Calculating Exact Ground States of Spin Glasses: A Polyhedral Approach

M. Grötschel, M. Jünger, G. Reinelt

Institut für Mathematik, Universität Augsburg
Memminger Str. 6, D-8900 Augsburg, FRG

Abstract

In this paper we describe recent developments in the theory and algorithm design of combinatorial optimization that are related to questions concerning ground states of spin glasses. In particular, we outline an approach, based on polyhedral combinatorics, that has led to the implementation of a cutting plane method for calculating exact ground states of spin glasses in the Ising model. With this method exact ground states for planar grids of size up to 40×40 with periodic boundary conditions and exterior magnetic field can be determined in reasonable running times.

1. The Max-Cut Problem

We now introduce a few notions of graph theory needed in the sequel and describe the max-cut problem to which the ground state problem of spin glasses can be reduced.

For our purposes, a graph $G = (V, E)$ consists of a finite nonempty set V of **nodes** and a set E of **edges** which are unordered pairs of different nodes, the **endnodes** of the edges. We denote an edge e with endnodes i and j simply by ij and say that i and j are joined by $e = ij$. This notation leads to no confusion since our graphs do not have parallel edges (several edges with the same pair of endnodes). The number of nodes of a graph G is called the **order** of G.

A **walk** in a graph G between two nodes u, v (short: (u, v)-**walk**) is a sequence of edges $i_1i_2, i_2i_3, \ldots, i_{k-2}i_{k-1}, i_{k-1}i_k$ such that $u = i_1$ and $v = i_k$. If $u = v$ the walk is called **closed**, otherwise it is called **open**. An open (i_1, i_k)-walk such that $i_p \neq i_q$ $(1 \leq p < q \leq k)$ is called a (i_1, i_k)-**path**, a closed (i_1, i_k)-walk such that $i_p \neq i_q$ $(1 \leq p < q < k)$ is called a **cycle**. So a cycle is a (i_1, i_{k-1})-path together with the edge i_1i_{k-1}.

Two nodes that are joined by an edge are called **adjacent** or **neighbors**. The set of edges having a node $v \in V$ as one of their endnodes is denoted by $\delta(v)$. The number $|\delta(v)|$ is the **degree** of node $v \in V$. More generally, if $W \subseteq V$, then $\delta(W)$ denotes the set of edges with one endnode in W and the other endnode in $V \setminus W$, i. e.,

$$\delta(W) = \delta(V \setminus W) = \{ij \in E \mid i \in W, j \in V \setminus W\}.$$

Any edge set of the form $\delta(W)$ is called a **cut**.

For a graph $G = (V, E)$ with weights $c_e \in \mathbb{R}$ for all $e \in E$, the weight $c(F)$ of an edge set $F \subseteq E$ is the sum of the weights of the elements of F, i. e.,

$$c(F) = \sum_{e \in F} c_e.$$

The **max-cut problem** is the following task. Given a graph $G = (V, E)$ with weights $c_e \in \mathbb{R}$ for all $e \in E$, find a cut $\delta(W)$, $W \subseteq V$, such that $c(\delta(W))$ is as large as possible.

Cut problems of various types are among the most important problems in combinatorial optimization; the max-cut problem, for instance, was one of the first to be shown $\mathcal{NP}$-complete — see KARP (1972). By considering weights $-c_e$ for all $e \in E$, of course also a minimum weight cut can be found via solving a max-cut problem. Let us point out here a subtlety in terminology. Whenever a min-cut problem is considered in combinatorial optimization (e. g., in the famous max-flow min-cut theorem), it is assumed that all weights c_e are nonnegative, that G is connected, and that a nonempty cut of minimum weight is to be found. In contrast to the max-cut problem, this problem can be solved in polynomial time (by any max-flow algorithm).

2. The Spin Glass Model

The mathematical approach we are going to outline can handle any spin glass model that is described by a Hamiltonian of the form

$$H = -\sum_{ij} J_{ij} S_i S_j,$$

where $S_i = \pm 1$ $(i = 0, 1, \ldots, n)$ is an Ising variable. Ising spins are fundamental for our Ansatz, but we can handle any type of interaction function (e. g., the Gaussian or the $\pm J$ model). There is no restriction on the dimension of the spin system, long range or short range models can be treated, exterior magnetic forces are allowed, we may have a random field. We are not restricted to special topological interaction structures like 2-dimensional or 3-dimensional grids.

The approach works in general, but — of course — knowing specific features of the model, definitely helps designing faster algorithms. We come back to this later.

Let us now describe the reduction of the ground state problem, i. e., finding a spin configuration of minimum energy, to a max-cut problem. We assume that we have a spin glass with n magnetic impurities $1, 2, \ldots, n$ and an exterior magnetic field 0. We set $V = \{0, 1, \ldots, n\}$ and consider V as the node set of a graph $G = (V, E)$, the **interaction graph** associated with the system. For a pair i, j of nodes, G contains an edge ij if the interaction J_{ij} between the magnetic impurities (resp. the field) is nonzero. With each impurity and with the exterior field i an Ising variable $S_i \in \{\pm 1\}$ is associated. For symmetry reasons the variable S_0 corresponding to the field may be fixed, to $S_0 = +1$, say. Given a spin configuration ω, i. e., an assignment of values $+1$ or -1 to the variables, the value of the Hamiltonian is

$$H(\omega) = -\sum_{ij \in E} J_{ij} S_i S_j.$$

In the case of a uniform field with strength h, this specializes to

$$H(\omega) = -\sum_{\substack{ij \in E \\ i,j>0}} J_{ij} S_i S_j - h \sum_{i=1}^{n} S_i.$$

Observe that each spin configuration ω induces a partition of the node set V of the interaction graph G into node sets V^+ and V^-, where $V^+ = \{i \in V \mid S_i = +1\}$ and $V^- = \{i \in V \mid S_i = -1\}$. So the energy of the spin configuration ω can be written in the form:

$$\begin{aligned} H(\omega) &= -\sum_{ij \in E(V^+)} J_{ij} S_i S_j - \sum_{ij \in E(V^-)} J_{ij} S_i S_j - \sum_{ij \in \delta(V^+)} J_{ij} S_i S_j \\ &= -\sum_{ij \in E(V^+)} J_{ij} - \sum_{ij \in E(V^-)} J_{ij} + \sum_{ij \in \delta(V^+)} J_{ij}, \end{aligned}$$

where, for $W \subseteq V$, $E(W) := \{ij \in E \mid i, j \in W\}$. Setting $C := \sum_{ij \in E} J_{ij}$ we see that

$$H(\omega) + C = 2 \sum_{ij \in \delta(V^+)} J_{ij},$$

and defining $c_{ij} := -J_{ij}$ for all $ij \in E$, the problem of minimizing H is, therefore, equivalent to maximizing

$$c(\delta(V^+)) := \sum_{ij \in \delta(V^+)} c_{ij}$$

over all $V^+ \subseteq V$. This problem is a weighted max-cut problem in the interaction graph G associated with the spin glass system. Thus, finding a ground state in the Ising model of a spin glass is equivalent to finding an optimum solution of the corresponding max-cut problem.

3. Complexity

A problem is said to be solvable in **polynomial time** if there is an algorithm that solves any instance of the problem in a running time that is bounded by a polynomial in the encoding length of the instance. Such an algorithm is often called a **good algorithm**. The class of problems solvable in polynomial time is denoted by $\mathcal{P}$. In the case of the max-cut problem, an instance is defined by specifying a graph $G = (V, E)$ and the weights $c_e \in E$. (The encoding of G needs at least $|V| + |E|$ bits. If the weights c_e are integers, for instance, the encoding length of the weights is $\sum_{e \in E}(\lceil \log_2(|c_e| + 1) \rceil + 1)$, binary encoding assumed.) There is no polynomial time algorithm known to solve the max-cut problem for general graphs. In fact, the max-cut problem is $\mathcal{NP}$-hard — see GAREY & JOHNSON (1979) for a formal treatment of the theory behind this.

Informally, $\mathcal{NP}$-hardness can be defined as follows. We say that a problem belongs to the class $\mathcal{NP}$, if one can verify in polynomial time that a proposed solution is indeed a solution. For the max-cut case, this means the following. For a graph G with

edge weights and an additional bound B we must be able to check whether a given edge set F is a cut $\delta(W)$ with weight $c(F)$ at least B. This is trivial of course, and in this sense, the max-cut problem belongs to $\mathcal{NP}$. Note that nothing is required about how to find a cut $\delta(W)$ with weight at least B. (We imagine that we have a "magic guessing module" that provides us with "good" cuts; our only task is to check in polynomial time that the guesses are really good.) $\mathcal{NP}$ is short hand for "nondeterministic polynomial time", and this name stems from a formal version of the "guessing" interpretation given above. We say that a problem Π is $\mathcal{NP}$-**hard** (or $\mathcal{NP}$-**complete**) if it has the following property: The existence of a polynomial time algorithm for Π implies the existence of a polynomial time algorithm for all problems in $\mathcal{NP}$. There are indeed such problems, and the max-cut problem is among them. The importance of $\mathcal{NP}$-hard problems is connected with one of the major open problems in algorithmic mathematics and computer science, namely the question whether $\mathcal{P} = \mathcal{NP}$ or not. Trivially, $\mathcal{P} \subseteq \mathcal{NP}$, but so far nobody was able to find a problem in $\mathcal{NP}$ that is not in $\mathcal{P}$. Now, the existence of a polynomial time algorithm for any $\mathcal{NP}$-hard problem would imply $\mathcal{P} = \mathcal{NP}$. Specifically, showing that there is a polynomial time or that there is no polynomial time algorithm for the max-cut problem would settle the $\mathcal{P} \stackrel{?}{=} \mathcal{NP}$ problem. This justifies saying that the max-cut problem is among the hardest problems in combinatorial optimization.

This observation implies that one cannot expect to be able to design an algorithm that works equally well for all possible instances of an $\mathcal{NP}$-hard problem. Restriction to particularly structured subproblems and design of special purpose algorithms for these is necessary. For the max-cut case, this means, that one should restrict the attention to special classes of graphs (for instance those that arise in various models of spin glasses) and design algorithms that utilize the structure of these graphs. Unfortunately, however, the max-cut problem is also $\mathcal{NP}$-hard for many important special types of graphs.

We will survey now what is known in this respect and how these results relate to the ground state problem of spin glasses.

A graph is **planar** if it can be drawn in the plane (a node is represented by a point, an edge ij by a line linking the points representing i and j) such that no two lines (representing edges) intersect, except possibly in their endpoints. ORLOVA & DORFMAN (1972) and HADLOCK (1975) have found a reduction of the max-cut problem in planar graphs to a so-called T-join problem by employing planar duality. The T-join problem can be solved in polynomial time for general graphs by an algorithm of EDMONDS & JOHNSON (1973) that ingeniously combines shortest path methods and matching techniques. A primal version of this algorithm was described in BARAHONA, MAYNARD, RAMMAL & UHRY (1982). It aimed — successfully — at solving 2-dimensional spin glass problems with nearest neighbor interactions. This algorithm is particularly useful to perform postoptimality analysis, e. g., to study existence of long range order. Planar spin glass problems on grids of size up to 50×50 have been handled with this method — see ANGLES D'AURIAC & MAYNARD (1984).

A graph G is said to be **contractible** to another graph H if H can be obtained from G by repeated deletion of nodes and edges and contraction of edges (an edge

is contracted by deleting it and identifying its endnodes). A graph $G = (V, E)$ is **bipartite** if its node set V can be partitioned into two nonempty subsets V_1, V_2 such that each edge $e \in E$ has one endnode in V_1 and the other in V_2. G is **complete bipartite** if each node of V_1 is joined to each node of V_2. The complete bipartite graph with $|V_1| = m$, $|V_2| = n$ is denoted by $K_{m,n}$. A graph is **complete** if every pair of nodes is joined by an edge. The complete graph of order n is denoted by K_n. The **k-sum** of two graphs G_1, G_2 is a glueing operation in which a complete subgraph of order k in G_1 is identified with a complete subgraph of order k in G_2, and afterwards, all edges of this complete subgraph are removed.

BARAHONA (1983) proved that if the max-cut problem can be solved in polynomial time for graphs G_1 and G_2 then it can also be solved in polynomial time for any 2-sum and any 3-sum of G_1 and G_2 (see GRÖTSCHEL & TRUEMPER (1986) for generalizations of this result). WAGNER (1937) proved that the class of graphs not contractible to K_5 (resp. to $K_{3,3}$) can be obtained by taking k-sums, $k = 2$ or 3, of planar graphs and a finite number of small special graphs. Combining the results of Barahona and Wagner yields that the max-cut problem is solvable in polynomial time for the class of graphs not contractible to K_5 and for the class of graphs not contractible to $K_{3,3}$. Both classes contain planar graphs since, by the Kuratowski-Wagner theorem, planar graphs are neither contractible to K_5 nor to $K_{3,3}$.

There are a few further classes of graphs resp. objective functions for which the max-cut problem is solvable in polynomial time. If all weights are nonpositive the empty cut is a maximum weight cut. But, as mentioned before, in this case one can even find a maximum weight nonempty cut using max-flow algorithms. If all edge weights are nonnegative, the max-cut problem can be solved for so-called weakly bipartite graphs — see GRÖTSCHEL & PULLEYBLANK (1981). This class of graphs contains the graphs not contractible to K_5 and those not contractible to $K_{3,3}$ — see FONLUPT, MAHJOUB & UHRY (1984). Further, there are polynomial time algorithms (that are probably not of too much practical value) for the max-cut problem in graphs of bounded tree-width (DRESS (1986)), graphs without long odd cycles (GRÖTSCHEL & NEMHAUSER (1984)), and graphs of bounded genus and the additional restriction that all edge weights satisfy $c_e \in \{0, +1, -1\}$ (BARAHONA (1981)).

The results described above imply the following. The ground state problem can be solved in polynomial time for all interaction graphs that are planar, so in particular for the standard 2-dimensional grid model. It can also be solved in polynomial time for all toroidal interaction graphs, provided interactions have value $\pm J$ only; a special case is the typical 2-dimensional grid model with periodic boundary conditions and $\pm J$ interactions. Moreover, it follows from results of PICARD & RATLIFF (1975) —see BARAHONA (1985) — that ground states of a random-field Ising ferromagnet can be computed in polynomial time. Here all nonzero interactions between magnetic spins are positive, while interactions with the exterior field may be positive or negative.

Let us now turn to the negative results. The above mentioned solvable cases seem to cover only few small classes of graphs. But there are a number of results showing that these classes cannot be enlarged significantly. The max-cut problem was shown to be $\mathcal{NP}$-complete for

- general graphs (KARP (1972)),
- cubic graphs (YANNAKAKIS (1978)); these are graphs where each node has degree exactly three,
- graphs not contractible to K_6 (BARAHONA (1983)),
- almost planar cubic graphs (BARAHONA (1983)); these are graphs that contain a node whose removal results in a planar cubic graph,
- 3-dimensional grid graphs (BARAHONA (1982)),
- two layer grid graphs with weights $0, \pm 1$ (BARAHONA (1982)),
- planar grid graphs with weights $0, \pm 1$ and a universal node (BARAHONA (1982)).

These results imply that the problem of determining ground states of spin glasses is $\mathcal{NP}$-complete, e. g., for the following interaction graphs

- 3-dimensional grids,
- 3-dimensional grids with two layers in the $\pm J$ model,
- planar grids with $\pm J$ interactions and with exterior magnetic field of strength $h = J$.

So most of the interesting spin glass configurations lead to $\mathcal{NP}$-hard models. Let us mention that one case is open. Is there a polynomial time algorithm for the max-cut problem in toroidal graphs (more specifically for toroidal grid graphs) with arbitrary weights? As mentioned before, this can be solved for the $\pm J$ model, but for arbitrary weights on a planar grid with periodic boundary conditions, no good algorithm is known.

4. Exact Methods

Let us recall that we focus on exact methods, i. e., algorithms that are designed with the intention to find maximum weight cuts and that end up with a proof of optimality. Of course, heuristics may also produce optimum solutions but one can never be sure of this. So, whenever we say that a problem is "solved" we mean that not just a feasible, but a true optimum solution is found and its optimality is proved. This makes a substantial difference, often overlooked in the physics literature on this subject.

As outlined in Section 3 there are good algorithms for a few special cases of the max-cut resp. ground state problem. Some of them, in particular those that are enumeration methods designed to handle particular types of graphs efficiently, can be applied to any graph to obtain maximum weight cuts. But outside their special range they exhibit exponential running time — not only in the worst case, it shows up always. Examples of this kind are the algorithms in GRÖTSCHEL & NEMHAUSER (1984) and DRESS (1986). These remarks also apply to the transfer matrix method described in MORGENSTERN & BINDER (1980) and MORGENSTERN (1983). The latter method is the only one of these for which computational experience has been reported. It can handle planar grids with exterior magnetic field of size up to 18×18. But even much faster (or parallel) computers with more memory than available today cannot push the "solvable grid size" much larger, say double it, due to the exponential explosion of time and space requirements.

A branch & bound method for general Ising models was proposed by HARTWIG, DASKE & KOBE (1984). It was applied to spin glass systems of up to 80 spins. An algorithm — in the spirit of the method we are going to describe — for 3-dimensional grids was described in BARAHONA & MACCIONI (1982). It can handle $5 \times 5 \times 5$ grids.

Assuming (as we do) that $\mathcal{P} \neq \mathcal{NP}$, there will never be an algorithm for the max-cut problem that runs fast on every type of graphs. It is vital for the empirical success of an algorithm for this problem to take special structures of the graphs considered into account and exploit them by studying their particular properties and by developing data structures that can handle these graphs quickly. Moreover, we believe that enumeration techniques should be avoided as much as possible. The reason is that enumerative methods usually have running times that are exponential for all problem instances and not only in the worst-case. We aim at methods that have stopping criteria, that is, if an optimum solution is found at an early stage, a proof of its optimality can be given. Such methods show much better performance empirically on the average. Of course, exponential running times will show up in some (hopefully only few) cases, otherwise we would have shown $\mathcal{P} = \mathcal{NP}$ — a unlikely event. To achieve such goals more (and frequently new) theory has to be developed, in particular special techniques that yield proofs of optimality.

To put this last remark in a general perspective, it is not only necessary to produce good lower bounds for the value of a max-cut problem (with heuristic methods, say), it is important to design algorithms that provide sharp upper bounds for this value. In fact, the method we will describe only produces upper bounds together with an optimum solution of a certain relaxed problem. If the optimum solution of the relaxed problem is a cut, an optimum solution of the max-cut problem is found, otherwise the relaxation is strengthened and we repeat.

5. Polyhedral Combinatorics

We will now describe the theoretical background of our method. It is based on ideas of polyhedral combinatorics. This is a subfield of combinatorial optimization which aims at describing combinatorial optimization problems as linear programs and solving these with special purpose methods. We outline the approach for the max-cut problem for general graphs.

Recall first that a **polytope** in $\mathbb{R}^n$ is the convex hull of finitely many points, or equivalently, a polytope is a bounded subset of $\mathbb{R}^n$ that is the intersection of finitely many halfspaces. Those points of a polytope P which are not representable as a convex combination of other points in P are the **vertices** of P.

The **dimension** of a polytope $P \subseteq \mathbb{R}^n$ is the maximum number of affinely independent points in P minus 1. P is **full-dimensional** if its dimension is n. An inequality $c^T x \leq \alpha$ is **valid** for $P \subseteq \mathbb{R}^n$ if $P \subseteq \{x \in \mathbb{R}^n \mid c^T x \leq \alpha\}$. If $c^T x \leq \alpha$ is valid then $F := \{x \in P \mid c^T x = \alpha\}$ is a **face** of P. A **facet** is a face of dimension one less than the dimension of P. An important theorem of polyhedral theory states that for a full-dimensional polytope every facet is defined by a unique (up to multiplication by a positive constant) inequality (i. e., if $F = \{x \in P \mid c^T x = \alpha\} = \{x \in P \mid d^T x = \beta\}$ is a facet of P and $c^T x \leq \alpha$ and $d^T x \leq \beta$ are valid for P then $c = \rho d$ for some $\rho > 0$),

and moreover, that every system of inequalities describing P completely must contain, for each facet F of P, at least one inequality defining F. This shows that in order to describe P in the form $P = \{x \mid Ax \leq b\}$ one has to know the inequalities defining facets of P. "Hunting" inequalities that define facets of certain polytopes associated with combinatorial optimization problems is one of the main issues in polyhedral combinatorics.

Let us now turn to the max-cut problem. Suppose a graph $G = (V, E)$ with edge weights c_{ij} for $ij \in E$ is given. We associate with G the real vector space $\mathbb{R}^E$, where the components of the vectors are indexed by the elements of E. For each cut $\delta(W)$, $W \subseteq V$, we define its **incidence vector** $\chi^{\delta(W)} \in \mathbb{R}^E$ by setting $\chi_e^{\delta(W)} = 1$ if $e \in \delta(W)$ and $\chi_e^{\delta(W)} = 0$ if $e \notin \delta(W)$. This yields a 1-1-correspondence of the cuts with certain $\{0,1\}$-vectors in $\mathbb{R}^E$. The **cut polytope** $\mathrm{CUT}(G)$ of G is the convex hull of all incidence vectors of cuts of G, i. e.,

$$\begin{aligned}\mathrm{CUT}(G) &= \mathrm{conv}\{\chi^{\delta(W)} \in \mathbb{R}^E \mid W \subseteq V\} \\ &= \{x \in \mathbb{R}^E \mid x = \lambda_1 \chi^{\delta(W_1)} + \ldots + \lambda_k \chi^{\delta(W_k)} \\ &\qquad \text{for some } k \leq |E| + 1, \text{ some } W_1, \ldots, W_k \subseteq V, \\ &\qquad \text{and some } \lambda_1, \ldots, \lambda_k \geq 0 \text{ with } \lambda_1 + \ldots + \lambda_k = 1\}.\end{aligned}$$

The problem of finding a cut $\delta(W)$ in G with $c(\delta(W))$ as large as possible can be written (considering c as a vector in $\mathbb{R}^E$) as the linear program

$$\max\{c^T x \mid x \in \mathrm{CUT}(G)\},$$

since the vertices of the polytope $\mathrm{CUT}(G)$ are exactly the incidence vectors of the cuts of G, and vice versa. In order to apply linear programming techniques to solve this linear program one has to represent $\mathrm{CUT}(G)$ as the solution set of an inequality system. General results in polyhedral combinatorics imply that, since the max-cut problem is $\mathcal{NP}$-hard, one cannot expect to find a complete system describing $\mathrm{CUT}(G)$. But — as we shall see later — also partial systems may be useful in solving the max-cut problem. The polyhedral structure of $\mathrm{CUT}(G)$ and the closely related bipartite subgraph polytope has, for instance, been studied in BARAHONA, GRÖTSCHEL & MAHJOUB (1985), BARAHONA & MAHJOUB (1983). We summarize here some of the results known about the facial structure of $\mathrm{CUT}(G)$ mentioned in BARAHONA & MAHJOUB (1983).

The cut polytope is full-dimensional, i. e., $\dim(\mathrm{CUT}(G)) = |E|$. This implies that each facet-defining inequality is unique up to multiplication by a constant. Since $\mathrm{CUT}(G)$ is in the unit hypercube of $\mathbb{R}^E$, the trivial inequalities $0 \leq x_e \leq 1$ are valid for $\mathrm{CUT}(G)$.

(5.1) Theorem. *For $e \in E$, the following statements are equivalent:*

(a) $x_e \geq 0$ defines a facet of $\mathrm{CUT}(G)$.

(b) $x_e \leq 1$ defines a facet of $\mathrm{CUT}(G)$.

(c) e does not belong to a cycle with three edges (triangle).

□

A consequence of (5.1) for the ground state problem, for instance, is that, for all edges e of a typical 2- or 3-dimensional grid model, the inequalities $0 \leq x_e \leq 1$ define facets of CUT(G) since grids do not contain triangels. But if an exterior magnetic field with nonzero interaction with all spins is added, none of these inequalities defines a facet of CUT(G).

We know from graph theory that a cut and a cycle intersect in an even number of edges. This observation yields that the **odd cycle inequalities**

$$x(F) - x(C \setminus F) \leq |F| - 1 \quad \text{for all cycles } C \subseteq E \text{ and all } F \subseteq C, |F| \text{ odd}$$

are satisfied by all incidence vectors of cuts. A **chord** of a cycle C is an edge of G which joins two nodes of C but does not belong to C.

(5.2) Theorem. *Let $C \subseteq E$ be a cycle and $F \subseteq C$, $|F|$ odd. Then*

$$x(F) - x(C \setminus F) \leq |F| - 1$$

defines a facet of CUT(G) *if and only if C has no chord.*

□

A graph is called a **bicycle p-wheel** if it consists of a cycle of length p and two further nodes adjacent to each other and to every node of the cycle.

(5.3) Theorem. *Let (W, F) be a bicycle $(2k+1)$-wheel, $k \geq 1$, contained in G. Then the inequality*

$$x(F) \leq 2(2k+1)$$

defines a facet of CUT(G).

□

(5.4) Theorem. *Let $K_p = (W, F)$ be a complete subgraph of order p of G. Then the K_p-inequality*

$$x(F) \leq \left\lceil \frac{p}{2} \right\rceil \left\lfloor \frac{p}{2} \right\rfloor$$

is valid for CUT(G); *this inequality defines a facet of* CUT(G) *if and only if p is odd.*

□

There are further classes of facets of CUT(G) known. In particular, there exist interesting methods to construct new facet defining inequalities from given facet defining inequalities. Especially notable are the techniques of changing the sign of a cut and of subdividing an edge. The exact definitions of these operations are technically a little complicated and therefore omitted.

Observe that, for many classes of graphs, the number of facet defining inequalities listed in Theorems (5.1),...,(5.4) grows exponentially with $|E|$. In fact, these are by far not all inequalities that define CUT(G) completely. Thus, the whole approach seems somehow ridiculous since we replace an optimization problem over $2^{|V|-1}$ points by an optimization problem over even more inequalities. However, there are general

results that show that the number of inequalities is not important. Important for the polynomial time solvability of a linear programming problem is whether one can check in polynomial time whether a given point satisfies all inequalities or not.

Let us state this more precisely for the cut polytope CUT(G). Let $\mathcal{K}$ be a class of linear inequalities valid for CUT(G); for instance, $\mathcal{K}$ could be any of the classes listed in (5.1),...,(5.4). The **separation problem** for $\mathcal{K}$ is the following:

> *Given $y \in \mathbb{Q}^E$, decide whether y satisfies all inequalities in $\mathcal{K}$, and if y does not, find an inequality in $\mathcal{K}$ violated by y.*

The importance of the separation problem stems from the fact that the polynomial time solvability of the separation problem for $\mathcal{K}$ implies the existence of a polynomial time algorithm for the optimization problem $\max\{c^T x \mid x$ satisfies all inequalities in $\mathcal{K}\}$ — see GRÖTSCHEL, LOVÁSZ & SCHRIJVER (1981, 1987). Although the optimization algorithm obtained this way (it utilizes the ellipsoid method) does not seem to be practically efficient, experience gained in the recent years shows that these optimization problems can indeed be solved reasonably well in practice by using the simplex method plus certain special techniques — see for instance CROWDER & PADBERG (1980), CROWDER, JOHNSON & PADBERG (1983), GRÖTSCHEL & HOLLAND (1985), GRÖTSCHEL, JÜNGER & REINELT (1984).

This is an astonishing recent development that shows a somewhat funny interaction between theory and practice. First, there were general cutting plane algorithms (of Gomory-type, say) that performed rather poor in practice. Then a few special cutting plane methods — based on the separation principle — were designed, but were not tried for many really large scale problems. After that, the ellipsoid method (and simultaneous diophantine approximation) came up and proofs were given that one can obtain (theoretically) good algorithms with it. (Most of this theory is summarized in GRÖTSCHEL, LOVÁSZ & SCHRIJVER (1987).) The combination of these techniques did not work in practice. But using the theory as a guideline, replacing the ellipsoid method by the simplex method and enhancing it with various heuristic features resulted in algorithms with quite good empirical behaviour.

Concerning the separation problem for the classes of inequalities listed in (5.1),..., (5.4) the following is known. We assume that a graph $G = (V, E)$ and a point $y \in \mathbb{Q}^E$ are given. We want to solve the separation problem for y.

Trivial inequalities (5.1): The separation problem is trivial. We simply substitute y into the inequalities $0 \leq x_e \leq 1$, $e \in E$.

Odd cycle inequalities (5.2): (We can assume here that $0 \leq y_e \leq 1$ holds.) We define a new graph $H = (V' \cup V'', E' \cup E'' \cup E''') = (W, F)$ which consists of two copies of G, say $G' = (V', E')$ and $G'' = (V'', E'')$, and the following additional edges E'''. For each edge $uv \in E$ we create the two edges $u'v''$ and $u''v'$. The edges $u'v' \in E'$ and $u''v'' \in E''$ get the weight y_{uv}, while the edges $u'v''$, $u''v' \in E'''$ get the weight $1 - y_{uv}$. For each pair of nodes $u', u'' \in W$ we calculate a shortest (with respect to the weights just defined) (u', u'')-path in H. Such a path contains an odd number of edges of E''' and corresponds to a closed walk in G containing u. Clearly, if the shortest of these (u', u'')-paths in H has length less than 1 there exists a cycle $C \subseteq E$ and an edge set

$F \subseteq C$, $|F|$ odd, such that y violates the corresponding odd cycle inequality. (C and F are easily constructed from a shortest path.) If the shortest of these (u', u'')-paths has length at least 1, then y satisfies all these inequalities. So the separation problem can be solved in polynomial time.

Bicycle wheel inequalities (5.3): GERARDS (1985) has shown that the separation problem for this class of inequalities can be reduced to a sequence of shortest path calculations in a similar way as described above for the odd cycle inequalities. Hence, this separation problem is also polynomially solvable.

K_p-inequalities (5.4): Trivially, for p fixed, one can check all K_p-inequalities in polynomial time by enumeration — but this is ridiculous. It is not known whether there is a polynomial time algorithm to solve the separation problem for all complete subgraph inequalities. It is also not known whether there is a good algorithm to solve the separation problem for the class of inequalities obtained from the K_5-inequalities by edge subdivision and changing the sign of a cut. The latter is a challenging open problem.

The remarks above show that the following LP-relaxation of the max-cut problem can be solved in polynomial time

$$\begin{array}{rl} \max c^T x & \\ 0 \leq x_e \leq 1 & \text{for all } e \in E, \\ x(F) - x(C \setminus F) \leq |F| - 1 & \text{for all cycles } C \subseteq E \text{ and} \\ & \text{all } F \subseteq C, |F| \text{ odd}, \\ x(F) \leq 2(2k+1) & \text{for all bicycle} \\ & (2k+1)\text{-wheels } (W, F). \end{array}$$

In the special application we are going to treat (planar grids with periodic boundary conditions and exterior magnetic filed), bicycle p-wheels do not occur; thus we can disregard these inequalities and concentrate on the remaining two classes. (Actually the same is true for the K_p-inequalities $p \geq 5$, but not for their edge subdivisions.) Let us therefore define the polytope

$$\begin{array}{rl} P_C(G) := \{x \in \mathbb{R}^E \mid 0 \leq x_e \leq 1 & \text{for all } e \in E, \\ x(F) - x(C \setminus F) \leq |F| - 1 & \text{for all cycles } C \subseteq E \text{ and all} \\ & F \subseteq C, |F| \text{ odd}\}. \end{array}$$

Observe that

$$\mathrm{CUT}(G) = \mathrm{conv}\{x \in P_C(G) \mid x \text{ integer}\},$$

so

$$\begin{array}{l} \max c^T x \\ x \in P_C(G) \\ x \text{ integer} \end{array}$$

is an integer programming formulation of the max-cut problem. Moreover, the following has been shown by BARAHONA & MAHJOUB (1983).

(5.5) Theorem. *$P_C(G) = \mathrm{CUT}(G)$ if and only if G is not contractible to K_5.*

□

This theorem and our remarks above prove that the max-cut problem is solvable in polynomial time for the class of graphs not contractible to K_5. Since — by the Wagner-Kuratowski theorem — planar graphs are those graphs which are not contractible to K_5 or $K_{3,3}$ we have $P_C(G) = \mathrm{CUT}(G)$ for all planar graphs. So this observation also implies the polynomial time solvability of the max-cut problem for planar graphs and graphs not contractible to K_5, respectively, in a way completely different from the methods of Orlova & Dorfman, Hadlock and Barahona.

Some observations in spin glass theory can be related to our approach. For the ground state problem in spin glasses, TOULOUSE (1977) introduced the concept of "frustrated contours" and states "There is no way of choosing the orientations of the site spins around a frustrated contour without frustrating at least one bond". If one translates this sentence into our terminology this suggests to consider the following system of inequalities

$$\sum_{ij \in C} x_{ij} \geq 1 \quad \text{for all frustrated contours } C \subseteq E,$$
$$x_{ij} \geq 0 \quad \text{for all } ij \in E,$$

where frustrated contours are cycles in the interaction graph G with an odd number of negative interactions. For the 2-dimensional grid case or any planar interaction graph, this system defines an integral polyhedron. This follows from the Chinese-Postman-Theorem of Edmonds and Johnson, as has been pointed out in BARAHONA, MAYNARD, RAMMAL & UHRY (1982). For random-field Ising ferromagnets the system also defines an integral polyhedron. This is a consequence of the max-flow min-cut theorem of Ford and Fulkerson and was shown by BARAHONA (1985). This system, however, is not integral for 3-dimensional Ising grid models, but it provides a tight LP-relaxation, as can be seen from the computational experience with it reported in BARAHONA & MACCIONI (1982). It is also not integral for planar or toroidal spin glass systems with exterior magnetic field, the case we are going to consider in the sequel.

6. A Cutting Plane Method

We shall now describe how the theory described in Section 5 can be applied. We want to design a so-called cutting plane algorithm that will be based on the simplex method. The simplex method — although not a polynomial time algorithm from a theoretical point of view — shows very good behaviour in practice and is therefore prefered to the ellipsoid (or other) methods.

The idea underlying such a cutting plane method is the following. We choose a system S of linear inequalities whose solution set P contains $\mathrm{CUT}(G)$ and for which the separation problem can be solved in polynomial time. Moreover, S has to have the property that $\mathrm{CUT}(G) = \mathrm{conv}\{x \in P \mid x \text{ integral}\} = \mathrm{conv}\{x \mid x \text{ satisfies all inequalities in } S \text{ and } x \text{ integral}\}$. Our aim is to solve $\max c^T x$, $x \in P$ (instead of

$\max c^T x$, $x \in \mathrm{CUT}(G)$). If the optimum solution over P is integral it is the incidence vector of a cut and thus we are done. If this is not the case we have to resort to branch & bound (in principle). The system S we choose is (usually) very large and it is impossible to write down all inequalities in polynomial time. Therefore we proceed as follows. We choose an initial "small" system $S_1 \subseteq S$ and solve the LP $\max c^T x$, x satisfies all inequalities in S_1. We check whether the optimum solution of this LP is the incidence vector of a cut. If this is so, we are done. If not, we enter a "separation phase" to find inqualities in S violated by the optimum solution. Such inequalities are called **cutting planes**, since they are used to cut off the present (infeasible) optimum LP-solution. If cutting planes have been found they are added to the current system S_1 and the process is repeated. If not, our separation algorithms yield a proof that the current optimum solution is an optimum solution for $\max c^T x$, $x \in P$ that is nonintegral. Then branch & bound must be applied.

Let us mention that the choice of the system S is very important. One should choose a system that is a rather tight relaxation of $\mathrm{CUT}(G)$ and, moreover, all inequalities in S should define facets of $\mathrm{CUT}(G)$ because facets define — in a sense that can be made precise — the deepest possible cutting planes. It has been observed empirically that cutting plane methods that use facet defining inequalities are by far superior compared with methods using just some valid inequalities. This fact is one of the reasons why the theory described in Section 5 was developed.

Our cutting plane algorithm for the max-cut problem was not implemented to treat all possible graphs. Our objective was the development of a special purpose computer code for max-cut problems coming up in the spin glass application. Although we exploited the special structure of this problem class (e. g., by designing special purpose cutting plane generation heuristics) our approach is general in principle. Only a few heuristics have to be exchanged and some data structures have to be modified in order to treat arbitrary graphs.

We study 2-dimensional Ising spin glasses on a grid with nearest neighbor interactions, an exterior magnetic field and periodic boundary conditions. This leads to the class of graphs consisting of a $k \times k$-grid embedded on a torus and a further node joined to all grid nodes. So, for $n := k^2$, our graphs have $n + 1$ nodes and $m = 3n$ edges.

Now we outline our strategy for generating and eliminating cutting planes. For the set S of inequalities to be considered we have chosen the trivial inequalities (5.1) and the odd cycle inequalities (5.2). The initial set S_1 consists of just the upper and lower bounds on the variables. The nonnegativity constraints and the upper bounds (of value 1) on the variables are automatically handled by the simplex method and no separation routine has to be designed. We implemented the exact separation algorithm for the odd cycle inequalities outlined in Section 5. Using Dijkstra's shortest path method and labelling techniques, an $O(n^3)$-implementation can be achieved. For practical purposes this is rather slow. Therefore, we have added faster heuristics for finding violated odd cycle inequalities in order to avoid calling the exact separation routine. These heuristics run in linear or at most quadratic time and usually find many odd cycle inequalities violated by the current optimum LP solution. But if they do not produce a violated odd cycle inequality there is no guarantee that such an inequality

does not exist. In this case the exact separation method has to be called. We describe the heuristics in the order we call them in the algorithm.

Suppose $y \in \mathbb{Q}^E$ is the optimum solution of the last linear program. We have to check whether y is the incidence vector of a cut, and if not, find odd cycle inequalities violated by y, if there are any.

CIBCAG (**C**heck **I**ntegrality, **B**ipartiteness, **C**ut, **a**nd **G**enerate Cutting Planes) For $0 \leq \varepsilon \leq \frac{1}{2}$ we define the graph $G_\varepsilon = (V, E_\varepsilon)$ as follows:

$$E_\varepsilon := \{e \in E \mid y_e \leq \varepsilon \text{ or } y_e \geq 1 - \varepsilon\}.$$

We try to 2-color the nodes of G_ε with red and green, say. First we pick an arbitrary node $v \in V$ and color it red. For all neighbors w of v in G_ε we do the following: If w is not colored, w receives the color of v if $y_{vw} \leq \varepsilon$, otherwise w receives the complementary color. If w is already colored, there are two cases. If w has the same color as v and $y_{vw} \leq \varepsilon$ or if v and w have complementary colors and $y_{vw} \geq 1 - \varepsilon$, we continue. Otherwise we have found a cycle C with an odd number of edges of value at least $1-\varepsilon$. Let F be the set of these edges. We check whether $y(F) - y(C \setminus F) > |F| - 1$. If this is the case, a violated odd cycle inequality is found. When all neighbors of v have been considered, we pick a new, colored node, consider its neighbours etc. and proceed in breadth first search manner.

If y is integral — which we check on the run — and not a cut, this procedure guarantees that a violated odd cycle inequality will be found. So, if for an integral y, CIBCAG does not produce a violated inequality, y is the incidence vector of a maximum weight cut in G. The breadth first search tree built up by CIBCAG allows us to generate the violated odd cycle inequalities efficiently. The worst-case running time of our implementation of CIBCAG depends on the structure of G_ε and is between $O(n)$ and $O(|E| \log n)$. Empirically it is $O(n)$ and extremely fast.

GEN4CYC (**Gen**erate **4**-**Cyc**les) Due to the special structure of our graph, the unchorded 4-cycles of the graph correspond exactly to the grid squares. There are n such 4-cycles. We scan through all these and check each of the eight associated odd cycle inequalities for violation. This can be done in $O(n)$ time.

GEN3CYC (**Gen**erate **3**-**Cyc**les) All 3-cycles (triangles) in G must contain the node 0 corresponding to the exterior magnetic field. By scanning through all grid edges vw we check the four possible odd cycle inequalities that can be derived from the triangle $0vw$. This algorithm has time complexity $O(n)$.

SHOC (**S**panning Tree **H**euristic for **O**dd **C**ycles) We calculate a maximum and a minimum weight spanning tree $T_{\max}$ and $T_{\min}$ of G where an edge e has weight y_e. For any non-tree edge e, we consider its fundamental cycle C and try to find an odd cardinality subset $F \subseteq C$ such that the corresponding odd cycle inequality is violated by y. In the "$T_{\max}$-case", we expect F to contain $C \setminus e$, and in the "$T_{\min}$-case", we hope $F = \{e\}$. Using Kruskal's algorithm, this heuristic runs in time $O(n \log n)$ on the average, and $O(n^2)$ in the worst case.

The above described heuristics are called in the following order.

- CIBCAG with user specified parameter EPSILON; if this way no cut can be generated, CIBCAG tries again with EPSILON = 0.49
- GEN4CYC if less than GENCYC-LIM cutting planes have been produced by CIBCAG
- GEN3CYC if less than GENCYC-LIM cutting planes have been produced by CIBCAG and GEN4CYC
- SHOC if less than SHOC-LIM cutting planes have been produced in the previous steps.

Finally the exact separation routine named OC (**O**dd **C**ycles) is called if all heuristics together found less than OC-LIM cutting planes. This kind of parameterization keeps the program flexible by allowing us to test various cutting plane generation strategies.

Elimination of cutting planes. To keep the linear program "handy" we eliminate inequalities in the following way: Whenever the objective function value has decreased by more than some DELTA (to be specified) compared to the previous solution value, all inequalities nonbinding at the current optimum solution are eliminated, otherwise no elimination is performed.

The features described above are the basic components of our cutting plane algorithm. The algorithm provably produces an optimum vertex of the polytope $P_C(G)$. If this algorithm finds an optimum solution which is integral, it provides a ground state of the spinglass. The algorithm, however, carries no guarantee to find an optimum integral solution. Formally, we could add a branch & bound phase to give such a guarantee. But, as shown in Section 7, it turned out that this is unnecessary.

7. Some Computational Results

To give an idea about what problem sizes can be handled by the method outlined in Section 6 we report here about some typical running times. Let us mention that it is a nontrivial task to tune such a cutting plane method so that it is practically efficient. That is, by performing computational experiments one has to find out which choices of all the parameters left open in our description, result in good running times and a numerically stable method. Crucial issues are the following:

(7.1) One has to determine a parameter MAXGEN which controls the maximum number of cutting planes added in one iteration. Choosing MAXGEN too small results in too many iterations of the cutting plane generation phase, choosing it too large produces too large linear programs, hardly solvable by the current LP-methods.

(7.2) One has to decide in which order the cutting plane heuristics are called and whether a further heuristic should be called, although a previous one found some cutting planes.

In our case, good choices for (7.1) and (7.2) turned out to be the most important factors. Further, but not so crucial prameters are:

(7.3) The parameter DELTA used to eliminate cutting planes.

(7.4) The choice of ε for the graph G_ε considered in the heuristic CIBCAG.

It is impossible to report here how these and other selections have been made and describe exactly what our parameters are. The paper BARAHONA, GRÖTSCHEL, JÜNGER & REINELT (1986) contains many details of this type. Nevertheless, we are still experimenting with our code, since the results are not completely satisfactory in one case to be explained later.

Tables 1 and 2 contain the computational results for 30×30 and 40×40 grids, respectively, with periodic boundary conditions and exterior magnetic field. The columns of these tables have to be interpreted as follows.

Column 1	$\hat{=}$	h	=	strength of the exterior magnetic field,
Column 2	$\hat{=}$	Ph	=	number of cutting plane phases = number of LP's solved,
Column 3	$\hat{=}$	4C	=	number of calls of GEN4CYC,
Column 4	$\hat{=}$	3C	=	number of calls of GEN3CYC,
Column 5	$\hat{=}$	OC	=	number of calls of OC,
Column 6	$\hat{=}$	#iter	=	total number of simplex pivots used for solving all the LP's,
Column 7	$\hat{=}$	size	=	number of rows of the final LP,
Column 8	$\hat{=}$	time	=	total computation time in min:sec including input, output etc.,
Column 9	$\hat{=}$	Energy	=	ground state energy per spin,
Column 10	$\hat{=}$	Mag	=	ground state magnetization per spin.

The magnetization per spin of a spin configuration is defined as the number of spins whose magnetic orientation agrees with that of the field minus the number of those with opposite orientation divided by the total number of spins.

Note that, in the current version of our code, the spanning tree heuristic SHOC described in Section 6 is not called at all. For our special toroidal grid graphs it did not contribute to running time improvements.

The experiments were executed on an IBM 3081-D32. The running times reported are total running times including input, output etc. Our code is written in ECL (an extension of PL/I) and uses IBM's LP-solver MPSX as a subroutine.

The results of Tables 1 and 2 are for the Gaussian model. The total sequence of experiments was conducted as follows. First, for all next neighbor pairs i, j on a toroidal $k \times k$ grid the interaction J_{ij} was chosen according to a Gaussian distribution with mean 0 and standard deviation 1. All spins interact with the exterior field with $h = 4.0$. Then we determine a ground state for this system. In the next run the spin interactions remained the same, only the strength of the field was decreased from 4.0 to 3.8. The optimum basic solution for $h = 4.0$ was used as the starting basis for $h = 3.8$. This was repeated until $h = 0.0$. The 2Δ-case showing up in the 30×30 table will be explained later.

30×30									
h	Ph	4C	3C	OC	#Iter	Size	Time	Energy	Mag
4.0	4	3	3	0	941	1203	1:45.87	-4.0290	0.9622
3.8	1	1	1	0	17	1212	0:11.96	-3.8371	0.9578
3.6	1	1	1	0	26	1249	0:12.62	-3.6470	0.9444
3.4	1	1	1	0	28	1291	0:13.24	-3.4591	0.9356
3.2	1	1	1	0	13	1331	0:12.12	-3.2734	0.9200
3.0	1	1	1	0	30	1359	0:12.55	-3.0916	0.8933
2.8	1	1	1	0	58	1422	0:14.06	-2.9139	0.8822
2.6	1	1	1	0	72	1472	0:13.99	-2.7416	0.8333
2.4	1	1	1	0	88	1544	0:15.28	-2.5784	0.8067
2.2	2	2	2	0	118	1682	0:26.49	-2.4200	0.7800
2.0	2	2	2	0	137	1794	0:25.72	-2.2652	0.7644
1.8	2	2	2	0	158	1905	0:27.20	-2.1164	0.7067
1.6	2	2	2	0	178	2027	0:30.14	-1.9797	0.6622
1.4	3	3	3	0	230	2190	0:42.01	-1.8511	0.6222
1.2	2	2	2	0	284	2358	0:36.00	-1.7310	0.5822
1.0	3	3	3	0	424	2593	0:56.78	-1.6205	0.5200
0.8	2	2	2	0	521	2738	0:54.17	-1.5212	0.4556
0.6	6	6	6	0	860	2846	2:03.89	-1.4373	0.3578
0.4	10	10	10	0	1411	3014	4:16.49	-1.3754	0.2467
0.2	7	7	7	0	2920	2990	6:55.81	-1.3334	0.1489
2Δ	49	49	49	2	4191	3048	22:38.29	-1.3187	0.0222
0.0	4	4	4	2	151	3080	5:03.82	-1.3187	0.0222

Table 1

40 × 40									
h	Ph	4C	3C	OC	#Iter	Size	Time	Energy	Mag
4.0	5	2	2	0	1623	1885	5:42.05	-4.0207	0.9625
3.8	1	1	1	0	36	1936	0:38.98	-3.8285	0.9575
3.6	2	2	2	0	60	2032	1:09.72	-3.6387	0.9425
3.4	1	1	1	0	49	2099	0:40.79	-3.4508	0.9325
3.2	1	1	1	0	68	2085	0:42.24	-3.2654	0.9187
3.0	2	2	2	0	93	2285	1:12.29	-3.0849	0.8925
2.8	1	1	1	0	106	2423	0:46.19	-2.9082	0.8737
2.6	2	2	2	0	124	2584	1:17.68	-2.7355	0.8537
2.4	2	2	2	0	172	2333	1:17.48	-2.5682	0.8200
2.2	2	2	2	0	212	2585	1:23.02	-2.4080	0.7825
2.0	2	2	2	0	210	2795	1:24.07	-2.2547	0.7512
1.8	2	2	2	0	271	3047	1:30.20	-2.1079	0.7175
1.6	2	2	2	0	325	3280	1:34.49	-1.9688	0.6700
1.4	4	4	4	0	493	3678	2:56.80	-1.8395	0.6262
1.2	2	2	2	0	589	4014	2:00.92	-1.7195	0.5787
1.0	4	4	4	0	751	4298	3:27.25	-1.6129	0.4975
0.8	3	3	3	0	960	4539	3:24.35	-1.5192	0.4475
0.6	3	3	3	0	1367	4743	4:33.53	-1.4375	0.3725
0.4	8	8	8	0	2739	4820	12:15.49	-1.3710	0.2837
0.2	7	7	7	0	5181	5035	19:55.33	-1.3236	0.2012

Table 2

Table 3 reports on a series of experiments with ± 1-iteractions on a 22×22-grid with periodic boundary conditions and exterior magnetic field. Column headings are the same as those for Tables 1 and 2, also the sequence of runs was executed analogously. The percentage of negative interactions ranges from 12% to 32% with an increase of 4% per step. The strength of the exterior field starts in each case with $h = 1.0$ and is decreased to 0.2 by 0.1. For each percentage of negative interactions the 1.0-field problem was solved "from scratch" whereas for the eight subsequent problems with lower field the optimum basic solution of the previous problem served as a starting basis. The interactions were chosen as follows. First we determined a random linear ordering of the grid edges. In the first run, the first 12% of the edges in this order received a -1-value all other edges a $+1$-value. In the second run the first 16% of the edges were considered negative interactions, all other edges positive interactions, etc. By comparing running times and the total number of pivots of the various instances one can see that by increasing the percentage of negative interactions and by decreasing the field strength the problem becomes harder for our code. Nevertheless, considering the fact that we deal with an $\mathcal{NP}$-hard problem, the running times seem quite reasonable.

22 × 22										
% neg	h	Ph	4C	3C	OC	#Iter	Size	Time	Energy	Mag
12%	1.0	3	3	3	0	201	302	0:12.64	-2.5496	0.9711
	0.9	1	1	1	0	49	311	0:02.44	-2.4525	0.9711
	0.8	0	0	0	0	0	311	0:01.33	-2.3554	0.9711
	0.7	0	0	0	0	1	311	0:01.33	-2.2583	0.9711
	0.6	1	1	1	0	8	315	0:02.48	-2.1620	0.9587
	0.5	0	0	0	0	0	315	0:01.33	-2.0661	0.9587
	0.4	2	2	2	0	46	350	0:04.43	-1.9702	0.9587
	0.3	2	2	2	0	66	389	0:05.10	-1.8744	0.9587
	0.2	1	1	1	0	28	399	0:03.13	-1.7810	0.9298
16%	1.0	6	6	6	0	295	422	0:22.34	-2.4215	0.9298
	0.9	1	1	1	0	58	443	0:03.44	-2.3285	0.9298
	0.8	0	0	0	0	0	443	0:01.35	-2.2355	0.9298
	0.7	0	0	0	0	2	443	0:00.00	-2.1426	0.9298
	0.6	2	2	2	0	38	474	0:05.22	-2.0496	0.9298
	0.5	0	0	0	0	1	474	0:01.32	-1.9566	0.9298
	0.4	2	2	2	0	67	517	0:05.12	-1.8636	0.9298
	0.3	2	2	2	0	82	558	0:05.62	-1.7727	0.9091
	0.2	1	1	1	0	88	585	0:04.67	-1.6843	0.8802
20%	1.0	5	5	5	0	463	548	0:26.08	-2.2893	0.9008
	0.9	2	2	2	0	72	587	0:05.88	-2.2017	0.8760
	0.8	0	0	0	0	2	587	0:01.34	-2.1140	0.8760
	0.7	1	1	1	0	12	589	0:00.00	-2.0264	0.8760
	0.6	2	2	2	0	52	614	0:05.45	-1.9388	0.8760
	0.5	1	1	1	0	27	629	0:04.05	-1.8512	0.8760
	0.4	2	2	2	0	120	682	0:07.09	-1.7686	0.8264
	0.3	3	3	3	0	151	755	0:11.16	-1.6868	0.8017
	0.2	2	2	2	0	156	833	0:09.57	-1.6083	0.7686
24%	1.0	6	6	6	0	604	718	0:33.61	-2.1612	0.8554
	0.9	2	2	2	0	118	759	0:06.90	-2.0764	0.8471
	0.8	1	1	1	0	11	766	0:03.56	-1.9917	0.8471
	0.7	1	1	1	0	20	775	0:02.86	-1.9070	0.8471
	0.6	2	2	2	0	75	785	0:05.22	-1.8248	0.8099
	0.5	1	1	1	0	52	806	0:05.11	-1.7438	0.8099
	0.4	4	4	4	0	195	890	0:13.92	-1.6661	0.7769
	0.3	3	3	3	0	206	975	0:13.74	-1.5963	0.6653
	0.2	4	4	4	0	239	1003	0:17.38	-1.5339	0.6033

Table 3

22 × 22										
% neg	h	Ph	4C	3C	OC	#Iter	Size	Time	Energy	Mag
28%	1.0	6	6	6	0	685	776	0:38.05	-2.0248	0.8223
	0.9	2	2	2	0	154	815	0:07.17	-1.9450	0.7975
	0.8	0	0	0	0	12	815	0:01.32	-1.8653	0.7975
	0.7	2	2	2	0	76	838	0:07.85	-1.7876	0.7769
	0.6	2	2	2	0	166	859	0:09.94	-1.7140	0.7149
	0.5	3	3	3	0	125	881	0:09.95	-1.6426	0.7149
	0.4	2	2	2	0	259	965	0:14.08	-1.5777	0.6488
	0.3	2	2	2	0	248	993	0:13.80	-1.5169	0.6074
	0.2	5	5	5	0	405	1079	0:27.86	-1.4612	0.5083
32%	1.0	9	8	8	0	915	903	0:55.34	-1.9421	0.7769
	0.9	3	3	3	0	190	941	0:10.53	-1.8678	0.7438
	0.8	1	1	1	0	52	961	0:04.95	-1.7967	0.7066
	0.7	1	1	1	0	108	1001	0:06.62	-1.7281	0.6860
	0.6	3	3	3	0	214	1037	0:12.99	-1.6645	0.6116
	0.5	2	2	2	0	133	1056	0:09.35	-1.6054	0.5826
	0.4	3	3	3	0	328	1113	0:21.37	-1.5521	0.4917
	0.3	1	1	1	0	234	1144	0:12.69	-1.5070	0.4504
	0.2	4	4	4	0	424	1178	0:29.46	-1.4675	0.4215

Table 3

Note that our code only solves an LP-relaxation of the max-cut problem and that there is no guarantee that the optimum LP-solution is integral (and thus a maximum weight cut), unless we enter an additional branch & bound phase. However, in all cases reported here the LP solution was indeed integral and no further enumerative technique had to be used. There are a few exceptions to this and they always occured at 0-field. This is a strange phenomenon that we cannot explain properly yet.

Recall that the max-cut problem is $\mathcal{NP}$-hard for toroidal graphs with exterior field, but solvable in polynomial time without exterior field and ± 1-interactions. It is not known whether this also holds for Gaussian interactions and 0-field. This made us expect the 0-field problem to be much easier. The contrary is the case, and our code is able to solve quite large (much larger than reported here) $k \times k$ toroidal grid problems with strong field (just look at the rather small running times in Tables 1 and 2), while 0-field problems take much longer. There is one trick that worked in the 30×30-problem of Table 1. Before solving a 0-field problem we solve a problem with $h = 2\Delta$, Δ a very small number. We observed frequently that the solution of this problem also gives a ground state for the $h = 0$ problem. This is the case in Table 1, but not in Table 2. Here, in fact, we obtained a nonintegral optimum solution in the 2Δ- and 0-field case. So we could not read a ground state from the linear program. The same phenomenon occurs, even more frequently, in the ± 1-model, and although the 0-field problem here can be solved in polynomial time it seems to be harder for our approach.

We are currently working on a feature that will be employed in cases of small fields. Our algorithm will be enhanced by max-cut heuristics that exploit the structure of fractional LP solutions and are based on fixing techniques similar to those used in standard branch & bound methods. Again, such observations show that structural knowledge has to be utilized to achieve reasonable running times. We are confident that the methods outlined above work so that we can solve 0-field problems as easily as the others.

Let us mention a few additional informations we can derive from our method.

Frequently, one is not only interested in calculating just one ground state; knowing all ground states or knowing certain properties of ground states (like the rigidity of bonds) are of equal importance. To obtain this type of information the algorithmic approach described in this paper is particularly well suited.

The computational experience described in this section shows that for all but a few ground state problems the optimum solution of our LP-relaxation of the max-cut problem is integral. This gives all tools of postoptimality analysis of linear programming to our disposal. For instance, using LP-duality we can often derive results like:

- the ground state is unique,
- a certain cluster of spins will have the same relative orientation in all ground states (rigidity of the ground states).

Actually, in the examples before, we frequently observed that the ground states are unique. If the ground state is not unique we can use reduced cost criteria to exhibit alternative ground states. We cannot produce all ground states (in reasonable time) since there may be exponentially many. Even so, LP-duality can be used to prove that certain edges have value one or zero in all ground states. From this information we can derive that certain clusters of spins have the same relative orientation in all ground states. And thus, our methods enable us to determine the existence of long distance order. Let us remark that information of this type is hardly obtainable from enumerative or branch & bound methods.

There are a number of variations of the ground state problem that are relevant for the understanding of spin glasses. Let us mention just one question due to A. J. Bray and M. A. Moore (personal communication with W. Kinzel). Suppose a spin glas (Gaussian model) on a toroidal $k \times k$-grid without exterior magnetic field is given. Let ω be a ground state and E_g be its energy. We pick a column of spins of the grid and we take an adjacent column of spins. The spins in the first column are fixed to its orientation in ω while the spins in the second column are fixed to the orientation opposite to the one in ω. This fixing can be achieved by adding an "artificial" exterior field and use $\pm M$ and zero forces, M a very large number. Let E_{+-} be the ground state energy subject to this side condition. Set $\Delta E := |E_g - E_{+-}|$. It has been conjectured that $(\Delta E)^2 \sim k^\gamma$. In particular, that γ is negative. For the straightforward generalizations of this question to 3-dimensional grids, it is believed that γ is positive. Using the algorithm described in this paper, estimates for γ can be obtained numerically.

It is also possible to contribute to an open problem mentioned in ANGLES D'AURIAC & MAYNARD (1984) by calculating ground states under the additional condition that the magnetization is zero. We can "misuse" the exterior magnetic field to for-

mulate this problem as an integer linear program, but further polyhedral studies are necessary before this approach can be used algorithmically.

8. Heuristic Methods

Proposals for heuristics to "solve" the max-cut problem are uncountable. The most successful ones — at least with respect to the applications in physics considered here — seem to be exchange methods, as proposed by Kernighan and Lin, for instance, and simulated annealing. We will not discuss the methods here in detail, but would like to mention that our cutting plane algorithm makes it possible to find out how good these heuristics really are. In fact, we have made a large number of computer experiments — exact details will be reported elsewhere — to compare the solutions found by simulated annealing and other heuristics with optimum solutions.

Exchange methods of Kernighan-Lin type did not show striking results, they were outperformed by other improvement techniques developed by us. In particular, these improvement methods are very fast and can handle problems of almost any size, a point of considerable importance in application oriented optimization. This is not so for simulated annealing.

To obtain a "good" simulated annealing code is a difficult matter by itself. Annealing schedules, experiments per temperature, restart features etc. have to be chosen carefully so that good solutions are obtained in "finite time". Of course, by increasing time bounds, one eventually will obtain better results, but then it might even be better to perform complete enumeration. We have compared simulated annealing with the results of other heuristics and we can summarize our findings in these experiments as follows:

(8.1) Very long running times (many steps at constant temperature, very slow cooling) lead to very good results. Minimum energy and magnetization of the ground state are closely approximated.

(8.2) Long running times (in our case, this means running times that are about as long as the cutting plane method needs to produce an exact ground state) result in relatively good approximations of the ground state energy and magnetization, but sometimes runaways, where simulated annealing fails, occur.

(8.3) Improvement heuristics obtain spin configurations with energies similar to those found by simulated annealing with strategy (8.2), but in much shorter time. However, for small fields, the magnetization of the ground state is not as well approximated as by simulated annealing.

(8.4) Giving to simulated annealing no more time than the improvement heuristics need, much worse results are obtained than by improvement methods.

Based on these observations we have determined a set of parameters that, using the running time bounds of (8.2), shows satisfactory behaviour. It works with restarts and typically has a behaviour as shown in Figure 1.

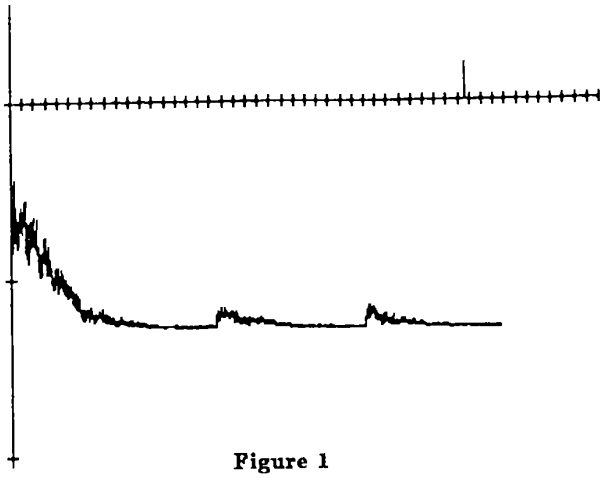

Figure 1

On the x-axis of Figure 1 the experiments are recorded linearly. The y-axis represents the energy. The long bar above the x-axis shows the experiment where the best spin configuration was found. After a certain number of experiments usually no significant changes of the energy of the spin configuration obtained by single spin flips occur and a restart is performed by accepting solutions with higher energy with higher probability. The time bound given to simulated annealing was about the running time our cutting plane code needed to solve the given problem exactly. This seems fair to us.

In Figure 2 we show one example of 11 experiments where the energy obtained by simulated annealing is compared to the minimum energy of a ground state. This is a Gaussian 30×30 problem with exterior field of strength $h = 0, 0.2, \ldots, 2.2$ and periodic boundary conditions. Nearest neighbor spin interaction values are chosen according to a Gaussian distribution and remain fixed for all the experiments. For $h = 0, 0.2, \ldots, 2.2$ the ground state energy per spin is determined with our cutting plane algorithm and approximated by simulated annealing. The lower curve gives the exact value of the ground state energy per spin, the upper curve the approximate value.

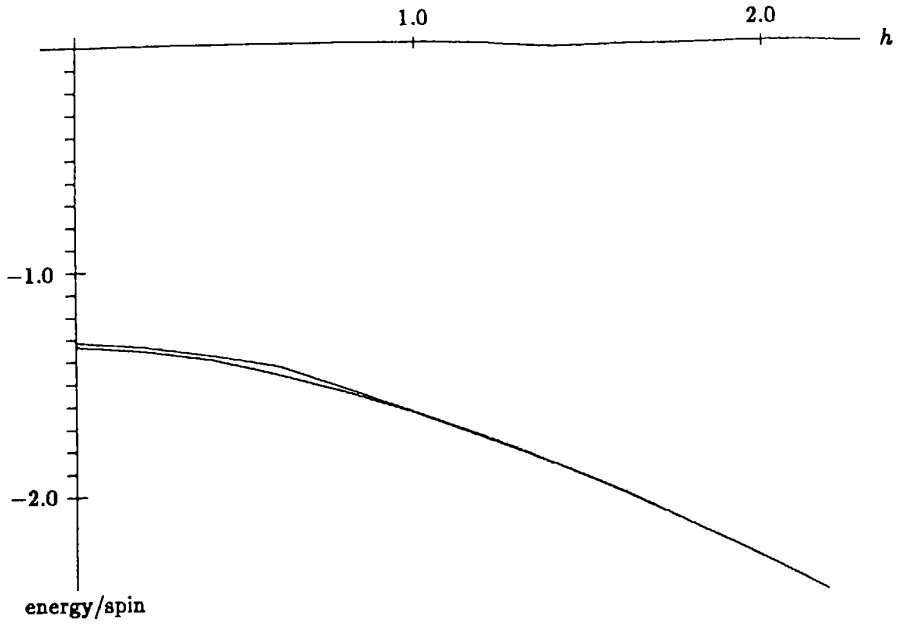

Figure 2

It turns out that the solution values of simulated annealing are good approximations of the minimum energy and get better with increasing strength of the exterior field. Moreover, the ground state magnetization is also closely approximated — see Figure 3 for the same series of experiments as described for Figure 2.

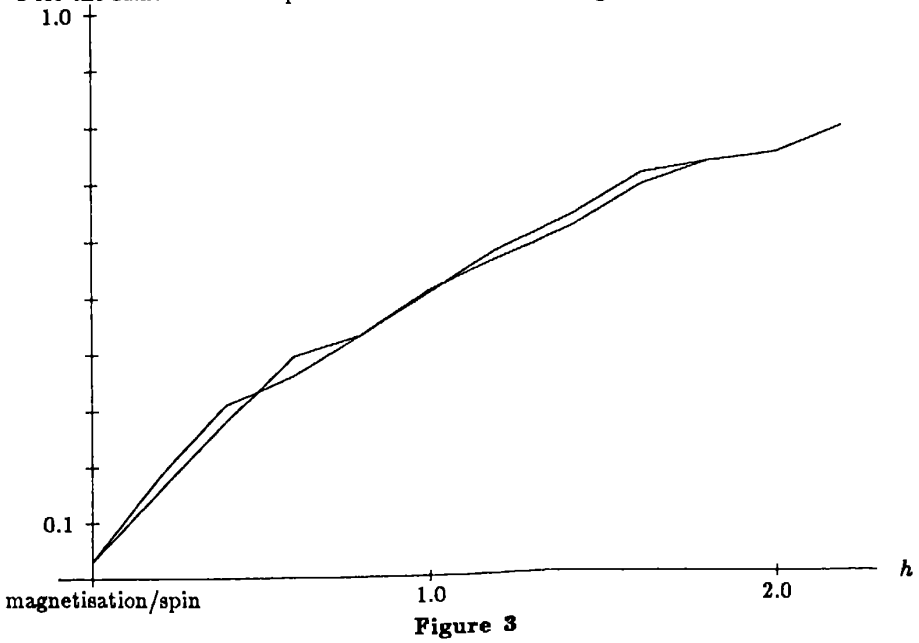

Figure 3

However, we also compared the configurations obtained by simulated annealing to optimum solutions (as we could prove with our algorithm these are frequently unique), and we realized that they rarely show structural similarities. One (random) example of this type is shown in Figures 4 and 5. The spin configuration in Figure 4 is an optimum solution to a Gaussian 30×30 grid problem with periodic boundary conditions and an exterior field of strenght 0; for the same instance the spin configuration shown in Figure 5 was found by simulated annealing.

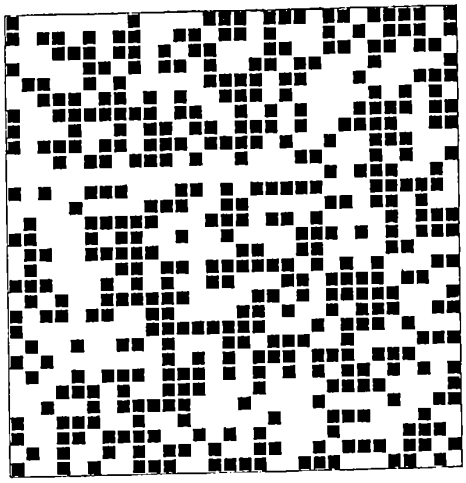

Figure 4

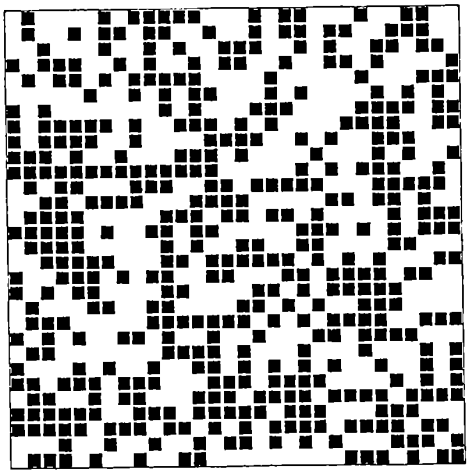

Figure 5

9. Final Remarks

The recent years have shown increasing interest of physicists in combinatorial optimization problems, in particular in the traveling salesman problem (TSP), and it seems, sometimes, that optimization is becoming a part of physics. Indeed, Toulouse (1983) states "... that the main spinoff of spin glass physics, so far, has been the transfer of Monte-Carlo annealing techniques to optimization problems". It is true that simulated annealing arose considerable excitement among mathematicians and a lot of research is devoted to it at present. Related procedures like Monte Carlo methods, the Metropolis algorithm, or the evolution techniques of Rechenberg have been around for quite some time but were neglected for several reasons. The main objection against these heuristics and against simulated annealing — that still holds — is that these methods are much too slow for practical purposes, say a TSP on several thousand cities to determine the route of an NC machine, and that they are inferior to special purpose heuristics developed for such problems. But the activities of physicists have given the area of stochastic methods a new impetus that has certainly put these methods into a new perspective.

TOULOUSE (1983) also writes "A less wellknown stream exists in the opposite direction with the use of Edmonds' algorithm to find various exact ground state properties of spin glasses and this stream may well grow in the future". We consider the work presented here as a contribution to the growth of this stream from optimization to physics. In fact, it would be fair to call it "the contribution of the traveling salesman problem to physics". Namely, the subarea of polyhedral combinatorics dealing with hard problems — on which our approach is based — was largly developed in terms of the TSP. Among important papers that have been written on this subject let us mention DANTZIG, FULKERSON & JOHNSON (1954), in many respects a seminal paper of this field. Research in this area has been quite active in the fifties, calmed down in the sixties, and was taken up again in the seventies to become one of the most flourishing branches in combinatorial optimization. Among the main contributions was the development of the polyhedral theory for the TSP — see, e. g., GRÖTSCHEL (1977), GRÖTSCHEL & PADBERG (1979a,b) — and the design and implementation of successful cutting plane algorithms — see for instance CROWDER & PADBERG (1980). Most of this development is documented in the recent book LAWLER, LENSTRA, RINNOOY KAN & SHMOYS (1985).

The methods developed for the TSP turned out to be general. In particular, the whole approach, the proof techniques, and the principle algorithmic ideas for cutting plane methods carry over to almost all other combinatorial optimization problems. The success in this area also motivated us to try the polyhedral approach to the max-cut problem with the aim to develop a practically efficient cutting plane algorithm for certain ground state problems. Our work is not complete yet, but the computational results reported here seem promising.

References

J. C. Angles d'Auriac & R. Maynard (1984), "On the random antiphase state in the $\pm J$ spin glass model in two dimensions", *Solid State Communications* 49 (1984) 785–790.

F. Barahona (1981), "Balancing signed toroidal graphs in polynomial time", Departamento de Matemáticas, Universidad de Chile, 1981.

F. Barahona (1982), "On the computational complexity of Ising spin glass models", *J. Phys. A: Math. Gen.* 15 (1982) 3241–3253.

F. Barahona (1983), "The max cut problem in graphs not contractible to K_5", *Operations Research Letters* 2 (1983) 107–111.

F. Barahona (1985), "Finding ground states in random-field Ising ferromagnets", *J. Phys. A: Math. Gen.* 18 (1985) L673–L675.

F. Barahona, M. Grötschel & A. R. Mahjoub (1985), "Facets of the bipartite subgraph polytope", *Mathematics of Operations Research* 10 (1985) 340–358.

F. Barahona, M. Jünger, M. Grötschel & G. Reinelt (1986), "An application of combinatorial optimization to statistical physics and circuit layout design", Preprint no. 95, Institut für Mathematik, Universität Augsburg, 1986.

F. Barahona & E. Maccioni (1982), "On the exact ground states of three-dimensional Ising spin glasses", *J. Phys. A. Math. Gen.* 15 (1982) L611–L615.

F. Barahona & A. R. Mahjoub (1983), "On the cut polytope", Report No. 83 271-OR, Institut für Ökonometrie und Operations Research, Universität Bonn, Bonn, W. Germany, 1983.

F. Barahona, R. Maynard, R. Rammal & J. P. Uhry (1982), "Morphology of ground states of a two dimensional frustration model", *J. Phys. A. Math. Gen.* 15 (1982) L673–L699.

H. Crowder, E. L. Johnson & M. Padberg (1983), "Solving large-scale zero-one linear programming problems", *Operations Research* 31 (1983) 803–834.

H. Crowder & M. Padberg (1980), "Solving large-scale symmetric travelling salesman problems to optimality", *Management Science* 26 (1980) 495–509.

G. B. Dantzig, D. R. Fulkerson & S. M. Johnson (1954), "Solution of a large-scale traveling-salesman problem", *Operations Research* 2 (1954) 393–410.

A. Dress (1986), "Computing spin-glass Hamiltonians", Preprint, Universität Bielefeld, 1986.

J. Edmonds & E. L. Johnson (1973), "Matching, Euler tours, and the Chinese postman", *Mathematical Programming* 5 (1973) 88–124.

J. Fonlupt, A. R. Mahjoub & J. P. Uhry (1984), "Composition of graphs and the bipartite subgraph polytope", R. R. no. 459, Laboratoire ARTEMIS (IMAG), Université de Grenoble, 1984.

M. R. Garey & D. S. Johnson (1979), *Computers and Intractability: A Guide to the Theory of $\mathcal{NP}$-completeness*, Freeman, San Francisco, 1979.

A. M. H. Gerards (1985), "Testing the odd bicycle wheel inequalities for the bipartite subgraph polytope", *Mathematics of Operations Research* 10 (1985) 359–360.

M. Grötschel (1977), *Polyedrische Charakterisierungen kombinatorischer Optimierungsprobleme*, Verlag Anton Hain, Meisenheim am Glan, 1977.

M. Grötschel & O. Holland (1985), "Solving matching problems with linear programming", *Mathematical Programming* 33 (1985) 243–259.

M. Grötschel, M. Jünger & G. Reinelt (1984), "A cutting plane algorithm for the linear ordering problem", *Operations Research* 32 (1984) 1195–1220.

M. Grötschel, L. Lovász & A. Schrijver (1981), "The ellipsoid method and its consequences in combinatorial optimization", *Combinatorica* 1 (1981) 169–197.

M. Grötschel, L. Lovász & A. Schrijver (1987), *Geometric Algorithms and Combinatorial Optimization*, Springer, Heidelberg, 1987 (to appear).

M. Grötschel & G. L. Nemhauser (1984), "A polynomial algorithm for the max-cut problem on graphs without long odd cycles", *Mathematical Programming* 29 (1984) 28–40.

M. Grötschel & M. W. Padberg (1979a), "On the symmetric travelling salesman problem I: inequalities", *Mathematical Programming* 16 (1979) 265–280.

M. Grötschel & M. W. Padberg (1979b), "On the symmetric travelling salesman problem II: lifting theorems and facets", *Mathematical Programming* 16 (1979) 281–302.

M. Grötschel & W. R. Pulleyblank (1981), "Weakly bipartite graphs and the max-cut problem", *Operations Research Letters* 1 (1981) 23–27.

M. Grötschel & K. Truemper (1986), "Decomposition and optimization over cycles in binary matroids", Preprint no. 108, Institut für Mathematik, Universität Augsburg, 1986.

F. Hadlock (1975), "Finding a maximum cut of a planar graph in polynomial time", *SIAM Journal on Computing* 4 (1975) 221–225.

A. Hartwig, F. Daske & S. Kobe (1984), "A recursive branch-and-bound algorithm for the exact ground state of Ising spin-glass models", *Computer Physics Communications* 32 (1984) 133–138.

R. M. Karp (1972), "Reducibility among combinatorial problems", in: R. E. Miller & J. W. Thatcher (eds.), *Complexity of Computer Computations*, Plenum Press, New York, 1972, 85–103.

E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan & D. Shmoys (1985), *The Traveling Salesman Problem*, Wiley, Chichester, 1985.

I. Morgenstern (1983), "Numerical simulations of spin glasses", in: J. L. van Hemmen & I. Morgenstern (eds.), Heidelberg Colloquium on Spin Glasses, Lecture Notes in Physics no. 192, Springer, Berlin, 1983, 305–327.

I. Morgenstern & K. Binder (1980), "Magnetic correlations in two-dimensional spin-glasses", *Physical Review B* 22 (1980) 288–302.

G. I. Orlova & Y. G. Dorfman (1972), "Finding the maximal cut in a graph", *Engrg. Cybernetics* 10 (1972) 502–506.

J. C. Picard & H. D. Ratliff (1975), "Minimum cuts and related problems", *Networks* 5 (1975) 357–370.

G. Toulouse (1977), "Theory of the frustration effect in spin glasses: I", *Communications on Physics* 2 (1977) 115–119.

G. Toulouse (1983), "Frustration and disorder, new problems in statistical mechanics, spin glasses in a historical perspective", in: J. L. van Hemmen & I. Morgenstern (eds.), Heidelberg Colloquium on Spin Glasses, Lecture Notes in Physics no. 192, Springer, Berlin, 1983, 2–17.

K. Wagner (1937), "Über eine Erweiterung des Satzes von Kuratowski", *Deutsche Mathematik* 2 (1937) 280–285.

M. Yannakakis (1978), "Node- and edge-deletion $\mathcal{NP}$-complete problems", Proc. 10th Ann. ACM Symposium on Theory of Computing, Association of Computing Machinery, New York, 1978, 253–264.