1. Introduction

The (bounded) GENERALIZED MAXIMUM SATISFIABILITY (GMS) problem, similar to the one defined by [Schaefer (1978)], covers a broad range of NP-complete problems, e.g. it is a generalization of INDEPENDENT SET, LINEAR INEQUALITY, HITTING SET, SET PACKING, MINIMUM COVER, etc. We present a new class of polynomial approximation algorithms for GMS, having natural performance bounds which are better or equal to all previously published or unpublished algorithms. The new algorithms are efficient, can be automatically generated and have a wide range of applications because of the central position of the GMS problem in the class of combinatorial optimization problems.

This paper is motivated by the observation that a large class of means for GMS can be computed efficiently, although the straightforward computation is exponential. By a mean the average number of satisfied clauses for a class of assignments is understood. The method which is used to find a short closed form for certain means is called "symmetrization" and was used earlier e.g. in [Erdős/Kleitman (1968)] or [Lieberherr/Specker (1979)]. Symmetrization is related to the "averaging method" described in [Lovasz (1979) problems 13.12, 13.41]. The averaging method guarantees the existence of a combinatorial object and in this paper it is shown how to find it efficiently in the case of GMS.

The algorithms presented here can also be considered as a successful attempt to make certain probabilistic existence arguments of the type given in [Erdős/Spencer (1974)] constructively efficient. As a simple example consider problem 13.41 in [Lovasz (1979)]. If an $r$-uniform hypergraph has at most $2^{r-1}$ edges then it is 2-colorable. (Note that 2-colorability of hypergraphs is NP-complete which follows from a general result in [Schaefer (1978)]). If the points of the hypergraph are colored with red and blue at random, independently of each other and with probability 1/2, a standard argument shows that the probability that such a random coloration contains a monochromatic edge is $<1$. Hence there exists a 2-coloring and the techniques described here allow to find it efficiently.

The fact that certain means can be computed efficiently allows to compute good approximations for GMS, since maximizations among means are efficient. A mean $m$ is formulated in such a way that an assignment satisfying at least $m$ clauses always exists. For the means which are considered here, their recursive definition implies a fast greedy algorithm to find such an assignment.

The results in this paper extend and complement those in [Lieberherr (80)], but can be read independently. There it was shown how to compute a certain function maxmean. Here we consider an infinite family of means, of which maxmean is only the first element. Also, in this paper the bounded GMS is discussed, which contains several interesting graph theoretic problems as special cases.

The results presented in this paper have several interpretations, including the following:
- A class of existence proofs is made efficiently constructive.
- A class of randomized algorithms is made deterministic and efficient.
- A relaxation of GMS is shown to be solvable in polynomial time. (see [Lieberherr (1980)])
- A new class of NP-equivalent construction problems is given. Namely the performance bound of the given algorithms is shown to be absolute P-optimal in the sense that if there is a polynomial algorithm which satisfies at least one clause more than the performance bound guarantees (if it is possible), then $P=NP$.
- A new class of combinatorial approximation algorithms is introduced which is based on "background" optimization. Instead of maximizing among all assignments we maximize among expected values for parametrized random solutions. It turns out that this "background" optimization is in two precise senses best possible, if $P\neq NP$. 
The given algorithms have the potential (supported by practical experience) to speed up the traditional backtrack algorithms for GMS problems considerably.

Plan for the paper:
Section 2 introduces definitions and the notation used. Elementary relationships between several means are given in Section 3 to motivate the maxmean functions and the use of partitions. Section 4 shows how to compute the various means efficiently by using the symmetrization method. Section 5 contains algorithms to find assignments which satisfy at least as many clauses as the means predict.

Theorem 1 and 2 of Section 6 show that at least maxmean (S) clauses can be satisfied in polynomial time, but that it is hard to satisfy more. Theorem 1 is built on renamings and theorem 2 on permutations. The bounded maximum $\psi$-satisfiability problem BGMS bounds the number of ones which are allowed in an assignment. BGMS is discussed in Section 7 where the main ideas of the paper are applied to this bounded problem.

2. Definitions

**GENERALIZED MAXIMUM SATISFIABILITY**

We start with an introductory example. Let $R(x, y, z)$ be a 3-place logical relation whose truth table is \{(1,0,0), (0,1,0), (0,0,1)\}, i.e. $R(x, y, z)$ is true iff exactly one of its three arguments is true. Consider the problem of deciding whether an arbitrary conjunction of clauses of the form $R(x, y, z)$ is satisfiable. Following [Schaefer (1978)], this problem is called the **ONE-IN-THREE SATISFIABILITY** problem. For example, the formula $R(b, c, d) \land R(b, c, a) \land R(a, b, c)$ is satisfiable, because it is made true by assigning the values 0,0,1,0 to the variables $a, b, c, d$ respectively. The **ONE-IN-THREE SATISFIABILITY** problem is NP-complete [Schaefer (1978)].

The similarity between this problem and the standard satisfiability problem for propositional formulas in conjunctive normal form (CNF) leads to the generalization which is the subject of this paper. Consider the problem of deciding whether a given CNF with three literals in each clause is satisfiable - a well-known NP-complete problem. Since a clause may contain any number of negated variables from 0 to 3, there are four distinct relations. They are defined by $R_0(a, b, c) = a$ or $b$ or $c$, $R_1(a, b, c) = not a$ or $b$ or $c$, $R_2(a, b, c) = not a$ or $not b$ or $c$, $R_3(a, b, c) = not a$ or $not b$ or $not c$. An input to the standard satisfiability problem is just a conjunction of clauses of the form $R_i(a, b, c), i \in \{0, 1, 2, 3\}$.

This sets the stage for the following generalization. Let $\psi = \{R_1, \ldots, R_m\}$ be any finite set of logical relations. A logical relation is defined to be any subset of $\{0, 1\}$ for some integer $r \geq 1$. The integer $r$ is called the **rank** of the relation. Define a $\psi$-**formula** to be any sequence of clauses, each of the form $R_i(\xi_1, \xi_2, \ldots)$, where $\xi_1, \xi_2, \ldots$ are distinct, non-negated variables whose number matches the rank of $R_i$, $i \in \{1, \ldots, m\}$. The $\psi$-**satisfiability problem** is the problem of deciding whether a given $\psi$-formula is satisfiable. The main result in [Schaefer (1978)] characterizes the complexity of the $\psi$-satisfiability problem for every finite set $\mathcal{R}$ of logical relations. An interesting feature of this characterization is that for any such $\psi$, the $\psi$-satisfiability problem is either polynomial-time decidable or NP-complete. The difficulty of approximating the $\psi$-satisfiability problem is the subject of this paper. The **MAXIMUM $\psi$-SATISFIABILITY** problem is defined by

**Instance:** a $\psi$-formula $S$.

**Question:** Find a (0,1)-assignment to the variables of $S$ which satisfies the maximum number of clauses.
Means

Let \( \psi \) be a set of relations and \( S \) a \( \psi \)-formula with \( n \) variables. \( \text{mean}_{\text{ALL}, \rho} (S) \) denotes the expected number of satisfied clauses if each variable is assigned 0 or 1 at random, independently of each other and with probability \( \rho (0 \leq \rho \leq 1) \). \( \text{mean}_k (S) \) is the average number of satisfied clauses among all assignments which set exactly \( k \) variables to 1. Let \( \text{maxmean} (S) = \max_{0 \leq k \leq n} \text{mean}_k (S) \). Consider a partition of the \( n \) variables into the first \( n_1 \) variables of type 1 and the next \( n_2 \) variables of type 2 \( (n = n_1 + n_2) \).

\( \text{mean}_{k_1, k_2}^{n_1, n_2} (S) \) is the average number of satisfied clauses among all assignments, for which \( k_1 \) of the \( n_1 \) variables and \( k_2 \) of the \( n_2 \) variables are set to 1.

\[
\text{maxmean}^{n_1, n_2} (S) = \max_{0 \leq k_1 \leq n_1} \max_{0 \leq k_2 \leq n_2} \text{mean}_{k_1, k_2}^{n_1, n_2} (S).
\]

The definition of \( \text{mean}_{k_1, k_2, k_3}^{n_1, n_2, n_3} (S) \) and \( \text{maxmean}_{k_1, k_2, k_3}^{n_1, n_2, n_3} (S) \) is straightforward.

Renamings

The renaming of a variable \( x \) with respect to value \( v \) is a substitution of \( e(x, v) = (x - v) \mod 2 \) for variable \( x \).

Let \( J \) be an assignment for formula \( S \). The renaming of formula \( S \) with respect to \( J \) is a substitution of \( e(x, J(x)) \) for all variables \( x \) in \( S \). The resulting formula is called the renamed formula with respect to \( J \).

Let \( R (x_1, \ldots, x_n) \) be a relation and let \( J \) be an assignment for \( x_1, \ldots, x_n \). The renamed relation \( R \) with respect to \( J \) is the relation \( L(R, J) \) defined by

\[
L(R, J)(e(x_1, J(x_1)), \ldots, e(x_n, J(x_n))) \rightarrow R(x_1, \ldots, x_n).
\]

By definition

\[
R(J(x_1), \ldots, J(x_n)) = L(R, J)(0, \ldots, 0).
\]

A set of relations \( \psi \) is said to be closed under renaming, if all relations, which can be generated from relations in \( \psi \) by renaming, are in \( \psi \). A set of relations is closed under restriction, if all relations, which can be generated from relations in \( \psi \) by substituting constants, are in \( \psi \).

Symmetry

Let \( SA(S, J) \) be the number of satisfied clauses in formula \( S \) under assignment \( J \). Let \( \pi_n \) be the full permutation group on the \( n \) variables of \( S \). For \( \sigma \in \pi_n \), let \( \sigma(S) \) be the permuted formula, which is the result of substituting \( \sigma(v) \) for all variables \( v \) in \( S \). A \( \psi \)-formula \( S \) is called symmetric if any permutation of the variables in the formula returns the same formula up to a permutation of the clauses. If \( S \) is a symmetric \( \psi \)-formula, then for all permutations \( \sigma \in \pi_n \) and all assignments \( J \) of \( S \):

\[
SA(S, J) = SA(\sigma(S), J) \quad \text{or equivalently} \quad \sigma^{-1}(\{v \in S \mid \sigma(v) \in S \}) = \{v \in S \mid \sigma^{-1}(v) \in S \},
\]

where the assignment \( \sigma^{-1}(J) \) is defined by \( \sigma^{-1}(J)(v) = J(\sigma^{-1}(v)) \) for all variables \( v \) in \( S \).

A symmetric \( \psi \)-formula which contains a relation of rank \( r \) contains at least \( \binom{n}{r} \) clauses. The notion of symmetry is easily generalized, if the variables are partitioned into classes. Let \( \Phi \) be a partition of the \( n \) variables of \( S \) into classes. Then \( S \) is said to be \( \Phi \)-symmetric, if any permutation of the variables, which preserves the classes, returns the same formula up to a permutation of the clauses.

A logical relation \( R \) of rank \( r \) is said to be symmetric, if for any permutation \( \sigma \in \pi_n \) of \( r \) variables: \( R(\xi_1, \xi_2, \ldots, \xi_r) \iff R(\sigma(\xi_1), \sigma(\xi_2), \ldots, \sigma(\xi_r)) \).
Complexity

Let $\psi$ be a set of relations. $cl(S)$ denotes the number of clauses in a $\psi$-formula $S$. A rational number $C (0 \leq C \leq 1)$ is said to be (relative) P-optimal with respect to $\psi$, if
1. there is a polynomial algorithm $Q_1$ satisfying at least $C \cdot cl(S)$ clauses for any $S \in \{\psi\text{-formulas}\}$ and
2. the set of $\psi$-formulas having an assignment satisfying at least $(C+\varepsilon) \cdot cl(S)$ clauses is NP-complete for any rational $\varepsilon > 0$.

Algorithm $Q_1$ is also called relative P-optimal.

A polynomial-time computable function $q : \{\psi\text{-formulas}\} \to N$ is said to be absolute P-optimal with respect to $\psi$, if
1. there is a polynomial algorithm $Q_2$ satisfying at least $q(S)$ clauses for any $S \in \{\psi\text{-formulas}\}$
and
2. a) The set of $\psi$-formulas $S$, which have an assignment satisfying more than $q(S)$ clauses, is in $P$, iff $NP = coNP$.
   b) There is a polynomial algorithm, which satisfies more than $q(S)$ clauses (if this is possible), iff $P = NP$.

Algorithm $Q_2$ is also called absolute P-optimal.

Of course, there are many trivial absolute P-optimal functions, but we consider interesting performance bounds defined by closed form expressions.

A decision, search or optimization problem $Q$ is $NP$-equivalent, whenever it can be shown that a polynomial algorithm exists for $Q$, iff $P = NP$ (by Turing reductions). (A $NP$-equivalent decision problem is by definition $NP$-complete with respect to Turing reductions.)

Relations

$R_{i,h}^S(x_1, x_2, \ldots, x_n) \iff \sum_{j=1}^{n} a_j Sb$ where $S \in \{\geq, \leq, =\}$ and $a_j \in \{-1, 0, 1\}$ and $j$ coefficients $a_i (1 \leq i \leq n)$ are $\pm 1$ and the first $i$ coefficients are $+1$.

Assume that the variables of $\psi$-formula $S$ are partitioned into two classes. Then

$R_{i_1, i_2, j_1}^S(x_1, x_2, \ldots, x_n) \iff \sum_{i=1}^{n_1} a_i x_i + \sum_{i=2}^{n_2} a_j x_j \geq b$

where $n_1 + n_2 = n$ and the variables $\{x_{i_1} \mid 1 \leq i_1 \leq n_1\}$ are in class 1 and the variables $\{x_{i_2} \mid 1 \leq i_2 \leq n_2\}$ are in class 2. $j_1$ coefficients $a_j (1 \leq i_1 \leq n_1)$ are $\pm 1$ with the first $i_1$ of them = $+1$, and similar, $j_2$ coefficients $a_j (1 \leq i_2 \leq n_2)$ are $\pm 1$ with the first $i_2$ of them = $+1$.

Further notation

$SA(S, J)$ is the number of satisfied clauses in formula $S$ under assignment $J$.

$S_{\nu=x} \in \{0,1\}$ denotes the $\psi$-formula which is obtained from $S$ after substituting $\nu$ for $x$. Note that $S_{\nu=x}$ might have clauses containing relations (even of rank 0) which are always satisfied or never satisfied. It is assumed that $\psi$ is closed under restriction.

$J_{\text{ALL 0}}$ is the assignment which assigns 0 to all variables.
General assumptions

It is assumed throughout the paper that the finite sets of relations $\Psi$ are closed under restriction and renaming (although this is sometimes not necessary). Also, only sets of relations $\Psi$ are considered for which the corresponding maximum $\Psi$-satisfiability problem is NP-equivalent.

3. Relationships between means

The algorithms discussed in this paper compute assignments which are guaranteed to exist by averaging arguments. It is the purpose of this section to discuss several averaging arguments (expressed as means) and to show their relationships. In section 4 it will be shown how the means can be computed efficiently.

Let $\text{mean}_{4LL,p}(S)$ denote the expected number of satisfied clauses, if each variable is assigned 0 or 1 at random, independently of each other and with probability $p (0 \leq p \leq 1)$. It turns out that $\text{mean}_{4LL,p}(S)$ can be easily computed, especially if $p = 1/2$. Let $\text{mean}_k(S)$ be the average number of satisfied clauses among all interpretations which set exactly $k$ variables to 1, and let $\text{maxmean}^n(S) = \max_{0 \leq k \leq n} \text{mean}_k(S)$. Since

$$\text{mean}_{4LL,p}(S) = \sum_{k=0}^n p^k (1-p)^{n-k} \binom{n}{k} \text{mean}_k(S)$$

$\text{maxmean}^n(S) \geq \text{mean}_{4LL,p}(S)$. Equality holds, if $\text{mean}_k(S)$ is a constant in $k$. Therefore, informally, an algorithm which guarantees to satisfy $\text{maxmean}^n(S)$ clauses can be expected to be a better approximation algorithm than one which only satisfies $\text{mean}_{4LL,p}(S)$ clauses.

Since

$$\sum_{l_2=0}^{l_1} \binom{n_1}{l_2} \binom{n_2}{l_1-l_2} = \binom{n_1+n_2}{l_1}$$

(Vandermonde's identity), it is natural to consider $\text{mean}_{n_1}^{n_1,n_2}(S)$ which is the following average: Partition the number of variables into two classes of size $n_1, n_2 (n_1 + n_2 = n = \text{number of variables in } S)$ and compute the average number of satisfied clauses among all assignments for which $l_2$ of the $n_1$ variables and $l_1 - l_2$ of the $n_2$ variables are set to 1.

Let

$$\text{maxmean}_{l_1}^{n_1,n_2}(S) = \max_{0 \leq l_2 \leq l_1} \text{mean}_{l_1}^{n_1,n_2}(S).$$

Since

$$\text{mean}_{l_1}^{n_1}(S) = \frac{1}{n} \sum_{l_2=0}^{l_1} \binom{n_1}{l_2} \binom{n_2}{l_1-l_2} \text{mean}_{l_1}^{n_1,n_2}(S).$$

$\text{maxmean}_{l_1}^{n_1,n_2}(S) \geq \text{mean}_{l_1}^{n_1}(S)$. Equality holds, if $\text{mean}_{l_1}^{n_1,n_2}(S)$ does not depend on $l_2$.

The following lemma summarizes relationships of this type.

Lemma

(1) $\text{maxmean}_{l_1}^{n_1,n_2}(S) \geq \text{mean}_{l_1}^{n_1+n_2}(S)$. Equality holds, if $\text{mean}_{l_1}^{n_1,n_2}(S)$ is constant in $l_2$. 


(2) $\max \text{mean}^{u_{1}, u_{2}}(S) \geq \max \text{mean}^{u_{1}+u_{2}}(S)$. Equality holds, iff for some $k_1, k_2 (0 \leq k_1 \leq n_1, 0 \leq k_2 \leq n_2)$

$$\max \text{mean}^{u_{1}, u_{2}}(S) = \text{mean}_{k_1+k_2}^{u_{1}, u_{2}}(S) = \text{mean}_{k_1+k_2}^{u_{1}+u_{2}}(S) = \max \text{mean}(S).$$

(3) $\text{mean}_{ALL,1/2}(S) \leq \max \text{mean}^{u_{1}, u_{2}}(S)$. Equality holds iff $\text{mean}_{k_1+k_2}^{u_{1}, u_{2}}(S)$ is constant in $k_1, k_2$.

Proof
(1) given above
(2) immediate
(3) observe that

$$\text{mean}_{ALL,1/2}(S) = \frac{1}{2^n} \sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} \binom{n_1}{i_1} \binom{n_2}{i_2} \text{mean}_{i_1,i_2}(S).$$

4. Computation of means

The computation of the means is based on the following idea. Consider a $\psi$-formula $S$ with $n_1$ variables of type 1 and $n_2$ variables of type 2. Let $\sigma$ be a permutation of the variables of $S$ which permute only variables of the same type, and apply $\sigma$ to $S$ to get $\sigma(S)$. Then $S$ and $\sigma(S)$ have the same $\text{mean}_{k_1+k_2}^{u_{1}, u_{2}}$. Let $\pi$ be the set of all $n_1! \cdot n_2!$ possible permutations of the variables of $S$ which permute only variables of the same type. Then the concatenation $\Sigma$ of all formulas $\sigma(S)$ for $\sigma \in \pi$ is symmetric and $\text{mean}_{k_1+k_2}^{u_{1}, u_{2}}(\Sigma) = |\pi| \cdot \text{mean}_{k_1+k_2}^{u_{1}, u_{2}}(S)$. Although $\Sigma$ is exponential in size, it has a nice structure, which allows to express $\text{mean}_{k_1+k_2}^{u_{1}, u_{2}}(\Sigma)$ in closed form. By dividing by $|\pi|$ we get

$$\text{mean}_{k_1+k_2}^{u_{1}, u_{2}}(S) = \sum_{\text{all relations } R \in S} \text{SAT}_{k_1+k_2}^{u_{1}, u_{2}}(R)$$

where

$$\text{SAT}_{k_1+k_2}^{u_{1}, u_{2}}(R) = \sum_{s_j=0}^{r_j(R)} q_{s_j}(R) \frac{k_j}{s_j} \frac{n_i - k_j}{r_j(R) - s_j}$$

$$\prod_{j=1}^{2} \left[ \frac{n_i}{r_j(R)} \right]$$

where

$r_j(R)$ is the rank of relation $R$ in class $j$,
$q_{s_j}(R)$ is the number of rows in the partitioned truth table of $R$ which contain exactly $s_j$ ones in class $j$,
$n_i$ is the number of variables in class $j$. 

$k_j$ is the number of variables in class $j$ which are set to 1.

This method can be used to compute $\text{mean}_{k_1,k_2}^n(S)$. It is clear that the complexity is a polynomial of small degree in $|S|$, if only relations of bounded rank $r$ are allowed.

5. Construction of assignments

The means $\text{mean}_1^n(S), \text{mean}_2^n(S), \ldots$ are rational numbers and it is clear from the definition that there must be an assignment for $S$ satisfying at least $\left[\text{mean}_1^n(S)\right]$, $\left[\text{mean}_2^n(S)\right]$, $\ldots$, clauses. This can be considered as a probabilistic existence argument: If $\text{mean}_1^n(S), \text{mean}_2^n(S), \ldots$ is greater than a given integer $c$, then with probability $> 0$ (for a suitable set of random assignments) there is an assignment satisfying at least $c+1$ clauses. It is now shown how to find such an assignment efficiently. The approach is to find a recurrence relation for $\text{mean}_k^n(S)$ and to transform it into a greedy algorithm.

The recurrence relation for $\text{mean}_k^n(S)$ is

$$\text{mean}_k^n(S) = \frac{k}{n} \text{mean}_{k-1}^{n-1}(S_{x=1}) + \frac{n-k}{n} \text{mean}_{k}^{n-1}(S_{x=0}).$$

The relevant recurrence relations for $\text{mean}_{k_1,k_2}^n(S)$ are:

Let $x$ be a variable of type 1 and $y$ be a variable of type 2.

$$\text{mean}_{k_1,k_2}^{n_1,n_2}(S) = \frac{k_1}{n_1} \text{mean}_{k_1-1,k_2}^{n_1-1,n_2}(S_{x=1}) + \frac{n_1-k_1}{n_1} \text{mean}_{k_1,k_2}^{n_1-1,n_2}(S_{x=0}).$$

$$\text{mean}_{k_1,k_2}^{n_1,n_2}(S) = \frac{k_2}{n_2} \text{mean}_{k_1,k_2-1}^{n_1,n_2-1}(S_{y=1}) + \frac{n_2-k_2}{n_2} \text{mean}_{k_1,k_2}^{n_1,n_2-1}(S_{y=0}).$$

These recurrence relations are implied by the observation that

$$n! \text{mean}_k^n(S) = k \cdot (n-1)! \text{mean}_{k-1}^{n-1}(S_{x=1}) + (n-k) \cdot (n-2)! \text{mean}_{k}^{n-2}(S_{x=0}).$$

This implies the following greedy algorithms:

Algorithm $\text{MEAN}_1^n$:
Input: $\psi$-formula $S$, integer $k$ ($0 \leq k \leq n$)
Output: Assignment which satisfies at least $\text{mean}_k^n(S)$ clauses.

for all variables $x$ in $S$ do
if $\text{mean}_{k-1}^{n-1}(S_{x=1}) > \text{mean}_k^{n-1}(S_{x=0})$
then $x := 1; k := k-1; S := S_{x=1}$
else $x := 0; S := S_{x=0}$

($\text{mean}_1^n(S)$ is defined to be zero.)

Algorithm $\text{MEAN}_{k_1,k_2}^{n_1,n_2}$:
Input: $\psi$-formula $S$ with a partition of its $n_1+n_2$ variables into 2 types, $n_1$ of type 1 and $n_2$ of type 2. Integers $k_1,k_2$ ($0 \leq k_1 \leq n_1, 0 \leq k_2 \leq n_2$)
Output: Assignment which satisfies at least $\text{mean}_{k_1,k_2}^{n_1,n_2}(S)$ clauses.

for all variables $x$ in $S$ of type 1 do
if \( \text{mean}_{n-1,n}^{x-1} (S_{x-1}) > \text{mean}_{n}^{x-1} (S_{x-1}) \)
then \( x := 1; k_{1} := k_{1} - 1; S_{x-1} := S_{x-1} \)
else \( x := 0; S_{x-0} := S_{x-0} \)
for all variables \( y \) in \( S \) of type 2 do
if \( \text{mean}_{n}^{y-1} (S_{y-1}) > \text{mean}_{n}^{y-1} (S_{y-0}) \)
then \( y := 1; k_{2} := k_{2} - 1; S_{y-1} := S_{y-1} \)
else \( y := 0; S_{y-0} := S_{y-0} \)

(\( \text{mean}_{n-1,n}^{x-1} (S) \) and \( \text{mean}_{n-1,n}^{y-1} (S) \) are defined to be zero)

The correctness of these algorithms is implied by the above recurrence relations.

6. Maximal means and their complexity

Since \( \text{mean}_{n}^{n} (S) \), \( \text{mean}_{n}^{n} (S) \) can be computed efficiently and since an assignment satisfying at least that many clauses can be found fast, it is natural to try some background optimization and consider the functions

\[
\text{maxmean}_{n} (S) = \max_{0 \leq k \leq n} \text{mean}_{n}^{k} (S),
\]

\[
\text{maxmean}_{n}^{n} (S) = \max_{0 \leq k_{1} \leq n_{1}, 0 \leq k_{2} \leq n_{2}} \text{mean}_{n}^{k_{1},k_{2}} (S).
\]

The maxmean functions depend on renamings of the formula \( S \), and if \( S \) is satisfiable, then there is a renaming \( R \) of \( S \), so that

\[
\text{maxmean}_{n}^{n} (R(S)) = \max_{0 \leq k_{1} \leq n_{1}, 0 \leq k_{2} \leq n_{2}} \text{mean}_{n}^{k_{1},k_{2}} (R(S)) = \cdots = cl(S).
\]

Therefore the following question is interesting: How difficult is it to find a renaming \( R \) for a formula \( S \), so that

\[
\text{maxmean} (R(S)) > \text{maxmean} (S) ?
\]

(In the following we use the abbreviation \( \text{maxmean} (S) \) for \( \text{maxmean}_{n}^{n} (S) \) or \( \text{maxmean}_{n}^{n} (S) \) or ... .)

This question is answered by

Theorem 1

1. If \( \text{maxmean} (S) \) is not an integer, then there is a polynomial algorithm which finds a renaming \( R \), so that

\[
\text{maxmean} (R(S)) > \text{maxmean} (S),
\]

namely a renaming \( R \) can be found in polynomial time, so that

\[
\text{maxmean} (R(S)) = \lfloor \text{maxmean} (S) \rfloor.
\]

2. If \( \text{maxmean} (S) \) is an integer, then

a) iff there is a polynomial algorithm to decide whether there is a renaming \( R \), so that

\[
\text{maxmean} (R(S)) > \text{maxmean} (S), \text{ then } NP = coNP.
\]

b) iff there is a polynomial algorithm which finds a renaming \( R \), so that

\[
\text{maxmean} (R(S)) > \text{maxmean} (S) \text{ (if there is such a renaming), then } P = NP.
\]

Proof of second part:
The key to the proof of this theorem is this question: Given a formula \( S \), how difficult is it to find a renaming \( R \) of \( S \), so that \( \text{maxmean}_{n}^{n} (R(S)) = \text{maxmean}_{n}^{n} (R(S)) ? \) (The
generalization to more than 2 variable types is straight-forward.) This problem is solved efficiently by the following algorithm $MAXMEAN_n^{+\mu_2}$. 

$$
\text{loop}
$$

1. Find $k_1, k_2$ so that $maxmean_n^{+\mu_2}(S) = mean_{k_1,k_2}^n(S)$. 
2. Apply $MEAN_{k_1,k_2}^n$ to find an assignment $J$ satisfying at least $mean_{k_1,k_2}^n(S)$ clauses. 
3. If assignment $J$ is no improvement over the previous assignment then exit. 
4. Rename $S$ so that the assignment $J_{ALL}0$, which assigns 0 to all variables, corresponds to assignment $J$. 

$$
\text{end}
$$

To prove part 2b) of the theorem, assume there is a polynomial algorithm $\Omega_1$, which returns an assignment satisfying more than the integer $maxmean_n^{+\mu_2}(S)$ clauses. Then the following algorithm is a polynomial algorithm for the maximum $\psi$-satisfiability problem. Algorithm $RED_1$: 

$$
\text{loop}
$$

1. Apply $MAXMEAN^{+\mu_2}_n$ to $S$; it returns a formula $S$ so that $maxmean(S) = SA(S, J_{ALL}0)$ is an integer; 
2. Apply $\Omega_1$: The assignment $\Omega_1(S)$ satisfies more than $maxmean(S)$ clauses, otherwise exit; 
3. Rename $S$ so that the assignment $J_{ALL}0$, which assigns 0 to all variables, corresponds to assignment $\Omega_1(S)$. 

$$
\text{end}
$$

This loop is executed at most $cl(S)$ times, where $cl(S)$ is the number of clauses in $S$. 

To prove part 2a) of the theorem, assume there is a polynomial algorithm $\Omega_2$, which decides whether more than $maxmean^{+\mu_2}_n(S)$ clauses can be satisfied. Then the following nondeterministic algorithm is polynomial for the complement of the NP-complete problem: Given $C$ and a $\psi$-formula $S$, is there an assignment which satisfies at least $C$ clauses?

Algorithm $RED_2$: 

1. Guess optimal assignment $J_{opt}$ for formula $S$; rename $S$, so that $J_{ALL}0$ corresponds to $J_{opt}$; (now $maxmean(S) = SA(S, J_{ALL}0)$). 
2. Apply $\Omega$ to determine that $J_{ALL}0$ is optimal. 

This would imply $NP = coNP$. 

It is conjectured that for any "non-trivial" $\psi$ the set of $\psi$-formulas which have an assignment satisfying more than $maxmean(S)$ clauses is NP-complete. Richard Statman pointed out a proof idea which works for sets of relations $\psi$ containing the disjunctions of length 2 and 3.

Proof of first part: 

Assume that $maxmean^{+\mu_2}_n(S)$ is not an integer (the generalization to more than 2 classes is straight-forward). Now algorithm $MAXMEAN^{+\mu_2}_n$ applied to $S$ will compute a renaming $R$, so that $maxmean^{+\mu_2}_n(R(S))$ is integer and $> maxmean^{+\mu_2}_n(S)$. 

A similar theorem can be proven for permutations/partitions instead of renamings. It will be employed to deal with graph theoretic optimization problems. The theorem does not hold for $maxmean_n(S)$, since $mean_n(S)$ does not depend on the order of the variables in
Theorem 2
1. Let $S$ be a $\psi$-formula, for which $\text{maxmean}_{1,n_2}(S)$ is not an integer.
   There is a polynomial algorithm which finds a permutation $\sigma$ of the $n$ variables of $S$ and numbers $\overline{n}_1, \overline{n}_2, \ldots, (\overline{n}_1 + \overline{n}_2 + \cdots = n)$, such that $\text{maxmean}_{1,n_2}(\sigma(S)) > \text{maxmean}_{1,n_2}(S)$.

2. Let $\text{maxmean}_{1,n_2}(S)$ be an integer.
   a) If there is a polynomial algorithm to decide, whether there is a permutation $\sigma$ of the variables of $S$ and numbers $\overline{n}_1, \overline{n}_2, \ldots, (\overline{n}_1 + \overline{n}_2 + \cdots = n)$, so that $\text{maxmean}_{1,n_2}(\sigma(S)) > \text{maxmean}_{1,n_2}(S)$, then $\text{NP} = \text{coNP}$.
   b) If there is a polynomial algorithm which finds a permutation $\sigma$ and numbers $\overline{n}_1, \overline{n}_2, \ldots, (\overline{n}_1 + \overline{n}_2 + \cdots = n)$, so that $\text{maxmean}_{1,n_2}(\sigma(S)) > \text{maxmean}_{1,n_2}(S)$, then $\text{P} = \text{NP}$.

7. Bounded maximum $\psi$-satisfiability

Many graph-theoretic optimization problems can be formulated as special cases of the bounded maximum $\psi$-satisfiability problem, including APPROXIMATE INDEPENDENT SET, GRAPH SEPARATION, APPROXIMATE SET PACKING, APPROXIMATE HITTING SET, APPROXIMATE MINIMUM COVER, APPROXIMATE VERTEX COVER, APPROXIMATE DOMINATING SET etc. (for definitions see [Garey/Johnson (1979)]). For this class of optimization problems efficient approximation algorithms with an absolute $P$-optimal performance bound are given.

Let $\psi$ be a finite set of logical relations.

**BOUNDED MAXIMUM $\psi$-SATISFIABILITY**

**Instance:** A $\psi$-formula $S$ with $n$ variables, two constants $L, U, 0 \leq L \leq U \leq n$.

**Question:** Find an assignment $J$ for $S$, so that the number $\text{ONE}(J)$ of ones in $J$ satisfies $L \leq \text{ONE}(J) \leq U$ and so that $J$ satisfies the maximum number of clauses in $S$.

In the following we only consider sets of relations $\psi$ for which this problem is $\text{NP}$-equivalent.

**Notation:** $(L, U) - \text{maxmean}_{1,n_2}(S)$ denotes

$$\max_{0 \leq k_1 \leq n_1, \ldots, 0 \leq k_2 \leq n_2} \text{mean}_{k_1,k_2}(S)$$

such that $L \leq \sum k_i \leq U$.

The following theorem has a similar flavor as theorems 1 and 2.

**Theorem 3**

Let $S$ be a $\psi$-formula.

1. Let $(L, U) - \text{maxmean}_{1,n_2}(S)$ be a rational number which is not integer. There is a polynomial algorithm which finds a permutation $\sigma$ of the $n$ variables of $S$ and numbers $\overline{n}_1, \overline{n}_2, \ldots, (\overline{n}_1 + \overline{n}_2 + \cdots = n)$, such that $(L, U) - \text{maxmean}_{1,n_2}(\sigma(S)) > (L, U) - \text{maxmean}_{1,n_2}(S)$ and $(L, U) - \text{maxmean}_{1,n_2}(\sigma(S))$ is integer.
2. Let \((L, U) - \maxmean n_1^n_2 \cdots (S)\) be an integer.
   a) If there is a polynomial algorithm to decide whether there is a permutation \(\sigma\)
of the variables of \(S\) and numbers \(\bar{n}_1, \bar{n}_2, \ldots, (\bar{n}_1 + \bar{n}_2 + \cdots = n)\), such that
   \((L, U) - \maxmean n_1^n_2 \cdots (\sigma(S)) > (L, U) - \maxmean n_1^n_2 \cdots (S)\)
   for \(0 \leq L, U \leq n\), then \(\text{NP} = \text{coNP}\).
   b) If there is a polynomial algorithm which finds a permutation \(\sigma\) and numbers
   \(\bar{n}_1, \bar{n}_2, \ldots, (\bar{n}_1 + \bar{n}_2 + \cdots = n)\), such that
   \((L, U) - \maxmean n_1^n_2 \cdots (\sigma(S)) > \maxmean n_1^n_2 \cdots (S)\)
   for \(0 \leq L, U \leq n\), then \(\text{P} = \text{NP}\).

   The proof is similar to the proofs of theorems 1 and 2. The following algorithm is relevant.

   **Input:** Instance of the bounded maximum \(\psi\)-satisfiability problem. Let \(S\) be the input
   formula with a partition of the variables into two classes of size \(n_1, n_2\). (The
   generalization to more than two classes is straightforward.) Assume
   \((L, U) - \maxmean n_1^n_2 \cdots (S)\) is not an integer.

   **Output:** A permutation \(\sigma\) of the variables and numbers \(\bar{n}_1, \bar{n}_2, \ldots, (\bar{n}_1 + \bar{n}_2 + \cdots = n)\), such that
   \((L, U) - \maxmean n_1^n_2 \cdots (\sigma(S)) > (L, U) - \maxmean n_1^n_2 \cdots (S)\)
   and
   \((L, U) - \maxmean n_1^n_2 \cdots (S)\) is an integer.

   **loop**
   1. Compute \(k_1, k_2\) such that
      \((L, U) - \maxmean n_1^n_2 \cdots (S) = \mean n_1^n_2 \cdots (S)\) and such that
      \(L \leq k_1 + k_2 \leq U\).
   2. Apply \(\text{MEAN}_{k_1}^{n_1} \cdots (S)\) to find an assignment \(J\) satisfying at least \(\mean n_1^n_2 \cdots (S)\) clauses:
   3. If assignment \(J\) is no improvement over the previous assignment, then exit.
   4. \(n_1 := k_1 + k_2; \quad n_2 := n - k_1 - k_2; \quad \text{permute the variables of } S, \text{ such that the first } n_1 \text{ variables are set 1 by } J.\)

   **end**

   Upon completion of this loop the relation
   \((L, U) - \maxmean n_1^n_2 \cdots (S) = \mean n_1^n_2 \cdots (S)\)
   holds.

   **7.1. Bounded maximum \(\psi\)-satisfiability of order 2**

   Let \(\psi\) be a finite set of relations and let \(r_{\max}\) be the maximal rank of the relations in
   \(\psi\). Then the corresponding bounded maximum \(\psi\)-satisfiability problem (BGMS) is said to
   be of order \(r_{\max}\).

   The BGMS of order two contains many graph-theoretical \(\text{NP}\)-equivalent optimization
   problems, including the following [Garey/Johnson(1979)]:

   **APPROXIMATE INDEPENDENT SET**

   \[\psi = \{R_{\delta - 1}\}, \quad U = n\]

   **GRAPH SEPARATION (also called: MINIMUM CUT INTO BOUNDED SETS)**

   \[\psi = \{R_{120}\}\]

   **APPROXIMATE SET PACKING**

   \[\psi = \{R_{\geq 1}\}, \quad U = n\]

   **APPROXIMATE HITTING SET**

   \[\psi = \{R_{\geq 1}\}, \quad L = 0\]
For these problems $\text{mean}_{k_1, k_2, \ldots, k_t}^n(S)$ is a homogeneous polynomial of degree two in $t$ variables:

$$\text{mean}_{k_1, k_2, \ldots, k_t}^n(S) = \sum_{i=1}^{t} a_i k_i^2 + \sum_{i<j}^{t} b_{i,j} k_i k_j$$

for certain coefficients $a_i, b_{i,j} (1 \leq i, j \leq t)$. To compute $(L, U) - \text{maxmean}_{k_1, k_2, \ldots, k_t}^n(S)$ is equivalent to finding the maximum of $\text{mean}_{k_1, k_2, \ldots, k_t}^n(S)$ at the integer lattice points in the polyhedron $[0 \leq k_i \leq n, (1 \leq i \leq t), L \leq \sum_{i=1}^{t} k_i \leq U]$. $\text{maxmean}_{k_1, k_2, \ldots, k_t}^n(S)$ can be computed with elementary methods without searching all integer lattice points in the above polyhedron.

8. Summary and Outlook

A new class of combinatorial approximation algorithms is given, which is based on "background" optimization. Instead of maximizing among all assignments (which is too time-consuming), we maximize among expected values for parametrized random solutions. The algorithms MAXMEAN*, MAXMEAN"1", ..., given in section 6 are P-optimal, efficient and can be generated automatically.

In another paper we show that algorithm MAXMEAN* also occurs naturally in the following extremal problem:

Let $\psi$ be a finite set of logical relations and let $\Gamma$ be a set of $\psi$-formulas which is closed under symmetrization. Assume that the $\psi$-satisfiability problem is NP-complete. Consider the fundamental constant

$$\tau_1 = \inf_{S \in \Gamma} \max_{\text{all assignments } J} \frac{SA(S, J)}{cl(S)}$$

Then

a) MAXMEAN* satisfies in polynomial time at least the fraction $\tau_1$ of the clauses in a given $\psi$-formula $S \in \Gamma$.

b) The set of $\psi$-formulas which have an assignment satisfying at least the fraction $\tau_1 + \epsilon$ of the clauses, is NP-complete for any rational $\epsilon > 0$.

Acknowledgement:

I would like to thank my wife Dr. R. Lieberher for stylistic improvements of the paper and my secretary S. Mairs for preparing it on the computer.
References


