Relaxations of the satisfiability problem using semidefinite programming

de Klerk, E.; Warners, J.P.; van Maaren, H.

Published in:
Journal of Automated Reasoning

Publication date:
2000

Journal of Automated Reasoning **24:** 37–65, 2000.

Relaxations of the Satisfiability Problem Using Semidefinite Programming

ETIENNE DE KLERK and HANS VAN MAAREN
Department of Technical Mathematics and Informatics, 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

JOOST P. WARNERS⋆
Department of Technical Mathematics and Informatics, Faculty of Information Technology and Systems, Delft University of Technology, P.O. Box 5031, 2600 GA Delft, The Netherlands, and SEN2, CWI, P.O. Box 94079, 1090 GB, Amsterdam, The Netherlands. e-mail: j.p.warners@its.tudelft.nl

Abstract. We derive a semidefinite relaxation of the satisfiability (SAT) problem and discuss its strength. We give both the primal and dual formulation of the relaxation. The primal formulation is an eigenvalue optimization problem, while the dual formulation is a semidefinite feasibility problem. We show that using the relaxation, a proof of the unsatisfiability of the notorious pigeonhole and mutilated chessboard problems can be computed in polynomial time. As a byproduct we find a new 'sandwich' theorem that is similar to the sandwich theorem for Lovász' ϑ-function. Furthermore, the semidefinite relaxation gives a certificate of (un)satisfiability for 2SAT problems in polynomial time. By adding an objective function to the dual formulation, a specific class of polynomially solvable 3SAT instances can be identified. We conclude with discussing how the relaxation can be used to solve more general SAT problems and with some empirical observations.

Key words: satisfiability problem, relaxation, semidefinite programming.

1. Introduction

The satisfiability problem of propositional logic (SAT) is the original NP complete problem [5]. Many algorithms for SAT have been developed, both complete and incomplete; see for an overview [14]. Most of the incomplete algorithms are aimed at proving satisfiability; using some hill-climbing strategy a satisfying solution is sought, and if one is found the algorithm terminates; if no solutions are found, no definite answers about the (un)satisfiability of the formula can be given. In this paper we are concerned primarily with developing an (incomplete) algorithm for detecting *un*satisfiability. It is based on elliptic approximations of propositional formulas. Elliptic approximations of propositional formulas were first introduced

⋆ Supported by the Dutch Organization for Scientific Research (NWO) under grant SION 612-33-001.

by van Maaren [25, 24]. Since then they have been used to recognize polynomially solvable instances of satisfiability [35] and to derive effective branching rules and relative clause-weightings [26, 36]. In this paper we use elliptic approximations to derive sufficient conditions for unsatisfiability of a formula. These conditions can be expressed in terms of an *eigenvalue optimization* problem, which in turn can be cast as a *semidefinite program* (see, e.g., [34, 8]). Semidefinite programming is a generalization of linear programming and can be solved in polynomial time (to a given accuracy). Recently, much attention has been devoted to this field. Using semidefinite programming, efficient approximation algorithms to various hard combinatorial optimization problems have been developed, including the *maximum-satisfiability* problem [13, 12, 32, 20]. Using duality theory, we show that the dual of our formulation is a semidefinite feasibility problem, which is related to the formulation used by Karloff and Zwick for 3SAT formulas [20]. However, it is our aim to use it for proving unsatisfiability rather than finding approximate MAX-SAT solutions. We show that using the relaxation, a proof of the unsatisfiability of the notorious pigeonhole problems can be obtained in polynomial time, in a truly automated way, that is, without additional problem-specific information at all. As a byproduct, we obtain a new 'sandwich theorem' that is similar to Lovász' famous ϑ-function [22]. Furthermore, we show that it is complete for 2SAT formulas, and we indicate how it can be used to help solving 3SAT problems. In particular, a certain class of polynomially solvable 3SAT formulas can be recognized by adding an objective function to the dual formulation [28, 35].

This paper is organized as follows. In the next subsection we introduce some notation. In Section 1.2 we discuss some techniques for satisfiability solving based on a linear model, while the subsection thereafter is concerned with nonlinear approximations of satisfiability problems. In Section 3 we formulate a sufficient condition for unsatisfiability in terms of the eigenvalues of an appropriate matrix. Subsequently, we give a brief introduction to semidefinite programming and show that, using semidefinite programming, a certificate of unsatisfiability can be computed in polynomial time. The remainder of the paper is concerned with the investigation of the strength of the relaxation. Several subclasses of satisfiability problems are considered, namely, 2SAT problems, a class of covering problems (to which the pigeonhole and mutilated chessboard problems belong), and 3SAT problems. We conclude with some empirical observations and suggestions for further research. We provide geometrical interpretations, where useful, of the notions introduced.

1.1. THE SATISFIABILITY PROBLEM (SAT)

We consider the satisfiability problem in conjunctive normal form (CNF). A propositional formula Φ in CNF is a conjunction of clauses, where each clause $\mathbf{C}_k$ is a disjunction of *literals*. Each literal is an *atomic proposition* (or *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 $\mathbf{C}_k$ is of the form

$$\mathbf{C}_k = \bigvee_{i \in I_k} p_i \vee \bigvee_{j \in J_k} \neg p_j,$$

with $I_k,\ J_k \subseteq \{1, \ldots, n\}$ disjoint. The satisfiability problem of propositional logic is to determine whether or not an assignment of truth values to the variables exists such that each clause evaluates to true (i.e., one of its literals is true) and thus the formula is true.

1.2. SAT AND LINEAR PROGRAMMING

1.2.1. *An Integer Linear Programming Formulation of SAT*

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

$$\sum_{i \in I_k} x_i - \sum_{j \in J_k} x_j \geq 2 - \ell(\mathbf{C}_k),$$

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 (ILP) formulation of the satisfiability problem can be stated as

$$(\mathrm{IP}_{SAT}) \text{ find } x \in \{-1, 1\}^n \text{ such that } Ax \geq b.$$

The matrix $A \in \mathbb{R}^{m \times n}$ is called the *clause-variable* matrix. By a_k^T the kth row of A is denoted. 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, $b_k = 2 - \ell(\mathbf{C}_k)$.

For the reader unfamiliar with the concepts of mathematical programming, let us try to provide some geometrical insight. To this end, we introduce some additional notation:

$\mathcal{B}$: $\{-1, 1\}$ hypercube of dimension n;

$\mathcal{B}_V$: the vertices of $\mathcal{B}$ – the *assignments*;

$\mathcal{P}$: the polyhedron $\{x \in \mathbb{R}^n \mid Ax \geq b\}$;

$\mathcal{P}_0$: $\mathcal{P} \cap \mathcal{B}$;

$\mathcal{P}_V$: $\mathcal{P} \cap \mathcal{B}_V$ – i.e., the *satisfying assignments*.

As a simple two-dimensional illustration, see Figure 1. It represents the above sets for a satisfiable and an unsatisfiable formula, respectively. Note that these particular examples do not correspond to actual SAT instances, since in two dimensions no meaningful examples can be constructed; as stated before, we merely aim to provide some geometrical insight. The box represents the unit hypercube $\mathcal{B}$; its vertices $\mathcal{B}_V$ are highlighted. Each of the lines (hyperplanes) corresponds

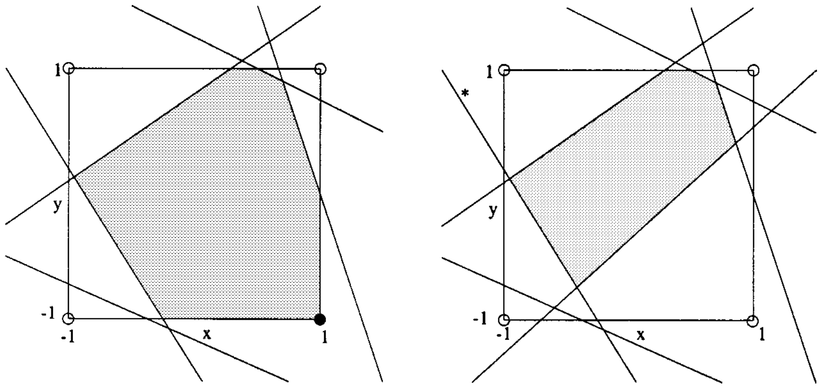

Figure 1. Illustration of LP approach to SAT solving.

to a linear constraint c.q. clause, expressing that any solution should lie either on or on a specified side of the hyperplane. The polyhedron $\mathcal{P}$ is defined by the intersection of half-spaces defined by the hyperplanes. The shaded area is the intersection of the unit hypercube and the polyhedron $\mathcal{P}$, i.e., $\mathcal{P}_0$. This is called the *feasible* region. The intersection of $\mathcal{P}_0$ and $\mathcal{B}_V$ yields $\mathcal{P}_V$ which is the set of satisfying assignments; it is highlighted by the black circle. In the left-hand example, $\mathcal{P}_V = \{(x = 1, y = -1)\}$, while in the right-hand example $\mathcal{P}_V = \emptyset$. Thus, solving ($\mathrm{IP}_{SAT}$) is equivalent to finding $x \in \mathcal{P}_V$ or showing that $\mathcal{P}_V$ is empty.

1.2.2. *Remarks on Solving* (IP_{SAT})

The most common approaches to solving integer linear programming problems include *branch-and-bound* and *branch-and-cut* algorithms. The integrality constraints are first relaxed to linear constraints (i.e., $x \in \{-1, 1\}$ is relaxed to $-1 \leq x \leq 1$), and subsequently the resulting *LP relaxation* is solved using the simplex method or an interior-point method (the latter of which runs in polynomial time). In other words, the problem of finding $x \in \mathcal{P}_0$ is solved. If the LP relaxation is infeasible (so $\mathcal{P}_0$ is empty), then the original ILP must also be infeasible, and thus we are done. Unfortunately, the LP relaxation may be feasible while the original ILP is infeasible (i.e., $\mathcal{P}_V$ is empty); this is illustrated by the right-hand picture of Figure 1. If the LP relaxation is feasible, usually one or a combination of the following techniques is applied:

1. *Branch*: a variable is chosen to be fixed at -1 or $+1$. This value is substituted in the LP relaxation and subsequently the (lower-dimensional) LP relaxation is solved. For example, if x is fixed at 1 in the right-hand side example of Figure 1, the LP relaxation of lower dimension is infeasible, since the intersection of $\mathcal{P}_0$ and the hyperplane $x = 1$ is empty; if, however, y is fixed at 1, the resulting LP relaxation is still feasible.

2. *Cut*: the LP relaxation is tightened by adding *cuts*, and again the LP relaxation is solved. A cut is a linear inequality that is redundant in the ILP formulation, but it cuts off a part of the feasible region of the LP relaxation (thus, denoting by $\mathcal{P}^*$ the tightened polytope, it holds that $\mathcal{P}^* \cap \mathcal{B} \subseteq \mathcal{P}_0$, while $\mathcal{P}^* \cap \mathcal{B}_V \equiv \mathcal{P}_V$; see also Section 6). As an example, consider the hyperplane marked with an asterisk in Figure 1; this is redundant for the ILP formulation because it does not cut off any vertices of the hypercube that are not already cut off by other hyperplanes. However, it does cut off a piece of the feasible region of the linear relaxation.

In general, procedures as sketched above will be effective only when the LP relaxations are *tight,* that is, when $\mathcal{P}_0$ is a good approximation of the *convex hull* of $\mathcal{P}_V$. The convex hull of $\mathcal{P}_V$ is the polytope that has only the elements of $\mathcal{P}_V$ as its vertices. Since LP algorithms provide a solution that is a vertex of the feasible region, the solution to the LP will be a satisfying assignment, or the LP relaxation will be infeasible, when $\mathcal{P}_0$ and the convex hull of $\mathcal{P}_V$ coincide. Unfortunately, the LP relaxation of (IP_{SAT}) is weak; it is easily checked that the trivial all-zero solution is always feasible when no unit clauses are present. Thus it is virtually useless for deciding *un*satisfiability; for finding satisfying assignments the formulation can sometimes be of use, since the solution to the LP relaxation could be either integer or rounded to a satisfying assignment. A specific kind of roundable solutions is the so-called linear autarky; see also Section 5.⋆

As mentioned previously, to better suit the LP approach toward proving unsatisfiability, the LP relaxation can be tightened by adding cuts. Without going into too many details, we mention that Hooker [17] shows that *resolution* can in fact be interpreted as the addition of cuts. Thus, it follows that by adding sufficiently many cuts, the feasible polytope becomes the convex hull of $\mathcal{P}_V$. Unfortunately, exponentially many cuts may be needed. Therefore, we are interested in deriving stronger relaxations of the satisfiability problem, using nonlinear rather than linear relaxations.

2. Elliptic Approximations of SAT Problems

2.1. A WEIGHTED ELLIPTIC APPROXIMATION

In [24] elliptic approximations for satisfiability problems are introduced. For each individual clause we define its *elliptic representation* by

$$\mathcal{E}_k = \{x \in \mathbb{R}^n \mid (a_k^T x - 1)^2 \leq (\ell(\mathbf{C}_k) - 1)^2\}. \tag{1}$$

A geometric interpretation is given in the next section. It is easy to check that any assignment $x \in \{-1, 1\}^m$ satisfying clause $\mathbf{C}_k$ lies in the interior or on the boundary of $\mathcal{E}_k$, since for such an assignment it holds that $b_k = 2 - \ell(\mathbf{C}_k) \leq a_k^T x \leq \ell(\mathbf{C}_k)$.

⋆ Note that an alternative LP relaxation, namely, the one based on polynomial representations of SAT problems, is capable of proving *un*satisfiability rather than satisfiability [37].

On the other hand, if x does not satisfy clause $\mathbf{C}_k$, it lies outside of $\mathcal{E}_k$ (since then $a_k^T x = -\ell(\mathbf{C}_k)$). Note that if $\ell(\mathbf{C}_k) \leq 2$, inequality in (1) may be replaced by equality.

Thus, basically the satisfiability problem can be expressed as finding a $\{-1, 1\}$ vector x lying in the intersection of m ellipsoids, that is,

$$x \in \bigcap_{k=1}^{m} \mathcal{E}_k \cap \{-1, 1\}^n \equiv \bigcap_{k=1}^{m} \mathcal{E}_k \cap \mathcal{B}_V.$$

However, it is hard to characterize the intersection of two or more ellipsoids explicitly; therefore we need to find another way to aggregate the information contained in the m separate ellipsoids. It appears plausible to take the sum over all these ellipsoids; this again yields an ellipsoid. Unfortunately, during summation the discriminatory properties of the separate ellipsoids are partly lost; therefore we speak of an *approximation* of a propositional formula. Rather than weighting each clause equally in the summation, let us associate a nonnegative weight w_k with each individual clause. Then any satisfying assignment $x \in \{-1, 1\}^n$ satisfies the following inequality:

$$\sum_{k=1}^{m} w_k (a_k^T x - 1)^2 \leq \sum_{k=1}^{m} w_k (\ell(\mathbf{C}_k) - 1)^2. \tag{2}$$

We define the *weighted elliptic approximation*:

$$\mathcal{E}(w) = \{x \in \mathbb{R}^n \mid x^T A^T W A x - 2w^T A x \leq r^T w\}, \tag{3}$$

where $w \in \mathbb{R}^m$, $W = \mathrm{diag}(w)$ (i.e., W is a diagonal matrix with the elements of the vector w on its diagonal) and $r_k = \ell(\mathbf{C}_k)(\ell(\mathbf{C}_k) - 2)$. We have the following theorem.

THEOREM 2.1. *Let Φ be a CNF formula with associated clause-variable matrix A and weighted elliptic approximation $\mathcal{E}(w)$. If $x \in \{-1, 1\}^n$ is a satisfying assignment of Φ, then $x \in \mathcal{E}(w)$ for any $w \geq 0$.*

Proof. The proof follows by expanding (2) to

$$\sum_{k=1}^{m} w_k (a_k^T x)^2 - 2 \sum_{k=1}^{m} w_k a_k^T x \leq \sum_{k=1}^{m} w_k \ell(\mathbf{C}_k)(\ell(\mathbf{C}_k) - 2),$$

which is equivalent to $x^T A^T W A x - 2w^T A x \leq r^T w$. Considering (3), we conclude that $x \in \mathcal{E}(w)$. □

2.2. ELLIPSOIDS AND EIGENVALUES

For the reader who is uncomfortable with quadratic forms such as discussed in the preceding section, let us now briefly consider some linear algebra to explain the name *elliptic* approximation and to provide some more insight in the structure of such quadratic forms. For a rigorous introduction to this field, see, for example, Strang [30].

Let Q be a real symmetric $n \times n$ matrix. Q has a so-called spectral decomposition $Q = S\Lambda S^T$, where Λ is a diagonal matrix containing the *eigenvalues* $\lambda_{\min} = \lambda_1 \leq \cdots \leq \lambda_n = \lambda_{\max}$ of Q, and the columns of S constitute an orthonormal base of corresponding eigenvectors. It holds that $S^T S = I$ (I denotes the identity matrix of appropriate dimension) and $Qs_i = \lambda_i s_i$, for any eigenvalue λ_i and corresponding eigenvector s_i. If all eigenvalues are strictly positive, Q is called *positive definite*. If all eigenvalues are nonnegative, Q is called positive *semi*definite. Equivalent definitions of positive semidefiniteness are that (i) $x^T Qx \geq 0$ for any x, and (ii) there exists a matrix R such that $Q = R^T R$.

Now consider the quadratic form

$$\mathcal{Q}(x) = x^T Qx - 2q^T x,$$

where $q \in \mathbb{R}^n$ and let us assume that Q is positive definite. Letting $z = S^T x$, we can rewrite $x^T Qx$ as follows:

$$x^T Qx = x^T S\Lambda S^T x = z^T \Lambda z = \sum_{i=1}^{n} \lambda_i z_i^2. \tag{4}$$

Considering (4), it is easy to see that the region $\{x \in \mathbb{R}^n \mid \mathcal{Q}(x) \leq \rho^2\}$ is a *bounded elliptic region* (i.e., it is an n-dimensional ellipsoid), which is nonempty if $\rho^2 \geq -q^T Q^{-1} q$. The family of ellipsoids associated with $\mathcal{Q}(x)$ is centered at $x = Q^{-1}q$, and its symmetry axes are parallel to the eigenvectors. The lengths of the axes are equal to $\rho/\sqrt{\lambda_i}$. This implies that the *longest* axis is the one parallel to the eigenvector corresponding to the *smallest* eigenvalue.

The elliptic approximation as introduced in the preceding section is not necessarily bounded, since we can guarantee only that the matrix $A^T WA$ is positive *semi*definite. This easily follows by noting that $A^T WA = R^T R$, where $R = W^{1/2}A$. Therefore, $x^T R^T Rx = \|Rx\|^2 \geq 0$. In [36] a proof is given that a formula is either easily satisfiable or allows a bounded elliptic approximation.

Let us finish this section by deriving a well-known and useful lower bound on the value of $x^T Qx$ over the vertices of the hypercube $\mathcal{B}_V$. Such a bound can easily be deduced from (4). Note that $x^T x = n$ for any $x \in \{-1, 1\}^n$ and also that $z^T z = x^T x = n$. Thus,

$$x^T Qx \geq \lambda_{\min} n, \quad \text{for any } x \in \{-1, 1\}^n. \tag{5}$$

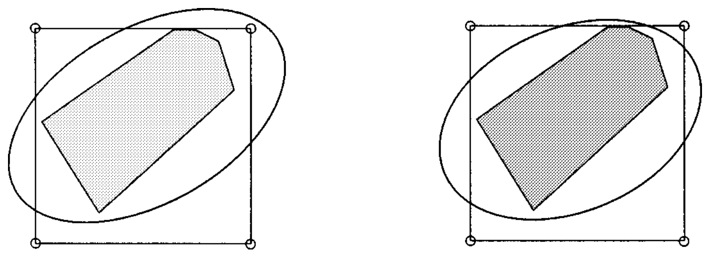

Figure 2. Elliptic approximations of the feasible polytope.

2.3. GEOMETRICAL INTERPRETATION OF ELLIPTIC APPROXIMATIONS

Let us now consider the geometrical interpretation of the elliptic approximations. As indicated in Figure 1 the polytope $\mathcal{P}_0$ strictly separates between the satisfying and contradictory assignments; that is, only satisfying assignments are contained in it. The elliptic approximation $\mathcal{E}(w)$ *circumscribes* $\mathcal{P}_0$. By adjusting the weights, its shape changes, but for any choice of weights $\mathcal{P}_0 \subset \mathcal{E}(w)$. Obviously, we are interested in finding a choice of weights such that $\mathcal{E}(w)$ is as tight as possible an approximation of $\mathcal{P}_0$. In particular, if $\mathcal{P}_V = \emptyset$ (so the formula under consideration is unsatisfiable), we want to find a set of weights such that $\mathcal{E}(w) \cap \mathcal{B}_V = \emptyset$ as well. See Figure 2. The elliptic approximation on the left-hand side is not tight, since a (contradictory) assignment is contained in it. The approximation on the right-hand side is tight.

In the next section we derive a condition on the weights for unsatisfiable formulas, under which the associated weighted elliptic approximation is tight.

3. Relating Unsatisfiability to Eigenvalues

3.1. A SUFFICIENT CONDITION FOR UNSATISFIABILITY

Let us again consider (3). By Theorem 2.1, for any satisfying assignment $x \in \{-1, 1\}^n$ it must hold that

$$x^T A^T W A x - 2w^T A x - r^T w \leq 0, \tag{6}$$

for all $w \geq 0$. Thus, we have a necessary condition for satisfiability. Reversing this argument gives a sufficient condition for unsatisfiability.

COROLLARY 3.1. *Let Φ be a CNF formula. If for some $w \geq 0$ it holds that $x \notin \mathcal{E}(w)$ for all $x \in \{-1, 1\}^n$, then Φ is unsatisfiable.*

Relaxing the integrality constraint to a spherical constraint, we obtain an alternative sufficient condition for unsatisfiability.

COROLLARY 3.2. *Let Φ be a CNF formula. If for some $w \geq 0$ it holds that $x \notin \mathcal{E}(w)$ for all x such that $x^T x = n$, then Φ is unsatisfiable.*

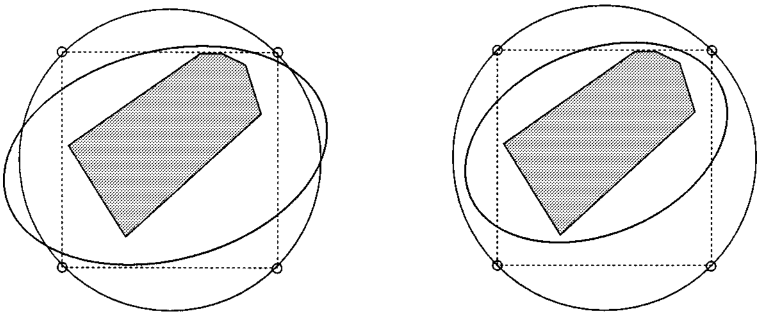

Figure 3. Illustration of Corollaries 3.1 and 3.2. On the left unsatisfiability is proved by Corollary 3.1; on the right unsatisfiability is proved by both corollaries.

Observe that the condition of the second corollary is weaker than that of the first, since $x \in \{-1, 1\}^n$ implies $x^T x = n$, but not the other way round.

Geometrically, we can interpret Corollary 3.2 as follows. Rather than trying to prove that a choice of weights exists such that no vertices of the unit hypercube are contained in the weighted elliptic approximation (this is still a difficult problem), our aim is to find weights such that the intersection of the *unit sphere* and the weighted elliptic approximation is empty; see Figure 3.

Note that the condition stated in Corollary 3.2 appears to be closely related to the eigenvalues of the matrix $A^T WA$ involved in the weighted elliptic approximation. Assuming that the ellipsoid is centered at the origin, a choice of weights is required such that the minimal eigenvalue is as large as possible, since the minimal eigenvalue is related to the length of the longest axis of the ellipsoid. In the following, we will see that this is indeed a fruitful approach. We first illustrate its usefulness with an example.

EXAMPLE. We consider the well-known *pigeonhole* formulas, which can be stated as follows:

Given $h+1$ *pigeons and* h *holes, decide whether it is possible to put each pigeon in at least one hole, while no two pigeons may be put in the same hole*.

For more details on this problem and its SAT encoding see Section 6. The set of clauses in such a formula can be divided into a set of *long* clauses and a set of *short* clauses. It can be shown that for the instance with h holes and $h+1$ pigeons, when all long clauses are given a weight of one, and all short clauses a weight of $h - 1 + \frac{2}{h+1}$, the minimal value of (6) over the sphere $x^T x = n$ is equal to $4(h - 1 + \frac{1}{h+1}) > 0$. These values can be explicitly computed by using the specific structure of the eigenspace of the matrix $A^T WA$ associated with the pigeonhole formulas. Thus by Corollary 3.2 it follows that the pigeonhole formulas are unsatisfiable. We conclude that, even by using only a low dimensional weight vector, pigeonhole formulas can be shown to be contradictory.

3.2. A CERTIFICATE OF UNSATISFIABILITY BASED ON EIGENVALUES

In this section we show how to reformulate Corollary 3.2 to obtain a condition for unsatisfiability in terms of eigenvalues. To this end we rewrite (6). Introducing an additional Boolean variable $x_{n+1} \in \{-1, 1\}$, we obtain the inequality

$$x^T A^T WAx - 2x_{n+1} w^T Ax - r^T w \leq 0. \tag{7}$$

Once again, if Φ is satisfiable, then inequality (7) is satisfied by some $\{x_1, \ldots, x_{n+1}\} \in \{-1, 1\}^{n+1}$ for all $w \geq 0$.

We can rewrite condition (7) as $\tilde{x}^T Q(w)\tilde{x} \leq 0$, where $\tilde{x} := [x_1, \ldots, x_n, x_{n+1}]$ and $Q(w)$ is the $(n+1) \times (n+1)$ matrix:

$$Q(w) := \begin{bmatrix} A^T WA - \frac{r^T w}{n} I & -A^T w \\ -w^T A & 0 \end{bmatrix}.$$

This is valid because $x \in \{-1, 1\}^n$ implies that $x^T(\frac{r^T w}{n} I)x = r^T w$. By this observation, it follows that we can further add a so-called correcting vector $u \in \mathbb{R}^n$ to $Q(w)$ to obtain

$$\tilde{Q}(u, w) := \begin{bmatrix} A^T WA - \frac{r^T w}{n} I - \mathrm{diag}(u) & -A^T w \\ -w^T A & e^T u \end{bmatrix}.$$

By e we denote the all-one vector; it is easy to verify that $\tilde{x}^T Q(w)\tilde{x} = \tilde{x}^T \tilde{Q}(u, w)\tilde{x}$.

Note that the family of ellipsoids defined by $\tilde{Q}(u, w)$ is in $(n+1)$-dimensional space, that is, a dimension higher than that of the feasible polytope $\mathcal{P}_0$. Because of the addition of the correcting vector, the ellipsoids in lower-dimensional space no longer necessarily circumscribe $\mathcal{P}_0$. Thus, for appropriate choices of u and w, the elliptic approximation defined by $\tilde{Q}(u, w)$ can be a *tighter* approximation of $\mathcal{P}_V$ than $\mathcal{P}_0$ is.

By Corollary 3.1 we are interested in minimizing $\tilde{x}^T \tilde{Q}(u, w)\tilde{x}$ over the $(n+1)$-dimensional $\{-1, 1\}$-vectors. Following Corollary 3.2, we relax the integrality constraint to a single spherical constraint $\tilde{x}^T \tilde{x} = n + 1$. In particular, if the optimal value to this problem is positive for some $w \geq 0$ and u, then Φ cannot be satisfiable. Thus we are facing the problem of finding choices of u and w such that the minimal value of $\tilde{x}^T \tilde{Q}(u, w)\tilde{x}$ over the unit sphere $\tilde{x}^T \tilde{x} = n + 1$ is positive. By (5) this is equivalent to finding a pair $(w \geq 0, u)$ such that the minimal eigenvalue of $\tilde{Q}(u, w)$ is positive.

DEFINITION 3.1. A pair $(w \geq 0, u)$ is called a (u, w)-*certificate of unsatisfiability* if the minimal eigenvalue of $\tilde{Q}(u, w)$ is positive.

Note that given a (u, w)-certificate of unsatisfiability, its validity can be verified in polynomial time by computing the minimal eigenvalue of $\tilde{Q}(u, w)$.

In the next section, we show that an approach to find a (u, w)-certificate of unsatisfiability can be cast in terms of a *semidefinite programming* problem.

4. A Semidefinite Relaxation of the SAT Problem

4.1. AN INTRODUCTION TO SEMIDEFINITE PROGRAMMING

Recently, much attention has been devoted to the field of semidefinite programming. It was shown that efficient approximation algorithms for hard combinatorial optimization problems can be obtained using semidefinite relaxations [13, 1], while there are also applications in control theory [34]. With interior-point methods, semidefinite programs can be solved (to a given accuracy) in polynomial time. For the reader who is unfamiliar with semidefinite programming, we review some of the basic concepts.

The standard primal (SP) and dual (SD) semidefinite programming formulations can be denoted by (see, e.g., [8])⋆

$$\text{(SP)}\quad \begin{array}{l} p^* = \inf \ \mathbf{Tr}\, CX \\ \quad \text{s.t. } \mathbf{Tr}\, A_k X = b_k, \\ \qquad X \succeq 0. \end{array} \qquad \text{(SD)}\quad \begin{array}{l} d^* = \sup b^T y \\ \quad \text{s.t. } \displaystyle\sum_{k=1}^m y_k A_k + S = C, \\ \qquad S \succeq 0. \end{array}$$

In the above programs, the A_k, C, X, and S are symmetric real $(n \times n)$-matrices and b and y are m-vectors. The matrix X denotes the primal decision variables, while (S, y) are the dual decision variables; S is also called a *slack*-variable. The constraint $X \succeq 0$ (resp. $S \succeq 0$) indicates that X (resp. S) must be positive semidefinite. Positive semidefiniteness of a matrix can be characterized in several ways; see Section 2.2. **Tr** denotes the *trace*-operator. The trace of a matrix A is equal to the sum of its diagonal elements. For products of matrices, it holds that

$$\mathbf{Tr}\, AB = \sum_{i=1}^n \sum_{j=1}^n A_{ij} B_{ij},$$

where $A, B \in \mathbb{R}^{n\times n}$. A useful easy-to-check property of the trace operator is $\mathbf{Tr}\, AB = \mathbf{Tr}\, BA$. Also, the trace of a matrix is equal to the sum of its eigenvalues.

When all the data matrices involved in the pair (SP, SD) are diagonal matrices, the semidefinite programming problem reduces to a linear programming problem. Note that the (nonlinear) constraints $X \succeq 0$, $S \succeq 0$, then reduce to nonnegativity constraints.

The duality theory for semidefinite programming is similar to – but slightly weaker than – the duality theory of linear programming. For the pair (SP, SD) weak duality holds; that is, $b^T y \leq \mathbf{Tr}\, CX$ if X is feasible for (SP) and y is feasible for (SD). As in linear programming, we call the nonnegative quantity $\mathbf{Tr}\, CX - b^T y$ the *duality gap*. It is easy to show that the duality gap equals $\mathbf{Tr}\, XS$ for feasible X, S.

⋆ Note that the matrices in this section denote general matrices that are not necessarily related to the clause-variable matrix A introduced before.

We say that (SP, SD) are in perfect duality if $p^* = d^*$, where we use the convention that $p^* = \infty$ if (SP) is infeasible, $p^* = -\infty$ if (SP) is unbounded, and so forth. Perfect duality is guaranteed if one of (SP) and (SD) is *strictly feasible*; (SP) (resp. (SD)) is strictly feasible if a strictly interior solution $X \succ 0$ (resp. $S \succ 0$) exists. Infeasibility of one then implies unboundedness of the other. Note that perfect duality always holds in linear programming; in the semidefinite programming case pathological duality effects can occur where, for example, (SP) is infeasible but (SD) has a finite optimal value (i.e., $d^* < \infty$).

If *both* (SP) and (SD) are strictly feasible, then optimal solutions (X, S) exist for (SP, SD) with duality gap zero (i.e., $\mathbf{Tr}\, XS = 0$). Such solutions are called *complementary*, since $\mathbf{Tr}\, XS = 0$ is equivalent to $XS = 0$ for positive semidefinite matrices. In linear programming, all solutions to the primal and dual problems are complementary, but for semidefinite programming (SP) and (SD) may have optimal solutions with positive duality gap, if strict feasibility does not hold. We speak of *strict infeasibility* of (SP) if there exists an *improving direction* for (SD), and vice versa. An improving direction for (SD) is defined as a pair $(\Delta y, \Delta S)$ such that $\sum_{k=1}^{m} \Delta y_k A_k + \Delta S = 0$ and $b^T \Delta y \geq 1$. The objective function of (SD) can therefore be increased indefinitely along the direction $(\Delta y, \Delta S)$, which means that (SD) must be unbounded if it is feasible. Improving directions for (SP) are defined similarly. Once again, strict infeasibility is the only kind of infeasibility that can occur in linear programming, but in semidefinite programming *weak infeasibility* is also possible. The semidefinite programs we consider in this paper cannot give rise to these pathological duality effects, and the duality relations used here will be no more complicated than in the linear programming case. For more details on semidefinite programming duality issues we refer to [8].

4.2. AN EIGENVALUE OPTIMIZATION PROBLEM

The problem of finding a (u, w)-certificate of unsatisfiability can be expressed in terms of an eigenvalue optimization problem. Note that, if μ is an eigenvalue of some matrix Q, then $\mu - \lambda$ is an eigenvalue of the matrix $Q - \lambda I$. This can be seen as follows: let s be the corresponding eigenvector, then $Qs = \mu s$ and $(Q - \lambda I)s = (\mu - \lambda)s$. Thus a valid approach for finding a (u, w)-certificate of unsatisfiability is to solve the following optimization problem.

$$\begin{aligned} &\sup (n+1)\lambda \\ \text{(P) s.t. } &\tilde{Q}(u, w) \succeq \lambda I, \\ &w \geq 0. \end{aligned}$$

If $\tilde{Q}(u, w) \succeq \lambda I$, then $\tilde{Q}(u, w) - \lambda I \succeq 0$. Thus it follows that $(w \geq 0, u)$ is a (u, w)-certificate of unsatisfiability, if (and only if) $\lambda > 0$. We call the optimal value of optimization problem (P) the *gap* of formula Φ (not to be confused with *duality gap*).

DEFINITION 4.1. The gap of a formula Φ is defined as the optimal value of the optimization problem (P).

$$\text{gap}(\Phi) := \sup_{w\geq 0,u} (n+1)\lambda_{\min}\big(\tilde{Q}(u,w)\big).$$

Thus, by Corollary 3.2, we have the following corollary.

COROLLARY 4.1. *If a formula Φ has a positive gap, it has a (u,w)-certificate of unsatisfiability and therefore Φ is a contradiction.*

We will show that the converse is also true if Φ is a 2-*SAT* formula. Furthermore, the formulas corresponding to a specific type of covering problems, which include the notorious pigeonhole problems and mutilated chessboards, have a positive gap. We emphasize that having a (u,w)-certificate of unsatisfiability is still merely a sufficient condition for unsatisfiability.

4.3. THE DUAL RELAXATION

Considering the eigenvalue optimization problem (P), we note that it fits the format (SD) (see Section 4.1). Thus, we can obtain the dual of the optimization problem (P) via the primal-dual pair (SP, SD). It can be simplified to the following *semidefinite feasibility problem*:

$$\text{(D)}\quad \begin{array}{ll} \text{find} & Y\in\mathbb{R}^{n\times n},\ y\in\mathbb{R}^n \\ \text{s.t.} & a_k^TYa_k - 2a_k^Ty \leq r_k,\ \ 1\leq k\leq m, \\ & \text{diag}(Y)=e, \\ & Y\succeq yy^T. \end{array}$$

Since (D) does not have an objective function, we adopt the convention that its optimal objective value is zero if a feasible solution exists, while it is $+\infty$ if (D) is infeasible.

To see that (D) is indeed the dual of (P), note that the constraint on the diagonal of Y follows by dualizing the correcting vector, while the first set of constraints is obtained by rewriting the condition

$$\mathbf{Tr}\begin{bmatrix} a_ka_k^T-\frac{r_k}{n}I & -a_k \\ -a_k^T & 0\end{bmatrix}\begin{bmatrix} Y & y \\ y^T & 1\end{bmatrix}\leq 0,$$

and using that $\text{diag}(Y)=e$. The last constraint of (P),

$$Y\succeq yy^T, \tag{8}$$

is called the *semidefinite constraint*. It follows using the well-known *Schur complement* reformulation (see, e.g., [18])

$$\begin{bmatrix} Y & y \\ y^T & 1 \end{bmatrix} \succeq 0 \Leftrightarrow Y - yy^T \succeq 0.$$

When a formula Φ is satisfiable with an assignment x, the solution $Y = xx^T,\ y = x$ is feasible in its associated dual relaxation (D); note that then

$$a_k^T Y a_k - 2a_k^T y = (a_k^T x)^2 - 2a_k^T x \le r_k,$$

which is exactly the elliptic representation of clause $\mathbf{C}_k$ (see (1)). Furthermore, $x \in \{-1, 1\}^n$, hence $\operatorname{diag}(Y) = e$, while (8) is satisfied as well.

Rather than by dualizing (P), (D) can be derived directly from the elliptic approximations of clauses in the following way. Expanding (1), it can be written as $a_k^T xx^T a_k - 2a_k^T x \le r_k$. This equation can be linearized by replacing the variables x_i by *vectors* $v_i \in \mathbb{R}^{n+1}$ with the additional requirement that $\|v_i\| = 1$, adding a homogenizing vector v_0, and letting $Y = V^T V$ and $y = V^T v_0$ (here V is the matrix with the vectors v_i as columns). The constraint on the diagonal of Y follows immediately, and the semidefinite constraint follows using that $[V\ v_0]^T[V\ v_0] \succeq 0$ and the Schur complement. If Φ allows the satisfying assignment x, a feasible solution in terms of V is constructed by setting all entries of the vectors v_i to zero, except the first entries: the first entry of the vectors v_i $(1 \le i \le n)$ is set to x_i, while the first entry of v_0 is set to one.

Finally, note that (D) is closely related to the MAX3SAT relaxation proposed by Karloff and Zwick [20]. It appears to be slightly weaker, since in their formulation the constraints are further disaggregated; see also Section 8. In addition, we are interested only in proving unsatisfiability, rather than in MAX-SAT solutions.

4.4. SOME PROPERTIES OF THE GAP

Let us now consider (P) and (D) to derive a number of properties of the gap. First note that (P) is strictly feasible, since there exist $w \ge 0$, u and λ such that the slack $\tilde{Q}(u, w) - \lambda I$ is positive definite. Thus, *perfect duality* holds, implying that unboundedness of (P) implies infeasibility of (D).

COROLLARY 4.2. *For any formula* Φ, $\operatorname{gap}(\Phi)$ *is either zero or infinity. If* $\operatorname{gap}(\Phi) = \infty$, *then there exists a* (u, w)*-certificate of unsatisfiability, implying the unsatisfiability of* Φ.

We have an even stronger duality result.

LEMMA 4.1. *For the primal-dual pair* (P, D) *exactly one of the following two duality relations holds:*

(1) (D) *is feasible and* (P, D) *have complementary optimal solutions;*

(2) (D) *is strictly infeasible and* (P) *is unbounded.*

Proof. (1) Suppose (Y, y) is a feasible solution of (D). The all-zero solution $u = 0$, $w = 0$, $\lambda = 0$ is feasible for (P) and since the slack is then all-zero as well, this constitutes a complementary (and therefore optimal) solution. Thus, strong duality holds.

(2) If (D) is infeasible, we conclude from the perfect duality of (P, D) that (P) is unbounded. Thus, it allows a solution $(w \geq 0, u)$ such that $\lambda > 0$. Using this solution, an improving direction for the objective function of (P) we can be construct, since $\lambda_{\min}(\tilde{Q}(\alpha u, \alpha w)) = \alpha\lambda_{\min}(\tilde{Q}(u, w))$. This implies that (D) is *strictly* infeasible (see Section 4.1). □

Since either complementary solutions exist for (P) and (D), or (D) is strictly infeasible, we have the following corollary (for a proof, see [8]).

COROLLARY 4.3. *Using semidefinite programming, one can decide in polynomial time which of the two duality relations holds. Thus, the existence of a* (u, w)*-certificate of unsatisfiability can be established in polynomial time.*

Finally, we have a result on monotonicity of the gap.

LEMMA 4.2. *Let* Φ *be a CNF-formula, and let* $\Psi \subseteq \Phi$*. Then it holds that* $\text{gap}(\Psi) \leq \text{gap}(\Phi)$.

Proof. Consider (P). Add to (P) the constraints that $w_k = 0$ for all clauses $\mathbf{C}_k$ that occur only in Φ. Solving this modified version of (P), $\text{gap}(\Psi)$ is obtained. Obviously, it is a more restricted version of $\text{gap}(\Phi)$, so the lemma follows. □

This leads immediately to the following corollary.

COROLLARY 4.4. *The gap is monotone under unit resolution.*

Proof. Unit resolution can be regarded as the addition of unit clauses to a formula. By the previous lemma, the gap cannot decrease. □

In the next sections the gap for some specific SAT problems is considered.

5. The Gap for 2SAT Problems

In this section we prove that $\text{gap}(\Phi) = \infty$ if Φ is an unsatisfiable 2SAT formula. It is well known that 2SAT problems are in fact solvable in linear time [2]. Furthermore, Goemans and Williamson [13] and Feige and Goemans [12] gave approximation algorithms with performance guarantee for the MAX2SAT problem, based on semidefinite programming. Hence it is not our aim to improve on any of the above algorithms; we merely want to show some properties of the gap approach. To this end we first characterize infeasibility of 2SAT formulas.

DEFINITION 5.1 (Autarky [27, 21]). Let Φ be any CNF formula. A vector $z \neq 0$ is called a linear autarky of Φ if $Az \geq 0$.

Two formulas Φ and Ψ are called *satisfiability-equivalent* if both are satisfiable or both are contradictory.

LEMMA 5.1. *A linear autarky z indicates a satisfiable subformula Ψ of Φ. The formula $\Phi \backslash \Psi$ is satisfiability-equivalent to Φ.*

Proof. For each linear inequality one of two possibilities is applicable:

(1) All $z_i = 0,\ i \in I_k \cup J_k$ (so $a_k^T z = 0$).
(2) At least one $z_i \neq 0,\ i \in I_k \cup J_k$ (so $a_k^T z \geq 0$).

In the second case, $a_k^T \operatorname{sgn}(z) \geq 2 - |I_k \cup J_k| = b_k$, implying that $\operatorname{sgn}(z)$ restricted to its nonzero entries is a satisfying assignment of the subformula Ψ that is obtained by taking all clauses of Φ associated with linear inequalities with property (2). Note that each clause $\mathbf{C}_k$ either is satisfied by $\operatorname{sgn}(z)$ or remains invariant. Thus, $\Phi \backslash \Psi$ contains a subset of clauses of Φ. If it is satisfiable with some assignment x, then Φ is satisfied by x extended with $\operatorname{sgn}(z)$. If it is unsatisfiable, then this means that a subformula of Φ is unsatisfiable, implying that Φ is unsatisfiable. □

We also make use of the notion of minimal unsatisfiability.

DEFINITION 5.2 (Minimal unsatisfiability). An unsatisfiable CNF formula Φ is called minimally unsatisfiable if a satisfiable formula is obtained by omitting any given clause from Φ.

By Lemma 4.2 we can restrict ourselves to the case that Φ is a minimally unsatisfiable formula.

LEMMA 5.2. *If a formula Φ is minimally unsatisfiable, then $Az \not\geq 0$ for all $0 \neq z \in \mathbb{R}^n$.*

Proof. If the condition in this lemma does not hold, the formula has a linear autarky, contradicting the fact that it is minimally unsatisfiable. □

COROLLARY 5.1. *If Φ is minimally unsatisfiable, then A is of full rank.*

Now we can prove the key lemma of this section.

LEMMA 5.3. *Let Φ be a minimally unsatisfiable $2SAT$ formula. Then* $\operatorname{gap}(\Phi) = \infty$.

Proof. From the semidefinite constraint (see (8)) we conclude that $a^T Y a \geq (a^T y)^2$ for any vector a. This implies that

$$(a_k^T y)^2 - 2a_k^T y \leq a_k^T Y a_k - 2a_k^T y \leq r_k \equiv 0$$

since $r_k = 0$ for 2SAT (for all $k = 1, \ldots, m$), from which it follows that $0 \leq a_k^T y \leq 2$. The minimal unsatisfiability of Φ implies that $y = 0$. Hence, $a_k^T Y a_k =$

0 for all $k = 1, \ldots, m$. Since $Y \succeq 0$ the a_k's must lie in the nullspace of Y,* and since A is of full rank ($\text{rank}(A) = n$) this implies $Y = 0$, contradicting the condition $\text{diag}(Y) = e$ of (D). We conclude that (D) is infeasible, implying that $\text{gap}(\Phi) = \infty$. □

The main theorem of this section follows easily.

THEOREM 5.1. *Let Φ be any 2SAT formula. It holds*

$$\text{gap}(\Phi) = \begin{cases} \infty & \text{if } \Phi \text{ unsatisfiable;} \\ 0 & \text{if } \Phi \text{ satisfiable.} \end{cases}$$

Proof. Let Φ be an unsatisfiable 2SAT formula, and let Ψ be a minimally satisfiable subformula of Φ. By Lemma 5.3, $\text{gap}(\Psi) = \infty$, Lemma 4.2 implies that $\text{gap}(\Phi) = \infty$. Conversely, assume now that Φ is a satisfiable 2SAT formula. Using any satisfying assignment x, we can construct a feasible solution $Y = xx^T$, $y = x$ to (D), implying that $\text{gap}(\Phi) = 0$. □

If $\text{gap}(\Phi) = 0$ we can use the dual solution (Y, y) to construct a satisfying assignment x. First we set the entries of x corresponding to nonzero entries of y to $x = \text{sgn}(y)$ as from the proof of Lemma 5.3 we know that y is an autarky. For all clauses that are not yet satisfied, a satisfying assignment can be constructed by considering Y^*, which denotes the matrix Y restricted to the rows and columns corresponding to the zero entries of y. Note that $a_k^T y = 0$ implies that $a_k^T Y a_k = 0$, so $Y_{ij} = -a_{ki}a_{kj}$ where $I_k \cup J_k = \{i, j\}$. So we can assume that each of the rows and columns of Y^* contains an off-diagonal element that is equal to ± 1. By fixing all ± 1 elements, such a matrix Y^* can be completed to a rank one $\{-1, 1\}$ matrix that is feasible in (D). From this matrix a $\{-1, 1\}$ solution x can easily be deduced. This construction is equivalent to the one given in [12].

6. The Gap for a Class of Covering Problems

In this section we consider SAT encodings of a particular class of covering problems and demonstrate that these can be shown to be contradictory by our SDP approach.

6.1. A SPECIFIC CLASS OF COVERING PROBLEMS

Let V be a set of n propositional variables. Let $\mathcal{S} = \{S_1, \ldots, S_N\}$ and $\mathcal{T} = \{T_1, \ldots, T_M\}$ be sets of subsets of V. Both $\mathcal{S}$ and $\mathcal{T}$ form a partition of V. Furthermore, let us assume that $M < N$. Consider the following CNF formula Φ_{CP}.

$$\bigvee_{i \in S_k} p_i, \quad 1 \le k \le N, \tag{9}$$

* Let $Y = VV^T$; then $0 = a_k^T Y a_k = \|V^T a_k\|^2$, or $V^T a_k = 0$ which implies $Y a_k = 0$.

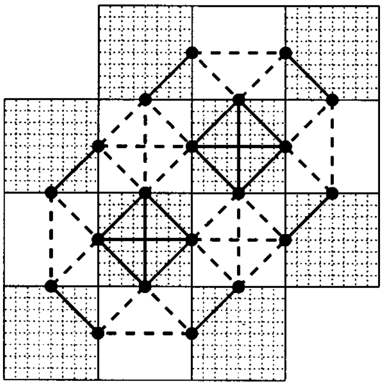

Figure 4. The mutilated chessboard for $s = 2$.

$$\neg p_i \vee \neg p_j, \quad i, j \in T_k, i \neq j, \ 1 \leq k \leq M. \tag{10}$$

First, let us give two examples of problems that fit in this format.

Pigeonhole Formulas. Let us state again the pigeonhole principle:

Given $h + 1$ *pigeons and* h *holes, decide whether it is possible to put each pigeon in at least one hole, while no two pigeons may be put in the same hole*.

We now argue that the standard encoding of the pigeonhole problem fits the format Φ_{CP}. For each pigeonhole pair a proposition is introduced; thus, we obtain a total of $h(h+1)$ variables in the set V. The long (positive) clauses (9) now express that each pigeon is put in at least one hole; thus there are exactly $N = h+1$ of such clauses that all have length h. The short (negative) clauses (10) model that no two pigeons may be put in the same hole simultaneously; for each hole there is a set of short clauses, giving rise to $M = h$ separate sets T_k, each of size $h + 1$. It is easy to see that each proposition occurs both exactly once in the sets S_k and exactly once in the sets T_k; thus both $\mathcal{S}$ and $\mathcal{T}$ are a partition of V.

Mutilated Chessboards. The problem of the mutilated chessboard can be expressed as follows:

Given a chessboard of size $2s \times 2s$ *squares. Two of its diagonally opposite vertices are removed. Can the resulting 'mutilated' chessboard be covered by rectangular dominoes of size* 2×1 (*i.e., a single domino covers exactly two adjacent squares*), *such that each square is covered exactly once?*

The standard satisfiability coding for this problem is obtained by introducing a proposition for each pair of adjacent squares; thus we need $4(2s^2 - s - 1)$ variables. For each square there is a positive clause (of length 2, 3, or 4) expressing that it must be covered at least once, and a set of negative (2-)clauses expressing that it may be covered at most once. It is strongly conjectured that resolution requires exponential running time on this problem as well (despite the absence of 'long' clauses) [33]. Taking a subset of this set of clauses, we obtain a formula of the form Φ_{CP}. For all the *black* squares we keep the positive clauses, while for all the *white* squares we use only the negative clauses. The first set corresponds to (9) and the second set to (10). See also the graph in Figure 4; the nodes correspond to individual variables, the drawn edges indicate the *positive* clauses (all cliques with drawn edges constitute one positive clause), and the dotted edges the *negative* clauses (each pair of nodes connected by a dotted edge corresponds to a negative clause). Again, it is easy to see that each variable occurs in exactly one of the $N = 2s^2$ positive clauses, and in exactly one of the $M = 2s^2 - 2$ sets of negative clauses.

Both of the above problems are easily concluded to be unsatisfiable. We verify that Φ_{CP} is unsatisfiable (when $M < N$), using *cutting planes* (this construction is well known; see, e.g., [6]). To this end we introduce some more notation. By e_S (e_{ij}) we denote the vector with ones in the positions $i \in S$ (i and j) and zeros elsewhere. The integer linear feasibility formulation can be stated as

$$(\mathrm{IP}_{CP})\quad \begin{array}{ll} \text{find} & x \in \{-1, 1\}^n \\ \text{s.t.} & e_{S_k}^T x \geq 2 - |S_k|, \quad 1 \leq k \leq N, \\ & e_{ij}^T x \leq 0, \quad i, j \in T_k, i \neq j,\ 1 \leq k \leq M. \end{array}$$

Obviously, it is easy to find a solution of the linear relaxation of this problem (i.e., when the integrality constraints are relaxed to $-e \leq x \leq e$), by setting all variables to 0. However, using cutting planes, one can show that for $\{-1, 1\}$-variables, the set of inequalities for a set T_k implies that

$$e_{T_k}^T x \leq 2 - |T_k|, \quad 1 \leq k \leq M. \tag{11}$$

(see [6]). A cutting plane is obtained by taking nonnegative combinations of the constraints, and subsequently adjusting the right-hand side of the resulting equality such that it is as sharp as possible. To determine this right-hand side, we use the fact that the variables are binary [4].

For completeness, let us review the derivation (11). Taking a subset $U \subseteq T_k$, $|U| = 3$, and summing the (three) inequalities associated with this set, we obtain the inequality $2e_U^T x \leq 0$. Since for any $\{-1, 1\}$ vector x it holds that $e_U^T x$ is *odd*, the right-hand side may be rounded down to the largest odd integer smaller than zero; thus we find that $e_U^T x \leq -1$. More generally, suppose we are given a set $U \subset T_k$ and an inequality $e_U^T x \leq 2 - |U|$. In addition, let $j \in T_k \backslash U$, and denote $\overline{U} = U \cup \{j\}$. Summing for all $i \in U$ the inequalities $e_{ij}^T x \leq 0$ (with weight 1) and the initial inequality with weight $|U| - 1$, we obtain that $|U| e_{\overline{U}}^T x \leq (|U| - 1)(2 - |U|)$. Dividing both sides by $|U|$, and rounding the right-hand side down to the nearest integer with same parity as $|\overline{U}|$ (thus the right-hand side becomes $1 - |U|$, which is valid, since $((|U| - 1)/|U|)(2 - |U|) < 3 - |U|$), we obtain

$$e_{\overline{U}}^T x \leq 1 - |U| = 2 - |\overline{U}|.$$

We conclude that (11) is indeed implied by the inequalities in (IP_{CP}).

Summing all the inequalities (11), and using that $\mathcal{T}$ partitions V, we find that $-n \le e^T x \le 2M - n$. Similarly, taking the sum over the first set of inequalities in (IP_{CP}), we have $2N - n \le e^T x \le n$. Combining, we get

$$2N \le e^T x + n \le 2M, \tag{12}$$

from which it follows that $N \le M$, implying the infeasibility of (IP_{CP}) and thus the unsatisfiability of Φ_{CP} when $M < N$. We conclude that using this cutting plane technique, a proof of unsatisfiability of Φ_{CP} can be constructed and verified in polynomial time.

Surprisingly, other techniques often require large running times to solve formulas of this type. Indeed, Haken [15] proves that no short resolution proof of the unsatisfiability of pigeonhole formulas exists. The length of any resolution proof is exponential in the size of the formula.

Note that to find the cutting plane proof sketched above efficiently, additional problem–specific information is used. In each step of the cutting plane proof, two linear inequalities are used to derive a new linear inequality. The cutting plane proof of the class (Φ_{CP}) is called a *treelike* cutting plane proof, as each inequality is used only once [19]. To construct such a short treelike cutting plane proof, additional searching is required, since the order in which linear inequalities are combined is crucial. For general CNF formulas, it is not clear how to efficiently find a (treelike) cutting plane proof, even if it is known that one exists. It is our aim to show in this section that using our semidefinite programming approach, we can prove formulas of the format Φ_{CP} to be contradictory in polynomial time, while no additional problem–specific information whatsoever is required. This is due to the fact that (Φ_{CP}) allows a (u, w)-certificate of unsatisfiability whose existence can be computed in polynomial time (Corollary 4.3). Let us consider the semidefinite relaxation of Φ_{CP}. The SDP relaxation can be denoted as (see also (D))

$$(\mathrm{SD}_{CP}) \qquad \begin{array}{rl} \text{find} & Y \in \mathbb{R}^{n\times n},\ y \in \mathbb{R}^n \\ \text{s.t.} & e_{S_k}^T Y e_{S_k} - 2e_{S_k}^T y \le |S_k|(|S_k| - 2), \quad 1 \le k \le N \\ & e_{ij}^T Y e_{ij} + 2e_{ij}^T y \le 0, \quad i, j, i \ne j \in T_k,\ 1 \le k \le M \\ & \mathrm{diag}(Y) = e, \\ & Y \succeq yy^T. \end{array}$$

We prove the following theorem.

THEOREM 6.1. *The semidefinite relaxation* (SD_{CP}) *of* Φ_{CP} *is infeasible* (*if* $M < N$). *Equivalently,* $\mathrm{gap}(\Phi_{CP}) = \infty$.

Proof. Note that from the semidefinite constraint (see (8)) it follows that $a^T Y a \ge (a^T y)^2$ for any n-vector a. Thus it follows that

$$(e_{S_k}^T y)^2 - 2e_{S_k}^T y \le e_{S_k}^T Y e_{S_k} - 2e_{S_k}^T y \le |S_k|(|S_k| - 2),$$

implying that

$$2 - |S_k| \le e_{S_k}^T y \le |S_k|, \quad 1 \le k \le N. \tag{13}$$

Now we consider the inequalities corresponding to the sets T_k. Taking the sum over all the inequalities corresponding to set T_k, k fixed, we find that

$$e_{T_k}^T Y e_{T_k} + (|T_k| - 2)e_{T_k}^T \mathrm{diag}(Y) + 2(|T_k| - 1)e_{T_k}^T y \leq 0.$$

To verify this, note that each diagonal element Y_{ii}, $i \in T_k$, occurs in exactly $|T_k|-1$ inequalities; similarly, each linear term y_i, $i \in T_k$, occurs in exactly $|T_k| - 1$ inequalities as well. Simplifying this expression using that $\mathrm{diag}(Y) = e$, we obtain

$$e_{T_k}^T Y e_{T_k} + 2(|T_k| - 1)e_{T_k}^T y \leq -|T_k|(|T_k| - 2).$$

Using the semidefinite constraint again, we conclude that

$$(e_{T_k}^T y)^2 + 2(|T_k| - 1)e_{T_k}^T y \leq e_{T_k}^T Y e_{T_k} + 2(|T_k| - 1)e_{T_k}^T y \leq -|T_k|(|T_k| - 2),$$

implying that

$$-|T_k| \leq e_{T_k}^T y \leq 2 - |T_k|, \quad 1 \leq k \leq M.$$

Summing these inequalities, we find that $-n \leq e^T y \leq 2M - n$ while from (13) we have that $2N - n \leq e^T y \leq n$, implying that $2N \leq e^T y + n \leq 2M$ (note that this is equivalent to what we obtained using cutting planes, see (12)). Thus we conclude that (SD_{CP}) is infeasible when $N > M$. □

Thus by Corollaries 4.2 and 4.3 we have the following corollary.

COROLLARY 6.1. *With semidefinite programming, the unsatisfiability of* Φ_{CP} *can be decided in polynomial time.*

It may be noted that the proof of Theorem 6.1 and the cutting plane refutation of Φ_{CP} are essentially very similar. Indeed, the cutting planes (11) are automatically implied in (SD_{CP}).

6.2. APPLICATION TO GRAPH COLORING

A famous result by Lovász is his ‘sandwich’ theorem [22], which states that for an undirected graph $G = (V, E)$ in polynomial time (using semidefinite programming), a number $\vartheta(G)$ can be computed that is bounded from above by the graph’s coloring number $\gamma(G)$ (i.e., the minimal number of colors required to color the vertices of the graph such that no two adjacent vertices have the same color), and from below by its clique number $\omega(G)$ (i.e., the maximal complete subgraph of G). Applying our result from the preceding section to the graph coloring problem (GCP), we obtain a similar result.

Suppose we are given a graph $G = (V, E)$ and a set of colors C. We introduce a proposition for each vertex-color combination. Then the GCP can be modeled as a formula Φ_{GCP} containing a set of $|V|$ long clauses, expressing that each vertex should be colored by at least one color, and a set of short clauses expressing that no two vertices may get the same color. We can construct the semidefinite relaxation of Φ_{GCP} in the usual way; we refer to it as (SD_{GCP}). Now let C^* be the smallest set of colors for which (SD_{GCP}) is feasible. Such a set must exist, since for $|C^*| \geq |V|$ the GCP and thus its relaxation (SD_{GCP}) are trivially feasible. We have the following theorem.

THEOREM 6.2. $\omega(G) \leq |C^*| \leq \gamma(G)$.

Proof. First note for any set of color C with $|C| < \omega(G)$, Φ_{GCP} has a subformula of the form Φ_{CP}; hence by Lemma 4.2 it has gap infinity. This subformula corresponds to the set of clauses corresponding to a clique of size $|C|+1$ or larger. Now consider set C^*. It holds that $|C^*| \geq \omega(G)$, since otherwise (SD_{GCP}) would be infeasible. Also, since removing one color from C^* implies infeasibility of (SD_{GCP}) (by assumption), it holds that $|C^*| \leq \gamma(G)$. This proves the theorem. □

So by applying a binary search on the size of C, a number similar to Lovász' ϑ-number can be computed.

In a forthcoming paper [9] the following theorem is proved.

THEOREM 6.3. $\omega(G) \leq |C^*| \leq \lceil \vartheta(G) \rceil \leq \gamma(G)$.

7. The Gap for 3SAT Problems

7.1. A MODIFIED FORMULATION

So far we have seen that several classes of formulas can be solved efficiently by the gap approach. Let us now turn our attention to the most general class of CNF formulas, the 3SAT problems. First we have a negative result for pure 3SAT problems (the class of CNF formulas in which all clauses have length 3).

LEMMA 7.1. *Suppose* Φ *is a pure* 3*SAT problem. It holds that* $\mathrm{gap}(\Phi) = 0$.

Proof. It is easy to verify that the solution $Y = I$, $y = 0$ is feasible in (D) since $a_k^T a_k = 3 = r_k$ for all clauses. By duality, $\mathrm{gap}(\Phi) = 0$. □

COROLLARY 7.1. *No pure* 3*CNF formula allows a* (u, w)*-certificate of unsatisfiability.*

Note that the proof of this lemma and the corollary easily extend to general CNF formulas in which no clauses of length one and/or two occur.

Of course, the gap can be computed in nodes of a branching tree; during the branching process 2-clauses are created, thus making it possible for the SDP approach to be successful in specific cases. Note that (also) in this respect, the SDP approach is stronger than the LP approach (as mentioned in Section 2.1). However, computing the gap is computationally rather expensive, and especially in a DPLL-like branching algorithm (including unit resolution, although this can be simulated by the semidefinite relaxation as well) [7] the overhead will be substantial with the current state-of-the-art implementations. Several possibilities in this respect are discussed in the next section.

For now, let us slightly reformulate our gap relaxation to be able to say a little more about 3SAT formulas. Consider again the elliptic representation $\mathcal{E}_k$ associated with a clause k (1). By introducing a *parity*-variable, the inequality is turned into an equality constraint,

$$\mathcal{E}_k^p = \{x \in \mathbb{R}^n, 0 \leq s_k \leq 1 \mid (a_k^T x - 1)^2 + 4s_k = 4\},$$

where for all feasible $\{-1, 1\}$ assignments to x it holds that $s_k \in \{0, 1\}$. The ellipsoid $\mathcal{E}_k^p$ has the semidefinite relaxation

$$a_k^T Y a_k - 2a_k^T y + 4s_k = 3. \tag{14}$$

Obviously, the trivial solution (Lemma 7.1) is still feasible when simply setting all s_k to 0. However, if we are now going to maximize the sum of the s_k's, a solution other than the trivial one may be obtained. Thus we define the *parity* gap. Consider the semidefinite optimization problem (D′).

$$(\mathrm{D}') \quad \begin{array}{ll} \max & \displaystyle\sum_{k=1}^{m} s_k \\ \text{s.t.} & a_k^T Y a_k - 2a_k^T y + 4s_k = 3, \quad 1 \leq k \leq m, \\ & \operatorname{diag}(Y) = e, \\ & Y \succeq yy^T, \\ & 0 \leq s_k \leq 1, \quad 1 \leq k \leq m. \end{array}$$

DEFINITION 7.1. The optimal value of optimization problem (D′) is called the *parity* gap of a formula Φ.

We then have the following lemma.

LEMMA 7.2. *If a pure* 3*CNF formula* Φ *has* parity *gap zero it is equivalent to an XOR-SAT formula and, as such, efficiently solvable.*

Proof. If the parity gap is zero, $s_k = 0$ for all $1 \leq k \leq m$. Suppose that a satisfying solution x exists for which some clause $\mathbf{C}_k$ is satisfied in exactly two literals, so $a_k^T x = 1$. If we use x, the solution $Y = xx^T,\ y = x$ is feasible in (D′), while $a_k^T Y a_k - 2a_k^T y = -1$, implying that $s_k = 1$. Thus the parity gap is not equal to zero. If (D′) has objective value zero, no such solution exists, implying that if a satisfying assignment exists, then it must satisfy each clause in either exactly one or all three literals (i.e., $a_k^T x \in \{-1, 3\},\ 1 \leq k \leq m$). Thus the formula is equivalent to an XOR-SAT formula [35]. Checking whether such a solution exists can be done in polynomial time [28, 35]. □

If a formula has parity gap zero, its equivalent XOR SAT formula is obtained by simply replacing all occurrences of the 'or' operator ('$\vee$') by the 'exclusive or' operator ('$\oplus$').

A specific class of formulas that appear likely to have parity gap zero is the class of *doubly balanced formulas* [24, 35]. By definition, for doubly balanced formulas it holds that A^TA is a diagonal matrix and $A^Te \equiv 0$. Eliminating the s_k variables from (D′), we can rewrite it to

$$
(\mathrm{D}'') \quad \begin{array}{ll} \min & \mathbf{Tr} \begin{bmatrix} A^TA & -A^Te \\ -e^TA & 0 \end{bmatrix} \begin{bmatrix} Y & y \\ y^T & 1 \end{bmatrix} \\ \text{s.t.} & -1 \le a_k^T Y a_k - 2a_k^T y \le 3, \quad 1 \le k \le m, \\ & \mathrm{diag}(Y) = e, \\ & Y \succeq yy^T. \end{array}
$$

It holds that $\mathrm{opt}(\mathrm{D}') = \frac{3}{4}m - \frac{1}{4}\mathrm{opt}(\mathrm{D}'')$. Note that the objective function of (D″) is essentially the same objective function as the one used by Goemans and Williamson in their MAX2SAT algorithm, only we are considering 3SAT formulas. If we use formulation (D″), it is straightforward to prove the next lemma.

LEMMA 7.3. *A doubly balanced formula has parity gap zero.*

Proof. By the definition of doubly balancedness and the constraint on the diagonal of Y, the objective function of (D″) reduces to the constant $3m$, implying that the parity gap is equal to zero. □

We also have an autarky result.

LEMMA 7.4. *If all* $s_k > 3/4$, $x = \mathrm{sgn}(y)$ *is a satisfiable assignment of* Φ.

Proof. Note that

$$(a_k^T y)^2 - 2a_k^T y + (4s_k - 3) \le 0.$$

This implies that

$$1 - 2\sqrt{1 - s_k} \le a_k^T y \le 1 + 2\sqrt{1 - s_k}.$$

If $s_k > 3/4$, it holds that $a_k^T y > 0$. If all s_k have this property, then y is a linear autarky. □

Note that y might be a linear autarky while not all s_k are larger than $3/4$. In this respect, a slightly stronger formulation is obtained by using *a single* slack variable t for all clauses, rather than a separate slack variable s_k for each of the clauses. The objective then becomes to maximize t; all s_k's must be replaced by t, and equality must be replaced by inequality. A drawback of this approach is that an optimum of zero implies only that a polynomially solvable *sub*formula is present. On the other hand, solving this subformula first may speed solving the full formula [37].

While it appears that the semidefinite relaxation as developed in this paper is not quite strong enough to completely solve 3SAT problems by itself, it can be used in various other ways. Apart from including it in a complete algorithm (see the next section), both its primal and dual solution can be used for heuristic purposes. This is briefly discussed in Section 8.

7.2. INCORPORATING THE GAP APPROACH IN A COMPLETE ALGORITHM

As mentioned in the preceding section, the gap can be computed in each node of a branching tree to detect unsatisfiability early, so as to reduce the size of the search tree. Even though at present it appears that this might be computationally too expensive, there are several ways to reduce computational cost. We mention two possibilities.

1. Instead of using a primal-dual algorithm, it is possible to use a dual scaling algorithm to exploit sparsity of $\tilde{Q}$ to the full [3]. Computational experience with MAX2SAT problems indicates that for small-sized problems this approach is competitive with other dedicated algorithms for the MAX2SAT problem [10].
2. For larger problems, spectral bundle methods can be used to solve the eigenvalue optimization problem (P) [16]. These have been shown to be able to handle problems with thousands of variables. Such methods solve only (P), so that the dual information is lost.

8. Remarks and Further Research

8.1. SOME EMPIRICAL OBSERVATIONS

Both the primal and dual solutions obtained by solving a semidefinite relaxations can be used for heuristic purposes. The dual solution (Y, y) might be used to try to obtain good approximate MAX-SAT solutions. In Figure 5, we illustrate the quality of the solutions thus obtainable. We restricted ourselves to a set of random pure 3SAT formulas with 100 variables and a varying number of clauses, and used the dual formulation with a single slack variable. The SDPs were solved using the public-domain solver SeDuMi [31]. Approximate solutions are constructed by (i) taking the dual solution and rounding y (the drawn line), and (ii) applying a Goemans–Williamson-like randomized rounding procedure (the dotted line). For comparison we also included the solutions obtained by simply drawing random solutions; note that this in fact corresponds to the Karloff–Zwick algorithm for approximating (pure) MAX3SAT solutions. To obtain a good ad hoc lower bound on the optimal solution, we applied a greedy weighted local search procedure. Considering the results, it is clear that the local search procedure gives the best results, but interestingly enough the drawn line is consistently above the dotted line, implying that the deterministic one-step procedure of rounding y gives better solutions than the randomized rounding procedure, which in turn is better than just drawing random solutions.

Let us finish this section by mentioning that using the primal optimal solution w, it is possible to identify hard subformulas. To this end, the sum of the weights

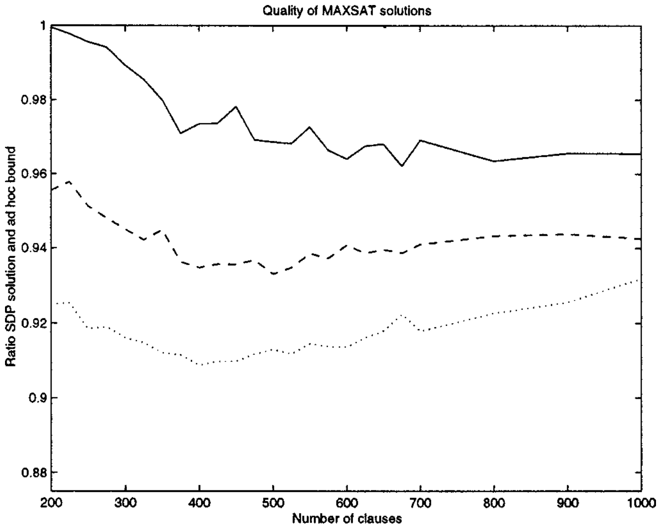

Figure 5. Comparison of MAXSAT solutions.

is bounded (this corresponds to using a single slack variable in the dual formulation). Removing the clauses with 'small' weights, we obtain a subformula that appears to be the 'core' of the original formula. This yields a technique for finding approximately minimal unsatisfiable subformulas.

8.2. MORE GENERAL CLAUSE MODELS

So far we have considered a fixed quadratic clause-model based on an elliptic approximation. Obviously there exist many other quadratic clause-models. Basically, all quadratic models $q(x)$ with the following property are valid.

PROPERTY 8.1. *If x is a satisfying assignment, then*

$$q(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} q_{ij} x_i x_j + \sum_{k=1}^{n} q_k x_k + q_0 \leq 0.$$

As an example, let us consider the clause $\mathbf{C} = p_1 \vee p_2 \vee p_3$. Substituting the satisfying assignments of $\mathbf{C}$ in $q(x)$, we obtain a set of 7 inequalities. These inequalities, which are linear in the coefficients q_{ij} and q_k, define a cone. The extreme rays of this cone give rise to the following generic set of valid quadratic cuts (S).

$$x_1x_2 + x_1x_3 - x_2 - x_3 \leq 0,$$
$$x_1x_2 + x_2x_3 - x_1 - x_3 \leq 0,$$
$$x_1x_3 + x_2x_3 - x_1 - x_2 \leq 0,$$
$$-x_1x_2 - x_1x_3 - x_2x_3 - 1 \leq 0,$$
$$-x_1x_2 + x_1 + x_2 - 1 \leq 0,$$

$$-x_1x_3 + x_1 + x_3 - 1 \leq 0,$$
$$-x_2x_3 + x_2 + x_3 - 1 \leq 0.$$

An equivalent set is given by Karloff and Zwick [20]. So, each valid quadratic model of **C** is a linear combination of this set of inequalities. In particular, the elliptic approximation is obtained by taking the sum of the first three inequalities (see (1) and use that $x^2 = 1$ for $x \in \{-1, 1\}$). Karloff and Zwick show that using the first three inequalities a $7/8$ approximation algorithm for MAX3SAT problems can be obtained. Thus, our relaxation is an aggregated version of the Karloff–Zwick relaxation.

Note that the SDP relaxations can be further strengthened by adding valid inequalities. An example of such cuts are the *triangle inequalities* [12]; note that many of these are in fact implied by (S) (namely those concerning variables that occur jointly in some clause). Another possibility is using (see also (1))

$$(a_k^T x - 1)(a_l^T x - 1) \leq (\ell(\mathbf{C}_k) - 1)(\ell(\mathbf{C}_l) - 1)$$

for any pair of clauses $\mathbf{C}_k,\ \mathbf{C}_l$ which share one or more variables. This is an example of a valid quadratic cut for *pairs* of clauses. Similar to (S), all valid quadratic cuts may be derived for pairs of clauses.

Generic procedures for deriving increasingly stronger relaxations of growing dimensions for binary programming problems in general, are given by Sherali and Adams [29] and Lovász and Schrijver [23]. These relaxations eventually describe the convex hull. While these results are of great theoretical interest, it is not entirely clear how to implement them in practice, due to the exponential size of the resulting relaxations. Using the elliptic approximations, we obtain only one quadratic inequality per clause.

References

1. Alizadeh, F.: Combinatorial optimization with interior point methods and semi-definite matrices, Ph.D. Thesis, University of Minnesota, Minneapolis, 1991.
2. Aspvall, B., Plass, M. F., and Tarjan, R. E.: A linear-time algorithm for testing the truth of certain quantified Boolean formulas, *Inform. Process. Lett.* **8**(3) (1979), 121–123.
3. Benson, S. J., Ye, Y., and Zhang, X.: Solving large-scale sparse semidefinite programs for combinatorial optimization, Technical Report, Computational Optimization Lab., Department of Management Science, University of Iowa, Iowa City, 1997.
4. Chvátal, V.: Edmonds polytopes and a hierarchy of combinatorial problems, *Discrete Math.* **4** (1973), 305–337.
5. Cook, S. A.: The complexity of theorem proving procedures, in *Proceedings of the 3rd annual ACM symposium on the Theory of Computing*, 1971, pp. 151–158.
6. Cook, W., Coullard, C. R., and Turan, G.: On the complexity of cutting plane proofs, *Discrete Appl. Math.* **18** (1987), 25–38.
7. Davis, M., Logemann, M., and Loveland, D.: A machine program for theorem proving, *Comm. ACM* **5** (1962), 394–397.
8. de Klerk, E.: Interior point methods for semidefinite programming, Ph.D. Thesis, Delft University of Technology, Delft, The Netherlands, 1997.
9. de Klerk, E., Pasechnik, D. V., and Warners, J. P.: Approximate graph colouring algorithms based on the ϑ-function, Technical Report, 1999. In preparation.

10. de Klerk, E. and Warners, J. P.: Semidefinite programming techniques for MAX-2-SAT and MAX-3-SAT: Computational perspectives, Technical Report 98–34, Department of Technical Mathematics and Informatics, Faculty of Information Technology and Systems, Delft University of Technology, Delft, The Netherlands, 1998.
11. Du, D., Gu, J., and Pardalos, P. M. (eds.): *Satisfiability Problem: Theory and Applications*, DIMACS Series in Discrete Math. and Comput. Sci. 35, Amer. Math. Soc., 1997.
12. Feige, U. and Goemans, M.: Approximating the value of two prover proof systems with applications to MAX 2SAT and MAX DICUT, in *Proc. Third Israel Symposium on Theory of Computing and Systems*, 1995, pp. 182–189.
13. Goemans, M. X. and Williamson, D. P.: Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, *J. ACM* **42**(6) (1995), 1115–1145.
14. Gu, J., Purdom, P. W., Franco, J., and Wah, B. W.: Algorithms for the satisfiability (SAT) problem: A survey, in Du et al. [11], pp. 9–151.
15. Haken, A.: The intractability of resolution, *Theoret. Comput. Sci.* **39** (1985), 297–308.
16. Helmberg, C. and Rendl, F.: A spectral bundle method for SDP, Technical Report ZIB, Preprint SC-97-37, Konrad-Zuse-Zentrum, Berlin, 1997.
17. Hooker, J. N.: Resolution vs. cutting plane solution of inference problems: some computational experience, *Oper. Res. Lett.* **7**(1) (1988), 1–7.
18. Horn, R. A. and Johnson, C. R.: *Matrix Analysis*, Cambridge University Press, 1985.
19. Impagliazzo, R., Pitassi, T., and Urquhart, A.: Upper and lower bounds for treelike cutting plane proofs, in *Proceedings of the 9th Annual IEEE Symposium on Logic in Computer Science*, 1994, pp. 220–228.
20. Karloff, H. and Zwick, U.: A 7/8-approximation algorithm for MAX3SAT? in *Proceedings of the 38th Symposium on the Foundations of Computer Science*, 1997, pp. 406–415.
21. Kullmann, O.: Investigations on autark assignments, Technical Report, Johann Wolfgang Goethe-Universität, Fachbereich Mathematik, 60054 Frankfurt, Germany, 1998. Submitted.
22. Lovász, L.: On the Shannon capacity of a graph, *IEEE Trans. Inform. Theory* **25** (1979), 1–7.
23. Lovász, L. and Schrijver, A.: Cones of matrices and set-functions and 0-1 optimization, *SIAM J. Optim.* **1**(2) (1991), 166–190.
24. Van Maaren, H.: Elliptic approximations of propositional formulae, Technical Report 96-65, Faculty of Technical Mathematics and Informatics, Delft University of Technology, Delft, The Netherlands, 1996. To appear in *Discrete Appl. Math.*
25. Van Maaren, H.: On the use of second order derivatives for the satisfiability problem, in Du et al. [11], pp. 677–687.
26. Van Maaren, H. and Warners, J. P.: Solving satisfiability problems using elliptic approximations – a note on volumes and weights, Technical Report 98-32, Department of Technical Mathematics and Informatics, Faculty of Information Technology and Systems, Delft University of Technology, Delft, The Netherlands, 1998.
27. Monien, B. and Speckenmeyer, E.: Solving satisfiability in less than 2^n steps, *Discrete Appl. Math.* **10** (1985), 287–295.
28. Schaefer, T. J.: The complexity of satisfiability problems, in *Proceedings of the Tenth Symposium on the Theory of Computing*, 1978, pp. 216–226.
29. Sherali, H. D. and Adams, W. P.: A hierarchy of relaxations between the continuous and convex hull representations for 0-1 programming problems, *SIAM J. Discrete Math.* **3**(3) (1990), 411–430.
30. Strang, G.: *Linear Algebra and Its Applications*, 3rd edn, Harcourt Brace Jovanovich, 1988.
31. Sturm, J. F.: Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones, Technical Report, Communications Research Laboratory, McMaster University, Hamilton, Canada, 1998.

32. Trevisan, L., Sorkin, G., Sudan, M., and Williamson, D.: Gadgets, approximation and linear programming, in *Proceedings of the 37th Symposium on the Foundations of Computer Science*, 1996, pp. 617–626.
33. Urquhart, A.: *Open Problem Posed at SAT'98*, Paderborn, Germany, 1998.
34. Vandenberghe, L. and Boyd, S.: Semidefinite programming, *SIAM Rev.* **38** (1996), 49–95.
35. Warners, J. P. and Van Maaren, H.: Recognition of tractable satisfiability problems through balanced polynomial representations, *Discrete Appl. Math.*, 1999. To appear.
36. Warners, J. P. and Van Maaren, H.: Solving satisfiability problems using elliptic approximations – effective branching rules, Technical Report 98-18, Department of Technical Mathematics and Informatics, Faculty of Information Technology and Systems, Delft University of Technology, Delft, The Netherlands, 1998. Accepted for publication in *Discrete Appl. Math.*
37. Warners, J. P. and Van Maaren, H.: A two-phase algorithm for solving a class of hard satisfiability problems, *Oper. Res. Lett.* **23**(3–5) (1999), 81–88.