Analyzing the MAX 2-SAT and MAX DI-CUT approximation algorithms of Feige and Goemans

Uri Zwick *

March 2, 2000

Abstract

We present a complete analysis of the MAX 2-SAT and MAX DI-CUT approximation algorithms of Feige and Goemans using various analytical and computational tools. By fine-tuning the rotation functions used we obtain minutely improved performance ratios for these problems. The rotation functions used for getting these improvements are essentially optimal as the performance ratios obtained using them almost completely match upper bounds that we obtain on the performance ratios that can be achieved using any rotation function. We also discuss possibilities of getting improved approximation algorithms for these problems.

1 Introduction

MAX CUT and MAX SAT are two of the most natural combinatorial optimization problems. An instance of MAX CUT is a graph. The goal is to partition the vertices of the graph into two sets such that the number, or the total weight, of the edges that cross the cut formed by this partition is maximized. An instance of MAX SAT is composed of a collection of clauses, each clause being a disjunction of literals. A literal is a variable or its negation. The goal is to assign the variables Boolean values such that the total number, or total weight, of the satisfied clauses is maximized. A clause is satisfied if at least one of its literals is assigned the value 1. These problems are long known to be NP-hard (see [GJ79]).

As polynomial time algorithms that solve all instances of MAX CUT and MAX SAT optimally do not exist, unless P=NP, alternative approaches should be used. One approach is to give up the polynomiality of the algorithms and try to obtain exponential, or perhaps some day sub-exponential, algorithms that are faster than exhaustive search. A second approach is to look for polynomial time algorithms that solve special types of instances optimally. A lot of work was done in these directions. Here, however, we are more interested in a third approach. We would like to construct *polynomial* time algorithms that find *almost* optimal solutions for *all* instances of these problems. Such algorithms are called *approximation algorithms*.

How well can we approximate MAX CUT and MAX SAT? We say that an approximation algorithm for a maximization problem has a *performance ratio* or a *performance guarantee* of α if for every instance of the problem the value of the solution returned by the algorithm is at least α times the value of the optimal

*Department of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel. E-mail: zwick@math.tau.ac.il.

solution. We also say in that case that the algorithm is an α-approximation algorithm for the problem. As we cannot solve the MAX CUT and MAX SAT problems optimally in polynomial time, the next best thing would have been to design, for every $\epsilon > 0$, a $(1-\epsilon)$-approximation algorithm. Such a collection of approximation algorithms is called a *Polynomial Time Approximation Scheme* (PTAS). Unfortunately, we now know that there is no PTAS for MAX CUT, nor for MAX SAT, unless P=NP. This is a very deep result. Its proof relies on very interesting relations between approximation algorithms and Interactive Proofs (IP) and Probabilistically Checkable Proofs (PCP), see Feige et al. [FGL+96], Arora and Safra [AS98] and Arora et al. [ALM+98].

Building on the results cited above, Håstad [Hås97], in another major development, showed that for any $\epsilon > 0$, there is no $(\frac{7}{8}+\epsilon)$-approximation algorithm for MAX SAT, unless P=NP. Håstad's result actually holds also for the MAX $\{3\}$-SAT problem, the restriction of the MAX SAT problem to instances in which all the clauses are of size 3. Trevisan et al. [TSSW96] use Håstad's result to derive strong inapproximablity results for other problems. They show, for example, that for any $\epsilon > 0$, there is no $(\frac{16}{17}+\epsilon)$-approximation algorithm for the MAX CUT problem. They also obtain bounds of $\frac{21}{22}$ and $\frac{12}{13}$, respectively, on the approximation ratios that can be obtained for the problems MAX 2-SAT and MAX DI-CUT. (These problems will be defined shortly.)

There are trivial $\frac{1}{2}$-approximation algorithms for MAX CUT and MAX SAT. A random cut cuts each edge of a graph with probability $\frac{1}{2}$. Similarly, a random assignment satisfies a clause of size k with probability $1-2^{-k}$. This immediately gives *randomized* $\frac{1}{2}$-approximation algorithms for MAX CUT and MAX SAT. It is easy to derandomize these algorithms and obtain deterministic $\frac{1}{2}$-approximation algorithms for these problems. This was done, long ago, by Johnson [Joh74].

Improved approximation algorithms for MAX SAT with a performance ratio of $\frac{3}{4}$ were obtained by Yannakakis [Yan94] and Goemans and Williamson [GW94]. Yannakakis uses network flow techniques while Goemans and Williamson use linear programming. (Further improvements will be mentioned in the sequel.) Linear Programming does not seem to give an improved approximation ratio for the MAX CUT problem.

An improved approximation ratio for the MAX CUT problem was obtained only by using *semidefinite programming* or *eigenvalue minimization* techniques. Semidefinite programs and eigenvalue minimization problems can be solved in polynomial time (see Alizadeh [Ali95] and the references there in). Delorme and Poljak [DP93a],[DP93b],[DP93c] showed that eigenvalue minimization problems can be used to obtain upper bounds on the size of maximum cuts. Goemans and Williamson [GW95] used a semidefinite programming relaxation of the MAX CUT problem, which is equivalent to the eigenvalue minimization problem of Delorme and Poljak, to obtain a vastly improved approximation ratio of about 0.87856 for the MAX CUT problem. The approximation algorithm of Goemans and Williamson uses a simple and elegant *rounding procedure* to round the solution of the semidefinite program into a cut. The solution of the semidefinite programming relaxation of a MAX CUT instance can be translated into n unit vectors $v_1,\ldots,v_n$ in $\mathbb{R}^n$, where n is the number of vertices in the graph. The rounding procedure of Goemans and Williamson [GW95] chooses a *random hyperplane* that passes through the origin. This hyperplane partitions the vectors, and therefore also the vertices of the graphs, into two sets. This partition is the cut returned by the algorithm. As can be seen, the approximation algorithm of Goemans and Williamson is randomized. It can be derandomized however, without affecting the performance ratio, as was shown by Mahajan and Ramesh [MR99].

In their paper, Goemans and Williamson [GW95] also obtain a 0.87856-approximation algorithm for MAX

2-SAT, and an 0.79607-approximation algorithm for the MAX DI-CUT problem. The MAX 2-SAT problem is the restriction of the MAX SAT problem to instances in which all the clauses are of size at most 2. The MAX DI-CUT problem is the analogue of the MAX CUT problem for *directed* graphs. Given a directed graph $G = (V, E)$, the goal is to partition the vertices of the graph into two sets S and $V - S$ such that the total number, or total weight, of the edges emanating from S and going into $V - S$ is maximized.

The performance guarantee of 0.87856 obtained by Goemans and Williamson [GW95] for the MAX CUT problem is still the best known. The MAX 2-SAT and MAX DI-CUT approximation algorithms of Goemans and Williamson were improved, however, by Feige and Goemans [FG95]. One of the important differences between the semidefinite programming relaxation of MAX CUT, and those of MAX 2-SAT and MAX DI-CUT is that in the latter, a vector v_0, which corresponds to, say, the Boolean value 0, plays a special role. In the improved algorithms of Feige and Goemans [FG95], the vectors obtained by solving the semidefinite programming relaxation of the instance are *rotated*, with respect to the special vector v_0, before being rounded using a random hyperplane. The rotations to be used are specified by a *rotation function*.

While, conceptually, the MAX 2-SAT and MAX DI-CUT approximation algorithms of Feige and Goemans [FG95] are almost as simple as the MAX CUT approximation algorithm of Goemans and Williamson [GW95], the analysis of the MAX 2-SAT and MAX DI-CUT approximation algorithms turns out to be much more complicated. Goemans and Williamson [GW95] give a complete analysis of their MAX CUT, MAX 2-SAT and MAX DI-CUT algorithms. They show that the performance ratio of their MAX CUT and MAX 2-SAT algorithms is $\alpha = \min_{0<\theta\leq\pi} \frac{2}{\pi}\frac{\theta}{1-\cos\theta} = 0.87856\ldots$, and that the performance ratio of their MAX DI-CUT algorithm is $\beta = \min_{0\leq\theta\leq\arccos(-1/3)} \frac{2}{\pi}\frac{2\pi-3\theta}{1+3\cos\theta} = 0.79607\ldots$. Things are much more complicated with the MAX 2-SAT and MAX DI-CUT algorithms of Feige and Goemans [FG95]. As these algorithms employ rotation functions, the optimization problems that have to be solved to obtain the performance ratios achieved by these algorithms are more complicated. Also, intertwined with the problem of computing the performance ratios achieved using given rotation functions is the even more difficult problem of finding the rotation functions that give the best, or almost the best, performance.

Feige and Goemans [FG95] give specific rotation functions using which they seem to obtain a 0.931090-approximation algorithm for MAX 2-SAT and a 0.859387-approximation algorithm for MAX DI-CUT. They also claim that these bounds are close to the best bounds that can be achieved by using their algorithms with any rotation function. Their analysis relies, however, non-rigorous numerical computations.

Karloff and Zwick [KZ97] gave an optimal $\frac{7}{8}$-approximation algorithm for the MAX 3-SAT problem, matching the inapproximability bound of Håstad [Hås97]. Their algorithm does not use rotations. Unfortunately, its analysis is again rather involved. (A few more words on that could be found in Section 5.) Halperin and Zwick [HZ99] gave an almost optimal 0.8721-approximation algorithm for the MAX 4-SAT problem. Asano and Williamson [AW00], based on a previous work of Zwick [Zwi99], give a 0.8331-approximation algorithm for MAX SAT. Finally, we mention that an improved approximation algorithm for *almost satisfiable* instances of MAX 2-SAT is given in Zwick [Zwi98b].

In this paper, we give a completely rigorous analysis of the MAX 2-SAT and MAX DI-CUT algorithms of Feige and Goemans [FG95]. As obtaining such a complete analysis using purely analytical means seems too difficult, at least at present, we rely, partially, on *rigorous* global optimization tools. Our analysis confirms the results obtained by Feige and Goemans [FG95]. Furthermore, by small adjustments to the rotation function we obtain slightly improved performance ratios: 0.931091 for MAX 2-SAT, and 0.859643 for MAX DI-CUT. These performance ratios match, almost completely, upper bounds that we obtain on the performance ratios that can be obtained, using the current algorithms, with *any* rotation function. Such upper bounds are not included in Feige and Goemans [FG95], though they were aware of their existence.

All the performance ratios cited above are obtained by solving constrained non-linear optimization problems in three real variables. Although these are continuous problems, and the optimal solutions are not expected to have any 'nice' form (i.e., they are probably not rational, not algebraic, etc.), there are completely rigorous ways of obtaining, in a finite number of steps, rigorous bounds on the optimal solutions of such optimization problems. These techniques take into account the fact that the computer arithmetic used is not exact. One of the main techniques used to obtain such rigorous bounds is *interval arithmetic* (see, e.g., Hansen [Han92]). A very brief summary of the main ideas used to obtain such rigorous bounds is given in Appendix A. To obtain our bounds we used a rigorous global optimization system called GlobSol [Kea96],[CK99]. We discuss the implication of using such a form of analysis in Section 5.

The rest of the paper is organized as follows. In Section 2 we analyze the MAX 2-SAT algorithm. In Section 3 we analyze the MAX DI-CUT algorithm. Each one of these sections is divided into several subsections. The first two subsections describe the approximation algorithm and the rotation functions used by Feige and Goemans [FG95]. The following subsections contain our upper bounds. Finally, the last subsections in these two sections describe our slightly improved rotation functions. In Section 4 we discuss possible improvements and some open problems. We end in Section 5 with some concluding remarks.

2 Approximating MAX 2-SAT

In this section we consider the MAX 2-SAT problem. 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 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.

2.1 The approximation algorithm

Feige and Goemans [FG95] use the semidefinite programming relaxation of the MAX 2-SAT problem given in Figure 1. This is a strengthening of the semidefinite relaxation of MAX 2-SAT given by Goemans and Williamson [GW95] which did not include the *triangle inequalities* $v_i \cdot v_j + v_i \cdot v_k + v_j \cdot v_k \geq -1$, for every $0 \leq i, j, k \leq 2n$. (For our purposes it is sufficient to add these inequalities only for $i = 0$.) This strengthened relaxation was suggested, in a different context, by Feige and Lovász [FL92]. (Feige and Lovász [FL92] did not use the relaxation to obtain an approximation algorithm for MAX 2-SAT.)

Let $v_0, \ldots, v_n$ be an (almost) optimal solution of this program. If these vectors are simply rounded using a random hyperplane, as done by Goemans and Williamson [GW95], the approximation ratio obtained

$$
\begin{array}{llll}
\text{Maximize} & \displaystyle\sum_{i<j} w_{ij} \frac{3 - v_0 \cdot v_i - v_0 \cdot v_j - v_i \cdot v_j}{4} & & \\
\text{subject to} & v_i \cdot v_j + v_i \cdot v_k + v_j \cdot v_k \ \geq \ -1 & , & 0 \leq i,j,k \leq 2n \\
& v_i \cdot v_{n+i} \ = \ -1 & , & 1 \leq i \leq n \\
& v_i \cdot v_i \ = \ 1 & , & 0 \leq i \leq n \\
& v_i \in \mathbb{R}^n & , & 0 \leq i \leq n
\end{array}
$$

Figure 1: The semidefinite programming relaxation of MAX 2-SAT used by Feige and Goemans

would still be $\alpha \simeq 0.87856$, as in the original algorithm of Goemans and Williamson [GW95]. To obtain an improved approximation ratio of about 0.931, Feige and Goemans first *rotate* the vectors $v_0, \ldots, v_n$ into new vectors $v'_0, \ldots, v'_n$, where $v'_0 = v_0$, using a simple process that will be described shortly, and then round these new vectors using a random hyperplane. Variables of the MAX 2-SAT instance whose vectors fall on the same side of the hyperplane as v_0 are assigned the value 0 (false), while variables whose vectors fall on the other side are assigned the value 1 (true).

The vectors $v_0, \ldots, v_n$ are rotated into the vectors $v'_0, \ldots, v'_n$ in the following way. Let $f : [0, \pi] \to [0, \pi]$ be a continuous function satisfying $f(0) = 0$ and $f(\pi - \theta) = \pi - f(\theta)$, for $0 \leq \theta \leq \pi$. (Note that this implies that $f(\pi) = \pi$ and $f(\frac{\pi}{2}) = \frac{\pi}{2}$.) Let $\theta_{ij} = \arccos(v_i \cdot v_j)$ be the angle formed by the vectors v_i and v_j, where $0 \leq i, j \leq n$. The vector v_i is rotated, in the 2-dimensional plane spanned by v_0 and v_i, until it forms an angle of $f(\theta_{0i})$ with v_0. The resulting vector is v'_i. (If $v_i = v_0$, then $v'_i = v_i$.) Let $\theta'_{ij} = \arccos(v'_i \cdot v'_j)$ be the angle formed by v'_i and v'_j. Clearly, $\theta'_{0i} = f(\theta_{0i})$, for every $1 \leq i \leq n$. Feige and Goemans [FG95] show that when $i \neq j$, we have

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

It is not difficult to check (see [FG95]) that the probability that a clause, say $x_1 \vee x_2$, is satisfied by the assignment produced by the above rounding procedure is

$$
\mathit{prob}(v_0, v_1, v_2) \ = \ \frac{\theta'_{01} + \theta'_{02} + \theta'_{12}}{2\pi} \ .
$$

The contribution of this clause to the value of the relaxation is

$$
\mathit{relax}(v_0, v_1, v_2) \ = \ \frac{3 - \cos \theta_{01} - \cos \theta_{02} - \cos \theta_{12}}{4} \ .
$$

As $v_0 = v'_0, v'_1$ and v'_2, and not v_0, v_1 and v_2, are the vectors rounded using a random hyperplane, $\mathit{prob}(v_0, v_1, v_2)$ depends on the angles $\theta'_{01}, \theta'_{02}, \theta'_{12}$ *after* the rotation, while $\mathit{relax}(v_0, v_1, v_2)$ depends on the angles $\theta_{01}, \theta_{02}, \theta_{12}$ between the vectors *before* the rotation. The same formulae hold, of course, for every clause $x_i \vee x_j$, all we have to do is replace the indices 1 and 2 by i and j. As $x_{n+i} = \bar{x}_i$, for $1 \leq i \leq n$, we do not have to consider separately clauses of the form $x_i \vee \bar{x}_j$ or $\bar{x}_i \vee \bar{x}_j$, these are just the clauses $x_i \vee x_{n+j}$ and $x_{n+i} \vee x_{n+j}$. This is also the reason for requiring that $f(\pi - \theta) = \pi - f(\theta)$, for $0 \leq \theta \leq \pi$. This condition ensures that unnegated and negated literals are treated in the same way. (In the rounding process we described, only the vectors $v_1, \ldots, v_n$ are rotated, and then rounded. If we were also to rotate the vectors $v_{n+1}, \ldots, v_{2n}$, then the vectors $v'_{n+1}, \ldots, v'_{2n}$ obtained would just be $-v'_1, \ldots, -v'_n$. This is because $\theta_{0,n+i} = \pi - \theta_{0i}$, which implies that $\theta'_{0,n+i} = f(\pi - \theta_{0i}) = \pi - f(\theta_{0i}) = \pi - \theta'_{0i}$, and therefore

$v'_{n+i} = -v'_i$. Thus, the literals x_i and $\bar{x}_i$ would be assigned different values by this procedure, unless the random hyperplane chosen passes through v_i, an event whose probability is 0.)

The performance ratio of an algorithm that uses a rotation function f is, therefore, at least

$$\beta(f) \;=\; \min_{v_0,v_1,v_2} \frac{prob(v_0,v_1,v_2)}{relax(v_0,v_1,v_2)} \;,$$

where the minimum is over all configurations of three vectors v_0, v_1 and v_2 that satisfy the constraints of the relaxation, and for which $relax(v_0,v_1,v_2) > 0$. By plugging in the expressions for $prob(v_0,v_1,v_2)$ and $relax(v_0,v_1,v_2)$, we get that

$$\beta(f) \;=\; \left[\begin{array}{lc} \displaystyle\min_{\theta_{01},\theta_{02},\theta_{12}} & \displaystyle\frac{2}{\pi}\,\frac{f(\theta_{01})+f(\theta_{02})+g(\theta_{01},\theta_{02},\theta_{12})}{3-\cos\theta_{01}-\cos\theta_{02}-\cos\theta_{12}} \\ \text{subject to} & \cos\theta_{01}+\cos\theta_{02}+\cos\theta_{12} \geq -1 \\ & \cos\theta_{01}-\cos\theta_{02}-\cos\theta_{12} \geq -1 \\ & -\cos\theta_{01}-\cos\theta_{02}+\cos\theta_{12} \geq -1 \\ & -\cos\theta_{01}+\cos\theta_{02}-\cos\theta_{12} \geq -1 \\ & 0 \leq \theta_{01},\theta_{02},\theta_{12} \leq \pi \end{array}\right] \;,$$

where

$$g(\theta_{01},\theta_{02},\theta_{12}) \;=\; \arccos\left(\cos f(\theta_{01})\cos f(\theta_{02}) + \frac{\cos\theta_{12}-\cos\theta_{01}\cos\theta_{02}}{\sin\theta_{01}\sin\theta_{02}}\sin f(\theta_{01})\sin f(\theta_{02})\right) \;.$$

The four inequality constraints in the definition of $\beta(f)$ come from the constraints $v_0\cdot v_i + v_0\cdot v_j + v_i\cdot v_j \geq -1$, where i and j take on the values $1, 2, n+1, n+2$. (Recall that $v_{n+1} = -v_1$ and $v_{n+2} = -v_2$.) We should actually exclude the point $\theta_{01} = \theta_{02} = \theta_{12} = 0$ from the feasibility region as it corresponds to the case $relax(v_0,v_1,v_2) = 0$. We will keep this issue in mind and see that it does not cause us any trouble. As θ_{01}, θ_{02} and θ_{12} are the angles between the vectors v_0, v_1 and v_2, we could also have included constraints like $\theta_{12} \leq \theta_{01} + \theta_{02}$, which correspond to the triangle inequality for distances on the unit sphere. These inequalities are implied, however, by the cosine inequalities. The expression defining $\beta(f)$ is quite complicated, even if the rotation function used is fairly simple.

2.2 The rotations of Feige and Goemans

Feige and Goemans [FG95] use rotation functions from the following family:

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

They claim that a 0.93109 approximation algorithm for MAX 2-SAT can be obtained by taking $\lambda = 0.806765$. A slightly better performance ratio can be obtained by taking $\lambda = 0.80676786784$. We carry out the analysis of the algorithm with this value of λ.

How can one *prove* that the performance ratio of the algorithm proposed by Feige and Goemans is indeed 0.93109? The optimization problem that has to be solved seems to be too complicated to be handled analytically. One can try of course to evaluate the the objective function

$$ratio(\theta_{01},\theta_{02},\theta_{12}) \;=\; \frac{f(\theta_{01})+f(\theta_{02})+g(\theta_{01},\theta_{02},\theta_{12})}{3-\cos\theta_{01}-\cos\theta_{02}-\cos\theta_{12}}$$

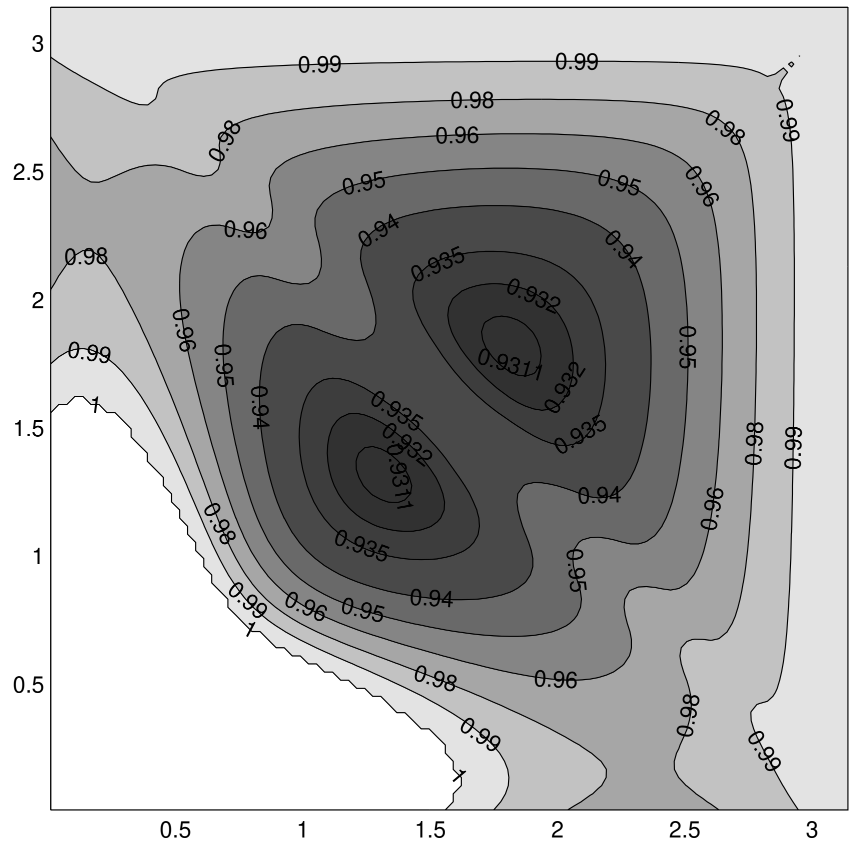

Figure 2: A contour plot of the function $ratio(\theta_{01}, \theta_{02})$ of the MAX 2-SAT problem with $\lambda = 0.80676786784$.

at points of the form $\theta_{01} = \frac{i\pi}{m}$, $\theta_{02} = \frac{j\pi}{m}$ and $\theta_{12} = \frac{k\pi}{m}$, for every $0 \le i, j, k \le m$, for a relatively large value of m, and find the smallest value obtained at points of this form that satisfy all the constraints. This process alone, however, does not yield rigorous lower bounds on $\beta(f)$ as smaller values of $ratio(\theta_{01}, \theta_{02}, \theta_{12})$, possibly much smaller, may be obtained in between these points. One can also try to solve the optimization problem using numerical optimization techniques. These methods converge, usually, to a *local* minimum of the function. There is no guarantee that the *global* minimum will be found.

To get an idea of how the function $ratio(\theta_{01}, \theta_{02}, \theta_{12})$ behaves, one can try to plot it in various ways. The values of $ratio(\theta_{01}, \theta_{02}, \theta_{12})$ at points of the form $\theta_{01} = \frac{i\pi}{m}$, $\theta_{02} = \frac{j\pi}{m}$ and $\theta_{12} = \frac{k\pi}{m}$ can be used, for example, to draw a contour plot of the function $ratio(\theta_{01}, \theta_{02}) = \min_{\theta_{12}} ratio(\theta_{01}, \theta_{02}, \theta_{12})$, were the minimum is over the admissible values of θ_{12}. Such a plot is given in Figure 2.

Feige and Goemans [FG95] used such numerical techniques to find out that the minimum of the function $ratio(\theta_{01}, \theta_{02}, \theta_{12})$, subject to the relevant constraints, seems to be 0.93109, and it seems to be attained near the two points $(1.32238, 1.32238.2.10396)$ and $(1.81921, 1.81921, 2.10396)$. This is also what the contour plot of Figure 2 seems to indicate. (The fact that the global minimum seems to be attained at two different points is not a coincidence. Note that $1.32238 + 1.81921 \approx \pi$. The parameter λ is chosen so that these two local minima would have an almost equal value.) These experiments indicate that it is very likely that the performance ratio of the algorithm is roughly 0.93109. They do not provide, however, a rigorous proof of this fact.

The estimate on $\beta(f)$ obtained by computing $ratio(\theta_{01},\theta_{02},\theta_{12})$ at points of the form $\theta_{01}=\frac{i\pi}{m}$, $\theta_{02}=\frac{j\pi}{m}$ and $\theta_{12}=\frac{k\pi}{m}$ can be turned into a rigorous bound on $\beta(f)$ if bounds are obtained on the partial derivatives of $ratio(\theta_{01},\theta_{02},\theta_{12})$, and if round-off errors are taken into account. Systems for Rigorous Global Optimization, such as GlobSol [Kea96],[CK99] and Numerica [VMD97],[Van98] use these, and much more sophisticated techniques, to obtain rigorous bounds on solutions of optimization problems. Some of the basic techniques used by such systems are described in Appendix A. More information can also be found in Hansen [Han92].

With the help of GlobSol, we obtained very sharp bounds on the global minimum of $ratio(\theta_{01},\theta_{02},\theta_{12})$ when $\lambda=0.80676786784$. Obtaining such bounds on $\beta(f_\lambda)$ with the help of GlobSol is not a straightforward task as the function $ratio(\theta_{01},\theta_{02},\theta_{12})$ to be minimized is not defined for every $0\le\theta_{01},\theta_{02},\theta_{12}\le\pi$. (The points in which $ratio(\theta_{01},\theta_{02},\theta_{12})$ is not defined do not satisfy all the required inequalities.) The exact way in which this was done is described in Appendix B. It turns out that the global minimum $\beta=\beta(f_\lambda)$ and the point $(\theta_{01},\theta_{02},\theta_{12})$ in which it is attained fall into one of the following two boxes:

$$\begin{array}{lcl}
\theta_{01}\in[1.819214250,1.819214253] & & \theta_{01}\in[1.315976123,1.315976126]\\
\theta_{02}\in[1.819214250,1.819214253] & \text{or} & \theta_{02}\in[1.315976123,1.315976126]\\
\theta_{12}\in[2.103957720,2.103957723] & & \theta_{12}\in[2.089617928,2.089617932]\\
\\
\beta\in[0.9310900681,0.9310900686] & & \beta\in[0.9310900680,0.9310900687]
\end{array}$$

This provides a *rigorous* proof that the performance ratio of the MAX 2-SAT algorithm of Feige and Goemans, with $\lambda=0.80676786784$, is at least 0.931090068, assuming the correctness of GlobSol. As expected, Feige and Goemans were right. Further comments on the nature and on the acceptability of such proofs appear in Section 5.

2.3 An upper bound

How close is the rotation function used by Feige and Goemans [FG95] to being optimal? In an attempt to answer this question we present in this section an *upper bound* $\bar\beta$ on the performance ratio that can be obtained for MAX 2-SAT using *any* rotation function.

Let θ be a fixed angle. Consider the two configurations

$$\begin{aligned}
\Theta_1 &= (\ \theta\ ,\ \ \theta\ \ ,\arccos(-1-2\cos\theta))\ ,\\
\Theta_2 &= (\pi-\theta\,,\,\pi-\theta\,,\arccos(-1-2\cos\theta))\ .
\end{aligned}$$

It is not difficult to check that both these configurations are feasible when $\frac{\pi}{2}\le\theta\le\pi$. The values of $ratio(\Theta_1)$ and $ratio(\Theta_2)$ depend only on θ and $f(\theta)$. (Recall that $f(\pi-\theta)=\pi-f(\theta)$.) Let $\psi=f(\theta)$. To make this dependency more aparent, we write these two ratios, from now on, as $ratio(\Theta_1,\psi)$ and $ratio(\Theta_2,\psi)$. For any fixed θ, the expression

$$\bar\beta(\theta)\ =\ \max_{0\le\psi\le\pi}\ \min\{ratio(\Theta_1,\psi),ratio(\Theta_2,\psi)\}$$

is clearly an upper bound on $\beta(f)$, for *any* rotation function f. Thus

$$\bar\beta\ =\ \min_{\frac{\pi}{2}\le\theta\le\pi}\bar\beta(\theta)\ =\ \min_{\frac{\pi}{2}\le\theta\le\pi}\ \max_{0\le\psi\le\pi}\ \min\{ratio(\Theta_1,\psi),ratio(\Theta_2,\psi)\}$$

is also such an upper bound. In other words, $\bar{\beta}$ is an upper bound on the performance ratio that can be obtained by the current algorithm using *any* rotation function, if the current mode of analysis is used. (It should be remembered that $\beta(f)$ is a *lower bound* on the performance ratio achieved when using the rotation function f. This lower bound may not be tight. More on this in Section 4.)

It is not difficult to see that for a given θ, the value of ψ at which $\bar{\beta}(\theta)$ is achieved is the one for which $ratio(\Theta_1,\psi) = ratio(\Theta_2,\psi)$. There is a unique such ψ as it is not difficult to check that if $\theta \ge \frac{\pi}{2}$, then $ratio(\Theta_1,\psi)$ is increasing in ψ while $ratio(\Theta_2,\psi)$ is decreasing in ψ. Thus

$$\bar{\beta} \;=\; \left[\begin{array}{ll} \min\limits_{\theta,\psi} & ratio(\Theta_1,\psi) \\ \text{subject to} & ratio(\Theta_1,\psi) = ratio(\Theta_2,\psi) \\ & \frac{\pi}{2} \le \theta \le \pi \;,\; 0 \le \psi \le \pi \end{array}\right] .$$

Using the formulae given in Section 2.1, after a few simplifications, we get that

$$ratio(\Theta_1,\psi) \;=\; \frac{1}{2\pi}\left(2\psi + \arccos\left(\cos^2\psi - \left(\frac{1+\cos\theta}{\sin\theta}\right)^2 \sin^2\psi\right)\right) \;=\; \frac{1}{2\pi}\left(2\psi + \arccos\left(1 - \frac{\sin^2\psi}{\sin^2\frac{\theta}{2}}\right)\right) \;,$$

$$ratio(\Theta_2,\psi) \;=\; \frac{1}{2\pi}\,\frac{2(\pi-\psi) + \arccos\left(1 - \frac{\sin^2\psi}{\sin^2\frac{\theta}{2}}\right)}{1+\cos\theta} \;.$$

Using GlobSol, we get that the optimal θ^* and ψ^*, and $\bar{\beta}$ satisfy

$$\begin{aligned} \theta^* &\in [1.82046072, 1.82046074] \;, \\ \psi^* &\in 1.93216265, 1.93216267] \;, \\ \bar{\beta} &\in [0.9310914878, 0.9310914899] \;. \end{aligned}$$

Thus, $\beta(f) \le 0.9310915$, for *any* rotation function f.

2.4 More on the upper bound

We can gain some more insight on the lower bound just obtained by solving the optimization problem defining $\bar{\beta}$ using a *Lagrange multiplier* z (see, e.g., Chapter 3 of Bertsekas [Ber99]). Let

$$L(\theta,\psi,z) \;=\; ratio(\Theta_1,\psi) - z\cdot(ratio(\Theta_1,\psi) - ratio(\Theta_2,\psi)) \;=\; (1-z)\cdot ratio(\Theta_1,\psi) + z\cdot ratio(\Theta_2,\psi)) \;.$$

Let θ^* and ψ^* be the angles at which the value of $\bar{\beta}$ is attained. Then, there exists z^* such that θ^*, ψ^* and z^* satisfy the following three equations:

$$\frac{\partial L}{\partial\theta}(\theta,\psi,z) \;=\; \frac{\partial L}{\partial\psi}(\theta,\psi,z) \;=\; \frac{\partial L}{\partial z}(\theta,\psi,z) \;=\; 0 \;.$$

(To show the existence of such a z^* we actually have to show that (θ^*,ψ^*) is a *regular* point. In our case, we have to show that $\frac{\partial}{\partial\theta}\left[ratio(\Theta_1,\psi) - ratio(\Theta_2,\psi)\right]|_{\theta^*,\psi^*} \neq 0$ or that $\frac{\partial}{\partial\psi}\left[ratio(\Theta_1,\psi) - ratio(\Theta_2,\psi)\right]|_{\theta^*,\psi^*} \neq 0$. This condition can be verified using the very tight bounds we have on θ^* and ψ^*.)

By a lenghty symbolic manipulation of these equations, done with the help of Mathematica [Wol99], we can eliminate z^* from these equations and get that θ^* and ψ^* should satisfy the following two equations:

$$\frac{\sin^2\psi}{\sin^2\frac{\theta}{2}} + \cos\left(\frac{2(\pi-2\psi)}{\cos\theta} - 2\psi\right) \;=\; 1 \;,$$

$$\frac{\cos\theta}{2\sin^4\frac{\theta}{2}}\sin^2\psi + (\frac{\pi}{2} - \psi)\left[2\sin\left(2\psi - \frac{2(\pi-2\psi)}{\cos\theta}\right) - \frac{\sin 2\psi}{\sin^2\frac{\theta}{2}}\right] \;=\; 0 \;.$$

A symbolic expression expressing z^* as a function of θ^* and ψ^* can also be obtained. As it is rather complicated, we do not bring it here. The upper bound $\bar{\beta}$ can be expressed very succinctly in terms of θ^* and ψ^*

$$\bar{\beta} \;=\; \frac{1 - \frac{2\psi^*}{\pi}}{\cos\theta^*} \;.$$

Solving the equations defining θ^*, ψ^* and z^* *numerically*, using the `FindRoot` function of Mathematica (see [Wol99]), with a working precision of 1000 decimal digits, we get that

$$\begin{aligned}
\theta^* &= 1.82046073248821624015\ldots \\
\psi^* &= 1.93216265987513186661\ldots \\
z^* &= 0.16692912073745081135\ldots \\
\bar{\beta} &= 0.93109148885525009015\ldots\;.
\end{aligned}$$

We should point out here that although all digits given above are likely to be correct, and we can compute many more, if we so desire, Mathematica does *not* give, as far as we know, a guarantee that this is really the case. Rigorous global optimization tools do not give yet such accuracy but their results are guaranteed to be correct (unless their implemetation is incorrect).

If a rotation function f is to yield an approximation algorithm with a performance ratio of $\bar{\beta}$, then it must satisfy $f(\theta^*) = \psi^*$. As $\theta = \theta^*$ is supposed to be the minimum of the function $ratio(\Theta_1) = ratio(\theta, \theta, \arccos(-1 - 2\cos\theta))$, we can easily get, by symbolically differentiating $ratio(\Theta_1)$ with respect to θ and equating to 0, an expression giving $f'(\theta^*)$ in terms of θ^* and ψ^*. As this expression is quite complicated we shall not present it here. By evaluating this expression numerically, using the numerically computed values of θ^* and ψ^* we get that $f'(\theta^*) = 1.41720914700094622305\ldots$. We shall use this fact in the next subsection.

The Lagrange multiplier z^* used to obtain the upper bound $\bar{\beta}$ has a very interesting interpretation. So far, we were assuming that the approximation algorithm uses a *fixed* rotation function. A more general algorithm would first solve the semidefinite programming relaxation and then, based on the vectors obtained, would choose the rotation function to use. Can we get in this way an improved approximation algorithm for MAX 2-SAT? It follows from the definition of z^* that

$$\bar{\beta} \;=\; \bar{\beta}(\theta^*) \;=\; \max_{\psi} \quad (1 - z^*) \cdot ratio(\Theta_1, \psi) + z^* \cdot ratio(\Theta_2, \psi) \;.$$

Thus, if for every clause $x_i \vee x_j$ with a non-zero contributation to the value of the relaxation, the vectors v_0, v_i, v_j are either in configuration Θ_1 or in configuration Θ_2, for $\theta = \theta^*$, and if the contribution of clauses that are in configuration Θ_1 is exactly $1 - z^*$ of the value of the relaxation, then no rotation function would give a ratio greater than $\bar{\beta}$ in this setting. It is not clear, however, whether there exists an instance that has optimal vectors $v_0, v_1, \ldots, v_n$ that satisfy all these conditions. We mention this in Section 4 as one of the open questions.

2.5 Optimal rotations

The performance ratio $0.9310900680\ldots$, obtained by the algorithm of Feige and Goemans described in Section 2.2, is extremely close to the upper bound $\bar{\beta} = 0.9310914888\ldots$ obtained in Sections 2.3 and 2.4 on the performance ratio that can be obtained using any rotation function. Can it be that by slightly changing λ we could obtain an algorithm for MAX 2-SAT with a performance ratio of $\bar{\beta}$?

The answer is no. The choice $\lambda = 0.80676786784$ is essentially the optinal one. (The optimal choice is $\lambda = 0.80676786784081143 06\ldots$)

We can show that no choice of λ would yield a performance ratio of $\bar{\beta}$ as follows. If we were to get a performance ratio of $\bar{\beta}$ then, as explained towards the end of Section 2.4, we shold have $f_\lambda(\theta^*) = \psi^*$, or in other words

$$\lambda \;=\; \frac{\psi^* - \theta^*}{\frac{\pi}{2}(1-\cos\theta^*) - \theta^*} \;=\; 0.80682645076384875167\ldots\;.$$

However, with this choice of λ we would have $f'_\lambda(\theta^*) = 1.42123957685877986893\ldots$, not as required.

Using a rotation function from a family of rotation functions determined by two parameters, such as the family

$$f_{\lambda_1,\lambda_2}(\theta) \;=\; \theta + \lambda_1\left(\frac{\pi}{2}(1-\cos\theta) - \theta\right) + \lambda_2\left(\frac{\pi}{2}(1-\cos\theta) - \theta\right)^3 \;,$$

we can make sure that both $f(\theta^*)$ and $f'(\theta^*)$ get the right values. By setting

$$\begin{aligned}\lambda_1 &= \;\;\,0.8106863176769092566\;,\\ \lambda_2 &= -0.2013776423882601184\;,\end{aligned}$$

we indeed get a performance ratio of 0.9310914888. This was rigorously verified using GlobSol, as was done in Section 2.2. We believe that by choosing λ_1 and λ_2 such that $f(\theta^*)$ and $f'(\theta^*)$ get *exactly* the right values, we obtain an approximation ratio of exactly $\bar{\beta}$. Although it is possible to *rigorously* prove this claim using slight extensions of the techniques used in this paper (we do not elaborate on this here), we did not do it. We feel that 0.9310914888 is close enough to the upper bound.

3 Approximating MAX DI-CUT

In this section we consider the MAX DI-CUT problem. 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 $1 \le i \le n$.

A problem which is very closely related to the MAX DI-CUT problem is the MAX 2-AND problem. An instance of MAX 2-AND in the Boolean variables $x_1, x_2, \ldots, x_n$ 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 them form $z_1 \wedge z_2$ where each z_j is either a variable x_k or its negation $\bar{x}_k$. Each such z_j is called a *literal.* (Note the similarity with MAX 2-SAT. The difference is that here a clause is the *conjunction*, and not the *disjunction*, of literals.) 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 *all* the literals appearing in it are 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 not difficult to see that MAX DI-CUT is a sub-problem of MAX 2-AND. An instance of MAX DI-CUT can be easily translated into an instance of MAX 2-AND in the following way. For each edge $i \to j$ with

$$\begin{array}{|lrcll|}\hline \text{Maximize} & \multicolumn{4}{l|}{\displaystyle\sum_{i,j} w_{ij} \frac{1 + v_0 \cdot v_i - v_0 \cdot v_j - v_i \cdot v_j}{4}} \\ \text{subject to} & v_i \cdot v_j + v_i \cdot v_k + v_j \cdot v_k & \geq & -1 & , \quad 0 \leq i,j,k \leq n \\ & v_i \cdot v_i & = & 1 & , \quad 0 \leq i \leq n \\ & v_i \in \mathbb{R}^n & & & , \quad 0 \leq i \leq n \\ \hline \end{array}$$

Figure 3: The semidefinite programming relaxaion of the MAX DI-CUT used by Feige and Goemans.

weight w in the input graph G, add a clause $\bar{x}_i \wedge x_j$ with weight w to the MAX 2-AND instance. There is then an obvious value preserving correspondence between directed cuts in G and 0-1 assignments to the variables $x_1, \ldots, x_n$, so the MAX DI-CUT instance and the resulting MAX 2-AND instance are completely equivalent.

Although the MAX 2-AND problem is, in principle, a more general problem than the MAX DI-CUT problem, all the approximation algorithms given in this paper work also for the MAX 2-AND problem.

An even more general problem is the MAX 2-CSP problem. An instance of the MAX 2-CSP problem is again 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 now of the form $f(z_1, z_2)$ where f is an *arbitrary* Boolean function and z_1 and z_2 are literals. There is, however, a simple value preserving reduction from MAX 2-CSP to MAX 2-AND (see, e.g., [Zwi98a]) so all the approximation algorithms given in this section can also be made to work for the MAX 2-CSP problem.

3.1 The approximation algorithm

Feige and Goemans [FG95] use the semidefinite programming relaxaion of the MAX DI-CUT problem given in Figure 3. Their approximation algorithm finds vectors $v_0, v_1, \ldots, v_n$ that (almost) maximize this program and then rotates them into vectors $v_0 = v'_0, v'_1, \ldots, v'_n$ using a rotation function f, as described in Section 2.1. (Different rotation function are used, as we shall see, in the MAX 2-SAT and MAX DI-CUT algorithms.) A directed cut is obtained by choosing a random hyperplane and putting on the 'left' side of the cut all the vertices that fall on the same side of the hyperplane as v_0. It is not difficult to check (see [GW95],[FG95]) that the probability that the edge $1 \to 2$ belongs to the directed cut produced is

$$\mathit{prob}(v_0, v_1, v_2) \;=\; \frac{-\theta'_{01} + \theta'_{02} + \theta'_{12}}{2\pi} \;.$$

The contribution of this edge to the value of the relaxation is

$$\mathit{relax}(v_0, v_1, v_2) \;=\; \frac{1 + \cos\theta_{01} - \cos\theta_{02} - \cos\theta_{12}}{4} \;.$$

The performance ratio of an algorithm that uses a rotation function f is, therefore, at least

$$\gamma(f) \;=\; \left[\begin{array}{ll} \displaystyle\min_{\theta_{01},\theta_{02},\theta_{12}} & \displaystyle\frac{2}{\pi}\,\frac{-f(\theta_{01})+f(\theta_{02})+g(\theta_{01},\theta_{02},\theta_{12})}{1+\cos\theta_{01}-\cos\theta_{02}-\cos\theta_{12}} \\ \text{subject to} & \cos\theta_{01}+\cos\theta_{02}+\cos\theta_{12}\geq -1 \\ & \cos\theta_{01}-\cos\theta_{02}-\cos\theta_{12} > -1 \\ & -\cos\theta_{01}-\cos\theta_{02}+\cos\theta_{12}\geq -1 \\ & -\cos\theta_{01}+\cos\theta_{02}-\cos\theta_{12}\geq -1 \\ & 0\leq \theta_{01},\theta_{02},\theta_{12}\leq \pi \end{array}\right] ,$$

where $g(\theta_{01},\theta_{02},\theta_{12})$ is as defined in Section 2.1. (We require strict inequality in the second inequality constraint to ensure that $\mathit{relax}(v_0,v_1,v_2)>0$ and to avoid division by 0.)

3.2 The rotations of Feige and Goemans

Feige and Goemans [FG95] suggest two rotation functions to be used with their algorithm. The first is a function from the family

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

with $\lambda = 0.5$. They claim that using this function, an approximation ratio of 0.857 is obtained. We let

$$\mathit{ratio}(\theta_{01},\theta_{02},\theta_{12}) \;=\; \frac{2}{\pi}\,\frac{-f(\theta_{01})+f(\theta_{02})+g(\theta_{01},\theta_{02},\theta_{12})}{1+\cos\theta_{01}-\cos\theta_{02}-\cos\theta_{12}} \;.$$

A contour plot of the function $\mathit{ratio}(\theta_{01},\theta_{02}) = \min_{\theta_{12}} \mathit{ratio}(\theta_{01},\theta_{02})$, were the minimum is over the admissible values of θ_{12}, is given in Figure 4. We verified, using GlobSol, that the performance ratio obtained using this rotation function is at least 0.85702758.

Feige and Goemans [FG95] claim that an improved approximation ratio of 0.859 can be obtained using a more complicated rotation function. This function is obtained by using a value of λ that depends on θ. More specifically, $\lambda = \lambda(\theta)$ is the piecewise linear function that passes through the points $\lambda(0) = \lambda(1.1) = 0.555$, $\lambda(1.2) = 0.53$, $\lambda(1.3) = 0.475$, $\lambda(1.4) = 0.425$, and which satisfies $\lambda(\pi-\theta) = \lambda(\theta)$, for $0\leq\theta\leq\pi$. (This function does not appear explicitly in [FG95]. It was communicated to us by Michel Goemans.) We verified, using GlobSol, that the performance ratio obtained using this rotation function is at least 0.859387.

3.3 An upper bound

In this section we derive an upper bound on the approximation ratio that can be obtained for MAX DI-CUT using the algorithm of Feige and Goemans with any rotation function. The technique is similar to the technique used in Section 2.3, tough several new ideas are needed to overcome some technical complications.

Let θ be a fixed angle. Consider the two configurations

$$\begin{aligned} \Theta_1 &= (\theta\,,\,\tfrac{\pi}{2}\,,\ \arccos(\cos\theta-1)\ )\ . \\ \Theta_2 &= (\theta\,,\,\theta\,,\arccos(2\cos\theta-1))\ , \end{aligned}$$

It is not difficult to check that both these configurations are feasible when $0\leq\theta\leq\frac{\pi}{2}$. The values of $\mathit{ratio}(\Theta_1)$ and $\mathit{ratio}(\Theta_2)$ depend only on θ and $f(\theta)$. (Recall that $f(\pi-\theta) = \pi - f(\theta)$ and that therefore $f(\frac{\pi}{2}) = \frac{\pi}{2}$.) Let $\psi = f(\theta)$. To make this dependency more aparent, we write these two ratios, from now on,

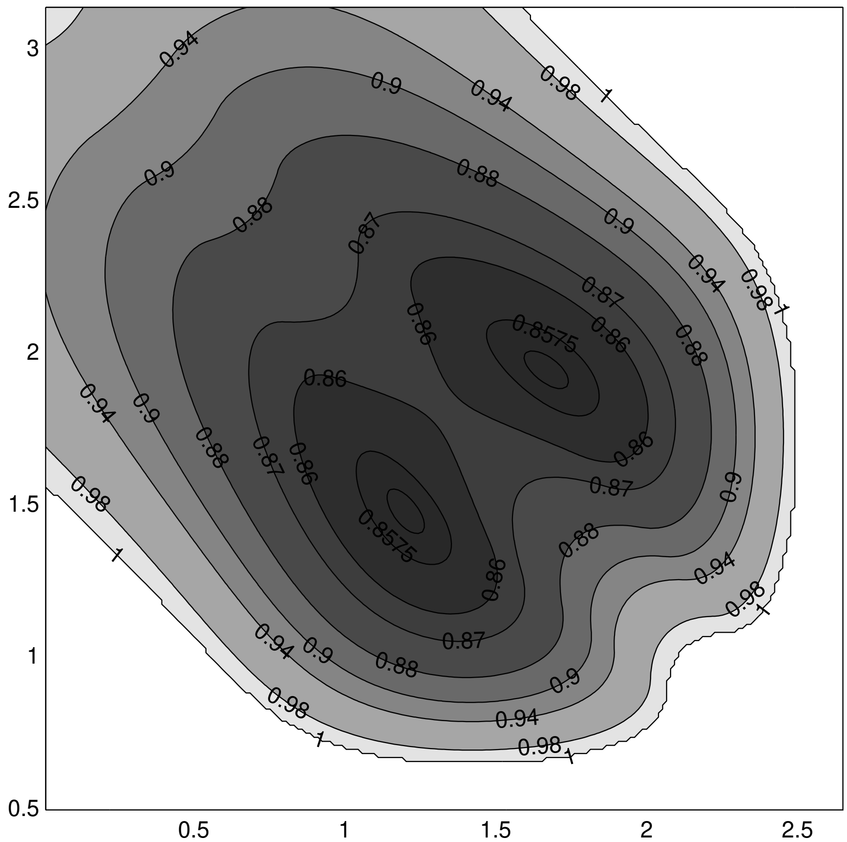

Figure 4: A contour plot of the function $\mathit{ratio}(\theta_{01}, \theta_{02})$ of the MAX DI-CUT problem with $\lambda = 0.5$.

as $\mathit{ratio}(\Theta_1, \psi)$ and $\mathit{ratio}(\Theta_2, \psi)$. As in Section 2.3, the following is an upper bound on the performance ratio that can be obtained using *any* rotation function:

$$\bar{\gamma} \;=\; \min_{0 \le \theta \le \frac{\pi}{2}} \; \max_{0 \le \psi \le \pi} \; \min\{\mathit{ratio}(\Theta_1, \psi), \mathit{ratio}(\Theta_1, \psi)\} \; .$$

Using the formulae given in Section 3.1, and using the trigonometric identity $\frac{1-\cos\theta}{\sin\theta} = \tan\frac{\theta}{2}$, we get that

$$\begin{aligned}
\mathit{ratio}(\Theta_1, \psi) \;&=\; \frac{1}{\pi}\left[\frac{\pi}{2} - \psi + \arccos\left(-\tan\tfrac{\theta}{2}\sin\psi\right)\right] \;, \\
\mathit{ratio}(\Theta_2, \psi) \;&=\; \frac{1}{\pi}\frac{\arccos\left(\cos^2\psi - \tan^2\frac{\theta}{2}\sin^2\psi\right)}{1-\cos\theta} \;=\; \frac{1}{\pi}\frac{\arccos\left(1 - \frac{\sin^2\psi}{\cos^2\frac{\theta}{2}}\right)}{1-\cos\theta} \;.
\end{aligned}$$

The task of obtaining rigorous bounds on $\bar{\gamma}$ turns out to be more difficult than the corresponding task, carried out in Section 2.3, of obtaining rigorous bounds on $\bar{\beta}$. The difficulty lies in the fact that the functions $\mathit{ratio}(\Theta_1, \psi)$ and $\mathit{ratio}(\Theta_2, \psi)$ this time are not monotone in ψ. Thus, for a given θ, there may be more than one ψ for which $\mathit{ratio}(\Theta_1, \psi) = \mathit{ratio}(\Theta_2, \psi)$. As a consequence, we cannot simply replace the maximization over ψ by the constraint $\mathit{ratio}(\Theta_1, \psi) = \mathit{ratio}(\Theta_2, \psi)$. We overcome this difficulty in the following way. We intorduce again the Lagrangian function

$$L(\theta, \psi, z) \;=\; (1-z)\cdot\mathit{ratio}(\Theta_1, \psi) + z\cdot\mathit{ratio}(\Theta_2, \psi) \; .$$

It is not difficult to see that

$$\bar{\gamma} \;=\; \min_{0\le\theta\le\frac{\pi}{2}} \; \max_{0\le\psi\le\pi} \; \min_{0\le z\le 1} \; L(\theta,\psi,z) \;.$$

We also let

$$\bar{\gamma}' \;=\; \min_{0\le\theta\le\frac{\pi}{2}} \; \min_{0\le z\le 1} \; \max_{0\le\psi\le\pi} \; L(\theta,\psi,z) \;.$$

It is not difficult to check that $\bar{\gamma} \le \bar{\gamma}'$. Let θ^*, ψ^* and z^* be such that $\bar{\gamma}' = L(\theta^*,\psi^*,z^*)$. We believe that $\bar{\gamma} = \bar{\gamma}' = L(\theta^*,\psi^*,z^*)$, and that θ^*, ψ^* and z^* satisfy the three equations

$$\frac{\partial L}{\partial \theta}(\theta,\psi,z) \;=\; \frac{\partial L}{\partial \psi}(\theta,\psi,z) \;=\; \frac{\partial L}{\partial z}(\theta,\psi,z) \;=\; 0 \;.$$

We have, however, no rigorous proof of this claim, so we do *not* rely on this claim in the sequel. Our strategy is the following. We solve the three equations given above, numerically, and obtain an approximate root θ_0, ψ_0 and z_0. Using θ_0 and z_0 we then obtain a *rigorous* upper bound on $\bar{\gamma}'$, which is also an upper bound on $\bar{\gamma}$. (The rigorousity of the upper bound obtained depends only on the rigorousity of the second step. The first step only 'guesses' the values θ_0 and z_0.) Using Mathematica we obtained the following approximate root of the three equations:

$$\begin{aligned} \theta_0 &= 1.21567288908303354835 \;, \\ \psi_0 &= 1.11092483031326593184 \;, \\ z_0 &= 0.24707933499676612120 \;. \end{aligned}$$

For these values of θ_0, ψ_0 and z_0 we get $L(\theta_0,\psi_0,z_0) = 0.86064152882557049833\ldots$. This, however, is still a non-rigorous estimate of $\bar{\gamma}$. Let $\bar{\gamma}'' = \max_{0\le\psi\le\pi} L(\theta_0,\psi,z_0)$. Let ψ_1 be the value of ψ attaining this maximum. Clearly $\bar{\gamma} \le \bar{\gamma}' \le \bar{\gamma}''$. Using GlobSol we find that

$$\begin{aligned} \psi_1 &\in [1.1109248301, 1.1109248305] \\ \bar{\gamma}'' &\in [0.8606415286, 0.8606415291] \;. \end{aligned}$$

We thus get the *rigorous* bound $\bar{\gamma} \le \bar{\gamma}'' \le 0.8606415291$. In an instance that matches this upper bound, about $1 - z_0 \simeq 0.7529$ of the edges should be in the configuration Θ_1 and about $z_0 \simeq 0.2471$ of them should be in the configuration Θ_2, where $\theta \simeq 1.21567$. (Again, it is not clear that such an instance exists. Recall that $\gamma(f)$ is just a *lower bound* on the approximation ratio obtained using the rotation function f.)

To obtain the upper bound we assumed that $f(\pi - \theta) = \pi - f(\theta)$ for every $0 \le \theta \le \pi$, so that $f(\frac{\pi}{2}) = \frac{\pi}{2}$. When working with problems like MAX 2-SAT and MAX 2-AND, it is very natural to impose these conditions, as they ensure that negated and unnegated literals are dealt with in the same way. Can we get a better approximation algorithm for MAX DI-CUT by using a rotation function that does not satisfy these conditions? It seems that the answer is no. We can show, for example, that the upper bounds obtained in this and in the next section hold also without these assumptions, although their derivations are slightly more complicated. We omit the details.

3.4 A sequence of upper bounds

In the previous section we obtained an upper bound on the performance ratio that can be obtained for MAX DI-CUT using any rotation function by considering the value of the rotation function $f(\theta)$ at one critical angle θ^*. A similar upper bound obtained in Section 2.3 for the MAX 2-SAT problem turned out to be tight. Things are different, again, with the MAX DI-CUT problem. It turns out that we can get better upper bounds by considering the values of the rotation function $f(\theta)$ at several critical angles $\theta_1^*, \theta_2^*, \ldots, \theta_k^*$. Let $\theta_1, \theta_2, \ldots, \theta_k$ be k angles such that $0 < \theta_1 < \theta_2 < \cdots < \theta_k < \frac{\pi}{2}$, and let $\theta_{k+1} = \frac{\pi}{2}$. Let $\psi_i = f(\theta_i)$, for $1 \le i \le k$. Note that $\psi_{k+1} = f(\theta_{k+1}) = \frac{\pi}{2}$. We let $\Theta = (\theta_1, \ldots, \theta_k)$ and $\Psi = (\psi_1, \ldots, \psi_k)$. For every $1 \le i, j \le k+1$, we let Θ_{ij} be the following configuration:

$$\Theta_{ij} \;=\; (\theta_i, \theta_j, \arccos(\cos\theta_i + \cos\theta_j - 1)) \;.$$

It is not difficult to check that Θ_{ij} is a feasible configuration and that

$$ratio(\Theta_{ij}, \Psi) \;=\; \frac{1}{\pi} \frac{-\psi_i + \psi_j + \arccos\left(\cos\psi_i \cos\psi_j - \tan\frac{\theta_i}{2}\tan\frac{\theta_j}{2}\sin\psi_i \sin\psi_j\right)}{1 - \cos\theta_j} \;.$$

For a given k, we consider the following set of configurations

$$C_k \;=\; \{\, \Theta_{i,k+2-i} \mid 1 \le i \le \lfloor k/2 \rfloor + 1 \,\} \;\cup\; \{\, \Theta_{k+1-i,i} \mid 1 \le i \le \lceil k/2 \rceil \,\} \;.$$

For example, $C_3 = \{\Theta_{14}, \Theta_{23}, \Theta_{22}, \Theta_{31}\}$ and $C_4 = \{\Theta_{15}, \Theta_{24}, \Theta_{33}, \Theta_{32}, \Theta_{41}\}$. The constant $\bar{\gamma}_k$, defined as

$$\bar{\gamma}_k \;=\; \min_{\theta_1,\ldots,\theta_k} \max_{\psi_1,\ldots,\psi_k} \min_{\Theta_{ij} \in C_k} ratio(\Theta_{ij}, \Psi) \;,$$

is again an upper bound on the performance ratio that can be obtained for MAX DI-CUT using any rotation function. (It is easy to check that $\bar{\gamma}_1$ is the upper bound defined in previous subsection.)

For brevity, we let $C_k = \{\Theta_1, \Theta_2, \ldots, \Theta_{k+1}\}$, where the order of these $k+1$ configurations is the one implicit in the definition of C_k. As in Section 3.3, we define the Lagrangian function

$$L(\Theta, \Psi, z) \;=\; ratio(\Theta_1, \Psi) - \sum_{i=1}^{k} z_i \cdot (ratio(\Theta_i, \Psi) - ratio(\Theta_{i+1}, \Psi)) \;.$$

It is again clear that

$$\bar{\gamma}_k \;=\; \min_{\theta_1,\ldots,\theta_k} \max_{\psi_1,\ldots,\psi_k} \min_{z_1,\ldots,z_k} L(\Theta, \Psi, z) \;,$$

where the minimum over $z_1, \ldots, z_k$ is under the constraints $1 \ge z_1 \ge z_2 \ge \cdots \ge z_k \ge 0$. As in Section 3.3, we believe that $\bar{\gamma}_k = \bar{\gamma}'_k$, where

$$\bar{\gamma}'_k \;=\; \min_{\theta_1,\ldots,\theta_k} \min_{z_1,\ldots,z_k} \max_{\psi_1,\ldots,\psi_k} L(\Theta, \Psi, z) \;,$$

and that if $\bar{\gamma}'_k = L(\Theta^*, \Psi^*, z^*)$ then, (Θ^*, Ψ^*, z^*) satisfy the $3k$ equations

$$\frac{\partial L}{\partial \theta_i}(\Theta, \Psi, z) \;=\; \frac{\partial L}{\partial \psi_i}(\Theta, \Psi, z) \;=\; \frac{\partial L}{\partial z_i}(\Theta, \Psi, z) \;=\; 0 \quad , \quad 1 \le i \le k \;.$$

Again, we have no proof of this claim at present, so we use these equations only as a heuristic for finding an approximate (Θ^*, Ψ^*, z^*). These values can then be used to obtain rigorous upper bounds on $\bar{\gamma}_k$. We did not carry out this second step in this case. We only computed the estimates of $\bar{\gamma}_k$ obtained by evaluating $L(\Theta^*, \Psi^*, z^*)$.

It is easy to verify that

$$L(\Theta, \Psi, z) \;=\; \sum_{i=1}^{k+1} a_i \cdot ratio(\Theta_i, \Psi) \;,$$

where $a_i = z_{i-1} - z_i$, for $1 \le i \le k+1$, where $z_0 = 1$ and $z_{k+2} = 0$. Note that $0 \le a_1, a_2, \ldots, a_{k+1} \le 1$ and that $\sum_{i=1}^{k+1} a_i = 1$. The a_i's give the proportional weights associated with the various configurations in an instance that matches the upper bound (if such an instance exists). In the sequel we give the values we obtained for a^*, instead of those obtained for z^*, as the values of a^* seem to be more informative. The corresponding values of z^* can be easily reconstructed from these values.

Using Mathematica, with a working precision of 100 decimal digits, we obtained the following estimates of $\bar{\gamma}_k$, for $1 \le k \le 8$:

$$
\begin{array}{rclcrcl}
\bar{\gamma}_1 & = & 0.86064152882557049833\ldots & \quad & \bar{\gamma}_5 & = & 0.85964474832821030805\ldots \\
\bar{\gamma}_2 & = & 0.85967326973683489935\ldots & & \bar{\gamma}_6 & = & 0.85964474630056827529\ldots \\
\bar{\gamma}_3 & = & 0.85964603062641909923\ldots & & \bar{\gamma}_7 & = & 0.85964474621778376358\ldots \\
\bar{\gamma}_4 & = & 0.85964479702793099730\ldots & & \bar{\gamma}_8 & = & 0.85964474621437901942\ldots
\end{array}
$$

As can be seen, slightly better upper bounds are obtained by using more and more points, though the improvements obtained are marginal. The critical angles $\theta_1^*, \ldots, \theta_8^*$ and $\psi_1^*, \ldots, \psi_8^*$ using which the estimate for $\bar{\gamma}_8$ was obtained are:

$$
\begin{array}{rclcrcl}
\theta_1^* & = & 1.18139587717550675254\ldots & \quad & \psi_1^* & = & 1.06966284202507978867\ldots \\
\theta_2^* & = & 1.29235161155383513547\ldots & & \psi_2^* & = & 1.21469857668774689861\ldots \\
\theta_3^* & = & 1.30576787540136961407\ldots & & \psi_3^* & = & 1.23220341315667012911\ldots \\
\theta_4^* & = & 1.30760701672554534766\ldots & & \psi_4^* & = & 1.23460174120156941752\ldots \\
\theta_5^* & = & 1.30795586663709469409\ldots & & \psi_5^* & = & 1.23505661947457880366\ldots \\
\theta_6^* & = & 1.30868543618098332387\ldots & & \psi_6^* & = & 1.23600789899692851440\ldots \\
\theta_7^* & = & 1.31357549964839172301\ldots & & \psi_7^* & = & 1.24238259543729115023\ldots \\
\theta_8^* & = & 1.34779882721322144442\ldots & & \psi_8^* & = & 1.28691718702660230001\ldots
\end{array}
$$

The corresponding weights $a_1^*, \ldots, a_9^*$ are:

$$
\begin{array}{rclcrcl}
a_1^* & = & 0.74003165264491210 94\ldots & \quad & a_6^* & = & 0.00012450541231465 23\ldots \\
a_2^* & = & 0.05524792819776304 80\ldots & & a_7^* & = & 0.00141770317464521 68\ldots \\
a_3^* & = & 0.00479519828268268 86\ldots & & a_8^* & = & 0.01609812641405112 53\ldots \\
a_4^* & = & 0.00042035919817037 46\ldots & & a_9^* & = & 0.18181210123017146 45\ldots \\
a_5^* & = & 0.00005242544528932 02\ldots & & & &
\end{array}
$$

It is interesting to note that many of the θ_i^*'s concentrate near $\theta \approx 1.31$, though the weights associated with the corresponding configurations are very small.

Instead of using the collection C_k as above, we could have used the larger collection $C'_k = \{\Theta_{ij} \mid 1 \le i, j \le k+1\}$. The results obtained using C'_k are identical to those obtained using C_k. (The optimal weights associated with configurations that are not in C_k would be 0.)

Finally, as was the case with MAX 2-SAT, it can be shown that no improvements can be obtained using rotation function that do not satisfy the condition $f(\pi - \theta) = \pi - f(\theta)$. We omit the details.

3.5 Almost optimal rotations

A performance ratio that almost matches the upper bound obtained in the previous section can be obtained using a piecewise linear rotation function $f(\theta)$ that passes through the points given in Figure 5. The value

θ	$f(\theta)$
0	0
1.0234866538333	0.8650372022203
1.1778245530705	1.0649981242812
1.2590574812375	1.1712260177359
1.3144090100621	1.2435377365940
1.4752427253552	1.4525213804115
$\frac{\pi}{2}$	$\frac{\pi}{2}$

Figure 5: A piecewise linear rotation function that yields a 0.859643-approximation algorithm for the MAX DI-CUT problem.

of $f(\theta)$ for $\frac{\pi}{2} \leq \theta \leq \pi$ is obtained using the relation $f(\theta) = \pi - f(\pi - \theta)$. The performance ratio obtained is 0.8596434254. This was again verified rigorously using GlobSol. A contour plot of the function $ratio(\theta_{01}, \theta_{02})$ with this almost optimal rotation function is given in Figure 6. Note that the plot is very different from the plot obtained using the rotation function $f_\lambda(\theta)$ with $\lambda = 0.5$, given in Figure 4

Piecewise linear rotation functions were also used in [HZ99]. The function given here was constructed using a Matlab program called `opt_dicut` which is a modification of the program `opt_funs` described in [HZ99].

4 Possible improvements and open problems

The open problems, and the avenues for obtaining improved approximation algorithms for the MAX 2-SAT and MAX DI-CUT problems fall into the following four categories.

4.1 Is the analysis tight?

We have shown that the MAX 2-SAT algorithm of Feige and Goemans [FG95] has a performance ratio of at least 0.931091, and that their approximation algorithm for MAX DI-CUT has a performance ratio of at least 0.859643. (Our bounds are slight improvements of similar bounds, obtained in a slightly less rigorous way by Feige and Goemans [FG95]). We do not know, however, how tight these bounds are. To obtain these bounds we used a *local* form of analysis, considering the clauses, or edges, of the instances one at a time. We have shown here that no better bounds can be obtained using such a local form of analysis. Can better bounds be obtained using a more *global* form of analysis?

For the MAX CUT approximation algorithm of Goemans and Williamson [GW95], Karloff [Kar96] (see also Alon and Sudakov [AS00]) has shown that the local analysis is also *globally* tight. Is this also the case for the MAX 2-SAT and MAX DI-CUT algorithms? To show, for example, that the local analysis of the MAX 2-SAT algorithm carried out here is globally tight, one would have to construct arbitrarily large MAX 2-SAT instances for each of which there exists an optimal solution $v_0, v_1, \ldots, v_n$ of the semidefinite programming relaxation such that for every clause $x_i \vee x_j$ of the instance, the vectors v_0, v_i and v_j are either in configuration Θ_1 of Section 2.3, or in configuration Θ_2, for the optimal $\theta^* = 1.82046073248821624015\ldots$. (The configuration $(0,0,0)$ which corresponds to the case $v_0 = v_i = v_j$ is also allowed.) Furthermore, the total weight of the clauses in configuration Θ_1 should be $1 - z^* = 0.75292066500323387879\ldots$ of the total

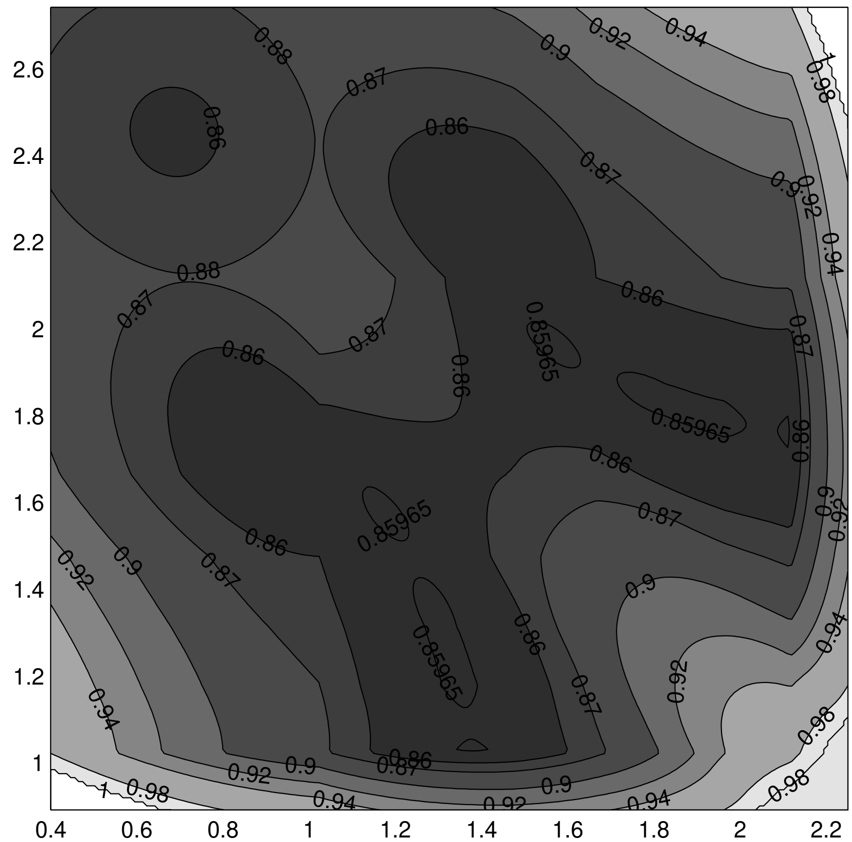

Figure 6: A contour plot of the function $ratio(\theta_{01}, \theta_{02})$ of the MAX DI-CUT problem with the almost optimal piecewise linear rotation function.

weight of all the clauses in the instance. These are fairly stringent conditions, and it is doubtful whether such instances exist. It would be enough, however, to construct a sequence of instances that tends toward the satisfaction of theses conditions. (This is how the tightness results of Karloff [Kar96] and of Alon and Sudakov [AS00] are obtained.)

4.2 Are there a better rounding procedures?

The MAX 2-SAT and MAX DI-CUT approximation algorithms of Feige and Goemans [FG95] use a fairly simple rounding procedure (though its analysis is quite involved). This rounding procedure rotates the vectors $v_0, v_1, \ldots, v_n$ obtained by solving the semidefinite programming relaxation of the instance as prescribed by a rotation function f. It then rounds the vectors obtained using a random hyperplane, as done by Goemans and Williamson [GW95].

We obtained here essentially optimal rotation functions, thereby slightly improving the performance ratios obtained by Feige and Goemans [FG95]. We also considered rounding procedures that use several rotation functions, or more generally, rounding procedures that choose the rotation function as a function of the optimal solution of the relaxation. The Lagrangian formulations of the upper bounds obtained show that such an approach cannot be used to obtain improved results, at least not using the current local mode of analysis.

Are there more general rounding procedures using which improved performance ratios can be obtained for MAX 2-SAT and MAX DI-CUT? We conjecture that the performance ratio for MAX DI-CUT can be improved to at least $\bar{\gamma} = 0.86064152882557049833\ldots$ of Section 3.3.

4.3 What is the quality of the relaxations?

Let I be an instance of, say, MAX 2-SAT. Let $opt(I)$ be the optimal value of this instance, and let $sdp(I)$ be the optimal value of the semidefinite programming relaxation of the instance I. Clearly $opt(I) \leq sdp(I)$, for every instance I. The *integrality ratio* of the semidefinite programming relaxation is defined to be $\eta = \inf_I opt(I)/sdp(I)$, where the infimum is taken, in this case, over all the MAX 2-SAT instances. The larger η is, the higher is the quality of the relaxation. All the approximation algorithms described in this paper compare the value of the solution produced to the value $sdp(I)$. Thus, if β is the performance of the best performance ratio obtained using the relaxation of the MAX 2-SAT problem given in Figure 1, and if η is the integrality gap of this relaxation, then clearly $\beta \leq \eta$. Thus, $0.931091 \leq \eta$. Feige and Goemans [FG95] show that $\eta \leq 64/65 \simeq 0.98462$. They also show that $\eta' \leq \frac{72}{65+5\sqrt{5}} \simeq 0.94513$, where η' is the integrality ratio of the relaxation of the MAX 2-SAT problem that only uses triangle inequalities of the form $v_0 \cdot v_i + v_0 \cdot v_j + v_i \cdot v_j \geq -1$, where $1 \leq i, j \leq 2n$. As our analysis only relies on these inequalities, we get that $0.931091 \leq \eta'$. The bounds we have on η' are therefore fairly tight. It would be interesting to obtained improved upper bounds on η, and on η', as these upper bounds give us limits on the performance ratios that can be obtained using these relaxations.

The corresponding bounds available on ξ and ξ', the integrality gaps of the relaxation of the MAX DI-CUT problem given in Figure 3, when all the triangle inequalities are used, or when only the triangle inequalities involving v_0 are used, are $0.859643 \leq \xi \leq 24/25 = 0.96$ and $0.859643 \leq \xi' \leq \frac{32}{25+5\sqrt{5}} \simeq 0.884455$, respectively. Obtaining improved upper bounds for ξ and ξ', and especially for ξ, is again a very interesting open problem. (The upper bound on ξ' is also the best known upper bound on the integrality ratio of the semidefinite programming relaxation of the MAX CUT problem *without* the triangle inequalities. For details see Feige and Goemans [FG95] and Goemans and Williamson [GW95].)

Finally, as pointed out by Feige and Goemans [FG95], stronger relaxations for the MAX 2-SAT and MAX DI-CUT problems can be obtained by imposing additional valid constraints. We may require, for example, that for every $0 \leq k_1, k_2, \ldots, k_5 \leq 2n$ we have $\sum_{i<j} v_{k_i} \cdot v_{k_j} \geq -2$. Such relaxations are expected to have higher integrality ratios, though it is not clear yet how to use such stronger relaxations to obtain improved approximation algorithms.

4.4 Improved inapproximability results?

As mentioned in the introduction, Håstad [Hås97] and Trevisan et al. [TSSW96] showed that no approximation algorithm for MAX 2-SAT can have a performance ratio of $\frac{21}{22} + \epsilon$, for any $\epsilon > 0$, and that no approximation algorithm for MAX DI-CUT can have an approximation ratio of $\frac{12}{13} + \epsilon$, for any $\epsilon > 0$, unless P=NP. Can these results be improved?

5 Concluding remarks

We do not believe that the improvements obtained here in the approximation ratio for MAX 2-SAT and MAX DI-CUT are important. What we believe *is* important is the fact we can *rigorously*, and with great accuracy, analyze the approximation algorithms that give these ratios, even through their analysis using traditional analytical means seems too difficult.

To carry out the analysis we use rigorous global optimization tools such as GlobSol [Kea96],[CK99]. (The other global optimization tool that we are aware of, Numerica [VMD97],[Van98], is a commercial product that was unavailable to us. Presumably Numerica could have been used instead of GlobSol.) The correctness of our results (or at least some of them) depends, therefore, on the correctness of these tools. The correctness of these tools depends, among other things, on the correctness of the compiler used to compile them, on the correctness of the operating system managing the computer on which these tools are run, and finally, on the correctness of the underlying hardware of the computer used. As each one of these is fairly complicated, it is probably realistic to expect that they all contain some bugs. We have to hope, therefore, that these bugs manifest themselves only in very rare situations and do not affect the results that we are interested in. We can increase our confidence in the results by trying several different tools. The results that we obtained using GlobSol, for example, are collaborated to some extent by computations done using Mathematica and Matlab.

We anticipate that the use of tools like GlobSol would became much more common in the future. This would, undoubtfully, increase the ease of using such tools and would help in making their reliability almost absolute. Rigorous global optimization tools may eventually be incorporated into more widely used systems such as Mathematica and Matlab.

We intend to use techniques that are similar to the techniques used here to obtain a rigorous analysis of more complicated approximation algorithms, such as the $\frac{7}{8}$-approximation algorithm for the MAX 3-SAT problem given by Karloff and Zwick [KZ97], and the $\frac{1}{2}$-approximation algorithm for the MAX 3-CSP problem given by Zwick [Zwi98a].

acknowledgment

We would like to thank Michel Goemans and Uri Feige for many insightful discussions over the years, and to Baker Kearfott for his valuable explanations and assistance with GlobSol.

References

[Ali95] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5:13–51, 1995.

[ALM+98] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45:501–555, 1998.

[AS98] S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM*, 45:70–122, 1998.

[AS00] N. Alon and B. Sudakov. Bipartite subgraphs and the smallest eigenvalue. *Combinatorics, Probability & Computing*, 2000. To appear.

[AW00] T. Asano and D.P. Williamson. Improved approximation algorithms for MAX SAT. In *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, California*, pages 96–105, 2000.

[Ber99] D.P. Bertsekas. *Nonlinear Programming.* Athena Scientific, second edition, 1999.

[CK99] G. Coreliss and R.B. Kearfott. *Global Solutions User Guide (draft).* Currently available from `http://www.mscs.mu.edu/~globsol/User_Guide/index.html`, 1999.

[DP93a] C. Delorme and S. Poljak. Combinatorial properties and the complexity of a max-cut approximation. *European Journal of Combinatorics*, 14(4):313–333, 1993.

[DP93b] C. Delorme and S. Poljak. Laplacian eigenvalues and the maximum cut problem. *Mathematical Programming*, 62(3, Ser. A):557–574, 1993.

[DP93c] C. Delorme and S. Poljak. The performance of an eigenvalue bound on the max-cut problem in some classes of graphs. *Discrete Mathematics*, 111(1-3):145–156, 1993.

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

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

[FL92] U. Feige and L. Lovász. Two-prover one-round proof systems: Their power and their problems (extended abstract). In *Proceedings of the 24rd Annual ACM Symposium on Theory of Computing, Victoria, Canada*, pages 733–744, 1992.

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

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

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

[Han92] E. Hansen. *Global optimization using Interval Analysis.* Marcel Dekker, 1992.

[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.

[HZ99] E. Halperin and U. Zwick. Approximation algorithms for MAX 4-SAT and rounding procedures for semidefinite programs. In *Proceedings of the 7th Conference on Integer Programming and Combinatorial Optimization, Graz, Austria*, pages 202–217, 1999.

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

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

[Kea96] R.K. Kearfott. *Rigorous global search: continuous problems.* Kluwer, 1996.

[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.

[MR99] S. Mahajan and H. Ramesh. Derandomizing approximation algorithms based on semidefinite programming. *SIAM Journal on Computing*, 28(5):1641–1663 (electronic), 1999.

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

[Van98] P. Van Hentenryck. A gentle introduction to Numerica. *Artificial Intelligence*, 103:209–235, 1998.

[VMD97] P. Van Hentenryck, L. Michel, and Y. Deville. *Numerica – A modeling language for global optimiztion.* MIT Press, 1997.

[Wol99] S. Wolfram. *The Mathematica book.* Cambridge University Press, fourth edition, 1999.

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

[Zwi98a] U. Zwick. Approximation algorithms for constraint satisfaction problems involving at most three variables per constraint. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, California*, pages 201–210, 1998.

[Zwi98b] U. Zwick. Finding almost-satisfying assignments. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing, Dallas, Texas*, pages 551–560, 1998.

[Zwi99] U. Zwick. Outward rotations: a tool for rounding solutions of semidefinite programming relaxations, with applications to max cut and other problems. In *Proceedings of the 31th Annual ACM Symposium on Theory of Computing, Atlanta, Georgia*, pages 679–687, 1999.

A Rigorous Global Optimization

The rigorous numerical bounds on the performance ratio are obtained using a rigorous global optimization tool called GlobSol [Kea96],[CK99]. (As mentioned, Numerica [VMD97],[Van98] could also have been used.) A rigorous global optimization tool is a program that computes rigorous bounds on the solutions of general non-linear constrained optimization problems of the form:

$$
\begin{array}{lrl}
\min & f(x_1,\ldots,x_n) & \\
\text{subject to} & g_j(x_1,\ldots,x_n) \le 0 & ,\quad 1 \le j \le m_1 \\
 & h_j(x_1,\ldots,x_n) = 0 & ,\quad 1 \le j \le m_2 \\
 & a_i \le x_i \le b_i & ,\quad 1 \le i \le n
\end{array}
$$

The optimization problem is specified by providing computer code for computing the function f and the functions g_j, for $1 \le j \le m_1$, and h_j, for $1 \le j \le m_2$, as well as the bounds a_i and b_i, for $1 \le i \le n$. The output of the global optimization tool is a list of boxes $XF_1,\ldots,XF_k$, where $XF_i = \Pi_{j=1}^{n+1}[\ell_{ij},u_{ij}]$. (Each box XF_i is an $(n+1)$-dimensional box $[\ell_{i1},u_{i1}] \times \cdots \times [\ell_{i,n+1},u_{i,n+1}]$.) We let $X_i = \Pi_{j=1}^{n}[\ell_{ij},u_{ij}]$ and $F_i = [\ell_{i,n+1},u_{i,n+1}]$. The following two conditions should be satisfied: (i) if $(x_1,\ldots,x_n) \in X_i$ then $f(x_1,\ldots,x_n) \in F_i$, and, more importantly, (ii) if (x^*,f^*) is an optimal solution of the optimization problem then there exists $1 \le i \le k$ such that $(x^*,f^*) \in XF_i$. Thus $\min_{i=1}^{k} \ell_{i,n+1}$ is a rigorous *lower* bound on the optimal value of the above minimization problem. Note that the above two conditions do not require a box XF_i to contain a feasible point, though the global optimization tool would undoubtedly try to identify boxes that do not contain feasible points and remove them from the list. The global optimization tool may be able to assert the existence of feasible points within some of the boxes. Without loss of generality, we assume that the first k' boxes, i.e., $XF_1,\ldots,XF_{k'}$, for some $k' \le k$, were found to contain feasible points. In this case, $\max_{i=1}^{k'} \ell_{i,n+1}$ is a rigorous *upper* bound on the optimal value of the optimization problem.

The user of a global optimization tool usually specifies an accuracy parameter ϵ. The global optimization tool then tries to output boxes such that $u_{ij} - \ell_{ij} \le \epsilon$, for every $1 \le i \le k$ and $1 \le j \le n+1$. The global optimization tool may fail to achieve this goal due to several reasons. The arithmetic it uses may not be accurate enough to produce boxes whose sides have length less than ϵ, or too much time or space may be needed to produce a list of slim enough boxes. In such a case, the global optimization tool should report failure. Note, however, that if a result is returned by the global optimization tool, then this result is guaranteed to be correct. For more exact definitions of the functionality of rigorous global optimization tools, see [Kea96],[CK99] and [VMD97],[Van98].

The main tool used by rigorous global optimization systems such as GlobSol and Numerica is *interval arithmetic* (see, e.g., Hansen [Han92]). Instead of performing the arithmetical, and other operations, on numbers, these operations are now performed on intervals. This automatically keeps track of the numerical errors that are inherent in these operations. For example, if $X_1 = [a_1,b_1]$ and $X_2 = [a_2,b_2]$ are two intervals, then their addition returns an interval $[\underline{a_1+a_2},\overline{b_1+b_2}]$, where $\underline{a_1+a_2}$ is machine representable number obtained by rounding a_1+a_2 downwards, while $\overline{b_1+b_2}$ is a machine representable number obtained by rounding b_1+b_2 upwards. Similarly, if $X=[a,b]$ is an interval, and SIN is an *interval extension* of the sine function, then $Y = \mathrm{SIN}(X)$ is an interval $Y=[c,d]$, where both c and d are machine representable numbers, such that $\{\sin(x) \mid a \le x \le b\} \subseteq Y$. Using interval extensions of all the arithmetical operations and of all the basic elementary functions, we can obtain interval extensions of all the functions that can be defined using them. Furthermore, a code for computing a function f can be easily converted into code for computing an interval extension F of f.

In particular, we can obtain interval extensions of the functions f, g_j and h_j that appear in the the definition of the global minimization problem mentioned above. We can then use the following naive, yet powerful, algorithm to obtain rigorous bounds on the solution of the minimization problem. We begin with a working list of boxes that contains the single box $\Pi_{i=1}^n[a_i, b_i]$, and an upper bound u on the value of the global minimum. Initially $u = +\infty$. Given a box X in the working list, we evaluate $F(X)$, $G_j(X)$, for $1 \le j \le m_1$, and $H_j(X)$, for $1 \le j \le m_2$, where the functions F, G_j and H_j are interval extensions of the functions f, g_j and h_j. If, using the (interval) values of $G_j(X)$ and $H_j(X)$, we can conclude that X does not contain feasible points, we can remove X from the list. Similarly, if $F(X) > u$ (we say that $[a, b] > c$ if and only if $a > c$), we can again eliminate X from the list, as no global minimum is contained in it. Otherwise, we try to find a feasible point $x \in X$. A naive way of doing that, for example, is simply to check if the midpoint of X, or if one of its corners, is feasible. (This should again be done using interval arithmetic. Note also that this approach is bound to fail if there are equality constraints.) If we find a feasible point $x \in X$, we evaluate $F([x, x])$. If $F([x, x]) = [a, b]$ and $b < u$, we can let $u \leftarrow b$. Then, if the box X and the interval $F(X)$ are slim enough, we can move the box $X \times F(X)$ to the output list. Otherwise, we split X into two sub-boxes X_1 and X_2, remove X from our working list and replace it by X_1 and X_2, and iterate. If the functions f, g_j and h_j are continuous, if the underlying arithmetic used is accurate enough, and if the splitting of X into X_1 and X_2 reduces the length of the largest size of these boxes, then this process is guaranteed to terminate. The running time may be large, however, and the output list produced may be very long. Nevertheless, this naive algorithm can already be used to rigorously solve some non-trivial problems.

GlobSol [Kea96],[CK99] and Numerica [VMD97],[Van98] use much more sophisticated algorithms to solve global optimization problems. Although in general, their running time may be exponential in the size of the instance (and they may sometime fail to achieve the requested accuracy), on many reasonable instances, such as the instances of interest to us here, they produce concise, and fairly accurate, solutions within a matter of seconds or minutes.

B Rigorous bounds on performance ratios

In this section we describe the way in which the rigorous bounds on the performance ratios were obtained. We use the original MAX 2-SAT approximation algorithm of Feige and Goemans as an example. The same technique can be applied to all the approximation algorithms described in this paper.

The optimization problem that defines $\beta(f)$ (see Section 2.1) seems to be a problem that can be solved directly using a global optimization tool like GlobSol. This, unfortunately, is not the case, at least with the current implementation of GlobSol which expects that the functions f, g_j and h_j that appear in the description of the optimization problem (see the generic global minimization problem given in the previous section) to be defined for every $(x_1, \ldots, x_n)$ such that $a_i \le x_i \le y_i$, for $1 \le i \le n$. In particular, $f(x_1, \ldots, x_n)$ should be defined at non-feasible points $(x_1, \ldots, x_n)$. Unfortunately, our objective function $ratio(\theta_{01}, \theta_{02}, \theta_{12})$ is not defined for every $0 \le \theta_{01}, \theta_{02}, \theta_{12} \le \pi$. This is due to two reasons. The first is that when $\theta_{01} = \theta_{02} = \theta_{12} = 0$, the denominator of $ratio(\theta_{01}, \theta_{02}, \theta_{12})$ vanishes and we get division by 0. The second, and more serious problem, is that for some non-feasible points $(\theta_{01}, \theta_{02}, \theta_{12})$, the argument of the arccos in the definition of $g(\theta_{01}, \theta_{02}, \theta_{12})$ assumes values that are not in the range $[-1, 1]$ and $g(\theta_{01}, \theta_{02}, \theta_{12})$ is therefore undefined (or is not real).

We therefore consider the slightly easier problem of showing that $\beta(f) \ge \beta$, for some fixed β. This is equivalent to showing that

$$\frac{f(\theta_{01})+f(\theta_{02})+g(\theta_{01},\theta_{02},\theta_{12})}{2\pi} \;\geq\; \beta\,\frac{3-\cos\theta_{01}-\cos\theta_{02}-\cos\theta_{12}}{4}\;,$$

or equivalently that

$$g(\theta_{01},\theta_{02},\theta_{12}) \;\geq\; h_\beta(\theta_{01},\theta_{02},\theta_{12})$$

where

$$h_\beta(\theta_{01},\theta_{02},\theta_{12}) \;=\; \frac{\beta\pi}{2}(3-\cos\theta_{01}-\cos\theta_{02}-\cos\theta_{12})-f(\theta_{01})-f(\theta_{02})\;,$$

for every $(\theta_{01},\theta_{02},\theta_{12})$ that satisfy the required inequalities. Recall that $g(\theta_{01},\theta_{02},\theta_{12})$ is the angle between the vectors v_1' and v_2' obtained by rotating v_1 and v_2 with respect to v_0. Thus, for every feasible $(\theta_{01},\theta_{02},\theta_{12})$ we have $g(\theta_{01},\theta_{02},\theta_{12})\in[0,\pi]$. (If $(\theta_{01},\theta_{02},\theta_{12})$ is not feasible, for example, if $\theta_{12}>\theta_{01}+\theta_{02}$, then there are no three vectors v_0, v_1 and v_2 such that the three angles between them are $(\theta_{01},\theta_{02},\theta_{12})$ and $g(\theta_{01},\theta_{02},\theta_{12})$ may be undefined.) If we knew that $h_\beta(\theta_{01},\theta_{02},\theta_{12})\in[0,\pi]$, for every feasible $(\theta_{01},\theta_{02},\theta_{12})$ then, as the function $\cos x$ is decreasing in the range $[0,\pi]$, we could have replaced the inequality $g(\theta_{01},\theta_{02},\theta_{12})\geq h_\beta(\theta_{01},\theta_{02},\theta_{12})$ by the inequality $\cos g(\theta_{01},\theta_{02},\theta_{12})\leq\cos h_\beta(\theta_{01},\theta_{02},\theta_{12})$, or equivalently

$$\begin{aligned}&\cos f(\theta_{01})\cos f(\theta_{02})+\frac{\cos\theta_{12}-\cos\theta_{01}\cos\theta_{02}}{\sin\theta_{01}\sin\theta_{02}}\sin f(\theta_{01})\sin f(\theta_{02})\\ &\leq\;\cos\left(\tfrac{\beta\pi}{2}(3-\cos\theta_{01}-\cos\theta_{02}-\cos\theta_{12})-f(\theta_{01})-f(\theta_{02})\right)\;.\end{aligned}$$

Finally, by multiplying both sides by $\sin\theta_{01}\sin\theta_{02}$, which is always non-negative, and moving things around, we obtain the final inequality

$$\begin{gathered}\mathit{diff}_\beta(\theta_{01},\theta_{02},\theta_{12}) \;=\\ \left(\cos\left(\tfrac{\beta\pi}{2}(3-\cos\theta_{01}-\cos\theta_{02}-\cos\theta_{12})-f(\theta_{01})-f(\theta_{02})\right)-\cos f(\theta_{01})\cos f(\theta_{02})\right)\sin\theta_{01}\sin\theta_{02}\\ -\;(\cos\theta_{12}-\cos\theta_{01}\cos\theta_{02})\sin f(\theta_{01})\sin f(\theta_{02})\quad\geq\quad 0\;.\end{gathered}$$

The advantage of using the function $\mathit{diff}_\beta(\theta_{01},\theta_{02},\theta_{12})$ is that, unlike $\mathit{ratio}(\theta_{01},\theta_{02},\theta_{12})$, it is defined for all $0\leq\theta_{01},\theta_{02},\theta_{12}\leq\pi$. Also, it only uses the trigonometric functions sin and cos, and avoids using the inverse cosine function. The disadvantage of the using $\mathit{diff}_\beta(\theta_{01},\theta_{02},\theta_{12})$ is that we have to work with a fixed value of β.

But, is it really the case that $h_\beta(\theta_{01},\theta_{02},\theta_{12})\in[0,\pi]$ for every feasible $(\theta_{01},\theta_{02},\theta_{12})$? It turns out that $h_\beta(\theta_{01},\theta_{02},\theta_{12})$ may actually be negative for some feasible $(\theta_{01},\theta_{02},\theta_{12})$. This, however, is not really a problem. If $h_\beta(\theta_{01},\theta_{02},\theta_{12})<0$ and $(\theta_{01},\theta_{02},\theta_{12})$ is feasible, then the inequality $g(\theta_{01},\theta_{02},\theta_{12})\geq h_\beta(\theta_{01},\theta_{02},\theta_{12})$ is automatically satisfied. It is thus enough to verify that $\mathit{diff}_\beta(\theta_{01},\theta_{02},\theta_{12})\geq 0$ for feasible points $(\theta_{01},\theta_{02},\theta_{12})$ that satisfy the additional constraint $h_\beta(\theta_{01},\theta_{02},\theta_{12})\geq 0$. Luckily, it is the case that $h_\beta(\theta_{01},\theta_{02},\theta_{12})\leq\pi$ for every feasible $(\theta_{01},\theta_{02},\theta_{12})$, at least for the values of β in which we are interested.

We have thus reduced the problem of showing that a given approximation algorithm for MAX 2-SAT achieves a performance ratio of β to proving the following two relations:

$$A \;=\; \left[\begin{array}{ll}\min & \mathit{diff}_\beta(\theta_{01},\theta_{02},\theta_{12})\\ \text{subject to} & \cos\theta_{01}+\cos\theta_{02}+\cos\theta_{12}\geq -1\\ & \cos\theta_{01}-\cos\theta_{02}-\cos\theta_{12}\geq -1\\ & -\cos\theta_{01}-\cos\theta_{02}+\cos\theta_{12}\geq -1\\ & -\cos\theta_{01}+\cos\theta_{02}-\cos\theta_{12}\geq -1\\ & h_\beta(\theta_{01},\theta_{02},\theta_{12})\geq 0\\ & 0\leq\theta_{01},\theta_{02},\theta_{12}\leq\pi\end{array}\right] \;\geq\; 0$$

and

$$
B \;=\; \left[\begin{array}{ll} \max & h_\beta(\theta_{01},\theta_{02},\theta_{12}) \\ \text{subject to} & \cos\theta_{01} + \cos\theta_{02} + \cos\theta_{12} \geq -1 \\ & \cos\theta_{01} - \cos\theta_{02} - \cos\theta_{12} \geq -1 \\ & -\cos\theta_{01} - \cos\theta_{02} + \cos\theta_{12} \geq -1 \\ & -\cos\theta_{01} + \cos\theta_{02} - \cos\theta_{12} \geq -1 \\ & 0 \leq \theta_{01},\theta_{02},\theta_{12} \leq \pi \end{array} \right] \;\leq\; \pi
$$

These two relations can now be verified using a rigorous global optimization package such as GlobSol, *and* some analytical arguments. For $\beta = 0.93109006$, for example, it turns out that indeed $A = 0$, where the global minimum is attained at $\theta_{01} = \theta_{02} = \theta_{12} = 0$. As for B, we get that $B \leq 2.8$, so there is a lot of slack here. For more information on how this is done, see the next section.

C Some GlobSol examples

As explained in the previous section, we cannot ask GlobSol directly to evaluate $\beta(f)$, as the function $ratio(\theta_{01},\theta_{02},\theta_{12})$ is not defined for every $0 \leq \theta_{01},\theta_{02},\theta_{12} \leq \pi$. We can, however, ask GlobSol to find the minimum of $ratio(\theta_{01},\theta_{02},\theta_{12})$ over a smaller region in which $ratio(\theta_{01},\theta_{02},\theta_{12})$ is defined at every point. This is the case, for example, for the region $1.25 \leq \theta_{01},\theta_{02} \leq 1.9$ and $2.05 \leq \theta_{12} \leq 2.15$ which, with hindsight, we know to contain the two global minima. The file `max2sat.f90` given in Figure 7 contains a description of the problem in the form required by GlobSol, i.e., as a Fortran 90 program that uses the data types of GlobSol. For more details on the required form, see [Kea96],[CK99]. Note that the description of the region over which the problem is to be solved is not specified here, but in a separate file that is not shown here. By running GlobSol on this problem we get the two output boxes given in Section 2.2.

These results do not show yet, rigorously, that the performance ratio of the MAX 2-SAT approximation algorithm is at least 0.931090068, as claimed in Section 2.2. To finish the proof we use the technique described in the previous section. The file `max2sat_check.f90`, given in Figure 8, describes the first optimization problem that has to be solved (the one that, in the notation of the previous section, gives the value A). By running GlobSol on this problem we get that the global minimum is at least -0.16×10^{-17}, and that the global minimum is attained when $0 \leq \theta_{01},\theta_{02},\theta_{12} \leq 7\times10^{-5}$. Furthermore, GlobSol determines that $diff_\beta(\theta_{01},\theta_{02},\theta_{12})$ is non-negative outside this box. This is almost what we want, but not quite. We have to show that the global minimum is 0, and that it is attained at $\theta_{01} = \theta_{02} = \theta_{12} = 0$. GlobSol is not able to obtain this result automatically as the first order partial derivatives of $diff_\beta(\theta_{01},\theta_{02},\theta_{12})$ vanish at $\theta_{01} = \theta_{02} = \theta_{12} = 0$. However, it turns out that it is not difficult to check, analytically, that $diff_\beta(\theta_{01},\theta_{02},\theta_{12})$ is non-negative for every $0 \leq \theta_{01},\theta_{02},\theta_{12} \leq 7\times10^{-5}$. We omit the straightforward, but somewhat tedious details.

The second optimization problem that has to be solved, the one that gives the value B, is of a similar, in fact simpler, form and it is handled in a similar manner. This completes the proof that the performace ratio of the Feige-Goemans approximation algorithm for MAX 2-SAT with $\lambda = 0.80676786784$ is indeed at least 0.931090068.

```
PROGRAM max2sat

! Find the performance ratio of the Feige and Goemans MAX 2-SAT algorithm

USE CODELIST_CREATION

   TYPE(CDLVAR)  , DIMENSION(3) :: tet
   TYPE(CDLLHS)  , DIMENSION(1) :: ratio
   TYPE(CDLINEQ) , DIMENSION(4) :: ineq

   TYPE(CDLVAR) :: pi , lambda , ftet1 , ftet2 , cosalp , gtet

   OUTPUT_FILE_NAME='max2sat.CDL'
   CALL INITIALIZE_CODELIST(tet)

   pi = ivl(3.14159265358979323846264338327D0 , 3.14159265358979323846264338333D0)
   lambda  =              ivl(0.80676786784D0 , 0.80676786784D0)

   ftet1 = (1-lambda)*tet(1) + lambda*(pi/2)*(1-COS(tet(1)))
   ftet2 = (1-lambda)*tet(2) + lambda*(pi/2)*(1-COS(tet(2)))

   cosalp   = ( COS(tet(3)) - COS(tet(1))*COS(tet(2)) ) / ( SIN(tet(1))*SIN(tet(2)) )
   gtet     =     ACOS( COS(ftet1)*COS(ftet2) + cosalp*SIN(ftet1)*SIN(ftet2) )
   ratio(1) = (2/pi) * ( ftet1+ftet2+gtet ) / ( 3-COS(tet(1))-COS(tet(2))-COS(tet(3)) )

   ineq(1) = - COS(tet(1)) - COS(tet(2)) - COS(tet(3)) - 1
   ineq(2) =   COS(tet(1)) + COS(tet(2)) - COS(tet(3)) - 1
   ineq(3) =   COS(tet(1)) - COS(tet(2)) + COS(tet(3)) - 1
   ineq(4) = - COS(tet(1)) + COS(tet(2)) + COS(tet(3)) - 1

   CALL FINISH_CODELIST

END PROGRAM max2sat
```

Figure 7: The file max2sat.f90

```
PROGRAM max2sat_check

! Verify that the performance ratio of the Feige and Goemans
! MAX 2-SAT algorithm is at least 0.931090068

USE CODELIST_CREATION

   TYPE(CDLVAR)  , DIMENSION(3) :: tet
   TYPE(CDLLHS)  , DIMENSION(1) :: diff
   TYPE(CDLINEQ) , DIMENSION(5) :: ineq

   TYPE(CDLVAR) :: pi , lambda , beta , ftet1 , ftet2 , temp

   OUTPUT_FILE_NAME='max2sat_check.CDL'
   CALL INITIALIZE_CODELIST(tet)

   pi = ivl(3.14159265358979323846264338327D0, &
            3.14159265358979323846264338328D0)

   lambda  = ivl(0.80676786784D0 , 0.80676786784D0)
   beta    = ivl(0.931090068D0   , 0.931090068D0  )

   ftet1 = (1-lambda)*tet(1) + lambda*(pi/2)*(1-COS(tet(1)))
   ftet2 = (1-lambda)*tet(2) + lambda*(pi/2)*(1-COS(tet(2)))

   temp = beta*pi/2*(3-COS(tet(1))-COS(tet(2))-COS(tet(3)))-ftet1-ftet2

   diff(1) = &
     (COS(temp) - COS(ftet1 )*COS(ftet2 )) * SIN(tet(1))*SIN(tet(2)) &
   - (COS(tet(3))-COS(tet(1))*COS(tet(2))) * SIN(ftet1 )*SIN(ftet2 )

   ineq(1) = - COS(tet(1)) - COS(tet(2)) - COS(tet(3)) - 1
   ineq(2) =   COS(tet(1)) + COS(tet(2)) - COS(tet(3)) - 1
   ineq(3) =   COS(tet(1)) - COS(tet(2)) + COS(tet(3)) - 1
   ineq(4) = - COS(tet(1)) + COS(tet(2)) + COS(tet(3)) - 1

   ineq(5) = - temp

   CALL FINISH_CODELIST

END PROGRAM max2sat_check
```

Figure 8: The file max2sat_check.f90