Complexity of Partial Satisfaction

K. J. LIEBERHERR AND E. SPECKER

Swiss Federal Institute of Technology, Zurich, Switzerland

ABSTRACT A conjunctive-normal-form expression (cnf) is said to be 2-satisfiable if and only if any two of its clauses are simultaneously satisfiable It is shown that every 2-satisfiable cnf has a truth assignment that satisfies at least the fraction h of its clauses, where $h = (\sqrt{5} - 1)/2 \sim 0.618$ (the reciprocal of the "golden ratio"). The proof is constructive in that it provides a polynomial-time algorithm that will find for any 2-satisfiable cnf a truth assignment satisfying at least the fraction h of its clauses. Furthermore, this result is optimal in that the constant h is as large as possible. It is shown that, for any rational $h' > h$, the set of all 2-satisfiable cnfs that have truth assignments satisfying at least the fraction h' of their clauses is an NP-complete set

KEY WORDS AND PHRASES doubly transitive permutations, golden mean, NP-complete, polynomial enumeration algorithm, polynomially constructive reductions, satisfiability

CR CATEGORIES 5.21, 5.25, 5.39

1. *Introduction*

The inefficiency of the known optimization algorithms for many optimization problems (especially those which are NP-complete [2, 8]) has stimulated research into the possibilities of proving "performance guarantees" for simple and efficient heuristic algorithms. The following "performance guarantee" for polynomial approximation algorithms has been studied extensively in the literature [5, 6]: The guarantee is quantified in terms of optimal solutions, stating that a particular algorithm constructs solutions that never differ in value from optimal by more than some fixed constant or by some constant percentage of the optimum value. When examples can be constructed that cause the algorithm to deviate from optimal by the maximal amount allowed by a proven performance bound, we may say that the "worst-case performance" of the algorithm is known exactly.

Given this type of analysis, one might hope to classify problems by the nature of the best performance bounds known for them. However, rather than provide us with a meaningful absolute ranking for any problem, such a classification may merely reflect the limitations of our current knowledge [5]. There might yet be undiscovered polynomial approximation algorithms that provide guarantees better than those presently known.

This paper presents a polynomial-time approximation algorithm for which the performance guarantee is provably the best possible (assuming that P $\neq$ NP) among the class of polynomial algorithms. We call such an algorithm P-optimal.

Authors' present addresses K. J Lieberherr, Department of Electrical Engineering and Computer Science, Princeton University, Princeton, NJ 08544, E. Specker, Department of Mathematics, Swiss Federal Institute of Technology, ETH Zentrum, Ch-8092 Zurich, Switzerland.

Informally, a polynomial-time approximation algorithm is *P-optimal* if the problem of guaranteeing better approximate solutions than those produced by the algorithm is NP-complete. This implies that if a polynomial approximation algorithm B is P-optimal and P $\neq$ NP, then there exists no polynomial-time algorithm which guarantees more than B.

A *conjunctive-normal-form expression* (cnf) is a finite sequence of clauses, with repetitions allowed, where each *clause* is a disjunction of different literals (a *literal* is either a variable A or its negation A').

An *interpretation* of a cnf s is an assignment of truth values ("true" and "false") to the variables of s. A cnf is *satisfied* by an interpretation iff every clause contains either a variable A to which true is assigned or a negated variable B' such that false is assigned to B.

If the sequence s is

$$\{A \vee B, A \vee B', A' \vee B'\},$$

it is satisfied by the assignment of true to A and false to B. There are, of course, cnfs which are not satisfiable. The simplest case is a cnf in which one clause is a variable—say A—and the other is its negation A'. The following is an example of an unsatisfiable but 2-satisfiable cnf:

$$\{A, B, C, A' \vee B' \vee C'\}.$$

This cnf contains a satisfiable subsequence which consists of three clauses.

This paper deals with the "maximum satisfiability" problem of [7]: Given a conjunctive-normal-form expression (cnf), with repeated clauses allowed, find a truth assignment that satisfies a maximum number of the clauses. Algorithm B2 in [7] satisfies at least $(|s| - \mathit{weight}(s))$ clauses of a cnf s [10]. *Weight*(s) is the sum of the weights of the clauses in s, and the weight of a clause c is $2^{-|c|}$, where $|c|$ is the number of literals in c. In [7] it is shown that there exist cnfs s in which $|s| - \mathit{weight}(s)$ is the *maximum* number satisfiable.

The current paper considers an algorithmic technique (called symmetrization) which allows a finer analysis. In the case of symmetrization, the worst-case performance bound depends not only on the lengths of the clauses in the input but also on the number of negated variables in each clause. We apply symmetrization to a special class, the "2-satisfiable" cnfs. In a 2-satisfiable cnf, unary clauses are allowed, but if clause A is present, then clause A' is forbidden. From a result in [7] or the analysis mentioned above we would expect that in every 2-satisfiable cnf we can satisfy at least the fraction $\frac{1}{2}$ of the clauses but not more than $\frac{3}{4}$. Indeed, the answer turns out to be ~0.618, the reciprocal of the golden ratio.

The symmetrization technique provides both a proof of this result and an efficient algorithm. Moreover, although the 2-satisfiable cnfs are mainly of theoretical interest, our computer experience with symmetrization indicates that this technique may also be a valuable practical method in general.

The main theoretical results are

THEOREM 1. *For a 2-satisfiable cnf s there exists a satisfiable subsequence of s that has at least $h \cdot |s|$ clauses (where h is the reciprocal of the golden mean, $h^2 + h - 1 = 0$, $h > 0$, and $|s|$ is the number of clauses in s).*

THEOREM 2. *Let h_0 be a number greater than h. Then there exists a 2-satisfiable cnf s containing no satisfiable subsequence of s that has at least $h_0 \cdot |s|$ clauses.*

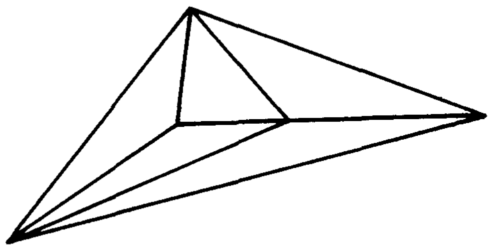

FIG 1 Graph G

COROLLARY 1. *There exists a polynomial algorithm (called ENUMERATE) which finds an interpretation J for every 2-satisfiable cnf s such that J satisfies at least* $h \cdot |s|$ *clauses.*

COROLLARY 2. *For any rational* h', $h < h' \le 1$, *the set of 2-satisfiable cnfs which have an interpretation satisfying the fraction* h' *of the clauses is NP-complete.*

Symmetrization as a (non-polynomial-time) constructive reduction is a well-known technique; for example, it is used in [3] for coloring problems of hypergraphs. Symmetrization is an instance of the following general technique: Given a problem Γ which has to be solved either exactly or approximately, transform Γ to a "simpler" problem so that the solution of the "simpler" problem easily allows solution of the original problem. Symmetrization is unusual in the sense that it simplifies by making larger.

Example. Consider the graph G (Figure 1) with five nodes and nine edges. The problem is to color this graph with three colors so that the fraction of the edges satisfying the coloring condition (adjacent nodes have different colors) is "close" to maximum. The "simpler" problem to which this graph is transformed is the complete graph with five nodes. It is easy to compute an optimal coloring with m colors for the complete graph with n nodes. Let $r = n \bmod m$. The fraction of the edges for which the coloring condition is satisfied by the optimum coloring with m colors is

$$d(n, m) = \frac{n^2(m-1) + r(r-m)}{m \cdot n \cdot (n-1)}.$$

Hence $d(5, 3) = \frac{4}{5}$. The simple, but crucial step follows now: If the optimum for the complete graph with five nodes is $\frac{4}{5}$, then there is a solution for the given problem such that the coloring condition is satisfied for at least the fraction $\frac{4}{5}$ of the edges (i.e., eight edges). This can be proved by considering the complete graph as the "overlapping" of 5! permutations of the original graph.

The remainder of the paper is organized as follows. In Section 2 we reduce Theorem 1 in two steps to a simpler form which can be proved directly; step 2 applies symmetrization. In the last part of Section 2 we prove Theorem 2. In Section 3 we present and analyze algorithm ENUMERATE, which applies symmetrization in a polynomially constructive form. We show that Algorithm B2 of [7] is in general unable to provide the best guarantee in the 2-satisfiable case. Section 4 contains the proof of the NP-completeness result, and the paper concludes with some open problems.

2. *Reductions*

We use three reductions to prove Theorem 1. For the definition of the reductions we need the following notions. The *length* of a clause is the number of occurrences of literals in it. A literal is said to be *positive* if it is a variable; otherwise it is said to be *negative*. We say that a cnf s has property GM if s has an interpretation which satisfies at least $h \cdot |s|$ clauses.

Theorem 1 states that each 2-satisfiable cnf has property GM. In the following we reduce the set of all 2-satisfiable cnfs to a set RED1, so that RED1 has property GM iff all 2-satisfiable cnfs have property GM.

Definition 1. RED1 is the subset of 2-satisfiable cnfs with the following properties:

(1) The clauses of length 1 only contain positive literals.
(2) The clauses of length 2 only contain negative literals.
(3) There are no other clauses.

PROPOSITION 1. *Each cnf in RED1 has property GM.*

LEMMA 1. *Proposition* 1 $\Rightarrow$ *Theorem* 1.

PROOF. Let s be an arbitrary 2-satisfiable cnf. We simplify s to a cnf $T(s)$ according to the following rules:

(1) For each variable L which occurs negated in a clause of length 1, replace all occurrences of the literals L and L' by their complements.
(2) In clauses containing positive literals, drop all but one positive literal.
(3) In clauses containing only negative literals, but more than one, drop all except two.

$T(s)$ is in RED1, and to each interpretation $I1$ of $T(s)$ corresponds an interpretation I of s which satisfies at least as many clauses in s as $I1$ in $T(s)$. Hence, if each cnf $T(s)$ in RED1 has property GM, each 2-satisfiable cnf s has property GM. □

A cnf s in RED1 can be described as follows: s contains n variables $V_1, V_2, \ldots, V_n$. The clause V_i of length 1 occurs x_i times. The clauses $V_j' \vee V_k'$ of length 2 occur $y_{j,k}$ times ($j < k$). (The clause $V_k' \vee V_j'$ ($j < k$) is identified with the clause $V_j' \vee V_k'$.) Hence a cnf of RED1 is determined by

(1) a natural number n (number of variables);
(2) n numbers $x_1, x_2, \ldots, x_n$ (repetition factors for the clauses of length 1);
(3) $\binom{n}{2}$ numbers $y_{12}, y_{13}, \ldots, y_{n-1,n}$ (repetition factors for the clauses of length 2).

In the following we define a subset RED2 of RED1 with the property that RED2 has property GM iff RED1 has property GM.

Definition 2. RED2 is the subset of cnfs of RED1 with the following properties:

(1) $x_1 = x_2 = \cdots = x_n$; that is, all x_j ($1 \leq j \leq n$) are equal.
(2) $y_{12} = y_{13} = \cdots = y_{n-1,n}$; that is, all $y_{j,k}$ ($1 \leq j < n, j < k \leq n$) are equal.

PROPOSITION 2. *Each cnf s in RED2 has property GM.*

LEMMA 2. *Proposition* 2 $\Rightarrow$ *Proposition* 1.

PROOF. This proof uses symmetrization. Let s be a cnf in RED1, and let W be a set which contains the variables of s. We construct a symmetrized cnf s' in RED2 which contains the variables in W, such that s' has property GM iff s has property GM.

The construction of s' is based on a doubly transitive permutation group PG of W. A permutation group is said to be *doubly transitive* if it is transitive on the ordered tuples [1, p. 139] (i.e., for all $A1$, $A2$, $B1$, $B2$ in W ($A1 \neq A2$, $B1 \neq B2$) there is a permutation Π in PG such that $\Pi(A1) = B1$ and $\Pi(A2) = B2$). The full permutation group which will be used in this proof is an example of a doubly transitive group.

To an element Π of PG we associate the cnf $\Pi(s)$ which is defined as the result of substituting $\Pi(A)$ for A (for all A in W). We define $S[\text{PG}](s)$ to be the concatenation of the sequences $\Pi(s)$ for all Π in PG. Let J be an interpretation of $S[\text{PG}](s)$ which satisfies the fraction h' of the clauses in $S[\text{PG}](s)$. It is obvious that there is at least one element Π in PG such that J satisfies at least the fraction h' of the clauses of $\Pi(s)$. The interpretation J', defined by $J' = J \circ \Pi^{-1}$ (i.e., $J'(A) = J(\Pi^{-1}(A))$) satisfies the same number of clauses of s as J satisfies clauses of $\Pi(s)$.

It remains to be shown that $S[\text{PG}](s)$ is an element of RED2. We use the following fact about permutation groups, which the reader can readily verify.

FACT 1. *Let PG be a doubly transitive permutation group on a set W, and let* $V1$, $V2$ *be elements of W* ($V1 \neq V2$). *Let* $g1[PG]$ *be the number of elements* Π *of PG such that* $\Pi(V1) = V1$, *and let* $g2\,[PG]$ *be the number of elements* Π *of PG such that* $\Pi(V1) = V1$ *and* $\Pi(V2) = V2$. *Then for all elements X, Y in W, there are exactly* $g1[PG]$ *elements* Π *of PG such that* $\Pi(X) = Y$. *Moreover, for all pairs* $(X1, X2)$, $(Y1, Y2)$ ($X1 \neq X2$, $Y1 \neq Y2$) *there are exactly* $g2\,[PG]$ *elements* Π *of PG such that* $\Pi(X1) = Y1$ *and* $\Pi(X2) = Y2$ [1, *Ch.* 39].

An immediate consequence of Fact 1 is the following. Let s be a cnf in RED1, and let W be a set which contains the variables of s. Assume that x clauses of s are of length 1 and that y clauses are of length 2. Let PG be a doubly transitive permutation group on W. Then the following hold for $S[\text{PG}](s)$:

(1) For each A in W, the clause A occurs $x \cdot g1[\text{PG}]$ times.
(2) For each pair A, B of different elements of W, the clause $A' \vee B'$ occurs $2 \cdot y \cdot g2[\text{PG}]$ times.

The factor 2 appears since the clauses $A' \vee B'$ and $B' \vee A'$ are identified.

Thus $s[\text{PG}](s)$ is an element of RED2 and the lemma is proved. □

PROPOSITION 3. *For all natural numbers* $n > 1$ *and all positive rational numbers a, there exists a natural number k* ($0 \leq k \leq n$) *such that*

$$\frac{k \cdot a + \binom{n}{2} - \binom{k}{2}}{n \cdot a + \binom{n}{2}} > h. \tag{1}$$

LEMMA 3. *Proposition* 3 $\Longrightarrow$ *Proposition* 2.

PROOF. A cnf s in RED2 is described by natural numbers n, x, and y, where n is the number of variables, x is the multiplicity of the clauses of length 1, and y is the multiplicity of the clauses of length 2. Note that there is an interpretation satisfying all literals if $y = 0$ or $n = 1$; so assume $y > 0$ and $n > 1$. If k is the number of variables in s which are set true, then

$$k \cdot x + \left(\binom{n}{2} - \binom{k}{2}\right) \cdot y$$

clauses are satisfied. The total number of clauses in s is

$$n \cdot x + \binom{n}{2} \cdot y.$$

We define $a = x/y$. This is well defined, since we assumed $y > 0$. The fraction of clauses satisfied is then $(k \cdot a + \binom{n}{2} - \binom{k}{2})/(n \cdot a + \binom{n}{2})$. Hence a cnf s (in RED2) described by the numbers n, x, and y has property GM iff (1) is true. □

PROOF OF PROPOSITION 3. We only use elementary calculus.

If $a \geq n - 1$, choose $k = n$; then (1) is satisfied. In the following we assume that $a < n - 1$. We put

$$f(n, a, k) = \frac{k \cdot a + \binom{n}{2} - \binom{k}{2}}{n \cdot a + \binom{n}{2}}.$$

Then Proposition 3 is equivalent to the following: For the solution $h[n]$ of the min-max problem

$$\min_{\substack{0<a<n-1 \\ a \text{ rational}}} \max_{\substack{0\leq k\leq n \\ k \text{ integer}}} f(n, a, k),$$

the inequality $h < h[n]$ holds. First we give an intuitive pseudoproof. In $f(n, a, k)$ we replace the expression $\binom{n}{2}$ by $n^2/2$ and $\binom{k}{2}$ by $k^2/2$ and call the resulting function $f_1(n, a, k)$. Hence

$$f_1(n, a, k) = \frac{2 \cdot k \cdot a + n^2 - k^2}{2 \cdot n \cdot a + n^2}.$$

$f_1(n, a, k)$ as a function of k is maximal if $k = a$. We substitute a for k in $f_1(n, a, k)$ and obtain

$$f_2(n, a) = \frac{a^2 + n^2}{2 \cdot n \cdot a + n^2}.$$

Note that

$$f_2(n, a) \geq \min_{\substack{n>0 \\ a\geq 0}} f_2(n, a) = h.$$

The minimum is reached for $a/n = h$.

To prove Proposition 3, we observe that

$$f(n, a, k) > h$$

iff

$$-k^2 + k \cdot (1 + 2 \cdot a) + n \cdot (n - 1) \cdot (1 - h) - 2 \cdot n \cdot h \cdot a > 0. \tag{2}$$

Let k_1 and k_2 be the two solutions of the quadratic polynomial in k on the left side of inequality (2). Note that the average $(k_1 + k_2)/2 = a + 1/2$. For this possibly nonintegral value of k, it is easy to show that (2) holds. Let

$$\begin{aligned} d(n, a) &= |k_1 - k_2| \\ &= ((1 + 2 \cdot a)^2 + 4 \cdot n \cdot (n - 1) \cdot (1 - h) - (8 \cdot n \cdot h \cdot a))^{1/2}. \end{aligned}$$

If $d(n, a) > 1$ ($n > 1$, $a \geq 0$), then there exists at least one integer k for which (1) holds. Therefore we prove that $d(n, a) > 1$ if $n > 1$. The minimum of $d(n, a)$ with respect to a is at

$$a_{\min} = n \cdot h - \tfrac{1}{2}.$$

We replace a in $d(n, a)$ by $a_{\min}$ and, by making liberal use of the identity $h^2 + h - 1 = 0$, we obtain a function

$$d_1(n) = \sqrt{4 \cdot n \cdot h - 4 \cdot n \cdot h^2}.$$

The identity $h \cdot (1 - h) = h^3$ implies that

$$d_1(n) = \sqrt{4 \cdot n \cdot h^3}.$$

Note that $d_1(n) > 1$ if $n > 1$. Therefore, if $n > 1$, the following holds for all real a: $1 < d_1(n) \leq d(n, a)$. Hence k can be chosen in the interval $J = [a + \frac{1}{2} - z, a + \frac{1}{2} + z]$, where $z = (n \cdot h^3)^{1/2}$.

J contains at least one integer if $n > 1$. Moreover, the breadth of the interval is proportional to $\sqrt{n}$ which shows that we have much freedom in choosing k. This remark (with the above proof showing it) is due to S.E. Knudsen. □

PROOF OF THEOREM 2. It is sufficient to define a sequence $s_1, s_2, \ldots, s_n, \ldots$ of 2-satisfiable cnfs such that the fraction h_n of satisfiable clauses tends to h for $n \to \infty$. Given such a sequence S and a number h_0 ($h < h_0 \leq 1$), there are infinitely many cnfs s in S such that s contains no satisfiable subsequence having $h_0 \cdot |s|$ or more clauses.

The sequence S we use contains cnfs which are all in RED2. Hence an element of S is described by a natural number n (the number of variables) and a rational number a (the quotient of the multiplicity of the clauses of length 1 and the multiplicity of the clauses of length 2). We give a sequence of rational numbers $a_2, a_3, \ldots, a_n, \ldots$ which describes a sequence S of cnfs of the reduced type within a constant multiple of the multiplicities.

We set

$$a_n = n \cdot \frac{F_n}{F_{n+1}},$$

where F_n is the nth Fibonacci number ($n \geq 1$, $F_1 = 1$, $F_2 = 1$, $F_n = F_{n-1} + F_{n-2}$). Observe that

$$\lim_{n \to \infty} \frac{a_n}{n} = h.$$

To motivate the definition of the sequence S, we recall that the minimum

$$\min_{\substack{n>0 \\ a \geq 0}} \frac{a^2 + n^2}{2 \cdot n \cdot a + n^2} = h$$

is reached for $a/n = h$. It is easy to check that for this sequence S, the maximal fraction of clauses which can be satisfied converges to h.

Therefore, for all $h_0 > h$ there exist infinitely many 2-satisfiable cnfs s containing no satisfiable subsequence of s which has at least $h_0 \cdot |s|$ clauses. □

3. *Algorithm ENUMERATE*

Theorem 1 guarantees for every 2-satisfiable cnf s an interpretation satisfying at least the fraction h of the clauses. However our proof is not polynomially constructive, since we used the full permutation group for symmetrizing. For the full permutation group the order (group size $= n!$) is an exponential function of the degree (set size $= n$). Fortunately there are sufficiently many "small" doubly transitive permutation groups for which the order is bounded by a small polynomial in the degree.

If n is a prime p, then there exists a well-known doubly transitive permutation group DT[p], the group of linear applications of the Galois field GF(p). Consider the $p \cdot (p - 1)$ permutations DT[p](q, r) on the set $\{1, 2, \ldots, p\}$, which are defined as follows:

$$\mathrm{DT}[p](q, r) = \mathrm{LAMBDA}\ i((q \cdot i + r) \bmod p),$$

where $1 \le q \le p-1$, $1 \le r \le p$, and $1 \le i \le p$. The reader may readily verify that DT[p] is indeed a doubly transitive permutation group [1, Ch. 67].

Let s be a cnf in RED2 which contains n variables. Let p be the first prime greater than or equal to n. By the Postulate of Bertrand [9, p. 22] we know that $p < 2 \cdot n$. Using the group DT[p] for symmetrizing s, we have to check at most $p \cdot (p-1)$ interpretations in order to find an interpretation which satisfies at least the fraction h of the clauses in s.

Let W be a set of n variables, and let p be the first prime greater than or equal to n. We describe a set $I(n)$ of interpretations of W which has the following properties:

(1) Cardinality($I(n)$) $\le p \cdot (p-1) \cdot (p+1) < p^3$.
(2) For every 2-satisfiable cnf s with variables in W there is an interpretation in $I(n)$ which satisfies at least $h \cdot |s|$ clauses in s.

Set

$$I(n) = \bigcup_{k=0}^{p} \bigcup_{q=1}^{p-1} \bigcup_{r=1}^{p} \mathrm{INT}(n, k, q, r),$$

where the interpretations INT(n, k, q, r) are defined in the following way. Let RP[k] be an arbitrary permutation of p variables. Then variable V_i in W $(1 \le i \le n)$ is set true by the interpretation INT(n, k, q, r) iff

$$((q \cdot (\mathrm{RP}[k](i)) + r) \bmod p) \le k.$$

Otherwise V_i is set false.

Note that we have defined a large number of different sets $I(n)$ because RP[k] can be chosen arbitrarily.

LEMMA 4. *Let s be a 2-satisfiable cnf in which each clause of length* 1 *contains a positive literal, and let n be the number of variables in s. Then $I(n)$ contains an interpretation satisfying $h \cdot |s|$ clauses.*

Lemmas 1 and 2 have been proved in such a way that they contain a proof of Lemma 4.

Note that we can enumerate the polynomial set $I(n)$ of "interesting" interpretations without knowing anything of s except the number of variables.

The following algorithm, called ENUMERATE, constructs an interpretation satisfying at least the fraction h of the clauses of a 2-satisfiable cnf s.

(1) For each variable L which occurs negated in a clause of length 1, replace all occurrences of the literals L and L' by their complements. (Afterward the clauses of length 1 only contain positive literals.)
(2) Compute the first prime p greater than or equal to n, and let W be a set of p variables containing all those from s.
(3) Enumerate the set $I(n)$ of interpretations of W, and choose an interpretation which satisfies the maximal number of clauses in s

Analysis of ENUMERATE. Algorithm ENUMERATE is a polynomial algorithm in the number l of occurrences of literals in the input cnf. By the Postulate of Bertrand [9, p. 22] the next prime of an integer n can be found in time $O(n^{3/2})$. ENUMERATE checks at most $8 \cdot n^3$ interpretations, where n is the number of variables of the input cnf. Thus the overall running time is $O(n^3 \cdot l)$.

ENUMERATE does not use the reduction which transforms a 2-satisfiable cnf to an element in RED1. If this reduction is used, the algorithm has to check at most $4 \cdot n^2$ interpretations, because the optimal k (number of variables set true) for a symmetrized cnf can be computed by using calculus. This would reduce the running time to $O(n^2 \cdot l)$ but might be expected to behave poorly in practice.

In [7] an algorithm B2 (for MAXIMUM SATISFIABILITY) has been introduced. Following [10], we also consider the variant RJ of B2.

The algorithms are based on the notions of weight of a cnf and the elimination of a literal in a cnf.

For a cnf s the weight $w(s)$ is defined to be $\sum_c 2^{-|c|}$, where the sum is taken over all clauses c of s. $|c|$ is the number of literals in c. It is convenient to allow clauses of length 0, which are always unsatisfied.

Let s be a cnf and L a literal; then $s[L]$ is the cnf obtained from s by the following process of elimination. A clause c of s containing L is dropped. A clause c of s containing L' is replaced by the clause c' obtained from c by dropping L' in c. If literal L is eliminated, then the variable V corresponding to L is set true if L is positive. Otherwise V is set false.

Algorithm B2 constructs an interpretation of a cnf s by iteratively eliminating the literals in s.

For a cnf $s0$ containing no variables, the weight $w(s0)$ is equal to the number of empty clauses in $s0$. Therefore, if s is a sequence of m clauses and $s0$ is the last cnf in the elimination process, the fraction of satisfied clauses is equal to $(m - w(s0))/m$.

Algorithm B2 chooses the literal L to be eliminated in such a way that $w(s[L]) \leq w(s)$. (This is possible since for all literals L in s the following equation holds: $(\frac{1}{2}) \cdot (w(s[L]) + w(s[L'])) = w(s)$). The fraction of satisfied clauses is therefore at least $(m - w(s))/m$.

Algorithm RJ chooses a literal $L0$ so that $w(s[L0]) \leq w(s[L])$ for all L in s.

We give an example of a 2-satisfiable cnf s_m for which RJ constructs (and B2 *could* construct) an interpretation such that only $(3 \cdot m + 1)/(5 \cdot m + 1) \sim 0.6$ clauses are satisfied. s_m is a sequence of $(5 \cdot m + 1)$ clauses $c_1, c_2, \ldots, c_{5m+1}$ in $2 \cdot m + 1$ variables $A, B_1, B_2, \ldots, B_{2 \cdot m}$:

$$c_i = \begin{cases} B_i & \text{for } i = 1, \ldots, 2 \cdot m; \\ A' \vee B'_{i-2 \cdot m} & \text{for } i = 2 \cdot m + 1, \ldots, 4 \cdot m; \\ A & \text{for } i = 4 \cdot m + 1, \ldots, 5 \cdot m + 1. \end{cases}$$

The weights are as follows:

$$\begin{aligned} w(s_m[A]) &= 2 \cdot m, \\ w(s_m[A']) &= 2 \cdot m + 1, \\ w(s_m[B_j]) &= 2 \cdot m + \tfrac{1}{4} \qquad (j = 1, \ldots, 2 \cdot m), \\ w(s_m[B'_j]) &= 2 \cdot m + \tfrac{3}{4} \qquad (j = 1, \ldots, 2 \cdot m). \end{aligned}$$

Therefore variable A is set true by RJ, and at most the fraction $(3 \cdot m + 1)/(5 \cdot m + 1) \sim 0.6$ of the clauses is satisfied (independent of the interpretations of the remaining variables).

4. *NP-Completeness*

In this section we prove

COROLLARY 3. *The set of 2-satisfiable cnfs which have an interpretation satisfying the fraction h' ($1 \geq h' > h$; h', e.g., rational) is NP-complete.*

PROOF. Let $h' = p/q$ be a rational number ($1 \geq h' > h$). We use the fact that the set of 2-satisfiable cnfs is NP-complete [2]. Therefore we give a polynomial transformation T which transforms a 2-satisfiable cnf s to a 2-satisfiable cnf $T(s)$ so that s is satisfiable iff $T(s)$ has an interpretation satisfying at least the fraction h' of the clauses.

Let s be a 2-satisfiable cnf containing m clauses. We may assume without loss of generality that $h'm < m - 1$. Since $h' > h$, there exist integers $m_1 > m_2$ and a 2-satisfiable cnf $t(m1, m2)$ containing $m1$ clauses of which only $m2$ are satisfiable and such that $m2/m1 < h'$ (Theorem 2). $T(s)$ contains $z_1 = m_1 p - m_2 q$ copies of s concatenated with $z_2 = m(q - p)$ copies of $t(m_1, m_2)$. In the following we describe how to compute z_1 and z_2 as a function of s and h'.

If in cnf s at most r ($r = m$ or $r = m - 1$) clauses can be satisfied, then at most the fraction

$$f(r, z_1, z_2) = \frac{r \cdot z_1 + m_2 \cdot z_2}{m \cdot z_1 + m_1 \cdot z_2}$$

of the clauses can be satisfied in cnf $T(s)$. The reduction T requires that

$$f(m - 1, z_1, z_2) < h' \qquad \text{and} \qquad f(m, z_1, z_2) \geq h'.$$

It is straightforward to check that both inequalities are satisfied if $z_1 = m_1 p - m_2 q$ and $z_2 = m(q - p)$, since $h'm < m - 1$. □

Remark. In [4] it is shown that the set of cnfs containing clauses of at most length 2 and which have an interpretation satisfying the fraction $\frac{7}{10}$ of the clauses is NP-complete. Using a reduction similar to that in [4] and the above method, it can be shown that the set of 2-satisfiable cnfs containing clauses of at most length 2 which have an interpretation satisfying the fraction h' ($1 > h' > h$; h', e.g., rational) of the clauses is NP-complete.

5. *Open Problems and Concluding Remarks*

So far we have discussed 2-satisfiable cnfs. Similar results hold for 1-satisfiable cnfs. A cnf is said to be 1-satisfiable if it does not contain an empty clause. It is obvious that a 1-satisfiable cnf has an interpretation which satisfies $|s|/2$ clauses. The constant 0.5 is optimal, and an interpretation satisfying half of the clauses can be found in polynomial time. For any rational t ($1 \geq t > 0.5$) the set of 1-satisfiable cnfs which have an interpretation satisfying $t \cdot |s|$ clauses is NP-complete.

These results suggest the following generalization: A cnf s is said to be *k-satisfiable* ($k = 1, 2, 3, \ldots$) iff any k of its clauses are satisfiable.

Conjecture C_k ($k = 3, 4, 5, \ldots$). There exists a constant r_k such that

(1) for any k-satisfiable cnf s there exists an interpretation of s satisfying at least $r_k \cdot |s|$ clauses;
(2) for any r greater than r_k there exists a k-satisfiable cnf s such that no interpretation satisfies $r \cdot |s|$ clauses;
(3) there is a polynomial algorithm which finds an interpretation guaranteed by (1);
(4) if $r_k < r$ and r is rational, the set of k-satisfiable cnfs having an interpretation which satisfies at least $r \cdot |s|$ clauses is NP-complete;
(5) r_k is algebraic.

We conjecture that $\lim_{k \to \infty} r_k = 1$.

Although the conjecture itself is only of theoretical interest, its proof might yield practical algorithms of an unknown type.

Symmetrization can be applied to a special class of systems of linear inequalities of the following type. Let s be a system of linear inequalities (sli) $c_1, c_2, \ldots, c_m$ in n variables $x_1, x_2, \ldots, x_n$, where inequality c_i is of the form

$$\sum_{j=1}^{n} a_{ij}x_j \geq b_i \qquad (a_{ij} \in \{1, 0, -1\},\ x_j \in \{0, 1\},\ b_i \text{ an integer},\ 1 \leq i \leq m,\ 1 \leq j \leq n).$$

The following question is of considerable practical interest: What is the complexity of the problem of constructing a (0, 1)-assignment J to the variables of an sli s so that J satisfies at least a given fraction of the inequalities. A partial answer can be found in [11, 12].

ACKNOWLEDGMENT. We wish to thank the referee for the numerous detailed comments, which have substantially improved our paper.

REFERENCES

(Note References [13, 14] are not cited in the text.)

1. CARMICHAEL, R.D. *Introduction to the Theory of Groups of Finite Order.* Dover, New York, 1937.
2. COOK, S.A The complexity of theorem-proving procedures Proc 3rd Ann. ACM Symp. on Theory of Computing, Shaker Heights, Ohio, 1971, pp 151-158.
3. ERDŌS, P, AND KLEITMAN, D J On coloring graphs to maximize the proportion of multicolored K-edges. *J Comb Theory 5*, 2 (1968), 164-169
4. GAREY, M.R., AND JOHNSON, D.S Some simplified NP-complete problems Proc. 6th Ann. ACM Symp on Theory of Computing, 1974, Seattle, Wash, pp. 47-63
5. GAREY, M R, AND JOHNSON, D.S. Approximation algorithms for combinatorial problems—an annotated bibliography. In *Algorithms and Complexity. Recent Results and New Directions*, J Traub, Ed., Academic Press, New York, 1976, 41-52
6. GAREY, M R, AND JOHNSON, D.S *Computers and Intractability.* Freeman, San Francisco, 1979
7. JOHNSON, D S. Approximation algorithms for combinatorial problems *J Comput Syst. Sci 9* (1974), 256-278.
8. KARP, R.M. Reducibility among combinatorial problems In *Complexity of Computer Computations*, R E. Miller and J W. Thatcher, Eds, Plenum Press, New York, 1972, pp 85-104.
9. LANDAU, E. *Handbuch der Lehre von der Verteilung der Primzahlen.* Chelsea Publishing, New York, 1974.
10. LIEBERHERR, K.J. Towards feasible solutions of NP-complete problems, Tech. Rep No. 14, Institute for Informatics, ETH Zentrum, CH-8092 Zurich, Switzerland, 1975
11. LIEBERHERR, K J. P-optimal heuristics *Theoret Comput. Sci. 10*, 2 (1980), 123-131.
12. LIEBERHERR, K J. Probabilistic combinatorial optimization Tech Rep. 281, Dep of Electrical Engineering and Computer Science, Princeton Univ, Princeton, N J, 1981
13. LIEBERHERR, K J, AND SPECKER, E Interpretations of 2-satisfiable conjunctive normal forms. *Not Am. Math Soc. 25*, 2 (1978), A-295.
14. LIEBERHERR, K.J, AND SPECKER, E Complexity of partial satisfaction. 20th Ann IEEE Symp on Foundations of Computer Science, San Juan, Puerto Rico, 1979, pp 132-139

RECEIVED AUGUST 1978, REVISED APRIL 1980; ACCEPTED APRIL 1980