Approximating MIN 2-SAT and MIN 3-SAT

Adi Avidor * *Uri Zwick* *

Abstract

We obtain substantially improved approximation algorithms for the MIN k-SAT problem, for $k = 2, 3$. More specifically, we obtain a 1.1037-approximation algorithm for the MIN 2-SAT problem, improving a previous 1.5-approximation algorithm, and a 1.2136-approximation algorithm for the MIN 3-SAT problem, improving a previous 1.75-approximation algorithm for the problem. These results are obtained by adapting techniques that were previously used to obtain approximation algorithms for the MAX k-SAT problem. We also obtain some hardness of approximation results.

1 Introduction

An instance of MIN SAT in the Boolean variables $x_1, \ldots, x_n$ is composed of a collection of *clauses* $C_1, \ldots, C_m$, and non-negative weights $w_1, \ldots, w_m$ associated with them. Each clause C_i is of the form $z_1 \vee z_2 \vee \ldots \vee z_{k_i}$ $(k_i \geq 1)$ where each z_j is either a variable x_l or its negation $\bar{x}_l$. Each such z_j is called a *literal*. The goal is to assign the variables $x_1, \ldots, x_n$ Boolean values 0 or 1 so that the total weight of the satisfied clauses is *minimized*. A clause is satisfied if at least one of the literals appearing in it is assigned the value 1. The more famous MAX SAT problem is defined similarly, with the goal being to *maximize* the total weight of the satisfied clauses. MIN k-SAT is the version of MIN SAT in which each clause has *at most* k literals. MIN k-SAT(t) is the version of MIN k-SAT where each variable appears, negated or unnegated, at most t times.

MIN SAT was first considered by Kohli *et al.* [KKM94]. They showed that even the unweighted version of MIN 2-SAT, restricted to instances in which all clauses are Horn clauses (i.e., containing at most one negated variable), is NP-hard. They also gave a simple randomized 2-approximation algorithm for the unweighted version of MIN SAT.

Marathe and Ravi [MR96] showed that the MIN SAT problem is in fact equivalent to the well studied MIN VC, the minimum weight vertex cover problem, in the sense that a ρ-approximation algorithm for one of them, for any $\rho \geq 1$, gives immediately a ρ-approximation algorithm for the other. As there are several 2-approximation algorithms for the MIN VC problem, see Bar-Yehuda and Even [BYE81, BYE85], Hochbaum [Hoc82, Hoc83], Monien and Speckenmeyer [MS85], we get immediately a variety of 2-approximation algorithms for the MIN SAT problem. It is conjectured by many that there is no better approximation algorithm for the MIN VC problem, and hence also for the MIN SAT problem, unless P=NP. Dinur and Safra [DS02], improving a result of Håstad [Hås01], showed that there is no $(10\sqrt{5} - 21 - \varepsilon)$-approximation algorithm for the MIN VC problem, for any $\varepsilon > 0$, unless P=NP. (Note that $10\sqrt{5} - 21 \simeq 1.3606$.)

Furthermore, Marathe and Ravi [MR96] showed that the MIN-VC(Δ) problem, i.e., minimum vertex cover problem for graphs with maximum degree Δ, is equivalent to the MIN Δ-SAT(2) problem, the version of the MIN SAT problem in which each clause is of size at most Δ, and each variable occurs at most twice. Approximation algorithm for the MIN VC(Δ) problem were given by Berman and Fujito [BF99] and by Halperin [Hal00]. Hardness results for this problem were obtained by Berman and Karpinski [BK99].

*School of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel. E-mail: {adi, zwick}@tau.ac.il. This research was supported by the ISRAEL SCIENCE FOUNDATION (grant no. 246/01).

	Inapproximability ratio	Approximation ratio
MIN 2-SAT	$\frac{15}{14} - \varepsilon$ [*]	1.1037 [*]
MIN 3-SAT	$\frac{7}{6} - \varepsilon$ [*]	1.2136 [*]
MIN k-SAT	$\frac{7}{6} - \varepsilon$ [*]	$2 - \frac{1}{2^{k-1}}$ [BTV99]
MIN SAT	1.3606 [DS02]	2

Table 1: Approximability and inapproximability results for MIN SAT. (Results obtained here are marked by [*].)

Marathe and Ravi [MR96] also show that the MIN k-SAT(t) problem is at least as easy as the MIN VC($k(t-1)$) problem. In other words, a ρ-approximation algorithm for MIN VC($k(t-1)$), for any $\rho \geq 1$ can be transformed into a ρ-approximation algorithm for MIN k-SAT(t).

Bertsimas *et al.* [BTV99] describe a linear programming based approximation algorithm for the MIN k-SAT problem. Their algorithm achieves a performance ratio of $2 - \frac{1}{2^{k-1}}$. For $k = 2$ they thus obtain a ratio of $\frac{3}{2}$, and for $k = 3$ a ratio of $\frac{7}{4}$. Opposed to previous works, where the variables of the linear programming were rounded *independently*, Bertsimias *et al.* [BTV99] round the variables in a *dependent* manner. Bazgan and Fernandez de la Vega [BdlV99] gave a PTAS for dense instances of MIN 2-SAT. The MIN 2-SAT problem, and other related problems were also considered by Hochbaum [Hoc00] and Hochbaum and Pathria [HP00].

We use semidefinite programming to approximate MIN 2-SAT and MIN 3-SAT. For MIN 2-SAT we improve the approximation ratio from 1.5 to 1.1037. For MIN 3-SAT we improve the approximation ratio from 1.75 to 1.2136. The idea behind our algorithm is as follows. We first assign each Boolean literal x_i a unit vector $\mathbf{v}_i \in \mathbb{R}^{n+1}$. In addition, we introduce the vector $\mathbf{v}_0$, that corresponds to the constant `false`. We then write a semidefinite programming relaxation for the problem. I.e., a minimization program which is linear in the products $\mathbf{v}_i \cdot \mathbf{v}_j$. We design the semidefinite program in a way that any valid assignment to x_i is a feasible point of the program. This program can then be solved, to any desired precision, in polynomial time. We then have to apply a Rounding Procedure (RP) on the vectors $\mathbf{v}_0, \ldots, \mathbf{v}_n$, i.e., we need to "round" the vectors to Boolean values. Rounding procedures are usually randomized algorithms.

Geomans and Williamson [GW95] were the first to suggest the semidefinite programming approach for approximating MAX CUT, MAX 2-SAT, and MIN DICUT. They rounded the solution to the semidefinite program by choosing a random hyperplane passing through the origin. The normal of this hyperplane is an $(n+1)$-dimensional standard normal random variable. Unfortunately, the random hyperplane technique, on its own, cannot be used to obtain approximation algorithms for the MIN k-SAT problem. Feige and Goemans [FG95] extended the random hyperplane technique while studying MAX 2-SAT and MAX DICUT. They *rotated* the vectors towards the vector $\mathbf{v}_0$, and then used random hyperplane rounding. Furthermore, they suggested, but did not explore, another extension to the hyperplane rounding: choose the normal of the hyperplane according to a distribution which is *skewed* towards (or from) the vector $\mathbf{v}_0$. The latter technique was explored for MAX 2-SAT and MAX DICUT first by Matuura and Matsui [MM01a, MM01b], and recently by Lewin *et al.* [LLZ02].

We show here that these techniques can also be applied to the MIN 2-SAT and MIN 3-SAT problems. Table 1 summarizes the approximability and inapproximability results obtained here, and in previous papers, for the MIN k-SAT problem.

2 Preliminaries

We let $x_{n+i} = \bar{x}_i$, for $1 \leq i \leq n$, and we also let $x_0 = 0$. Each clause of a MIN 2-SAT instance is then of the form $x_i \vee x_j$ where $0 \leq i, j \leq 2n$ (if $i = 0$ the clause is of length 1). An instance of MIN 2-SAT can then be

$$\begin{array}{lll}
\text{Min} & \displaystyle\sum_{i,j} w_{ij} \frac{3 - \mathbf{v}_0 \cdot \mathbf{v}_i - \mathbf{v}_0 \cdot \mathbf{v}_j - \mathbf{v}_i \cdot \mathbf{v}_j}{4} & \\
\text{s.t.} & \mathbf{v}_0 \cdot \mathbf{v}_i + \mathbf{v}_0 \cdot \mathbf{v}_j + \mathbf{v}_i \cdot \mathbf{v}_j \geq -1 \quad , & 1 \leq i,j \leq 2n \\
& \mathbf{v}_i \cdot \mathbf{v}_{n+i} = -1 \quad , & 1 \leq i \leq n \\
& \mathbf{v}_i \in \mathbb{R}^{n+1} \ , \ \mathbf{v}_i \cdot \mathbf{v}_i = 1 \quad , & 0 \leq i \leq 2n
\end{array}$$

Figure 1: A semidefinite programming relaxation of MIN 2-SAT

encoded as an array $(w_{ij})_{0 \leq i,j \leq 2n}$, where w_{ij} is interpreted as the weight of the clause $x_i \vee x_j$. In a similar way, an instance of MIN 3-SAT is encoded as an array $(w_{ijk})_{0 \leq i,j,k \leq 2n}$, where w_{ijk} is interpreted as the weight of the clause $x_i \vee x_j \vee x_k$, where $0 \leq i,j,k \leq 2n$. Note that if $i = 0, 0 < j,k \leq 2n$, then the clause is of length two, and if $i = j = 0, 0 < k \leq 2n$, then the clause is of length one. The goal is then to assign the variables $x_1, \ldots, x_{2n}$ Boolean values, such that $x_{n+i} = \bar{x}_i$ and such that the total weight of the satisfied clauses is minimized.

3 Semidefinite Programming Relaxation for MIN 2-SAT

The semidefinite programming relaxation of the MIN 2-SAT problem is given in Figure 1. In this relaxation, a unit vector $\mathbf{v}_i \in \mathbb{R}^{n+1}$ is assigned to each literal x_i, where $0 \leq i \leq 2n$. As $x_0 = 0$, the vector $\mathbf{v}_0$ corresponds to the constant 0 (`false`). The vectors $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{2n}$ are unit vectors on the unit sphere in $\mathbb{R}^{n+1}$. To ensure $x_{n+i} = \bar{x}_i$, we require $\mathbf{v}_i \cdot \mathbf{v}_{n+i} = -1$, for $1 \leq i \leq n$. This is equivalent to $\mathbf{v}_{n+i} = -\mathbf{v}_i$, since $\mathbf{v}_i$ are unit vectors ($1 \leq i \leq n$). As the value of a solution of the semidefinite program depends only on the inner products between the vectors, we may assume, with out loss of generality, that $\mathbf{v}_0 = (1, 0, \ldots, 0) \in \mathbb{R}^{n+1}$. Note that the semidefinite program in Figure 1 is a relaxation of MIN 2-SAT: Let $x \in \{0,1\}^{2n+1}$ be a valid assignment (i.e., $x_{n+i} = \bar{x}_i$). Let $\mathbf{v}_i = \mathbf{v}_0$ if $x_i = 0$, and $\mathbf{v}_i = -\mathbf{v}_0$ if $x_i = 1$, where $0 \leq i \leq 2n$. Then, it is easy to see that for $0 \leq i,j \leq 2n$ we have

$$\frac{3 - \mathbf{v}_0 \cdot \mathbf{v}_i - \mathbf{v}_0 \cdot \mathbf{v}_j - \mathbf{v}_i \cdot \mathbf{v}_j}{4} = x_i \vee x_j$$

In addition, the constraints $\mathbf{v}_0 \cdot \mathbf{v}_i + \mathbf{v}_0 \cdot \mathbf{v}_j + \mathbf{v}_i \cdot \mathbf{v}_j \geq -1$, called the 'triangle constraints', hold as $\mathbf{v}_i$ and $\mathbf{v}_j$ are assigned the values $\pm\mathbf{v}_0$, for $0 \leq i,j \leq 2n$.

4 Semidefinite Programming Relaxation for MIN 3-SAT

The semidefinite programming relaxation of the MIN 3-SAT problem is given in Figure 2. In a similar way to MIN 2-SAT, a unit vector $\mathbf{v}_i \in \mathbb{R}^{n+1}$ is assigned to each literal x_i, where $0 \leq i \leq 2n$. In addition, a scalar z_{ijk} is assigned to each clause, where $0 \leq i,j,k \leq 2n$. Note that the semidefinite program in Figure 2 is a relaxation of MIN 3-SAT: Let $x \in \{0,1\}^{2n+1}$ be a valid assignment. Let $\mathbf{v}_i = \mathbf{v}_0$ if $x_i = 0$, and $\mathbf{v}_i = -\mathbf{v}_0$ if $x_i = 1$, where $0 \leq i \leq 2n$. Then, it is easy to see that for $0 \leq i,j,k \leq 2n$ we have

$$\max \left\{ \begin{array}{cc} \frac{3 - \mathbf{v}_0 \cdot \mathbf{v}_i - \mathbf{v}_0 \cdot \mathbf{v}_j - \mathbf{v}_i \cdot \mathbf{v}_j}{4}, & \frac{3 - \mathbf{v}_0 \cdot \mathbf{v}_i - \mathbf{v}_0 \cdot \mathbf{v}_k - \mathbf{v}_i \cdot \mathbf{v}_k}{4} \\ \frac{3 - \mathbf{v}_0 \cdot \mathbf{v}_j - \mathbf{v}_0 \cdot \mathbf{v}_k - \mathbf{v}_j \cdot \mathbf{v}_k}{4}, & 0 \end{array} \right\} = x_i \vee x_j \vee x_k \ .$$

Again, the 'triangle constraints' hold.

5 Rounding Procedures

We now describe the rounding procedures we use in this paper, namely *RHP*, *ROT$_f$* and *THRESH$_g$*. For a survey of rounding procedures see Lewin *et al.* [LLZ02]. The basic rounding technique, introduced by Goemans and

$$\begin{array}{llcl}
\text{Min} & \displaystyle\sum_{i,j,k} w_{ijk} z_{ijk} & & \\
\text{s.t.} & z_{ijk} \geq \dfrac{3 - \mathbf{v}_0 \cdot \mathbf{v}_i - \mathbf{v}_0 \cdot \mathbf{v}_j - \mathbf{v}_i \cdot \mathbf{v}_j}{4} & , & 0 \leq i,j,k \leq 2n \\
 & z_{ijk} \geq \dfrac{3 - \mathbf{v}_0 \cdot \mathbf{v}_i - \mathbf{v}_0 \cdot \mathbf{v}_k - \mathbf{v}_i \cdot \mathbf{v}_k}{4} & , & 0 \leq i,j,k \leq 2n \\
 & z_{ijk} \geq \dfrac{3 - \mathbf{v}_0 \cdot \mathbf{v}_j - \mathbf{v}_0 \cdot \mathbf{v}_k - \mathbf{v}_j \cdot \mathbf{v}_k}{4} & , & 0 \leq i,j,k \leq 2n \\
 & z_{ijk} \geq 0 & , & 0 \leq i,j,k \leq 2n \\
 & \mathbf{v}_0 \cdot \mathbf{v}_i + \mathbf{v}_0 \cdot \mathbf{v}_j + \mathbf{v}_i \cdot \mathbf{v}_j \geq -1 & , & 1 \leq i,j \leq 2n \\
 & \mathbf{v}_i \cdot \mathbf{v}_{n+i} = -1 & , & 1 \leq i \leq n \\
 & \mathbf{v}_i \in \mathbb{R}^{n+1} \ , \ \mathbf{v}_i \cdot \mathbf{v}_i = 1 & , & 0 \leq i \leq 2n
\end{array}$$

Figure 2: A semidefinite programming relaxation of MIN 3-SAT

Williamson [GW95], is the *random hyperplane rounding* (*RHP*) technique. In this rounding procedure, the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ are rounded using a random hyperplane:

RHP: Let $\mathbf{r} = (r_0, r_1, \ldots, r_n)$ be a standard $(n+1)$-dimensional normal random variable. For $1 \leq i \leq n$, let $x_i \leftarrow 0$ if $\mathbf{v}_0$ and $\mathbf{v}_i$ lie on the same side of the hyperplane with the normal vector $\mathbf{r}$, and let $x_i \leftarrow 1$ otherwise.

Since we require $\mathbf{v}_{n+i} = -\mathbf{v}_i$ for $1 \leq i \leq n$, *RHP* preserves consistency, in the sense that x_i and $x_{n+i} = \bar{x}_i$ are always assigned opposite values. As mentioned before, *RHP* was generalized by Feige and Goemans [FG95] to the *ROT$_f$* rounding procedure, parameterized by a *rotation* function f, as defined below:

Definition 5.1 *A continuous function $f : [0, \pi] \to [0, \pi]$ is called a* rotation *function if $f(\pi - \theta) = \pi - f(\theta)$ for all $\theta \in [0, \pi]$.*

ROT$_f$: Let $\theta_i = \arccos(\mathbf{v}_0 \cdot \mathbf{v}_i)$ be the angle between $\mathbf{v}_0$ and $\mathbf{v}_i$. Rotate $\mathbf{v}_i$ into a vector $\mathbf{v}'_i \in Span(\mathbf{v}_0, \mathbf{v}_i)$ that forms an angle of $\theta'_i = f(\theta_i)$ with $\mathbf{v}_0$. Round the vectors $\mathbf{v}_0, \mathbf{v}'_1, \ldots, \mathbf{v}'_n$ using the *RHP* rounding procedure.

Again, since f is a rotation function consistency is preserved. For convenience we let $\mathbf{v}'_0 = \mathbf{v}_0$.

In addition to the rotation technique, Feige and Goemans [FG95] suggested taking the random variable r_0 from an *arbitrary* distribution, rather than the standard normal distribution. In other words, choose the normal of the random hyperplane with a distribution which is *skewed* towards (or away) the vector $\mathbf{v}_0$. The *THRESH$_g$* rounding procedure is a special case of using *rotation* combined with a *skewed* hyperplane distribution. In *THRESH$_g$*, the random variable r_0 is assigned a *constant* value. We adopt a different but equivalent definition of *THRESH$_g$* proposed by [LLZ02]. To avoid confusion, we note that *THRESH$_g$* is a member of the family $\mathcal{THRESH}^-$ of [LLZ02]. The *THRESH$_g$* rounding procedure, is parameterized by an odd function $g : [-1, 1] \to \mathbb{R}$, that is, a function satisfying $g(-x) = -g(x)$, as follows

THRESH$_g$: Let $\mathbf{r} = (0, r_1, \ldots, r_n)$, where $r_1, r_2, \ldots, r_n$ are independent standard normal variables. For $1 \leq i \leq n$, let $x_i \leftarrow 1$ if $\mathbf{v}_i \cdot \mathbf{r} \leq g(\mathbf{v}_0 \cdot \mathbf{v}_i)$ and $x_i \leftarrow 0$ otherwise.

Again, the 'oddness' of the function g is used in order to preserve consistency. Lewin *et al.* [LLZ02] showed that if a rounding procedure is characterized by a rotation function f and if the random variable r_0 is assigned the constant value a, then the rounding procedure is equivalent to *THRESH$_g$* with $g(x) = -a\,(\cot f(\arccos x))\,\sqrt{1 - x^2}$.

Though is seems that THRESH_g is a generalization of ROT_f, the families $\mathcal{ROT}$ and $\mathcal{THRESH}$ are disjoint. This claim can be derived using observations made in [LLZ02].

6 Analysis

In this section we describe the computation of the performance ratio obtained using a given rounding procedure RP. This computation is later used in Section 7 to compute the ratios of our approximation algorithms. In addition, we explain here how to show that a given rounding procedure is an essentially optimal rounding procedures from a given family of rounding procedures.

Let $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_k \in \mathbb{R}^{n+1}$ be vectors corresponding to some clause $x_1 \vee x_2 \vee \ldots \vee x_k$ of MIN k-SAT $(k = 2, 3)$. Denote by $prob_{RP}(\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_k)$ the probability that the clause $x_1 \vee x_2 \vee \ldots \vee x_k$ is satisfied when the vectors $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_k$ are rounded using the rounding procedure RP. Denote by $value(\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_k)$ the contribution of the clause to the value of the semidefinite program. For MIN 2-SAT we have

$$value(\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2) \;=\; \frac{(3 - \mathbf{v}_0 \cdot \mathbf{v}_i - \mathbf{v}_0 \cdot \mathbf{v}_j - \mathbf{v}_i \cdot \mathbf{v}_j)}{4} \,,$$

and for MIN 3-SAT we have

$$value(\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3) \;=\; \max \left\{ \begin{array}{cc} \frac{3 - \mathbf{v}_0 \cdot \mathbf{v}_i - \mathbf{v}_0 \cdot \mathbf{v}_j - \mathbf{v}_i \cdot \mathbf{v}_j}{4}, & \frac{3 - \mathbf{v}_0 \cdot \mathbf{v}_i - \mathbf{v}_0 \cdot \mathbf{v}_k - \mathbf{v}_i \cdot \mathbf{v}_k}{4} \\ \frac{3 - \mathbf{v}_0 \cdot \mathbf{v}_j - \mathbf{v}_0 \cdot \mathbf{v}_k - \mathbf{v}_j \cdot \mathbf{v}_k}{4}, & 0 \end{array} \right\} .$$

Denote

$$ratio_{RP}(\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_k) \;=\; \frac{prob_{RP}(\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_k)}{value(\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_k)} \,.$$

We denote by $\alpha_k(RP)$ the performance ratio of the approximation algorithm that uses the rounding procedure RP. We denote by $\Omega_{n+1}^{(k)}$ the vector $(k+1)$-tuples $(\mathbf{v}_0, \ldots, \mathbf{v}_k)$ that satisfy all the constraints of MIN k-SAT semidefinite programming relaxation and have a strictly positive value. For MIN 2-SAT and MIN 3-SAT we can write:

$$\Omega_{n+1}^{(k)} = \left\{ (\mathbf{v}_0, \ldots, \mathbf{v}_k) \;\middle|\; \begin{array}{rrl} \mathbf{v}_i \in \mathbb{R}^{n+1}, \quad \mathbf{v}_i \cdot \mathbf{v}_i = & 1, & 0 \le i \le k \\ \mathbf{v}_0 \cdot \mathbf{v}_i + \mathbf{v}_0 \cdot \mathbf{v}_j + \mathbf{v}_i \cdot \mathbf{v}_j \ge & -1, & 1 \le i, j \le k \\ -\mathbf{v}_0 \cdot \mathbf{v}_i - \mathbf{v}_0 \cdot \mathbf{v}_j + \mathbf{v}_i \cdot \mathbf{v}_j \ge & -1, & 1 \le i, j \le k \\ -\mathbf{v}_0 \cdot \mathbf{v}_i + \mathbf{v}_0 \cdot \mathbf{v}_j - \mathbf{v}_i \cdot \mathbf{v}_j \ge & -1, & 1 \le i, j \le k \\ \mathbf{v}_0 \cdot \mathbf{v}_i - \mathbf{v}_0 \cdot \mathbf{v}_j - \mathbf{v}_i \cdot \mathbf{v}_j \ge & -1, & 1 \le i, j \le k \\ value(\mathbf{v}_0, \ldots, \mathbf{v}_k) > & 0 & \end{array} \right\} .$$

Let $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_n$ be an optimal solution of the semidefinite program corresponding to a MIN k-SAT instance $(w_{i_1 \ldots i_k})$. As the semidefinite program is a relaxation of MIN k-SAT, the expression $\sum_{i_1, \ldots, i_k} w_{i_1 \ldots i_k} \cdot value(\mathbf{v}_0, \mathbf{v}_{i_1}, \ldots, \mathbf{v}_{i_k})$ is a lower bound on the value of the optimum solution. We assume that it is bigger than 0. (The case in which it is 0 is easily dealt with.) The expected value of the assignment produced using rounding procedure RP is:

$$\sum_{i_1, \ldots, i_k} w_{i_1 \ldots i_k}\, prob_{RP}(\mathbf{v}_0, \mathbf{v}_{i_1}, \ldots, \mathbf{v}_{i_k}) \,.$$

Therefore, we get:

$$\begin{aligned}\alpha_k(RP) &\le \sup_{(w_{i_1,\ldots,i_k})} \frac{\sum_{i_1,\ldots,i_k} w_{i_1\ldots i_k}\, prob_{RP}(\mathbf{v}_0,\mathbf{v}_{i_1},\ldots,\mathbf{v}_{i_k})}{\sum_{i_1,\ldots,i_k} w_{i_1\ldots i_k}\, value(\mathbf{v}_0,\mathbf{v}_{i_1},\ldots,\mathbf{v}_{i_k})} \\ &\le \sup_{(\mathbf{u}_0,\mathbf{u}_1,\ldots,\mathbf{u}_k)\in\Omega^{(k)}_{n+1}} \frac{prob_{RP}(\mathbf{u}_0,\mathbf{u}_1,\ldots,\mathbf{u}_k)}{value(\mathbf{u}_0,\mathbf{u}_1,\ldots,\mathbf{u}_k)} \\ &= \sup_{(\mathbf{u}_0,\mathbf{u}_1,\ldots,\mathbf{u}_k)\in\Omega^{(k)}_{n+1}} ratio_{RP}(\mathbf{u}_0,\mathbf{u}_1,\ldots,\mathbf{u}_k)\ .\end{aligned}$$

The computation of $prob_{RP}(\mathbf{v}_0,\mathbf{v}_1,\ldots,\mathbf{v}_k)$ for the rounding procedures ROT_f and $THRESH_g$ is described in subsections 6.1 and 6.2 . For each of our rounding procedures, the probability $prob_{RP}(\mathbf{v}_0,\mathbf{v}_1,\ldots,\mathbf{v}_k)$ depends only on the inner products $\mathbf{v}_i\cdot\mathbf{v}_j$, for $0\le i,j\le k$. As we are only interested in the inner products of the vectors $\mathbf{v}_0,\mathbf{v}_1,\ldots,\mathbf{v}_k$, we may assume $\mathbf{v}_0,\mathbf{v}_1,\ldots,\mathbf{v}_k$ lie in $\mathbb{R}^{k+1}$. In other words, we may apply our maximization in $\Omega^{(k)}_{k+1}$, rather than $\Omega^{(k)}_{n+1}$, and get

$$\alpha_k(RP)\ \le\ \beta_k(RP)\ \stackrel{def}{=}\ \sup_{(\mathbf{u}_0,\mathbf{u}_1,\ldots,\mathbf{u}_k)\in\Omega^{(k)}_{k+1}} ratio_{RP}(\mathbf{u}_0,\mathbf{u}_1,\ldots,\mathbf{u}_k)\ .$$

For a family $\mathcal{F}$ of rounding procedures (e.g., the family $\bigcup_f ROT_f$), $\beta_k(RP)$ (where $RP\in\mathcal{F}$) can be bounded from below as follows. Let

$$\{(\mathbf{v}_0^{(1)},\mathbf{v}_1^{(1)},\ldots,\mathbf{v}_k^{(2)}),(\mathbf{v}_0^{(2)},\mathbf{v}_1^{(2)},\ldots,\mathbf{v}_k^{(2)}),\ldots,(\mathbf{v}_0^{(r)},\mathbf{v}_1^{(r)},\ldots,\mathbf{v}_k^{(r)})\}\subseteq\Omega^{(k)}_{k+1}$$

be a finite set of r *configurations*. Then, for every $RP\in\mathcal{F}$ we have

$$\beta_k(RP)\ \ge\ \max_{j=1}^{r}\ \inf_{R\in\mathcal{F}}\ ratio_R(\mathbf{v}_0^{(j)},\mathbf{v}_1^{(j)},\ldots,\mathbf{v}_k^{(j)})\ .$$

We use this simple bounding technique to show that our rounding procedures are essentially the optimal rounding procedures from the $\mathcal{ROT}$ and $\mathcal{THRESH}$ families.

6.1 Computing probabilities for ROT_f

Let $\mathbf{v}'_0,\mathbf{v}'_1,\ldots,\mathbf{v}'_k$ be the vectors obtained by rotating the vectors $\mathbf{v}_0,\mathbf{v}_1,\ldots,\mathbf{v}_k$ using a rotation function f. Then, $prob_{ROT_f}(\mathbf{v}_0,\mathbf{v}_1,\ldots,\mathbf{v}_k) = prob_{RHP}(\mathbf{v}'_0,\mathbf{v}'_1,\ldots,\mathbf{v}'_k)$. Denote the angle between the vectors $\mathbf{v}_i$ and $\mathbf{v}_j$ by $\theta_{ij} = \arccos(\mathbf{v}_i\cdot\mathbf{v}_j)$, and the angle between the vectors $\mathbf{v}'_i$, and $\mathbf{v}'_j$ by θ'_{ij}. Then, $\theta'_{0i} = f(\theta_{0i})$, for $1\le i\le k$. The angle θ'_{ij}, for $1\le i,j\le k$, can be calculated using the cosine rule for spherical triangles as ([FG95], [Ber87]):

$$\cos(\theta'_{ij}) = \cos(\theta'_{0i})\cos(\theta'_{0j}) + \frac{cos(\theta_{ij}) - \cos(\theta_{0i})\cos(\theta_{0j})}{\sin(\theta_{0i})\sin(\theta_{0j})}\sin(\theta'_{0i})\sin(\theta'_{0j})\ .$$

For clauses of size two, we denote by $prob(\mathbf{v}'_i/\mathbf{v}'_j)$ the probability that the random hyperplane separates $\mathbf{v}'_i$ and $\mathbf{v}'_j$. Then $prob(\mathbf{v}'_i/\mathbf{v}'_j) = \theta'_{ij}/\pi$, as $prob_{RHP}(\mathbf{v}'_0,\mathbf{v}'_1,\mathbf{v}'_2) = \frac{1}{2}(prob(\mathbf{v}'_0/\mathbf{v}'_1) + prob(\mathbf{v}'_0/\mathbf{v}'_2) + prob(\mathbf{v}'_1/\mathbf{v}'_2))$ we get

$$prob_{RHP}(\mathbf{v}'_0,\mathbf{v}'_1,\mathbf{v}'_2) = \frac{\theta'_{01}+\theta'_{02}+\theta'_{12}}{2\pi}\ .$$

For clauses of size three, the calculation of the probability is harder. Karloff and Zwick [KZ97] showed that

$$prob_{RHP}(\mathbf{v}'_0,\mathbf{v}'_1,\mathbf{v}'_2,\mathbf{v}'_3) = 1 - \frac{Vol(\lambda_{01},\lambda_{02},\lambda_{12},\lambda_{03},\lambda_{13},\lambda_{23})}{\pi^2}\ ,$$

where

$$(\lambda_{01},\lambda_{02},\lambda_{12},\lambda_{03},\lambda_{13},\lambda_{23}) = (\pi,\pi,\pi,\pi,\pi,\pi) - (\theta'_{23},\theta'_{13},\theta'_{03},\theta'_{12},\theta'_{02},\theta'_{01})\ ,$$

and $Vol(\lambda_{01}, \lambda_{02}, \lambda_{12}, \lambda_{03}, \lambda_{13}, \lambda_{23})$ is the volume of a spherical tetrahedron in which the angles between the faces are $\lambda_{01}, \lambda_{02}, \lambda_{12}, \lambda_{03}, \lambda_{13}, \lambda_{23}$. These angles are called the dihedral angles of the spherical tetrahedron. No closed-form formula for the volume is known. We compute the volume by using Hsiang's method of infinitesimal symmetrization [Hsi88], which uses one dimensional numerical integration.

Note that for both types of clauses the probability depends on the inner products $\mathbf{v}_i \cdot \mathbf{v}_j$ only (for $0 \le i, j \le k$), as the angles θ'_{ij} are a function of these inner products.

6.2 Computing probabilities for *THRESH$_g$*

We denote by $\phi_\ell(\mathbf{x})$ the probability density function of an ℓ-dimensional standard normal random variable, that is, $\phi_\ell(\mathbf{x}) = (2\pi)^{-\ell/2} \exp(-||\mathbf{x}||^2/2)$, where $||\mathbf{x}||^2 = \sum_{i=1}^{\ell} x_i^2$. Define the set $F(\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_k) \subseteq \mathbb{R}^{n+1}$ as

$$F(\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_k) = \{ \ \mathbf{r} = (0, r_1, \ldots, r_n) \ | \ \mathbf{v}_i \cdot \mathbf{r} \ge g(\mathbf{v}_i \cdot \mathbf{v}_0), \quad 1 \le i \le k \} \ .$$

$F(\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_k)$ is the set of all values for the random variable $\mathbf{r}$, for which the clause $x_1 \vee x_2 \vee \ldots \vee x_k$ is *not* satisfied when using *THRESH$_g$* . Hence,

$$1 - prob_{THRESH_g}(\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_k) = \int\limits_{r \in F(\mathbf{v}_0, \ldots, \mathbf{v}_k)} \phi_{n+1}(\mathbf{r})\, d\mathbf{r} \ .$$

This $(n+1)$-dimensional integral may be reduced to a d-dimensional integral, where $d+1$ is the dimension of the subspace spanned by $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_k$. Assume, without loss of generality, that $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_d$ form a basis of this subspace. Let $\{\mathbf{u}_{d+1}, \ldots, \mathbf{u}_n\}$ be an orthonormal basis to the space perpendicular to $Span(\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_d)$. Denote by U the matrix whose rows are the vectors $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_d, \mathbf{u}_{d+1}, \ldots, \mathbf{u}_n$, and by V the matrix whose rows are the vectors $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_d$. Apply the substitution $\mathbf{s} = U\mathbf{r}$ to the integral. Then, $\mathbf{r} = U^{-1}\mathbf{s}$. As the first coordinate of $\mathbf{r}$ is 0, so is the first coordinate of $\mathbf{s}$. Denote $\mathbf{s} = (s_0, \ldots, s_n)^T$, $\mathbf{s}' = (s_0, \ldots, s_d)^T = V\mathbf{r}$, and $\mathbf{s}'' = (s_{d+1}, \ldots, s_n)^T$. Then,

$$UU^T = \begin{pmatrix} VV^T & 0 \\ 0 & I_{n-d} \end{pmatrix} \ .$$

Hence, the Jacobian of the transformation is

$$\det U^{-1} = \sqrt{(\det UU^T)^{-1}} = \sqrt{(\det VV^T)^{-1}} \ ,$$

and the integrand becomes

$$\begin{aligned} \phi_{n+1}(\mathbf{r}) &= \phi_{n+1}(U^{-1}\mathbf{s}) \\ &= (2\pi)^{-(n+1)/2} \, e^{-\mathbf{s}^T (UU^T)^{-1} \mathbf{s}/2} \\ &= (2\pi)^{-(d+1)/2} \, e^{-\mathbf{s}'^T (VV^T)^{-1} \mathbf{s}'/2} \, \phi_{n-d}(\mathbf{s}'') \ . \end{aligned}$$

The first d conditions $\mathbf{v}_i \cdot \mathbf{r} \ge g(\mathbf{v}_i \cdot \mathbf{v}_0)$, become $s_i \ge g(\mathbf{v}_i \cdot \mathbf{v}_0)$, for $1 \le i \le d$. The other $k-d$ conditions $\mathbf{v}_i \cdot \mathbf{r} \ge g(\mathbf{v}_i \cdot \mathbf{v}_0)$ can be written as $\mathbf{v}_i^T V^T (VV^T)^{-1} s'^T \ge g(\mathbf{v}_i^T \mathbf{v}_0)$, for $d+1 \le i \le k$. Denote by $F'(\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_k)$ the set of all vectors $\mathbf{s}' \in \mathbb{R}^{d+1}$ that satisfy the above conditions together with the condition $s_0 = 0$. Thus, $1 - prob_{THRESH_g}(\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_k)$ is equal to

$$\int\limits_{\substack{s' \in F'(\mathbf{v}_0, \ldots, \mathbf{v}_k) \\ s'' \in \mathbb{R}^{n-d}}} (2\pi)^{-(d+1)/2} \, e^{-\mathbf{s}'^T (VV^T)^{-1} \mathbf{s}'/2} \ \phi_{n-d}(\mathbf{s}'') \, \frac{d\mathbf{s}' d\mathbf{s}''}{\sqrt{\det(VV^T)}}$$

$$= (2\pi)^{-(d+1)/2} \left(\det(VV^T)\right)^{-1/2} \int\limits_{F'(\mathbf{v}_0, \ldots, \mathbf{v}_k)} e^{-\mathbf{s}'^T (VV^T)^{-1} \mathbf{s}'/2} d\mathbf{s}' \ .$$

The last equality follows from the fact that $\int_{\mathbf{s}''\in\mathbb{R}^{n-d}} \phi_{n-d}(\mathbf{s}'')d\mathbf{s}'' = 1$. As $s_0 = 0$, the last d-dimensional integral is in fact a $(d-1)$-dimensional integral. Note that the integral depends only on the inner products $\mathbf{v}_i \cdot \mathbf{v}_j$, for $0 \le i, j \le k$.

In our implementation for MIN 2-SAT we use one dimensional numerical integration, as we use the function $\Phi(x) = \int_{-\infty}^{x} \phi_1(t)dt$ in the integrand.

7 Algorithms

In this section we describe our algorithms for MIN 2-SAT and MIN 3-SAT. For both problems we give algorithms, which use a *ROT*$_f$ rounding procedure. For MIN 2-SAT we give a better algorithm which uses a *THRESH*$_g$ rounding procedure. The performance ratios of our algorithms are calculated by bounding $\beta_k(RP)$ from above. We also obtain lower bounds for $\beta_k(RP)$ among rounding procedures from the $\mathcal{ROT}$ family, and the $\mathcal{THRESH}$ family. These lower bounds show that our choices of rotation functions are almost optimal. The calculation of $\beta_k(RP)$ is a global optimization task. So is the lower bound of $\beta_k(RP)$ for a given family of rounding procedures. These bounds were calculated by using the optimization toolbox of `Matlab`. We could, in principle, obtain computer-assisted proofs of these bounds, e.g., using $\mathcal{R}$eal$\mathcal{S}$earch [Zwi02], which is a generic software for proving inequalities. However, this would require a huge amount of work.

For both the $\mathcal{ROT}$ and the $\mathcal{THRESH}$ families, we need to conduct a search for a good rotation function. We briefly describe below how this search is conducted. As $\beta_k(RP)$ depends only on the inner product of the vectors, we identify a configuration by the angles between its vectors. In our search, we assume that the rotation function is piecewise linear. We represent our rotation function by a set of p points $\{(\theta_1, f(\theta_1)), \ldots, (\theta_p, f(\theta_p))\}$. It turns out that $\beta_k(RP)$ can be easily bounded from below using a *small* set of configurations, typically three to four configurations. These configurations usually fulfill equality in one (of more) of the triangle inequalities. Therefore, we first bound $\beta_k(RP)$ for all rotation functions by finding these "bad" configurations. In this process we also obtain the optimal rotation values for these angles. These angles, and their rotation values are a good starting point for the search of a good piecewise linear rotation function. At each stage of the search, we obtain a set of "bad" configurations for the current rotation function (typically hundreds of configurations). We then optimize the break points $\{(\theta_1, f(\theta_1)), \ldots, (\theta_p, f(\theta_p))\}$ to perform better on the current "bad" configurations. Next we optimize the "bad" configurations to a new set of configurations that perform worse on the new optimized rotation function. We repeat this process until we find a rotation function which is close enough to our lower bound. The results of our searches are given in Subsections 7.1, 7.2 and 7.3.

7.1 An Approximation Algorithm for MIN 2-SAT using *ROT*$_f$

The algorithm first solves the semidefinite programming relaxation of Figure 1. It then rounds the solution using the rounding procedure *ROT*$_{f_{2R}}$, which rotates the vectors using the piecewise linear rotation function f_{2R} in Table 2(a). The algorithm achieves a performance ratio of at most 1.115682, and the rotation function f_{2R} is nearly optimal in the $\mathcal{ROT}$ family as

$$1.115674 \;<\; \inf_f \beta_2(ROT_f) \;<\; \beta_2(ROT_{f_{2R}}) < 1.115682 \,.$$

The lower bound may of 1.115674 be obtained in the following way. Let

$$\theta_1 = 1.2897 \;,\; \theta_2 = 1.3688 \;,\; \theta_3 = 1.3889 \;,\; \theta_4 = \pi/2 \;,$$

and denote a configuration Θ_{ij} by $\Theta_{ij} = (\theta_i, \pi - \theta_j, \arccos(1 - \cos(\theta_i) + \cos(\pi - \theta_j)))$. Then optimize the values of any rotation function on the angles $\theta_1, \theta_2, \theta_3, \theta_4$ to minimize the maximum ratio of the four configurations $\{\Theta_{14}, \Theta_{22}, \Theta_{23}, \Theta_{13}\}$. Note that for each configuration Θ_{ij}, one of the triangle constraints is tight.

θ	$f_{2R}(\theta)$
0.000000	0.000000
0.632970	0.000000
1.247303	1.022423
1.336455	1.177710
1.483008	1.429392
1.570796	1.570796

(a)

θ	$f_{2T}(\theta)$
0.000000	0.000000
0.275281	0.554272
0.929855	0.979610
1.295874	1.319045
1.441122	1.455178
1.537966	1.541859
1.570796	1.570796

(b)

Table 2: Rotation functions for the MIN 2-SAT approximation algorithms. (a) The piecewise linear rotation function f_{2R} used for the 1.115682-approximation algorithm. (b) The piecewise linear rotation function f_{2T} used for the 1.103681-approximation algorithm.

7.2 An Approximation Algorithm for MIN 2-SAT using $\textit{THRESH}_g$

The algorithm first solves the semidefinite programming relaxation of Figure 1. It then rounds the solution using the rounding procedure $\textit{THRESH}_{g_{2T}}$, with $g_{2T}(x) = -a\,(\cot f_{2T}(\arccos x))\sqrt{1-x^2}$, where $a = 1.5$ and f_{2T} is the piecewise linear function given in Table 2(b). The algorithm achieves a performance ratio of at most 1.103681. The rounding procedure $\textit{THRESH}_{g_{2T}}$ is an essentially optimal rounding procedure from the family $\mathcal{THRESH}$, as

$$1.103639 \;<\; \inf_g \beta_2(\textit{THRESH}_g) \;<\; \beta_2(\textit{THRESH}_{g_{2T}}) \;<\; 1.103681 \;.$$

To obtain the lower bound of 1.103639 we found the values of the function g at the cosines of the four angles

$$\theta_1 = 1.3483 \;,\; \theta_2 = 1.3893 \;,\; \theta_3 = 1.5061 \;,\; \theta_4 = \pi/2$$

that minimize the maximum ratio of $\{\Theta_{14}, \Theta_{23}, \Theta_{31}, \Theta_{32}\}$, where the configuration Θ_{ij} is the same as that defined in Section 7.1.

7.3 An Approximation Algorithm for MIN 3-SAT using $\textit{ROT}_f$

The algorithm first solves the semidefinite programming relaxation of Figure 2. It then rounds the solution using the rounding procedure $\textit{ROT}_{f_{3R}}$, which rotates the vectors using the piecewise linear rotation function f_{3R} given in Table 3. The algorithm achieves a performance ratio of at most 1.21358, and the rotation function f_{3R} is nearly optimal in the $\mathcal{ROT}$ family as

$$1.21145 \;<\; \inf_f \beta_3(\textit{ROT}_f) \;<\; \beta_3(\textit{ROT}_{f_{3R}}) \;<\; 1.21358 \;.$$

We obtain the lower bound using the three angles

$$\theta_1 = 1.2965 \;,\; \theta_2 = 1.3490 \;,\; \theta_3 = \pi/2 \;,$$

and the configurations $\{\Theta_{13}, \Theta_{21}, \Theta_{22}\}$. Here each configuration is a 6-tuple $\Theta_{ij} = (\theta_i, \theta_i, \theta''_{ij}, \pi - \theta_j, \theta'_{ij}, \theta'_{ij})$, $\theta'_{ij} = \arccos(1 - \cos(\theta_i) + \cos(\pi - \theta_j))$, and $\theta'_{ij} = \arccos(-1 + 2\min\{|\cos\theta'_{ij}|, |\cos\theta_i|\})$. Note that both angles θ'_{ij}, and θ''_{ij} make at least one of the triangle constraints tight.

θ	$f_{3R}(\theta)$
0.000000	0.000000
0.704216	0.000000
1.048568	0.600507
1.248896	0.964164
1.250766	0.986882
1.320512	1.120350
1.382893	1.240981
1.570796	1.570796

Table 3: The piecewise linear rotation function used for the 1.21358 algorithm for MIN 3-SAT.

8 Hardness Results

In this section we use a slight modification of the gadget technique of Bellare *et al.* [BGS98] and Trevisan *et al.* [TSSW00] to obtain the following result:

Theorem 8.1 *For any $\varepsilon > 0$, MIN 2-SAT cannot be approximated within a factor of $15/14 - \varepsilon$, and MIN 3-SAT cannot be approximated within a factor of $7/6 - \varepsilon$, unless $P = NP$.*

Let $f : \{0,1\}^k \to \{0,1\}$ be a Boolean function of k variables. MIN CSP(f) is the problem of minimizing the total weight of satisfied clauses, where each clause is of the form $f(z_1, \ldots, z_k)$, and the z_i's are variables or negation of variables. We allow a variable to appear (with or without negation) several times in the same clause. In a similar way we define MAX CSP(f). Specifically, MAX CSP(XOR_3) is the problem in which each clause is of the form $z_1 \oplus z_2 \oplus z_3$, and MIN CSP($OR_k$) is MIN k-SAT.

Trevisan *et al.* [TSSW00] formalize the notion of a gadget for reducing one maximization problem MAX CSP(g) into a second maximization problem MAX CSP(f). Here we adapt their definition for the case in which we want to reduce a maximization problem MAX CSP(g) into a minimization problem MIN CSP(f):

Definition 8.2 (Gadget) *Let $g : \{0,1\}^\ell \to \{0,1\}$ and $f : \{0,1\}^k \to \{0,1\}$ be two Boolean functions. A weighted set of constraints $\Gamma = \{w_i \times f(z_{i1}, \ldots, z_{ik})\}$, where the w_i's are non-negative weights and the z_{ij}'s are taken from set $\{x_1, \ldots, x_l, y_1 \ldots, y_m, \bar{x}_1, \ldots, \bar{x}_l, \bar{y}_1, \ldots, \bar{y}_m\}$, is said to be an* α-gadget from g to f *iff the following two condition are satisfied:*

$$(i) \quad g(x_1, \ldots, x_\ell) = 1 \Longrightarrow \min_{y_1,\cdots,y_m} \sum_i w_i f(z_{i1}, \ldots, z_{ik}) = \alpha$$

$$(ii) \quad g(x_1, \ldots, x_\ell) = 0 \Longrightarrow \min_{y_1,\cdots,y_m} \sum_i w_i f(z_{i1}, \ldots, z_{ik}) \geq \alpha + 1$$

The variables $y_1, \ldots, y_m$ are called auxiliary variables.

An α-gadget is said to be strict, *if an equality holds in condition (ii).*

A (strict) α-gadget from g to f is said to be optimal, *iff for any $\alpha' < \alpha$ there is no (strict) α'-gadget from g to f.*

A 7-gadget from XOR_3 to OR_2, called Γ_{OR_2}, is given in Figure 3, and a 3-gadget from XOR_3 to OR_3, called Γ_{OR_3}, is given in Figure 4. Γ_{OR_2} has four auxiliary variables, while Γ_{OR_3} has no auxiliary variables. Both Γ_{OR_2} and Γ_{OR_3} are strict. In addition, as stated in Theorem 8.5 below, Γ_{OR_2} and Γ_{OR_3} are optimal.

Theorem 8.1 follows immediately from the following theorem and from the existence of the gadgets Γ_{OR_2}, and Γ_{OR_3}.

$\frac{2}{3}\times\ x_1 \vee y_1$	$\frac{1}{3}\times\ \bar{x}_1 \vee \bar{y}_1$	$\frac{2}{3}\times\ x_1 \vee y_2$	$\frac{1}{3}\times\ \bar{x}_1 \vee \bar{y}_2$
$\frac{2}{3}\times\ \bar{x}_1 \vee y_3$	$\frac{1}{3}\times\ x_1 \vee \bar{y}_3$	$\frac{2}{3}\times\ \bar{x}_1 \vee y_4$	$\frac{1}{3}\times\ x_1 \vee \bar{y}_4$
$\frac{2}{3}\times\ x_2 \vee y_1$	$\frac{1}{3}\times\ \bar{x}_2 \vee \bar{y}_1$	$\frac{2}{3}\times\ \bar{x}_2 \vee y_2$	$\frac{1}{3}\times\ x_2 \vee \bar{y}_2$
$\frac{2}{3}\times\ x_2 \vee y_3$	$\frac{1}{3}\times\ \bar{x}_2 \vee \bar{y}_3$	$\frac{2}{3}\times\ \bar{x}_2 \vee y_4$	$\frac{1}{3}\times\ x_2 \vee \bar{y}_4$
$\frac{2}{3}\times\ x_3 \vee y_1$	$\frac{1}{3}\times\ \bar{x}_3 \vee \bar{y}_1$	$\frac{2}{3}\times\ \bar{x}_3 \vee y_2$	$\frac{1}{3}\times\ x_3 \vee \bar{y}_2$
$\frac{2}{3}\times\ \bar{x}_3 \vee y_3$	$\frac{1}{3}\times\ x_3 \vee \bar{y}_3$	$\frac{2}{3}\times\ x_3 \vee y_4$	$\frac{1}{3}\times\ \bar{x}_3 \vee \bar{y}_4$

Figure 3: Γ_{OR_2} : The strict 7-gadget from XOR_3 to OR_2.

$1\times\ \bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3$	$1\times\ \bar{x}_1 \vee x_2 \vee x_3$	$1\times\ x_1 \vee \bar{x}_2 \vee x_3$	$1\times\ x_1 \vee x_2 \vee \bar{x}_3$

Figure 4: Γ_{OR_3} : The strict 3-gadget from XOR_3 to OR_3.

Theorem 8.3 *If there exists an α-gadget from XOR_3 to f, then for any $\varepsilon > 0$ there is no $\left(\frac{2\alpha+1}{2\alpha} - \varepsilon\right)$-approximation algorithm for MIN CSP(f), unless $P = NP$.*

Proof: Let $\{w_i \times f(z_{i1}, \ldots, z_{ik})\}$ be an α-gadget from XOR_3 to f. Assume, for the sake of contradiction, that MIN CSP(f) has a $\left(\frac{2\alpha+1}{2\alpha} - \varepsilon\right)$-approximation algorithm for some $\varepsilon > 0$.

We assume, without loss of generality, that all our MAX CSP(XOR_3) instances have total clauses' weight of 1. Håstad [Hås01] showed that for any $\varepsilon' > 0$, distinguishing between instances of maximal weight of at most $\frac{1}{2}+\varepsilon'$ and instances of maximal weight of at least $1-\varepsilon'$ is NP-hard. Choose $\varepsilon' > 0$ such that $\alpha + \frac{1}{2} - \varepsilon' > (\alpha+\varepsilon')(\frac{2\alpha+1}{2\alpha} - \varepsilon)$. Let I_{max} be an instance of MAX CSP(XOR_3). We construct an instance I_{min} of MIN CSP(f). We do so by replacing each clause, $w \times x_1 \oplus x_2 \oplus x_3$ (or $w \times x_1 \oplus x_2 \oplus \bar{x}_3$) of I_{max} by the set of clauses $\{(w \cdot w_i) \times f(z_{i1}, \ldots, z_{ik})\}$ which are induced by the α-gadget. For each XOR_3 clause we introduce new auxiliary variables. By the definition of an α-gadget, if the maximal weight of I_{max} is at most $\frac{1}{2} + \varepsilon'$, then the minimal weight of I_{min} is at least $\alpha(\frac{1}{2} + \varepsilon') + (\alpha+1)(\frac{1}{2} - \varepsilon') = \alpha + \frac{1}{2} - \varepsilon'$. In addition, if the maximal weight of I_{max} is at least $1 - \varepsilon'$, then the minimal weight of I_{min} is at most $\alpha(1-\varepsilon') + (\alpha+1)\varepsilon' = \alpha + \varepsilon'$. Apply the $\left(\frac{2\alpha+1}{2\alpha} - \varepsilon\right)$-approximation algorithm to I_{min}. If the maximal weight of I_{max} is at most $\frac{1}{2} + \varepsilon'$, then the algorithm yields an assignment of I_{min} of weight at least $\alpha + \frac{1}{2} - \varepsilon'$. If the maximal weight of I_{max} is at most $1 - \varepsilon'$, then the algorithm yields an assignment of I_{min} of weight at least $(\alpha+\varepsilon')(\frac{2\alpha+1}{2\alpha} - \varepsilon)$. By our choice of ε' these cases can be easily distinguished in contradiction with Håstad's result. □

In a similar way it can proven that a *strict* α-gadget from g to f implies that, for any $\varepsilon > 0$, an $\frac{\alpha+1-\beta-\varepsilon}{\alpha+1-\gamma}$-approximation algorithm for MIN CSP(f) can be used to distinguish between MAX CSP(g) instances with maximal weight at most β from instances with maximal weight at least γ (assuming the total weight of the clauses in the MAX CSP(g) instances is 1).

Next we would like to show that our gadgets from XOR_3 to OR_k are the best possible. We first prove the following lemma which is similar to Corollary 3.8 in Trevisan *et al.* [TSSW00].

Lemma 8.4 *For any Boolean function $f : \{0,1\}^k \to \{0,1\}$, with $k \geq 1$, there exists an optimal gadget from XOR_3 to f, that has at most 5 auxiliary variables.*

Proof: Let $\Gamma = \{w_i \times f(z_{i1}, \ldots, z_{ik})\}$ be an α-gadget from XOR_3 to f with auxiliary variables $y_1, \ldots, y_m$, where $m > 5$. We construct an α-gadget $\Gamma' = \{w_i \times f(z'_{i1}, \ldots, z'_{ik})\}$ from XOR_3 to f, that has only 5 auxiliary variables

$y'_1, \ldots, y'_5$. Here the z_{ij}'s are taken from the set $\{x_1, x_2, x_3, y_1 \ldots, y_m, \bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{y}_1, \ldots, \bar{y}_m\}$, while the z'_{ij}'s are taken from the set $\{x_1, x_2, x_3, y'_1 \ldots, y'_5, \bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{y}'_1, \ldots, \bar{y}'_5\}$. Consider condition (i) of Definition 8.2. There exist four assignments of $x = (x_1, x_2, x_3)$, that satisfy XOR_3, namely: $(0,0,1), (0,1,0), (1,0,0)$ and $(1,1,1)$. For each such triple x there exists an assignment of the auxiliary variables $y_1(x), \ldots, y_m(x)$, that attains the minimum in condition (i). That is, condition (i) defines a mapping from satisfying assignments of XOR_3 to the auxiliary variables $y_1, \ldots, y_m$. We treat each of $y_1(x), \ldots, y_m(x)$ as a column vector of length four. We now build a mapping between satisfying assignments of XOR_3, to the auxiliary variables $y'_1, \ldots, y'_5$ of the gadget Γ'. We define the mapping by the following 4×8 matrix

$$A_{XOR_3} = \begin{pmatrix} 0 & 0 & 1 & \; & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & & 1 & 0 & 0 & 0 & 0 \end{pmatrix} .$$

The matrix A_{XOR_3} is called the witness matrix for XOR_3. The first 3 columns of A_{XOR_3} are associated with the variables x_1, x_2, x_3, while the last 5 columns are associated with the auxiliary variables $y'_1, \ldots, y'_5$. Note that the columns of A_{XOR_3} contain all possible binary 4-tuples, up to negation. Hence, any of the vectors $y_1(x), \ldots, y_m(x)$ is equal, up to negation, to one of the columns of A_{XOR_3}. We now use this correspondence between the auxiliary variables $y_1(x), \ldots, y_m(x)$ to the columns of A_{XOR_3} to construct the gadget Γ': any occurrence of the auxiliary variable y_i in Γ, is replaced by the variable from $x_1, x_2, x_3, y'_1 \ldots, y'_5$ (or $\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{y}'_1, \ldots, \bar{y}'_5$) that is associated with the column $y_i(x)$ (or the column $\overline{y_i(x)}$). By our construction, for each satisfying assignment (x_1, x_2, x_3) of XOR_3, there is an assignment of $y'_1, \ldots, y'_5$, the one obtained by taking the row in A_{XOR_3} that starts with (x_1, x_2, x_3), which satisfies $\sum_i w_i f(z'_{i1}, \ldots, z'_{ik}) = \alpha$. In addition,

$$\min_{y'_1, \ldots, y'_5} \sum_i w_i f(z'_{i1}, \ldots, z'_{ik}) \geq \min_{y_1, \ldots, y_m} \sum_i w_i f(z_{i1}, \ldots, z_{ik}) .$$

The last inequality holds as the minimization in the left hand side can be viewed as the same minimization as in the right hand side, but with additional constraints of some variables being equal to others. Hence, Γ' satisfies conditions (i) and (ii) of Definition 8.2, and therefore, Γ' is an α-gadget. □

In a similar way it can be shown that if g is an ℓ-variable Boolean function, with p satisfying assignments, and f is any Boolean function, then there exists an optimal gadget from g to f with at most $2^{p-1} - \ell$ auxiliary variables. In this case (see the proof of the last lemma) the witness matrix A_g, has p rows and 2^{p-1} columns: the columns of all length p binary vectors up to negation. The first ℓ columns of A_g correspond to the satisfying assignments of g and the last $2^{p-1} - \ell$ correspond to the values of the auxiliary variables that attain the minimum in condition (i) (of Definition 8.2).

We conclude by proving the next theorem.

Theorem 8.5 *The gadgets Γ_{OR_2} and Γ_{OR_3} are optimal.*

Proof: Let A_{XOR_3} be the witness matrix of XOR_3 as discussed in the proof of Lemma 8.4. Let $A^{(i)}_{XOR_3}$ be the i-th row of A_{XOR_3} (where $1 \leq i \leq 4$). By the proof of Lemma 8.4, we may restrict ourselves to gadgets with 5 auxiliary variables $y_1, \ldots, y_5$, such that the minimum in condition (i) of Definition 8.2 is achieved, when $(x_1, x_2, x_3, y_1, \ldots, y_5) = A^{(i)}_{XOR_3}$. For brevity, we let

$$(z_1, \ldots, z_{16}) = (x_1, x_2, x_3, y_1, \ldots, y_5, \bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{y}_1, \ldots, \bar{y}_5) .$$

We first handle the case of OR_2. Consider the linear program (LP) below, with $1 + \binom{16}{2} = 121$ variables (α and $w_{j_1 j_2}$, for $1 \le j_1 < j_2 \le 16$), and $4 + 4 \cdot 2^5 + 4 \cdot 2^5 = 260$ constraints:

$$\begin{array}{lll}
 & \min \quad \alpha & (LP) \\
 & s.t. & \\
\forall i \in \{1, \ldots, 4\} & : (x_1, x_2, x_3, y_1, \ldots, y_5) = A^{(i)}_{XOR_3} & \\
 & \displaystyle\sum_{1 \le j_1 < j_2 \le 16} w_{j_1 j_2}(z_{j_1} \vee z_{j_2}) \ = \ \alpha & \\
\forall (x_1, x_2, x_3) \in \{0,1\}^3 & : x_1 \oplus x_2 \oplus x_3 = 1 \quad \forall (y_1, \ldots, y_5) \in \{0,1\}^5 & \\
 & \displaystyle\sum_{1 \le j_1 < j_2 \le 16} w_{j_1 j_2}(z_{j_1} \vee z_{j_2}) \ \ge \ \alpha & \\
\forall (x_1, x_2, x_3) \in \{0,1\}^3 & : x_1 \oplus x_2 \oplus x_3 = 0 \quad \forall (y_1, \ldots, y_5) \in \{0,1\}^5 & \\
 & \displaystyle\sum_{1 \le j_1 < j_2 \le 16} w_{j_1 j_2}(z_{j_1} \vee z_{j_2}) \ \ge \ \alpha + 1 &
\end{array}$$

Any gadget from XOR_3 to OR_2 is a feasible solution of the linear program (LP). We solved this linear program and got the 7-gadget Γ_{OR_2}. Therefore, Γ_{OR_2} is an optimal 7-gadget from XOR_3 to OR_2.

A similar linear program for constructing an optimal gadget from XOR_3 to OR_3 can be obtained. In this case the linear program will have $1 + \binom{16}{3} = 560$ variables, namely, α and $w_{j_1 j_2, j_3}$, where $1 \le j_1 < j_2 < j_3 \le 16$, and the same number of constraints as before. This linear program can be obtained by replacing each summation $\sum_{1 \le j_1 < j_2 \le 16} w_{j_1 j_2}(z_{j_1} \vee z_{j_2})$ in (LP) by the summation $\sum_{1 \le j_1 < j_2 < j_3 \le 16} w_{j_1 j_2 j_3}(z_{j_1} \vee z_{j_2} \vee z_{j_3})$. We solved the linear program and got the 3-gadget Γ_{OR_3}. Therefore, Γ_{OR_3} is an optimal 3-gadget from XOR_3 to OR_3. □

Similar linear programs, as in the proof of Theorem 8.5, for constructing optimal gadgets from arbitrary g to arbitrary f, can be obtained. In the general case, if g is an ℓ-variable Boolean function with p satisfying assignments, and f is a k-variable Boolean function, then the linear program would have $1 + \binom{2^p}{k}$ variables and $p + 2^{2^{p-1}}$ constraints.

9 Concluding remarks and open problems

We obtained substantially improved approximation algorithms for the MIN 2-SAT and the MIN 3-SAT problems. Our algorithms are randomized, but they can be derandomized using the techniques of Mahajan and Ramesh [MR99]. Obtaining further improved approximability and inapproximability results for these problems are challenging open problems.

References

[BdlV99] C. Bazgan and W. Fernandez de la Vega. A polynomial time approximation scheme for dense min2sat. In *Proc. 12th Int. Symp. on Fundamentals of Computation Theory, Lecture Notes in Comput. Sci. 1684*, pages 91–99. Springer-Verlag, 1999.

[Ber87] M. Berger. *Geometry II*. Springer-Verlag, 1987.

[BF99] P. Berman and T. Fujito. On approximation properties of the independent set problem for low degree graphs. *Theory of Computing Systems*, 32(2):115–132, 1999.

[BGS98] M. Bellare, O. Goldreich, and M. Sudan. Free bits, PCPs, and nonapproximability—towards tight results. *SIAM Journal on Computing*, 27:804–915, 1998.

[BK99] P. Berman and M. Karpinski. On some tighter inapproximability results (extended abstract). In *Proceedings of the 26th International Colloquium on Automata, Languages and Programming, Prague, Czech Republic*, pages 200–209, 1999.

[BTV99] D. Bertsimas, C. Teo, and R. Vohra. On dependent randomized rounding algorithms. *Oper. Res. Lett.*, 24(3):105–114, 1999.

[BYE81] R. Bar-Yehuda and S. Even. A linear-time approximation algorithm for the weighted vertex cover problem. *Journal of Algorithms*, 2(2):198–203, 1981.

[BYE85] R. Bar-Yehuda and S. Even. A local-ratio theorem for approximating the weighted vertex cover problem. *Annals of Discrete Mathematics*, 25:27–45, 1985.

[DS02] I. Dinur and S. Safra. The importance of being biased. In *Proceedings of the 33th Annual ACM Symposium on Theory of Computing, Montréal, Canada*, pages 33–42, 2002.

[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 3rd Israel Symposium on Theory and Computing Systems, Tel Aviv, Israel*, pages 182–189, 1995.

[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.

[Hal00] E. Halperin. Improved approximation algorithms for the vertex cover problem in graphs and hypergraphs. In *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, California*, pages 329–337, 2000.

[Hås01] J. Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48:798–859, 2001.

[Hoc82] D.S. Hochbaum. Approximation algorithms for the set covering and vertex cover problems. *SIAM Journal on Computing*, 11(3):555–556, 1982.

[Hoc83] D.S. Hochbaum. Efficient bounds for the stable set, vertex cover and set packing problems. *Discrete Applied Mathematics*, 6(3):243–254, 1983.

[Hoc00] D.S. Hochbaum. Instant recognition of polynomial time solvability, half itegrality and 2-approximations. In *Proceedongs of APPROX'00, Saarbrüken, Germany*, pages 2–14, 2000.

[HP00] D. Hochbaum and A. Pathria. Approximating a generalization of MAX 2SAT and MIN 2SAT. *Discrete Applied Mathematics*, 107(1-3):41–59, 2000.

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

[KKM94] R. Kohli, R. Krishnamurti, and P. Mirchandani. The minimum satisfiability problem. *Discrete Mathematics*, 7:275–283, 1994.

[KZ97] 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, Florida*, pages 406–415, 1997.

[LLZ02] M. Lewin, D. Livnat, and U. Zwick. Improved rounding techniques for the MAX 2-SAT and MAX DI-CUT problems. In *Proceedings of the 9th Conference on Integer Programming and Combinatorial Optimization, Cambridge, Massachusetts*, pages 67–82, 2002.

[MM01a] S. Matuura and T. Matsui. 0.863-approximation algorithm for MAX DICUT. In *Approximation, Randomization and Combinatorial Optimization: Algorithms and Techniques, Proceedongs of APPROX-RANDOM'01, Berkeley, California*, pages 138–146, 2001.

[MM01b] S. Matuura and T. Matsui. 0.935-approximation randomized algorithm for MAX 2SAT and its derandomization. Technical Report METR 2001-03, Department of Mathematical Engineering and Information Physics, the University of Tokyo, Japan, September 2001.

[MR96] M.V. Marathe and S.S. Ravi. On approximation algorithms for the minimum satisfiability problem. *Information Processing Letters*, 58:23–29, 1996.

[MR99] S. Mahajan and H. Ramesh. Derandomizing approximation algorithms based on semidefinite programming. *SIAM Journal on Computing*, 28:1641–1663, 1999.

[MS85] B. Monien and E. Speckenmeyer. Ramsey numbers and an approximation algorithm for the vertex cover problem. *Acta Informatica*, 22(1):115–123, 1985.

[TSSW00] L. Trevisan, G.B. Sorkin, M. Sudan, and D.P. Williamson. Gadgets, approximation, and linear programming. *SIAM Journal on Computing*, 29:2074–2097, 2000.

[Zwi02] U. Zwick. Computer assisted proof of optimal approximability results. In *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, California*, pages 496–505, 2002.