Sums of squares based approximation algorithms for MAX-SAT

H. van Maaren, L. van Norden, and M.J.H. Heule [1]

Delft University of Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
Mekelweg 4, 2628 CD Delft, The Netherlands

Abstract

We investigate the Semidefinite Programming based Sums of squares (SOS) decomposition method, designed for global optimization of polynomials, in the context of the (Maximum) Satisfiability problem. To be specific, we examine the potential of this theory for providing tests for unsatisfiability and providing MAX-SAT upper bounds. We compare the SOS approach with existing upper bound and rounding techniques for the MAX-2-SAT case of Goemans and Williamson [12] and Feige and Goemans [10] and the MAX-3-SAT case of Karloff and Zwick [14], which are based on Semidefinite Programming as well. We prove that for each of these algorithms there is a SOS-based counterpart which provides upper bounds at least as tight, but observably tighter in particular cases. Also, we propose a new randomized rounding technique based on the optimal solution of the SOS Semidefinite Program which we experimentally compare with the appropriate existing rounding techniques. Further we investigate the implications to the decision variant SAT and compare experimental results with those yielded from the higher lifting approach of Anjos [1–3].

We give some impression of the fraction of so called unit constraints in the various SDP relaxations. From a mathematical viewpoint these constraints should be easily dealt with in an algorithmic setting, but seem hard to be avoided as extra constraints in a SDP setting. Finally we briefly indicate whether this work could have implications in finding counter examples to uncovered cases in Hilberts Positivstellensatz.

Key words: (Maximum) Satisfiability, Semidefinite Programming, sums of squares, approximation algorithms

[1] Supported by the Dutch Organization for Scientific Research (NWO) under grant 617.023.306

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 2, or $n = 2$ and the degree is 4. Despite these restrictive constraints, explicit counter examples for the non-covered cases are rare, although Blekherman [5] proved that there must be many of them. On the other side Parrilo [17] 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 MAX-SAT using the SOS approach which gives tighter bounds than the existing ones.

Before entering the specific MAX-SAT context we first explain the SOS formalism: 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 program (1).

A benefit of the approach above is the possibility of 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 finding a SOS decomposition. 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. Each polynomial that is non-negative on $\{-1, 1\}^n$ can be written as a SOS modulo the ideal $I_{\mathcal{B}}$ generated by the polynomials $x_1^2 - 1, \ldots, x_n^2 - 1$. This result is a special case of a theorem by Putinar [18]. 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. 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

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

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, $SP(x_1,\ldots,x_n)$ 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 formally. 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 decision method (in theory) for the question whether p can be written as a SOS using $M_1,\ldots,M_k$ as a basis of monomials. This decision method can be seen as a polynomial time decision method, only if a prescribed precision is specified. Testing (5) for feasibility, within a given precision with respect to some metric on matrix spaces, can be done in polynomial time, which in turn depends on the precision specified. The constraint $M^T S M = p$ in fact results in a set of linear constraints in the entries of the matrix S. This is illustrated in example 1. To find a lower bound on the minimum of $p(x_1,\ldots,x_n)$ we solve the program

$$\begin{aligned} & \max \alpha \\ \text{s.t. } & p - \alpha = M^T S M \\ & \alpha \in \mathbb{R}, S \succeq 0 \end{aligned} \tag{6}$$

Note that a $S \succeq 0$ has a Cholesky decomposition $S = L^T L$. 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.

Now we come to discuss the above in a MAX-SAT related context (for a general survey on the relations between Semidefinite Programming and satisfiability we refer to Anjos [3]). First we shall associate polynomials to CNF formulas: 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 minima of F_ϕ and $F_\phi^{\mathcal{B}}$ yield upper bounds for the MAX-SAT-solution.

The following two examples illustrate the construction of F_ϕ and $F_\phi^{\mathcal{B}}$ and the corresponding SDP's.

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

$$X \vee Y, \quad X \vee \neg Y, \quad \neg X$$

The polynomial F_ϕ is in this case

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

In order to attempt to find the maximal α such that $F_\phi - \alpha$ can be rewritten as a SOS it suffices to work with the monomial basis $M^T = (1, x, y, xy)$. The program we need to solve is

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

Let s_{ij} be the entry in S on row i and column j. When substituting F_ϕ, M and S program (7) becomes

$$
\begin{aligned}
&\max \alpha && (8)\\
&\text{s.t. } \frac{3}{8} + \frac{1}{4}x + \frac{3}{8}x^2 + \frac{1}{8}y^2 + \frac{1}{4}xy^2 + \frac{1}{8}x^2y^2 - \alpha\\
&= s_{11} + (s_{12}+s_{21})x + (s_{13}+s_{31})y + s_{22}x^2 + s_{33}y^2 + s_{44}x^2y^2\\
&+(s_{24}+s_{42})x^2y + (s_{43}+s_{34})xy^2 + (s_{14}+s_{41}+s_{23}+s_{32})xy\\
&\alpha \in \mathbb{R}, S \succeq 0
\end{aligned}
$$

The linear equalities for the s_{ij} are obtained by comparing the coefficients of the monomials on both sides of the equation. For example if we consider the monomial xy^2 we have the equality

$$s_{43} + s_{34} = \frac{1}{4}$$

and for the monomial xy we have

$$s_{14} + s_{41} + s_{23} + s_{32} = 0$$

SDP (8) has optimal solution $\alpha = \frac{1}{3}$, from which we may conclude that $2\frac{2}{3}$ is an upper bound for the MAX-SAT solution of ϕ. Notice that $F_\phi = \frac{1}{3} + \frac{3}{8}\left(x + \frac{1}{3}\right)^2 + \frac{1}{8}(xy - y)^2$, i.e. $F_\phi - \frac{1}{3}$ is a SOS. 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

$$X \vee Y \vee Z, \quad X \vee Y \vee \neg Z, \quad \neg Y \vee \neg T, \quad \neg X, \quad T$$

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

The Semidefinite Program (SDP)

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

can be rewritten using $x^2 \equiv y^2 \equiv z^2 \equiv t^2 \equiv 1$ modulo $I_{\mathcal{B}}$ as

$$\begin{aligned}
&\max \alpha && (11)\\
&\text{s.t. } \frac{3}{2} + \frac{1}{4}x - \frac{1}{4}t + \frac{1}{4}xy + \frac{1}{4}yt - \alpha \equiv \\
&s_{11} + s_{22} + s_{33} + s_{44} + (s_{12} + s_{21})x + (s_{13} + s_{31})y + (s_{14} + s_{41})t + \\
&(s_{23} + s_{32})xy + (s_{24} + s_{42})tx + (s_{43} + s_{34})yt \ \text{ modulo } \ I_{\mathcal{B}} \\
&\alpha \in \mathbb{R}, S \succeq 0
\end{aligned}$$

Program (11) gives output $\alpha = 0.793$, from which we may conclude that 4.207 is an upper bound for the MAX-SAT solution of ϕ. The SDP

$$\begin{aligned}
&\max \alpha && (12)\\
&\text{s.t.} F_{\phi}^{\mathcal{B}} - \alpha \equiv (1, x, y, t, xy, xt, yt) S (1, x, y, t, xy, xt, yt)^T \ \text{ modulo } \ I_{\mathcal{B}} \\
&\alpha \in \mathbb{R}, S \succeq 0
\end{aligned}$$

gives output $\alpha = 1$. Note that the second SDP gives a tighter upper bound, because the basis contains more monomials.

Below we formulate some properties of the polynomials F_{ϕ} and $F_{\phi}^{\mathcal{B}}$. Let m be the number of clauses and n the number of variables in CNF-formula 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 MAX-SAT 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

$$m - \max \alpha \tag{13}$$
$$s.t.\ \ F_\phi - \alpha \quad is\ a\ SOS$$
$$\alpha \in \mathbb{R}$$

and

$$m - \max \alpha \tag{14}$$
$$s.t.\ \ F_\phi^{\mathcal{B}} - \alpha \equiv M^T S M \ \ modulo\ \ I_{\mathcal{B}}$$
$$\alpha \in \mathbb{R}, S \succeq 0$$

give upper bounds for the MAX-SAT 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 nonnegative. 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 (14) is the basis for the search for MAX-SAT upper bounds in this paper. Theorems 1.5 and 1.8, program (13), could serve as a starting point 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 (13) must have $\alpha = 0$ for a satisfiable formula (F_ϕ is a SOS itself by construction). For an unsatisfiable formula ϕ, the optimal 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 give a counterexample but gives a proof of the unsatisfiability of the instance. We will report on some experiments and prove a theorem relating complexity issues to the existence of counterexamples of Hilbert's Positivstellensatz of a specific form in Section 8.

We close this section with a classification based on Theorem 1. Let ϕ be a CNF-formula. As we have seen before 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. No direct computational gain is to be expected here, since tractable computations with ideals presume working with a Gröbner basis. Establishing such a basis for the ideal I_ϕ would take in general a double exponential procedure. Nevertheless, we state the theorem as such, since it connects satisfiability to what could be called the algebraic concept of "Artinean rings".

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

At the end of this section we will define the notation of five monomial bases that are used throughout the remainder of this paper. $1, x_1, \ldots, x_n$ are contained in each of these bases. A product $x_i x_j$ occurs in the polynomial related to the CNF-formula (at least before adding terms with the same monomial) if X_i and X_j occur in a same clause. This makes these monomials probably good choices to include in the monomial basis.

Definition 1
- *M_{GW} is the monomial basis containing $1, x_1, \ldots, x_n$ (applicable for MAX-2-SAT).*
- *M_p is the monomial basis containing $1, x_1, \ldots, x_n$ and all monomials $x_i x_j$ for variables X_i and X_j appearing in a same clause (applicable for MAX-2-SAT and MAX-3-SAT).*
- *M_{ap} is the monomial basis containing $1, x_1, \ldots, x_n$ and monomials $x_i x_j$ for each pair of variables X_i and X_j (applicable for MAX-2-SAT and MAX-3-SAT).*
- *Monomial basis M_t contains $1, x_1, \ldots, x_n$ and the monomials $x_i x_j x_k$ such that X_i, X_j and X_k occur in a same clause (applicable for MAX-3-SAT).*
- *Monomial basis M_{pt} contains $1, x_1, \ldots, x_n$, all monomials $x_i x_j$ for variables X_i and X_j appearing in a same clause and all monomials $x_i x_j x_k$ such that X_i, X_j and X_k occur in a same clause (applicable for MAX-3-SAT).*

The corresponding SDP's with computation modulo $I_{\mathcal{B}}$ are called SOS_{GW}, SOS_p, SOS_{ap}, SOS_t and SOS_{pt} respectively. Note that when M_t is selected as monomial basis, monomials of degree two are obtained as products of variables and monomials of degree three modulo $I_{\mathcal{B}}$. Secondly, note that M_{GW} can only be used as monomial basis if all clauses have length smaller than or equal to two. If a CNF-formula ϕ contains a clause of length three, $F_\phi^{\mathcal{B}}$ probably contains a monomial of the form $x_i x_j x_k$ which does not occur in $M_{GW}^T S M_{GW}$.

In section 2 we prove that SOS_{GW} gives upper bounds equal to the ones obtained by the approach by Goemans and Williamson. Furthermore, we prove that adding monomials to the monomial basis has the same, and possibly stronger, effect as adding related valid inequalities to the SDP of Goemans and Williamson. In section 2.3 we prove that SOS_{ap} always finds an upper bound equal to the optimum for a class of problems having worst case known performance for the approach of Feige and Goemans. Experimental results on the quality of the different upper bounding techniques for randomly generated MAX-2-SAT instances are presented in section 2.4. Section 3 gives the computational complexities of the different approaches given a particular SDP algorithm (Sedumi [19]). In section 4 we present experimental results on random MAX-3-SAT and a proof showing that the upper bounds obtained by SOS_{pt} are at least as tight as the one obtained by the method by Karloff and Zwick . Section 5 experimentally compares SOS_t and SOS_{pt} with a relaxation by Anjos for proving unsatisfiability of 3-SAT instances. The new SOS-based rounding procedure used for obtaining lower bounds is described and experimentally investigated in section 6. In section 7 we give experimental evidence that in many cases a considerable fraction of the constraints in the SOS SDP's only fixes variables. Section 8 deals with the interrelationship between the complexity of solving SDP's and counterexamples to uncovered cases of Hilbert's Positivstellensatz stemming from unsatisfiable CNF-formulas.

2 SDP-based upper bounds for MAX-2-SAT

Although the SOS approach provides upper bounds for general MAX-SAT instances, 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 [12] and Feige and Goemans [10]. In this section, we consider SOS_{GW}.

Semidefinite Programming formulations come in pairs: the so-called primal and dual formulations, see for example [8] and [9]. Here we present a generic formulation of a primal Semidefinite problem P, and its dual program D. In the context of this paper, D and P give the same optimal value.

Consider the program P

$$
\begin{aligned}
&\min \operatorname{Tr}(CX) && (15)\\
&\text{s.t. } \operatorname{diag}(X) = e && (16)\\
&\operatorname{Tr}(A_j X) \geq 1, \quad j = 1, \ldots, k && (17)\\
&X \succeq 0
\end{aligned}
$$

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

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

$\text{diag}(X) = e$ 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

$$\max \sum_{i=1}^{p}\gamma_i + \sum_{j=1}^{k} y_j \tag{18}$$
$$\text{s.t.Diag}(\gamma) + \sum_{j=1}^{k} y_j A_j + U = C$$
$$U \succeq 0, y_j \geq 0$$

in which the y_j's are the dual variables corresponding to constraints (17) and the γ_i's the dual variables corresponding to constraint (16). U is a symmetric square matrix and $\text{Diag}(\gamma)$ 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_{GW} and Goemans-Williamson approach

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

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

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 inner products $v_i \bullet v_j$. They make $F_\phi^{\mathcal{B}}$ homogeneous by adding a dummy vector v_0 in order to make the linear terms in $F_\phi^{\mathcal{B}}$ quadratic as well. For example, $3x_i$ is replaced by $3(v_i \bullet v_0)$. 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

$$\min \hat{F}_\phi^{\mathcal{B}}(v_0, v_1, \ldots, v_n) \tag{20}$$
$$s.t. \ \|v_i\| = 1, v_i \in \mathbb{R}^{n+1}$$

To transform (20) to a Semidefinite Program, $v_i \bullet v_j$ is replaced by t_{ij}. Let b_{ij} be the coefficient of $(M_{GW})_i(M_{GW})_j$ 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 be 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.

Remark The use of v_0, apart from the fact that it is used to homogenize expressions, can also be used to give meaning to a rounding scheme and its logical function is used differently in different approaches. For instance Goemans and Williams take the value 1 to represent "false". In our rounding schemes discussed later we take the value 1 to represent "truth", which seems the approach taken by the majority of authors.

Consequently, (20) is equivalent to the following Semidefinite Program

$$\begin{aligned} & c_0 + \min \sum_{i=1}^{n+1} \sum_{j=1}^{n+1} f_{ij} t_{ij} \\ & \text{s.t. } \; t_{ii} = 1, T \succeq 0 \end{aligned} \tag{21}$$

or in matrix notation,

$$\begin{aligned} & c_0 + \min \operatorname{Tr}(M(F)T) \\ & \text{s.t. } \; \operatorname{diag}(T) = e, T \succeq 0 \end{aligned} \tag{22}$$

While Goemans and Williamson relax the input arguments of $F_\phi^{\mathcal{B}}$, the SOS-approach is a relaxation by replacing non-negativity by being a SOS. The next theorem proves that SOS_{GW} gives the same upper bound for MAX-2-SAT as program (22).

Theorem 4 *SOS_{GW} gives the same upper bound as the algorithm of Goemans and Williamson.*

PROOF. In the Goemans-Williamson SDP (22) we only have the constraints of type (16), and not of type (17). This implies that we have to deal only with the γ_j-variables. The size of the variable matrix T is $n + 1$.

The dual problem of the Goemans-Williamson-Semidefinite Program (22) is

$$\begin{aligned} & c_0 + \max \sum_{i=1}^{n+1} \gamma_i \\ & \text{s.t. } \; \operatorname{Diag}(\gamma) + U = M(F) \\ & U \succeq 0, \gamma_i \; \text{ free} \end{aligned} \tag{23}$$

We start the proof with the program

$$\begin{aligned} &\max \alpha \\ &\text{s.t. } F_\phi^{\mathcal{B}} - \alpha \equiv M^T S M \ \text{ modulo } \ I_{\mathcal{B}} \\ &S \succeq 0, \alpha \in \mathbb{R} \end{aligned} \tag{24}$$

with monomial basis $M = M_{GW} = ((M_{GW})_1, \ldots, (M_{GW})_{n+1}) = (1, x_1, \ldots, x_n)$ and prove that it is equal to (23). Let s_{ij} be the (i, j)-th element of matrix S.

We can reformulate program (24) as

$$\begin{aligned} &\max \alpha \\ &\text{s.t. } \sum_{i=1}^{n+1} \sum_{j=1}^{n+1} s_{ij} (M_{GW})_i (M_{GW})_j \equiv F_\phi^{\mathcal{B}} - \alpha \ \text{ modulo } \ I_{\mathcal{B}} \\ &S \succeq 0, \alpha \in \mathbb{R} \end{aligned} \tag{25}$$

Consider the constraint in program (25) for the coefficient of the constant. On the left hand side we have $\sum_{i=1}^{n+1} s_{ii}$ because $((M_{GW})_i)^2 \equiv 1$ modulo $I_{\mathcal{B}}$. On the right hand side we have $c_0 - \alpha$. This results in the equality

$$\alpha = c_0 - \sum_{i=1}^{n+1} s_{ii} \tag{26}$$

In the matrix formulation (27), on both left and right hand sides we have a matrix with on position (i, j), $i \neq j$, the coefficient of $(M_{GW})_i (M_{GW})_j$ using symmetry. Substituting (26) and using matrix notation we can reformulate (25) as

$$\begin{aligned} &c_0 + \max \sum_{i=1}^{n+1} -s_{ii} \\ &\text{s.t. } \ S - \operatorname{diag}(S) = M(F) \\ &S \succeq 0 \end{aligned} \tag{27}$$

Identifying γ_i with matrix entries $-s_{ii}$ and U with S, it is immediate that (27) is equivalent to (23).

Hence, we proved that (24) with monomial basis M_{GW} equals (23). It can be concluded that the Goemans-Williamson SDP and SOS_{GW} are dual problems providing the same upper bounds for MAX-2-SAT instances.

Still, there is something more to say about these two different approaches.

Program (21) has $\frac{1}{2}(n+1)(n+2)$ variables t_{ij} (not $(n+1)^2$ because T is symmetric).

In SOS_{GW} (14), each product of two different monomials yields a unique monomial. This means that each equality is of the form

$$s_{ij} + s_{ji} = c$$

for some constant c. For each pair $(i, j), i \neq j$ there is such an equality. Due to symmetry this implies that in fact only the diagonal elements are essentially variable, because all 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.

In the experiments we tried several Semidefinite Programming solvers like Sedumi [19], DSDP [4], CSDP [7] and SDPA [11], but none of them could fully benefit from this fact. However, we found that CSDP performed best on SDP's of the form (14). In section 7 we investigate how many constraints in the SOS approach are in fact 'unit' constraints.

2.2 Adding valid inequalities vs adding monomials

Feige and Goemans [10] 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 of 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

$$1 + x_i + x_j + x_i x_j \geq 0 \tag{28}$$

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 inequality $1 + t_{i,n+1} + t_{j,n+1} + t_{ij} \geq 0$ which is added to (21). All these inequalities can be added, 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 these valid inequalities compare with SOS_p. 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

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

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 (28) 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 Goemans-Williamson program (21) together with the four triangle inequalities of the form (28). The experiments in section 2.4 support this fact.

Feige and Goemans [10] further showed that adding for each triple of variables X_i, X_j and X_k to (21) the valid inequalities

$$\begin{aligned} &1 + t_{ij} + t_{ik} + t_{jk} \geq 0, \quad 1 - t_{ij} + t_{ik} - t_{jk} \geq 0 \\ &1 + t_{ij} - t_{ik} - t_{jk} \geq 0, \quad 1 - t_{ij} - t_{ik} + t_{jk} \geq 0 \end{aligned} \tag{29}$$

improves the tightness of the relaxation. Note that

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

Hence, the effect of adding the four inequalities (29) 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 5.

Finally, note that adding all inequalities of the form (29) 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 feature into a computational benefit as well.

It is clear that invoking a new monomial in our SOS approach increases the matrix size as well: it requires an extra row and an extra column. Hence the key question here is whether SDP software can be developed dealing with unit constraints (see section 7) efficiently. At this moment we cannot say that going from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$ really generates computational gain.

Theorem 5 *Adding monomials $x_i x_j$, $x_i x_k$ and $x_j x_k$ 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 (29) to the Goemans-Williamson SDP (21).*

PROOF. Without loss of generality we consider the triangle inequality

$$1 + x_1x_2 - x_1x_3 - x_2x_3 \geq 0 \tag{30}$$

In the notation of (21) 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

$$\begin{aligned} &\min \mathrm{Tr}(M(F)T) \\ &\text{s.t. } \mathrm{diag}(T) = e \\ &\mathrm{Tr}(AT) \geq 1 \\ &T \succeq 0 \end{aligned} \tag{31}$$

Assume that F is an homogeneous 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 (31) is the following

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

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

Program (32) can be reformulated as

$$\begin{aligned} &\max \gamma_1 + \gamma_2 + \gamma_3 + y \\ &\text{s.t. } M(F) - \mathrm{Diag}(\gamma) - yA \succeq 0 \\ &y \geq 0 \end{aligned} \tag{33}$$

Now suppose that $(\hat{\gamma}_1, \hat{\gamma}_2, \hat{\gamma}_3, \hat{y})$ is an optimal solution for (33). We will show that from this optimal solution a feasible solution for

$$\begin{aligned} &\max \alpha \\ &\text{s.t. } \; MSM^T \equiv F - \alpha \quad \text{modulo} \;\; I_{\mathcal{B}} \end{aligned} \tag{34}$$

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.

Program (34) with monomial basis M_1 can be reformulated as

$$\begin{aligned} &\max \left(-\sum_{i=1}^{4} s_{ii} \right) \\ &s_{12} + s_{21} + s_{34} + s_{43} = 2a \\ &s_{13} + s_{31} + s_{24} + s_{42} = 2b \\ &s_{23} + s_{32} + s_{14} + s_{41} = 2c \\ &S \succeq 0 \end{aligned} \tag{35}$$

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

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

with Δ the positive Semidefinite matrix

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

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 $Z + \frac{1}{2}\hat{y}\Delta \succeq 0$ because $Z \succeq 0$, $\Delta \succeq 0$ and $\hat{y} \geq 0$. The matrix $Z + \frac{1}{2}\hat{y}\Delta$

$$Z + \frac{1}{2}\hat{y}\Delta = \begin{pmatrix} -\hat{\gamma}_1 - \frac{1}{2}\hat{y} & a - \frac{1}{2}\hat{y} & b + \frac{1}{2}\hat{y} & -\frac{1}{2}\hat{y} \\ a - \frac{1}{2}\hat{y} & -\hat{\gamma}_2 - \frac{1}{2}\hat{y} & c + \frac{1}{2}\hat{y} & -\frac{1}{2}\hat{y} \\ b + \frac{1}{2}\hat{y} & c + \frac{1}{2}\hat{y} & -\hat{\gamma}_3 - \frac{1}{2}\hat{y} & \frac{1}{2}\hat{y} \\ -\frac{1}{2}\hat{y} & -\frac{1}{2}\hat{y} & \frac{1}{2}\hat{y} & \frac{1}{2}\hat{y} \end{pmatrix}$$

satisfies the constraints in (35) and $-\sum_{i=1}^{4} s_{ii} = \hat{\gamma}_1 + \hat{\gamma}_2 + \hat{\gamma}_3 + \hat{y}$.

Because the optimal solution of Goemans-Williamson SDP with Feige-Goemans valid inequalities equals a feasible solution of SOS_p, it can be concluded that the SOS approach gives at least as tight upper bounds.

At this point, we proved in this section that the upper bounds obtained by SOS_{GW} and the ones obtained by the approach of Goemans and Williamson are equal. Furthermore, we proved that SOS_p gives upper bounds that are at least as tight as the ones obtained by the Feige-Goemans approach with four valid inequalities for each pair of variables occurring in a same clause. Furthermore, we showed that SOS_{ap} provides upper bounds at least as tight as the Feige-Goemans approach with all inequalities of the form (29) added.

2.3 SOS_{ap} on worst known case for Feige-Goemans

The CNF formula of ϕ_{FGn} 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 [10] present ϕ_{FG5} 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-numbered variables to false. It is not possible to satisfy all clauses for odd n. In this section, we show that the SOS_{ap} finds the optimal MAX-SAT solution of ϕ_{FGn}.

Theorem 6 *Let n be an odd number. The polynomial $F_{\phi_n}^{\mathcal{B}} - 1$ with*

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

is a sum of squares if we choose as monomial basis M_{ap}.

PROOF. As initial step we start with ϕ_{FG3}. The polynomial $F_{\phi_3}^{\mathcal{B}}$ is

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

Define $F_3(x) = 1 + x_1x_2 + x_2x_3 + x_3x_1$. $F^{\mathcal{B}}_{\phi_3} - 1$ is a sum of squares modulo $I_{\mathcal{B}}$, because

$$\frac{1}{2}\left(\frac{1}{2}F_3(x)\right)^2 \equiv \frac{1}{2}F_3(x) \text{ modulo } I_{\mathcal{B}}$$

We use this fact to prove by induction that $F^{\mathcal{B}}_{\phi_n} - 1$ is a sum of squares relative to the monomial basis considered. Assume that the polynomial $F^{\mathcal{B}}_{\phi_{n-2}} - 1$ related to $\phi_{FG(n-2)}$ 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 SOS_{ap} 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 SOS_{ap} solves (36) to optimality.

2.4 Experimental results on random MAX-2-SAT upper bounds

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 (28) are added.

Variant FG_{ap}**:** For *each* pair of variables the four inequalities of the type (28) 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 (29) added.

We compare the upper bounds resulting from these variants with the upper bounds obtained from the Semidefinite Program (14) with monomial basis M_p. 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 present results on small-scale problems only in this section, because solving the SDP's of SOS_p takes a lot of time with the SDP-solvers currently available.

Table 1
10 variables, random MAX-2-SAT, bound ratios

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

In initial experiments we used a set of 900 randomly generated instances with 10 variables and different clause-variable densities d. The (clause-variable) density is the number of clauses divided by the number of variables. 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 MAX-SAT solution divided by the upper bound found. The bound ratio can be seen as a size independent measure for the quality of the upper bounds.

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. The first column indicates the density, the second column the bound ratio for SOS_p, the third column gives the results of SOS_{ap}, the fourth gives the Goemans-Williamson bound ratio, the fifth gives the bound ratio of variant FG_m, the next the bound ratio of variant FG_{4p}, then the bound ratio of FG_{ap} and the last column gives the bound ratio 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 individual instance involved in our experiments. For the selected set of instances, SOS_p turns out to be almost always, except for one instance, at least as good as the best Feige-Goemans variant FG_{pt} while this is not forced by Theorem 5. In these experiments with MAX-2-SAT instances with 10 variables, the SDP's of each variant are solved by Sedumi [19]. Table 2 gives the same type of results for instances with 25 variables but only for the methods that are most relevant and computationally not too expensive. For the instances with 25 variables GW and FG_{ap} are solved by Sedumi. The SDP's of SOS_p are solved by CSDP [7], because this solver is faster, more accurate and uses less memory when solving the SDP's of SOS_p .

Table 2
25 variables, random MAX-2-SAT, bound ratios

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

The detailed results on the experiment with instances with 25 variables are given in Tables 3. In these tables, the first column gives the density, the second gives the optimal MAX-SAT solution. In the third column is indicated the number of instances for which the upper bound found by SOS_p equals the optimal MAX-SAT solution, in which a small numerical error of at most 10^{-5} is allowed. Columns four and five respectively give the average and minimal bound ratio for SOS_p. The last is an indication for the worst case result on the test set. Columns 6, 7 and 8 give similar results for FG_{ap}. The last column mentions the number of instances with the MAX-SAT solution equal to the number in the second column. From Tables 3 is clear that SOS_p finds an upper bound equal to the optimal MAX-SAT solution considerably more often than FG_{ap}.

3 Complexity of different approaches wrt short step Semidefinite optimization algorithms

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 related computational complexity of the different upper bound variants in

Table 3
25 variables, MAX-2-SAT, bound ratios and frequencies of finding optimum

d	OPT	#E SOS_p	Av SOS_p	Min SOS_p	#E FG_{ap}	Av FG_{ap}	Min FG_{ap}	#I
1.5	36	2	0.996438	0.989314	0	0.984777	0.980163	3
	37	27	0.999780	0.994716	8	0.996815	0.990767	29
2.0	47	5	1	1	1	0.990664	0.987512	5
	48	24	0.999098	0.994749	5	0.995868	0.988283	30
	49	45	0.999896	0.995205	27	0.998918	0.989843	46
2.5	57	0	0.994421	0.994421	0	0.990546	0.990546	1
	58	0	0.999722	0.999722	0	0.990159	0.990159	1
	59	10	0.998753	0.992482	0	0.993466	0.986547	15
	60	20	0.999471	0.996081	3	0.996686	0.991281	24
	61	41	0.999901	0.995850	26	0.999238	0.994349	42
	62	15	1	1	13	0.999833	0.997988	15
3.0	69	10	0.999827	0.998096	0	0.994511	0.990119	11
	70	16	0.999802	0.996639	1	0.997208	0.990343	17
	71	28	0.999909	0.997371	12	0.998443	0.992015	29
	72	29	0.999951	0.998523	17	0.999285	0.996044	30
	73	10	1	1	9	0.999829	0.998291	10
	74	3	1	1	3	1	1	3
3.5	78	0	0.993485	0.993485	0	0.983817	0.983817	1
	79	1	1	1	0	0.990783	0.990783	1
	80	3	0.999334	0.998357	0	0.994561	0.991726	6
	81	20	0.999460	0.996669	2	0.995872	0.989009	26
	82	22	0.999734	0.996179	4	0.997746	0.993598	24
	83	19	1	1	6	0.998670	0.995682	19
	84	13	1	1	12	0.999866	0.998259	13
	85	8	1	1	7	0.99985	0.999879	8
	86	2	1	1	1	0.999274	0.998544	2
4.0	90	5	0.999629	0.997774	0	0.992851	0.990062	6
	91	15	0.999755	0.997380	1	0.995239	0.987905	17
	92	28	0.999964	0.998966	6	0.998158	0.992194	29
	93	21	0.999919	0.998221	5	0.998694	0.995554	22
	94	12	1	1	6	0.999628	0.997897	12
	95	10	1	1	7	0.999941	0.999478	10
	96	4	1	1	4	1	1	4
4.5	100	0	0.998557	0.997645	0	0.990234	0.989394	3
	101	4	0.999535	0.998409	0	0.993362	0.989456	7
	102	18	0.999878	0.997691	0	0.995885	0.989874	19
	103	23	1	1	7	0.998316	0.994168	23
	104	15	0.999882	0.998112	3	0.998851	0.994588	16
	105	15	1	1	6	0.998962	0.997240	15
	106	13	1	1	12	0.999903	0.998735	13
	107	4	1	1	4	1	1	4
5.0	110	1	1	1	0	0.994511	0.994511	1
	111	8	0.999885	0.998964	0	0.994919	0.992738	9
	112	14	1	1	1	0.997640	0.994068	14
	113	24	1	1	2	0.997601	0.993335	24
	114	19	0.999999	0.999973	7	0.998725	0.993769	20
	115	11	1	1	5	0.999383	0.997228	11
	116	14	1	1	8	0.999580	0.996685	14
	117	4	1	1	3	0.999897	0.999586	4
	118	3	1	1	3	1	1	3

this section. Let ϕ be a CNF formula with n variables and m clauses.

In the Goemans-Williamson approach the size of the variable matrix is $n+1$ and the number of constraints is also $n+1$ implying a computational complexity

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

The variant FG_m also has a variable matrix of size $n+1$, but the number of constraints is now $n+1+m$ yielding

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

Hence, for fixed clause-variable density FG_m has complexity $\mathcal{O}(n^2\sqrt{n})$. Using that the number of pairs of variables occurring in a same clause is of the same order as the number of clauses for 2-SAT, we have

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

In the SDP of FG_{ap} the size of the variable matrix is $n+1$ and the number of constraints $n+1+4p$ with the number of pairs p equal to $\frac{1}{2}n(n-1)$. For the computational complexity $CP_{FG_{ap}}$ we have

$$CP_{FG_{ap}} = \mathcal{O}((4n^2+2n+2)\sqrt{n}) = \mathcal{O}\left(n^2\sqrt{n}\right)$$

The variable matrix of FG_{pt} is of size $n+1$, and the number of constraints equals $n+1+2n(n-1)+\frac{2}{3}n(n-1)(n-2)$ giving

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

SOS_p is applicable for both MAX-2-SAT and MAX-3-SAT, but with a slightly different complexity. The size of the variable matrix equals the size $|M|$ of the monomial basis. For MAX-2-SAT the size of the basis is smaller than or equal to $1+n+m$. The number of constraints is of $\mathcal{O}(|M|^2)$. Hence for MAX-2-SAT the computational complexity CP_{SOSp} is

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

Note that for instances with a fixed clause-variable density d, i.e. $m = dn$, the computational complexity of FG_{ap} and SOS_p is of the same order for MAX-2-SAT. For MAX-3-SAT the size of the monomial basis is at most $1+n+3m$ giving the complexity

$$CP_{SOSp} = \mathcal{O}\left((n+3m)^2\sqrt{n+3m}\right)$$

Finally, SOS_{ap} has variable matrix size $n+1+\frac{1}{2}n(n-1)$ and $n+1+\frac{1}{2}n(n-1)+\frac{1}{6}n(n-1)(n-2)$ constraints. The number of constraints is obtained by observing that in $M^TSM = F_\phi - \alpha$ there are linear constraints for the constant, each single variable, each pair of variables and each triple of variables. Because the variable matrix is of size $\mathcal{O}(n^2)$ the complexity is

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

The variants SOS_t and SOS_{pt}, when applied to MAX-3-SAT, have the following complexities

$$CP_{SOSt} = \mathcal{O}\left((n+m)^2\sqrt{n+m}\right)$$
$$CP_{SOSpt} = \mathcal{O}\left((n+4m)^2\sqrt{n+4m}\right)$$

4 SDP-based upper bounds for MAX-3-SAT

In contrary to the Goemans-Williamson and Feige-Goemans approaches the SOS-approach is directly applicable for MAX-3-SAT. Karloff and Zwick [14] present an algorithm based on Semidefinite Programming that guarantees a 7/8-approximation of MAX-3-SAT. Karloff and Zwick [14] prove that this is the case for satisfiable instances and provide strong evidence that it is also the case for unsatisfiable MAX-3-SAT instances. Zwick [20] completes the proof that the method is also a 7/8-approximation for unsatisfiable instances. Håstad [13] proved that for any $\epsilon > 0$ there does not exist a polynomial time $(\frac{7}{8}+\epsilon)$-approximation algorithm for MAX-3-SAT unless $\mathcal{P} = \mathcal{NP}$. This result implies that the algorithm by Karloff and Zwick is as tight as possible for the complete class of MAX-3-SAT instances.

We will start with a short description of the Semidefinite Program of Karloff and Zwick for unweighted MAX-3-SAT. Literals are numbered from 1 to $2n$ in the order $X_1, \ldots, X_n, \neg X_1, \ldots, \neg X_n$. Let z_{ijk} be a Boolean variable being 1 if the clause with literals i, j and k is satisfied and 0 otherwise. $v_1, \ldots, v_n$ are vectors in $\mathbb{R}^{n+1}$ corresponding to the literals $X_1, \ldots, X_n$, and $v_{n+1}, \ldots, v_{2n}$ correspond to the literals $\neg X_1, \ldots, \neg X_n$. v_0 is a vector corresponding to FALSE. The program presented by Karloff and Zwick [14] is

$$\max \sum_{i,j,k} z_{ijk} \tag{40}$$
$$z_{ijk} \leq \frac{4-(v_0+v_i)\cdot(v_j+v_k)}{4} \tag{41}$$
$$z_{ijk} \leq \frac{4-(v_0+v_j)\cdot(v_i+v_k)}{4} \tag{42}$$
$$z_{ijk} \leq \frac{4-(v_0+v_k)\cdot(v_i+v_j)}{4} \tag{43}$$
$$z_{ijk} \leq 1 \tag{44}$$
$$v_{n+i} = -v_i \tag{45}$$

Just like in the Goemans-Williamson approach inproducts $v_i v_j$ are replaced by variables t_{ij} to obtain the SDP. The sum in (40) is taken over all i, j, k such that there is a clause with literals i, j and k. Constraint (45) implies that v_i corresponding to X_i must be the opposite of v_{n+i} corresponding to $\neg X_i$. It is easy to check that one of (41) to (43) forces z_{ijk} to 0 if the vectors take values corresponding to an assignment that does not satisfy the clause with literal i, j and k.

In this section, we prove that SOS_{pt} gives at least as tight upper bounds as the approach of Karloff and Zwick and present experimental results on a set of randomly generated MAX-3-SAT instances.

4.1 Karloff-Zwick inequalities vs monomials in SOS_{pt}

Theorem 7 *Each constraint of the type (41), (42) or (43) in the SDP of Karloff and Zwick can be represented as an inequality that states that a sum of squares is non-negative with respect to the monomial basis*

$$M = \{1, x_i, x_j, x_k, x_i x_j, x_i x_k, x_j x_k, x_i x_j x_k\}$$

PROOF. In the SDP of Karloff and Zwick the z_{ijk} are variables having the interpretation of being 1 if the corresponding clause is satisfied and 0 if not.

For the clause $X_i \vee X_j \vee X_k$ the expression

$$1 - \frac{1}{8}(1 - x_i)(1 - x_j)(1 - x_k) \tag{46}$$

is equivalent to z_{ijk} in the sense that is also 1 if the clause is satisfied and 0 otherwise. Inequality (41) is for this clause equal to

$$1 - \frac{1}{4}(-1 + x_i)(x_j + x_k) \geq 1 - \frac{1}{8}(1 - x_i)(1 - x_j)(1 - x_k)$$

which is equivalent to

$$(1 - x_i)(1 - x_j)(1 - x_k) + 2(1 - x_i)(x_j + x_k) \geq 0$$

This can be simplified to

$$(1 - x_i)(1 + x_j + x_k + x_j x_k) \geq 0.$$

Hence we have to show that $(1-x_i)(1+x_j+x_k+x_jx_k)$ is a sum of squares modulo $I_{\mathcal{B}}$ invoking the designated monomials. This can be seen by

$$
\begin{aligned}
&(1-x_i)(1+x_j+x_k+x_jx_k) \equiv \qquad (47)\\
&\frac{1}{8}\left(1-x_i+x_j+x_k-x_ix_j-x_ix_k+x_jx_k-x_ix_jx_k\right)^2 \text{ modulo } I_{\mathcal{B}}
\end{aligned}
$$

The other constraints (42) and (43) can be dealt with analogously. This completes the proof.

The next corollary is a consequence of the above theorem. The reasoning is similar as in the proof of Theorem 5.

Corollary 1 *SOS_{pt} gives at least as tight upper bounds as the approach of Karloff and Zwick.*

4.2 Experimental comparison on random MAX-3-SAT upper bounds

We would like to compare the upper bound obtained by SOS_t, SOS_p and SOS_{pt} with the upper bound found by the Karloff-Zwick Semidefinite Program on randomly generated MAX-3-SAT instances with 10 and 15 variables and different densities. For the selected instances the upper bound obtained by the Karloff-Zwick SDP is always equal to the number of clauses. Hence, in the remainder we will present only the results of SOS_t, SOS_p and SOS_{pt}.

In Table 4 the first column gives the number of variables, the second column the density, the third column the bound ratio for SOS_{pt}. Column 4 gives the number of instances for which the upper bound of SOS_{pt} equals the optimal MAX-SAT solution up to a given precision. Columns 5 to 8 give similar results for SOS_p and SOS_t. The last column gives the number of instances used. From Table 4 it is clear that SOS_{pt} comes very close to the optimal MAX-SAT solution for the instances of the size considered in the experiment. In fact, in most cases the SOS based upper bound equals the MAX-SAT solution value.

5 Experimental results of SOS_t and SOS_{pt} on (the decision variant of) 3-SAT

The SOS approach is not only useful as a method to approximate MAX-SAT, but can also be used to prove the unsatisfiability of an instance. If it returns an upper bound smaller than the number of clauses (minus some small value γ),

Table 4
10 and 15 variables MAX-3-SAT, different SOS upper bounds of unsat instances

n	d	SOS_{pt}	#E	SOS_p	#E	SOS_t	#E	#I
10	4.0	0.99997691	17	0.999486	9	0.988574	0	18
	5.0	1	56	0.999871	40	0.993319	1	56
	6.0	1	84	0.999987	75	0.995770	10	84
	7.0	1	98	0.999977	94	0.997125	23	98
	8.0	1	99	0.999998	98	0.997873	30	99
	9.0	1	100	0.999997	99	0.997672	23	100
15	4.0	0.99997251	9	0.999288	2	0.985297	0	10
	4.5	1	32	0.999806	15	0.989574	0	32
	5.0	0.99999802	55	0.999934	41	0.993318	0	58
	6.0	1	92	0.999984	83	0.996899	8	92
	7.0	0.99999959	99	0.99999902	98	0.997871	18	100
	8.0	0.99999954	98	0.99999606	96	0.998280	18	100
	9.0	0.99999677	99	0.99999623	98	0.998556	30	100

the instance is definitely unsatisfiable. The value γ is necessary to compensate for numerical imprecisions. In this section we use a set of randomly generated 3-SAT instances with 15 and 20 variables and densities varying between 4.0 and 5.0, 100 instances for each size and density.

For each of the unsatisfiable instances in the set of selected instances, we computed the upper bound for SOS_t. Secondly, we computed the upper bound of SOS_{pt}. We selected besides SOS_{pt} also SOS_t because SOS_t can be solved much faster than SOS_{pt} . Based on these upper bounds we can count the number of instances that is proved to be unsatisfiable, i.e. the number of instances for which the upper bound is smaller than the number of clauses (minus γ).

Anjos [2] proposed a new SDP relaxation for SAT, which significantly improved on earlier SDP relaxations and can prove that a CNF-formula is unsatisfiable for formulas containing clauses of any length. Higher liftings are used to obtain the relaxation. The experiments in his paper show that for instances with up to 260 variables and 400 clauses satisfying assignments or proofs of unsatisfiability can be obtained. We compare the SOS approach with this rank-3-SDP relaxation [2] for satisfiability on a set of unsatisfiable instances. The method by Anjos proves unsatisfiability by proving infeasibility

[2] We thank Miguel Anjos for providing us with the code of his relaxation

of his SDP, which we will describe below.

Let P be the set containing $1, x_1, \ldots, x_n$, all $x_i x_j$ such that variables X_i and X_j occur in a same clause and all $x_i x_j x_k$ such that variables X_i, X_j and X_k occur in a same clause. Let v be a column vector containing these monomials and $Y = vv^T$. Let I denote the set of indices of the variables in a monomial. $Y_{\emptyset,I} = \prod_{i \in I} x_i$. Define s_{ij} to be 1 if X_j is contained in clause i and -1 if $\neg X_j$ is contained in clause j. The rank-3-relaxation is to find a symmetric positive semi-definite $|P| \times |P|$ matrix Y with ones on the diagonal satisfying

$$s_{j1} Y_{\emptyset,\{i_1\}} + s_{j2} Y_{\emptyset,\{i_2\}} - s_{j1} s_{j2} Y_{\emptyset,\{i_1,i_2\}} = 1 \tag{48}$$

for each clause j of length two, with i_1 and i_2 the indices of the variables in clause j and

$$\begin{aligned} s_{j1} Y_{\emptyset,\{i_1\}} + s_{j2} Y_{\emptyset,\{i_2\}} + s_{j3} Y_{\emptyset,\{i_3\}} - s_{j1} s_{j2} Y_{\emptyset,\{i_1,i_2\}} - s_{j1} s_{j3} Y_{\emptyset,\{i_1,i_3\}} \\ - s_{j2} s_{j3} Y_{\emptyset,\{i_2,i_3\}} + s_{j1} s_{j2} s_{j3} Y_{\emptyset,\{i_1,i_2,i_3\}} = 1 \end{aligned} \tag{49}$$

for each clause j of length three, with i_1, i_2 and i_3 the indices of the variables in clause j, and furthermore the constraints

$$Y_{\emptyset,I_1} = Y_{I_2,I_3}, Y_{\emptyset,I_2} = Y_{I_1,I_3}, Y_{\emptyset,I_3} = Y_{I_1,I_2} \tag{50}$$

for each I_1, I_2 and I_3 such that I_1, I_2 and I_3 are contained in the set of indices of variables in some clause j and $I_1 \triangle I_2 = I_3$(i.e. I_3 contains all elements that are in I_1 or I_2 but not in both). Hence, the SDP of Anjos adds a new variable (to be interpreted as Boolean) for each 'clause pair' and each 'clause triple'. Note that the size of the matrices in the Anjos' SDP and the size of the matrices in M_{pt} are identical. Also the computational complexity is of the same order as the one of SOS_{pt}.

Table 5 shows the results of the three approaches on the set of instances with 15 variables. The first column gives the density, the second one the number of instances out of the 100 generated instances that are unsatisfiable. The third, fourth and fifth column give respectively the number of instances proved unsatisfiable by SOS_t, SOS_{pt} and the rank-3-relaxation of Anjos. Table 6 shows the same type of results for the instances with 20 variables.

From Tables 5 and 6 we might conclude that SOS_{pt} is able to prove unsatisfiability up to larger sizes than the rank-3-relaxation of Anjos and that the ability of SOS_t to prove unsatisfiability breaks down early. Now we want to give some impression about the computation times involved. Unfortunately, we were not able to run the rank-3-relaxation under the same computational environment as SOS_t and SOS_{pt}, because it involves MATLAB routines. Hence, the figures

Table 5
15 variables 3-SAT

d	#unsats	SOS_t	SOS_{pt}	$A3$
4.0	10	6	10	10
4.1	17	11	17	17
4.2	19	11	19	19
4.3	24	15	24	24
4.4	24	20	24	24
4.5	32	28	32	32
4.6	35	32	35	35
4.7	40	35	40	40
4.8	45	43	45	45
4.9	54	51	54	54
5.0	58	54	58	58

Table 6
20 variables 3-SAT

d	#unsats	SOS_t	SOS_{pt}	$A3$
4.0	19	0	19	7
4.1	25	0	25	12
4.2	30	0	30	19
4.3	36	0	36	22

below only give a brief indication of the trade-off between quality and computational effort. For the instances with 15 variables and density 4.3, SOS_t needs on average 35.1 seconds, SOS_{pt} 549.9 seconds and the rank-3-relaxation by Anjos needs on average 5.8 seconds.

As we have compared in the above Anjos 3 relaxation with our SOS_{pt} approach it must be said that comparison of Anjos 2 [1] and SOS_t could have made sense as well. They compare naturally in that sense that Anjos 2 introduces an extra "Boolean" variable for each product of three original Boolean variables, while SOS_t introduces a new monomial for each triple of original Boolean variables. Both approaches can be characterized as "clause wise" extensions, since both look at triples of variables appearing in a same clause. Similarly, SOS_p could have been compared with an Anjos type of relaxation where to each pair of original Boolean variables an extra "Boolean" is introduced. We restricted ourselves however to the experiments as described above, simply because computational times involved are far from "neglecting".

6 Rounding procedures based on SOS_p and SOS_t

Goemans and Williamson present in their paper a rounding procedure for obtaining assignments that give a lower bound on the MAX-SAT solution. The performance guarantee obtained by this rounding procedure and the upper bound obtained by their SDP is 0.87856. Feige-Goemans [10] improve on this with their approach having a performance guarantee of 0.931. Lewin, Livnat and Zwick present a skewed rounding procedure yielding a performance guarantee of 0.940. In this section we present a rounding procedure for finding lower bounds based on the solutions of SOS_p and SOS_t.

We describe our rounding procedure for SOS_p, but the procedure can be generalized to any monomial basis. This SOS-rounding takes as input the optimal solution S of SOS_p. It can be shown that such an optimal S has an eigenvalue 0. Next the procedure determines an orthogonal basis of eigenvectors $V_1, \ldots, V_N$ of the optimal matrix S corresponding to eigenvalue 0. These eigenvectors might be algebraically inconsistent. With this we mean that for example the entries corresponding to x_1 and x_2 might be positive while the entry corresponding to $x_1 x_2$ might be negative.

The eigenvectors corresponding to eigenvalue 0 are the most relevant vectors because vectors with eigenvalue 0 correspond with solutions: if $\hat{V}$ is an eigenvector of S corresponding to eigenvalue 0, $M^T S M = 0$ if M_k is replaced by the numerical value of $\hat{V}_k$. If such an eigenvector is algebraically consistent and Boolean it realizes the optimal MAX-SAT solution.

Next, we come to the description of our randomized rounding procedure. A random point $\lambda = (\lambda_1, \ldots, \lambda_N)$ on the N-dimensional unit sphere is generated uniformly. We use the method by Knuth for generating uniformly distributed random points on the N-dimensional unit sphere [15] which uses standard normal distributed variables. The *sign* operator on a number x is defined to return 1 if $x \geq 0$ and -1 otherwise. Let P be defined as

$$P = \sum_{i=1}^{N} \lambda_i V_i. \tag{51}$$

The sign operator on a vector is defined as applying the sign operator on each individual entry. Let $SP = sign(P)$. $SP_2, \ldots, SP_{n+1}$ give an assignment for the variables $X_1, \ldots, X_n$ in the CNF-formula. The entry SP_1 corresponds to the first monomial in the monomial basis, being 1. Therefore, if $SP_1 = -1$ we reverse the assignment of the variables.

It can be proved that the above rounding procedure gives identical results as the rounding procedure of Goemans-Williamson, if the monomial basis M_{GW} is used [3]. Preliminary experiments showed that using the optimal matrix of SOS_p and the rounding procedure with a uniformly distributed λ does not necessarily improve on the Goemans-Williamson approach. The reason for this is probably that the eigenvectors obtained are algebraically inconsistent (apart from the fact that their entries are not necessarily Boolean). This might influence the quality of the rounding procedure negatively.

The idea to improve on this rounding procedure is that we want to try to get the entries in P corresponding to the products of variables as small as possible, to minimize the influence of the algebraic inconsistency as much as possible. Let V be the matrix containing $V_1, \dots, V_N$ as columns and p be the number of monomials of the form $x_i x_j$ in the monomial basis. The $p \times N$ matrix B is the matrix containing the rows $n+2$ to $n+1+p$ of V corresponding to the monomials of degree 2. We want to find vectors λ such that the last p entries in P are relatively small.

Therefore, we start with computing an orthogonal basis of eigenvectors $U_1, \dots, U_N$ and eigenvalues $\mu_1, \dots, \mu_N$ of the matrix $B^T B$. Let $\mu_1 \leq \mu_2 \leq \cdots \leq \mu_N$. We have $B^T B U_i = \mu_i U_i$. Hence, we have

$$\|BU_i\|^2 = (BU_i)^T (BU_i) = U_i^T B^T B U_i = \mu_i U_i^T U_i = \mu_i \|U_i\|^2 = \mu_i$$

and can conclude $\|BU_i\| = \sqrt{\mu_i}$. The goal is that we want to find a λ such that $B\lambda$ is small. We uniformly generate a random vector w on the N-dimensional sphere. For each $i = 1, \dots, N$ we determine a so-called scaling factors ξ_i. This results in a new skewed vector $\tilde{w}$ with $\tilde{w}_i = w_i \xi_i$. The λ from the linear combination (51) is now taken as

$$\lambda = \sum_{k=1}^{N} \tilde{w}_k U_k.$$

Note that

$$\|B\lambda\| \leq \sum_{k=1}^{n} \tilde{w}_k \sqrt{\mu_k} = \sum_{k=1}^{N} \xi_k w_k \sqrt{\mu_k}$$

In the experiments we used scaling factors ξ_i such that $\xi_i \leq \xi_j$ if $i < j$. For i with μ_i relatively large we want to keep ξ_i small in order to keep $\|B\lambda\|$ small.

6.1 Skewed rounding on MAX-2-SAT instances based on SOS_p

For the experiments in this section we generated 100 instances with each of the following number of variables and densities: 40 variables and density 3.0,

[3] We thank Etienne de Klerk for noticing this

35 variables and densities 3.0 and 5.0, 30 variables and densities 3.0, 5.0 and 7.0 and 25 variables and densities 3.0, 5.0, 7.0 and 9.0. We selected instances of this size because the corresponding SDP's can be solved in reasonable time with current SDP solvers. For each of the instances we solve SOS_p and apply SOS_p-based rounding with a uniform random vector and five weighted rounding procedures as described above. The three scaling factors we use are ρ_i^2, ρ_i^5, ρ_i^{15}, ρ_i^N with $\rho_i = (1-(\mu_i-\mu_1))$. Also we investigate scaling factors $2^{-(i-1)}$. In the first four the scaling is dependent on the size of the eigenvalues of B^TB. The last factor depends only on the ranking of the eigenvalues. Note that the first four are increasingly steeper. ρ_i^5 is selected for its good performance in preliminary experiments and the others are selected to study the effect of the 'steepness' of the scaling factors. Scaling factors ρ_i^5 seems to offer a good balance between the influence of the eigenvectors corresponding to small and larger eigenvalues. A much steeper function for the scaling factors is not very desirable because the vectors with larger eigenvalues get a scaling factor that is nearly zero and hardly contribute to the linear combination. The λ-vectors obtained are much more similar to each other and consequently the number of different possible assignments found is considerably reduced. As a consequence the chance of finding an optimal solution is lower. A flatter function like for example ρ_i^2, yields a larger diversity of λ-vectors and assignments but the number of times the (almost) optimal assignments are found is relatively low. For instance-wise comparison we implemented the Goemans-Williamson rounding on the Goemans-Williamson SDP. Each of the rounding procedures is run 1000 times.

Let S_R be the rounding procedure with a uniform random vector λ without any scaling.

Table 7 gives the average over the instances of the number of the times the optimum is found in the 1000 tries. This gives an indication for the number of tries that is necessary to find the optimum with high probability. Table 7 shows that the rounding procedures based on SOS_p find the optimum on average in about four times as much out of 1000 tries than the approach of Goemans and Williamson.

Next, we define the observed performance guarantee of the procedures based on SOS_p for a particular instance as the average number of clauses satisfied in a try divided by the upper bound obtained by SOS_p. The observed performance guarantee of the Goemans-Williamson approach for an instance is defined as the average number clauses satisfied in a try divided by the upper bound obtained by the Goemans-Williamson approach. Table 8 gives for each set of instances the average over the instances of the observed performance guarantees.

Figure 1 shows the performance guarantees and the results of the Goemans-

Table 7
MAX-2-SAT, rounding on instances with different sizes and densities, frequencies of finding the optimum

n	d	S_R	ρ_i^2	ρ_i^5	ρ_i^{15}	ρ_i^N	$2^{-(i-1)}$	GW
25	3.0	504.26	562.66	643.72	835.16	749.41	787.17	287.04
25	5.0	544.65	567.7	609.92	756.12	662.17	677.11	236.32
25	7.0	630.55	645.64	668.18	753.04	689.81	821.34	196.39
25	9.0	749.62	757.55	769.40	821.81	781.94	871.82	175.51
30	3.0	383.09	429.26	503.82	661.97	578.46	655.95	209.06
30	5.0	495.13	513.21	522.98	669.68	588.05	731.90	166.45
30	7.0	624.12	635.49	651.34	712.95	704.30	800.34	131.95
35	3.0	333.64	384.02	471.09	652.95	586.54	642.95	188.98
35	5.0	515.09	531.57	559.19	662.64	600.75	733.81	122.49
40	3.0	250.94	287.85	356.33	530.83	484.63	533.27	143.98

Table 8
MAX-2-SAT, rounding on instances with different sizes and densities, observed performance guarantee

n	d	S_R	ρ_i^2	ρ_i^5	ρ_i^{15}	ρ_i^N	$2^{-(i-1)}$	GW
25	3.0	0.979	0.985	0.991	0.996	0.995	0.995	0.954
25	5.0	0.984	0.987	0.991	0.996	0.994	0.994	0.963
25	7.0	0.993	0.994	0.995	0.997	0.996	0.998	0.968
25	9.0	0.995	0.996	0.996	0.998	0.997	0.998	0.971
30	3.0	0.967	0.977	0.984	0.989	0.988	0.989	0.951
30	5.0	0.983	0.986	0.989	0.993	0.992	0.995	0.961
30	7.0	0.992	0.993	0.994	0.996	0.996	0.998	0.966
35	3.0	0.967	0.977	0.984	0.989	0.988	0.990	0.951
35	5.0	0.985	0.987	0.990	0.994	0.993	0.995	0.960
40	3.0	0.953	0.967	0.977	0.985	0.986	0.985	0.949

Williamson approach and the SOS_p approach for a typical random 2-SAT instance with 25 variables and density 9.0. From right to left the vertical lines give the optimum solution, which can be found for instances of this size by the algorithm of Borchers and Furman [6], the performance guarantee of the approach of Lewin, Livnat and Zwick [16], the performance guarantee of the method of Feige and Goemans [10] and the performance guarantee of the Goemans-Williamson approach [12]. The leftmost non-vertical graph represents the performance of the Goemans-Williamson rounding for this particular instance. Each point (x, y) of the graph indicates the fraction y of the tries for which the assignment found satisfies more than x clauses. The rightmost non-vertical graph gives the performance of SOS_p with scaled rounding with scaling factors $2^{-(i-1)}$. Unfortunately, no actual implementations of the rounding procedures of Feige and Goemans and Lewin, Livnat and Zwick were available in order to get a more complete view here.

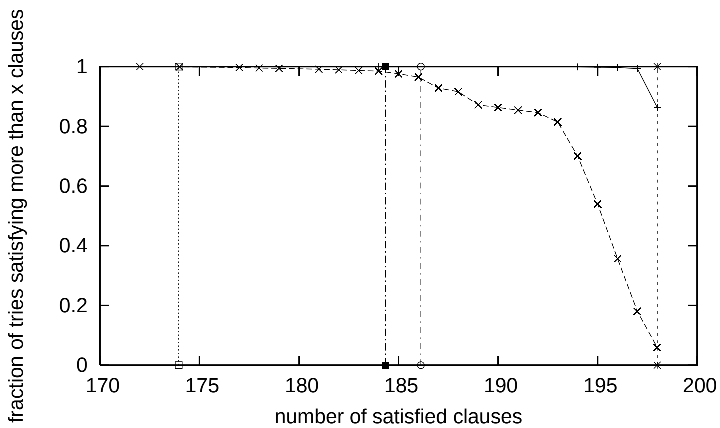

Fig. 1. 25 variables, density 9.0, observed vs proven performance

From this subsection, we might conclude that the rounding procedure based on SOS_p with any of the scaling factors investigated has better observed performance guarantee and larger fraction of tries for which the optimum is found than the approach of Goemans and Williamson. The observed performance guarantee is on average better than the proven performance guarantees of the approaches of Feige and Goemans and Lewin, Livnat and Zwick. We observe that the steepness of the scaling factors is of influence, but not necessarily with identical impact on both aspects: the frequency of finding particular good solutions on one side and yielding high observed performance on the other hand.

6.2 A rounding procedure for MAX-3-SAT based on SOS_t

In this section, we use a similar rounding procedure as in subsection 6.1, but instead of SOS_p we use SOS_t. In this rounding procedure the entries that we want to get relatively small are the entries corresponding to the monomials of degree 3. We chose SOS_t and not SOS_{pt} because SOS_t can be solved much faster than SOS_{pt} and the size of the monomial basis is comparable to the size of the monomial basis of SOS_p in the MAX-2-SAT case. For the experiments on the rounding procedures for random MAX-3-SAT instances we generated 100 unsatisfiable instances with 20 variables for each of the densities 4.0, 5.0, 6.0, 7.0, 8.0 and 9.0. We test the same variants of SOS-rounding as in the MAX-2-SAT case and compare the results with the Goemans-Williamson rounding applied on the SDP as presented by Karloff and Zwick [14]. We apply each of the rounding procedures a 1000 times on each of the instances.

Let SRT denote the rounding procedure with a uniform random vector λ without scaling.

Table 9 gives the average over the instances of the number of times out of the 1000 that the optimal number of clauses is attained. In this case, the observed performance guarantee for the procedures based on SOS_t is defined as the average number of clauses satisfied in a try divide by the upper bound obtained by SOS_t. The observed performance guarantee of the approach of Karloff and Zwick is defined as the average number of clauses satisfied in a try divided by the upper bound obtained by their SDP. Table 10 gives the average over the instances of the observed performance guarantees of the different approaches. Table 9 shows a trend break from density 7.0 to density 8.0. The average number of times the optimum is found decreases from density 4.0 up till density 7.0 and to density 8.0 there is an increase. The reason can be found in the number of eigenvectors corresponding to eigenvalue 0. Among the instances with density 8.0 and 9.0 are instances with a relatively small number of eigenvectors corresponding to eigenvalue 0, which is not the case for instances with smaller densities. The rounding procedures perform observably better when the number of eigenvectors corresponding to eigenvalue 0 is small. This might explain why the quality of the rounding starts performing much better from density 8.0 and higher.

Considering both tables shows a similar effect of the scaling as in Section 6.1, but considerably less pronounced. Note that the observed performance guarantee of the algorithm of Karloff and Zwick tends to its proven performance guarantee for larger densities. Notable is that the observed performance guarantee of the algorithm of Karloff and Zwick decreases with increasing density while the observed performance guarantee of the SOS-based procedures increases. This is mainly caused by the fact that the upper bounds tend to be better with larger densities using SOS_t.

To give some indication about the times needed to solve the SDP's, we remark that for the instances with 20 variables and density 7.0, the SDP of Karloff and Zwick takes on average about 6.2 seconds when solved by CSDP. On the same instances SOS_t takes on average 2883.7 seconds on the same machine and with the same solver.

From the above observations we might conclude that the choice for the scaling factors depends on the goal: finding for as many instances as possible a relatively good solution or having a high probability of finding a particularly good solution in a try. The reason that scaling factors ρ_i^N perform relatively bad for these instances is that the number of eigenvectors corresponding to eigen-

Table 9
MAX-3-SAT, 20 variables, average time optimum found

d	S_{RT}	ρ_i^2	ρ_i^5	ρ_i^{15}	ρ_i^N	$2^{-(i-1)}$	KZ
4.0	6.67	25.49	44.42	52.08	51.34	49.81	13.8
5.0	2.05	8.75	16.82	19.11	15.34	19.36	4.47
6.0	0.89	5.50	13.81	28.75	37.27	20.07	1.4
7.0	7.20	9.22	11.59	10.48	8.09	13.78	0.62
8.0	128.54	140.45	144.31	147.11	148.95	147.19	0.35
9.0	228.51	231.35	243.83	237.35	237.3	236.76	0.06

Table 10
MAX-3-SAT, 20 variables, observed performance guarantee

d	S_{RT}	ρ_i^2	ρ_i^5	ρ_i^{15}	ρ_i^N	$2^{-(i-1)}$	KZ
4.0	0.924	0.941	0.949	0.953	0.954	0.952	0.931
5.0	0.921	0.937	0.945	0.950	0.949	0.949	0.924
6.0	0.922	0.939	0.948	0.954	0.955	0.953	0.916
7.0	0.928	0.943	0.952	0.956	0.956	0.955	0.908
8.0	0.942	0.954	0.961	0.964	0.965	0.964	0.900
9.0	0.953	0.963	0.968	0.972	0.972	0.972	0.896

value 0 tends to be quite large, yielding a very steep function such that only very few eigenvectors make a non-negligible contribution to the linear combination. Relatively flat scaling factors like for example ρ_i^2 yield λ-vectors that are relatively much different among each other. This leads to many different assignments in the tries. Among these assignment are very good ones and very bad ones, but the chance on a very good one is not very large. Steeper scaling factors like ρ_i^{15} for example, give assignments that are not very different from each other and most are good. Because most assignments are quite similar one might miss the optimal solution in any try, although the assignment are on average very good.

Tables 7, 8, 9 and 10 show that is more difficult to find the optimal solution with the rounding procedure for MAX-3-SAT than for MAX-2-SAT for instances of the size considered.

The above instances are pure MAX-3-SAT instances. Because the upper bound found using the algorithm of Karloff and Zwick equals the number of clauses for all of these instances, we investigate a set of instances with mixed clause lengths. We present in Tables 12 and 11 the results of the algorithm of Karloff and Zwick and SOS_{pt} on a set of instances with 20 variables and 80 clauses of different lengths. The first column contains the number of clauses of length one,

Table 11
MAX-3-SAT, 20 variables, 80 clauses (C_1 unit clauses, C_2 binary clauses, C_3 ternary clauses), number of times optimum found using 1000 tries

$\|C_1\|$	$\|C_2\|$	$\|C_3\|$	KZ	S_{RT}	ρ_i^2	ρ_i^5	ρ_i^{15}	ρ_i^N	$2^{-(i-1)}$
5	10	65	405.72	530.16	548.2	582.22	700.04	658.76	806.78
5	20	55	466.26	545.98	567.14	596.86	712.8	683.76	798.96
5	25	50	536.28	514.02	529.88	573.94	677.96	637.92	776.9
10	5	65	539.04	568.32	585.75	596	725.42	686.36	813.2
10	10	60	546.24	548.34	565.94	592.9	702.22	680.38	782
10	15	55	476.96	454.74	475.3	514.14	652.1	618.92	703
10	20	50	556.2	472.6	488.04	524.34	650.78	589.66	741.04
15	5	60	607.18	512.62	530.48	560.78	668.26	626.2	754.66

Table 12
MAX-3-SAT, 20 variables, 80 clauses (C_1 unit clauses, C_2 binary clauses, C_3 ternary clauses), observed performance guarantee

$\|C_1\|$	$\|C_2\|$	$\|C_3\|$	KZ	S_{RT}	ρ_i^2	ρ_i^5	ρ_i^{15}	ρ_i^N	$2^{-(i-1)}$
5	10	65	0.963	0.983	0.984	0.987	0.993	0.992	0.996
5	20	55	0.969	0.982	0.985	0.987	0.993	0.992	0.995
5	25	50	0.974	0.981	0.983	0.986	0.992	0.992	0.995
10	5	65	0.975	0.983	0.985	0.987	0.993	0.993	0.996
10	10	60	0.975	0.98	0.982	0.985	0.992	0.992	0.995
10	15	55	0.97	0.972	0.976	0.981	0.989	0.988	0.991
10	20	50	0.977	0.98	0.983	0.986	0.992	0.99	0.994
15	5	60	0.981	0.982	0.984	0.987	0.992	0.991	0.995

the second the number of clauses of length two and the third one the number of clauses of length three. Table 12 contains the observed performance guarantees and Table 11 the average number of times out of 1000 tries that the optimal solution is found. The fourth column contains the results of the algorithm of Karloff and Zwick, followed by the results of SOS_{pt} with unweighted rounding, rounding with respectively ρ_i^2, ρ_i^5, ρ_i^{15} and ρ_i^N. The last column contains the results of rounding with scaling factor $2^{-(i-1)}$.

From Table 12 and 11 it is clear that the performance guarantee of each of the SOS_{pt}-variants is better than Karloff and Zwick's. The variant with the best performance also obtains the optimal solution more often than Karloff and Zwick's method.

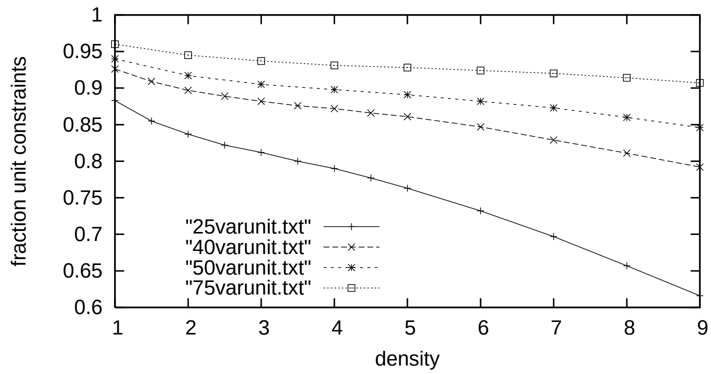

Fig. 2. Fraction of unit constraints, 2-SAT

7 Fraction of unit constraints in SOS_p, SOS_t and SOS_{pt}

In this section we investigate the type of constraints in SOS_p and show that many of the constraints are of the form $s_{ij} + s_{ji} = c$ for some constant c. Because of the symmetry of the matrix S, these constraints simply fix the two matrix entries concerned. SDP-solvers that make use of the large fraction of these 'unit' constraints in an efficient way might be able to solve these SDP's much faster than current SDP-solvers. The results show that efforts should be made to design such solvers.

We generated random 2-SAT instances with each of the densities 1.0, 2.0, 3.0, ..., 9.0 and with 25, 40, 50 and 75 variables. For each of these densities and sizes we determined the average percentage of unit constraints of the type $s_{ij} + s_{ji} = c$ in the corresponding SDP SOS_p.

Figure 2 illustrates that the fraction of unit constraints increases with problem size. The percentage of unit constraints decreases with increasing density for fixed size.

A small-scale experiment with randomly generated unsatisfiable 3-SAT instances with 20 variables and density 4.0 and density 5.0 showed that for these instances on average respectively 55.4% and 53.1% of the constraints in SOS_{pt} are unit constraints. Analogously to the 2-SAT case this percentage is expected to be larger for larger instances. A small set of instances with 40 variables and density 4.25 having 76.8% unit constraints illustrates this expectation to be valid. The percentage of unit constraints in SOS_t is considerably larger. For the same set of instances SOS_t has on average 95.9% unit constraints.

8 Counterexamples to uncovered cases of Hilbert's Positivstellensatz?

The theory discussed in the introduction might be used to try to find a counterexample to one of the uncovered cases of Hilbert's Positivstellensatz. Note that there are finitely many different k-SAT CNF-formulas with n variables for fixed k. We showed in Section 1 that a formula ϕ is unsatisfiable if and only if there exists an $\epsilon > 0$ such that $F_\phi(x) \geq \epsilon$ for each $x \in \mathbb{R}^n$. Let Φ_n^k be the set of unsatisfiable k-SAT formula with n variables. Let ϵ_n^k be defined as

$$\epsilon_n^k = \min_{\phi \in \Phi_n^k} \left\{ \min_x F_\phi(x) \right\} \tag{52}$$

ϵ_n^k is always strictly positive. Practically, it is not possible to determine ϵ_n^k because it is the minimum over the exponentially many minima of all unsatisfiable k-SAT instances with n variables. Even finding the exact minimum for a given ϕ is not doable with current software.

Definition 2 *An unsatisfiable k-SAT formula ϕ with n variables is a CUC if $F_\phi - \epsilon_n^k$ is not a sum of squares.*

Alternatively, we can conclude that an unsatisfiable k-SAT formula ϕ is a CUC if the program

$$\begin{aligned} &\max \alpha \\ \text{s.t.} F_\phi - \alpha \ &\text{is a SOS} \\ &\alpha \in \mathbb{R} \end{aligned} \tag{53}$$

returns an α smaller than or equal to ϵ_n^k. To decide this a Semidefinite Programming solver is necessary that returns a solution with precision at most ϵ_n^k.

In some small-scale experiments we compute the maximum of program (53) for each of the considered instances with the SDP solver CSDP [7]. We selected this solver because its precision turned out best among a small set of solvers we tried. We generated a set of 2-SAT instances with 10 variables with a density varying between 1.0 and 3.0. For all 1500 sample instances except two, the SDP (53) gives solutions such that any α corresponding to a satisfiable instance of a particular density is smaller than any α corresponding to an unsatisfiable instance of the same size and density although this difference may be very small. This implies that the considered satisfiable and unsatisfiable 2-SAT formulas are almost perfectly separated by their value of α. The precision of the solver is not sufficiently adequate because for some of the satisfiable instances it returns a positive α which is not allowed based on Theorem 1. Based on these experiments there is no reason to suppose that polynomials of degree 4 coming from unsatisfiable 2-SAT formulas may yield the desired counterexamples. On the other hand, due to the numerical imprecision the opposite cannot be concluded either. Only much more accurate optimization methods, or much more accurate implementations of SDP-algorithms, could result in a more final conclusion.

In a small-scale experiment with 100 randomly generated 3-SAT instances with 10 variables and density 4.0 and 100 instances with density 5.0, we obtained similar results. We found an almost perfect separation of satisfiable and unsatisfiable instances based on the value of α but again no final conclusion can be made whether unsatisfiable random 3-SAT instances may provide CUCs. Although the experiments described give an unsatisfactory result concerning CUC's, they certainly indicate that the SOS approach has strong separating power regarding satisfiability. Because k-SAT is known to be $\mathcal{NP}$-complete (for $k \geq 3$) and the construction of the monomial basis M_ϕ using the Newton polytope for F_ϕ can be carried out in polynomial time, we can conclude with the following theorem:

Theorem 8 *If* $\max\left\{\alpha | F_\phi - \alpha = M_\phi^T S M_\phi, S \succeq 0\right\} > 0$ *can be decided in polynomial time (open) and* $\mathcal{NP} \neq \mathcal{P}$*, infinitely many CUC's exist for each* $k \geq 3$*.*

Corollary 2 *Under the same conditions, infinitely many polynomials of degree* $2k(k \geq 3)$*, coming from unsatisfiable instances, are non-negative but not sums of squares.*

9 Conclusions

In this paper, we compare the SOS (Sums Of Squares) approaches with existing upper bound and rounding techniques for the MAX-2-SAT case of Goemans and Williamson [12] and Feige and Goemans [10] and the MAX-3-SAT case of Karloff and Zwick [14], which are based on Semidefinite Programming as well. We prove that for each of these algorithms there is a SOS-based counterpart which provides upper bounds at least as tight, but observably tighter in particular cases.

We conclude that a combination of the rounding schemes of Goemans and Williamson and of Feige and Goemans with the appropriate SOS based upper bound techniques proposed, leads to polynomial time algorithms for MAX-2-SAT having a performance ratio guarantee at least as good as the ones proven by Goemans and Williamson and Feige and Goemans, but observably better in particular cases. A similar conclusion can be drawn with respect to the Karloff and Zwick algorithm for MAX 3-SAT. Also, the experiments with the newly proposed randomized rounding schemes for SOS_p and SOS_t seem promising.

Further, first experiments on the decision variant give reasons to believe that the sums of squares approaches presented yield better decisive results than the natural equivalents of Anjos [1–3], at least for the unsatisfiable cases. Our exposure of the fraction of “unit constraints” in the SDP relaxations is added to motivate researchers to develop SDP algorithms, where such constraints could be invoked directly, instead of added to the (already huge) list of “non-trivial” constraints. Finally, we gave some first considerations which could be helpful to decide whether polynomial transforms of SAT instances could provide counterexamples to uncovered cases of Hilberts Positivstellensatz, which would be rather interesting to know, since the known counterexamples stem from a completely different nature [5].

References

[1] M.F. Anjos. On semidefinite programming relaxations for the satisfiability problem. *Mathematical Methods of Operations Research*, 60(3):349–367, 2004.

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

[3] M.F. Anjos. Semidefinite optimization approaches for satisfiability and maximum-satisfiability problems. *Journal on Satisfiability, Boolean Modeling and Computation*, 1:1–47, 2005.

[4] Steven J. Benson, Yinyu Ye, and Xiong Zhang. Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM Journal on Optimization*, 10(2):443–461, 2000.

[5] G. Blekherman. There are significantly more nonnegative polynomials than sums of squares. *Israel Journal of Mathematics*, 153:355, 2005.

[6] B. Borchers and J. Furman. A two-phase exact algorithm for MAX-SAT and weighted MAX-SAT problems. *Journal of Combinatorial Optimization*, 2(4):299–306, 1999.

[7] B. Borchers. CSDP : A C library for semidefinite programming. Technical report, New Mexico Tech, 1997.

[8] 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.

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

[10] 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.

[11] 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.

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

[13] J. Håstad. Some optimal inapproximability results. *Journal of the Association of Computing Machinery*, 48:798–859, 2001.

[14] 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.

[15] D.E. Knuth. *The art of computer programming, vol. 2: seminumerical algorithms*. Addison-Wesley, 1969.

[16] M. Lewin, D. Livnat, and U. Zwick. Improved rounding techniques for the max-2-sat and max-di-cut problems. In *Proceedings of 9th Integer Programming and Combinatorial Optimization Conference*, pages 67–82, 2002.

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

[18] M. Putinar. Positive polynomials on compact semi-algebraic sets. *Indiana University Mathematics Journal*, 42(3), 1993.

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

[20] U. Zwick. Computer assisted proof of optimal approximability results. In *Proceedings of 13th SODA*, pages 496–505, 2002.