Approximation algorithms for constraint satisfaction problems involving at most three variables per constraint

Uri Zwick *

July 7, 1997

Abstract

An instance of MAX 3CSP is a collection of m clauses of the form $f_i(z_{i1}, z_{i2}, z_{i3})$, where the z_{ij}'s are literals, or constants, from the set $\{0, 1, x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n\}$, and the f_i's are arbitrary Boolean functions depending on (at most) three variables. Each clause has a non-negative weight w_i associated with it. A solution to the instance is an assignment of 0-1 values to the variables $x_1, \ldots, x_n$ that maximizes $\sum_{i=1}^{n} w_i f_i(z_{i1}, z_{i2}, z_{i3})$, the total weight of the satisfied clauses. The MAX 3CSP problem is clearly a generalization of the MAX 3SAT problem. (In an instance of the MAX 3SAT problem $f_i(z_{i1}, z_{i2}, z_{i3}) = z_{i1} \vee z_{i2} \vee z_{i3}$ for every i.)

Karloff and Zwick have recently obtained a $\frac{7}{8}$-approximation algorithm for MAX 3SAT. Their algorithm is based on a new semidefinite relaxation of the problem. Håstad showed that no polynomial time algorithm can achieve a better performance ratio, unless P=NP. Here we use similar techniques to obtain a $\frac{1}{2}$-approximation algorithm for MAX 3CSP. The performance ratio of this algorithm is also optimal, as follows again from the work of Håstad. We also obtain better performance ratios for several special cases of the problem. Our results include:

- A $\frac{1}{2}$-approximation algorithm for MAX 3AND, the problem in which each clause is of the form $z_{i1} \wedge z_{i2} \wedge z_{i3}$. This result is optimal and it implies the result for MAX 3CSP.
- A $\frac{2}{3}$-approximation algorithm for MAX 3MAJ, the problem in which each clause is of the form $\texttt{MAJ}(z_{i1}, z_{i2}, z_{i3})$, where $\texttt{MAJ}$ is the *majority* function. This result is again optimal.
- A $\frac{3}{2+\sqrt{2}}$-approximation algorithm for MAX 3NAE, the problem in which each clause is of the form $\texttt{NAE}(z_{i1}, z_{i2}, z_{i3})$, where $\texttt{NAE}(x_1, x_2, x_3) = 1$ if and only if x_1, x_2 and x_3 are not all equal. (We assume here that the same literal, or constant, is not allowed to appear more than once in the same clause.)

Finally, building on ideas of Trevisan, we get a $\frac{5}{8}$-approximation algorithm for *satisfiable* instances of MAX 3CSP. We conjecture that this result is optimal. Our results imply limits on the power of *non-adaptive* verifiers of probabilistically checkable proofs (PCP's) that read only three bits of the proof. We also obtain a $\frac{3}{2\pi}\arccos(-\frac{1}{3})$-approximation algorithm for *satisfiable* instances of MAX 3NAE. (Note that $\frac{3}{2+\sqrt{2}} \simeq 0.87868$ while $\frac{3}{2\pi}\arccos(-\frac{1}{3}) \simeq 0.91226$.)

We should mention here that for most of the performance ratios claimed above we do not have complete analytical proofs. Each such performance ratio relies on one or more inequalities in six real variables. These inequalities involve the volume function of *spherical tetrahedra*, a non-elementary function. Numerical methods were used to verify the validity of these inequalities. Our results may therefore be viewed, at least for the time being, as experimental results. The results stated above for MAX 3NAE are among the ones for which we *do* have complete analytical proofs.

*Department of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel. Email: `zwick@math.tau.ac.il`. This work was done while this author was visiting ICSI and UC Berkeley.

1 Introduction

Recent years have seen two major breakthroughs in the field of approximation algorithms. On one end, an impressive series of works, that includes [FGL+96],[AS92], [ALM+92],[BGLR93],[BGS95], [Hås97], established, among other things, that the exists a constant $\gamma < 1$ such that MAX E3SAT, the version of MAX SAT in which each clause contains *exactly* three literals, cannot be approximated with a performance ratio of at least γ, unless P=NP. In the last work of this series, Håstad obtains this result with $\gamma = \frac{7}{8} + \epsilon$, for every $\epsilon > 0$. A random assignment satisfies, on average, $\frac{7}{8}$ of the total weight of the clauses of a MAX E3SAT instance. Håstad's result is therefore tight.

In a second breakthrough, Goemans and Williamson [GW95] used semidefinite programming to obtain a 0.87856-approximation algorithm for MAX CUT. Feige and Goemans [FG95] followed with a 0.93109-approximation algorithm for MAX 2SAT. Karloff and Zwick [KZ97] recently obtained a $\frac{7}{8}$-approximation algorithm for MAX 3SAT, [1] the version of MAX SAT in which each clause is of length *at most* three. Here we continue this line of research and obtain a few more optimal approximation algorithms. In particular, we obtain a $\frac{1}{2}$-approximation algorithm for MAX 3CSP, improving a 0.387-approximation algorithm of Trevisan, Sorkin, Sudan and Williamson [TSSW96], and a $\frac{5}{8}$-approximation algorithm for *satisfiable* instances of MAX 3CSP, improving a 0.514-approximation algorithm of Trevisan [Tre97]. As discussed in [TSSW96] and [Tre97], this imply limits on the power of *non-adaptive* verifiers of probabilistically checkable proofs (PCP's) that read only three bits of the proof.

Many other papers deal with issues related to the ones investigated here. Following is a partial list of such papers: [Asa97], [AOH96], [CFZ97], [Cre95], [CT96], [GW94], [Joh74], [KLP96], [Kar96], [KMSV94], [KST97], [KSW97], [MR95], [PY91], [Pet94], [Sch78] and [Yan94].

2 Constraints involving at most three variables

In the sequel, we denote by MAX CSP(f), where f is a 3-variable Boolean function, the subproblem of MAX 3CSP in which all the constraints are of the form $f(z_{i1}, z_{i2}, z_{i3})$.

We say that two Boolean functions $f(x_1, \ldots, x_k)$ and $g(x_1, \ldots, x_k)$ are of the same *type* if and only if there exist a permutation $\pi : \{1, \ldots, k\} \to \{1, \ldots, k\}$ and constants $a_1, \ldots, a_k \in \{0, 1\}$ such that

$$g(x_1, \ldots, x_k) \;=\; f(x_{\pi(1)} \oplus a_1, \ldots, x_{\pi(k)} \oplus a_k) \quad \text{for every } x_1, \ldots, x_k \in \{0, 1\}.$$

Clearly, if f and g are of the same type, then the problems MAX CSP(f) and MAX CSP(g) are equivalent. The three types of Boolean functions depending on two variables are shown in Figure 1. They are represented by the three functions $\texttt{OR}_2 = x_1 \vee x_2$, $\texttt{XOR}_2 = x_1 \oplus x_2$ and $\texttt{AND}_2 = x_1 \wedge x_2$. Goemans and Williamson [GW95] obtained a 0.87856-approximation algorithm for MAX CSP($\texttt{XOR}_2$), which includes MAX CUT as a subproblem. Feige and Goemans [FG95] obtained a 0.93109-approximation algorithm for MAX CSP($\texttt{OR}_2$), also known as the MAX 2SAT problem, and a 0.859-approximation algorithm for MAX CSP($\texttt{AND}_2$), which includes MAX DICUT as a subproblem. The upper bounds on the approximability of these problems were obtained by Trevisan, Sorkin, Sudan and Williamson [TSSW96]. They are explained in more detail in Section 10. We note here that the inapproximability bounds given here for MAX CSP($\texttt{XOR}_2$) and MAX CSP($\texttt{AND}_2$) are stronger, i.e., smaller, than those known for MAX CUT and MAX DICUT. MAX CUT and MAX DICUT are subproblems of these problems and may therefore be easier to approximate. The best inapproximability bounds known for MAX CUT and MAX DICUT are $\frac{16}{17}$ and $\frac{12}{13}$, respectively.

A simple computer program can be used to show that there are 16 types of Boolean functions that depend on 3 variables. They are given in Figure 2. An instance of MAX 3CSP may thus contain constraints of

[1] More precisely, [KZ97] obtain a $(\frac{7}{8} - \epsilon)$-approximation algorithm for satisfiable instances of MAX 3SAT and provide compelling analytical and numerical evidence that suggest that the performance ratio of the algorithm is $\frac{7}{8} - \epsilon$ for *all* instances of the problem, for every $\epsilon > 0$. It is possible to show that the ϵ in the above expressions can be eliminated.

constraint type	performance ratio	inapproximability bound
$\texttt{OR}_2 = x_1 \vee x_2$	0.93109	$\frac{21}{22}$
$\texttt{XOR}_2 = x_1 \oplus x_2$	0.87856	$\frac{11}{12}$
$\texttt{AND}_2 = x_1 \wedge x_2$	0.859	$\frac{9}{10}$

Figure 1: The 3 types of 2-variable constraints.

AND	=	$x_1 \wedge x_2 \wedge x_3$
EQU	=	$x_1x_2x_3 \vee \bar{x}_1\bar{x}_2\bar{x}_3$
AXR	=	$x_1 \wedge (x_2 \oplus x_3)$
AOR	=	$x_1 \wedge (x_2 \vee x_3)$
AOA	=	$x_2x_3 \vee x_1\bar{x}_2\bar{x}_3$
TWO	=	$x_1x_2\bar{x}_3 \vee x_1\bar{x}_2x_3 \vee x_1x_2\bar{x}_3$
XOR	=	$x_1 \oplus x_2 \oplus x_3$
MAJ	=	$x_1x_2 \vee x_1x_3 \vee x_2x_3$

XAD	=	$x_1 \oplus x_2x_3$
SEL	=	$x_1\bar{x}_3 \vee x_2x_3$
OAD	=	$x_1 \vee x_2x_3$
XOA	=	$(x_1 \oplus x_2) \vee x_2x_3$
NTW	=	$(\bar{x}_1 \vee \bar{x}_2)\bar{x}_3 \vee (x_1 \equiv x_2)x_3$
NAE	=	$(x_1 \oplus x_2) \vee (x_1 \oplus x_3) \vee (x_2 \oplus x_3)$
OXR	=	$x_1 \vee (x_2 \oplus x_3)$
OR	=	$x_1 \vee x_2 \vee x_3$

Figure 2: The 16 types of 3-variable constraints.

20 different types: the 16 types shown in Figure 2, the 3 types shown in Figure 1, and constaints of the form z_i, where z_i is a literal. In the rest of the paper we describe approximation algorithms for MAX 3CSP, as well as for the problems MAX CSP(f), where f is one of the 16 functions given in Figure 2.

We now try to acquaint ourselves with the functions appearing in Figure 2. The functions `AND`, `XOR` and `OR` need no introduction. The other symmetric functions are $\texttt{EQU}(x_1, x_2, x_3)$, which returns 1 iff $x_1 = x_2 = x_3$, $\texttt{TWO}(x_1, x_2, x_3)$, which returns 1 iff exactly *two* of x_1, x_2 and x_3 are 1, $\texttt{NAE}(x_1, x_2, x_3)$, which returns 1 iff x_1, x_2 and x_3 are *not all equal*, and $\texttt{NTW}(x_1, x_2, x_3)$, which returns 1 iff the number of 1's is *not two*. Note that the function $\texttt{ONE}(x_1, x_2, x_3)$, which returns 1 iff the number of 1's in exactly *one*, is of the same type as `TWO`, as $\texttt{ONE}(x_1, x_2, x_3) = \texttt{TWO}(\bar{x}_1, \bar{x}_2, \bar{x}_3)$. The functions $\texttt{AXR} = x_1 \wedge (x_2 \oplus x_3)$, $\texttt{AOR} = x_1 \wedge (x_2 \vee x_3)$, $\texttt{XAD} = x_1 \oplus x_2x_3$, $\texttt{OAD} = x_1 \vee x_2x_3$ and $\texttt{OXR} = x_1 \vee (x_2 \oplus x_3)$ are also farily natural. Finally, $\texttt{SEL}(x_1, x_2, x_3)$ is the *selection* function which outputs x_1, if $x_3 = 0$, and x_2, if $x_3 = 1$. We occasionally attach the subscript 3 to these function names to remind us that they all depend on three variables.

3 Semidefinite relaxations

All our approximation algorithms use the *cannonical* semidefinite relaxations of constraint satisfaction problems introduced by Karloff and Zwick [KZ97].

For ease of notation, we encode an instance of MAX CSP(f) in the following way. Let $x_1, \ldots, x_{n-1}$ be the variables appearing in the instance. (Note that the number of Boolean variables is assumed to be $n-1$ and not n). We let x_0 stand for the Boolean constant 1 and $x_n, \ldots, x_{2n-1}$ stand for the negations $\bar{x}_0, \ldots, \bar{x}_{n-1}$. (Note that x_n stands for the Boolean constant 0.) The instance is then represented as an array $\{w_{ijk}\}$ of non-negative weights, where w_{ijk}, for $0 \le i, j, k < 2n$, is the weight of the clause $f(x_i, x_j, x_k)$. Thus, $w_{0,7,n+4}$, for example, is the weight of the clause $f(0, x_7, \bar{x}_4)$, and $w_{5,n,n+5}$ is the weight of the clause $f(x_5, 1, \bar{x}_5)$. Unless specifically prohibited, clauses are allowed to contain constants, and they are allowed to contain more than one occurrence of the same variable, as in the second example above.

The semidefinite program used as a relaxation for an instance $\{w_{ijk}\}$ of MAX 3AND is given in Figure 3. The variables of the program are the n unit vectors $v_0, v_1, \ldots, v_{n-1}$, in $\mathbb{R}^n$, corresponding to the constant 1

$$
\begin{array}{lll}
\text{Maximize} & \displaystyle\sum_{i,j,k} w_{ijk} \cdot z_{ijk} & \\
\text{subject to} & z_{ijk} \le \frac{1+v_0 \cdot v_i + v_0 \cdot v_j + v_i \cdot v_j}{4} & \forall\, 0 \le i,j,k < 2n \\
& z_{ijk} \le \frac{1+v_0 \cdot v_i + v_0 \cdot v_k + v_i \cdot v_k}{4} & \forall\, 0 \le i,j,k < 2n \\
& z_{ijk} \le \frac{1+v_0 \cdot v_j + v_0 \cdot v_k + v_j \cdot v_k}{4} & \forall\, 0 \le i,j,k < 2n \\
& z_{ijk} \le \frac{1+v_i \cdot v_j + v_i \cdot v_k + v_j \cdot v_k}{4} & \forall\, 0 \le i,j,k < 2n \\
& v_i \in S^{n-1} & \forall\, 0 \le i < 2n \\
& v_{n+i} = -v_i & \forall\, 0 \le i < n
\end{array}
$$

Figure 3: A semidefinite relaxation of MAX 3AND.

$$
v_i \cdot v_j + v_i \cdot v_k + v_j \cdot v_k \ge -1 \quad \forall\, 0 \le i,j,k < 2n
$$

Figure 4: Possible additional constraints.

and the variables $x_1, \ldots, x_{n-1}$, and scalars z_{ijk} corresponding to the clauses of the instance. For convenience, we let $v_{n+i} = -v_i$, for $0 \le i < n$. (In the relaxation, S^{n-1} stands for the unit sphere in $\mathbb{R}^n$.) We should note here that while in [KZ97] v_0 was used to represent the constant 0, here we find it more convenient to let v_0 represent the constant 1.

To verify that this is indeed a relaxation of MAX 3AND, we note that for any truth assignment $a_1, \ldots, a_{n-1}$, we can generate a feasible point of the semidefinite program in the following way: We define $v_i = (-1, 0, \ldots, 0)$, if $a_i = 0$, and $v_i = (1, 0, \ldots, 0)$, if $a_i = 1$. We also let $v_0 = (1, 0, \ldots, 0)$. We then define $z_{ijk} = 1$, if $a_i \wedge a_j \wedge a_k = 1$, and $z_{ijk} = 1$, otherwise. (We use here the convention that $a_0 = 1$ and $a_{n+i} = \bar{a}_i$, for $0 \le i < n$.) It is not difficult to check that this is indeed a feasible point of the program.

We may, if we want to, add the additional constraints given in Figure 4. These are the additional constraints used by Feige and Goemans [FG95]. We are allowed to add these constraints as the points constructed in the previous paragraph satisfy them. In the case of MAX 3AND, the addition of the additional constraints does not lead to an improved performance ratio. For some of the other problems, they do help.

The semidefinite relaxations that we use for problems MAX CSP(f), with different choices of f, are all similar to the one shown in Figure 3. The only differene is that different bounds are used on the z_{ijk}'s and that sometimes we also use the additional constraints of Figure 4. The relaxations used for all other types of Boolean constraints involving at most three variables are given in Appendix A.

How are these relaxations obtained? This is described in [KZ97]. Very briefly, to any Boolean function $f : \{-1,1\}^k \to \{0,1\}$ we attach a polytope *polytope*(f) in $\mathbb{R}^{\frac{k(k+1)}{2}+1}$. (For convenience we use $\{-1,1\}^k$, instead of $\{0,1\}^k$, as the domain of f, with -1 representing FALSE and 1 representing TRUE.) The constraints of the semidefinite relaxation correspond to the *facets* of *polytope*(f). (We assume here that *polytope*(f) is non-degenerate). If $\sum_{ij} \alpha_{ij} y_{ij} + \beta z \le \gamma$ is a facet of the polytope, the semidefinite relaxation will include constraints of the form $z \le \frac{\gamma - \sum_{ij} \alpha_{ij} v_i \cdot v_j}{\beta}$.

Before definning *polytope*(f), we need the following definition. Given a vector $x = (x_1, \ldots, x_k) \in \{-1,1\}^k$, we let $\mathrm{prod}(x) = (x_0 x_1, x_0 x_2, \ldots, x_{k-1} x_k)$, where $x_0 = 1$. Note that $\mathrm{prod}(x) \in \{-1,1\}^{\frac{k(k+1)}{2}}$. The polytope *polytope*(f) is now defined as follows:

$$
\mathit{polytope}(f) \;=\; \mathrm{conv}(\, \{\, (\mathrm{prod}(x), f(x)) \mid x \in \{-1,1\}^k \,\}\,)\ ,
$$

where $(\mathrm{prod}(x), f(x))$ denotes a vector of length $\frac{k(k+1)}{2} + 1$ obtained by appending $f(x)$ to $\mathrm{prod}(x)$, and

$\text{conv}(\{v_1, \ldots, v_\ell\})$ is the *convex hull* of the set $\{v_1, \ldots, v_\ell\}$. The points $(\text{prod}(x), f(x))$, for $x \in \{-1,1\}^k$, are therefore the vertices of *polytope*(f).

It is not difficult to see (and it is shown in [KZ97]) that the obtained semidefinite programs are indeed relaxations of the Boolean constraint satisfaction problems. Furthermore, they are the *strongest* semidefinite relaxations within a natural class of relaxations that are allowed to 'consider' the Boolean constraints only one at a time. For details, the reader is referred to [KZ97].

4 Rounding using a random hyperplane

A good relaxation is only the first component of a good approximation algorithm. The second component is a good *rounding* procedure. In our case, a rounding procedure should convert the vectors $v_0, \ldots, v_{n-1}$ obtained as an optimal solution to the semidefinite program into a truth assignment $a_1, \ldots, a_{n-1}$ to the variables $x_1, \ldots, x_{n-1}$.

The first rounding procedure that comes to mind, in connection with semidefinite programs, is the rounding procedure suggested by Goemans and Williamson [GW95]. This rounding procedure chooses a random *hyperplane*. A variable x_i is assigned the value 1 iff v_i and v_0 lie on the same side of the hyperplane.

This rounding procedure performs well for MAX CUT (no better rounding procedure is known for MAX CUT). This is also the rounding procedure used in [KZ97] to obtain the $\frac{7}{8}$-approximation algorithm for MAX 3SAT. Slightly more complicated rounding procedures, involving *rotations*, are used by Feige and Goemans [FG95] in their MAX 2SAT and MAX DICUT algorithms.

An even simpler, seemingly stupid, rounding procedure, that should nonetheless be considered, is the 'rounding' procedure that does not even look at the vectors $v_0, \ldots, v_{n-1}$ and simply chooses the truth values assigned to the variables $x_1, \ldots, x_{n-1}$ independently at random.

The performance ratio obtained using these two rounding procedures, for the 16 different types of constraints, are given in the second and third columns of Figure 5. The way in which these performance ratios were computed is described in the rest of this section and in Appendix C.

The performance ratio achieved for MAX CSP(f) using random assignments is simply the probability $r_f = \Pr[\ f(x_1, x_2, x_3) = 1\ ]$, where x_1, x_2 and x_3 are assigned independent random bits. For certain functions f, we have to assume here that the three literals appearing in each clause belong to different variables, and are therefore independent. For such functions, clauses like $f(x_1, x_2, 0)$, $f(x_1, x_1, x_2)$ or $f(x_1, x_2, \bar{x}_1)$ are not allowed. For some other functions f, there is no harm in allowing such clauses, as their satisfiability probability is either 0, in which case the clauses can be ignored, or at least as high as r_f.

Finding the performance ratio achieved using random hyperplane rounding is harder. A lower bound α_f on this performance ratio can be obtained as follows:

$$\alpha_f \;=\; \inf_{\mathit{relax}_f(v_0,v_1,v_2,v_3)>0} \frac{\mathit{prob}_f(v_0,v_1,v_2,v_3)}{\mathit{relax}_f(v_0,v_1,v_2,v_3)}\,,$$

where the infimum is taken over all the feasible configurations of the vectors v_0, v_1, v_2, v_3 for which $\mathit{relax}_f(v_0, v_1, v_2, v_3) > 0$. Here $\mathit{prob}_f(v_0, v_1, v_2, v_3)$ is the probability that rounding the vectors v_0, v_1, v_2, v_3 using a random hyperplane yields an assignment a_1, a_2, a_3 to x_1, x_2, x_3 for which $f(a_1, a_2, a_3) = 1$. As for $\mathit{relax}_f(v_0, v_1, v_2, v_3)$, this is the maximal feasible value of z_{123}, given the vectors v_0, v_1, v_2, v_3. For $f = \texttt{AND}_3$, for example, we get

$$\mathit{relax}_{\texttt{AND}_3}(v_0,v_1,v_2,v_3) \;=\; \min\left\{ \begin{array}{cc} \dfrac{1+v_0\cdot v_1+v_0\cdot v_2+v_1\cdot v_2}{4}\,, & \dfrac{1+v_0\cdot v_1+v_0\cdot v_3+v_1\cdot v_3}{4} \\ \dfrac{1+v_0\cdot v_2+v_0\cdot v_3+v_2\cdot v_3}{4}\,, & \dfrac{1+v_1\cdot v_2+v_1\cdot v_3+v_2\cdot v_3}{4} \end{array} \right\}.$$

constraint type	random assignment	random hyperplane	improved rounding	inapproximability bound
AND	$\frac{1}{8}$	0	$\frac{1}{2}$	$\frac{1}{2}$
EQU	$\frac{1}{4}$	0.796070		$\frac{7}{8}$
AXR	$\frac{1}{4}$	0.404973	$\frac{1}{2}$	$\frac{1}{2}$
AOR	$\frac{3}{8}$	0.733368		$\frac{3}{4}$
AOA	$\frac{3}{8}$	0.733368		$\frac{3}{4}$
TWO	$\frac{3}{8}$	0.488862	$\frac{1}{2}$	$\frac{1}{2}$
XOR	$\frac{1}{2}$	$\frac{1}{2}$		$\frac{1}{2}$
MAJ	$\frac{1}{2}$	0.666158	$\frac{2}{3}$	$\frac{2}{3}$
XAD	$\frac{1}{2}$	0.666158		$\frac{2}{3}$
SEL	$\frac{1}{2}$	0.796070	0.859	$\frac{9}{10}\big/\frac{11}{12}$
OAD	$\frac{5}{8}$	0.810686		$\frac{5}{6}$
XOA	$\frac{5}{8}$	0.810686		$\frac{5}{6}$
NTW	$\frac{5}{8}$	$\frac{5}{8}$		$\frac{5}{8}$
NAE	$\frac{3}{4}$	0.878567	$\frac{3}{2+\sqrt{2}}$	$\frac{11}{12}\big/\frac{15}{16}$
OXR	$\frac{3}{4}$	$\frac{3}{4}$		$\frac{3}{4}$
OR	$\frac{7}{8}$	$\frac{7}{8}$		$\frac{7}{8}$

Figure 5: The performance ratios obtained using the different rounding procedures and upper bounds on the achievable performance ratios for the 16 types of 3-variable constraints.

For some functions f, computing $prob_f(v_0, v_1, v_2, v_3)$ is fairly easy. For example

$$prob_{\mathtt{NAE}_3}(v_0, v_1, v_2, v_3) \;=\; \frac{\theta_{12}+\theta_{13}+\theta_{23}}{2\pi} \quad , \quad prob_{\mathtt{EQU}_3}(v_0, v_1, v_2, v_3) \;=\; 1-\frac{\theta_{12}+\theta_{13}+\theta_{23}}{2\pi} \;,$$

$$prob_{\mathtt{SEL}_3}(v_0, v_1, v_2, v_3) \;=\; 1-\frac{\theta_{01}+\theta_{02}-\theta_{13}+\theta_{23}}{2\pi} \;.$$

where $\theta_{ij} = \arccos(v_i \cdot v_j)$ is the angle between the vectors v_i and v_j. For the other functions f, computing $prob_f(v_0, v_1, v_2, v_3)$ is much harder. It follows from [KZ97] that

$$prob_{\mathtt{AND}_3}(v_0, v_1, v_2, v_3) \;=\; \frac{\mathrm{Vol}(\lambda_{01}, \lambda_{02}, \lambda_{12}, \lambda_{03}, \lambda_{13}, \lambda_{23})}{\pi^2} \;,$$

where $\lambda_{01} = \pi - \theta_{23}$, $\lambda_{02} = \pi - \theta_{13}$, $\lambda_{12} = \pi - \theta_{03}$, $\lambda_{03} = \pi - \theta_{12}$, $\lambda_{13} = \pi - \theta_{02}$, $\lambda_{23} = \pi - \theta_{01}$, and $\mathrm{Vol}(\lambda_{01}, \lambda_{02}, \lambda_{12}, \lambda_{03}, \lambda_{13}, \lambda_{23})$ is the volume of a *spherical tetrahedron* with *dihedral angles* $\lambda_{01}, \lambda_{02}, \lambda_{12}, \lambda_{03}$, λ_{13} and λ_{23}. Unfortunately, $\mathrm{Vol}(\lambda_{01}, \lambda_{02}, \lambda_{12}, \lambda_{03}, \lambda_{13}, \lambda_{23})$ seems to be a non-elementary function and no closed-form formula, in terms of the basic elementary functions, is known for it. Things are not hopeless, however, as the partial derivatives $\partial\mathrm{Vol}/\partial\lambda_{ij}$ are elementary functions. See Appendix B.

All other $prob_f(v_0, v_1, v_2, v_3)$ can be expressed using the volume function. For example

$$prob_{\mathtt{MAJ}_3}(v_0, v_1, v_2, v_3) \;=\; 3-\frac{2\theta_{01}+2\theta_{02}+2\theta_{03}+\theta_{12}+\theta_{13}+\theta_{23}}{2\pi}-\frac{2\mathrm{Vol}(\lambda_{01}, \lambda_{02}, \lambda_{12}, \lambda_{03}, \lambda_{13}, \lambda_{23})}{\pi^2} \;.$$

Computing $prob_f(v_0, v_1, v_2, v_3)$, for given v_0, v_1, v_2, v_3 is not easy. Solving the optimization problem used to define α_f is even harder. In most cases, we use numerical methods to compute α_f. We then provide analytical evidence supporting the claim that the obtained solution is indeed correct. See Appendix C.

It was shown in [KZ97], using a combination of analytical and numerical methods, that $\alpha_{\mathtt{OR}_3} = \frac{7}{8}$, resulting in a $\frac{7}{8}$-approximation algorithm for MAX 3SAT (see footnote on page 1). We should remark here that while in the analysis of the performance of random assignments we had to assume, in some cases, that two literals of the same variable are not allowed to appear in the same clause, such a restriction is *not* required when analyzing the performance ratio of the random hyperplane rounding procedure. In the case of MAX 3SAT, if all clauses are of length 3, then a random assignment achieves a performance ratio of $\frac{7}{8}$. If clauses of length 1 and 2 are allowed, then the performance ratio of random assignments drops to $\frac{1}{2}$. A clause $x_1 \vee x_2$ of length 2 can be represented, however, as a clause $x_1 \vee x_1 \vee x_2$ of length 3. This *is* allowed when considering the random hyperplane rounding procedure. The algorithm of [KZ97] achieves, therefore, a performance ratio of $\frac{7}{8}$ even when clauses of length 1 and 2 are allowed.

A random assignment satisfies, on average, $\frac{1}{8}$ of the total weight of the clauses of a MAX 3AND instance. What performance ratio do we get using random hyperplanes? The disappointing answer is 0 ! This does not look very promising. In the next section we show, however, that this can be easily fixed. We show there that if we generate two assignments, one of them being a completely random one while the other is obtained by rounding the vectors $v_0, \ldots, v_{n-1}$ using a random hyperplane, and then take the better one, we obtain a $\frac{1}{2}$-approximation algorithm for MAX 3AND.

Let us check now how does random hyperplane rounding perform for the other types of constraints. A quick look at Figure 5 shows that in all other 15 cases, random hyperplane rounding does at least as well as selecting a random assignment. This does not sound very impressive. It should be remembered, however, that the performance ratios given for random hyperplane rounding are valid even when clauses are allowed to contain constants and more than one occurrence of the same variable. For the problems MAX CSP(f), where $f = \mathtt{XOR}_3$, $\mathtt{NTW}_3$, $\mathtt{OXR}_3$ and $\mathtt{OR}_3$ we get optimal approximation algorithms. For the functions $\mathtt{NTW}_3$, $\mathtt{OXR}_3$ and $\mathtt{OR}_3$, the approximation algorithm that uses hyperplane rounding is stronger than the naive algorithm that simply picks a random assigmnets. In the case of $\mathtt{NTW}_3$, for example, we are allowed to use clauses of the form $\mathtt{NTW}(x_1, \bar{x}_1, \bar{x}_2) = x_2$ and $\mathtt{NTW}(x_1, \bar{x}_2, 1) = x_1 \oplus x_2$. In the case of $\mathtt{OXR}_3$, we are allowed, for example, to use clauses of the form $\mathtt{OXR}(0, x_1, x_2) = x_1 \oplus x_2$. The fact that random hyperplane rounding achieves performance ratios $\frac{1}{2}$ for $\mathtt{XOR}_3$ and $\frac{5}{8}$ for $\mathtt{NTW}_3$ will be important in Section 9, where an improved approximation algorithm for *satisfiable* instances of MAX 3CSP is presented.

In all the other 11 cases, random hyperplane rounding performs better than naive random choice, even if only simple clauses are allowed. While none of the obtained results is optimal, it should be noted that in most cases, the obtained performance ratio is very close to optimality. In the case of $\mathtt{MAJ}_3$, for example, we get a ratio of 0.666158 which is extremely close to the inapproximability bound of $\frac{2}{3}$. In Section 6 we show how to obtain an optimal $\frac{2}{3}$-approximation algorithm for MAX CSP($\mathtt{MAJ}_3$) using a modified rounding procedure. It should also be noted that the performance ratio obtained for $\mathtt{NAE}_3$ is exactly the performance ratio of the MAX CUT algorithm of Goemans and Williamson [GW95] (see also [KLP96]), and that the performace ratio that we get for $\mathtt{EQU}$ and $\mathtt{SEL}$ is exactly the performance ratio of the MAX DICUT algorithm of Goemans and Williamson [GW95]. In Section 7 we obtain an improved approximation algorithm for MAX CSP($\mathtt{NAE}_3$). This algorithm is our only algorithm in which we cannot allow a variable to appear twice in a clause. Improved approximation algorithms for all other problems are mentioned in Section 8.

The $\frac{1}{2}$-approximation algorithm for MAX 3AND yields immediately a $\frac{1}{2}$-approximation algorithm for MAX 3CSP. This follows from the fact that any constraint of the form $f(z_1, z_2, z_3)$ can be translated into a set of $\mathtt{AND}_3$ constraints of which no two are simultaneously satisfiable. (In the language of Section 10, there is a 1-gadget from f to $\mathtt{AND}_3$.) If, for example, the satisfying assignments of f are $(1,0,0)$, $(0,1,1)$ and $(0,0,0)$, then the clause $f(z_1, z_2, z_3)$ can be replaced by the clauses $z_1 \wedge \bar{z}_2 \wedge \bar{z}_3$, $\bar{z}_1 \wedge z_2 \wedge z_3$ and $z_1 \wedge z_2 \wedge z_3$.

5 A $\frac{1}{2}$-approximation algorithm for MAX 3AND

As mentioned in the previous section, the performance ratio obtained by rounding an optimal solution of the semidefinite relaxation given in Figure 3 using a random hyperplane is 0. To see this, we consider a configuration v_0, v_1, v_2, v_3 in which $\theta_{ij} = \arccos(-\frac{1-\epsilon}{3})$, for every $i < j$, where ϵ is a small constant. It is not difficult to check that such a configuration is realizable. (A given set of angles can be realized as a set of angles between a set of vectors if and only if the matrix $\{\cos\theta_{ij}\}$ is positive semidefinite.) It is easy to check that $\mathit{relax}_{\mathtt{AND}_3}(v_0, v_1, v_2, v_3) = \frac{\epsilon}{4}$ and that $\mathit{prob}_{\mathtt{AND}_3}(v_0, v_1, v_2, v_3) = \mathrm{Vol}(\lambda, \lambda, \lambda, \lambda, \lambda, \lambda)$, where $\lambda = \arccos(\frac{1-\epsilon}{3})$. Using the formula

$$\mathrm{Vol}(\lambda, \lambda, \lambda, \lambda, \lambda, \lambda) \;=\; 3\int_{\cos\lambda}^{\frac{1}{3}} \frac{\arccos\left(\frac{x}{1-2x}\right)}{\sqrt{1-x^2}}\, dx \;=\; 9(\frac{1}{3} - \cos\lambda)^{3/2} - \frac{567}{40}(\frac{1}{3} - \cos\lambda)^{5/2} + \ldots\,,$$

obtained by integrating the partial derivatives $\partial\mathrm{Vol}/\partial\lambda_{ij}$, we get that $\mathit{prob}_{\mathtt{AND}_3}(v_0, v_1, v_2, v_3) \sim 9(\frac{\epsilon}{3})^{3/2}$ and therefore $\mathit{prob}_{\mathtt{AND}_3}(v_0, v_1, v_2, v_3)/\mathit{relax}_{\mathtt{AND}_3}(v_0, v_1, v_2, v_3) \sim \frac{3}{4}(\frac{\epsilon}{3})^{1/2}$. When ϵ tends to 0, the ratio tends to 0.

A ratio approaching 0 is also attained at other points. In all of them, however, $\mathit{relax}_{\mathtt{AND}_3}(v_0, v_1, v_2, v_3)$ also approaches 0 and the contribution of the corresponding clause to the objective function of the semidefinite program is negligible. For such clauses, a random assignment performs exceptionally well. This suggests the strategy outlined in the previous section. We use a rounding procedure that generates *two* assignments and then chooses the best one among them. The first assignment is simply a random assignment. The second assignment is obtained by rounding the vectors $v_0, \ldots, v_{n-1}$ using a random hyperplane. We claim that this algorithm achieves an optimal performance ratio of $\frac{1}{2}$.

Analyzing this algorithm directly is difficult. We analyze instead the following rounding procedure:

- With probability $\frac{\pi}{4}$ round the vectors $v_0, \ldots, v_{n-1}$ using a random hyperplane.
- With probability $1 - \frac{\pi}{4}$ choose a random assignment.

Clearly, the algorithm that does both things and chooses the better assignment behaves at least as well as this algorithm. A lower bound $\alpha'_{\mathtt{AND}_3}$ on the performance ratio of the algorithm that uses this combined rounding procedure can be obtained as follows:

$$\alpha'_{\mathtt{AND}_3} = \inf_{\mathit{relax}_{\mathtt{AND}_3}(v_0,v_1,v_2,v_3)>0} \frac{\frac{\pi}{4}\cdot \mathit{prob}_{\mathtt{AND}_3}(v_0, v_1, v_2, v_3) + (1-\frac{\pi}{4})\cdot\frac{1}{8}}{\mathit{relax}_{\mathtt{AND}_3}(v_0, v_1, v_2, v_3)}\,.$$

It is easy to see that $\alpha'_{\mathtt{AND}_3} \le \frac{1}{2}$. To see this we consider the configuration in which v_0, v_1, v_2, v_3 are all perpendicular to one other, i.e., $\theta_{ij} = \frac{\pi}{2}$, for every $i < j$. For this configuration we get $\mathit{prob}_{\mathtt{AND}_3}(v_0, v_1, v_2, v_3) = \frac{1}{8}$ and $\mathit{relax}_{\mathtt{AND}_3}(v_0, v_1, v_2, v_3) = \frac{1}{4}$. It follows therefore that $\alpha'_{\mathtt{AND}_3} \le \frac{1}{2}$.

Showing that $\alpha'_{\mathtt{AND}_3} = \frac{1}{2}$ is much harder. A relatively simple manipulation shows that it is equivalent to the following inequality involving the volume function:

$$\mathrm{Vol}(\lambda_{01}, \lambda_{02}, \lambda_{12}, \lambda_{03}, \lambda_{13}, \lambda_{23}) + \frac{\pi}{2}\cdot(\cos\lambda_{01} + \cos\lambda_{02} + \cos\lambda_{03}) \;\ge\; \frac{\pi^2}{8}\,,$$

for every 6-tupe of dihedral angles satisfying:

$$\begin{aligned}
\cos\lambda_{01} + \cos\lambda_{02} &\ge \cos\lambda_{13} + \cos\lambda_{23}\,,\\
\cos\lambda_{01} + \cos\lambda_{03} &\ge \cos\lambda_{12} + \cos\lambda_{23}\,,\\
\cos\lambda_{02} + \cos\lambda_{03} &\ge \cos\lambda_{12} + \cos\lambda_{13}\,.
\end{aligned}$$

We have no analytical proof of this inequality, but we have made extensive numerical experiments that strongly suggest that the inequality is valid. In fact, the point $(\lambda_{01}, \lambda_{02}, \lambda_{12}, \lambda_{03}, \lambda_{13}, \lambda_{23}) = (\frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2})$ seems to be the only point in which the inequality is tight. See the discussion in Appendix C.

6 A $\frac{2}{3}$-approximation algorithm for MAX 3MAJ

As mentioned, the performace ratio obtained for MAX 3MAJ using random hyperplane rounding is $\alpha_{\mathtt{MAJ}_3} \simeq 0.666158$. This is extremely close to the inapproximability bound of $\frac{2}{3}$ that will be presented in Section 10. The ratio $\alpha_{\mathtt{MAJ}_3} \simeq 0.666158$ is attained when $\theta_{01} = \theta_{02} = \theta_{03} \simeq 1.537234 \simeq 0.489316\pi$ and $\theta_{12} = \theta_{13} = \theta_{23} \simeq 1.604359 \simeq 0.510684\pi$.

To obtain an optimal $\frac{2}{3}$-approximation algorithm for MAX 3MAJ we use a rounding procedure similar to one used by Feige and Goemans [FG95] for MAX 2SAT and for MAX DICUT. Let $f : [0, \pi] \to [0, \pi]$ be a function satisfying $f(\pi - \theta) = \pi - f(\theta)$, for every $0 \le \theta \le \pi$, and $f(0) = 0$. Before rounding the vectors $v_0, \ldots, v_{n-1}$, we *rotate* them with respect to v_0. The vector v_i is rotated in the plane containing v_0 and v_i so that it would now form an angle of $f(\theta_{0i})$ with v_0, where θ_{0i} is the original angle between v_0 and v_i. We denote the resulting set of vectors by $v'_0, \ldots, v'_{n-1}$. The vectors $v'_0, \ldots, v'_{n-1}$, and not $v_0, \ldots, v_{n-1}$, are then rounded by choosing a random hyperplane. The Boolean variable x_i is assigned the value 1 if and only if v'_i and $v'_0 = v_0$ lie on the same side of the random hyperplane chosen.

Feige and Goemans [FG95] use rotation functions of the form:

$$f_\lambda(\theta) = (1 - \lambda) \cdot \theta + \lambda \cdot \frac{\pi}{2}(1 - \cos\theta) \ .$$

Their 0.93109-approximation algorithm for MAX 2SAT is obtained using $\lambda = 0.806765$. A $\frac{2}{3}$-approximation algorithm for MAX 3MAJ can be obtained by using $\lambda = \frac{2}{3} \cdot \frac{\pi-3}{\pi-2} \simeq 0.0826872$. This value of λ was chosen so that $f'_\lambda(\frac{\pi}{2}) = \frac{\pi}{3}$. The derivative of the function $f(\theta)$ at $\theta = \frac{\pi}{2}$ is the improtant factor here. The exact shape of the function $f(\theta)$ makes little difference. We can also use, for example, the function

$$g_{\lambda'}(\theta) = \theta - \lambda' \cdot \sin 2\theta \ ,$$

with $\lambda' = \frac{1}{2}(\frac{\pi}{3} - 1) \simeq 0.0235988$. We note that the values used for λ and λ' are fairly small and the rotation used here is 'gentle', compared to the rotation used by Feige and Goemans [FG95] for MAX 2SAT.

7 A $\frac{3}{2+\sqrt{2}}$-approximation algorithm for MAX 3NAE

An instance of MAX 3NAE can be easily converted into an equivalent instance of MAX CSP($\mathtt{XOR}_2$). Simply replace each clause $\mathtt{NAE}(z_1, z_2, z_3)$ with weight w with the three clauses $z_1 \oplus z_2$, $z_1 \oplus z_3$ and $z_2 \oplus z_3$, each with a weight of $w/2$. Thus MAX 3NAE is at least as easy to approximate as MAX CSP($\mathtt{XOR}_2$).

An instance of MAX CSP($\mathtt{XOR}_2$) can be easily transformed into an equivalent instance of MAX 3NAE. Simply replace each clause $z_1 \oplus z_2$ with the clause $\mathtt{NAE}(z_1, z_1, z_2)$. MAX 3NAE and MAX CSP($\mathtt{XOR}_2$) are therefore equivalent problems. No wonder, therefore, that the performance ratio that we got for MAX 3NAE was exactly the performance ratio that [GW95] got for MAX CUT and MAX CSP($\mathtt{XOR}_2$).

Note, however, that while casting MAX CSP($\mathtt{XOR}_2$) intances as MAX 3NAE instances, we used clauses of the form $\mathtt{NAE}(z_1, z_1, z_2)$ in which the same literal appears twice in the same clause. We now show that if clauses of the form $\mathtt{NAE}(z_1, z_1, z_2)$, where z_1 is either a literal or a constant, are not allowed, then the problem becomes slightly easier to approximate. We present a $\frac{3}{2+\sqrt{2}}$-approximation algorithm for this version. Note that $\frac{3}{2+\sqrt{2}} \simeq 0.878679$ is marginally larger that $\alpha_{\mathtt{XOR}_2} = \min \frac{2}{\pi} \frac{\theta}{1-\cos\theta} \simeq 0.878567$. Nonetheless, we think that this result is important as it may eventually be used to obtain an improved approximation algorithm for the MAX CUT problem. It immediately implies, for example, a slightly improved performance ratio for MAX CUT in graphs that contain many triangles.

To obtain the improved performance ratio, we impose the additional constraints of Figure 4 and then use the rounding procedure used in Section 5, i.e., we consider a random assignment and an assignment obtained using random hyperplane rounding and take the better one. We analyze the variant of this rounding procedure in which random hyperplane rounding is used with probability $\frac{3}{4} \cdot \frac{\pi}{\sqrt{2}+1} \simeq 0.975968$, and a random assignment is used with probability $1-\frac{3}{4} \cdot \frac{\pi}{\sqrt{2}+1} \simeq 0.024032$. For brevity, we let $\epsilon = 1-\frac{3}{4} \cdot \frac{\pi}{\sqrt{2}+1}$. A lower bound on the performance ratio obtained using this combined rounding procedure is

$$\alpha'_{\mathtt{NAE}_3} = \inf \frac{(1-\epsilon) \cdot \dfrac{\theta_{12}+\theta_{13}+\theta_{23}}{2\pi} + \epsilon \cdot \dfrac{3}{4}}{\dfrac{3-\cos\theta_{12}-\cos\theta_{13}-\cos\theta_{23}}{4}} ,$$

where the infimum is over all $0 \le \theta_{01}, \theta_{02}, \theta_{12} \le \pi$ such that $\cos\theta_{12}+\cos\theta_{13}+\cos\theta_{23} \ge -1$, $\cos\theta_{12} - \cos\theta_{13} - \cos\theta_{23} \ge -1$, $-\cos\theta_{12}+\cos\theta_{13}-\cos\theta_{23} \ge -1$ and $-\cos\theta_{12}-\cos\theta_{13}+\cos\theta_{23} \ge -1$. As this expression does not involve volumes, we are able this time to show analylically that $\alpha'_{\mathtt{NAE}_3} = \frac{3}{2+\sqrt{2}}$. See Appendix D. The worst ratio is attained when $\theta_{12}=\theta_{13}=\frac{3\pi}{4}$ and $\theta_{23}=0$.

8 Other approximation algorithms

As MAX CSP($\mathtt{AXR}_3$) and MAX CSP($\mathtt{TWO}_3$) are subproblems of MAX 3CSP, we get an optinal performance ratio of $\frac{1}{2}$ for these two problems. An improved performance ratio of 0.859 for MAX CSP($\mathtt{SEL}_3$) is obtained by reducing an instance of this problem into an equivalent instance of MAX 2CSP. A clause $\mathtt{SEL}(z_1, z_2, z_3) = z_1\bar{z}_3 \vee z_1 z_2$ is simply replaced by the two clauses $z_1 \wedge \bar{z}_3$ and $z_2 \wedge z_3$. Note that these two clauses cannot be satisfied simultaneously.

Slightly improved performance ratios for $\mathtt{AOR}_3$, $\mathtt{AOA}_3$, $\mathtt{XAD}_3$, $\mathtt{OAD}_3$ and $\mathtt{XOA}_3$ can be obtained using rotations, as described in Section 6. The exact details will appear in the full version of the paper.

It is interesting to note that the performance ratio obtained for $\mathtt{XAD}_3$ using random hyperplane rounding is the same as that obtained for $\mathtt{MAJ}_3$. While for $\mathtt{MAJ}_3$, we were able to improve the performance ratio to the optimal value of $\frac{2}{3}$ using rotations, this does not seem to work for $\mathtt{XAD}_3$.

Another interesting observation is that the performance ratios obtained using random hyperplane rounding for $\mathtt{AOR}_3$ and $\mathtt{AOA}_3$, and for $\mathtt{OAD}_3$ and $\mathtt{XOA}_3$ are the same, although we have not been able to show the equivalence of the corresponding problems. Slightly different performance ratios are obtained for these problems when rotations are considered.

9 Approximating satisfiable instances

An instance of MAX CSP(f) is said to be *satisfiable* if there is an assignment that satisfies all the clauses. Satisfiable instances are sometimes easier to solve. Given a satisfiable instance of MAX 2CSP, for example, we can find a satisfiable assignment in polynomial time. For some other problems, like MAX 3SAT, finding a satisfiable assignment remains NP-hard. In fact, satisfiable instances of MAX 3SAT are as hard to approximate as general MAX 3SAT instances. Sometimes, the problem remains NP-hard but becomes easier to approximate.

In this section we build on nice ideas of Trevisan [Tre97] and obtain an improved approximation algorithm for *satisfiable* instances of MAX 3CSP. It is not difficult to see that clauses involving $\mathtt{AND}_3$, $\mathtt{EQU}_3$, $\mathtt{AXR}_3$, $\mathtt{AOR}_3$ and $\mathtt{AOA}_3$, as well as those involving $\mathtt{AND}_2$ and $\mathtt{XOR}_2$, can be easily removed from satisfiable instances. If a satisfiable instance includes, for example, a clause $\mathtt{AXR}(z_1, z_2, z_3)$, we can immediately deduce that $z_1 = 1$ and that $z_3 = \bar{z}_2$. We can therefore replace all the occurrences of the literal z_1 by 1, and all the occurrences

algorithm	TWO	$\mathtt{XOR}_3$	MAJ	XAD	SEL	OAD	XOA	NTW	$\mathtt{OR}_2$	NAE	OXR	$\mathtt{OR}_3$
SDP	0.649	$\frac{1}{2}$	0.736	0.736	$\frac{5}{6}$	0.824	0.824	$\frac{5}{8}$	0.912	0.912	$\frac{3}{4}$	$\frac{7}{8}$
RGE	$\frac{3}{4}$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{5}{8}$	$\frac{5}{8}$	$\frac{5}{8}$	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{7}{8}$

Figure 6: Approximation satisfiable instances of MAX 3CSP.

of the literal $\bar{z}_1$ by 0. Similarly, we can replace all the occurrences of z_3 by $\bar{z}_2$ and all the occurrences of $\bar{z}_3$ by z_2. We may therefore concentrate on the other types of constraints.

A lower bound β_f on the performance ratio of the algorithm obtained by solving the semidefinite relaxation of MAX CSP(f) and then rounding the vectors using a random hyperplane can be obtained as follows:

$$\beta_f \;=\; \inf_{relax_f(v_0,v_1,v_2,v_3)=1} prob_f(v_0,v_1,v_2,v_3)\;.$$

The values of β_f for the functions f that we are interested in appear in the second row of Figure 6. All these bounds have closed forms that are given in Appendix E. We mention here in passing that many of these bounds can be improved using a more sophisticated rounding procedure. We do not have space to elaborate on this here. It is important to note that the performance ratios given in the second row are attained *simultaneously*, using the same rounding procedure.

A quick look at Figure 6 reveals that for any constraint type, except $\mathtt{XOR}_3$, we obtain a ratio of at least $\frac{5}{8}$. If all the constraints are of this type, the problem is easy, as we can find a satisfying assignment using *Gaussian elimination*. What happens, however, if we have a mixture of constraint types? Trevisan [Tre97] suggests the following approach: Use Gaussian elimination to find a basis of the affine linear subspace containing all the solutions to the $\mathtt{XOR}_3$-type constraints. Here we supplement this idea with the observation that for any clause of the form $\mathtt{TWO}(z_1,z_2,z_3)$, we can add to the linear system the constraint $z_1\oplus z_2\oplus z_3=0$. We now consider each clause $f(z_1,z_2,z_3)$ of the instance. If there exists a satifying assignment of $f(z_1,z_2,z_3)$ which is inconsistent with the linear subspace of solutions, i.e., if no point in the subspace yields this assignment to z_1,z_2 and z_3, then the clause $f(z_1,z_2,z_3)$ can be simplified. For example, if the clause is $\mathtt{NTW}(z_1,z_2,z_3)$ and in no point of the solution subspace we have $(z_1,z_2,z_3)=(0,0,0)$, we can replace the constraint $\mathtt{NTW}(z_1,z_2,z_3)$ with the constraint $\mathtt{XOR}_3(z_1,z_2,z_3)$. We can now add a new equation to the linear system and obtain, perhaps, a slightly smaller space of solutions. We repeat this process until all satisfiable assignments of all the clauses of the instance are consistent with the affine subspace of solutions. We then pick a *random* point of this solution subspace. It is not difficult to verify that the different types of constraints are now satisfied with the probabilities given in the third row of Figure 6. The important things to note is that all the $\mathtt{XOR}_3$-type clauses are now satisfied, and that each $\mathtt{TWO}(z_1,z_2,z_3)$ clause is satisfied with probability $\frac{3}{4}$, as in each assignment generated we have $z_1\oplus z_2\oplus z_3=0$. All other clauses are satisfied with exactly the same probability that they would have been satisfied, had the assignment been chosen completely at random, and not from the affine subspace.

We thus have two approximation algorithms whose performace ratios are given in Figure 6. We obtain a combined algorithm by running both algorithms, comparing the quality of the two assignments produced and choosing the better one. We claim that the performance ratio of this combined algorithm, for satisfied clauses of MAX 3CSP, is $\frac{5}{8}$. This follows from the fact that the performance ratio of this algorithm is at least as good as the performace ratio of the algorithm that runs SDP, our first algorithm, with probability $\frac{3}{4}$, and algorithm RGE, our second algorithm, with probability $\frac{1}{4}$. It is interesting to note that the hardest satisfiable instances of MAX 3CSP turn out to be instances in which all clauses are $\mathtt{NTW}_3$ clauses.

10 Hardness results

The inapproximability bounds given in Figure 1 and 5 are obtained using *gadgets* (see [BGS95] and [TSSW96]). Most of these gadgets are new. They are presented in Appendix F. The larger inapproximability bounds given for $\mathtt{SEL}_3$ and $\mathtt{NAE}_3$ in Figure 5 correspond to the case in which only simple clauses are allowed.

11 Concluding remarks

We find it remarkable that semidefinite programming yields optimal approximation algorithms for so many constraint satisfaction problems that involve at most three variables per constraint. Many interesting and challenging open problems still remain, however. The most important ones are perhaps improving either the preformance guarantees, or the inapproximability bounds, for MAX CUT and MAX 2SAT.

Finally, we note again that most performance ratios claimed in this paper rely on certain inequalities involving the volume function of spherical tetrahedra (see the bottom of page 7 for an example of such an inequality). While compelling analytical and numerical evidence convinced us of the validity of these inequalities, this work will not be complete without complete analytical proofs of these inequalities. We believe that such proofs can be obtained and we are working in that direction.

Acknowledgment

We would like to thank Greg Sorkin for supplying some of the gadgets presented in Appendix F.

References

[ALM+92] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and hardness of approximation problems. In *Proceedings of the 33rd Annual IEEE Symposium on Foundations of Computer Science, Pittsburgh, Pennsylvania*, pages 14–23, 1992.

[AOH96] T. Asano, T. Ono, and T. Hirata. Approximation algorithms for the maximum satisfiability problem. In *Proceedings of the 5th Scandinavian workshop on algorithm Theory*, pages 100–111, 1996.

[AS92] S. Arora and M. Safra. Probabilistic checking of proofs: a new characterization of NP. In *Proceedings of the 33rd Annual IEEE Symposium on Foundations of Computer Science, Pittsburgh, Pennsylvania*, pages 2–13, 1992.

[Asa97] T. Asano. Approximation algorithms for MAX SAT: Yannakakis vs. Goemans-Williamson. In *Proceedings of the 3nd Israel Symposium on Theory and Computing Systems, Ramat Gan, Israel*, 1997.

[BGLR93] M. Bellare, S. Goldwasser, C. Lund, and A. Russell. Efficient probabilistically checkable proofs and applications to approximation. In *Proceedings of the 25rd Annual ACM Symposium on Theory of Computing, San Diego, California*, pages 294–304, 1993. See Errata in STOC'94.

[BGS95] M. Bellare, O. Goldreich, and M. Sudan. Free bits, PCPs and non-approximability—towards tight results. In *Proceedings of the 36rd Annual IEEE Symposium on Foundations of Computer*

Science, Milwaukee, Wisconsin, pages 422–431, 1995. Full version available as E-CCC Technical Report number TR95-024.

[BH81] J. Böhm and H. Hertel. *Polyedergeometrie in n-dimensionalen Räumen konstanter Krümmung.* Birkhäuser, 1981.

[CFZ97] J. Chen, D. Friesen, and H. Zheng. Tight bound on Johnson's algorithm for max-SAT. In *Proceedings of the 12th Annual IEEE Conference on Computational Complexity, Ulm, Germany*, 1997. To appear.

[Cox35] H.S.M. Coxeter. The functions of Schläfli and Lobatschefsky. *Quarterly Journal of of Mathematics (Oxford)*, 6:13–29, 1935.

[Cox57] H.S.M. Coxeter. *Non-Euclidean Geometry.* The University of Toronto Press, 1957.

[Cre95] N. Creignou. A dichotomy theorem for maximum generalized satisfiability problems. *Journal of Computer and System Sciences*, 51:511–522, 1995.

[CT96] P. Crescenzi and L. Trevisan. Max NP-completeness made easy. Manuscript, 1996.

[FG95] U. Feige and M.X. Goemans. Approximating the value of two prover proof systems, with applications to MAX-2SAT and MAX-DICUT. In *Proceedings of the 3nd Israel Symposium on Theory and Computing Systems, Tel Aviv, Israel*, pages 182–189, 1995.

[FGL+96] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Interactive proofs and the hardness of approximating cliques. *Journal of the ACM*, 43:268–292, 1996.

[GJ79] M.R. Garey and D.S. Johnson. *Computers and Intractability, a guide to the Theory of NP-Completeness.* W.H. Freeman and Company, 1979.

[GW94] M.X. Goemans and D.P. Williamson. New 3/4-approximation algorithms for the maximum satisfiability problem. *SIAM Journal on Discrete Mathematics*, 7:656–666, 1994.

[GW95] M.X. Goemans and D.P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115–1145, 1995.

[Hås97] J. Håstad. Some optimal inapproximability results. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing, El Paso, Texas*, pages 1–10, 1997.

[Hsi88] W.Y. Hsiang. On infinitesiaml symmetrization and volume formula for spherical or hyperbolic tetrahedrons. *Quarterly Journal of Mathematics (Oxford)*, 39:463–468, 1988.

[Hsi92] W.Y. Hsiang. On the development of quantitative geometry from Pythagoras to Grassmann. In D.-Z. Du and F.K. Hwang, editors, *Computing in Euclidean Geometry*, pages 1–21. World Scientific Publishing Co., 1992.

[Joh74] D.S. Johnson. Approximation algorithms for combinatorical problems. *Journal of Computer and System Sciences*, 9:256–278, 1974.

[Kar96] H. Karloff. How good is the Goemans-Williamson MAX CUT algorithm? In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing, Philadelphia, Pennsylvania*, pages 427–434, 1996.

[Kel91] R. Kellerhals. The dilogarithm and volumes of hyperpolic polytopes. In L. Lewin, editor, *Structural Properties of Polylogarithms*, Mathematical Surveys and Monographs, Volume 37, pages 301–336. American Mathematical Society, 1991.

[KLP96] V. Kann, J. Lagergren, and A. Panconesi. Approximability of maximum splitting of k-sets and some other APX-complete problems. *Information Processing Letters*, 58:105–110, 1996.

[KMSV94] S. Khanna, R. Motwani, M. Sudan, and U. Vazirani. On syntactic versus computational views of approximability. In *Proceedings of the 35rd Annual IEEE Symposium on Foundations of Computer Science, Santa Fe, New Mexico*, pages 819–830, 1994.

[KST97] S. Khanna, M. Sudan, and L. Trevisan. Constraint satisfaction: The approximability of minimization problems. In *Proceedings of the 12th Annual IEEE Conference on Computational Complexity, Ulm, Germany*, pages 11–20, 1997. Full version available as E-CCC Technical Report number TR96-064.

[KSW97] S. Khanna, M. Sudan, and D.P. Williamson. A complete classification of the approximability of maximization problems derived from Boolean constraint satisfaction. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing, El Paso, Texas*, 1997. To appear. Full version available as E-CCC Technical Report number TR96-062.

[KZ97] H. Karloff and U. Zwick. A $(7/8-\epsilon)$-approximation algorithm for MAX 3SAT? In *Proceedings of the 38rd Annual IEEE Symposium on Foundations of Computer Science, Miami Beach, Florida*, 1997. To appear.

[MR95] S. Mahajan and H. Ramesh. Derandomizing semidefinite programming based approximation algorithms. In *Proceedings of the 36rd Annual IEEE Symposium on Foundations of Computer Science, Milwaukee, Wisconsin*, pages 162–169, 1995.

[Pet94] E. Petrank. The hardness of approximation: Gap location. *Computational Complexity*, 4:133–157, 1994.

[PY91] C.H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43:425–440, 1991.

[Sch58] L. Schläfli. On the multiple integral $\int^n dx\, dy \ldots dz$, whose limits are $p_1 = a_1x + b_1y + \ldots + h_1z > 0, p_2 > 0, \ldots, p_n > 0$, and $x^2 + y^2 + \ldots + z^2 < 1$. *Quarterly Journal of Mathematics (Oxford)*, 2:269–300, 1858. Continued in Vol. 3 (1860), pp. 54–68 and pp. 97-108.

[Sch78] T.J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing, San Diego, California*, pages 216–226, 1978.

[Tre97] L. Trevisan. Approximating satisfiable satisfiability problems. In *Proceedings of the 5th European Symposium on Algorithms, Graz, Austria*, 1997. To appear.

[TSSW96] L. Trevisan, G.B. Sorkin, M. Sudan, and D.P. Williamson. Gadgets, approximation, and linear programming (extended abstract). In *Proceedings of the 37rd Annual IEEE Symposium on Foundations of Computer Science, Burlington, Vermont*, pages 617–626, 1996.

[Yan94] M. Yannakakis. On the approximation of maximum satisfiability. *Journal of Algorithms*, 17:475–502, 1994.

A Semidefinite relaxations

The 8 vertices of the polytope *polytope*($\texttt{AND}_3$), for example, are given in Figure 7. (The vertices are the 7-tuples in the righthand side of the table.) The 8 facets of *polytope*($\texttt{AND}_3$) are given in Figure 8. Note that the last 4 facets give upper bounds on z, and they are exactly the bounds used in the relaxation of MAX 3AND, with y_{ij} replaced by $v_i \cdot v_j$. The first 4 facets give lower bounds on z and they are therefore discarded. (We could include them in the relaxation but there is no need to.) The facets that give lower bounds on z can be eliminated automatically by adding the cone vector $(0,0,0,0,0,0,-1)$. We denote the resulting polytope by *polytope*$'$($\texttt{AND}_3$). The 20 facets of *polytope*$'$($\texttt{AND}_3$) are given in Figure 9. It is interesting to note that the first 16 facets of *polytope*$'$($\texttt{AND}_3$) are exactly the additional constraints given in Figure 4.

The relaxations of all the other types of constraints are obtained in a similar manner. The relaxations of the identity function $\texttt{IDN}_1(x_1) = x_1$, of $\texttt{OR}_2$, $\texttt{XOR}_2$, $\texttt{AND}_2$ and of $\texttt{EQU}_3$, $\texttt{SEL}_3$ and $\texttt{NAE}_3$ are especially simple. Only one facet of the corresponding polytope supplies an upper bound on z. The obtained relaxations are:

$$
\begin{aligned}
relax_{\texttt{IDN}_1}(v_0,v_1) &= \frac{1+v_0\cdot v_1}{2} \\
relax_{\texttt{XOR}_2}(v_0,v_1,v_2) &= \frac{1-v_1\cdot v_2}{2} \\
relax_{\texttt{OR}_2}(v_0,v_1,v_2) &= \frac{3+v_0\cdot v_1+v_0\cdot v_2-v_1\cdot v_2}{4} = 1-\frac{(v_0-v_1)\cdot(v_0-v_2)}{4} \\
relax_{\texttt{AND}_2}(v_0,v_1,v_2) &= \frac{1+v_0\cdot v_1+v_0\cdot v_2+v_1\cdot v_2}{4} = \frac{(v_0+v_1)\cdot(v_0+v_2)}{4} \\
relax_{\texttt{EQU}_3}(v_0,v_1,v_2,v_3) &= \frac{1+v_1\cdot v_2+v_1\cdot v_3+v_2\cdot v_3}{4} = \frac{(v_1+v_2)\cdot(v_1+v_3)}{4} \\
relax_{\texttt{NAE}_3}(v_0,v_1,v_2,v_3) &= \frac{3-v_1\cdot v_2-v_1\cdot v_3-v_2\cdot v_3}{4} = 1-\frac{(v_1+v_2)\cdot(v_1+v_3)}{4} \\
relax_{\texttt{SEL}_3}(v_0,v_1,v_2,v_3) &= \frac{2+v_0\cdot v_1+v_0\cdot v_2-v_1\cdot v_3+v_2\cdot v_3}{4} = \frac{2+v_0\cdot(v_1+v_2)-v_3\cdot(v_1-v_2)}{4}
\end{aligned}
$$

The relaxations of $\texttt{IDN}_1$, $\texttt{OR}_2$, $\texttt{XOR}_2$ and $\texttt{AND}_2$ are identical to the ones used in [GW95]. The relaxation of $\texttt{NAE}_3$ is identical to the one used in [KLP96]. All other relaxations are the minimum of four bounds, as is the case for $\texttt{AND}_3$. The facets of *polytope*(f) used in these relaxations are given in Figures 10 to 21. We thus get, for example,

$$
\begin{aligned}
relax_{\texttt{AND}_3}(v_0,v_1,v_2,v_3) &= \min\left\{\begin{array}{cc} \frac{(v_0+v_1)\cdot(v_0+v_2)}{4}, & \frac{(v_0+v_1)\cdot(v_0+v_3)}{4} \\ \frac{(v_0+v_1)\cdot(v_0+v_3)}{4}, & \frac{(v_1+v_2)\cdot(v_1+v_3)}{4} \end{array}\right\}, \\
relax_{\texttt{XOR}_3}(v_0,v_1,v_2,v_3) &= \min\left\{\begin{array}{cc} \frac{\|-v_0+v_1+v_2+v_3\|^2-2}{2}, & \frac{\|v_0-v_1+v_2+v_3\|^2-2}{2} \\ \frac{\|v_0+v_1-v_2+v_3\|^2-2}{2}, & \frac{\|v_0+v_1+v_2-v_3\|^2-2}{2} \end{array}\right\}, \\
relax_{\texttt{MAJ}_3}(v_0,v_1,v_2,v_3) &= \min\left\{\begin{array}{cc} \frac{3+2v_0\cdot v_1-v_1\cdot v_2-v_1\cdot v_3+v_2\cdot v_3}{4}, & \frac{3+2v_0\cdot v_2-v_1\cdot v_2+v_1\cdot v_3-v_2\cdot v_3}{4} \\ \frac{3+v_1\cdot v_2+2v_0\cdot v_3-v_1\cdot v_3-v_2\cdot v_3}{4}, & \frac{2+2v_0\cdot(v_1+v_2+v_3)+(v_1+v_2)\cdot(v_1+v_3)}{4} \end{array}\right\}, \\
relax_{\texttt{OR}_3}(v_0,v_1,v_2,v_3) &= \min\left\{\begin{array}{cc} \frac{4+(v_0-v_1)\cdot(v_2+v_3)}{4}, & \frac{4+(v_0-v_2)\cdot(v_1+v_3)}{4} \\ \frac{4+(v_0-v_3)\cdot(v_1+v_2)}{4}, & 1 \end{array}\right\}.
\end{aligned}
$$

It is interesting to note that the relaxation of $\texttt{AND}_3(x_1,x_2,x_3)$ is simply the minimum of the relaxations of $x_1 \wedge x_2$, $x_1 \wedge x_3$, $x_2 \wedge x_3$ and $\texttt{EQU}(x_1,x_2,x_3)$. The relaxation of $\texttt{XOR}_3$ is also fairly intuitive.

x_1	x_2	x_3	x_0x_1	x_0x_2	x_1x_2	x_0x_3	x_1x_3	x_2x_3	f
-1	-1	-1	-1	-1	$+1$	-1	$+1$	$+1$	0
-1	-1	$+1$	-1	-1	$+1$	$+1$	-1	-1	0
-1	$+1$	-1	-1	$+1$	-1	-1	$+1$	-1	0
-1	$+1$	$+1$	-1	$+1$	-1	$+1$	-1	$+1$	0
$+1$	-1	-1	$+1$	-1	-1	-1	-1	$+1$	0
$+1$	-1	$+1$	$+1$	-1	-1	$+1$	$+1$	-1	0
$+1$	$+1$	-1	$+1$	$+1$	$+1$	-1	-1	-1	0
$+1$	$+1$	$+1$	$+1$	$+1$	$+1$	$+1$	$+1$	$+1$	1

Figure 7: The 8 vertices of *polytope*(`AND`$_3$).

						$-z$	$\leq$	0
$+y_{01}$	$+y_{02}$			$+y_{13}$	$+y_{23}$	$-4z$	$\leq$	0
$+y_{01}$		$+y_{12}$	$+y_{03}$		$+y_{23}$	$-4z$	$\leq$	0
	$+y_{02}$	$+y_{12}$	$+y_{03}$	$+y_{13}$		$-4z$	$\leq$	0
$-y_{01}$	$-y_{02}$	$-y_{12}$				$+4z$	$\leq$	1
$-y_{01}$			$-y_{03}$	$-y_{13}$		$+4z$	$\leq$	1
	$-y_{02}$		$-y_{03}$		$-y_{23}$	$+4z$	$\leq$	1
		$-y_{12}$		$-y_{13}$	$-y_{23}$	$+4z$	$\leq$	1

Figure 8: The facets of *polytope*(`AND`$_3$).

$-y_{01}$	$+y_{02}$	$+y_{12}$					$\leq$	1
$+y_{01}$	$-y_{02}$	$+y_{12}$					$\leq$	1
$+y_{01}$	$+y_{02}$	$-y_{12}$					$\leq$	1
$-y_{01}$	$-y_{02}$	$-y_{12}$					$\leq$	1
$-y_{01}$			$+y_{03}$	$+y_{13}$			$\leq$	1
$+y_{01}$			$-y_{03}$	$+y_{13}$			$\leq$	1
$+y_{01}$			$+y_{03}$	$-y_{13}$			$\leq$	1
$-y_{01}$			$-y_{03}$	$-y_{13}$			$\leq$	1
	$-y_{02}$		$+y_{03}$		$+y_{23}$		$\leq$	1
	$+y_{02}$		$-y_{03}$		$+y_{23}$		$\leq$	1
	$+y_{02}$		$+y_{03}$		$-y_{23}$		$\leq$	1
	$-y_{02}$		$-y_{03}$		$-y_{23}$		$\leq$	1
		$-y_{12}$		$+y_{13}$	$+y_{23}$		$\leq$	1
		$+y_{12}$		$-y_{13}$	$+y_{23}$		$\leq$	1
		$+y_{12}$		$+y_{13}$	$-y_{23}$		$\leq$	1
		$-y_{12}$		$-y_{13}$	$-y_{23}$		$\leq$	1
$-y_{01}$	$-y_{02}$	$-y_{12}$				$+4z$	$\leq$	1
$-y_{01}$			$-y_{03}$	$-y_{13}$		$+4z$	$\leq$	1
	$-y_{02}$		$-y_{03}$		$-y_{23}$	$+4z$	$\leq$	1
		$-y_{12}$		$-y_{13}$	$-y_{23}$	$+4z$	$\leq$	1

Figure 9: The facets of *polytope*$'$(`AND`$_3$).

	$+y_{02}$	$-y_{12}$	$-y_{03}$	$+y_{13}$	$+2y_{23}$	$+4z$	$\leq$	2
	$-y_{02}$	$+y_{12}$	$+y_{03}$	$-y_{13}$	$+2y_{23}$	$+4z$	$\leq$	2
$-2y_{01}$	$+y_{02}$	$+y_{12}$	$+y_{03}$	$+y_{13}$		$+4z$	$\leq$	2
$-2y_{01}$	$-y_{02}$	$-y_{12}$	$-y_{03}$	$-y_{13}$		$+4z$	$\leq$	2

Figure 10: The facets of *polytope*($\texttt{AXR}_3$) used in the relaxation.

$-y_{01}$						$+2z$	$\leq$	1
$-y_{01}$		$-y_{12}$	$-y_{03}$		$+y_{23}$	$+4z$	$\leq$	2
$-y_{01}$	$-y_{02}$			$-y_{13}$	$+y_{23}$	$+4z$	$\leq$	2
$-2y_{01}$	$-y_{02}$	$-y_{12}$	$-y_{03}$	$-y_{13}$		$+4z$	$\leq$	2

Figure 11: The facets of *polytope*($\texttt{AOR}_3$) used in the relaxation.

					$-y_{23}$	$+2z$	$\leq$	1
$-y_{01}$		$+y_{12}$	$-y_{03}$		$-y_{23}$	$+4z$	$\leq$	2
$-y_{01}$	$-y_{02}$			$+y_{13}$	$-y_{23}$	$+4z$	$\leq$	2
	$-y_{02}$	$+y_{12}$	$-y_{03}$	$+y_{13}$	$-2y_{23}$	$+4z$	$\leq$	2

Figure 12: The facets of *polytope*($\texttt{AOA}_3$) used in the relaxation.

$+y_{01}$	$+y_{02}$	$-y_{12}$	$-2y_{03}$	$+2y_{13}$	$+2y_{23}$	$+4z$	$\leq$	3
$+y_{01}$	$-2y_{02}$	$+2y_{12}$	$+y_{03}$	$-y_{13}$	$+2y_{23}$	$+4z$	$\leq$	3
$-2y_{01}$	$+y_{02}$	$+2y_{12}$	$+y_{03}$	$+2y_{13}$	$-y_{23}$	$+4z$	$\leq$	3
$-2y_{01}$	$-2y_{02}$	$-y_{12}$	$-2y_{03}$	$-y_{13}$	$-y_{23}$	$+4z$	$\leq$	3

Figure 13: The facets of *polytope*($\texttt{TWO}_3$) used in the relaxation.

$+y_{01}$	$-y_{02}$	$+y_{12}$	$-y_{03}$	$+y_{13}$	$-y_{23}$	$+2z$	$\leq$	2
$-y_{01}$	$+y_{02}$	$+y_{12}$	$-y_{03}$	$-y_{13}$	$+y_{23}$	$+2z$	$\leq$	2
$-y_{01}$	$-y_{02}$	$-y_{12}$	$+y_{03}$	$+y_{13}$	$+y_{23}$	$+2z$	$\leq$	2
$+y_{01}$	$+y_{02}$	$-y_{12}$	$+y_{03}$	$-y_{13}$	$-y_{23}$	$+2z$	$\leq$	2

Figure 14: The facets of *polytope*($\texttt{XOR}_3$) used in the relaxation.

		$-y_{12}$	$-2y_{03}$	$+y_{13}$	$+y_{23}$	$+4z$	$\leq$	3
	$-2y_{02}$	$+y_{12}$		$-y_{13}$	$+y_{23}$	$+4z$	$\leq$	3
$-2y_{01}$		$+y_{12}$		$+y_{13}$	$-y_{23}$	$+4z$	$\leq$	3
$-2y_{01}$	$-2y_{02}$	$-y_{12}$	$-2y_{03}$	$-y_{13}$	$-y_{23}$	$+4z$	$\leq$	3

Figure 15: The facets of *polytope*($\texttt{MAJ}_3$) used in the relaxation.

	$+y_{02}$		$-y_{03}$	$+2y_{13}$	$+y_{23}$	$+4z$	$\leq$	3
	$-y_{02}$	$+2y_{12}$	$+y_{03}$		$+y_{23}$	$+4z$	$\leq$	3
$-2y_{01}$	$-y_{02}$		$-y_{03}$		$-y_{23}$	$+4z$	$\leq$	3
$-2y_{01}$	$+y_{02}$	$+2y_{12}$	$+y_{03}$	$+2y_{13}$	$-y_{23}$	$+4z$	$\leq$	3

Figure 16: The facets of *polytope*(`XAD`$_3$) used in the relaxation.

$-y_{01}$			$-y_{03}$	$+y_{13}$		$+4z$	$\leq$	3
$-y_{01}$	$-y_{02}$	$+y_{12}$				$+4z$	$\leq$	3
$-2y_{01}$		$+y_{12}$		$+y_{13}$	$-y_{23}$	$+4z$	$\leq$	3
$-2y_{01}$	$-y_{02}$		$-y_{03}$		$-y_{23}$	$+4z$	$\leq$	3

Figure 17: The facets of *polytope*(`OAD`$_3$) used in the relaxation.

$-y_{01}$	$-y_{02}$	$+y_{12}$				$+4z$	$\leq$	3
	$-y_{02}$	$+2y_{12}$	$-y_{03}$		$-y_{23}$	$+4z$	$\leq$	3
$-y_{01}$		$+2y_{12}$	$-y_{03}$	$-y_{13}$		$+4z$	$\leq$	3
		$+y_{12}$		$-y_{13}$	$-y_{23}$	$+4z$	$\leq$	3

Figure 18: The facets of *polytope*(`XOA`$_3$) used in the relaxation.

$+y_{01}$	$+y_{02}$	$-y_{12}$	$+y_{03}$	$-y_{13}$	$-y_{23}$	$+2z$	$\leq$	2
$+2y_{01}$	$-y_{02}$	$+y_{12}$	$-y_{03}$	$+y_{13}$	$-2y_{23}$	$+4z$	$\leq$	4
$-y_{01}$	$+2y_{02}$	$+y_{12}$	$-y_{03}$	$-2y_{13}$	$+y_{23}$	$+4z$	$\leq$	4
$-y_{01}$	$-y_{02}$	$-2y_{12}$	$+2y_{03}$	$+y_{13}$	$+y_{23}$	$+4z$	$\leq$	4

Figure 19: The facets of *polytope*(`NTW`$_3$) used in the relaxation.

	$-y_{02}$	$+y_{12}$	$-y_{03}$	$+y_{13}$		$+4z$	$\leq$	4
$-2y_{01}$	$+y_{02}$	$+y_{12}$	$-y_{03}$	$-y_{13}$	$+2y_{23}$	$+4z$	$\leq$	4
$-2y_{01}$	$-y_{02}$	$-y_{12}$	$+y_{03}$	$+y_{13}$	$+2y_{23}$	$+4z$	$\leq$	4
	$+y_{02}$	$-y_{12}$	$+y_{03}$	$-y_{13}$		$+4z$	$\leq$	4

Figure 20: The facets of *polytope*(`OXR`$_3$) used in the relaxation.

						$+z$	$\leq$	1
	$-y_{02}$	$+y_{12}$	$-y_{03}$	$+y_{13}$		$+4z$	$\leq$	4
$-y_{01}$		$+y_{12}$	$-y_{03}$		$+y_{23}$	$+4z$	$\leq$	4
$-y_{01}$	$-y_{02}$			$+y_{13}$	$+y_{23}$	$+4z$	$\leq$	4

Figure 21: The facets of *polytope*(`OR`$_3$) used in the relaxation.

B The volume of spherical tetrahedra

A spherical tetrahedron is characterized by either its six lengths $\theta_{01}, \theta_{02}, \theta_{12}, \theta_{03}, \theta_{13}, \theta_{23}$, or its six dihedral angles $\lambda_{01}, \lambda_{02}, \lambda_{12}, \lambda_{03}, \lambda_{13}, \lambda_{23}$. The lengths and the dihedral angles are related by the following relations; if i, j, k, ℓ is a permutation of $0, 1, 2, 3$ then:

$$\cos\lambda_{ij} = \frac{\begin{vmatrix} 1 & \cos\theta_{ij} & \cos\theta_{i\ell} \\ \cos\theta_{ji} & 1 & \cos\theta_{j\ell} \\ \cos\theta_{ki} & \cos\theta_{kj} & \cos\theta_{k\ell} \end{vmatrix}}{\begin{vmatrix} 1 & \cos\theta_{ij} & \cos\theta_{ik} \\ \cos\theta_{ji} & 1 & \cos\theta_{jk} \\ \cos\theta_{ki} & \cos\theta_{kj} & 1 \end{vmatrix}^{1/2} \cdot \begin{vmatrix} 1 & \cos\theta_{ij} & \cos\theta_{i\ell} \\ \cos\theta_{ji} & 1 & \cos\theta_{j\ell} \\ \cos\theta_{\ell i} & \cos\theta_{\ell j} & 1 \end{vmatrix}^{1/2}},$$

$$\cos\theta_{ij} = -\frac{\begin{vmatrix} 1 & -\cos\lambda_{ij} & -\cos\lambda_{ki} \\ -\cos\lambda_{ij} & 1 & -\cos\lambda_{\ell i} \\ -\cos\lambda_{kj} & -\cos\lambda_{\ell j} & -\cos\lambda_{\ell k} \end{vmatrix}}{\begin{vmatrix} 1 & -\cos\lambda_{ij} & -\cos\lambda_{kj} \\ -\cos\lambda_{ij} & 1 & -\cos\lambda_{\ell j} \\ -\cos\lambda_{kj} & -\cos\lambda_{\ell j} & 1 \end{vmatrix}^{1/2} \cdot \begin{vmatrix} 1 & -\cos\lambda_{ij} & -\cos\lambda_{ki} \\ -\cos\lambda_{ij} & 1 & -\cos\lambda_{\ell i} \\ -\cos\lambda_{ki} & -\cos\lambda_{\ell i} & 1 \end{vmatrix}^{1/2}},$$

where it is assumed that $\theta_{ij} = \theta_{ji}$ and that $\lambda_{ij} = \lambda_{ji}$ for every $i \neq j$.

We let $\mathrm{Vol}(\lambda_{01}, \lambda_{02}, \lambda_{12}, \lambda_{03}, \lambda_{13}, \lambda_{23})$ denote the volume of a spherical tetrahedron with dihedral angles $\lambda_{01}, \lambda_{02}, \lambda_{12}, \lambda_{03}, \lambda_{13}, \lambda_{23}$. Although the function $\mathrm{Vol}(\lambda_{01}, \lambda_{02}, \lambda_{12}, \lambda_{03}, \lambda_{13}, \lambda_{23})$, giving the volume of a spherical tetrahedron as a function of its six dihedral angles, is a complicated function, its partial derivatives have a surprisingly simple form:

Theorem B.1 (Schläfli (1858) [Sch58]) *For every* $0 \le i < j \le 3$ *we have* $\dfrac{\partial \mathrm{Vol}}{\partial \lambda_{ij}} = \dfrac{\theta_{ij}}{2}$.

In other words, the partial derivative of the volume with respect to the dihedral angle that corresponds to the edge ij is just half the length of this edge! For a more modern proof see Coxeter [Cox57]. An immediate consequence of this formula is that the volume is an increasing function of the dihedral angles.

The volume function itself can be obtained by (numerically) integrating its partial derivatives. A more direct approach, also involving integration, was obtained by Hsiang [Hsi88]. We used Hsiang's method to compute volumes numerically. Other relevant references are: [BH81], [Cox35], [Hsi92], [Kel91].

C Rounding using a random hyperplane

The angles in which the ratios given in Figure 5 are attained are given in Figure 22. The ratios were obtained using the following technique:

1. The ratio $prob_f/relax_f$ was computed for every feasible point $(\theta_{01}, \theta_{02}, \theta_{03}, \theta_{12}, \theta_{13}, \theta_{23})$ such that θ_{ij} is of the form $\frac{k\pi}{n}$, for $0 \le k \le n$. The volumes that appear in the definition of $prob_f$ were obtained using numerical integration. A typical value of n used here is $n = 20$. Note that as the complexity of this search in $O(n^6)$, it is too time consuming to use much larger values of n. In some cases, the complexity can be reduced to $O(n^5)$ or to $O(n^4)$ by noting that the point in which the minimum of $prob_f/relax_f$ is attained must satisfy certain properties. In such cases larger values of n were used.

	θ_{01}	θ_{02}	θ_{03}	θ_{12}	θ_{13}	θ_{23}	*prob*	*relax*	*ratio*
AND	$\text{acos}(-\frac{1}{3})$	$\text{acos}(-\frac{1}{3})$	$\text{acos}(-\frac{1}{3})$	$\text{acos}(-\frac{1}{3})$	$\text{acos}(-\frac{1}{3})$	$\text{acos}(-\frac{1}{3})$	0	0	0
EQU	–	–	–	0.926800	0.926800	0.926800	0.557485	0.700297	0.796070
AXR	π	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	0	0	0	0.404973
AOR	1.263213	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	1.878380	0.477700	0.651378	0.733368
AOA	1.263213	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	1.263213	0.477700	0.651378	0.733368
TWO	1.741976	1.741976	1.741976	1.399616	1.399616	1.399616	0.304192	0.622244	0.488862
XOR	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$
MAJ	1.537234	1.537234	1.537234	1.604359	1.604359	1.604359	0.516384	0.775168	0.666158
XAD	1.537234	1.537234	1.537234	1.604359	1.604359	1.604359	0.516384	0.775168	0.666158
SEL	0.926800	0.926800	0.926800	0.926800	0	0.926800	0.557485	0.700297	0.796070
OAD	1.355211	1.355211	1.355211	1.786382	1.786382	1.786382	0.738081	0.910439	0.810686
XOA	1.355211	1.355211	1.786382	1.786382	1.355211	1.355211	0.738081	0.910439	0.810686
NTW	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{5}{8}$	1	$\frac{5}{8}$
NAE	–	–	–	2.331122	2.331122	0	0.742019	0.844579	0.878567
OXR	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{3}{4}$	1	$\frac{3}{4}$
OR	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{7}{8}$	1	$\frac{7}{8}$

Figure 22: The worst angles for random hyperplane rounding.

2. If $(\theta^*_{01}, \theta^*_{02}, \theta^*_{03}, \theta^*_{12}, \theta^*_{13}, \theta^*_{23})$ is one of the points in which the lowest ratios above were attained, then a similar exhuastive search was carried out in the vicinity of this point.

3. In addition to the global and local exhuastive searches mentioned above, an attempt was also made to find the minimum of $prob_f/relax_f$ using numerical minimization methods (Matlab was used for this purpose). The starting points used were the worst points found using the exhaustive search as well as many random points. Almost all these searches converged to the points given in Figure 22.

4. After the first three steps above, it was easy to identify the conjectured global minimum point of $prob_f/relax_f$. Though we were usually not able to prove analytically that this point is indeed the global minimum of the function, we verfied, using a Taylor series expansion of the volume function, that the obtained point is at least a local minimum of the function.

5. Finally, an attempt was made to find other local minima of the function $prob_f/relax_f$. To that end, we attempted to find all the critical points of the function. As the partial derivatives of the volume function are elementary functions, the critical points satisfy equations that can be written down explicitly without the use of the volume function. These equations are fairly complicated, however, it is not clear how to solve them exactly. We solved the equations numerically (again using Matlab). Note that this numerical solution does *not* involve numerical integration. A small number of critical points were obtained and it was verified that in all of them the ratio is larger than the ratio in the conjectured global minimum. We believe that we have identified all the critical points of the function

prob$_f$/*relax*$_f$. Proving that will provide, of course, a proof that we obtained the global minimum of the function *prob*$_f$/*relax*$_f$.

We note that the 0.93109 and 0.859 performance ratios of the MAX 2SAT and MAX DICUT algorithms of Feige and Goemans [FG95] were also obtained using numerical methods. No analytical proofs were provided, even though these problems are much simpler than the ones considered here.

We also note that although we have no closed form formula for $\mathrm{Vol}(\lambda_{01},\lambda_{02},\lambda_{12},\lambda_{03},\lambda_{13},\lambda_{23})$, this function is very well behaved. It is continuously differentiable, it is monotone, and its partial derivatives are bounded; they always lie in the interval $[0,\frac{\pi}{2}]$. This makes the possibility that we have incorrectly identified the global minimum of *prob*$_f$/*relax*$_f$ extremely unlikely.

As mentioned in the concluding remarks of the paper, we are currently working on complete analytical proofs for all the claims made in this extended abstract. Some progress in this respect was made in [KZ97].

D A $\frac{3}{2+\sqrt{2}}$-approximation algorithm for MAX 3NAE

Let

$$F(\theta_{12},\theta_{13},\theta_{23}) \;=\; (1-\epsilon)\cdot\frac{\theta_{12}+\theta_{13}+\theta_{23}}{2\pi}+\epsilon\cdot\frac{3}{4}-\gamma\cdot\frac{3-\cos\theta_{12}-\cos\theta_{13}-\cos\theta_{23}}{4}\;,$$

where $\gamma=\frac{3}{2+\sqrt{2}}$ and $\epsilon=1-\frac{3}{4}\cdot\frac{\pi}{\sqrt{2}+1}$. The claim that $\alpha'_{\mathtt{NAE}_3}=\frac{3}{2+\sqrt{2}}$ is equivalent to the claim that $F(\theta_{12},\theta_{13},\theta_{23})\ge 0$, for every $0\le\theta_{12},\theta_{13},\theta_{23}\le\pi$ such that

$$\begin{aligned}\cos\theta_{12}+\cos\theta_{13}+\cos\theta_{23} &\;\ge\; -1\;,\\ \cos\theta_{12}-\cos\theta_{13}-\cos\theta_{23} &\;\ge\; -1\;,\\ -\cos\theta_{12}+\cos\theta_{13}-\cos\theta_{23} &\;\ge\; -1\;,\\ -\cos\theta_{12}-\cos\theta_{13}+\cos\theta_{23} &\;\ge\; -1\;.\end{aligned}$$

The proof is separated into five cases:

case I: None of the constraints is tight.

We look for critical points of the function $F(\theta_{12},\theta_{13},\theta_{23})$. As $\frac{\partial F}{\partial\theta_{12}}=\frac{1-\epsilon}{2\pi}-\frac{\gamma}{4}\sin\theta_{12}$, we get that in any critical point we have $\sin\theta_{12}=\frac{2}{\gamma}\cdot\frac{1-\epsilon}{\pi}=\frac{\sqrt{2}}{2}$ and thus $\theta_{12}=\frac{\pi}{4}$ or $\theta_{12}=\frac{3\pi}{4}$. Similarly, in any critical point θ_{13} and θ_{23} should also be $\frac{\pi}{4}$ or $\frac{3\pi}{4}$. Up to symmetries, there are, therefore, four critical points that have to be considered. They are $\Theta_1=(\frac{\pi}{4},\frac{\pi}{4},\frac{\pi}{4})$, $\Theta_2=(\frac{\pi}{4},\frac{\pi}{4},\frac{3\pi}{4})$, $\Theta_3=(\frac{\pi}{4},\frac{3\pi}{4},\frac{3\pi}{4})$ and $\Theta_4=(\frac{3\pi}{4},\frac{3\pi}{4},\frac{3\pi}{4})$. It is easy to check that $F(\Theta_1)=\frac{3}{8}(6\sqrt{2}-7)+\frac{9\pi}{32}(1-\sqrt{2})\simeq 0.190993$, $F(\Theta_2)=\frac{3}{8}(4\sqrt{2}-5)+\frac{3\pi}{32}(1-\sqrt{2})\simeq 0.124324$ and that $F(\Theta_3)=\frac{3}{8}(2\sqrt{2}-3)+\frac{3\pi}{32}(\sqrt{2}-1)\simeq 0.0576561$. Finally, $F(\Theta_4)=\frac{9\pi}{32}(\sqrt{2}-1)-\frac{3}{8}\simeq -0.00901211$ but, luckily, Θ_4 is not a feasible point as $3\cos\frac{3\pi}{4}=-\frac{3}{2}\sqrt{2}\simeq -2.12132<-1$.

case II: $\theta_{23}=0$.

It is easy to see that $\theta_{23}=0$ together with the four other constraints implies $\theta_{12}=\theta_{13}$. We have therefore, to consider points of the form $(\theta,\theta,0)$. For such points we have

$$F(\theta,\theta,0) \;=\; (1-\epsilon)\cdot\frac{\theta}{\pi}+\epsilon\cdot\frac{3}{4}-\gamma\cdot\frac{1-\cos\theta}{2}\;.$$

By differentiating we get that the two critical points are at $\Theta_5=(\frac{\pi}{4},\frac{\pi}{4},0)$ and $\Theta_6=(\frac{3\pi}{4},\frac{3\pi}{4},0)$. The values of F at these points are $F(\Theta_5)=\frac{3}{8}(4-\pi)(\sqrt{2}-1)\simeq 0.133336$ and $F(\Theta_6)=0$. The point Θ_6 is, as we shall see, the global minimum of the function. We also have to consider the points $\Theta_7=(0,0,0)$

and $\Theta_8 = (\pi, \pi, 0)$. It is easy to check that $F(\Theta_7) = \frac{3}{4} - \frac{9\pi}{16}(\sqrt{2}-1) \simeq 0.0180242$ and that $F(\Theta_8) = \frac{3}{4}(2\sqrt{2}-3) + \frac{3\pi}{16}(\sqrt{2}-1) \simeq 0.115312$. The cases $\theta_{12} = 0$ or $\theta_{13} = 0$ are identical to this case.

case III: $\theta_{23} = \pi$.

In this case the other constraints imply that $\theta_{13} = \pi - \theta_{12}$. We have to consider, therefore, points of the form $(\theta, \pi-\theta, \pi)$. It is not difficult to verify that for every $0 \le \theta \le \pi$ we have $F(\theta, \pi-\theta, \pi) = 1 - \frac{\epsilon}{4} - \gamma = \frac{3}{4}(2\sqrt{2}-3) + \frac{3\pi}{16}(\sqrt{2}-1) \simeq 0.115312$.

case IV: $\cos\theta_{12} + \cos\theta_{13} + \cos\theta_{23} = -1$.

In this case we have $\theta_{23} = \arccos(-1-\cos\theta_{12}-\cos\theta_{13})$ and

$$F'(\theta_{12},\theta_{13}) \;=\; F(\theta_{12},\theta_{13},\theta_{23}) \;=\; (1-\epsilon)\cdot\frac{\theta_{12}+\theta_{13}+\arccos(-1-\cos\theta_{12}-\cos\theta_{13})}{2\pi} + \epsilon\cdot\frac{3}{4} - \gamma\ ,$$

$$\frac{\partial F'}{\partial \theta_{12}} = \frac{1-\epsilon}{2\pi}\cdot\left[1 - \frac{\sin\theta_{12}}{\sqrt{1-(-1-\cos\theta_{12}-\cos\theta_{13})^2}}\right] \;=\; \frac{1-\epsilon}{2\pi}\cdot\left[1-\frac{\sin\theta_{12}}{\sin\theta_{23}}\right]\ .$$

A similar expression is obtained for $\frac{\partial F'}{\partial\theta_{13}}$. At a critical point we therefore have $\sin\theta_{12} = \sin\theta_{13} = \sin\theta_{23}$. Up to symmetries, there are two possibilties, either $\theta_{12} = \theta_{13} = \theta_{23}$, or $\theta_{12} = \theta_{13} = \pi - \theta_{23}$. Recalling that $\cos\theta_{12} + \cos\theta_{13} + \cos\theta_{23} = -1$, we get, in the first case, that $\theta_{12} = \theta_{13} = \theta_{23} = \arccos(-\frac{1}{3})$ and $F(\theta_{12},\theta_{13},\theta_{23}) \simeq 0.029681$. In the second case we get $\theta_{12} = \theta_{13} = \pi$ and $\theta_{23} = 0$. Cases in which $\theta_{23} = 0$ were already covered.

case IV: $-\cos\theta_{12} - \cos\theta_{13} + \cos\theta_{23} = -1$.

In this case we have $\theta_{23} = \arccos(-1+\cos\theta_{12}+\cos\theta_{13})$ and

$$F''(\theta_{12},\theta_{13}) \;=\; (1-\epsilon)\cdot\frac{\theta_{12}+\theta_{13}+\arccos(-1+\cos\theta_{12}+\cos\theta_{13})}{2\pi} + \epsilon\cdot\frac{3}{4} - \gamma\cdot\frac{2-\cos\theta_{12}-\cos\theta_{13}}{2}\ ,$$

$$\frac{\partial F''}{\partial\theta_{12}} \;=\; \frac{1-\epsilon}{2\pi}\cdot\left[1+\frac{\sin\theta_{12}}{\sin\theta_{23}}\right] - \frac{\gamma}{2}\cdot\sin\theta_{12}\ ,$$

$$\frac{\partial F''}{\partial\theta_{13}} \;=\; \frac{1-\epsilon}{2\pi}\cdot\left[1+\frac{\sin\theta_{13}}{\sin\theta_{23}}\right] - \frac{\gamma}{2}\cdot\sin\theta_{13}\ .$$

At a critical point we therefore have $\sin\theta_{12} = \sin\theta_{13}$. Thus either $\theta_{12} = \theta_{13}$, or $\theta_{12} = \pi - \theta_{13}$. If $\theta_{12} = \pi - \theta_{13}$ then $\theta_{23} = \pi$ and this is a case that we have already considered. We may therefore assume that $\theta_{12} = \theta_{13}$. We then have $\theta_{12} = \theta_{13} = \theta$ and $\theta_{23} = \arccos(-1+2\cos\theta)$, where $0 \le \theta \le \frac{\pi}{2}$. It turns out that there are two critical points in this case: a local maximum at $\theta \simeq 0.668963$, with $F \simeq 0.186444$, and a local minimum at $\theta \simeq 1.488969$, with $F \simeq 0.0716386$.

If two of the additional constraints are tight then one of the angles is either 0 or π. Thus, it is not difficult to verify that this covers all the cases that need to be considered. This completes therefore the proof that $\alpha'_{\texttt{NAE}_3} = \frac{3}{2+\sqrt{2}}$.

We note in passing that this analytical proof is already quite lengthy, even though the inequality we set off to prove is quite simple and it involves only three angles. This should give the reader an idea of the difficulty of getting an analytical proof of the more complicated inequalities that we need to prove for the other types of constraints. These inequalities involve six angles and contain, as one of their terms, the volume function of spherical tetrahedra. Nonetheless, we believe that complete analytical proof of all the other inequalities will eventually be obtained.

E Approximating satisfiable instances

The performance ratios obtained using random hyperplane rounding for satisfiable instances are:

$$\beta_{\texttt{TWO}_3} = \frac{6}{\pi}\arccos(-\frac{1}{3}) - 3 \simeq 0.649041\,,$$

$$\beta_{\texttt{MAJ}_3} = \beta_{\texttt{XAD}_3} = \frac{9}{2\pi}\arccos(-\frac{1}{3}) - 2 \simeq 0.736781\,,$$

$$\beta_{\texttt{OAD}_3} = \beta_{\texttt{XOA}_3} = \frac{3}{\pi}\arccos(-\frac{1}{3}) - 1 \simeq 0.824520\,,$$

$$\beta_{\texttt{OR}_2} = \beta_{\texttt{NAE}_3} = \frac{3}{2\pi}\arccos(-\frac{1}{3}) \simeq 0.912260\,.$$

We should mention here that there exist fairly simple polynomial time algorithms that given satisfiable instances of MAX 3SEL and MAX 3MAJ, find assignments that satisfy these instances. This follows from the fact that the two problems can be reduced to 2SAT. A constraint $\texttt{MAJ}(z_1, z_2, z_3)$ can be replaced by the three constraints $z_1 \vee z_2$, $z_1 \vee z_3$ and $z_2 \vee z_3$. A constraint $\texttt{SEL}(z_1, z_2, z_3)$ can be replaced by the two constraints $z_1 \vee z_3$ and $z_2 \vee \bar{z}_3$.

F Hardness results

Let f and g be two Boolean function. Here we assume that f and g depend on at most three variables. A weighted set of constraints $\{w_i \times g(z_{i1}, z_{i2}, z_{i3})\}$, where the w_i's are non-negative weights and the z_{ij}'s are either constants or literals belonging to the variables $x_1, x_2, x_3, y_1, \ldots, y_k$ is said to be an *α-gadget* from f to g iff the following two conditions are satisfied:

$$\begin{array}{lll} (i) & f(x_1,x_2,x_3)=1 \implies & \max_{y_1,\ldots,y_k} \sum w_i\, g(z_{i1},z_{i2},z_{i3}) \;=\; \alpha\,, \\ (ii) & f(x_1,x_2,x_3)=0 \implies & \max_{y_1,\ldots,y_k} \sum w_i\, g(z_{i1},z_{i2},z_{i3}) \;\le\; \alpha - 1\,. \end{array}$$

In other words, if (x_1, x_2, x_3) is a satisfying assignment of f, then the best assignment of Boolean values to the *auxiliary* variables $y_1, \ldots, y_k$ satisfies clauses of total weight exactly α. If, on the other hand, (x_1, x_2, x_3) is not a satisfying assignment of f, then any assignment to $y_1, \ldots, y_k$ satisfies clauses of total weight at most $\alpha - 1$. An α-gadget is said to be *strict* if, in addition to the conditions above, whenever (x_1, x_2, x_3) is not a satisfying assignment of f, there is an assignment for $y_1, \ldots, y_k$ that satisfies clauses of total weight exactly $\alpha - 1$.

Gadgets have been used informally for years. The definition presented above was formalized by Bellare, Goldreich and Sudan [BGS95]. Trevisan *et al.* [TSSW96] show that optimal gadgets can be found by solving, sometimes very large, linear programs.

Håstad [Hås97] showed that for every $\epsilon > 0$, it is NP-hard to distinguish instances of MAX 3XOR in which a fraction of $1 - \epsilon$ of all the clauses can be simultaneously satisfied, from those in which only a fraction of $\frac{1}{2} + \epsilon$ of the clauses can be simultaneously satisfied. Thus the approximation threshold of MAX 3XOR is exactly $\frac{1}{2}$. It is not difficult to see that if there is an α-gadget from $\texttt{XOR}_3$ to f, then $\frac{2\alpha-1}{2\alpha}$ is an inapproximability bound for MAX CSP(f). In other words, if there is a γ-approximation algorithm of MAX CSP(f) with $\gamma > \frac{2\alpha-1}{2\alpha}$ then $P = NP$. (See [TSSW96] for the simple proof of this fact.) All our hardness results are obtained in this way using the gadgets given in Figure 23. The gadgets for $\texttt{EQU}_3$, $\texttt{SEL}_3$ and $\texttt{NAE}_3$ were custom made by Greg Sorkin. All other gadgets were obtained using a simple program that looks for gadgets without auxiliary variables. The gadgets for $\texttt{EQU}_3$, $\texttt{SEL}_3$ and $\texttt{NAE}_3$ are optimal. The other gadgets are optimal if auxiliary variables are not allowed. We intend to check whether they can be improved by alloweing the use of auxiliary variables.

$\texttt{AND}(x_1, x_2, x_3)$, $\texttt{AND}(\bar{x}_1, \bar{x}_2, x_3)$, $\texttt{AND}(\bar{x}_1, x_2, \bar{x}_3)$, $\texttt{AND}(x_1, \bar{x}_2, \bar{x}_3)$

1-gadget

$\frac{3}{2} \times \texttt{EQU}(x_1, y_1, 0)$	$\frac{3}{2} \times \texttt{EQU}(\bar{x}_2, \bar{x}_3, \bar{y}_1)$	$\frac{1}{2} \times \texttt{EQU}(\bar{x}_1, x_2, \bar{y}_2)$	$\frac{1}{2} \times \texttt{EQU}(\bar{x}_3, \bar{y}_2, 0)$
$1 \times \texttt{EQU}(x_2, \bar{x}_3, \bar{y}_2)$	$1 \times \texttt{EQU}(\bar{x}_1, x_3, \bar{y}_3)$	$1 \times \texttt{EQU}(\bar{x}_1, \bar{y}_2, 0)$	$1 \times \texttt{EQU}(\bar{x}_2, \bar{y}_3, 0)$

4-gadget

$\texttt{AXR}(\bar{x}_1, x_2, x_3)$, $\texttt{AXR}(x_1, x_2, \bar{x}_3)$

1-gadget

$\texttt{OXR}(x_3, x_1, x_2)$, $\texttt{OXR}(\bar{x}_3, x_1, \bar{x}_2)$

2-gadget

$\texttt{AOR}(\bar{x}_1, x_2, x_3)$, $\texttt{AOR}(\bar{x}_1, \bar{x}_2, \bar{x}_3)$, $\texttt{AOR}(x_1, x_2, \bar{x}_3)$, $\texttt{AOR}(x_1, \bar{x}_2, x_3)$

2-gadget

$\texttt{AOA}(x_1, \bar{x}_2, \bar{x}_3)$, $\texttt{AOA}(\bar{x}_1, \bar{x}_2, x_3)$, $\texttt{AOA}(x_3, x_1, x_2)$, $\texttt{AOA}(\bar{x}_3, \bar{x}_1, x_2)$

2-gadget

$\texttt{TWO}(\bar{x}_1, x_2, \bar{x}_3)$, $\texttt{TWO}(x_1, \bar{x}_2, x_3)$, $\texttt{TWO}(x_1, x_2, \bar{x}_3)$, $\texttt{TWO}(\bar{x}_1, \bar{x}_2, \bar{x}_3)$

1-gadget

$\texttt{MAJ}(\bar{x}_1, x_2, \bar{x}_3)$, $\texttt{MAJ}(x_1, \bar{x}_2, x_3)$, $\texttt{MAJ}(x_1, x_2, \bar{x}_3)$, $\texttt{MAJ}(\bar{x}_1, \bar{x}_2, \bar{x}_3)$

$\frac{3}{2}$-gadget

$\texttt{XAD}(\bar{x}_1, x_2, \bar{x}_3)$, $\texttt{XAD}(x_1, \bar{x}_2, x_3)$, $\texttt{XAD}(x_1, x_2, \bar{x}_3)$, $\texttt{XAD}(\bar{x}_1, \bar{x}_2, \bar{x}_3)$

$\frac{3}{2}$-gadget

$1 \times \texttt{SEL}(\bar{x}_1, \bar{x}_2, y_3)$	$\frac{1}{2} \times \texttt{SEL}(x_1, x_2, y_2)$	$\frac{1}{2} \times \texttt{SEL}(x_1, x_3, y_2)$	$\frac{1}{2} \times \texttt{SEL}(x_1, \bar{x}_3, \bar{y}_1)$
$\frac{1}{2} \times \texttt{SEL}(x_1, \bar{y}_1, \bar{y}_4)$	$\frac{1}{2} \times \texttt{SEL}(\bar{x}_1, x_2, y_1)$	$\frac{1}{2} \times \texttt{SEL}(\bar{x}_1, \bar{y}_1, y_4)$	$\frac{1}{2} \times \texttt{SEL}(x_2, x_3, \bar{y}_4)$
$\frac{1}{2} \times \texttt{SEL}(x_2, \bar{x}_3, \bar{y}_1)$	$\frac{1}{2} \times \texttt{SEL}(\bar{x}_2, x_3, y_2)$	$\frac{1}{2} \times \texttt{SEL}(\bar{x}_2, \bar{x}_3, y_4)$	$\frac{1}{2} \times \texttt{SEL}(x_3, y_2, \bar{y}_3)$
$\frac{1}{2} \times \texttt{SEL}(\bar{x}_3, y_2, y_3)$	$\frac{1}{2} \times \texttt{SEL}(y_3, y_4, y_5)$	$\frac{1}{2} \times \texttt{SEL}(y_3, y_4, \bar{y}_5)$	

6-gadget

$\texttt{OAD}(x_2, x_1, \bar{x}_3)$, $\texttt{OAD}(\bar{x}_2, x_1, x_3)$, $\texttt{OAD}(x_3, \bar{x}_1, x_2)$, $\texttt{OAD}(\bar{x}_3, \bar{x}_1, \bar{x}_2)$

3-gadget

$\texttt{XOA}(\bar{x}_1, \bar{x}_2, x_3)$, $\texttt{XOA}(x_1, \bar{x}_2, \bar{x}_3)$, $\texttt{XOA}(x_2, x_3, x_1)$, $\texttt{XOA}(x_2, \bar{x}_3, \bar{x}_1)$

3-gadget

$\texttt{NTW}(\bar{x}_1, \bar{x}_2, x_3)$, $\texttt{NTW}(\bar{x}_1, x_2, \bar{x}_3)$, $\texttt{NTW}(x_1, \bar{x}_2, \bar{x}_3)$, $\texttt{NTW}(x_1, x_2, x_3)$

$\frac{4}{3}$-gadget

$\texttt{NAE}(x_1, \bar{y}_1, 0)$	$\texttt{NAE}(x_2, \bar{y}_1, 0)$	$\texttt{NAE}(\bar{x}_2, y_2, 0)$	$\texttt{NAE}(x_3, y_3, 0)$
$\texttt{NAE}(x_1, x_2, \bar{y}_3)$	$\texttt{NAE}(x_1, \bar{y}_1, y_2)$	$\texttt{NAE}(x_2, x_3, \bar{y}_1)$	$\texttt{NAE}(\bar{x}_3, y_1, y_2)$

8-gadget

{ $\texttt{OR}(x_1, x_2, x_3)$, $\texttt{OR}(\bar{x}_1, \bar{x}_2, x_3)$, $\texttt{OR}(\bar{x}_1, x_2, \bar{x}_3)$, $\texttt{OR}(x_1, \bar{x}_2, \bar{x}_3)$

3-gadget

Figure 23: Gadgets from $\texttt{XOR}_3$.