Towards an SDP-based Algorithm for the Satisfiability Problem

Miguel F. Anjos

Operational Research Group, School of Mathematics, University of Southampton, U.K.

The Satisfiability Problem (SAT)

The satisfiability (SAT) problem is a central problem in mathematical logic, computing theory, and artificial intelligence. We consider instances of SAT specified by a set of boolean variables $x_1, \ldots, x_n$ and a propositional formula $\Phi = \bigwedge_{j=1}^{m} C_j$, with each clause C_j having the form $C_j = \bigvee_{k \in I_j} x_k \vee \bigvee_{k \in \bar{I}_j} \bar{x}_k$ where $I_j, \bar{I}_j \subseteq \{1, \ldots, n\}$, $I_j \cap \bar{I}_j = \emptyset$, and $\bar{x}_i$ denotes the negation of x_i. Given such an instance, the SAT problem asks whether there is a truth assignment to the variables such that the formula is satisfied.

Semidefinite Programming (SDP)

Semidefinite programming (SDP) refers to the class of optimization problems where a linear function of a symmetric matrix variable X is optimized subject to linear constraints on the elements of X and the additional constraint that X must be positive semidefinite. A primal-dual pair for SDP has the form:

$$\begin{array}{ll|ll} \max & C \cdot X & \min & b^T y \\ \text{s.t.} & A_i \cdot X = b_i,\ i = 1, \ldots, m & \text{s.t.} & Z = \sum_{i=1}^{m} y_i A_i - C \\ & X \succeq 0 & & Z \succeq 0 \end{array}$$

where $A \cdot B = \sum_{i,j} A_{i,j} B_{i,j} = \operatorname{trace}(B^T A)$, and $X \succeq 0$ denotes that X is symmetric positive semidefinite.

Solving SAT Using SDP – The Gap Relaxation

We study the application of SDP to the basic SAT problem, particularly for proving unsatisfiability.

In [5, 6], de Klerk, van Maaren, and Warners introduced an SDP relaxation for SAT, the Gap relaxation. This SDP relaxation is based on the elliptic approximations of clauses introduced in [9] and characterizes unsatisfiability for several interesting classes of SAT problems, such as mutilated chessboard and pigeonhole instances. However, it cannot detect unsatisfiability when all the clauses have length three or higher.

We now introduce some notation. Let 1 denote TRUE and -1 denote FALSE, and for clause C_j and $k \in I_j \cup \bar{I}_j$, let $s_{j,k} = 1$ if $k \in I_j$ and $s_{j,k} = 1$ if $k \in \bar{I}_j$. Then clause C_j is satisfied $\Leftrightarrow s_{j,k} x_k = 1$ for some $k \in I_j \cup \bar{I}_j \Leftrightarrow \prod_{k \in I_j \cup \bar{I}_j} (1 - s_{j,k} x_k) = 0$.

Consider the simple example

$$(x_1 \vee x_2) \wedge (x_2 \vee \bar{x}_3 \vee x_4)$$

For the first clause, the satisfiability condition is $x_1 + x_2 - x_1 x_2 = 1$ which involves the terms x_1, x_2, and $x_1 x_2$. Similarly, for the second clause, we have $x_2 - x_3 + x_4 + x_2 x_3 - x_2 x_4 + x_3 x_4 - x_2 x_3 x_4 = 1$.

The Gap relaxation for this example is

$$\begin{array}{ll} \text{find} & X \succeq 0 \\ \text{s.t.} & \\ & X_{1,2} - X_{0,1} - X_{0,2} + 1 = 0 \\ & X_{2,4} - X_{3,4} - X_{2,3} - X_{0,2} + X_{0,3} - X_{0,4} \leq 0 \\ & \\ & X = \begin{pmatrix} 1 & X_{0,1} & X_{0,2} & X_{0,3} & X_{0,4} \\ X_{0,1} & 1 & X_{1,2} & X_{1,3} & X_{1,4} \\ X_{0,2} & X_{1,2} & 1 & X_{2,3} & X_{2,4} \\ X_{0,3} & X_{1,3} & X_{2,3} & 1 & X_{3,4} \\ X_{0,4} & X_{1,4} & X_{2,4} & X_{3,4} & 1 \end{pmatrix} \end{array}$$

where the entry $X_{0,i}$ represents x_i and the entry $X_{i,j}$ represents $x_i x_j$.

Note that for the Gap relaxation, the satisfiability condition for clauses involving more than two variables must be relaxed to a quadratic inequality since there are no entries in X to represent terms such as $x_2 x_3 x_4$. Hence the Gap relaxation is always feasible when the SAT instance has no clauses of length less than three, and is unable to detect unsatisfiability for such instances.

The Higher Liftings Paradigm

Note that the rows and columns of the matrix variable in the Gap relaxation are indexed by the binary variables themselves:

$$X = \begin{pmatrix} 1 & X_{0,1} & \cdots & X_{0,n} \\ & 1 & & \\ & & \ddots & X_{i,j} \\ & & & 1 \end{pmatrix}$$

We consider semidefinite relaxations with the rows and columns of the matrix variable indexed by subsets of the set of variables. These higher liftings

- have strong theoretical properties (see [4, 7, 8]). In particular, using all 2^n subsets means that we are optimizing over the convex hull of the ± 1 feasible solutions;

- but the size of the liftings grows very rapidly with the number of binary variables. As a consequence, only second liftings for max-cut problems with only up to 27 binary variables were successfully solved in [1].

Goals of This Research

- Find "partial" liftings which are more amenable to practical computation than the complete higher liftings, while preserving (some of) their theoretical strength.
- Hence develop improved SDP relaxations for SAT, and employ them to obtain a practical SDP-based algorithm for solving the general satisfiability problem.

An Improved SDP Relaxation for SAT

Let $\mathcal{P}$ denote the set of all nonempty sets $I \subseteq \{1, \ldots, n\}$ such that the term $\prod_{i \in I} x_i$ appears in the instance's satisfiability conditions. Then we introduce new variables $x_I := \prod_{i \in I} x_i$ for each $I \in \mathcal{P}$, define the vector $v := (1, x_{I_1}, \ldots, x_{I_{|\mathcal{P}|}})^T$, and define the rank-one matrix $Y := vv^T$ whose rows and columns are indexed by $\emptyset \cup \mathcal{P}$. By construction, $Y_{\emptyset,I} = x_I$ for all $I \in \mathcal{P}$. Furthermore, $Y_{I_1,I_2} = Y_{I_3,I_4}$ whenever $I_1 \Delta I_2 = I_3 \Delta I_4$, where $I_i \Delta I_j$ denotes the symmetric difference of I_i and I_j. The tradeoff involved in adding such constraints to the SDP is that as the number of constraints increases, the semidefinite relaxations become computationally more expensive to solve. We use the smaller set of constraints:

$$Y_{\emptyset,I_1} = Y_{I_2,I_3}, \quad Y_{\emptyset,I_2} = Y_{I_1,I_3}, \quad \text{and} \quad Y_{\emptyset,I_3} = Y_{I_1,I_2}. \tag{1}$$

for all the triples $\{I_1, I_2, I_3\} \subseteq \mathcal{P}$ such that $(I_1 \cup I_2 \cup I_3) \subseteq (I_j \cup \bar{I}_j)$ for some clause j and satisfying the symmetric difference condition above. For our example, the improved SDP relaxation is

$$\begin{array}{ll} \text{find} & Y \succeq 0 \\ \text{s.t.} & \\ & Y_{\emptyset,x_1} + Y_{\emptyset,x_2} - Y_{\emptyset,x_{12}} = 1 \\ & Y_{\emptyset,x_2} - Y_{\emptyset,x_3} + Y_{\emptyset,x_4} + Y_{\emptyset,x_{23}} - Y_{\emptyset,x_{24}} + Y_{\emptyset,x_{34}} - Y_{\emptyset,x_{234}} = 1 \end{array}$$

$$Y = \begin{array}{c} \begin{array}{cccccccccc} \scriptstyle\emptyset & \scriptstyle\{1\} & \scriptstyle\{2\} & \scriptstyle\{1,2\} & \scriptstyle\{3\} & \scriptstyle\{4\} & \scriptstyle\{2,3\} & \scriptstyle\{2,4\} & \scriptstyle\{3,4\} & \scriptstyle\{2,3,4\} \end{array} \\ \begin{pmatrix} 1 & x_1 & x_2 & x_{12} & x_3 & x_4 & x_{23} & x_{24} & x_{34} & x_{234} \\ & 1 & x_{12} & x_2 & * & * & * & * & * & * \\ & & 1 & x_1 & x_{23} & x_{24} & x_3 & x_4 & x_{234} & x_{34} \\ & & & 1 & * & * & * & * & * & * \\ & & & & 1 & x_{34} & x_2 & x_{234} & x_4 & x_{24} \\ & & & & & 1 & x_{234} & x_2 & x_3 & x_{23} \\ & & & & & & 1 & x_{34} & x_{24} & x_4 \\ & & & & & & & 1 & x_{23} & x_3 \\ & & & & & & & & 1 & x_2 \\ & & & & & & & & & 1 \end{pmatrix} \end{array}$$

The asterisk elements are not involved in any of the linear equality constraints (but they are constrained by positive semidefiniteness). The motivation for the particular choice of constraints in (1) is that they suffice to prove:

Theorem 1. *Given any propositional formula in CNF, consider the SDP relaxation constructed as presented. Then*
- *If the SDP relaxation is infeasible, then the formula is unsatisfiable.*
- *If the SDP relaxation is feasible, and Y is a feasible matrix such that $\operatorname{rank} Y \leq 3$, then a truth assignment satisfying the formula can be obtained from Y.*

Thus we can use the SDP relaxation to prove *either* satisfiability *or* unsatisfiability of the given SAT instance. But due to the CPU time required to solve the improved SDP relaxations, we can tackle only small SAT instances, so...

Is the SDP-based Approach Competitive?

We solved the improved SDP relaxation on a 2.4GHz Pentium IV with 1.5Gb of RAM for 3 unsatisfiable instances of SAT that were unsolved during the SAT2003 competition. The SDP relaxation is infeasible for all 3 instances, and in particular it was able to prove in less than two hours (the time limit used in the competition) the unsatisfiability of the smallest unsatisfiable instance that remained unsolved during the competition.

Problem Name	# of variables and clauses	Improved SDP proved UNSAT	Total CPU seconds
hgen8-n260-01*	260 / 391	Yes	6922
hgen8-n260-02	260 / 404	Yes	7438
hgen8-n260-03	260 / 399	Yes	7662

*the smallest UNSAT instance unsolved at SAT 2003 (7200 sec. timeout)

These results show the potential of our relaxation for complementing existing techniques for SAT. Since the computational time required is dominated by the effort required to solve the SDPs, future research will consider how the structure of the SDP relaxations could be specifically exploited.

References

[1] M.F. Anjos. *New Convex Relaxations for the Maximum Cut and VLSI Layout Problems.* PhD thesis, University of Waterloo, 2001.

[2] M.F. Anjos. An improved semidefinite programming relaxation for the satisfiability problem. *Math. Program.*, (Ser. A), to appear.

[3] M.F. Anjos. On Semidefinite Programming Relaxations for the Satisfiability Problem. *Math. Meth. Oper. Res.*, to appear.

[4] M.F. Anjos and H. Wolkowicz. Strengthened semidefinite relaxations via a second lifting for the max-cut problem. *Discrete Appl. Math.*, 119(1–2):79–106, 2002.

[5] E. de Klerk and H. van Maaren. On semidefinite programming relaxations of $(2+p)$-SAT. *Ann. Math. Artif. Intell.*, 37(3):285–305, 2003.

[6] E. de Klerk, H. van Maaren, and J.P. Warners. Relaxations of the satisfiability problem using semidefinite programming. *J. Automat. Reason.*, 24(1-2):37–65, 2000.

[7] J.B. Lasserre. An explicit equivalent positive semidefinite program for nonlinear 0-1 programs. *SIAM J. Optim.*, 12(3):756–769 (electronic), 2002.

[8] M. Laurent. Semidefinite relaxations for max-cut. In M. Grötschel, editor, *The Sharpest Cut, Festschrift in Honor of M. Padberg's 60th Birthday.* SIAM, to appear.

[9] H. van Maaren. Elliptic approximations of propositional formulae. *Discrete Appl. Math.*, 96/97:223–244, 1999.