Complexity of Partial Satisfaction II

Karl Lieberherr[*]

GTE Laboratories

Ernst Specker

Swiss Federal Institute of Technology
Zurich

ABSTRACT

What is easy and when does it become hard to find a solution of a problem? We give a sharp answer to this question for various generalizations of the well-known maximum satisfiability problem. For several maximum ψ-satisfiability problems we explicitly determine algebraic numbers τ_ψ $(0 < \tau_\psi < 1)$, which separate NP-complete from polynomial problems. The fraction τ_ψ of the clauses of a ψ-formula can be satisfied in polynomial time, while the set of ψ-formulas which have an assignment satisfying the fraction τ' $(\tau' > \tau_\psi, \tau'$ rational) of the clauses is NP-complete.

November 27, 1985

Complexity of Partial Satisfaction II

Karl Lieberherr[*]

GTE Laboratories

Ernst Specker

Swiss Federal Institute of Technology
Zurich

1. Introduction

We continue our study of the (generalized) satisfiability problem [Lieberherr/Specker (1981), Lieberherr (1982)].

One often recurring theme in computer science is the following: Given an algorithmic problem, find an algorithm which is optimal with respect to a certain measure. The motivation for looking for such algorithms is that the algorithm designer has a guarantee that his algorithm is best-possible in a certain precise sense. Some typical measures are time, space, comparisons, $A \cdot T^2$ etc. For many measures it is hard to prove that a given algorithm is optimal.

We have analyzed the algorithmic problem of finding good approximate solutions for generalized satisfiability problems. The measure we used to compare algorithms is the quality of the approximations which they find. In [Lieberherr (1982)] we describe an efficient algorithm MAXMEAN* which is best possible with respect to our measure for a large class of problems. In this paper we provide a framework for the analysis of MAXMEAN* and we apply our method to several special cases.

We investigate combinatorial optimization problems of the following form: Given a sequence of constraints, find an assignment which satisfies as many as possible. This constraint satisfaction problem appears in many applications like time table scheduling, laying out graphs in a grid, decoding of linear codes, minimizing PLA's etc.

Maximization problems of this type are naturally formulated as maximum ψ-satisfiability problems [Schaefer (1978)]. ψ is a finite set of logical relations

[*] Supported by National Science Foundation grant MCS80-04490 and Forschungsinstitut fuer Mathematik, ETH Zurich. Most of this research was done while the first author was at Princeton University.
Author's addresses: Karl Lieberherr, GTE Laboratories, 40 Sylvan Road, Waltham MA 02254. Ernst Specker, Dept. of Mathematics, ETH-Zentrum, CH-8092 Zurich, Switzerland. v.3

$R_1, \cdots, R_m$ which are used to express the constraints. A ψ-formula S with n variables is a finite sequence of clauses each of the form $R_i(x_1, \cdots, x_{r_i})$. r_i is the rank of R_i and $x_1, \cdots x_{r_i}$ are a subset of the variables of S. The maximum ψ-satisfiability problem consists of finding, for any ψ-formula S, an assignment to the n variables satisfying the maximum number of the clauses.

Let τ_ψ be the fraction of the clauses which can be satisfied efficiently in any ψ-formula S. It is shown in [Lieberherr (1982)] that the following algorithm MAXMEAN* guarantees to satisfy the fraction τ_ψ in time $O(|S| \cdot |clauses(S)|)$, where $|clauses(S)|$ is the number of clauses in S.

Algorithm MAXMEAN*

Input: A ψ-formula S with n variables.

Output: An assignment satisfying at least the fraction τ_ψ of the clauses.

max_assignment := 0;
loop
compute k such that

$$\max_{0 \le k' \le n} mean_{k'}(S) = mean_k(S)$$

{ $mean_k(S)$ is the average fraction of satisfied clauses in S among all assignments having exactly k ones. $mean_k(S)$ is a polynomial in k which can be efficiently computed }
for all variables $x \in S$ **do**
if $mean_{k-1}(S_{x=1}) > mean_k(S_{x=0})$
then $J[x] := 1; k := k-1; S := S_{x=1}$
else $J[x] := 0; S := S_{x=0}$
{ $mean_{-1}(S) = mean_0(S), mean_{n+1}(S) = mean_n(S)$ }
h := *SATISFIED* (S, J); { *SATISFIED* (S, J) is the number of satisfied clauses in S under assignment J }
if $h >$ *max_assignment* **then** *max_assignment* := h **else exit** ;
rename all variables in S which are assigned 1 by J;
end ;

Already after one iteration of the outermost loop of *MAXMEAN** the fraction τ_ψ of the clauses is satisfied by assignment J [Lieberherr (1982)].

From the definition of τ_ψ it follows that MAXMEAN* is a polynomial $(1-\tau_\psi)$-approximate algorithm for the maximum ψ-satisfiability problem, i.e. MAXMEAN* comes within $1-\tau_\psi$ of the optimal assignment. It is an open problem whether there are polynomial ϵ'-approximate algorithms for ϵ'

$<1-\tau_\psi$. However it is shown in [Lieberherr (1982)] that it is NP-complete to decide whether more than the fraction τ_ψ of the clauses can be satisfied in a given ψ-formula.

In the following we outline the contents of this paper. In [Lieberherr (1982)] we reduce the determination of τ_ψ for a given ψ to a discrete minimax problem. We show, that the discrete minimax problem can be reduced to a continuous minimax problem which is considerably easier to solve (Theorem 2.1).

We determine τ_ψ for several maximum ψ-satisfiability problems, each requiring a different proof technique. In Theorem 3.1 we analyze special systems of linear inequalities, i.e. a special case of the (0,1)-integer programming problem.

Theorem 3.1

Let R_j be the relation of rank r which holds, if exactly j of the r variables are assigned one. Let $\psi=\{R_0, \cdots, R_r\}$. Then algorithm MAXMEAN* satisfies the fraction $\frac{1}{r+1}$ of the clauses in a ψ-formula S in time $O(|S| \, |\mathit{clauses}(S)|)$. The set of ψ-formulas having an assignment satisfying the fraction $\tau' > \frac{1}{r+1}$ of the clauses is NP-complete (τ' rational).

The proof first uses the above reduction (Theorem 2.1) and continues with an averaging trick which simplifies a part of the problem to the computation of an integral. The integral is solved by partial integration. The mean value theorem from calculus and some further minimax manipulations complete the proof.

The following theorem analyzes subclasses of the regular satisfiability problem.

Theorem 4.1

Let $F(p,q)$ be the class of propositional formulas in conjunctive normal form which contain in each clause at least p positive or at least q negative literals ($p, q \geq 1$). Let α be the solution of $(1-x)^p = x^q$ in (0,1) and let $\tau_{p,q} = 1-\alpha^q$. Then algorithm MAXMEAN* satisfies the fraction $\tau_{p,q}$ of the clauses in any formula $\in F(p,q)$ in time $O(|S| \, |\mathit{clauses}(S)|)$. The set of formulas $\in F(p,q)$ which have an assignment satisfying the fraction $\tau' > \tau_{p,q}$ of the clauses is NP-complete (τ' rational).

The proof is involved and is decomposed into 3 simplifying reductions.

In the last part of the paper we partially solve a problem which was left open in [Lieberherr/Specker (1981)]. We give an efficient algorithm which guarantees to satisfy 2/3 of the clauses in a 3-satisfiable conjunctive normal form.

2. Reduction to a continuous min-max-problem

In the following we sketch how the computation of τ_ψ can be simplified to a discrete minimax problem involving polynomials (a more detailed explanation is in [Lieberherr (1982)]).

τ_ψ is by definition the fraction of the clauses which can be satisfied in all ψ-formulas. First we consider ψ-formulas with at most n variables and let $\tau_{n,\psi}$ be the fraction of clauses which can be satisfied in all such formulas.

For computing $\tau_{n,\psi}$ we determine the worst-case formulas, i.e. the formulas where the smallest fraction of the clauses can be satisfied (by the optimal assignment) among all ψ-formulas with n variables. It is easy to prove that these formulas are symmetric, i.e. they are invariant under permutations of the variables, upto a permutation of the clauses.

Fortunately the worst-case formulas have a nice structure and therefore it is easy to compute an optimal assignment for them. For computing an optimal assignment for a symmetric formula we only have to compute the maximum of a polynomial. This polynomial can be derived by elementary combinatorial analysis.

In this section we prove a theorem which simplifies the computation of τ_ψ to the solution of a continuous minimax problem which does not involve a limit operation. Let $\psi = \{R_1, R_2, \ldots, R_m\}$ be a finite set of relations and let S be a symmetric ψ-formula in which the fraction t_{R_i} of the clauses contains clauses involving relation R_i. In order to compute $\tau_{n,\psi}$ we have to find the worst assignment to the parameters $t_{R_1} \cdots t_{R_m}$ which makes the optimal fraction of satisfiable clauses as small as possible.

It follows from the above discussion that (assuming that $\sum_{i=1}^{m} t_{R_i} = 1$, $t_{R_i} \geq 0$ $(1 \leq i \leq m)$)

$$\tau_\psi = \lim_{n \to \infty} \tau_{n,\psi},$$

$$\tau_{n,\psi} = \min_{\substack{t_{R_i}(1 \leq i \leq m) \\ rational}} \max_{\substack{0 \leq k \leq n \\ integer}} \sum_{i=1}^{m} t_{R_i} \cdot SAT_k^n(R_i)$$

$$SAT_k^n(R) = \sum_{s=0}^{r(R)} q_s(R) \frac{(k)_s (n-k)_{r(R)-s}}{(n)_{r(R)}}$$

where

t_R is the fraction of clauses containing relation R

$r(R)$ is the rank of R

$q_s(R)$ is the number of satisfying rows in the truth table of R which contain s ones

$(\alpha)_\beta$ $\frac{\alpha!}{(\alpha-\beta)!}$, where α,β are positive integers, $\alpha \geq \beta$.

Let

$$\tau_\psi' = \min_{\substack{real \\ 1 \leq i \leq m}} t_{R_i} \qquad \max_{\substack{real \\ 0 \leq x \leq 1}} \sum_{i=1}^{m} t_{R_i} appSAT_x(R_i)$$

$$\sum_{i=1}^{m} t_{R_i} = 1 \,,\; t_{R_i} \geq 0$$

$$appSAT_x(R) = \sum_{s=0}^{r(R)} q_s(R)\, x^s (1-x)^{r(R)-s} \,.$$

Theorem 2.1

$$\tau_\psi = \tau_\psi' \,.$$

τ_ψ is defined as the solution of a discrete minimax problem since the maximization is over integers. However τ_ψ' is expressed as the solution of a continuous minimax problem since both the minimization and maximization are over reals. Furthermore the formula for τ_ψ' does not involve a limit operation. Therefore the definition of τ_ψ' is easier to evaluate.

We need the following definitions for the proof of theorem 2.1. Let S be a ψ-formula containing relation $R_i\,(1 \leq i \leq m)$ for the fraction t_{R_i} of the clauses. Let $\vec{t} = (t_{R_1}, \ldots, t_{R_m})$.

$$mean_k^n(\vec{t}) = \sum_{i=1}^{m} t_{R_i} SAT_k^n(R_i).$$

Let

$$appmean_x(\vec{t}) = \sum_{i=1}^{m} t_{R_i} appSAT_x(R_i).$$

Lemma 2.2

Let ψ be a finite set of m relations and let $\vec{t}=(t_{R_1}, \ldots, t_{R_m})$ be a vector whose components add up to 1.

(1) $\lim_{j \to \infty} mean_{j:k}^{j:n}(\vec{t}) = appmean_{\frac{k}{n}}(\vec{t})$

(2) For all real $x\,(0 \leq x \leq 1)$:

$$\lim_{n \to \infty} mean_{\lceil nx \rceil}(\vec{t}) = appmean_x(\vec{t})$$

Proof

(1)
We have to show that

$$\lim_{j \to \infty} \frac{(jk)_s\,(j\,(n-k))_{r-s}}{(jn)_r} = \left\{\frac{k}{n}\right\}^s \left\{1-\frac{k}{n}\right\}^{r-s}.$$

This follows from

$$\lim_{j \to \infty} \frac{(jk)(jk-1)\cdots(jk-s+1)}{(jn)(jn-1)\cdots(jn-s+1)} = \left\{\frac{k}{n}\right\}^s$$

and

$$\lim_{j \to \infty} \frac{(j\,(n-k))(j\,(n-k)-1)\cdots(j\,(n-k)-r+s-1)}{(jn-s)(jn-s-1)\cdots(jn-r+1)} = \left\{1-\frac{k}{n}\right\}^{r-s}.$$

(2)
Follows from

$$\lim_{n \to \infty} \frac{\lceil nx \rceil}{n} = x$$

and (1).

Proof of Theorem 2.1

We have to show that for any $\vec{t}$

$$A = \lim_{n \to \infty} \max_{\substack{0 \leq k \leq n \\ integer}} mean_k^n(\vec{t}) =$$

$$B = \max_{\substack{0 \leq x \leq 1 \\ real}} appmean_x(\vec{t}).$$

Let $x_{\max}$ be the maximal x in the definition of B. For each n, define $k(n)=\lceil x_{\max} \rceil$. Then by Lemma 2.2(2)

$$A \geq \lim_{n\to\infty} mean^{n}_{k(n)}(\vec{t})=appmean_{x_{\max}}(\vec{t})=B.$$

Hence $A \geq B$.

To show that $A \leq B$, define the sequence $k(n), n=1,2,\cdots$ such that $mean^{n}_{k(n)}(\vec{t}) = \max_{0\leq k\leq n} mean^{n}_{k}(\vec{t})$. Let $k'(n')$, n' ranging over an increasing subsequence of the natural numbers, be an infinite subsequence of $k(n)$ such that

$$\lim_{n'\to\infty} \frac{k'(n')}{n'} = x$$

for some real x. Then by Lemma 2.2(2)

$$A = \lim_{n'\to\infty} mean^{n'}_{k'(n')}(\vec{t}) = \lim_{n\to\infty} mean^{n}_{\lceil xn \rceil}(\vec{t}) = appmean_{x}(\vec{t}) \leq B.$$

3. Special (0,1)-Integer Programming

In this section we analyze special systems of linear equalities. The computation of the constant τ_{ψ} for this case requires new methods. One reason is that here we are dealing with sets of relations which contain r relations (r a variable) and not just two relations.

Let R_j be the relation of rank r which holds if exactly j of the r variables are true.

Theorem 3.1

Let $\psi = \{R_0, R_1, \ldots, R_r\}$. Then

(i) In any ψ-formula the fraction $\frac{1}{r+1}$ of the clauses can be satisfied.

(ii) There is a polynomial algorithm MAXMEAN* which finds an assignment satisfying at least the fraction $\frac{1}{r+1}$ of the clauses in a ψ-formula.

(iii) For any rational $\tau' > \frac{1}{r+1}$ the set of ψ-formulas having an assignment satisfying at least the fraction τ' of the clauses is NP-complete.

Proof of 3.1(i).

Since $q_s(R_j) = 0$ if $s \neq j$ and $q_j(R_j) = \binom{r}{j}$ we have

$$appmean_{x}(S) = \sum_{j=0}^{r} t_j \binom{r}{j} x^{j}(1-x)^{r-j} \quad .$$

By Theorem 1 it is sufficient to show that

$$\frac{1}{r+1} = t_j \left(0 \leq \min_{\substack{j \leq r \\ real}}\right) \quad 0 \leq \max_{\substack{x \leq 1 \\ real}} \quad appmean_x(\vec{t}).$$

$$\sum_{j=0}^{r} t_j = 1$$

ψ has the property that it is not necessary to choose the maximal x in the above formula in order to compute $\tau_\psi = \frac{1}{r+1}$. Therefore we perform an averaging process in the following lemma.

Lemma 3.2

$$\int_0^1 x^j (1-x)^{r-j}\, dx = \frac{1}{\binom{r}{j}(r+1)}$$

Proof

Let

$$f_{j,r} = \int_0^1 x^j (1-x)^{r-j}\, dx$$

We show the Lemma by induction.

$$f_{0,r} = \int_0^1 (1-x)^r\, dx = -\frac{1}{r+1}(1-x)^{r+1} \mid_0^1 = \frac{1}{r+1}$$

For the induction step we use partial integration

$$\begin{aligned} u &= \frac{1}{j+1} x^{j+1} \\ u' &= x^j \\ v &= (1-x)^{r-j} \\ v' &= -(r-j)(1-x)^{r-j-1} \\ f_{j,r} &= \int_0^1 u'\, v\, dx = u \cdot v \mid_0^1 - \int_0^1 uv'\, dx \\ &= \frac{r-j}{j+1} f_{j+1,r} \end{aligned}$$

Hence,

$$f_{j+1,r} = \frac{j+1}{r-j} f_{j,r} \; (0 \leq j < r)$$

and therefore inductively

$$f_{j+1,r} = \frac{j+1}{r-j} f_{j,r} = \frac{j+1}{r-j} \; \frac{j \cdot (j-1) \cdots 1}{r\,(r-1) \cdots (r-j+1)} \cdot \frac{1}{r+1} = \frac{1}{\binom{r}{j+1}} .$$

Lemma 3.3

Let

$$\vec{t} = (t_0, t_1, \ldots, t_r)\left(\sum_{j=0}^{r} t_j = 1\right)$$

and let

$$appmean_x \left\{ \vec{t} \right\} = \sum_{j=0}^{r} t_j \binom{r}{j} x^j (1-x)^{r-j} .$$

Then there is $x_0 (0 \leq x_0 \leq 1)$ such that

$$appmean_{x_0} \left\{ \vec{t} \right\} = \frac{1}{r+1} .$$

Proof

Consider

$$\int_0^1 appmean_x \left\{ \vec{t} \right\} dx$$

$$= \sum_{j=0}^{r} t_j \binom{r}{j} \int_0^1 x^j (1-x)^{r-j} \, dx$$

$$= \sum_{j=0}^{r} t_j \binom{r}{j} \frac{1}{\binom{r}{j}(r+1)} = \frac{1}{r+1} .$$

The claim follows from the mean value theorem of calculus.

Lemma 3.4

$$\min_{\vec{t}} \max_{0 \leq x \leq 1} appmean_x \left\{ \vec{t} \right\} = \max_{\vec{t}} \min_{0 \leq x \leq 1} appmean_x \left\{ \vec{t} \right\} = \frac{1}{r+1}$$

Proof

Let

$$\vec{b} = \left\{ \frac{1}{r+1}, \frac{1}{r+1}, \ldots, \frac{1}{r+1} \right\} (r+1 \textit{ dimensional}).$$

Then

$$appmean_x \left\{ \vec{b} \right\} = \frac{1}{r+1} \sum_{j=0}^{r} \binom{r}{j} x^j (1-x)^{r-j}$$

$$= \frac{1}{r+1}$$

Therefore

$$\min_{\vec{t}} \; 0 \leq x \leq 1 \; \max \; appmean_x \left\{ \vec{t} \right\} \leq \frac{1}{r+1} \;,$$

since for $\vec{t} = \vec{b}$ only the fraction $\frac{1}{r+1}$ can be satisfied (independent of x).

Also

$$\max_{\vec{t}} \; \min_{x} \; appmean_x \left\{ \vec{t} \right\} \geq \frac{1}{r+1} \;,$$

since for $\vec{t} = \vec{b}$ the minimal x satisfies the fraction $\frac{1}{r+1}$. On the other hand for any $\vec{t}$ there is an x_0 such that

$$appmean_{x_0} \left\{ \vec{t} \right\} = \frac{1}{r+1} \;.$$

Therefore

$$\min_{\vec{t}} \; 0 \leq x \leq 1 \; \max \; appmean_x \left\{ \vec{t} \right\} \geq \frac{1}{r+1}$$

and

$$\max_{\vec{t}} \; 0 \leq x \leq 1 \; \min \; appmean_x \left\{ \vec{t} \right\} \leq \frac{1}{r+1} \;.$$

Proof of 3.1(ii) and 3.1(iii).

Algorithm MAXMEAN* guarantees to satisfy the fraction $\frac{1}{r+1}$ in polynomial time. It follows from a general result in [Schaefer (1978)] that the ψ-satisfiability problem is NP-complete (for the ψ under discussion). Then (iii) follows from Theorem 1.2 in [Lieberherr (1982)].

4. Satisfiability

Let $F(p,q)$ be the following class of propositional formulas in conjunctive normal form: Each clause in a formula in $F(p,q)$ contains at least p positive or q negative literals $(p,q \geq 1)$.

Let α be the solution of $(1-x)^p = x^q$ in $(0,1)$ and let $\tau_{p,q} = 1 - \alpha^q$.

Theorem 4.1

(i) In any formula in $F(p,q)$ the fraction $\tau_{p,q}$ of the clauses can be satisfied.

(ii) There is a polynomial algorithm MAXMEAN* which finds an assignment satisfying at least the fraction $\tau_{p,q}$ of the clauses in a formula in $F(p,q)$.

(iii) For any rational $\tau' > \tau_{p,q}$ the set of formulas in $F(p,q)$ having an assignment satisfying at least the fraction τ' of the clauses is NP-complete.

This theorem and its proof extend the results and methods given in [Lieberherr/Specker (1981)]. The proof of Theorem 4.1(i) is given by a sequence of simplifying reductions. Each reduction is presented as a Proposition j. The corresponding Lemma j claims that Proposition j implies the previous proposition (in the first step: Theorem 4.1(i))

Theorem 4.1(ii) is a special case of a general result proven in [Lieberherr (1981)]. The proof of Theorem 4.1(iii) is based on a result by [Schaefer (1978)] and the technique given in [Lieberherr/Specker (1981)].

Simplifying Reductions

Proposition 4.2

For all integers $n > \min(p,q)$ and for all positive integers t_1, t_2 there is an integer k $(0 \leq k \leq n)$ such that $g_{EXACT}(n,k,t_1,t_2) =$

$$\frac{\frac{t_1}{(n)_p}(n-k)_p + \frac{t_2}{(n)_q}(k)_q}{t_1 + t_2} \leq 1 - \tau_{p,q}$$

Lemma 4.2

Proposition 4.2 ==> Theorem 4.1(i)

Proof

Using the techniques given in [Lieberherr/Specker (1981)] it is easy to show that the class $F(p,q)$ can be reduced to $F'(p,q) =$ {formulas having only clauses containing either exactly p positive literals or exactly q negative literals}. Furthermore, it is sufficient to consider only symmetric formulas in $F'(p,q)$.

Let S be a symmetric formula in $F'(p,q)$ which contains t_1 clauses of the form $A_1 \vee A_2 \vee \cdots \vee A_p$ and t_2 clauses of the form $not\ A_1 \vee not\ A_2 \vee \cdots \vee not\ A_q$. Then the fraction of unsatisfied clauses if k variables are set to 1 is, by elementary counting methods, given by $g_{EXACT}(n,k,t_1,t_2)$.

Note that $g_{EXACT}(n,k,t_1,t_2)$ is the expected fraction of unsatisfied clauses among all assignments which set k variables to 1. It is denoted by $mean_k'(S)$ for a given formula S.

First we give an outline of the proof for Proposition 4.2.

Outline:

Let $x = \frac{k}{n}$ and substitute r^s for any expression of the form $(r)_s$ in $g_{EXACT}(n,k,t_1,t_2)$. The resulting expression for the fraction of *unsatisfied* clauses is

$$g_{APPROX}(x,t_1,t_2) = \frac{t_1(1-x)^p + t_2 x^q}{t_1 + t_2} .$$

Since for all positive integers r,k,n $(k \leq n)$

$$\frac{(k)_r}{(n)_r} \leq \left(\frac{k}{n}\right)^r ,$$

the inequality

$$g_{EXACT}(n,k,t_1,t_2) \leq g_{APPROX}(\frac{k}{n},t_1,t_2)$$

holds. Therefore it is sufficient to show that for all n and all positive integers t_1,t_2 there is an integer k such that

$$g_{APPROX}(\frac{k}{n},t_1,t_2) \leq 1 - \tau_{p,q} .$$

W.l.o.g. we set $t_2 = 1$, since g_{APPROX} is homogeneous in t_1,t_2.

Take the derivative of g_{APPROX} with respect to x, set it to zero and solve for t_1:

$$t_1 = \frac{q}{p} \frac{x^{q-1}}{(1-x)^{p-1}}$$

Substitute for t_1 in g_{APPROX} :

$$g_{APP}(x) = \frac{q \cdot x^{q-1}(1-x)^p + p \cdot x^q (1-x)^{p-1}}{q \cdot x^{q-1} + p(1-x)^{p-1}} .$$

g_{APP} has the following intuitive meaning. Consider a formula S in $F'(p,q)$ with n variables and define $k_{\min}$ by

$$\min_{0 \leq k \leq n} mean_k'(S) = mean_{k_{\min}}'(S) .$$

In any such a formula S at most the fraction $g_{APP}(\frac{k_{\min}}{n})$ of the clauses can be unsatisfied. This holds since the second derivative of g_{APPROX} with respect to x is positive. if $p \geq 1$ or $q \geq 1$ and $t_1 \neq 0$ and $t_2 \neq 0$. Therefore it is sufficient to show that for all positive integers and all real x $(0 \leq x \leq 1)$

$$g_{APP}(x) \leq 1 - \tau_{p,q} .$$

Compute the extremal points of g_{APP} with respect to x in (0,1). There is only one which is given by the solution of $(1-x)^p = x^q$.
Substituting x^q for $(1-x)^p$ in g_{EXACT} yields

$$g_{EXACT} = 1 - x^q .$$

Therefore the fraction $\tau_{p,q} = 1 - \alpha^q$ can be satisfied in any formula in $F'(p,q)$.

The following simple heuristic method, which was also observed by John Scranton, gives the correct result.

Choose x such that the fraction of satisfied clauses is independent of t_1,t_2. The resulting condition for x is

$$(1-x)^p = x^q .$$

For such an x, the fraction of satisfied clauses is independent (in the limit) of the formula we consider and it is $\tau_{p,q}$.

Now we continue with the proof of Proposition 4.2.

Proposition 4.3

For all integers $n > \min(p,q)$ and all positive integers t_1, t_2 there is an integer $k\,(0 \leq k \leq n)$ such that

$$g_{APPROX}(x, t_1, t_2) = \frac{t_1(1-x)^p + t_2 x^q}{t_1 + t_2} \leq 1 - \tau_{p,q}$$

Lemma 4.3

Proposition 4.3 ==> Proposition 4.2

Proof

Note that for all positive integers $r, k, n\,(k \leq n)$ since

$$\frac{(k)_r}{(n)_r} \leq \left\{\frac{k}{n}\right\}^r$$

$$(1 - \frac{1}{k})(1 - \frac{2}{k})\cdots(1 - \frac{r-1}{k}) \leq (1 - \frac{1}{n})(1 - \frac{2}{n})\cdots(1 - \frac{r-1}{n}) .$$

If we let $x = \frac{k}{n}$ and replace $\frac{(n-k)_p}{(n)_p}$ by $(1-x)^p$ and $\frac{(k)_q}{(n)_q}$ by x^q we increase g_{EXACT}. Therefore $g_{EXACT} \leq g_{APPROX}$ which proves Lemma 4.3.

Proposition 4.4

For all real x $(0 \leq x \leq 1)$

$$g_{APP} = \frac{x^{q-1}(1-x)^{p-1}[q(1-x) + px]}{q \cdot x^{q-1} + p(1-x)^{p-1}} \leq \alpha^q$$

Lemma 4.4

Proposition 4.4 ==> Proposition 4.3

Proof

W.l.o.g. we set $t_2 = 1$ since g_{APPROX} is homogeneous in t_1, t_2. Take the derivative of g_{APPROX} with respect to x, set it to zero and solve for t_1:

$$t_1 = \frac{q}{p} \frac{x^{q-1}}{(1-x)^{p-1}}$$

If we substitute for t_1 in g_{APPROX} we get g_{APP}. Note that the second derivative of g_{APPROX} with respect to x is

$$t_1 \cdot p \cdot (p-1)(1-x)^{p-2} + t_2 \cdot q \cdot (q-1)x^{q-2},$$

which is positive for any $x\,(0 < x < 1)$, if $p > 1$ and $t_1 \neq 0$ or $q > 1$ and $t_2 \neq 0$.

Proof of Proposition 4.4

We show first that the derivative of $g_{APP}(x)$ is zero in (0,1) iff x satisfies $(1-x)^p = x^q$.

Let $A = x^q, B = (1-x)^p$. Then

$$g_{APP}(x) = \frac{A'B - AB'}{A' - B'}$$

The numerator of the derivative of $g_{APP}(x)$ is

$$(A''B' - A'B'')(A - B).$$

The first factor

$$A''B' - A'B''$$
$$= -pq(q-1)x^{q-2}(1-x)^{p-1} - q \cdot p(p-1)x^{q-1}(1-x)^{p-2}$$
$$= x^{q-2}(1-x)^{p-2}(-(q-1)(1-x) - (p-1)x)$$

has no zeros in (0,1).

Since $(1-x)^p = x^q$ has only one solution α in (0,1) the rational function $g_{APP}(x)$ has one extremal point in (0,1) with value $g_{APP}(\alpha) = \alpha^q$. Since $g_{APP}(0) = g_{APP}(1) = 0$ the function $g_{APP}(x)$ is maximal for $x = \alpha$.

The proof of Proposition 4.4 uses differentiation. We give now a different proof which does not use differentiation and which provides further insight into the problem.

Proposition 4.5

For all real x, β $(0 \leq x, \beta \leq 1)$

$$g_2(x,\beta) = x^q \left[(1-x)^p (px - qx + q) + (1-\beta)^p q(x-1)\right] - (1-x)^p \beta^q \cdot p \cdot x \leq 0$$

Lemma 4.5

Proposition 4.5 ===> Proposition 4.4

Proof

Multiply both sides of $g_{APP}(x) \leq \alpha^q$ by the denominator of $g_{APP}(x)$ and shift all terms to the left of the inequality sign. The resulting inequality is $g_2(x,\beta) \leq 0$ if we make liberal use of $(1-\alpha)^p = \alpha^q$ (a crucial point) and if we substitute β for α.

Proof of Proposition 4.5

So far a proof of Proposition 4.5 was obtained only for special cases.

I) $p=1, q \geq 1$.

Note that

$$g_2(x,\beta) = (x-\beta)^2 \sum_{i=1}^{q-1} x^{q-i-1}(i-q)\beta^{i-1}$$

Hence $g_2(x,\beta)$ is non-positive for $0 \leq x, \beta \leq 1$.

Example: $(p=1)$

$g_2(x,\beta)$ is proportional to (the deleted factor has a positive sign)

$-(x-\beta)^2$ for $q=2$.

$-(x-\beta)^2(2x+\beta)$ for $q=3$.

$-(x-\beta)^2(3x^2+2x\alpha+\alpha^2)$ for $q=3$.

II) $p=2$.

$g_2(x,\beta)$ is proportional to (the deleted factor has a positive sign)

$(x-\beta)^2(x(x-4)+2\beta(x-1))$ for $q=3$.

$(x-\beta)^2(x^2(x-3)+2\beta x(x-1)+\beta^2(x-1))$ for $q=4$.

$(x-\beta)^2(x^3(3x-8)+6\beta x^2(x-1)+4\beta^2 x(x-1)+2\beta^3(x-1))$ for $q=5$.

Unfortunately this technique does not generalize for $p \geq 3$ but it is conjectured that Proposition 4.5 holds in general.

The formulas obtained by the alternate proof method have interesting applications. The following theorem allows us to predict which fraction of the clauses can be satisfied in every formula if the index of the maximal $mean_k(S)$ is fixed.

Theorem 4.6

Let S be a formula in $F'(1,q)(q>1)$ for which

$$\max_{0 \leq k \leq n} mean_k(S) = mean_0(S).$$

Then assignment $J_{ALL\ 0}$, which assigns false to all variables satisfies all clauses. In general, if $\max_{0 \leq k \leq n} mean_k(S) = mean_{k'}(S)$ then the assign-

ment which assigns true to k' variables satisfies at least the fraction

$$1 - \alpha^q - \frac{(\frac{k'}{n} - \alpha)^2 \sum_{i=1}^{q-1} (i-q)\alpha^{i-1}(\frac{k'}{n})^{q-i-1}}{q \cdot (\frac{k'}{n})^{q-1} + 1}$$

of the clauses.

Proof

Consider $g_{APP}(x) - \alpha^q$ for $p=1$ (after multiplying with $q \cdot x^{q-1} + 1$):

$$qx^{q-1}(1-x) + x^q - \alpha^q \cdot (q \cdot x^{q-1}+1) = (x-\alpha)^2 \sum_{i=1}^{q-1} x^{q-i-1}(i-q)\alpha^{i-1}.$$

Let

$$h(x) = \frac{(x-\alpha)^2 \sum_{i=1}^{q-1} x^{q-i-1}(i-q)\alpha^{i-1}}{q \cdot x^{q-1}+1} .$$

Now,

$$h(0) = \alpha^2(-1 \cdot \alpha^{q-2}) = -\alpha^q .$$

Proof of Theorem 4.1(ii)

Algorithm MAXMEAN* guarantees to satisfy the fraction $\tau_{p,q}$ in polynomial time.

Proof of Theorem 4.1(iii)

The fact that the satisfiability problem for formulas in $F(p,q)$ is NP-complete follows from a general result of [Schaefer (1978)]. Then the proof can be adapted from [Lieberherr/Specker (1981)].

Extensions

The technique used to prove Theorem 4.1(i) is suitable to determine τ_ψ for other sets ψ which contain only two relations. The fraction of satisfied clauses in a symmetric formula which contains t_1 clauses with the first relation and t_2 clauses with the second relation is given by (in approximated form)

$$h_1(x, t_1, t_2) = \frac{t_1 R_1(x) + t_2 R_2(x)}{t_1 + t_2} ,$$

where R_1 and R_2 are polynomials which depend on the two relations.

W.l.o.g. $t_2 = 1$. If we take the derivative of h_1 with respect to x and solve for t_1 we get

$$t_1 = \frac{-R_2'(x)}{R_1'(x)} .$$

Substituting in h_1 we get

$$h_2(x) = \frac{R_1'(x)\cdot R_2(x) - R_1(x)R_2'(x)}{R_1'(x) - R_2'(x)} .$$

The numerator of the derivative of $h_2(x)$ is given by

$$(R_1(x) - R_2(x))(R_1''(x)R_2'(x) - R_1'(x)R_2''(x)) .$$

If the second factor has no zeros in (0,1) then the fraction $R_1(\alpha)$ can always be satisfied, where α is the solution of $R_1(x) = R_2(x)$ in (0,1) which is the global minimum of h_2.

5. Partial Solution of the 3-Satisfiability Problem

In [Lieberherr/Specker (1981)] the following problem was left open. A formula S of the propositional calculus in conjunctive normal form (cnf) is said to be *3-satisfiable,* if any triple of clauses is satisfiable. We find a lower bound on the fraction τ_3 of the clauses which can always be satisfied in a 3-satisfiable formula by showing in the following that $\tau_3 \geq 2/3$. Unfortunately we have not been able to determine τ_3 exactly. The motivation for studying *k-satisfiable* formulas is the relationship to polynomial approximation schemes for satisfiability [Lieberherr/ Specker (1981), Huang/ Lieberherr (1981)].

The problem with 3-satisfiable formulas is that they are not closed under symmetrization. If we take a 3-satisfiable formula S and symmetrize it with the full permutation group then the symmetrized formula is in general not 3-satisfiable.

To show that $\tau_3 \geq 2/3$ we construct a class RED_1 of formulas so that

1. RED_1 contains all 3-satisfiable formulas (but some are not 3-satisfiable)
2. in any formula in RED_1 at least the fraction 2/3 of the clauses can be satisfied.

Consider any 3-satisfiable formula S. Without loss of generality we assume that clauses of length 1 only contain positive literals (this can be enforced by renamings). Now we partition the variables into two classes. The first class contains only variables which occur in clauses of length 1. The second class contains all other variables. A clause is said to be type T_{ij}^{q} if its j variables are in class q and i of them are positive. A clause is said to be of

type $T^{qr}_{i_1 j_1 i_2 j_2}$ if it contains j_1 variables of class q and j_2 variables of class r and if i_1 of the j_1 variables are positive and i_2 of the j_2 variables are positive.

Definition

RED_1 is the following subset of cnfs: The variables are partitioned into 2 classes (A-variables and B-variables) and only the following clause types occur:

$$T^{1}_{11}, T^{1}_{03}, T^{12}_{01\,11}, T^{12}_{01\,01}, T^{2}_{02}, T^{2}_{12}, T^{2}_{22} .$$

This definition is of interest since for proving that $\tau_3 \geq 2/3$ it is sufficient to minimize among the formulas in RED_1.

Theorem 5.1

(i) In any 3-satisfiable cnf at least the fraction 2/3 of the clauses can be satisfied. (ii) There is a polynomial algorithm to find such an assignment.

Proposition 5.2

In any cnf in RED1 at least the fraction 2/3 of the clauses can be satisfied.

Lemma 5.2

Proposition 5.2 ==> Theorem 5.1(i)

Proof

Any 3-satisfiable cnf is easily reduced to a formula in RED_1 by deleting literals. Deleting literals makes a formula harder for satisfying many clauses.

Definition:

Let RED_2 be the subset of cnfs of RED_1 which do not contain clauses with types T^{2}_{02}, T^{2}_{12} and T^{2}_{22}.

Proposition 5.3

In any cnf in RED2 at least the fraction 2/3 of the clauses can be satisfied.

Lemma 5.3

Proposition 5.3 ==> Proposition 5.2

Proof

In a cnf containing clauses of exactly length 2 at least the fraction 3/4 of the clauses can be satisfied (a random assignment satisfies 3/4). Therefore deleting clauses of the above three types does not make it easier to satisfy many clauses.

We prove now Proposition 5.3 by a sequence of further reductions. Let S be a formula in RED_2 which contains t_1 clauses of type T^1_{11}, t_2 clauses of type $T^{12}_{01\,11}$, t_3 clauses of type T^1_{03} and t_4 clauses of type $T^{12}_{01\,11}$. The worst-case formulas (regarding the fraction of satisfiable clauses) in RED_2 are those which are symmetric in the A-variables and B-variables. Among those formulas the formulas with $t_2 = t_4$ are hardest. In a formula in RED_2 with $t_2 = t_4$ the fraction

$$1 - \frac{\frac{t_1}{n}(n-k) + \frac{t_2}{n}k + \frac{t_3}{\binom{n}{3}}\binom{k}{3}}{t_1 + 2t_2 + t_3}$$

of the clauses are satisfied if k of the n A-variables are set to 1. Therefore, we have to show

Proposition 5.4

For all integers n and for all positive integers t_1, t_2, t_3 there is an integer k $(0 \leq k \leq n)$ such that

$$\frac{\frac{t_1}{n}(n-k) + \frac{t_2}{n}k + \frac{t_3}{(n)_3}(k)_3}{t_1 + 2t_2 + t_3} \leq \frac{1}{3}.$$

Lemma 5.4

Proposition 5.4 ==> Proposition 5.3

Proof

Given above.

Proposition 5.5

For all integers n and for all positive integers t_1, t_2, t_3 there is an integer $k\,(0 \leq k \leq n)$ such that

$$w_1 = \frac{\frac{t_1}{n}(n-k) + \frac{t_2}{n}k + \frac{t_3}{n^3}k^3}{t_1 + 2t_2 + t_3} \leq \frac{1}{3}.$$

Lemma 5.5

Proposition 5.5 ==> Proposition 5.4

Proof

Observe that $\frac{(k)_r}{(n)_r} \leq \frac{k^r}{n^r}$ if $k \leq n$.

Proposition 5.6

For all x $(0 \leq x \leq 1)$ and all positive integers t_3

$$w_2 = \frac{1 - 2t_3x^3}{3 + t_3(1-6x^2)} \leq \frac{1}{3}.$$

Lemma 5.6

Proposition 5.6 ==> Proposition 5.5

Proof

W.l.o.g. let $t_1 = 1$ and substitute x for $\frac{k}{n}$ in w_1. Take the derivative of w_1 with respect to x, set it to zero and solve for t_2:

$$t_2 = 1 - 3t_3x^2.$$

By substituting $1 - 3t_3x^2$ for t_2 in w_1 we get w_2.

Proposition 5.6 is easily proven directly by case analysis.

References

Huang1981a.

M.A. Huang and K.J. Lieberherr, "Towards an approximation scheme for generalized satisfiability," *Report 292, Dep. of EECS, Princeton University*, 1981.

Lieberherr1981a.

K. Lieberherr and E. Specker, "Complexity of partial satisfaction," *Journal of the ACM*, vol. 28, no. 2, pp. 411-421, 1981.

Lieberherr1982a.

K.J. Lieberherr, "Algorithmic extremal problems in combinatorial

optimization," *Journal of Algorithms*, vol. 3, pp. 225-244, 1982.

Schaefer1978a.

T. Schaefer, "The complexity of satisfiability problems," *Proc. 10th Annual ACM Symposium on Theory of Computing*, pp. 216-226, 1978.