Annals of Mathematics and Artificial Intelligence **37:** 285–305, 2003.

On semidefinite programming relaxations of $(2+p)$-SAT

E. de Klerk and H. van Maaren

Faculty of Information Technology and Systems, Delft University of Technology, P.O. Box 5031, 2600 GA Delft, The Netherlands
E-mail: {e.deklerk,h.vanmaaren}@its.tudelft.nl

Recently, de Klerk, van Maaren and Warners [10] investigated a relaxation of 3-SAT via semidefinite programming. Thus a 3-SAT formula is relaxed to a semidefinite feasibility problem. If the feasibility problem is infeasible then a certificate of unsatisfiability of the formula is obtained. The authors proved that this approach is exact for several polynomially solvable classes of logical formulae, including 2-SAT, pigeonhole formulae and mutilated chessboard formulae. In this paper we further explore this approach, and investigate the strength of the relaxation on $(2+p)$-SAT formulae, i.e., formulae with a fraction p of 3-clauses and a fraction $(1-p)$ of 2-clauses. In the first instance, we provide an empirical computational evaluation of our approach. Secondly, we establish approximation guarantees of randomized and deterministic rounding schemes when the semidefinite feasibility problem is feasible, and also present computational results for the rounding schemes. In particular, we do a numerical and theoretical comparison of this relaxation and the stronger relaxation by Karloff and Zwick [15].

Keywords: approximation algorithms, satisfiability, semidefinite programming, randomized rounding

1. Introduction

Recently, de Klerk et al. [10] proposed a relaxation of the satisfiability problem (SAT) to a semidefinite programming (SDP) feasibility problem. The authors showed that this approach – called the *gap relaxation* – is exact[1] for certain classes of polynomially solvable formulae including:

- 2-SAT;
- unsatisfiable formulae from graph colouring instances where clique constraints imply unsatisfiability;
- pigeonhole formulae;
- 'mutilated chessboard' formulae.

The last two classes of formulae are special cases of a general class of infeasible assignment problems for which the gap relaxation is exact. Pigeonhole formulae are of

[1] We call the relaxation exact if it is feasible if and only if the formula is satisfiable.

independent interest, since their solution by the resolution algorithm requires exponential time [12].[2]

Semidefinite programming relaxations of combinatorial optimization problems dates back to work by Lovász [16] in the seventies.

The gap relaxation in [10] is closely related to the types of MAX-SAT relaxations studied by Karloff and Zwick [15], Zwick [23], and Halperin and Zwick [13]. The work of these authors in turn employs the ideas of semidefinite approximation algorithms and associated randomized rounding, as introduced in the seminal work of Goemans and Williamson [11] on approximation algorithms for the MAX-CUT and other problems.

The gap relaxation is by no means the strongest possible relaxation of SAT formulae using SDP. In fact, Lovász and Schrijver have shown how to obtain exact descriptions of the feasible integer polytope of the maximum stable set problem. (Recall that there is a simple polynomial time reduction from the maximum stable set problem to SAT.) Of course, the resulting SDP's may be exponentially large in the original problem size.

The gap relaxation should therefore be seen as a trade-off between practically implementable and theoretically strong relaxations. Only one constraint is introduced per clause, and these constraints have special (rank one) structure in the resulting semidefinite feasibility problem, which can be exploited by interior point algorithms. By contrast, the $7/8$ MAX-3-SAT relaxation Karloff and Zwick requires 3 constraints per 3-clause, and these constraints do not have rank one in the resulting SDP. We will show that – as might be expected – the gap relaxation can be solved more quickly, but that the relaxation by Karloff and Zwick can detect unsatisfiability in more cases.

A disadvantage of the gap (and Karloff–Zwick) relaxation is that it is always feasible for 3-SAT formulae if there are no 2-clauses present. In other words, in order to detect unsatisfiability, 2-clauses must be present. For this reason $(2+p)$-SAT formulae provide a natural test environment for this approach.

If the gap relaxation is feasible, then one can still use a solution of the relaxation in an attempt to generate a truth assignment for the SAT-formula in question (rounding schemes). In this paper we will also explore the theoretical properties and practical performance of some rounding schemes. Thus we show that the best rounding scheme satisfies roughly 91% of the 2-clauses and 70% of the 3-clauses (in expectation). This is worse than the $7/8$ guarantee which is obtained from rounding the Karloff–Zwick relaxation, but we show that rounding the gap relaxation can be useful in generating truth assignments in practice.

Outline

This paper is organized as follows. In section 2 we review how CNF formulae may be relaxed to semidefinite feasibility problems. We discuss the issues involved in solving these relaxations using interior point methods in section 3. This is followed by some numerical results on detecting unsatisfiability in $(2+p)$-SAT formulae in section 4.

[2] Polynomial time cutting plane proofs are known for the pigeonhole formulae [6], but the gap relaxation solves these formulae in a fully automated way, without the need for problem-dependent cutting planes.

We consider the situation where the gap relaxation is feasible in section 5, and derive performance guarantees for different rounding schemes which aim at constructing truth assignments. Finally, we present some numerical results for the rounding schemes in section 7, as well as a numerical comparison of the relative strength of the gap and Karloff–Zwick relaxations.

2. Boolean quadratic clause representations and their relaxations

We consider the satisfiability problem in conjunctive normal form (CNF). A propositional formula Φ in CNF is a conjunction of clauses, where each clause $\mathbf{C}_i$ is a disjunction of *literals*. Each literal is an *atomic proposition* (or *logical variable*) or its negation ($\neg$). Let m be the number of clauses and n the number of atomic propositions. A clausal propositional formula is denoted as $\Phi = \mathbf{C}_1 \wedge \mathbf{C}_2 \wedge \cdots \wedge \mathbf{C}_m$, where each clause $\mathbf{C}_i$ is of the form

$$\mathbf{C}_i = \bigvee_{j \in I_i} p_j \vee \bigvee_{j \in J_i} \neg p_j,$$

with $I_i, J_i \subseteq \{1, \ldots, n\}$ disjoint. The satisfiability (SAT) problem of propositional logic is to determine whether or not an assignment of truth values to the logical variables exists such that each clause evaluates to true (i.e., one of its literals is true) and thus the formula is true. The MAX-k-SAT problem is to find the maximum number of clauses which can be simultaneously satisfied, where the clause length is at most k; the MAX-$\{k\}$-SAT problem involves clauses of length exactly k. Similar definitions hold for k-SAT and $\{k\}$-SAT.

We will also refer to $(2+p)$-SAT ($0 \leqslant p \leqslant 1$), which is the class of 3-SAT formulae where a fraction $(1-p)$ of the clauses are 2-clauses and the remaining clauses are 3-clauses. Finally, we will call $\alpha = m/n$ the clause/variable ratio.

Associating a $\{-1, 1\}$-variable x_i with each logical variable p_i, a clause $\mathbf{C}_k$ can be written as a linear inequality in the following way.

$$C_k(x) = \sum_{i \in I_k} x_i - \sum_{j \in J_k} x_j \geqslant 2 - \ell(\mathbf{C}_k), \tag{1}$$

where $\ell(\mathbf{C}_k)$ denotes the length of clause k, i.e., $\ell(\mathbf{C}_k) = |I_k \cup J_k|$. Using matrix notation, the integer linear programming formulation of the satisfiability problem can be stated as

$$\text{find } x \in \{-1, 1\}^n \quad \text{such that } Ax \geqslant r.$$

The matrix $A \in \mathbb{R}^{m \times n}$ is called the *clause-variable* matrix. We have that $a_k^T x = C_k(x)$, where a_k^T denotes the kth row of A. Obviously, $a_{ki} = 1$ if $i \in I_k$, $a_{ki} = -1$ if $i \in J_k$, while $a_{ki} = 0$ for any $i \notin I_k \cup J_k$. Furthermore, $r_k = 2 - \ell(\mathbf{C}_k)$.

In order to apply a SDP relaxation to a CNF formula, one must represent each clause as a Boolean quadratic (in)equality. Subsequently, one or more quadratic (in)equalities are formulated for a given clause $\mathbf{C}_j$ which are satisfied if x corresponds to a truth assignment for clause $\mathbf{C}_j$.

There are many ways to do this. Karloff and Zwick [15] derived a set of seven valid quadratic functions which represent all possible valid quadratic inequalities for 3-clauses (by taking linear combinations of the seven inequalities); see also [10]. As an example, let us consider the clause $p_1 \vee p_2 \vee p_3$. All valid quadratic inequalities for this clause are nonnegative aggregations of the following seven quadratic inequalities:

$$\begin{aligned} x_1x_2 + x_1x_3 - x_2 - x_3 &\leqslant 0, \\ x_1x_2 + x_2x_3 - x_1 - x_3 &\leqslant 0, \\ x_1x_3 + x_2x_3 - x_1 - x_2 &\leqslant 0, \\ -x_1x_2 - x_1x_3 - x_2x_3 - 1 &\leqslant 0, \\ -x_1x_2 + x_1 + x_2 - 1 &\leqslant 0, \\ -x_1x_3 + x_1 + x_3 - 1 &\leqslant 0, \\ -x_2x_3 + x_2 + x_3 - 1 &\leqslant 0, \\ x_1, x_2, x_3 &\in \{-1, 1\}. \end{aligned}$$

The analogous inequalities for other possible 3-clauses are obtained by replacing x_i by $-x_i$ if p_i appears negated in the clause.

More recently, Halperin and Zwick [13] derived a similar set of generic quadratic inequalities for 4-clauses. Van Maaren and Warners [21,22] and de Klerk et al. [10] considered so-called elliptic representations of clauses. For our example clause $p_1 \vee p_2 \vee p_3$ the elliptic representation takes the form:

$$(x_1 + x_2 + x_3 - 1)^2 \leqslant 4, \quad x_1, x_2, x_3 \in \{-1, 1\}.$$

Using $x_i^2 = 1$ and simplifying we get

$$\sum_{i \neq j} x_i x_j - \sum_{i=1}^{3} x_i \leqslant 0. \tag{2}$$

This quadratic inequality is simply the sum of the first three of the seven generic inequalities listed above. Karloff and Zwick [15] showed that by representing a 3-clause by the first three inequalities a $7/8$ approximation algorithm is obtained for MAX-3-SAT, by using the Goemans–Williamson randomized rounding technique [11].

Let us assume that

$$x^T A_i x + 2b_i^T x + c_i \leqslant 0, \quad x \in \{-1, 1\}^n \tag{3}$$

is a valid Boolean quadratic representation of a clause $\mathbf{C}_i$. In other words, (3) holds if and only if x corresponds to a truth assignment for clause $\mathbf{C}_i$. Equation (3) can be rewritten as

$$\operatorname{Tr}\begin{bmatrix} A_i & b_i \\ b_i^T & c_i \end{bmatrix}\begin{bmatrix} xx^T & x \\ x^T & 1 \end{bmatrix} \leqslant 0, \quad x \in \{-1, 1\}^n, \tag{4}$$

where 'Tr' denotes the trace operator. We can relax (4) to

$$\mathrm{Tr}\begin{bmatrix} A_i & b_i \\ b_i^T & c_i \end{bmatrix}\begin{bmatrix} X & y \\ y^T & 1 \end{bmatrix} \leqslant 0, \tag{5}$$

where X is now a symmetric $n \times n$ matrix which satisfies

$$X - yy^T \succeq 0, \tag{6}$$

and $X_{jj} = 1$ $(j = 1, \ldots, n)$. Note that the entries X_{ij} correspond to the products $x_i x_j$ and y_i corresponds to x_i. Also note that (6) is equivalent to

$$\begin{bmatrix} X & y \\ y^T & 1 \end{bmatrix} \succeq 0, \tag{7}$$

by the Schur complement theorem.

We therefore have a general procedure by which we can relax a clause to one or more linear matrix inequalities. The only non-mechanical step is to select which valid quadratic (in)equalities will be used to represent each clause.

The gap relaxation

In this paper we consider the gap relaxation by de Klerk et al. [10] which uses the inequalities of the type (2) for 3-clauses, and for the generic 2-clause

$$p_1 \vee p_2$$

uses the valid quadratic equality

$$(x_1 + x_2 - 1)^2 = 1.$$

Again, if p_1 is negated, then x_1 is replaced by $-x_1$ in (8), etc. Simplifying using $x_i^2 = 1$ as before, we get

$$x_1x_2 - x_1 - x_2 = -1. \tag{8}$$

In the relaxation the generic 2-clause $p_1 \vee p_2$ corresponds to a 3×3 principal submatrix of the matrix in (7), namely,

$$\begin{bmatrix} 1 & X_{12} & y_1 \\ X_{12} & 1 & y_2 \\ y_1 & y_2 & 1 \end{bmatrix},$$

and (8) implies

$$X_{12} - \sum_{i=1}^{2} y_i = -1.$$

Similarly, the generic 3-clause $p_1 \vee p_2 \vee p_3$ corresponds to a 4×4 principal submatrix of the matrix in (7), namely,

$$\begin{bmatrix} 1 & X_{12} & X_{13} & y_1 \\ X_{12} & 1 & X_{23} & y_2 \\ X_{13} & X_{23} & 1 & y_3 \\ y_1 & y_2 & y_3 & 1 \end{bmatrix},$$

and (2) implies

$$X_{12} + X_{13} + X_{23} - \sum_{i=1}^{3} y_i \leqslant 0.$$

Note that the identity matrix is always feasible for the gap relaxation of $\{3\}$-SAT formulae. This means that unsatisfiability can only be detected if 2-clauses are also present.

Definition 2.1 (Gap relaxation). Formally, we can define the gap relaxation in terms of the clause-variable matrix; for the gap relaxation, the parameters in (5) become

$$A_i = a_i a_i^T, \quad b_i = -a_i, \quad c_i = -\ell(\mathbf{C}_i)\big(\ell(\mathbf{C}_i) - 2\big), \qquad i = 1, \ldots, m.$$

Moreover, for 2-clauses, the inequality sign in (5) becomes equality.

3. Implementational issues

The semidefinite feasibility problem resulting from the gap relaxation (or Karloff–Zwick relaxation) may be solved using interior point methods. The most reliable way of detecting infeasibility using interior point methods is via the technique of self-dual embeddings. For a detailed discussion see De Klerk et al. [8,9]. The self-dual embedding approach is implemented in the software SeDuMi by Sturm [20], which we used to obtain the results presented in the next section.

The fact that the coefficient matrix in (5) has rank one for the gap relaxation can be exploited by interior point methods and in particular by the dual-scaling method by Benson et al. [4].

In order to give an indication of the computational times involved, we have listed some CPU times in table 1 for the computation of the gap relaxation and the related relaxation by Karloff and Zwick (K–Z) for $(2+p)$-SAT formulae with $p = 0.5$, $\alpha = 2.2$ and the numbers of variables (n) ranging from 100 to 300. Computation was done on a Pentium III (450 MHz) with SeDuMi 1.04 running under Matlab 5.3. The version of the implementation by Benson et al. [4] which we used could only handle rank one constraints, and was therefore not suitable to solve the K–Z relaxation.[3]

Note that the dual scaling method is faster on small formulae, but slower than SeDuMi on the larger ones. The reason is that the number of interior point iterations

[3] This method has now been extended to handle constraint matrices of all ranks (see [5]).

Table 1
Typical CPU times (seconds) for solving the gap relaxation and the relaxation by Karloff–Zwick (K–Z) using two different interior point solvers.

# vars.	Gap: SeDuMi	Gap: Benson et al.	K–Z: SeduMi
100	14.8	5.65	36.5
150	43.7	33.3	140
200	133	229	394
250	264	360	781
300	645	1088	1453

required by the dual scaling method grows quickly with the problem size (although the time per iteration remains low compared to SeDuMi).

In our own experience the dual scaling method of Benson et al. [4] therefore does not give superior performance on our feasibility problems at this time. However, at the time of writing significant progress has been made with these methods by using a conjugate gradient method to solve the Newton system at each iteration [7]. This approach was shown to be very promising for other combinatorial relaxations, like the MAX-CUT relaxation, where relaxations for problems with up to 14000 variables can be solved. Dual scaling methods therefore remain a promising alternative.

4. Computational results: detecting unsatisfiability for $(2+p)$-SAT

This section contains a selection of computational results we have obtained for random $(2+p)$-SAT formulae.

In particular, the following figures indicate the effectiveness of the relaxation in detecting infeasibility in random $(2+p)$-SAT formulae; the actual fraction of unsatisfiable formulae is plotted by a solid line as a function of the clause/variable ratio α for various values of p. The dashed line indicates the fraction where unsatisfiability is detected (where the gap relaxation is infeasible). Finally, the dash–dot line shows the fraction of formulae where the 2-SAT part is already unsatisfiable. At least one hundred random formulae were generated for each pair of values (α, p); if fewer than 20 of these formulae were unsatisfiable for a given pair of (α, p), then more formulae were generated until 20 unsatisfiable formulae had been found.

In figure 1 we consider $p = 0.3$ for formulae with 125 variables. Note that the gap relaxation is quite successful at detecting unsatisfiability in this case. The fraction of 3-clauses is increased to $1/2$ in figure 2. Note that the success rate of the relaxation now decreases, and this is even more noticeable in figure 3 where $p = 0.7$.

This trend is not surprising, since the relaxation approach is exact for 2-SAT yet always feasible for 3-SAT. One would therefore expect the performance to worsen as the fraction of 3-clauses increases. A positive aspect is that the fraction of unsatisfiable formulae where unsatisfiability is detected is still significantly higher than the fraction of unsatisfiable formulae where the 2-SAT part is already unsatisfiable.

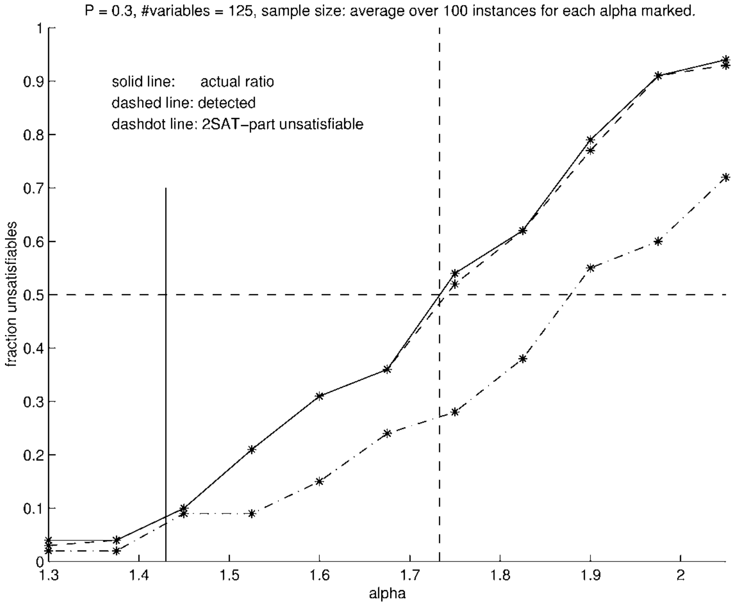

Figure 1. The fraction of formulae where unsatisfiability is detected as a function of the clause/variable ratio α for $p = 0.3$.

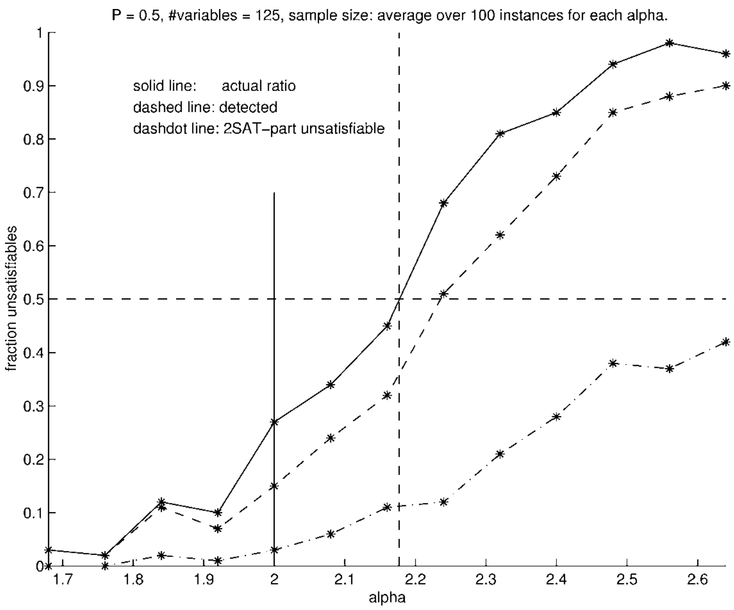

Figure 2. The fraction of formulae where unsatisfiability is detected as a function of the clause/variable ratio α for $p = 0.5$.

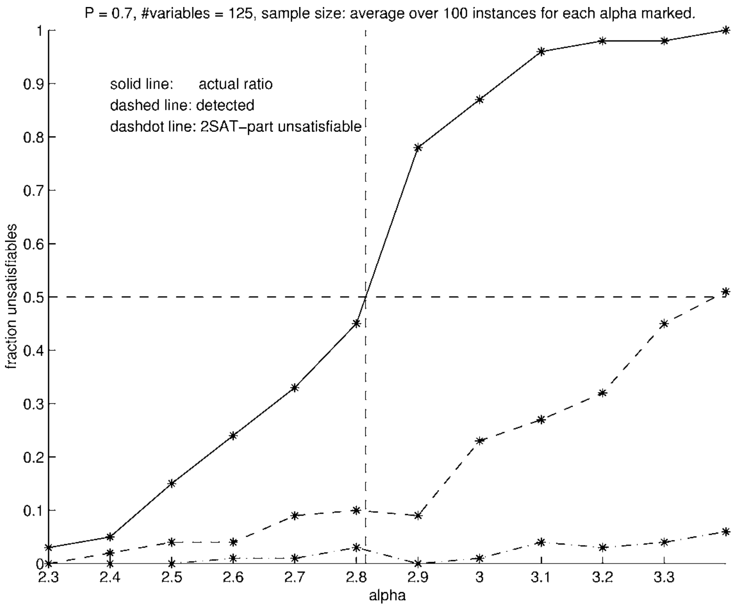

Figure 3. The fraction of formulae where unsatisfiability is detected as a function of the clause/variable ratio α for $p = 0.7$.

In figure 4 we plot the failure rate of the gap relaxation approach as a function of p. Here, we selected random formulae with 125 and 200 variables, respectively, such that the clause/variable ratio α is as close as possible to the empirically observed threshold value from [18].

The failure rate is defined as the number of unsatisfiable formulae where unsatisfiability is *not* detected as a fraction of the total number of unsatisfiable formulae. Note that one sees a threshold behavior – the effectiveness of the gap relaxation decreases rapidly after $p \approx 0.4$.

Finally, we show a histogram (figure 5) which indicates the fraction of failures of the relaxation as a function of α for formulae with 200 variables and $p = 0.5$. Note that the largest failure rates coincide with the empirically observed threshold value of $\alpha \approx 2.1$ [18]. Exact solution approaches for SAT usually experience difficulty with formulae where α is close to the empirically observed threshold value [18]. It is interesting to view the results in figure 5 in the light of some resent results on the 'threshold' behaviour of $(2+p)$-SAT. In [18,19], extensive experiments on the threshold behaviour of $(2+p)$-SAT formulas are carried out and the results explained through an analogy with properties of glassy or granular materials, as studied in statistical mechanics. It is claimed from this analogy that the average $(2+p)$-SAT formula behaves like a 2-SAT formula below a certain value of p (approximately 0.4) and like a 3-SAT formula above that value, with respect to computational costs (of resolution based methods used for solving) and with respect to the nature of the phase transition.

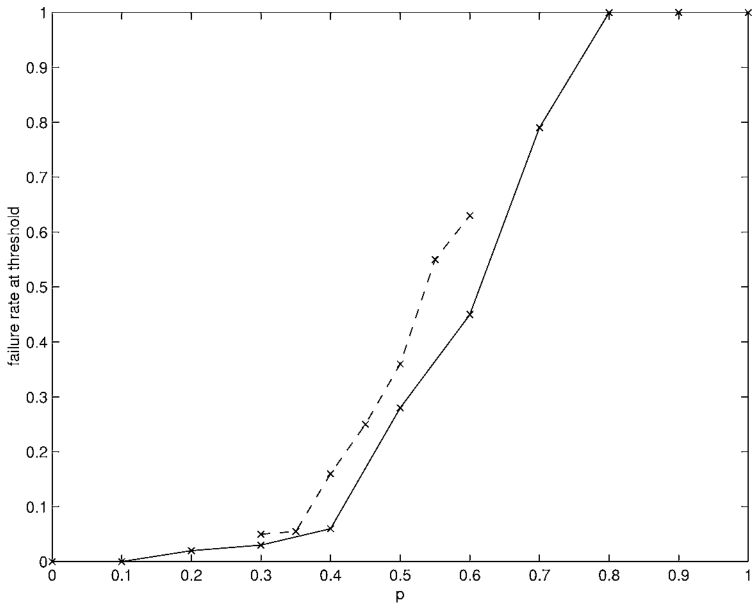

Figure 4. The failure rate of the gap relaxation approach as a function of p for random $(2+p)$-SAT formulae with clause/variable ratio near the threshold value. Solid line: 125 variables, dashed line: 200 variables.

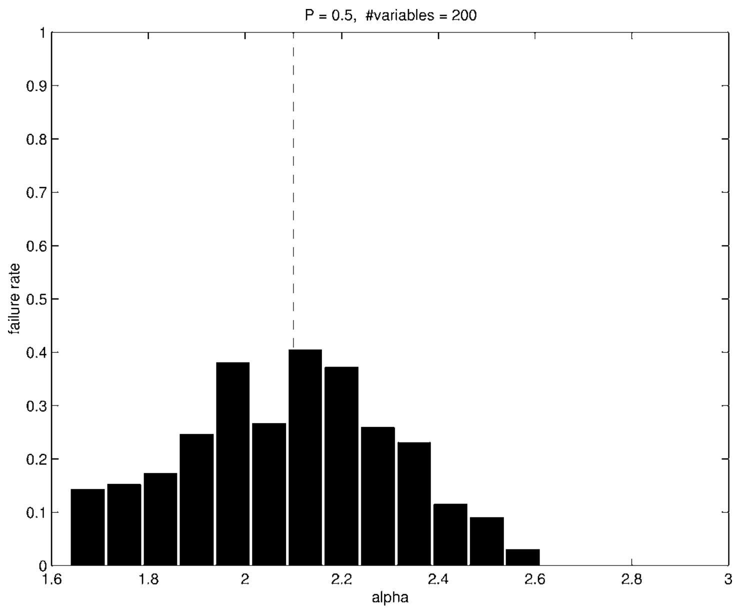

Figure 5. The fraction of failures (to detect unsatisfiability) as a function of the clause/variable ratio α for $p = 0.5$. The vertical dashed line marks the observed threshold value from [19] for this class of formulae.

More recently, Achlioptas et al. [1] have derived some rigorous bounds for the phase transition: they prove that for $p \leqslant 0.4$, asymptotically, with probability $1 - o(1)$, a random $(2 + p)$-SAT formula is satisfiable if and only if its 2-SAT part is satisfiable. Moreover, they prove that this situation no longer holds for some value $0.4 < p < 0.695$.

Our results (cf. figure 4), also show a ‘phase transition’: below 0.4 the gap relaxation method seems reliable, but above that value the fraction of failures rapidly increases. The results in [1,18,19] suggest that this behaviour is to be expected for large formulae. Experiments with large formulae are needed to confirm this, something which is not currently within the current computational possibilities.

5. Rounding procedures

If the gap relaxation of a given formula is feasible, then one may use a feasible solution of the relaxation in an attempt to generate a truth assignment for the formula (if one exists). Moreover, it is possible to establish performance guarantees for such ‘rounding procedures’.

5.1. *Deterministic rounding*

Recall that the vector y in (7) gives a truth assignment if X, y are feasible in (5) and X is a rank one matrix.

In general, X will not be a rank one solution, but one can still check whether the rounded vector $\mathrm{sign}(y)$ yields a truth assignment.

We will refer to this heuristic as *deterministic rounding*. It was shown in [10] that deterministic rounding satisfies each 2-clause, say $\mathbf{C} = p_i \vee p_j$, for which $y_i \neq 0$ or $y_j \neq 0$ in the solution of the gap relaxation. The unresolved 2-clauses can subsequently be satisfied in a trivial way (see [10] for details). In other words, deterministic rounding can always be used to satisfy a fraction $1 - p$ of the clauses (namely the 2-clauses).

5.2. *Randomized rounding with hyperplanes*

Recall that the entry X_{ij} in the matrix in (7) corresponds to the product $x_i x_j$ of logical variables. In fact, we may write

$$X_{ij} = v_i^T v_j,$$

where v_i and v_j are columns from the Choleski decomposition of the matrix in (7).

This shows that the product $x_i x_j$ is in fact relaxed to the inner product $v_i^T v_j$, that is, we associate a vector v_i with each literal p_i.

The vector y in (7) can similarly be seen as a vector of inner products:

$$y_i = v_T^T v_i,$$

where one can attach a special interpretation to the vector v_T as ‘truth’ vector: in a rank one solution, if $v_i = v_T$ then p_i is TRUE, and if $v_i = -v_T$ then p_i is FALSE.

This interpretation suggests a rounding scheme (which was introduced by Karloff and Zwick [15] as an extension of the ideas of Goemans and Williamson [11]):

1. Take the Choleski factorization of the matrix in (7).
2. Choose a random hyperplane through the origin.
3. If v_i lies on the same side of the hyperplane as v_T, then set p_i to TRUE; otherwise set p_i to FALSE.

We will refer to this procedure as *randomized rounding*; this heuristic can be derandomized using the techniques in [17].

6. Approximation guarantees for the rounding schemes

In this section we give a review of the analysis required to establish performance guarantees for the randomized rounding procedure. The relevant methodology is largely due to Karloff and Zwick [15]; we give a self-contained presentation of their approach here because we will use it to analyze a new type of rounding scheme in section 6.3.

6.1. Randomized rounding for 2-clauses

We again only consider the two clause $p_1 \vee p_2$ without loss of generality. Let the vectors v_1, v_2 be associated with the literals of $p_1 \vee p_2$. The randomized rounding procedure will fail to satisfy this clause if all three vectors $v_1, v_2, -v_T$ lie on the same side of the random hyperplane (see figure 6).

In general, we want to know what the probability is that a set of given vectors lie on the same side of a random hyperplane. The probability that two given unit vectors v_1, v_2 lie on the same side of a random hyperplane is easy: it only depends on the angle

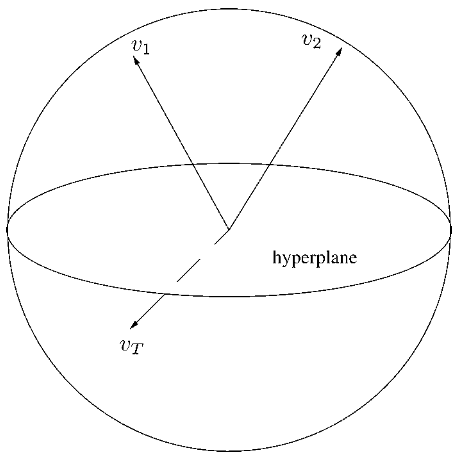

Figure 6. The situation where v_1 and v_2 are separated from v_T by a random hyperplane.

$\arccos(v_1^T v_2)$ between these vectors and is given by $1 - \arccos(v_1^T v_2)/\pi$ [11]. One can use this observation to treat the three vector case using inclusion–exclusion; this is done in [15]. We present a different derivation here which can be generalized to more vectors. The key is to consider the normal vector r to the randomized hyperplane. The clause will not be satisfied if the three vectors all have a positive (or all have a negative) inner product with r.

Note that the Gram matrix of $v_1, v_2, -v_T$ is the following matrix:

$$\bar{X}_2 := \begin{bmatrix} 1 & X_{12} & -y_1 \\ X_{12} & 1 & -y_2 \\ -y_1 & -y_2 & 1 \end{bmatrix},$$

and the gap relaxation requires

$$X_{12} - \sum_{i=1}^{2} y_i = -1. \tag{9}$$

The vectors $v_1, v_2, -v_T$ can be viewed as three points on the 2-dimensional unit sphere

$$\mathcal{S}^2 := \{x \in \mathbb{R}^3 \mid \|x\| = 1\},$$

and thus define a so-called *spherical triangle* (say $\mathbf{S}$) in the space $\mathcal{S}^2$.

The associated dual spherical triangle is defined as

$$\mathbf{S}^* := \{r \in \mathbb{R}^3\colon\ r^T v_i \geqslant 0 (i = 1, 2), r^T v_T \geqslant 0\}$$

which, together with $-1 \times \mathbf{S}^*$ form the set of normal vectors for which all three vectors lie on the same side of the associated plane.

The probability that the clause is not satisfied is therefore given by:

$$p^{(2)} = 2\frac{\text{area}(\mathbf{S}^*)}{\text{area}(\mathcal{S}^2)} = \frac{\text{area}(\mathbf{S}^*)}{2\pi}.$$

It is well known that the area of a spherical triangle is given by its *angular excess*[4] [2]. The dihedral angels of $\mathbf{S}^*$ are given in terms of the edge lengths of $\mathbf{S}$ (see, e.g., [2] and equal $(\pi - \arccos(X_{12}))$, $(\pi - \arccos(-y_1))$, and $(\pi - \arccos(-y_2))$. It follows that the angular excess (i.e., area) of $\mathbf{S}^*$ is given by:

$$\begin{aligned} \text{area}(\mathbf{S}^*) &= \big((\pi - \arccos(X_{12})) + (\pi - \arccos(-y_1)) + (\pi - \arccos(-y_2))\big) - \pi \\ &= 2\pi - \arccos(X_{12}) - \arccos(-y_1) - \arccos(-y_2), \end{aligned}$$

so that

$$p^{(2)} = 1 - \frac{1}{2\pi}\big(\arccos(X_{12}) + \arccos(-y_1) + \arccos(-y_2)\big). \tag{10}$$

[4] The angular excess is the difference between the sum of the (dihedral) angles of the spherical triangle and π.

We are therefore interested in the optimization problem:

$$\max_{\bar{X}_2} p^{(2)}$$

subject to $\bar{X}_2 \succeq 0$ and (9).

Since $p^{(2)}$ is a strongly quasi-concave function of $\bar{X}_2$, and the feasible region is convex, this optimization problem can therefore be solved to global optimality, because each local optimum is also global in this case (see theorem 3.5.9 in [3]). The optimal solution is given by

$$\bar{X}_2 = \begin{bmatrix} 1 & -1/3 & -1/3 \\ -1/3 & 1 & -1/3 \\ -1/3 & -1/3 & 1 \end{bmatrix},$$

in which case the clause is satisfied by randomized rounding with probability $3/(2\pi)\arccos(-1/3) \approx 0.91226$, by (10).

6.2. *Randomized rounding for* 3*-clauses*

This analysis is perfectly analogous to the analysis of the 2-clause case, the only complication being that the probability function is much more complicated.

Let the vectors v_1, v_2, v_3 be associated with the literals of $p_1 \vee p_2 \vee p_3$. As before, the randomized rounding procedure will fail to satisfy this clause if all four vectors $v_1, v_2, v_3, -v_T$ lie on the same side of the random hyperplane.

What is the probability of this event? This question has been answered by Karloff and Zwick [15]. Once again, we consider the normal vector r to the randomized hyperplane. The clause will not be satisfied if the four vectors all have a positive (or negative) inner product with r.

Note that the Gram matrix of $v_1, v_2, v_3, -v_T$ is the following matrix:

$$\bar{X}_3 := \begin{bmatrix} 1 & X_{12} & X_{13} & -y_1 \\ X_{12} & 1 & X_{23} & -y_2 \\ X_{13} & X_{23} & 1 & -y_3 \\ -y_1 & -y_2 & -y_3 & 1 \end{bmatrix}, \tag{11}$$

and that the gap relaxation requires

$$X_{12} + X_{13} + X_{23} - \sum_{i=1}^{3} y_i \leqslant 0. \tag{12}$$

The vectors $v_1, v_2, v_3, -v_T$ can be viewed as four points on the 3-dimensional unit hypersphere

$$\mathcal{S}^3 := \{x \in \mathbb{R}^4 \mid \|x\| = 1\},$$

and thus define a so-called *spherical tetrahedron* (say $\mathbf{S}$) in the space $\mathcal{S}^3$.

The associated dual spherical tetrahedron is defined as

$$\mathbf{S}^* := \left\{ r \in \mathbb{R}^4 \colon r^T v_i \geqslant 0 \ (i = 1, \ldots, 3), r^T v_T \geqslant 0 \right\}$$

which, together with $-\mathbf{S}^*$ form the set of normal vectors for which all four vectors lie on the same side of the hyperplane.

The probability that the clause is not satisfied is therefore given by:

$$p^{(3)} = 2 \frac{\text{volume}(\mathbf{S}^*)}{\text{volume}(\mathcal{S}^3)}.$$

The relative volume as a function of $\bar{X}_3$ is given by the following integral (see [2]):

$$\frac{\text{volume}(\mathbf{S}^*)}{\text{volume}(\mathcal{S}^3)} = \frac{1}{\sqrt{\det(\bar{X}_3)\pi^4}} \int_0^\infty \cdots \int_0^\infty \mathrm{e}^{-y^T \bar{X}_3^{-1} y} \mathrm{d}y_1 \cdots \mathrm{d}y_4.$$

In order to establish the worst-case performance for our relaxation we therefore have to solve the following optimization problem

$$\max_{\bar{X}_3} p^{(3)}$$

subject to $\bar{X}_3 \succeq 0$ and (12).

The volume function cannot be written in closed form, but can be simplified to a one dimensional integral [14] for spherical tetrahedra. Surprisingly, the gradient of the volume function is explicitly known (see [2,15]).

The optimization problem we consider has a convex feasible region but the objective function (to be maximized) is not concave. It is therefore difficult to find the global maximum, but it can be done due to the small problem size. We give only a sketch of the proof here.[5]

- First show that the global optimal solution is not positive definite. Assuming the contrary, the semidefiniteness constraints are redundant and only the single linear inequality (12) has to be considered. Since the gradient is known we can write down the first order (necessary) optimality conditions. Using these conditions we can show that the identity matrix is the only positive definite solution of the optimality conditions. If this stationary point would be globally optimal, then we would have an $7/8$ approximation guarantee. Unfortunately this is not the case.
- If the global optimal solution is only positive *semi*definite, then the volume calculation reduces to calculating a triangular area on a sphere or a circular arch length. Thus the analysis for a 3-clause reduces to that of 2-clause in this case.

[5] Since we have only one linear constraint, namely (12), the analysis is simpler than for the Karloff–Zwick relaxation (where there are three linear inequalities).

After following this procedure, we find that the optimal solution is given by the rank 2 matrix:

$$\bar{X}_3 = \begin{bmatrix} 1 & 1 & -1/2 & -1/2 \\ 1 & 1 & -1/2 & -1/2 \\ -1/2 & -1/2 & 1 & 1 \\ -1/2 & -1/2 & 1 & 1 \end{bmatrix},$$

in which case the clause is satisfied by randomized rounding with probability $(1/\pi)\arccos(-1/2) = 2/3$.[6]

Note, however, that the clause $p_1 \vee p_2 \vee p_3$ would have been satisfied by deterministic rounding of this solution. This suggests the idea of combining randomized and deterministic rounding in order to improve the approximation guarantee.

6.3. *Combined randomized and deterministic rounding*

We consider the worst case for randomized rounding of 3-clauses which are *not* satisfied by deterministic rounding. In other words, $y_i \leqslant 0$ $(i = 1, \ldots, 3)$.

This corresponds to

$$\bar{X}_3 = \begin{bmatrix} 1 & -0.75 & -0.75 & 0.5 \\ -0.75 & 1 & 1 & 0 \\ -0.75 & 1 & 1 & 0 \\ 0.5 & 0 & 0 & 1 \end{bmatrix},$$

in which case the clause is satisfied by randomized rounding with probability ≈ 0.8025.

Now the reasoning is as follows: suppose that deterministic rounding satisfies a fraction $1 - x$ of all the 3-clauses. Then the expected number of satisfied clauses via *subsequent* randomized rounding is:

$$0.8025x + 2/3(1 - x).$$

It now follows that if $x > 0.2934$ then it is worthwhile to do randomized rounding. In particular, we can always obtain an $1 - x > 0.7066$ approximation guarantee. This is a little better than the $2/3$ obtained if one only does randomized rounding.

We state this result as a theorem.

Theorem 6.1. Let an $(2 + p)$-SAT formula be given for which the gap relaxation is feasible. Either the randomized or the deterministic rounding scheme satisfies a fraction of at least

$$\max\{1 - p, 0.7066p + 0.91226(1 - p)\}$$

of the clauses.

---

[6] We wish to stress that it was not obvious a priori that the approximation guarantee for the gap relaxation would be strictly worse than $7/8$. Our analysis therefore shows that one cannot simplify the Karloff–Zwick relaxation to the gap relaxation and still retain a $7/8$ approximation algorithm.

Note that this guarantee is significantly worse than the (optimal) $7/8$ guarantee of Karloff and Zwick [15], showing that the gap relaxation is indeed weaker than the relaxation in [15].

7. Computational results for a combined procedure

We now show computational results where we do the following for a given $(2 + p)$-SAT formula.

1. Check if the gap relaxation is feasible; if not, then a certificate of unsatisfiability is obtained – STOP; else go to step 2.
2. Perform the rounding schemes described in the previous section. If a truth assignment is obtained, STOP; else go to step 3.
3. Check if the Karloff–Zwick relaxation is feasible; if not, then a certificate of unsatisfiability is obtained – STOP.

We wish to see how useful such a combined approach is. In particular, we wish to see how often the Karloff–Zwick relaxation detects unsatisfiability when the gap relaxation does not.

The results are presented as pie-charts in figures 7–11 for 200 random instances of $(2 + p)$-SAT with $p = 0.3, 0.4, 0.5, 0.6, 0.7$. The clause-variable ratio α is chosen near the threshold value for each value of p. As a consequence about half of the 200 formulae are unsatisfiable for each value of p. The number of variables n is 125. 'K–Z' stand for the Karloff–Zwick relaxation, and 'GAP' for the gap relaxation.

It is clear from the figures that both the relaxations and the rounding procedure are very reliable for $a = 0.3$ and $\alpha = 0.4$. Note that the K–Z relaxation is only slightly better than the gap relaxation at detecting unsatisfiability in these cases.

For $p = 0.5$ the picture changes. Already 12% of the formulae cannot be solved, and the K–Z relaxation detects unsatisfiability in 6% more formulae than the gap relaxation. For $p = 0.6$ and $p = 0.7$ these trends are further amplified: for $p = 0.7$ the

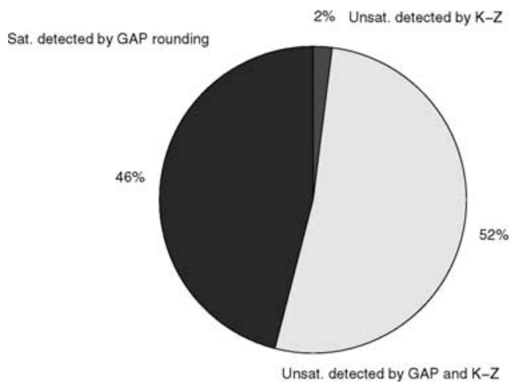

Figure 7. Computational results for $(2 + p)$-SAT formulae with $n = 125$, $p = 0.3$ and $\alpha = 1.74$.

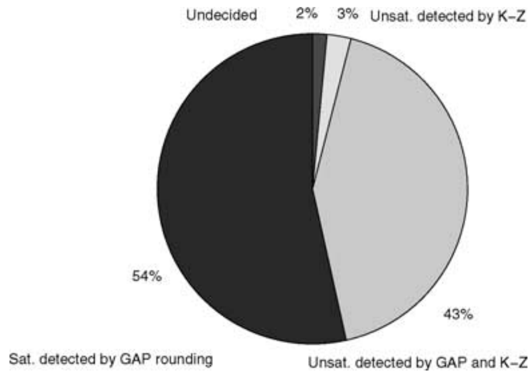

Figure 8. Computational results for $(2 + p)$-SAT formulae with $n = 125$, $p = 0.4$ and $\alpha = 1.94$.

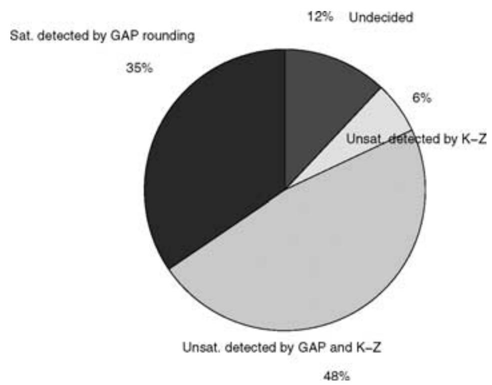

Figure 9. Computational results for $(2 + p)$-SAT formulae with $n = 125$, $p = 0.5$ and $\alpha = 2.18$.

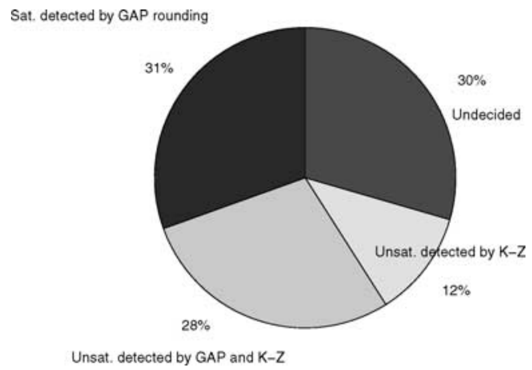

Figure 10. Computational results for $(2 + p)$-SAT formulae with $n = 125$, $p = 0.6$ and $\alpha = 2.47$.

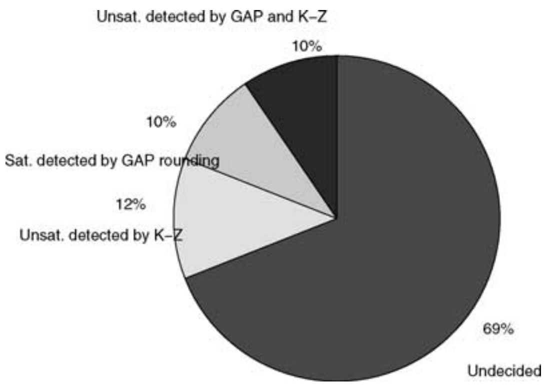

Figure 11. Computational results for $(2 + p)$-SAT formulae with $n = 125$, $p = 0.7$ and $\alpha = 2.82$.

K–Z approach detects unsatisfiability of an additional 12% of the formulae. However, in all fairness one should note that both the K–Z and gap approaches have a high failure rate for $p = 0.7$ (about half the formulae are unsatisfiable). Also, the rounding scheme becomes less efficient for larger p, as one may expect.

8. Conclusions and future research

In this paper we have studied the strength of the semidefinite programming relaxation of de Klerk et al. [10] on $(2 + p)$-SAT formulae, i.e., 3-SAT formulae with a fraction p of 3-clauses and a fraction $(1 - p)$ of 2-clauses. Thus a given logical formula is relaxed to a semidefinite feasibility problem. If the feasibility problem is infeasible, a certificate of unsatisfiability is obtained. The resulting semidefinite programming problem is computationally more attractive than the stronger relaxation by Karloff and Zwick [15] (see table 1). However, the latter method detects unsatisfiability for more instances than the gap relaxation does. This suggests a combined approach where cutting planes corresponding to the Karloff–Zwick relaxation are added to the gap relaxation, if necessary.

Numerical experiments on random instances show that the gap relaxation is quite useful in detecting unsatisfiability if the value of p is small, but becomes less so as p approaches unity. The approach also has a higher failure rate for formulae where the clause/variable ratio is close to the observed threshold values. For $p \leqslant 0.4$ the stronger relaxation of Karloff and Zwick does not perform significantly better. For $p \geqslant 0.5$ the latter method performs notably better than the gap relaxation, but then the failure rate is high for both approaches.

The gap relaxation can currently be solved using interior point methods for formulae with a few hundred variables and clauses. It is a topic for future research to better exploit sparsity in these methods; in particular, the potential use of dual scaling as opposed to primal–dual methods seems promising.

Two rounding heuristics were analysed for the case where the relaxation is feasible, namely the randomized rounding of Goemans–Williamson [11], and a deterministic rounding scheme. It was shown that – by using both heuristics and taking the best solution – a fraction of at least $\max\{1-p, 0.7066p+0.91226(1-p)\}$ clauses are satisfied (in expectation). This is worse than the $7/8$ rounding scheme for the relaxation by Karloff and Zwick [15], but numerical tests show that the rounding scheme analysed here is useful in practice for low values of p ($p \leqslant 0.4$).

Acknowledgements

The first author would like to thank Professor Uri Zwick for making his code for calculating the volume of spherical tetrahedrons available to him. The authors also thank Lucie Aarts, Vesna Balaban, and Sander van der Drift for performing many of the computational experiments presented in section 4. The authors also thank Steve Benson for his assistance with the dual scaling algorithm DSDP.

References

[1] D. Achlioptas, L.M. Kirousis, E. Kranakis and D. Krizank, Rigorous results for random $(2+p)$-SAT, Theoretical Computer Science 265(1–2) (2001) 109–129.

[2] D.V. Alekseevskij, E.B. Vinberg and A.S. Solodovnikov, Geometry of spaces of constant curvature, in: *Geometry II*, ed. E.B. Vinberg, Encyclopedia of Mathematical Sciences, Vol. 29 (Springer, Berlin, 1993).

[3] M.S. Bazarraa, H.D. Sherali and C.M. Shetty, *Nonlinear Programming: Theory and Algorithms*, 2nd edn. (Wiley, New York, 1993).

[4] S.J. Benson, Y. Ye and X. Zhang, Solving large-scale sparse semidefinite programs for combinatorial optimization, SIAM Journal on Optimization 10 (2000) 443–461.

[5] S.J. Benson and Y. Ye, DSDP3: Dual scaling algorithm for general positive semidefinite programming, Working paper, Computational Optimization Laboratory, Department of Management Science, University of Iowa, Iowa City (2001).

[6] W. Cook, C.R. Coullard and G. Turan, On the complexity of cutting plane proofs, Discrete Applied Mathematics 18 (1987) 25–38.

[7] C. Choi and Y. Ye, Solving sparse semidefinite programs using the dual scaling algorithm with an iterative solver, Working paper, Computational Optimization Laboratory, Department of Management Science, University of Iowa, Iowa City (2000).

[8] E. de Klerk, C. Roos and T. Terlaky, Initialization in semidefinite programming via a self-dual, skew-symmetric embedding, OR Letters 20 (1997) 213–221.

[9] E. de Klerk, C. Roos and T. Terlaky, Infeasible-start semidefinite programming algorithms via self-dual embeddings, in: *Topics in Semidefinite and Interior-Point Methods*, eds. P.M. Pardalos and H. Wolkowicz, Fields Institute Communications Series, Vol. 18 (American Mathematical Society, 1998) pp. 215–236.

[10] E. de Klerk, H. van Maaren and J.P. Warners, Relaxations of the satisfiability problem using semidefinite programming, Journal of Automated Reasoning 24 (2000) 37–65.

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

[12] A. Haken, The intractability of resolution, Theoretical Computer Science 39 (1985) 297–308.

[13] E. Halperin and U. Zwick, Approximation algorithms for MAX 4-SAT and rounding procedures for semidefinite programs, Journal of Algorithms 40 (2001) 184–211.

[14] W.-Y. Hsiang, On infinitesimal symmetrization and volume formula for spherical or hyperbolic tetrahedrons, The Quarterly Journal of Mathematics. Oxford 39(2) (1988) pp. 463–468.

[15] H. Karloff and U. Zwick, A 7/8-approximation algorithm for MAX 3SAT?, in: *Proc. 38th FOCS* (1997) 406–415.

[16] L. Lovász, On the Shannon capacity of a graph, IEEE Transactions on Information Theory 25 (1979) 1–7.

[17] S. Mahajan and H. Ramesh, Derandomizing semidefinite programming based approximation algorithms, in: *Proc. of the 36th Annual IEEE Symposium on Foundations of Computer Science* (1995) pp. 162–169.

[18] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman and L. Troyansky, Determining computational complexity from characteristic 'phase transitions', Nature 400 (1999) 133–137.

[19] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman and L. Troyansky, $2+P$-SAT: relation of typical-case complexity to the nature of the phase transition, Random Structure and Algorithms 15 (1999) 414–435.

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

[21] H. Van Maaren, Elliptic approximations of propositional formulae, Discrete Applied Mathematics 96–97 (1999) 223–244.

[22] J.P. Warners and H. Van Maaren, Recognition of tractable satisfiability problems through balanced polynomial representations, Discrete Applied Mathematics 99 (2000) 229–244.

[23] U. Zwick, Outward rotations: a new tool for rounding solutions of semidefinite programming relaxations, with applications to MAX CUT and other problems, in: *Proc. 31st STOC* (1999) pp. 679–687.