Improved rounding techniques for the MAX 2-SAT and MAX DI-CUT problems *

Michael Lewin, *Dror Livnat*, and *Uri Zwick*

School of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel.
E-mail: {mikel,dror,zwick}@tau.ac.il.

Abstract. Improving and extending recent results of Matuura and Matsui, and less recent results of Feige and Goemans, we obtain improved approximation algorithms for the MAX 2-SAT and MAX DI-CUT problems. These approximation algorithms start by solving semidefinite programming relaxations of these problems. They then *rotate* the solution obtained, as suggested by Feige and Goemans. Finally, they round the rotated vectors using random hyperplanes chosen according to *skewed* distributions. The performance ratio obtained by the MAX 2-SAT algorithm is at least 0.940, while that obtained by the MAX DI-CUT algorithm is at least 0.874. We show that these are essentially the best performance ratios that can be achieved using any combination of prerounding rotations and skewed distributions of hyperplanes, and even using more general families of rounding procedures. The performance ratio obtained for the MAX 2-SAT problem is fairly close to the inapproximability bound of about 0.954 obtained by Håstad. The performance ratio obtained for the MAX DI-CUT problem is very close to the performance ratio of about 0.878 obtained by Goemans and Williamson for the MAX CUT problem.

1 Introduction

In a seminal paper, Goemans and Williamson [GW95] used semidefinite programming to obtain a 0.878-approximation algorithm for the MAX CUT and MAX 2-SAT problems, and a 0.796-approximation algorithm for the MAX DI-CUT problem. These algorithms solve semidefinite programming relaxations of the problems, and then round the obtained solution using random hyperplanes. The normal vector of this random hyperplane is uniformly distributed over the n-dimensional sphere. Feige and Goemans [FG95] improved the last two of these results by obtaining a 0.931-approximation algorithm for the MAX 2-SAT problem, and a 0.859-approximation algorithms for the MAX DI-CUT problem. (A tighter analysis of these algorithms is presented in Zwick [Zwi00].) The

* This research was supported by the ISRAEL SCIENCE FOUNDATION (grant no. 246/01).

improved approximation ratios are obtained by *rotating* the vectors obtained as the solutions of the semidefinite programming relaxations before rounding them using a random hyperplane. The normal of the random hyperplane used is again uniformly distributed over the $(n+1)$-dimensional sphere. In their paper, Feige and Goemans [FG95] suggest several other techniques that may be used to obtain improved approximation algorithms for these problems. One of these suggestions is that the normal of the random hyperplane used to round the vectors should be chosen according to a distribution that is *not* uniform on the $(n+1)$-dimensional sphere, but rather *skewed* toward, or away, from v_0. (In the relaxation of the MAX 2-SAT problem, the vector v_0 is used to represent the value `false`.) Feige and Goemans [FG95], however, do not explore this possibility. The first to explore this possibility were Matuura and Matsui [MM01a,MM01b]. Overcoming technical difficulties that stand in the way of employing the skewed distribution idea, they obtain in [MM01a] a 0.863-approximation algorithm for the MAX DI-CUT problem, and in [MM01b] a 0.935-approximation algorithm for the MAX 2-SAT problem. In [MM01b] they also show, partially relying on results of Mahajan and Ramesh [MR99], that these algorithms can be derandomized.

Matuura and Matsui [MM01a,MM01b] obtained their approximation algorithms by using skewed distributions taken from a specific, easy to work with, family of skewed distributions. We conduct a *systematic search* for skewed distributions that yield the *best* performance ratios for the MAX 2-SAT and MAX DI-CUT problems. As a result, we obtain further improved approximation ratios, namely 0.937 for MAX 2-SAT and 0.866 for MAX DI-CUT. These are essentially the best performance ratios that can be used by using skewed distribution of hyperplanes on their own.

We obtain further improved results, however, by *combining* the skewed hyperplane technique with pre-rounding rotations. For the MAX 2-SAT problem we get an approximation ratio of at least 0.940. For the MAX DI-CUT problem we get an approximation ratio of at least 0.874. These are essentially the best ratios that can be obtained using *any* combination of pre-rounding rotations and skewed distributions of hyperplanes, and even using more general families of rounding procedures.

Håstad [Hås97], using gadgets of Trevisan *et al.* [TSSW00], showed that for any $\epsilon > 0$, if there is a $(\frac{21}{22}+\epsilon)$-approximation algorithm for the MAX 2-SAT problem, or a $(\frac{12}{13}+\epsilon)$-approximation algorithm for the MAX DI-CUT problem, then P=NP. (Note that $\frac{21}{22} \simeq 0.95454$ and that $\frac{12}{13} \simeq 0.92308$. There are still gaps, therefore, between the best approximation and inapproximability ratios for the MAX 2-SAT and MAX DI-CUT problems. The gap for the MAX 2-SAT problem, between 0.940 and 0.954, is quite small. The gap for the MAX DI-CUT is larger. But, the ratio of 0.874 we obtain for the MAX DI-CUT problem is very close to the best ratio, of about 0.878, obtained by Goemans and Williamson [GW95] for the MAX CUT problem, which may be seen as a subproblem of the MAX DI-CUT problem.

The rest of this paper is organized as follows. In the next section we formally define the MAX 2-SAT and MAX DI-CUT problems and describe their semidefinite programming relaxations. We also briefly sketch there the approximation algorithms of [GW95] and [FG95]. In Section 3 we discuss families of rounding procedures that can be used to round configuration of vectors obtained by solving semidefinite programming relaxations. In Section 4 we explain how to compute certain probabilities that appear in the analysis of approximation algorithms that use rounding procedures from the families defined in Section 3. In Section 5 we describe the computation of lower bounds on the performance ratios achieved by such algorithms. In Section 6 we describe the essentially best approximation algorithms that can be obtained using rounding procedures from the family $\mathcal{SKEW}$. In Section 7 we describe our best approximation algorithms. These algorithms use rounding procedures from the family $\mathcal{ROT}$+$\mathcal{SKEW}$. In Section 8 we show that these algorithms are essentially the best algorithms that can be obtained using rounding procedures from $\mathcal{ROT}$+$\mathcal{SKEW}$. We end in Section 9 with some concluding remarks.

2 Preliminaries

An instance of MAX 2-SAT in the Boolean variables $x_1, \ldots, x_n$ is composed of a collection of *clauses* $C_1, \ldots, C_m$ with non-negative weights $w_1, \ldots, w_m$ assigned to them. Each clause C_i is of the form z_1 or of the form $z_1 \vee z_2$ where each z_j is either a variable x_k or its negation $\bar{x}_k$. (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 maximized. A clause is satisfied if at least one of the literals appearing in it is assigned the value 1. (If a variable is assigned the value 0 then its negation is assigned the value 1, and vice versa.)

It is convenient to let $x_{n+i} = \bar{x}_i$, for $1 \leq i \leq n$, and also to let $x_0 = 0$. Each clause is then of the form $x_i \vee x_j$ where $0 \leq i, j \leq 2n$. An instance of MAX 2-SAT can then be encoded as an array (w_{ij}), where $0 \leq i, j \leq 2n$, where w_{ij} is interpreted as the weight of the clause $x_i \vee x_j$. The goal is then

$$
\begin{array}{l}
\text{Max } \frac{1}{4}\sum_{i,j} w_{ij}(3 - v_0\cdot v_i - v_0\cdot v_j - v_i\cdot v_j) \\
\qquad\qquad \begin{array}{cll}
v_0\cdot v_i + v_0\cdot v_j + v_i\cdot v_j \ge -1 & , & 1 \le i,j \le 2n \\
v_i\cdot v_{n+i} = -1 & , & 1 \le i \le n \\
v_i \in \mathbb{R}^{n+1}\ ,\ v_i\cdot v_i = 1 & , & 0 \le i \le 2n
\end{array}
\end{array}
$$

Fig. 1. A semidefinite programming relaxation of MAX 2-SAT

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 maximized.

The semidefinite programming relaxation of the MAX 2-SAT problem used by Feige and Goemans [FG95] is given in Figure 1. In this relaxation, a unit vector v_i is assigned to each literal x_i, where $0 \le i \le 2n$. As $x_0 = 0$, the vector v_0 corresponds to the constant 0 (`false`). To ensure consistency, we require $v_i \cdot v_{n+i} = -1$, for $1 \le i \le n$. As the value of a solution of the SDP program depends only on the inner products between the vectors, we can assume, with out loss of generality, that $v_0 = (1, 0, \ldots, 0) \in \mathbb{R}^{n+1}$. Note that as the 'triangle constraints' holds for any $1 \le i, j \le 2n$, and as $v_i = -v_{n+i}$, for $1 \le i \le n$, we get that if v_0, v_i and v_j appear in a feasible solution of the SDP, then we have:

$$
\begin{array}{rl}
v_0\cdot v_i + v_0\cdot v_j + v_i\cdot v_j \ge -1 & , \\
-v_0\cdot v_i - v_0\cdot v_j + v_i\cdot v_j \ge -1 & , \\
-v_0\cdot v_i + v_0\cdot v_j - v_i\cdot v_j \ge -1 & , \\
v_0\cdot v_i - v_0\cdot v_j - v_i\cdot v_j \ge -1 & .
\end{array}
$$

We can strengthen the relaxation by adding the constraints $v_i \cdot v_j + v_i \cdot v_k + v_j \cdot v_k \ge -1$, for any $1 \le i, j, k \le 2n$, but we do not know how to take advantage of these additional constraints in our analysis, so we do not know whether this helps.

An instance of MAX DI-CUT is a directed graph $G = (V, E)$, with non-negative weights assigned to its edges. The goal is to find a partition of V into $(S, V - S)$, where $S \subseteq V$, such that the total weight of the edges that cross this cut, i.e., edges $u \to v$ such that $u \in S$ but $v \notin S$, is maximized.

It is convenient to assume that $V = \{1, 2, \ldots, n\}$. An instance of MAX DI-CUT can then be represented as an array (w_{ij}), where $1 \le i, j \le n$, where w_{ij} is interpreted as the weight of the edge $i \to j$. Note that $w_{ij} \ge 0$, for every $1 \le i, j \le n$. We can also assume that $w_{ii} = 0$, for $1 \le i \le n$.

The semidefinite programming relaxation of the MAX DI-CUT problem used by Feige and Goemans [FG95] is given in Figure 2. Here, there is

$$
\begin{array}{l}
\text{Max } \frac{1}{4}\sum_{i,j} w_{ij}(1 + v_0\cdot v_i - v_0\cdot v_j - v_i\cdot v_j) \\
\begin{array}{rcl}
v_0\cdot v_i + v_0\cdot v_j + v_i\cdot v_j \geq -1 & , & 1\leq i,j\leq n \\
-v_0\cdot v_i - v_0\cdot v_j + v_i\cdot v_j \geq -1 & , & 1\leq i,j\leq n \\
-v_0\cdot v_i + v_0\cdot v_j - v_i\cdot v_j \geq -1 & , & 1\leq i,j\leq n \\
v_0\cdot v_i - v_0\cdot v_j - v_i\cdot v_j \geq -1 & , & 1\leq i,j\leq n \\
v_i\in \mathbb{R}^{n+1}\ ,\ v_i\cdot v_i = 1 & , & 0\leq i\leq n
\end{array}
\end{array}
$$

Fig. 2. A semidefinite programming relaxation of MAX DI-CUT

a unit vector v_i corresponding to each vertex $1 \leq i \leq n$ of the graph, and there is also a vector v_0, corresponding to the side S of the directed cut. Again, we can add more triangle constraints, but it is not clear whether this helps.

The approximation algorithms of Goemans and Williamson [GW95] solve weakened versions of the SDP relaxations, they do not use the triangle constraints, and then *round* the resulting solution using a random hyperplane. More specifically, they choose a random vector $r = (r_0, r_1, \ldots, r_n)$ according a standard $(n+1)$-dimensional normal distribution. The random variables r_i, for $0 \leq i \leq n$, are i.i.d. random variables distributed according to the standard normal distribution. The normalized vector $r/||r||$ is then uniformly distributed on S^n, the unit sphere in $\mathbb{R}^{n+1}$. For a MAX 2-SAT instance, the literal x_i is assigned the value 0 if and only if v_i and v_0 lie on the same side of the hyperplane whose normal is r, or equivalently, if $sgn(v_i \cdot r) = sgn(v_0 \cdot r)$. For a MAX DI-CUT instance, a vertex i is placed in S if and only if v_i and v_0 lie on the same side of the random hyperplane.

The approximation algorithms of Feige and Goemans [FG95] solve the SDP relaxations given in Figures 1 and 2. Before rounding the resulting vectors, they rotate them using a rotation function $f : [0,\pi] \to [0,\pi]$ that satisfies $f(\pi - \theta) = \pi - f(\theta)$, for $0 \leq \theta \leq \pi$. The vector v_i, forming an angle $\theta_i = \arccos(v_0 \cdot v_i)$ with v_0, is rotated, in the 2-dimensional plane spanned by v_0 and v_i, into a vector v_i' whose angle $\theta_i' = \arccos(v_0 \cdot v_i')$ with v_0 satisfies $\theta_i' = f(\theta_{0i})$.

3 Families of rounding procedures

Feige and Goemans [FG95] suggest the following very general family of rounding procedures:

$\mathcal{GEN}$: Let $F : \mathbb{R}^2 \times [-1,1] \to \{0,1\}$ be an arbitrary function. Let $r = (r_0, r_1, \ldots, r_n)$ be a standard $(n+1)$-dimensional random variable. For $1 \le i \le n$, let $x_i \leftarrow F(v_0 \cdot r, v_i \cdot r, v_0 \cdot v_i)$.

For consistency, the function F should satisfy $F(x, -y, -z) = 1 - F(x, y, z)$ for every $x, y \in \mathbb{R}$ and $z \in [-1,1]$. In the spirit of the RPR^2 rounding procedure of Feige and Langberg [FL01], this can be further generalized to:

$\mathcal{GEN}^+$: Let $F : \mathbb{R}^2 \times [-1,1] \to [0,1]$ be an arbitrary function. Let $r = (r_0, r_1, \ldots, r_n)$, where $r_0, r_1, \ldots, r_n$ are independent standard normal variables. For $1 \le i \le n$, let $x_i \leftarrow 1$, independently with probability $F(v_0 \cdot r, v_i \cdot r, v_0 \cdot v_i)$.

The basic random hyperplane rounding technique of Goemans and Williamson [GW95] is of course a very special case of general family with

$$F(v_0 \cdot r, v_i \cdot r, v_0 \cdot v_i) \;=\; [\,(v_0 \cdot r)(v_i \cdot r) \le 0\,]\;,$$

where $[a \le b]$ gets the value 1 if $a \le b$, and 0, otherwise. We can assume, without loss of generality that $v_0 = (1, 0, \ldots, 0)$, and that $r_0 \ge 0$. The expression for F is then further simplified to

$$F(v_0 \cdot r, v_i \cdot r, v_0 \cdot v_i) \;=\; [\,(v_i \cdot r) \le 0\,]\;.$$

To obtain their improved MAX 2-SAT and MAX DI-CUT algorithms, Feige and Goemans [FG95] do not use the full generality of the $\mathcal{GEN}$ family. They use, instead the following more restricted family:

$\mathcal{ROT}$: Let $f : [0,\pi] \to [0,\pi]$ be an arbitrary *rotation* function. Let $\theta_i = \arccos(v_0 \cdot v_i)$ be the angle between v_0 and v_i. Rotate v_i into a vector v_i' that forms an angle of $\theta_i' = f(\theta_i)$ with v_0. Round the vectors $v_0, v_1', \ldots, v_n'$ using the random hyperplane technique of Goemans and Williamson, i.e., let $x_i \leftarrow [\,(v_0 \cdot r)(v_i' \cdot r) \le 0\,]$, for $1 \le i \le n$.

For consistency, the function f should satisfy $f(\pi - \theta) = \pi - f(\theta)$. Feige and Goemans [FG95] also suggest, but do not analyze, the following *skewed hyperplane* rounding technique:

$\mathcal{SKEW}$: Let $r = (r_0, r_1, \ldots, r_n)$ be a *skewed* normal vector, i.e., $r_1, r_2, \ldots, r_n$ are still independent standard normal variables, but r_0 is chosen according to a different distribution. Again, let $x_i \leftarrow [\,(v_0 \cdot r)(v_i \cdot r) \le 0\,]$, for $1 \le i \le n$.

It may seem, at first sight, that $\mathcal{SKEW}$ is not a sub-family of $\mathcal{GEN}$, as r_0 in $\mathcal{SKEW}$ is not a standard normal variable. We will see shortly, however, that $\mathcal{SKEW}$ is indeed a sub-family of $\mathcal{GEN}$, and it is not necessary, therefore, to further generalize $\mathcal{GEN}$ by allowing r_0 to be chosen according to a non-normal distribution.

Matuura and Matsui [MM01a,MM01b] take up the challenge of analyzing this technique, in a slightly different form. They obtain some improved results but do not consider *all* rounding procedures from this family, and hence do not exploit the full power of $\mathcal{SKEW}$. In Section 6 we obtain the essentially best algorithms for the MAX 2-SAT and MAX DI-CUT problems that can be obtained using rounding procedures for $\mathcal{SKEW}$.

We obtained further improved algorithms, however, by considering combinations of pre-rounding rotations and skewed hyperplane distributions. We denote the family of rounding procedures containing all these combinations by $\mathcal{ROT}+\mathcal{SKEW}$. In Section 7 we obtain the essentially best algorithms for the MAX 2-SAT and MAX DI-CUT problems that can be obtained using rounding procedures from this family. These are the best approximation algorithms that we get for these two problems. We have not been able to improve these algorithms using more general rounding techniques. Extensive numerical experiments that we have conducted suggest that, at least for the MAX 2-SAT problem, no further improvements are possible, even using rounding procedures from $\mathcal{GEN}^+$.

Our next task will be to determine the form of the function $F(x, y, z)$ that corresponds to rounding procedures from the family $\mathcal{ROT}+\mathcal{SKEW}$. Before continuing, it would be convenient to present an equivalent formulation of $\mathcal{GEN}$ (and $\mathcal{GEN}^+$). We assume, as stated, that $v_0 = (1, 0, \ldots, 0)$. Any vector $r = (r_0, r_1, \ldots, r_n) \in \mathbb{R}^{n+1}$ can thus be written as

$$r = r_0 v_0 + r^\perp ,$$

where, $r^\perp = r - r_0 v_0 = (0, r_1, \ldots, r_n)$. Clearly, $v_0 \cdot r^\perp = 0$. We can therefore replace the three parameters $v_0 \cdot r, v_i \cdot r$ and $v_0 \cdot v_i$, by the three parameters $r_0, v_i \cdot r^\perp$ and $v_0 \cdot v_i$. As $v_i \cdot r = r_0 + v_i \cdot r^\perp$, we can easily move from one set of parameters to the other. Instead of using an arbitrary function $F(v_0 \cdot r, v_i \cdot r, v_0 \cdot v_i)$, we can, therefore, use an arbitrary function $G(r_0, v_i \cdot r^\perp, v_0 \cdot v_i)$. Note that r_0 is now an argument of G. By applying an appropriate function on r_0 we can get a random variable distributed according to any desired distribution.

Let v_i' be the vector obtained by rotating the vector v_i, with respect to v_0, using the rotation function f. Let $g(x) = f(\arccos x)$, for $-1 \le x \le 1$. Also let $x_i = v_0 \cdot v_i$. As v_i' lies in the 2-dimensional space spanned by v_0 and v_i, we get that $v_i = \alpha v_0 + \beta v_i$, for some $\alpha, \beta \in \mathbb{R}$. By the definition of the rotation process, we have $\beta \geq 0$. We can get two equations for α and β:

$$\begin{aligned} v_i' \cdot v_i' \;&=\; (\alpha v_0 + \beta v_i) \cdot (\alpha v_0 + \beta v_i) \;=\; \alpha^2 + \beta^2 + 2\alpha\beta x_i \;=\; 1 \;, \\ v_0 \cdot v_i' \;&=\; v_0 \cdot (\alpha v_0 + \beta v_i) \;=\; \alpha + \beta x_i \;=\; g(x_i) \;. \end{aligned}$$

Solving these equations, keeping in mind that $\beta \geq 0$, we get that:

$$\begin{aligned} \alpha \;&=\; g(x_i) - x_i\sqrt{\tfrac{1-g^2(x_i)}{1-x_i^2}} \;=\; \cos\theta_i' - \cot\theta_i \sin\theta_i' \;, \\ \beta \;&=\; \sqrt{\frac{1-g^2(x_i)}{1-x_i^2}} \;=\; \frac{\sin\theta_i'}{\sin\theta_i} \;. \end{aligned}$$

As

$$v_i' \cdot r \;=\; (\alpha v_0 + \beta v_i) \cdot (r_0 v_0 + r^{\perp}) \;=\; (\alpha + \beta x) r_0 + \beta v_i \cdot r^{\perp} \;,$$

the condition $v_i' \cdot r \leq 0$ of $\mathcal{ROT}$+$\mathcal{SKEW}$ is equivalent to:

$$v_i \cdot r^{\perp} \;\leq\; -\frac{\alpha + \beta x_i}{\beta} r_0 \;=\; -\cot\theta_i' \sin\theta_i \, r_0 \;=\; -\cot f(\theta_i) \sin\theta_i \, r_0 \;.$$

We can also define $v_i^{\perp} = v_i - (v_0 \cdot v_i)$. Clearly $v_0 \cdot v_i^{\perp} = 0$, and therefore $v_i \cdot r^{\perp} = v_i^{\perp} \cdot r^{\perp}$. The above condition is, therefore, also equivalent to the condition $v_i^{\perp} \cdot r^{\perp} \leq -\cot f(\theta_i) \sin\theta_i \, r_0$.

This leads us to the definition of the following family of rounding procedures that properly contains $\mathcal{ROT}$+$\mathcal{SKEW}$:

$\mathcal{THRESH}$: Let $r = (r_0, r_1, \ldots, r_n)$, where $r_1, r_2, \ldots, r_n$ are independent standard normal variables, and r_0 is an independent random variable distributed according to an arbitrary distribution. (Without loss of generality, r_0 assumes only nonnegative values.) Let $r^{\perp} = (0, r_1, \ldots, r_n)$. Let $T : \mathbb{R} \times [-1, 1] \to \mathbb{R}$ be a *threshold* function. Let $x_i \leftarrow [\, v_i \cdot r^{\perp} \leq T(r_0, v_0 \cdot v_i) \,]$, for $1 \leq i \leq n$.

From the above discussion, we see that $\mathcal{ROT}$+$\mathcal{SKEW}$ is just the subfamily of $\mathcal{THRESH}$ obtained by requiring the threshold function T to be of the form $T(r_0, x_i) = r_0 S(x_i)$, i.e., multiplicative in r_0. (We simply take $S(x) = -\cot f(\arccos x)\sqrt{1-x^2}$, or $f(\theta) = \arctan(-\frac{\sin\theta}{S(\cos\theta)})$.)

Interestingly, our best rounding procedures for the MAX 2-SAT and the MAX DI-CUT problems choose r_0 according to a very simple distribution. They just pick a *fixed* value for r_0! They fall, therefore, to the following sub-family of $\mathcal{THRESH}$:

$\mathcal{THRESH}^-$: Let $r = (0, r_1, \ldots, r_n)$, where $r_1, r_2, \ldots, r_n$ are independent standard normal variables. Let $S : [-1,1] \to \mathbb{R}$ be a *threshold* function. Let $x_i \leftarrow [\, v_i \cdot r \leq S(v_0 \cdot v_i)\,]$, for $1 \leq i \leq n$.

In rounding procedures from $\mathcal{THRESH}^-$, the normal vector r is chosen in the n-dimensional space orthogonal to v_0 and $r^\perp = r$. $\mathcal{THRESH}^-$ is the sub-family of $\mathcal{ROT}$+$\mathcal{SKEW}$ in which r_0 assumes a fixed value. Note that the exact value assumed by r_0 is not really important. Let us represent a rounding procedure from $\mathcal{ROT}$+$\mathcal{SKEW}$ using a pair (a, f), where a is the fixed value assumed by r_0, and f is the rotation function used. Then, it is easy to check that the rounding procedure (a_1, f_1) is equivalent to the rounding procedure (a_2, f_2), where $\cot f_2(\theta) = \frac{a_1}{a_2} \cot f_1(\theta)$. Thus, the choice of the fixed value assumed by r_0 is, in fact, arbitrary. In our best rounding procedures, the rotation functions used are linear. The exact value assumed by r_0 does play a role in that case. Note that the exact values assumed by r_0 are also important when no rotation function is used, as is the case in Section 6.

Our best algorithms for the MAX 2-SAT and MAX DI-CUT problems use rounding procedures from the family $\mathcal{ROT}$+$\mathcal{SKEW}$. Our MAX 2-SAT algorithm, however, is essentially the best algorithm that can be obtained using any rounding procedure from $\mathcal{THRESH}$. Our MAX DI-CUT algorithm is essentially the best algorithm that can be obtained using any rounding procedure from $\mathcal{THRESH}^-$. At least for the MAX 2-SAT problem, we believe that no better rounding procedure can be found even in $\mathcal{GEN}^+$, the most general family of rounding procedures defined above. (This belief is supported by some numerical experimentation. We will elaborate on this point in the full version of the paper.)

4 Computing probabilities

The analysis of most semidefinite programming based approximation algorithm is *local* in nature. In the analysis of the MAX 2-SAT and MAX DI-CUT algorithms that are presented in the next sections, we only have to evaluate the probability that a certain clause is satisfied (in the case of MAX 2-SAT) or that a certain edge is cut (in the case of MAX DI-CUT). Let $prob(v_1, v_2)$, where $v_1, v_2 \in \mathbb{R}^{n+1}$ be the probability that the corresponding variables x_1 and x_2 both receive the value 1 when the vectors v_1 and v_2 are rounded using the chosen rounding procedure. Then, the probability that a 2-SAT clause is satisfied is $1 - prob(-v_1, -v_2)$ and the probability that a directed edge is cut is $prob(-v_1, v_2)$.

To evaluate such a probability, it is enough to consider the projection of r on the three dimensional space spanned by v_0, v_1 and v_2. Let $r' = (r_0, r_1', r_2')$ be the projection of r on this space. It is easy to see that r_0, r_1', r_2' are still independent random variables, and that r_1' and r_2' are standard normal variables. We may assume, therefore, without loss of generality, that v_0, v_1 and v_2 lie in $\mathbb{R}^3$. We may also assume that $v_1 = (a_0, a_1, 0)$ and $v_2 = (b_0, b_1, b_2)$, for some $a_0, a_1, b_2 \in \mathbb{R}^+$ and $b_0, b_1 \in \mathbb{R}$. We also let $r = (r_0, r_1, r_2)$ and $r^\perp = (0, r_1, r_2)$.

Suppose now that the vectors v_1 and v_2 are rounded using a rounding procedure from $\mathcal{GEN}^+$, the most general family of rounding procedures considered in the previous section. Let $G : \mathbb{R}^2 \times [-1, 1] \to [0, 1]$ be the rounding function used. By definition, we then have:

$$\mathit{prob}(v_1, v_2) \;=\; \int_0^\infty \left[\int_{-\infty}^\infty \int_{-\infty}^\infty G(x, a_1 y, a_0) G(x, b_1 y + b_2 z, b_0) \phi(y)\phi(z) dy dz \right] f(x) dx \ ,$$

where $\phi(y) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-x^2/2}$ is the density function of a standard normal variable, and $f(x)$ is the density function of the random variable r_0. If r_0 is a discrete random variable that assumes the value s_k with probability p_k, for $1 \le k \le \ell$, then the outer integral is replaced by a sum:

$$\mathit{prob}(v_1, v_2) \;=\; \sum_{k=1}^{\ell} p_k \left[\int_{-\infty}^\infty \int_{-\infty}^\infty G(s_k, a_1 y, a_0) G(s_k, b_1 y + b_2 z, b_0) \phi(y)\phi(z) dy dz \right] \ .$$

If the rounding procedure belongs to $\mathcal{THRESH}$, so that $G(r_0, v_i \cdot r^\perp, v_0 \cdot v_i) = [v_i \cdot r^\perp \le T(r_0, v_0 \cdot v_i)]$, then we have a further simplification:

$$\mathit{prob}(v_1, v_2) \;=\; \sum_{k=1}^{\ell} p_k \left[\iint_{\substack{a_1 y \le T(s_k, a_0) \\ b_1 y + b_2 z \le T(s_k, b_0)}} \phi(y)\phi(z) dy dz \right] \ .$$

Note that the conditions $a_1 y \le T(s_k, a_0)$ and $b_1 y + b_2 z \le T(s_k, b_0)$, which are equivalent to the conditions $v_1 \cdot r^\perp \le T(r_0, v_0 \cdot v_1)$ and $v_2 \cdot r^\perp \le T(r_0, v_0 \cdot v_2)$, define a wedge in the y-z plane, the plane perpendicular to v_0. (Note that $T(s_k, a_0)$ and $T(s_k, b_0)$ here are constant.) This wedge integral can be evaluated numerically. It is not difficult to reduce it to single integration over the normal distribution function $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \phi(y) dy$.

When the rounding procedure is from the family $\mathcal{SKEW}$, or $\mathcal{SKEW}+\mathcal{ROT}$, further simplifications are possible, and $\mathit{prob}(v_1, v_2)$ can be evaluated using a single integration, even if r_0 is a continuous random distribution. More details would appear in the full version of the paper.

5 Computing performance ratios

In this section we describe the standard way in which we get a lower bound on the performance ratio achieved by our algorithms. For concreteness, we consider the MAX 2-SAT problem. The details for the MAX DI-CUT problem are analogous.

Let $P(v_i, v_j) \ = \ 1 - prob(-v_i, -v_j)$ be the probability that the clause $x_i \vee x_j$ is satisfied by the assignment produced by the chosen rounding procedure. Let

$$V(v_i, v_j) \ = \ \tfrac{1}{4}(3 - v_0 \cdot v_i - v_0 \cdot v_j - v_i \cdot v_j)$$

be the contribution of this clause to the value of the semidefinite program. As shown by Goemans and Williamson [GW95], the performance ratio of the MAX 2-SAT algorithm is at least

$$\alpha \ = \ \min_{(v_i,v_j)\in\Omega} \frac{P(v_i, v_j)}{V(v_i, v_j)} \ ,$$

where Ω is the set of all $(v_i, v_j) \in \mathbb{R}^3 \times \mathbb{R}^3$ that satisfy all the constraints of the semidefinite program. We evaluated such minimums numerically, using the optimization toolbox of Matlab.

To obtain an upper bound the performance ratio that can be obtained using *any* rounding procedure from a given family, we select a finite set of configurations $C = \{(v_1^i, v_2^i) \mid 1 \le i \le t\}$. We then estimate, numerically, that maximum, over all rounding procedures from the given family, of the minimum

$$\bar{\alpha} \ = \ \min_{i=1}^{t} \frac{P(v_1^i, v_2^i)}{V(v_1^i, v_2^i)} \ .$$

This bounding technique is implicit in Feige and Goemans [FG95] and is described explicitly in Zwick [Zwi00].

The computation of α, for a given rounding procedure, and of $\bar{\alpha}$, for a given family of rounding procedure, are very difficult global optimization problems. We have made extensive experiments to verify the numerical bounds that we state in this paper. However, we cannot claim them as theorems. It is possible, at least in principle, to obtain rigorous proofs for our claims using a tool such as $\mathcal{R}$eal$\mathcal{S}$earch (see [Zwi02]), but this would require a tremendous amount of work.

As mentioned, our best algorithms for the MAX 2-SAT and MAX DI-CUT problems simply set r_0 to be a constant. This may seem very surprising. We next try to explain why this is not so surprising after all.

Let C be a set of t configurations used to obtain an upper bound of $\bar{\alpha}$ on the performance ratio that can be achieved using rounding procedures from a given family. Consider now the search for rounding procedures that actually achieve a performance ratio of at least $\bar{\alpha}$ on each one of the t configurations of the set. Such optimal rounding procedures, for C, may choose r_0 according to an arbitrary continuous distribution. However, we can discretize the problem and assume that r_0 assumes only ℓ different values, where ℓ may be a huge number. If ℓ is large enough, we loose very little in this process. Let us assume that r_0 assumes the value s_i with probability p_i, for $1 \le i \le \ell$.

Let R_{ij} be the ratio obtained for the j-th configuration when r_0 assumes the value s_i. To find the optimal probabilities in which the values $s_1, s_2, \ldots, s_\ell$ should be chosen, we solve the following linear program:

$$\begin{array}{lll} \max \alpha & & \\ \text{s.t.} & \alpha \le \sum_{i=0}^{\ell} R_{ij} p_i & , \ 1 \le j \le t \\ & \sum_{i=0}^{\ell} p_i = 1 & \\ & p_i \ge 0 & , \ 1 \le i \le \ell \end{array}$$

This linear program has a basic optimal solution with at most t of the p_i's are non-zero. (Note that there are $\ell+1$ variables and only $t+1$ inequalities other than the inequalities $p_i \ge 0$, for $1 \le i \le \ell$. Thus, at least $\ell - t$ of these inequalities must be satisfied with equality.)

Thus, we see that when C is of size t, it is not necessary to let r_0 assume more than t different values. The sets of configurations that we use to bound the performance ratio that can be achieved are very small. For MAX 2-SAT, for example, we use only two configurations.

6 Algorithms from $\mathcal{SKEW}$

In this section we describe the essentially best approximation algorithms for the MAX 2-SAT and MAX DI-CUT problems that can be obtained using rounding procedures from $\mathcal{SKEW}$.

6.1 A MAX 2-SAT algorithm

To specify our algorithms, we only need to specify how the random variable r_0 is chosen. The probability distribution of r_0 used by our MAX 2-SAT algorithm is given in Table 1(a). The distribution of r_0 is very discrete in nature. With a probability of more than $\frac{1}{2}$, namely 0.5166, it assumes the fixed value 1.635. It assumes each of three other values with

prob.	value
0.0430	$N(0,1)$
0.0209	0.145
0.0747	0.345
0.3448	0.755
0.5166	1.635

(a)

prob.	value
0.0714	$N(0,1)$
0.0644	0.170
0.3890	0.640
0.4752	1.520

(b)

Table 1. (a) The probability distribution of r_0 used to obtain a 0.937-approximation algorithm for the MAX 2-SAT problem. (b) The probability distribution of r_0 used to obtain a 0.866-approximation algorithm for the MAX DI-CUT problem.

smaller probabilities. Only with the relatively small probability of 0.0430 it is chosen according to the standard normal distribution. (If we want r_0 to assume only nonnegative values, we take, with this probability, the absolute value of a standard normal variable.) This small probability of assuming a standard normal distribution allows the algorithm to easily achieve good approximation ratios on clauses which contribute very small values to the overall value of the semidefinite program.

6.2 A MAX DI-CUT algorithm

The probability distribution of r_0 used by our MAX DI-CUT algorithm is given in Table 1(b). Again, the distribution of r_0 is very discrete in nature. With a probability of almost $\frac{1}{2}$, namely 0.4752, it assumes the fixed value 1.52. It assumes only two other values with smaller probabilities. Only with the relatively small probability of 0.0714, r_0 is chosen according to the standard normal distribution which again is needed to handle edges which have small values in the semidefinite solution.

Numerical calculations shows that the performance ratio achieved using this rounding procedure is at least 0.866. Again, more details would appear in the full version of the paper.

The approximation algorithm for MAX DI-CUT presented here, as most other MAX DI-CUT approximation algorithms, is also an approximation algorithm for MAX 2-AND.

7 Algorithms from $\mathcal{ROT}+\mathcal{SKEW}$

In this section we describe our best approximation algorithms for the MAX 2-SAT and MAX DI-CUT problems. These algorithms use rounding

θ	$f(\theta)$
0	0.88955971
1.24676999	1.42977923
1.30701991	1.45654677
1.36867009	1.48374672
1.39760143	1.49640918
1.44728317	1.51791721
1.50335282	1.54210593
$\pi/2$	$\pi/2$

Table 2. The piecewise linear rotation function used for the 0.8740 algorithm for the MAX DI-CUT problem.

procedures from the $\mathcal{ROT}$+$\mathcal{SKEW}$ family. These algorithms are essentially the best algorithms that can be obtained using rounding procedures from this family, and even from larger families.

7.1 A MAX 2-SAT algorithm

To specify a rounding procedure from the $\mathcal{ROT}$+$\mathcal{SKEW}$ family, we have to specify a rotation function f and the distribution of r_0. The rounding procedure used by our best MAX 2-SAT algorithm has a very simple representation. The 'random' variable r_0 assumes the value 2 and the rotation function f is linear:

$$f(\theta) \;=\; 0.58831458\,\theta + 0.64667394\;.$$

Numerical calculations show that the performance ratio achieved by this algorithm is at least 0.9401.

The worst performance ratio of about 0.9401 is achieved on two configurations. It is convenient to represent a configuration (v_1, v_2) using a triplet $(\theta_{01}, \theta_{02}, \theta_{12})$, where $\theta_{ij} = \arccos(v_i \cdot v_j)$ is the angle between v_i and v_j, for $0 \le i < j \le 2$. The angles corresponding to the first of the these two worst configurations (v_1, v_2) are roughly $(1.401, 1.401, 2.294)$. The second configuration is then just $(-v_1, -v_2)$, which is roughly represented by the triplet $(1.741, 1.741, 2.294)$.

7.2 A MAX DI-CUT algorithm

The rounding procedure used by our best MAX DI-CUT algorithm has again a fairly simple representation. The 'random' variable r_0 assumes the constant value 2.5. The rotation function f is now a piecewise linear function defined by the points given in Table 2.

Numerical calculations show that the performance ratio achieved by this algorithm is at least 0.8740.

8 Upper bounds on achievable ratios

In this section we explain our claim that the approximation algorithm presented in Section 7.1 is essentially the best MAX 2-SAT algorithm that can be obtained using a rounding procedure from $\mathcal{ROT}+\mathcal{SKEW}$, or even from $\mathcal{THRESH}$. (We also believe that it is essentially the best rounding procedure even from $\mathcal{GEN}^+$, but our evidence in support of this claim is less conclusive at this stage.) We also explain our claim that the approximation algorithm presented in Section 7.2 is essentially the best approximation algorithm that can be obtained using a rounding procedure from $\mathcal{ROT}+\mathcal{SKEW}$, or even from $\mathcal{THRESH}^-$.

We would like to stress that these claims are based on difficult numerical computations, carried out using the optimization toolbox of Matlab. We do not claim, therefore, that they are theorems.

Another point that should be explained is the following. Our analysis of the various rounding procedures is very local in nature and it produces only *lower bounds* on the actual performance ratios. We do not know whether these lower bounds are tight. The local analysis of the MAX CUT algorithm of Goemans and Williamson [GW95] was shown to be tight by Karloff [Kar99] (see also [AS00] and [ASZ02]). It is not known whether the analysis of the MAX 2-SAT and MAX DI-CUT algorithms of Feige and Goemans [FG95] (see also [Zwi00]) is tight. When we say here that a certain rounding procedure is essentially optimal, what we actually mean is that no better *lower bound* on the performance ratio can be obtained, using the local form of analysis, for any other rounding procedure from the family.

8.1 Upper bound for MAX 2-SAT

Let $C = \{(v_1, v_2), (-v_1, -v_2)\}$ be the set of the two worst configurations for the MAX 2-SAT algorithm of Section 7.1. To show that no rounding procedure from $\mathcal{THRESH}$ can be used to obtain a better approximation algorithm for MAX 2-SAT, it is enough to show that no rounding procedure can achieve a better performance ratio even on the set of these two configurations.

A rounding procedure from $\mathcal{THRESH}$ is defined using a threshold function $T : \mathbb{R} \times [-1,1] \to \mathbb{R}$ and a distribution for r_0. To ensure that the literals x_i and $\bar{x}_i$ are assigned opposite values, this function should satisfy $T(x,-y) = -T(x,y)$. As the set C contains only two configurations, it follows from the discussion in Section 5 that there is an optimal rounding procedure for it in which r_0 assumes only two different values. Let us denote these two values by s_1 and s_2. Let p be the probability that r_0 gets the value s_1.

Let $\Theta_1 = (\theta, \theta, \theta_{12})$ and $\Theta_2 = (\pi - \theta, \pi - \theta, \theta_{12})$ be the triplets corresponding to the configurations (v_1, v_2) and $(-v_1, -v_2)$. (We have $\theta \simeq 1.401$ and $\theta_{12} = \arccos(2\cos\theta - 1) \simeq 2.294$.) To determine the behavior of this rounding procedure on the configurations (v_1, v_2) and $(-v_1, -v_2)$ we only need to specify the values of the thresholds $T(s_1, \cos\theta)$ and $T(s_2, \cos\theta)$. (Note that $T(s_i, -\cos\theta) = -T(s_i, -\cos\theta)$, for $i = 1, 2$.) The search for the best rounding procedure from $\mathcal{THRESH}$ is therefore reduced to the search for the optimal setting of the three parameters $T(s_1, \cos\theta)$, $T(s_2, \cos\theta)$ and p. (Note that the values of s_1 and s_2 are not important.) We have numerically searched for the best setting of these three parameters and could not improve on the result obtained using the algorithm of Section 7.1. This algorithm is, therefore, the best algorithm that can be obtained using *any* rounding procedure from $\mathcal{THRESH}$.

A rounding procedure from $\mathcal{GEN}^+$ is characterized by a function $G : \mathbb{R}^2 \times [-1,1] \to [0,1]$ that satisfies $G(x,-y,-z) = 1 - G(x,y,z)$. As before, we may assume that an optimal rounding procedure for C assigns r_0 only two distinct values s_1 and s_2. To determine the behavior of the rounding procedure on the configurations (v_1, v_2) and $(-v_1, -v_2)$ we have to specify p, the probability that r_0 assumes the value s_1, and the two univariate functions $g_1(x) = G(s_1, x, \cos\theta)$ and $g_2(x) = G(s_2, x, \cos\theta)$. We have attempted to improve the performance ratio obtained for C using functions $g_1(x)$ and $g_2(x)$ that are piecewise linear. The attempt failed. This seems to suggest that no improvement can be obtained using rounding procedures from $\mathcal{GEN}^+$. (We plan to elaborate on this point in the full version of the paper.)

8.2 Upper bound for MAX DI-CUT

The upper bound on the performance ratio that can be obtained for the MAX DI-CUT problem using rounding procedures from the family $\mathcal{THRESH}^-$ is obtained using similar ideas. Things are more complicated here, however, as we need to use more configurations. To obtain an upper bound that almost matches the performance of the MAX DI-CUT algorithm of Section 7.2 we need to use a collection of six configurations.

For every $k \geq 1$, Zwick [Zwi00] describes a family C_k of $k+1$ configurations that are parametrized by k angles $\theta_1, \theta_2, \ldots, \theta_k$. For example, for $k = 6$, we have $C_6 = \{\Theta_{17}, \Theta_{26}, \Theta_{35}, \Theta_{44}, \Theta_{43}, \Theta_{52}, \Theta_{61}\}$, where $\Theta_{ij} = (\theta_i, \theta_j, \arccos(\theta_i + \theta_j - 1)$, for $1 \leq i, j \leq 7$, and $\theta_7 = \frac{\pi}{2}$. To obtain our best upper bounds we used the values

$$\begin{gathered} \theta_1 = 1.24677 \ , \ \theta_2 = 1.30703 \ , \ \theta_3 = 1.36867 \ , \\ \theta_4 = 1.39760 \ , \ \theta_5 = 1.44728 \ , \ \theta_6 = 1.50335 \ . \end{gathered}$$

To specify a rounding procedure from $\mathcal{THRESH}^-$ we only need to specify a threshold function $S : [-1, 1] \to \mathbb{R}$. To determine the behavior of this rounding procedure on the configurations of C_6, we only need to specify six numbers, namely $S(\cos\theta_i)$, for $1 \leq i \leq 6$. We have numerically searched for the best setting of these six parameters and could not improve on the result obtained using the algorithm of Section 7.2. This algorithm is, therefore, the best algorithm that can be obtained using *any* rounding procedure from $\mathcal{THRESH}^-$. We hope to determine, in the near future, whether any improvements are possible using rounding procedures from $\mathcal{THRESH}$.

9 Concluding remarks

We presented improved approximation algorithms for the MAX 2-SAT and MAX DI-CUT problems using combinations of pre-rounding rotations and skewed hyperplane rounding. We showed that the performance ratios achieved by these algorithms are essentially the best ratios that can be achieved using this technique. Obtaining further improved approximation ratios would require additional ideas.

Our investigation also sheds some new light on the nature a general family of rounding procedures suggested by Feige and Goemans [FG95].

Acknowledgement

We would like to thank Uri Feige and Michael Langberg for stimulating discussions on skewed hyperplane rounding techniques, and to Shiro Matuura and Tomomi Matsui for sending us a copy of [MM01b].

References

[AS00] N. Alon and B. Sudakov. Bipartite subgraphs and the smallest eigenvalue. *Combinatorics, Probability & Computing*, 9:1–12, 2000.

[ASZ02] N. Alon, B. Sudakov, and U. Zwick. Constructing worst case instances for semidefinite programming based approximation algorithms. *SIAM Journal on Discrete Mathematics*, 15:58–72, 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.

[FL01] U. Feige and M. Langberg. The RPR^2 rounding technique for semidefinite programs. In *Proceedings of the 28th International Colloquium on Automata, Languages and Programming, Crete, Greece*, pages 213–224, 2001.

[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 29th Annual ACM Symposium on Theory of Computing, El Paso, Texas*, pages 1–10, 1997. Full version available as E-CCC Report number TR97-037.

[Kar99] H. Karloff. How good is the Goemans-Williamson MAX CUT algorithm? *SIAM Journal on Computing*, 29:336–350, 1999.

[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.

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

[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.

[Zwi00] U. Zwick. Analyzing the MAX 2-SAT and MAX DI-CUT approximation algorithms of Feige and Goemans. Submitted for publication, 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.