Perspectives in Mathematics
Anniversary of Oberwolfach 1984

Developments in Combinatorial Optimization

MARTIN GRÖTSCHEL

Universität Augsburg, Mathematisches Institut,
Memminger Straße 6, D-89 Augsburg (FRG)

Abstract

This paper describes developments in combinatorial optimization in the last thirty years and outlines trends of future research. Section 1 introduces a few representative problems of the subject and mentions some applications. Polynomial time solvability and $\mathcal{NP}$-completeness of combinatorial optimization problems are discussed in Section 2. Polyhedral combinatorics, and the theory and practice of cutting planes are surveyed in detail in Section 3. Many of the landmarks of these topics are mentioned, open problems and future developments are outlined. Section 4 describes some of the relations of combinatorial optimization to other branches of mathematics. In particular, some of the major recent breakthroughs that arose from applying the results of other fields to combinatorial optimization (and vice versa) are mentioned. Moreover, lists of promising research areas and concrete open problems are given.

1 Introduction and Applications

The roots of combinatorial optimization lie in easy-looking problems (mostly of economical or technical nature) of the following kind.

(1.1) Given n cities and distances between these, find a roundtrip through all cities of shortest total length (the TRAVELLING SALESMAN PROBLEM).

(1.2) Given a road network connecting two cities A and B, find a shortest route (with respect to time or distance) from city A to B (the SHORTEST PATH PROBLEM).

(1.3) Determine the layout of a printed circuit board so that no two lines (or as few lines as possible) intersect—except in their endpoints (the PLANARITY PROBLEM).

(1.4) Find a simultaneous permutation of the rows and columns of an (n, n)-matrix such that the sum of the entries above the main diagonal is as large as possible (TRIANGULATION OF INPUT OUTPUT MATRICES).

(1.5) Given m machines and n jobs which consist of a given sequence of operations on some of the machines, suppose that for each operation a

processing time on the associated type of machine is given and that each job has a due date. Find a feasible assignment of operations to machines such that as few of the due dates as possible are violated (a SCHEDULING PROBLEM).

(1.6) Determine the routes of the garbage collection trucks of a city so that all side conditions with respect to working time, capacity of trucks etc. are satisfied and the total distance covered by all trucks is minimized (a ROUTING PROBLEM).

(1.7) Given, at a university or school, a number of courses, class-rooms and teachers, assign teachers to courses and courses to classrooms so that no two courses are in the same classroom at the same time, no two teachers give the same course, teachers are able to give the course etc. (an ASSIGNMENT PROBLEM).

(1.8) Given a pipeline system between a "source" and a "sink", determine the maximal amount of "flow" from the source to the sink through the network subject to capacity constraints etc. (a FLOW PROBLEM).

The problems mentioned above are examples of so-called combinatorial optimization problems. Formally, a *combinatorial optimization problem* can be described by a set of *instances* and a *task*. Each instance is given by a pair (S, c) where S is a finite set and $c: S \to \mathbb{R}$ any function, and for each instance (S, c) the task is to find an element $s \in S$ whose function value $c(s)$ is maximum (or minimum).

The elements of S are called *feasible solutions*, an element of S maximizing (or minimizing) c over S is called *optimal solution* of the instance (S, c). The function c is called *objective function.*

Clearly, every combinatorial optimization problem can be solved by *enumeration*, i.e. by scanning through S, evaluating for each $s \in S$ the objective function $c(s)$, and choosing the element s^* with highest (or lowest) value $c(s^*)$. Thus for a combinatorial optimization problem to be nontrivial we have to assume that the set S and the function c of every instance are structured in some way, i.e. that S and c are describable in much less space than the cardinality of S.

In very many cases S is a set of subsets of a finite set E, and the objective function is specified by giving a value $c(e)$ to each element $e \in E$ and setting $c(T) := \sum_{e \in T} c(e)$ for every set $T \in S$. Problems of this type are called *linear objective combinatorial optimization problems*. These are the best-studied and most important combinatorial optimization problems, and we will restrict our attention to this class of problems in the sequel. Thus we will focus on problems where each instance is given by a finite set E (the *ground set*), a (usually implicitly defined) set $\mathcal{F} \subseteq 2^E$ of feasible solutions and a function $c: E \to \mathbb{R}$ where the task is to find a set $F \in \mathcal{F}$ such that $c(F) := \sum_{e \in F} c(e)$ is maximum or minimum.

For most of the problems (1.1), ..., (1.8) it is trivial to see how they can be phrased in the way defined above. Consider, for instance, the travelling

salesman problem (1.1). With each instance of this problem we associate a complete graph $K_n = (V, E)$ with n nodes (representing the cities) where each pair of distinct nodes is linked by an edge (representing a road connection). The "ground set" is the edge set E. The feasible solutions $\mathcal{F} \subseteq 2^E$ are the "roundtrips" or "tours", which are sets of n edges forming a cycle which passes through every node (hamiltonian cycles). The "value" of each edge $ij \in E$ is the distance c_{ij} between its two endnodes (cities) i and j, and so the "length" of a tour T is $c(T) := \sum_{ij \in T} c_{ij}$. The task is to find a tour T^* such that $c(T^*)$ is minimum.

Most of the problems studied in the early days of this subject came from operations research, industrial management, computer science and military applications. But problems of this kind arise almost everywhere, and therefore combinatorial optimization has found successful applications in fields like archeology, biology, chemistry, geography, linguistics, physics, sociology and others.

This survey is not meant as an overview of the applications of combinatorial optimization. The reader interested in this should consult the appropriate sections of the classified bibliographies Kastning (1976), Hausmann (1978), von Randow (1982), the book Roberts (1978), or the papers Balas & Padberg (1975), Grötschel (1982), Iri (1983) which explain various applications (of special types) of combinatorial optimization problems.

2 Polynomial Time Solvability and $\mathcal{NP}$-Completeness

The theory of combinatorial optimization—at least in the way I view the subject—aims at a better mathematical understanding of the type of problems introduced in paragraph 1 with the ultimate goal to provide tools for the design of "efficient" algorithms for solving these problems.

The notion of efficiency needs, of course, some clarification. In the early days of combinatorial optimization the efficiency of an algorithm was usually tested empirically by programming the algorithm and running the code on several data sets, measuring time and storage space needed, and fitting these measures to some curves. Such empirical comparisons are still of great value for those who are using implemented algorithms in practice, but from a theoretical viewpoint they are quite unsatisfactory.

A theoretically more appealing concept of efficiency was brought into the field from complexity theory. The notions "solvable in polynomial time" and "NP-complete" introduced in the late sixties have considerably changed the way combinatorial optimizers look at their subject and put new research topics into focus.

In short and informally these concepts can be described as follows. First one has to fix a model of computation. Usually Turing machines or RAM machines are used (for a nonexpert in this field it is sufficient to consider a real world computer). Then one has to decide how the problem instances are to be encoded for the machine and how the "size" of an instance has to be measured.

The standard way to encode numbers (say integers) is to use their binary representation and so the *input length or size* of an integer is the number of digits of this representation. Graphs (a large number of combinatorial optimization problems can be formulated as problems concerning graphs) are usually encoded by means of adjacency or edge lists. The number of nodes and edges of a graph is a convenient measure of the input size of a graph. In this way (other structures can be handled similarly) with every instance of a combinatorial optimization problem an input size, which is an integral number, can be associated.

An algorithm for a combinatorial optimization problem is said to run in *polynomial time* (or to be a "good algorithm") if there is a polynomial $p: \mathbb{N} \to \mathbb{N}$ such that for every instance of the problem of input size at most n the running time of the algorithm is at most $p(n)$. The running time is measured by counting the "steps" the algorithm has to perform until termination, where steps are elementary operations (like comparisons, additions and multiplications) on the machine model considered. By $\mathcal{P}$ we denote the class of problems which can be solved by a polynomial time algorithm. Of course, an algorithm with running time n^{1000} is by no means good in practice, but, for large enough instances, it is still a lot better than an algorithm which performs 2^n steps. On the other hand, it has turned out that for many of the practically relevant problems—once their polynomial time solvability was discovered—good algorithms with low degree polynomial time bounds could be found, say of order at most n^3 or n^4.

The notion of polynomial time solvability was introduced by Edmonds (1965a) and Cobham (1965). It is customary to call problems which are solvable in polynomial time *easy*.

For the definition of "difficult" problems a few more technicalities are necessary. The success of this so-called "theory of $\mathcal{NP}$-completeness" rests on a fundamental result of Cook (1971) whose far reaching consequences for combinatorial optimization were recognized and popularized by Karp (1972).

To explain the ideas behind this theory let us consider a combinatorial minimization problem Π. We now want to solve the following decision problem. Given an instance (P, c) of Π and an additional number B, decide whether there is a feasible solution, say S, whose value $c(S)$ is at most as large as B. (Clearly, if we could find an optimum solution to (P, c) in polynomial time we could solve this decision problem in polynomial time by calculating the optimum value and comparing it with B.)

We say that problem Π belongs to the class of problems $\mathcal{NP}$ if Π has the following property. There is a polynomial time algorithm which does the following. If for an instance (P, c) and a bound B there is a solution $S \in P$ with $c(S) \leqq B$, then the algorithm can verify in polynomial time that $S \in P$ and $c(S) \leqq B$.

Note that the algorithm is not required to find S. The only thing the algorithm has to do is, given (P, c), B and S, check whether $S \in P$ and $c(S) \leqq B$ in polynomial time. The problems in $\mathcal{NP}$ are called *solvable in nondeterministic polynomial time*. The name stems from an equivalent definition of the class $\mathcal{NP}$

where algorithms are considered which are allowed to make guesses (nondeterministic steps).

For example, consider the travelling salesman problem (1.1). Suppose an instance of this problem and a bound B are given. Now we guess a set T of edges and mark it on the map with a red pencil. We choose a starting point and follow the red edges. If we return to the starting point, have passed all cities exactly once and have encountered all red edges then T is a roundtrip. Now we add up the distances associated with the red edges to obtain $c(T)$ and compare this value with B. This procedure is clearly a polynomial time algorithm of the type we consider, so the travelling salesman problem belongs to $\mathcal{NP}$.

It is obvious that $\mathcal{P} \subseteq \mathcal{NP}$. And it is intuitively convincing that not everything that can be guessed can also be constructed. But despite enormous research efforts in the last decade it could not be decided yet whether $\mathcal{P} \neq \mathcal{NP}$ or not. In my opinion this question is one of the major open problems in mathematics. $\mathcal{P} \neq \mathcal{NP}$ (together with further known results) would imply that there is a host of combinatorial optimization problems relevant for real world applications which are inherently intractable.

The class $\mathcal{NP}$ contains a further important class of problems which are called $\mathcal{NP}$-complete. This class is denoted by $\mathcal{NPC}$. Let us call a problem Π $\mathcal{NP}$-complete (or simply *hard*) if it has the following property: $\Pi \in \mathcal{NP}$, and if Π can be solved in polynomial time, then every problem in $\mathcal{NP}$ can be solved in polynomial time. The $\mathcal{NP}$-complete problems are in a sense the hardest problems in $\mathcal{NP}$, since in order to show that $\mathcal{P} = \mathcal{NP}$ it suffices to prove for just one $\mathcal{NP}$-complete problem that it is in $\mathcal{P}$.

The classification of combinatorial optimization problems into hard and easy ones was one of the main streams of research of this field in the seventies. It turned out that most of the practically relevant problems are in fact $\mathcal{NP}$-complete. This, of course, had significant implications on the directions of further theoretical and algorithmic investigations about which we will report in the subsequent sections.

The results of the studies done in this area of complexity theory are documented in the excellent book Garey & Johnson (1979) where for several hundred generic problems (and some thousand variants) their membership in $\mathcal{P}$ and $\mathcal{NPC}$ is recorded. An ongoing guide of David Johnson in the Journal of Algorithms documents the current progress in this subject.

In the meantime for almost all major problems it has been decided whether they are in $\mathcal{P}$ or in $\mathcal{NPC}$. Today's research in this area mainly concentrates on exploring the borderlines between $\mathcal{P}$ and $\mathcal{NPC}$, to unify results and get a deeper understanding of "difficulty".

Almost no significant progress with respect to the $\mathcal{P} \neq \mathcal{NP}$ problem has been made. There are two major (and usually powerful) techniques available to attack this problem. With "simulation" one could try to prove $\mathcal{P} = \mathcal{NP}$, while "diagonalization" might be a good tool to show that $\mathcal{P} \neq \mathcal{NP}$. Baker, Gill & Solovay (1975), however, showed in a beautiful paper (using oracle techniques

which we do not want to describe here) that these two methods do not suffice to solve this important question. It is even conceivable that this problem cannot be solved within the framework of formal set theory, see Hartmanis & Hopcroft (1976). Yet, almost all researchers in combinatorial optimization assume $\mathcal{P} \neq \mathcal{NP}$ as a working hypothesis.

The theory of $\mathcal{NP}$-completeness has provided a rough but very useful classification scheme for combinatorial optimization problems. When studying a problem, nowadays a standard first step is to check whether it belongs to $\mathcal{P}$ or is $\mathcal{NP}$-complete.

Graph theory and combinatorial optimization were two of the main areas of application of the theory of $\mathcal{NP}$-completeness. In the meantime complexity theory has "invaded" other mathematical fields as well, in particular disciplines like number theory or algebra where "sizes of numbers" or "lengths of proofs" are considered.

I think that today complexity theory plays two important roles. It provides a language to distinguish between hard and easy problems and its concepts are convenient tools for the analysis of algorithms.

3 Cutting Planes and Polyhedral Combinatorics

A simple idea but one of the most fruitful approaches in combinatorial optimization is to formulate combinatorial optimization problems as integer linear programs or even as linear programs. Theoretically this is rather easy, but to make this idea an algorithmic success quite a number of new mathematical concepts and algorithmic design techniques had to be developed. For more detailed surveys of this subject the reader should consult the excellent papers Pulleyblank (1983) und Schrijver (1983). Moreover, the book Schrijver (1984a) will appear soon which treats the whole area in depth and contains almost all of the results known to date.

3.1 *Polyhedra Associated with Combinatorial Optimization Problems*

We shall now describe a method with which a polyhedron can be associated with (almost) every combinatorial optimization problem. Suppose we have a minimization problem where each instance is given by a ground set E, a set $\mathcal{I} \subseteq 2^E$ of feasible solutions, and an objective function $c: E \to \mathbb{R}$. Let $\mathbb{R}^E$ denote the real $|E|$-dimensional vector space where for every vector $x \in \mathbb{R}^E$ its components are indexed by the elements of E, i.e. $x = (x_e)_{e \in E}$. For every subset $F \subseteq E$ we define its *incidence* (or characteristic) *vector* $\chi^F = (\chi^F_e)_{e \in E}$ by

$$\chi^F_e = 1 \text{ if } e \in F \quad \text{and} \quad \chi^F_e = 0 \text{ if } e \notin F.$$

With the set $\mathcal{F}$ of feasible solutions we associate the convex hull of the incidence vectors of the elements of $\mathcal{F}$, i.e. a polytope $P_{\mathcal{F}} \subseteq \mathbb{R}^E$ defined by

(3.1) $P_{\mathcal{F}} := \text{conv}\{\chi^F \in \mathbb{R}^E | F \in \mathcal{F}\}$.

The combinatorial optimization problem

(3.2) $\min\{c(F) | F \in \mathcal{F}\}$

can now be written (considering c as a vector in $\mathbb{R}^E$) as

(3.3) $\min\{c^T x | x \in P_{\mathcal{F}}\}$.

Every feasible solution $F \in \mathcal{F}$ corresponds to a vertex of the polytope $P_{\mathcal{F}}$ and vice versa. As $P_{\mathcal{F}}$ is a polytope, problem (3.3) is a linear program, and it is well-known that, for every objective function, the program (3.3) has an optimum solution which is a vertex of $P_{\mathcal{F}}$. Thus, by solving (3.3) one can obtain an optimum incidence vector of a set $F \in \mathcal{F}$ which in turn is an optimum solution of (3.2).

Problem (3.3) can of course be solved in finite time by generating all the vertices of $P_{\mathcal{F}}$ (these are implicitly given through $\mathcal{F}$) and selecting the one with the best value. But this way nothing has been gained by the new representation of the combinatorial optimization problem.

Linear programs are usually given by a linear objective function and a system of linear equations and inequalities, and all (nontrivial) algorithms solving linear programming problems require such systems as input. Thus, in order to use the powerful tools and methods of linear programming it is necessary to find a linear system describing $P_{\mathcal{F}}$. Theoretically, the theorem of Weyl guarantees that for every polyhedron P there are a matrix A and a vector b such that $P = \{x | Ax \leqq b\}$. There are even constructive proofs of this theorem, but they are not effective in the sense that one can "easily read" from $\mathcal{F}$ an inequality system describing $P_{\mathcal{F}}$.

So in order to transform a combinatorial optimization problem Π into a linear programming problem one has to solve the following problem:

(3.4) *For every instance $(\mathcal{F}, c)$ of Π find a system of linear equations and inequalities describing the associated polytope $P_{\mathcal{F}}$.*

One of the (at present) most flourishing branches of combinatorial optimization, called *polyhedral combinatorics*, considers this task as one of its central research topics. We shall now survey some of the ideas developed in this area and some of the successes of this approach. And we shall mention interesting open research problems.

3.2 *Some Examples*

A *graph* $G=(V,E)$ consists of a finite nonempty set V of *nodes* and a set E of *edges* which are two-element subsets of V. (For ease of exposition we do not consider loops and multiple edges here.) The two elements of V, say i and j, forming an edge $e \in E$ are called the endnodes of e, and we write $e=ij$ instead of $e=\{i,j\}$. A *digraph* $D=(V,A)$ consists of a finite nonempty set V of nodes and a set $A \subseteq V \times V$ called *arcs*. If $a=(i,j) \in A$ then i is called the *tail* of arc a and j is called the *head* of a, i and j are also called the *endnodes* of a.

If $G=(V,E)$ $(D=(V,A))$ is a graph (digraph) and $W \subseteq V$ a set of nodes then $E(W)$ $(A(W))$ denotes the set of edges (arcs) with both endnodes in W; $\delta(W) \subseteq E$ denotes the set of edges with one endnode in W and the other in $V \setminus W$; $\delta^+(W) \subseteq A$ (resp. $\delta^-(W) \subseteq A$) denotes the set of arcs with tail (resp. head) in W and head (resp. tail) in $V \setminus W$. We write $\delta(v)$ instead of $\delta(\{v\})$ for $v \in V$.

Matchings

One of the first positive results in this area concerns matchings in bipartite graphs. It is generally attributed to Birkhoff (1946) and von Neumann (1953). A graph $G=(V,E)$ is called *bipartite* if V can be partitioned into two nonempty, disjoint subsets V_1, V_2 such that each edge has one of its endnodes in V_1 and the other in V_2, i.e. if $E=\delta(V_1)=\delta(V_2)$. A *matching* in G is a set $M \subseteq E$ of edges such that no two edges of M have a common endnode. The Birkhoff-von Neumann theorem characterizes the polytope associated with the matchings in a bipartite graph.

(3.5) **Theorem.** *Let $G=(V,E)$ be a bipartite graph. Then the convex hull of the incidence vectors of the matchings of G (the* **matching polytope** *of G) is the polytope defined by the following system of inequalities:*

$$(1) \qquad x_e \geqq 0 \quad \text{for all } e \in E,$$

$$(2) \qquad \sum_{e \in \delta(v)} x_e \leqq 1 \quad \text{for all } v \in V. \qquad \square$$

This theorem has some interesting consequences in graph theory. It implies, for instance, via LP-duality König's matching theorem

> „*The maximum cardinality of a matching in a bipartite graph is equal to the minimum cardinality of a set of nodes that meets all edges.*"

or Hall's famous marriage theorem.

It is easy to see that the system (1), (2) of (3.5) is not sufficient for the description of the matching polytope of nonbipartite graphs. For instance, if G is a cycle of length three then the vector $\frac{1}{2}(1,1,1)$ is a vertex of the polytope defined

by (1), (2). A major breakthrough was obtained by Edmonds, cf. Edmonds (1965a), (1965b), who found a complete system for the matching polytope in general.

(3.6) **Theorem.** *Let $G=(V,E)$ be a graph. Then the convex hull of the incidence vectors of the matchings of G is the polytope defined by*

$$(1) \qquad x_e \geqq 0 \qquad \text{for all } e \in E$$

$$(2) \quad \sum_{e \in \delta(v)} x_e \leqq 1 \qquad \text{for all } e \in E$$

$$(3) \quad \sum_{e \in E(W)} x_e \leqq (|W|-1)/2 \qquad \text{for all } W \subseteq V, |W| \text{ odd.} \quad \square$$

Theorem (3.6) is a deep result with many consequences in graph theory and combinatorial optimization. It has meanwhile found generalizations in many directions. The reader interested in this should consult section 6 of Schrijver (1983).

Theorem (3.6) looks—at first sight—quite useless from a linear programming point of view since the number of constraints (3) is exponential in the input size of G. So it already takes exponential time to input the inequalities (1), (2), (3), which means that no algorithm requiring the full system (1), (2), (3) of (3.6) can run in time polynomial in the input size of G. However, and this was the second major achievement, Edmonds was able to devise an algorithm for the solution of linear programs with constraint system (1), (2), (3) which runs in time polynomial in $|E|$. Edmonds exploited LP-duality theory, the complementary slackness theorem, and the fact that the system (1), (2), (3) has a "nice" implicit description which made it possible to avoid the use of all inequalities at once and to generate inequalities whenever necessary. We shall come back to this point in Section 3.4.

Matroids and Generalizations

Another class of polyhedra which is "well understood" is a class of polytopes associated with matroids. A matroid M on E is a pair $(E,\mathcal{I})$ where $\mathcal{I}$ is a subset of the set of all subsets of E satisfying

(3.7) $\emptyset \in \mathcal{I}$,

(3.8) $J \subseteq I \in \mathcal{I} \Rightarrow J \in \mathcal{I}$,

(3.9) $I, J \in \mathcal{I}, |I| < |J| \Rightarrow \exists e \in J \setminus I$ with $I \cup \{e\} \in \mathcal{I}$.

The elements of $\mathcal{I}$ are called the *independent sets* of M.

For instance, if $G=(V,E)$ is a graph, then the set $(E,\mathcal{I})$ where $\mathcal{I} := \{F \subseteq E \mid F \text{ is a forest (i.e. } F \text{ contains no cycle)}\}$ is a matroid on the edge set E of G (the so-called *forest matroid* of a graph). If A is a matrix with entries from some field K and with column index set E, then call a set $I \subseteq E$ independent if the column vectors of A corresponding to the indices of I are linearly independent. The pair $(E,\mathcal{I})$ defined this way is a matroid, the so-called *matric matroid.*

We recommend Welsh (1976) for more information about matroids. The survey paper Iri (1983) and the forthcoming book Recski (1984) particularly focus on applications of matroid theory.

Given a matroid $M=(E,\mathcal{I})$ and a function $c: E \to \mathbb{R}$, then the matroid optimization problem is to find an independent set I such that $c(I)$ is as large as possible. (A special case of this, e.g., is the problem to find a maximum (or minimum) forest in a graph.) The matroid polytope P_M is defined as follows

$$P_M := \text{conv}\{\chi^I \in \mathbb{R}^E \mid I \in \mathcal{I}\}.$$

It is very easy to solve matroid optimization problems with the famous *greedy algorithm.* (Set $I := \emptyset$. Choose an element $e \in E$ such that c_e is as large as possible. If $c_e \geqq 0$ and $I \cup \{e\} \in \mathcal{I}$ set $I := I \cup \{e\}$. Remove e from E and continue until E is empty.) Edmonds (1971) interpreted the greedy algorithm as a linear programming algorithm and derived from this the following characterization of the matroid polytope.

(3.10) **Theorem.** *Let $M=(E,\mathcal{I})$ be a matroid and P_M be the associated polytope, then*

$$P_M = \{x \in \mathbb{R}^E \mid \begin{array}{lll} (1) & x_e \geqq 0 & \text{for all } e \in E \\ (2) & \sum_{e \in F} x_e \leqq r(F) & \text{for all } F \subseteq E\}. \end{array} \quad \square$$

Above, for every set $F \subseteq E$, $r(F)$ denotes the largest cardinality of an independent set contained in F. The number $r(F)$ is called the *rank* of F. Edmonds (1970) was able to extend Theorem (3.10) to the intersection of two matroids as follows.

Let $M_1=(E,\mathcal{I}_1)$ and $M_2=(E,\mathcal{I}_2)$ be two matroids on the same ground set E. The pair $(E,\mathcal{I})$ with

$$\mathcal{I} := \mathcal{I}_1 \cap \mathcal{I}_2$$

is called the intersection of the matroids M_1 and M_2, and

$$P_{\mathcal{I}} := \text{conv}\{\chi^I \in \mathbb{R}^E \mid I \in \mathcal{I}\}$$

is called the *2-matroid intersection polytope.*

(3.11) **Theorem.**

$$P_{\mathcal{I}} = P_{M_1} \cap P_{M_2}, \text{ in particular}$$

$$P_{\mathcal{I}} = \{x \in \mathbb{R}^E \mid \ (1) \quad x_e \geqq 0 \qquad \text{for all } e \in E$$

$$(2) \ \sum_{e \in F} x_e \leqq \min\{r_1(F), r_2(F)\} \quad \text{for all } F \subseteq E\}. \ \square$$

In (2) above r_i denotes the rank function of M_i, $i = 1,2$. This theorem has many interesting special cases. For instance, it gives a complete linear description of the branching polytope of a digraph (see Edmonds (1967)), but the Birkhoff-von Neumann Theorem (3.5), König's matching theorem, Hall's marriage theorem, and Fulkerson's branching theorem can also be derived from (3.11). Optimizing over the intersection of three or more matroids is $\mathcal{NP}$-complete. This suggests that a similar result extending (3.11) to the intersection of three or more matroid polytopes is unlikely.

The results of Edmonds were the starting point of a new branch of combinatorial optimization, the theory of submodular functions. For a finite set E a function $f: 2^E \to \mathbb{R}$ is called *submodular* if

(3.12) $f(S \cup T) + f(S \cap T) \leqq f(S) + f(T)$ for all $S, T \subseteq E$.

The connection with matroids is the fact that matroid rank functions (see (3.10)) are special submodular functions satisfying in addition $r(\emptyset) = 0$, $S \subseteq T \subseteq E \Rightarrow r(S) \leqq r(T)$, and $r(S) \leqq |S|$ for all $S \subseteq E$. It was observed that most of the properties of matroids resp. matroid polyhedra are in fact consequences of the submodularity of the rank function.

With every submodular function $f: 2^E \to \mathbb{R}$ (without loss of generality we may assume $f(\emptyset) = 0$) one can associate the polyhedron

$$P_f := \{x \in \mathbb{R}^E \mid \sum_{e \in F} x_e \leqq f(F) \quad \text{for all } F \subseteq E\}.$$

It is possible to solve $\max\{c^T x \mid x \in P_f\}$ by means of an appropriately modified greedy algorithm, and one can show that the vertices of P_f are integral valued if f is integral valued. Moreover, the matroid intersection theorem (3.11) also extends to the intersection of two such polyhedra, see Edmonds & Giles (1977).

In the recent few years there has been an inflation of frameworks based on submodular functions combined with various graph theory concepts (in particular network flow theory). They aim at a unification of those parts of combinatorial optimization for which polyhedral results (like the one described above) or min-max theorems (like König's matching theorem) exist. Concepts like polymatroids (Edmonds (1970)), submodular flows (Edmonds & Giles (1977)), lattice polyhedra (Hoffman & Schwartz (1978)), generalized polymatroids (Frank (1984)), kernel systems (Frank (1979)), base polyhedra of submodu-

lar systems (Fujishige (1983)), polymatroid network flows (Hassin (1978)), (Lawler & Martel (1982)) and others are competing for the attraction of further investigators. Schrijver (1984b), (1984c) has surveyed, compared and extended these efforts.

It turns out that most of these frameworks are in a certain sense equivalent and that they indeed unify large parts of the existing theory. But it seems to me that the final word is not said yet and that it may take some more time to find a "best possible" setting of the theory of submodular functions (combined with graphs and digraphs) within combinatorial optimization.

All the structure results quoted so far are—of course—closely related to the algorithmic aspects of the optimization problems mentioned. A truly positive result of this theory is that most of the optimization problems associated with matroids, intersections of two matroids (polymatroids etc.) can be solved in polynomial time provided that one can check in polynomial time whether a set is independent in a matroid (an element of a polymatroid etc.) or not.

The most general result in this respect is due to Grötschel, Lovász & Schrijver (1981). It is based on the ellipsoid method (see Section 3.4) and shows that submodular functions can be minimized in polynomial time. More precisely,

(3.12) **Theorem.** *Let $f: 2^E \to \mathbb{Z}$ be a submodular function. Suppose that a positive integer B is known with $|f(S)| \leqq B$ for all $S \subseteq E$. Then there exists an algorithm which finds a set $S^* \subseteq E$ such that $f(S^*) \leqq f(S)$ for all $S \subseteq E$ and which runs in time polynomial in $|E|$, $\lceil \log B \rceil$ and the time necessary to evaluate the function f.* □

So in particular, if there is an algorithm to evaluate the function f in time polynomial in $|E|$ and $\lceil \log B \rceil$ (this is the case for all practically relevant problems), then the algorithm (3.12) is polynomial in the sense described in Section 2.

A further interesting aspect of submodular functions is that it is $\mathcal{NP}$-complete to maximize them. The same phenomenon is also known for convex functions $f: \mathbb{R}^n \to \mathbb{R}$ which are easy to minimize but difficult to maximize. It turns out that there is a close connection linking submodular and convex functions. In a sense submodular functions can be viewed as the discrete analogues of convex functions. An interesting survey illuminating this aspect and describing more results about and applications of submodular functions is the paper by Lovász (1983).

Matching theory and matroid theory are in several respects well-understood, thus there have been continuing attempts to unify these theories. It seems that the best setting of such a generalization is the following. Suppose $G = (V, E)$ is a graph and $M = (V, \mathcal{I})$ is a matroid on the node set V. If $c: E \to \mathbb{R}$ is an objective function then the *matroid matching problem* is to find an *independent matching* $F \subseteq E$ of maximum total weight $c(F)$. Here an independent matching is a set $F \subseteq E$ which is a matching of G, for which $V(F) := \{i \in V | i$ is contained in some edge of $F\}$ is independent in M. (It is easy to see that the graph matching

problem and the matroid intersection problem are special cases of the matroid matching problem.) However, it turned out, that the matroid matching problem is $\mathcal{NP}$-complete, see Lovász (1981). But Lovász, on the positive side, found a polynomial time algorithm to obtain a maximum cardinality matroid matching in case the given matroid is matric. This algorithm is one of the most involved algorithms known in this field. It is based on a geometric representation of the matroid matching problem and uses a number of ingenious new algorithms for problems in affine and projective geometry.

The weighted version of the matric matroid case is not solved yet. It seems plausible that this problem can also be solved in polynomial time, but what—probably—is lacking here, is a better understanding of the convex hull of the incidence vectors of the independent matchings, i.e. a description of this polytope by means of equations and inequalities.

Another outgrowth of matroid theory is the theory of greedoids. Greedoids have been defined in an attempt to obtain a better insight into the combinatorial structures for which the greedy method works.

A greedoid is a pair $(E, \mathcal{F})$ where E is a finite set and $\mathcal{F}$ is a subset of 2^E which satisfies (3.7) and (3.9). So a matroid is a greedoid satisfying (3.8) in addition. The main advocates of greedoids are B. Korte and L. Lovász who are currently working on a research program which aims at characterizing those combinatorial optimization problems which are greedoids, and obtaining richer substructures within the class of greedoids which give rise to certain min-max relations or polynomial time algorithms. For more information see Korte & Lovász (1983).

Shortest Paths, Cuts, and Flows

Up to now we have only considered combinatorial optimization problems which are solvable in polynomial time (with a few exceptions mentioned in side remarks). We shall now turn to a problem which is solvable in polynomial time only if the objective function is restricted in some way, and we shall indicate here some of the subtleties coming up in polyhedral combinatorics.

An instance of the shortest path problem can be described as follows.

Given a directed graph $D=(V, A)$, a function $c: A \to \mathbb{R}$, and two different nodes $s, t \in V$. An *(s, t)-path* P in D is a set of arcs $\{a_1, a_2, \ldots, a_k\}$ such that the tail of a_1 is s, the head of a_k is t, and the head of arc a_i is the tail of arc a_{i+1}, $i=1, \ldots, k-1$. Moreover, we require that no node appears in P more than twice as an endnode of an arc a_i, $i=1, \ldots, k$. (If P is an (s, t)-path and $(t, s) \in A$ then $P \cup \{(t, s)\}$ is a *directed cycle.*) The task is to find an (s, t)-path P such that $c(P)$ is as small as possible.

To treat the shortest path problem from a polyhedral point of view the first idea—of course—is to consider the polyhedron

$$P_{SP}(D) := \text{conv}\{\chi^P \in \mathbb{R}^A \mid P \subseteq A \text{ is an } (s,t)\text{-path}\}$$

and solve $\min c^T x$, $x \in P_{SP}(D)$. However, there is no explicit linear description of $P_{SP}(D)$ known to date. And it is very likely that we will never be able to obtain such a characterization since the shortest path problem (in this general formulation) is $\mathcal{NP}$-complete.

On the other hand this problem can be solved in polynomial time if the objective function is nonnegative. (In fact, one can do a little better, namely the problem is solvable in polynomial time if D contains no directed cycle such that the sum of the arc weights of the directed cycle is negative, see for instance Lawler (1976). We do not want to go into these details here.) And in this case a polyhedral result is available. The polytope one should consider is

$$P_S(D) := \text{conv}\{\chi^P \in \mathbb{R}^A \mid P \subseteq A \quad \text{contains an } (s,t)\text{-path}\}.$$

It is clear that for every objective function $c: A \to \mathbb{R}$ with $c(a) \geqq 0$, for all $a \in A$ there is always an optimum solution of $\min\{c^T x \mid x \in P_S(D)\}$ which is the incidence vector of an (s,t)-path. So, in this case the shortest path problem can also be solved via the linear program $\min\{c^T x \mid x \in P_S(D)\}$. The polytope $P_S(D)$ can be described as follows.

(3.13) **Theorem.** *Let $D = (V, A)$ be a digraph, and let s, t be two different nodes of V. Then*

$$P_S(D) = \{x \in \mathbb{R}^A \mid (1) \quad 0 \leqq x_a \leqq 1 \text{ for all } a \in A,$$
$$(2) \quad \sum_{a \in \delta^+(W)} x_a \geqq 1 \text{ for all } W \subseteq V \text{ with } s \in W, t \notin W\}. \quad \square$$

A set $\delta^+(W)$ of arcs with $s \in W$ and $t \notin W$ is called an (s,t)-*cut*. Another interesting combinatorial optimization problem is, given a digraph $D = (V, A)$ and arc capacities $c(a) \geqq 0$ for all $a \in A$, to find a minimum capacity (s,t)-cut. A polyhedral characterization of (s,t)-cuts is the following.

(3.14) **Theorem.** *Let $D = (V, A)$ be a digraph, and s, t be different nodes of V. Let*

$$P_{CT}(D) := \text{conv}\{\chi^B \in \mathbb{R}^A \mid B \subseteq A \text{ contains an } (s,t)\text{-cut}\}, \text{ then}$$

$$P_{CT}(D) = \{x \in \mathbb{R}^A \mid (1)\ 0 \leqq x_a \leqq 1 \text{ for all } a \in A,$$
$$(2) \quad \sum_{a \in P} x_a \geqq 1 \text{ for all } (s,t)\text{-paths } P \subseteq A\}. \quad \square$$

Analogously to shortest paths there is no characterization of the convex hull of the incidence vectors of (s,t)-cuts of a diagraph known (the general cut problem, i.e. c not restricted to nonnegative vectors, is also $\mathcal{NP}$-complete).

The striking similarity of Theorems (3.14) and (3.13) is not just a coinci-

dence. The (s,t)-paths and the (s,t)-cuts form what is called a *blocking pair*. This is a polarity relation, introduced by Fulkerson, which has found many nice applications in combinatorial optimization. The blocking theory has revealed many interesting connections between problems which were formerly considered quite unrelated. We recommend Fulkerson (1971) for a survey of this theory and the related theory of antiblocking polyhedra.

The (s,t)-cuts are important because of their relation to flows. An (s,t)-*flow* in a digraph $D=(V,A)$ is a vector $x \in \mathbb{R}^A$ satisfying

$$x_a \geqq 0 \qquad \text{for all } a \in A$$
$$\sum_{a \in \delta^-(v)} x_a = \sum_{a \in \delta^+(v)} x_a \quad \text{for all } v \in V \setminus \{s,t\}.$$

The value of an (s,t)-flow x is the net amount of flow leaving s, i. e. this value is equal to

$$\sum_{a \in \delta^+(s)} x_a - \sum_{a \in \delta^-(s)} x_a$$

which is clearly equal to $\sum_{a \in \delta^-(t)} x_a - \sum_{a \in \delta^+(t)} x_a$. We say that an (s,t)-flow is subject to capacity $c: A \to \mathbb{R}_+$ if $x_a \leqq c_a$ for all $a \in A$. One of the most celebrated theorems in combinatorial optimization is the following one due to Ford & Fulkerson (1956) and Elias, Feinstein & Shannon (1956).

(3.15) **Max-Flow Min-Cut Theorem.** *Let $D=(V,A)$ be a directed graph, let $s,t \in V$, $s \neq t$, and let $c: A \to \mathbb{R}_+$ be a capacity function. Then the maximum value of an (s,t)-flow subject to the capacity c is equal to the minimum capacity of an (s,t)-cut. If all capacities c_a, $a \in A$, are integer then there exists an integer optimum flow.* □

Theorem (3.15) in particular implies that for a digraph $D=(V,A)$, two nodes $s,t \in V$, $s \neq t$, and capacities c_a, $a \in A$ the linear program (called *network flow problem*).

$$\max \sum_{a \in \delta^+(s)} x_a - \sum_{a \in \delta^-(s)} x_a$$
$$\sum_{a \in \delta^+(v)} x_a - \sum_{a \in \delta^-(v)} x_a = 0 \text{ for all } v \in V \setminus \{s,t\}$$
$$0 \leqq x_a \leqq c_a \text{ for all } a \in A$$

has an integral optimum solution whenever all the capacities c_a are integral.

Ford & Fulkerson (1956) gave an algorithm to compute a maximum (s,t)-flow and a minimum (s,t)-cut which is based on the min-max relation of

(3.15). Edmonds & Karp (1972) showed that this algorithm, with some modifications, is polynomial. This algorithm is one of the combinatorial algorithms which is most frequently used in practice. The reason is that quite a large number of real world problems can be formulated as network-flow problems.

Hard Combinatorial Optimi tion Problems

We now turn our attention to combinatorial optimization poblems which are $\mathcal{NP}$-complete. With each of these problems we can associate a class of polyhedra as described in section 3.1. For instance, consider the travelling salesman problem (TSP) (1.1). With each instance of (1.1) we associate the complete graph $K_n=(V,E)$ with $V=\{1,2,\ldots,n\}$. Each tour is a hamiltonian cycle of K_n. Thus the vertices of the (symmetric) *travelling salesman polytope*

$$Q_T^n := \text{conv}\{\chi^T \in \mathbb{R}^E \mid T \subseteq E \text{ is the edge set of a hamiltonian cycle}\}$$

correspond to the feasible solutions of the n-city problem, and each instance of the TSP can be solved — in principle — via the LP

$$\min\ c^T x,\ x \in Q_T^n$$

where $c \in \mathbb{R}^E$ is a vector describing the distances between the cities.

Enormous research efforts have gone into describing the polytopes associated with hard problems. Up to date no single example of a hard combinatorial optimization problem could be found for which an explicit linear characterization of the associated class of polytopes could be determined. In retrospect, this is no surprise since complexity theory provides good reasons to believe that for $\mathcal{NP}$-complete problems no descriptions of the type discussed in the foregoing subsections for easy problems can ever be obtained, see Karp & Papadimitriou (1982) und Papadimitriou (1984) for precise versions of this statement.

The research effort spent on this type of investigations, however, was not in vain. For many of the practically relevant problems it was possible to determine large classes of facets. These classes of facets could be incorporated into cutting plane algorithms (to be described in Section 3.5) which for quite a number of problems seem to be the practically most efficient methods available at present.

Moreover, it was also possible to use this polyhedral information to determine further special cases of hard problems for which polynomial time algorithms exist. The number of results in this area is so vast that it is impossible to survey the main results and give proper credit. To give at least some examples I would like to mention two of m favourite problems.

The travelling salesman polytope Q_T^n is one of the best studied polytopes. Grötschel & Padberg (1984) have collected all the known results on Q_T^n and

described in Padberg & Grötschel (1984) their algorithmic uses. Parts of these results can be summarized in the following.

(3.16) **Theorem.** *Let $n \geqq 6$. Then the travelling salesman polytope Q_T^n is contained in the polytope defined by the following system of equations and inequalities:*

(1) $0 \leqq x_e \leqq 1$ *for all* $e \in E$

(2) $\sum_{e \in \delta(v)} x_e = 2$ *for all* $v \in V$

(3) *subtour elimination constraints*

$$\sum_{e \in E(W)} x_e \leqq |W| - 1 \text{ for all } W \subseteq V,\ 3 \leqq |W| \leqq n-3 \text{ and } 1 \in W.$$

(4) *comb constraints*

$$\sum_{e \in E(H)} x_e + \sum_{i=1}^{k} \sum_{e \in E(T_i)} x_e \leqq |H| + \sum_{i=1}^{k} (|T_i| - 1) - \frac{k+1}{2}$$

(5) *clique tree inequalities*

$$\sum_{j=1}^{s} \sum_{E(H_j)} x_e + \sum_{i=1}^{t} \sum_{e \in E(T_i)} \leqq \sum_{j=1}^{s} |H_i| + \sum_{i=1}^{t} (|T_i| - t_i) - \frac{t+1}{2}.$$

(In (4) a comb consists of a node set H (called handle) and node sets $T_1, \ldots, T_k$ (called teeth) such that

i) $|H \cap T_i| \geqq 1$ for $i = 1, \ldots, k$,
ii) $|T_i \setminus H| \geqq 1$ for $i = 1, \ldots, k$,
iii) $T_i \cap T_j = \emptyset$ $1 \leqq i < j \leqq k$,
iv) $k \geqq 3$ and k odd,
v) $1 \in H$.

In (5) a clique tree consists of a set of node sets $H_1, \ldots, H_s \subseteq V$ (called handles) and a set of node sets $T_1, \ldots, T_t$ (called teeth) satisfying

i) $T_i \cap T_j = \emptyset$ $1 \leqq i < j \leqq t$,
ii) $H_i \cap H_j = \emptyset$ $1 \leqq i < j \leqq s$, and $s \geqq 2$,
iii) for each $i \in \{1, 2, \ldots, t\}$, $2 \leqq |T_i| \leqq n-2$ and some $v \in T_i$ belongs to no H_j for $j = 1, \ldots, s$,
iv) for each $j \in \{1, 2, \ldots, s\}$ the number of T_i having nonempty intersection with H_j is odd and at least three,
v) for $i \in \{1, 2, \ldots, t\}$ and $j \in \{1, 2, \ldots, s\}$, if $H_j \cap T_i \neq \emptyset$, then $H_j \cap T_i$ is an articulation set of the subgraph C of K_n with node set $\bigcup_{j=1}^{s} H_j \cup \bigcup_{i=1}^{t} T_i$ and edge set $\bigcup_{j=1}^{s} E(H_j) \cup \bigcup_{i=1}^{t} E(T_i)$, moreover C is connected.

And where t_i is the number of handles tooth T_i intersects.

Moreover, each of the inequalities of the system (1), (3), (4) and (5) defines a facet of Q_T^n, and no two of these inequalities are equivalent with respect to Q_T^n. The dimension of Q_T^n is $|E|-n$, and (2) is a minimal equation system for the affine hull of Q_T^n. □

It is obvious that the number of facets of Q_T^n described in Theorem (3.16) is incredibly large. Nevertheless these are by far not all of the facets of Q_T^n.

The second example I would like to mention is the ($\mathcal{NP}$-complete) *acyclic subdigraph problem.* An instance of this problem can be described as follows. Given a digraph $D=(V,A)$ with arc weights $c_a \in \mathbb{R}$ for all $a \in A$. Find an arc set $B \subseteq A$ which contains no directed cycle (i. e. B is an *acyclic arc set*) such that $c(B)$ is as large as possible. The associated polyhedron is

$$P_{AC}(D) := \text{conv}\,\{\chi^B \in \mathbb{R}^A \mid B \subseteq A \text{ is acyclic}\}.$$

The following has been proved in Grötschel, Jünger & Reinelt (1982).

(3.17) **Theorem.** *Let $D=(V,A)$ be a digraph. Then each of the following inequalities defines a facet of $P_{AC}(D)$. No two of these inequalities are equivalent with respect to $P_{AC}(D)$.*

(1) $x_{ij} \geqq 0$ *for all* $(i,j) \in A$,

(2) $x_{ij} \leqq 1$ *for all* $(i,j) \in A$ *with* $(j,i) \notin A$,

(3) $\sum_{(i,j) \in C} x_{ij} \leqq |C|-1$ *for all directed cycles* $C \subseteq A$,

(4) k-fence inequalities

$$\sum_{e \in F} x_e \leqq |F|-k+1 \qquad \text{for all } k\text{-fences } F \subseteq A,$$

(5) Möbius ladder inequalities

$$\sum_{e \in M} x_e \leqq |M| - \frac{k+1}{2} \qquad \text{for all Möbius ladders } M \subseteq A,$$

In (4) a subdigraph $(V(F),F)$ of D is a ***simple k-fence*** *if $V(F)$ consists of two disjoint node sets $U=\{u_1,\ldots,u_k\}$, $W=\{w_1,\ldots,w_k\}$ of cardinality k and F consists of all arcs (u_i,w_i), $i=1,\ldots,k$ and all arcs (w_j,u_i) $i,j=1,\ldots,k, i \neq j$, and where a k-fence is a digraph which can be obtained from a simple k-fence by repeated subdivision of arcs.*

In (5) a Möbius ladder is defined as follows. Let $C_1,C_2,\ldots,C_k$ be a sequence of different dicycles in a digraph $D=(V,A)$ such that the following holds:

(i) $k \geqq 3$ *and k is odd.*

(ii) C_i *and* C_{i+1} $(i=1,\ldots,k-1)$ *have a directed path* P_i *in common,* C_1 *and* C_k *have a dipath* P_k *in common.*

(iii) *Given any dicycle* C_j, $j \in \{1,\ldots,k\}$, *set*
$J = \{1,\ldots,k\} \cap (\{j-2, j-4, j-6, \ldots\} \cup \{j+1, j+3, j+5, \ldots\})$.
Then every set $\left(\bigcup_{i=1}^{k} C_i\right) \setminus \{e_i \mid i \in J\}$ *contains exactly one dicycle (namely* C_j*), where* e_i, $i \in J$, *is any are contained in the dipath* P_i.

(iv) *The largest acyclic arc set in* $\bigcup_{i=1}^{k} C_i$ *has cardinality* $\left|\bigcup_{i=1}^{k} C_i\right| - \frac{k+1}{2}$.

Then we call the arc set $M = \bigcup_{i=1}^{k} C_i$ *a* ***Möbius-ladder.***

Let us set

$$P_C(D) := \{x \in \mathbb{R}^A \mid x \text{ satisfies (1), (2), (3)}\}.$$

The class of digraphs D with $P_C(D) = P_{AC}(D)$ is called *weakly acyclic*. In Section 3.4 we will show that programs over $P_C(D)$ can be solved in polynomial time. Thus, since for this class of digraphs $\max\{c^T x \mid x \in P_C(D)\}$ equals $\max\{c^T x \mid x \in P_{AC}(D)\}$ $(c \geqq 0)$, for weakly acyclic digraphs the acyclic subdigraph problem can be solved in polynomial time. This class, for instance, contains the planar digraphs as an (important) subclass. This indicates — and we shall formulate this in more detail later — that polyhedral results can be used to obtain good algorithms, in particular, for special cases of hard problems.

3.3 *Integer Linear Programming and Cutting Planes*

One of the main approaches in the fifties and sixties was to consider combinatorial optimization problems as integer linear programs and use the simplex method together with so-called cutting planes to solve these problems. The idea of this technique is the following.

Given an instance of a combinatorial optimization problem, find a matrix A and a vector b such that

$$(3.18)\quad \{x \in \mathbb{R}^n \mid Ax \leqq b,\ x \text{ integer}\}$$

corresponds to the feasible solutions of the instance; i. e., in the cases we consider, the set defined above should consist of all incidence vectors of feasible solutions.

For instance, if $G = (V, E)$ is a graph with edge weights $c_e \in \mathbb{R}$ for all $e \in E$ then the feasible solutions of the following integer linear program (cf. (3.5), (3.6)).

$$
\begin{aligned}
&\max\ c^T x \\
&\sum_{e \in \delta(v)} x_e \leqq 1 && \text{for all } v \in V \\
&x_e \geqq 0 && \text{for all } e \in E \\
&x_e \text{ integer} && \text{for all } e \in E
\end{aligned} \tag{3.19}
$$

are exactly the incidence vectors of the matchings of G. Similarly, if $K_n = (V, E)$ is a complete graph and $c_e \in \mathbb{R}$ for all $e \in E$, then (cf. (3.16)) the feasible solutions of

$$
\begin{aligned}
&\min\ c^T x \\
&\sum_{e \in \delta(v)} x_e = 2 && \text{for all } v \in V \\
&\sum_{e \in E(W)} x_e \leqq |W| - 1 && \text{for all } W \subseteq V,\ 3 \leqq |W| \leqq n-3 \\
&0 \leqq x_e \leqq 1 && \text{for all } e \in E \\
&x_e \text{ integer} && \text{for all } e \in E
\end{aligned} \tag{3.20}
$$

are the incidence vectors of the hamiltonian cycles in K_n.

In general, it is not too difficult to find such an integer linear programming formulation for any combinatorial optimization problem.

Gomory's Algorithm

The algorithmic aspect behind such formulations is the following. Let us remove the integrality stipulations from (3.18) and solve the linear program $\max\{c^T x \mid Ax \leqq b\}$ (with the simplex method). (We call this LP the *linear programming relaxation* of the combinatorial optimization problem.) In case the optimum solution x^* of the LP is integral, then x^* is an optimum solution of the integer linear program and thus an optimum solution of the combinatorial problem is found. If x^* is not integral one would like to cut off x^* from $\{x \in \mathbb{R}^n \mid Ax \leqq b\}$ by adding a further inequality (called *cutting plane*), say $a^T x \leqq a_0$, in such a way that

$$
\begin{aligned}
&x^* \notin \{x \in \mathbb{R}^n \mid Ax \leqq b,\ a^T x \leqq a_0\} \text{ and} \\
&\{x \in \mathbb{Z}^n \mid Ax \leqq b\} = \{x \in \mathbb{Z}^n \mid Ax \leqq b,\ a^T x \leqq a_0\}.
\end{aligned} \tag{3.21}
$$

Gomory (1958), (1960) has devised a very simple method with which such an inequality can be read from the simplex tableau corresponding to the optimum solution x^*. And moreover, Gomory proved that by adding a finite number of his type of cutting planes an optimum solution of the corresponding integer program can be found.

However, practical computational experience revealed that Gomory's algorithm is very inefficient, and to my knowledge, there is no commercial code for integer programming problems which uses these cutting planes. There have been a number of attempts in the late sixties to devise new types of cutting planes different from the ones Gomory proposed, but there have been no computationally significant improvements.

This failure was one of the reasons that led to the developments discussed in Section 3.2. Consider for instance Edmond's characterization of the matching polytope (3.6) and the integer LP (3.19). Edmonds proved that the only inequalities that have to be added to (3.19) to obtain the matching polytope are the inequalities (3) of (3.6). Gomory's result shows that after a finite number of applications of his procedure all these inequalities can be obtained, but — and that turned out to be the case in practice — zillions of redundant inequalities may have to be added as well.

This observation suggests that Gomory's approach is too general. It is probably more effective to concentrate on particular problems and to characterize the cutting planes necessary for those problems (i.e. to describe the facets of the convex hull of the incidence vectors of the feasible solutions) in order to use this special type of problem specific cutting planes in LP-based algorithms. This idea will be further explored in Sections 3.4 and 3.5.

The Closure of $\{x \mid Ax \leqq b\}$

Gomory's idea has been systematized and been brought into the form of a nice theorem by Chvátal (1973) and — in more generality — by Schrijver (1980). To describe this, let us define for a given rational (m,n)-matrix A and a rational vector b

(3.22) $P := \{x \in \mathbb{R}^n \mid Ax \leqq b\}$ and $P_I := \text{conv}\{x \in \mathbb{Z}^n \mid Ax \leqq b\}$.

Let a_i, $i = 1,\ldots,n$ denote the column vectors of A. If $\lambda \in \mathbb{R}^m$, $\lambda \geqq 0$ then the inequality

$$\sum_{i=1}^{n} (\lambda^T a_i)\, x_i \leqq \lambda^T b$$

is satisfied by all points in P, and clearly, every integral vector contained in P satisfies

(3.23) $$\sum_{i=1}^{n} [\lambda^T a_i]\, x_i \leqq [\lambda^T b]$$

where $[\alpha]$ denotes the largest integer not larger than α. Thus, (3.23) is a valid inequality for P_I. Let us denote by $cl(P)$ (closure of P) the set of vectors x in $\mathbb{R}^n$ satisfying $Ax \leqq b$ and all inequalities of type (3.23) derived from $Ax \leqq b$ in the

way described above. To shorten notation we write $cl^r(P)$ for $cl(cl(\ldots cl(P)\ldots))$ where the closure operation cl is applied r times iteratively.

(3.24) **Theorem.** *Let P and P_I be as defined above. Then $cl(P)$ is a polyhedron containing P_I. Moreover, there exists a number $r \in \mathbb{N}$ such that*

$$P_I = cl^r(P). \quad \square$$

The smallest integer r with $P_I = cl^r(P)$ is called the *Chvátal rank* of P.

Total Unimodularity

It has been observed in the fifties that some matrices A associated with combinatorial optimization problems have the property that $\{x \in \mathbb{R}^n \mid Ax \leqq b\}$ is a polyhedron with only integral vertices if and only if b is integral. This property is of particular importance, since it implies that if $b \in \mathbb{Z}^m$ the polyhedra P and P_I defined in (3.22) coincide, i.e. no cutting planes have to be added to obtain P_I from P.

Hoffman & Kruskal (1956) introduced the following class of matrices. A matrix A is called *totally unimodular* if the determinants of all its square submatrices are 0, 1 or -1, and they showed:

(3.25) **Theorem.** *Let A be an integral matrix. Then A is totally unimodular if and only if for every integral vector b the polyhedron $\{x \in \mathbb{R}^n \mid Ax \leqq b, x \geqq 0\}$ has integral vertices only.* $\square$

This in particular implies that for a totally unimodular matrix A and integral vectors c and b the linear program $\{\max c^T x \mid Ax \geqq b, x \geqq 0\}$ as well as its dual linear program $\min \{b^T y \mid A^T y \geqq c, y \geqq 0\}$ have integral optimum solutions, provided feasible solutions exist.

Prime examples of totally unimodular matrices are the node-edge incidence matrices of bipartite graphs (these are the matrices defined by the left hand sides of the inequalities (2) of (3.5) and the node-arc incidence matrices of digraphs. (The matrices of network flow problems, cf. (3.15), are submatrices of such matrices.) Total unimodularity of these matrices is easy to prove by induction. Thus, in particular, the important Theorems (3.5) and (3.15) can be considered as consequences of Theorem (3.25).

There are quite a number of other characterizations of totally unimodular matrices, see Schrijver (1984b) for a survey. But none of these is a good characterization in the sense that it allows to check in polynomial time whether a matrix is totally unimodular or not. This problem has recently been solved by Seymour (1980) in the following way.

Tutte (1965) proved that a matrix A is totally unimodular if and only if the matric matroid defined by A over the reals is regular (see Welsh (1976) for a definition). Seymour then showed that a matroid is regular if and only if it can be

constructed via three (rather simple) types of compositions starting from three types of matroids: forest matroids, duals of forest matroids and a particular 10-element matroid. Edmonds pointed out that this characterization can be used to decompose in polynomial time a given matric matroid into the three types of matroids or to show that the matroid is not regular, i.e. that the matrix is not totally unimodular.

Another interesting line of research building on the proof techniques developed for the treatment of the concepts defined above is the investigation of so-called *totally dual integral systems* (TDI-systems). These generalize totally unimodular systems in various ways. We do not go into details here and refer to the survey papers Edmonds & Giles (1984), Pulleyblank (1983) and the forthcoming book Schrijver (1984a).

3.4 *Separation Problems and the Ellipsoid Method*

We have already seen that cutting off points by hyperplanes is an old idea in combinatorial optimization which has led (among others) to the developments described in the foregoing sections. A new impetus came into the field through the ellipsoid method.

The ellipsoid method is an algorithm developed by Shor (1970), (1977) which has been considered for some time in nonlinear (in particular nondifferentiable) optimization. Khachiyan (1979) observed that the ellipsoid method can be modified (using some observations from linear algebra, number theory and complexity theory) in such a way that it yields a polynomial time method for the solution of linear programming problems. This result caused great and well-deserved excitement in the world of mathematical programming.

It turned out that the ellipsoid method has even more potential and that it is particularly suited for the design of polynomial time algorithms for combinatorial optimization problems. These observations have been made by Karp & Papadimitriou (1982), Padberg & Rao (1984) and Grötschel, Lovász & Schrijver (1981). We do not want to explain the ellipsoid method (this algorithm is only used as a proof technique and can be replaced by other algorithms like the new "simplex method" of Yamnitsky & Levin (1982)), but we would like to mention the most important consequences which have led to new fields of research.

To present the results correctly we would have to introduce quite a technical machinery. We want to avoid this and state the results in a slightly imprecise form, making, however, the essence of them clear. Let us introduce the following three problems:

(3.26) **Optimization Problem.** *Given a polytope $P \subseteq \mathbb{R}^n$ and a vector $c \in \mathbb{Q}^n$. Find a vector $x^* \in P$ maximizing $c^T x$ over P or prove that P is empty.*

(3.27) **Separation Problem.** *Given a polytope $P \subseteq \mathbb{R}^n$ and a vector $y \in \mathbb{Q}^n$. Decide whether $y \in P$, and if $y \notin P$, find a vector $d \in \mathbb{Q}^n$ such that $d^T y > d^T x$ for all $x \in P$ (i.e. find a hyperplane separating y from P):*

(3.28) **Membership Problem.** *Given a polytope $P \subseteq \mathbb{R}^n$ and a vector $y \in \mathbb{Q}^n$. Decide whether y belongs to P or not.*

Clearly, if one can solve the separation problem for P then one can solve the membership problem for P.

In order to speak about polynomial time algorithms for these problems we have to specify the input lengths. For the vectors c in (3.26) resp. y in (3.27), (3.28) this is just the length of their binary encoding (rationals r are encoded by encoding the numerator p and the denominator q of a coprime representation $r = \frac{p}{q}$.) An important consequence of the ellipsoid method is that we do not need to know all the facet defining inequalities or all the vertices of P explicitly to define the input length of P. It is sufficient to consider an upper bound on the maximum input length of a vertex or a facet defining inequality of P and to use this number as the input length. Thus, for the case of 0/1-polytopes $P \subseteq \mathbb{R}^n$, which we are particularly interested in, we can simply use the natural number n as input length of P, despite the fact that P may have a number of vertices and facets which is exponential in n (like the matching polytope (3.6) and the travelling salesman polytope (3.16)).

So, if we speak of a polynomial time algorithm to solve problem (3.26), (3.27) or (3.28) we mean that this algorithm runs in time polynomial in the input length of P (as defined above) and the input length of c resp. y. The following theorem is one of the most useful consequences of the ellipsoid method.

(3.29) **Theorem.** *(a) Let $P \subseteq \mathbb{R}^n$ be a polytope with rational vertices, then the optimization problem (3.26) for P can be solved in polynomial time if and only if the separation problem (3.27) for P can be solved in polynomial time.*

(b) Let $P \subseteq \mathbb{R}^n$ be a full-dimensional polytope with rational vertices for which an interior point is known in advance. Then the optimization problem (3.26) for P can be solved in polynomial time if and only if the membership problem (3.28) for P can be solved in polynomial time. □

Note that the condition necessary for the validity of (3.29) (b) is almost always fulfilled in the case of the 0/1-polytopes we consider. Most of these polytopes are full-dimensional and contain the zero vector and all unit vectors, so $\frac{1}{n+1}(1,\ldots,1)$ is an interior point of these polytopes. If the polytopes are not full-dimensional (like the travelling salesman polytope) then usually an equality representation of the affine hull is known and the polytope can be projected to make it full-dimensional (in a lower dimensional space).

Theorem (3.29) thus states that in order to be able to optimize over P in polynomial time it suffices to be able to decide in polynomial time whether a given point belongs to P or not. The latter problem looks much easier but — as (3.29) states — has the same degree of difficulty as the former.

We want to describe two applications of this theorem which also show how good algorithms for some combinatorial optimization problems can be used to design good algorithms for others.

Consider the problem of finding a minimum capacity (s,t)-cut in a digraph $D=(V,A)$ with capacities $c_a \geqq 0$ for all $a \in A$. We know that the shortest (s,t)-path problem can be solved in polynomial time, for instance with the method of Dijkstra (1959). By Theorem (3.14) we can find a minimum (s,t)-cut by solving the linear program

$$\begin{aligned} &\min c^T x \\ (3.30)\quad &(1)\ 0 \leqq x_a \leqq 1 \quad \text{for all } a \in A, \\ &(2)\ \sum_{a \in P} x_a \geqq 1 \quad \text{for all } (s,t)\text{-paths } P \subseteq A. \end{aligned}$$

To be able to solve (3.30) in polynomial time it suffices by Theorem (3.29) to solve the separation problem for the 0/1-polytope $P_{CT}(D)$ defined by the constraints (1), (2) of (3.30) in polynomial time.

So, given a vector $y \in \mathbb{Q}^n$ we have to check in polynomial time whether y satisfies these two systems of inequalities. To check (1) is trivial. If one of the components of y is smaller than 0 or larger than 1 we obtain a separating inequality $x_a \geqq 0$ or $x_a \leqq 1$. In order to check the inequality system (2) of (3.30) we may therefore assume that y satisfies (1). Now we consider the components y_a of y as "lengths" of the arcs $a \in A$, and we calculate a shortest (s,t)-path P^* in D with respect to the length vector $y=(y_a)$ (by Dijkstra's method in polynomial time). Now if $\sum_{a \in P^*} y_a \geqq 1$ then, since P^* is a shortest path, y satisfies all inequalities (2), otherwise the inequality $\sum_{a \in P^*} x_a \geqq 1$ separates y from the polytope $P_{CT}(D)$.

This shows how a polynomial time minimum capacity cut algorithm can be derived via Theorem (3.29) from a polynomial time shortest path algorithm.

In the second example we consider the acyclic subdigraph problem. By Theorem (3.17) we know that the convex hull $P_{AC}(D)$ of the incidence vectors of acyclic subdigraphs of a digraph $D=(V,A)$ satisfies

$$\begin{aligned} P_{AC}(D) \subseteq P_C(D) := \{x \in \mathbb{R}^n \mid (1)\ & 0 \leqq x_a \leqq 1 && \text{for all } a \in A \\ (2)\ & \sum_{a \in C} x_a \leqq |C|-1 && \text{for all directed cycles } C \subseteq A\}. \end{aligned}$$

Given a vector $y \in \mathbb{Q}^n$ we can easily check inequalities (1), and to check (2) we may assume that y satisfies (1). We now define new "lengths"

$$w_a := 1 - y_a \quad \text{for all } a \in A.$$

For each arc $(i,j) \in A$ we calculate a shortest (j,i)-path P_{ji} in D with respect to the length vector w. Clearly, for each arc $(i,j) \in A$, $C_{ij} := P_{ji} \cup \{(i,j)\}$ is the shortest directed cycle (with respect to w) in D containing (i,j). Let C^* be a shortest of these cycles C_{ij}, $(i,j) \in A$. Suppose $\sum_{a \in C^*} w_a \geqq 1$ then $\sum_{a \in C^*} y_a \leqq |C^*| - 1$ and thus — by our construction — y satisfies all directed cycle inequalities (2). If $\sum_{a \in C^*} w_a < 1$ then $\sum_{a \in C^*} y_a > |C^*| - 1$ and the inequality $\sum_{a \in C^*} x_a \leqq |C^*| - 1$ separates y from $P_C(D)$. Hence by using $|A|$ times a shortest path algorithm we can design a polynomial time separation algorithm for $P_C(D)$.

From Theorem (3.29) we can conclude that we can optimize over $P_C(D)$ in polynomial time. This shows that for the class of weakly acyclic digraphs the acyclic subdigraph problem can be solved in polynomial time.

Note that in both applications described above the number of facets and the number of vertices of the polytopes is exponential in the input size of D. Still one can optimize in polynomial time.

In various applications it is important to test membership or solve the separation problem. The design of such algorithms has been neglected for a long time. Now that the ellipsoid method has shown the polynomial time equivalence of these problems to the optimization problem (3.26), a considerable amount of research is spent on inventing truly good combinatorial separation algorithms which do not suffer from the numerical disadvantages of the ellipsoid method.

For instance, Padberg & Rao (1982) have shown how to solve the separation problem for the matching polytope (3.6), Cunningham (1984) has designed a good combinatorial algorithm to solve this problem for matroid polytopes (3.10). But there is still a lot to do. A major achievement would be the invention of a good combinatorial algorithm for the polymatroid separation problem. This would imply a good combinatorial algorithm for the minimization of submodular functions (3.12). The only known polynomial time method for this problem described in Grötschel, Lovász & Schrijver (1981) utilizes the ellipsoid method.

Generalizations of the ellipsoid method and further applications of this method to problems in number theory, geometry, and combinatorial optimization are described in the forthcoming book Grötschel, Lovász & Schrijver (1985).

We want to mention a further example. In an outstanding paper H. W. Lenstra jun. (1983) has shown that for fixed $n \in \mathbb{N}$, given a rational (m,n)-matrix A and a vector $b \in \mathbb{Q}^m$, one can check in polynomial time whether or not there is an integral vector x satisfying $Ax \leqq b$. This result implies that integer linear programming problems in fixed dimension can be solved in polynomial time. This result follows quite easily from a newly developed method to find a reduced basis of a lattice (see Lenstra, Lenstra & Lovász (1982)) and the ellipsoid method.

More generally, in fixed dimension one can even minimize a convex function over the integral points of a convex body in polynomial time (cf. Grötschel, Lovász & Schrijver (1985)).

3.5 *Cutting Plane Algorithms in Practice*

In the foregoing sections we have mainly concentrated on theoretical issues. Now we turn to computer implementations of the ideas described before. We have outlined Gomory's cutting plane algorithm and mentioned its practical failure. The ellipsoid method together with the investigations of special combinatorial polytopes has shed a new light on this subject and seems to indicate that good cutting plane algorithms might exist provided that the cutting planes are selected carefully. We shall describe now what this means in practice.

The approach works in the same way for $\mathcal{NP}$-complete problems and problems in $\mathcal{P}$. We also want to point out that this approach dates back to the fifties where Dantzig, Fulkerson & Johnson (1954) have used it for the solution of a large Travelling Salesman Problem. The methods described in this paper have been neglected for a long time and found a revival only recently. We outline the basic issues of this technique by means of the $\mathcal{NP}$-complete acyclic subdigraph problem.

Each instance of this problem is given by a digraph $D=(V,A)$ and arc weights c_a, $a \in A$. As described in Section 3.1 resp. 3.2 we associate with each instance of this problem a polytope $P_{AC}(D)$ and want to solve

$$(3.31)\quad \max\{c^T x \mid x \in P_{AC}(D)\}.$$

By Theorem (3.17) we know that $P_{AC}(D)$ is contained in the polytope defined by

$$(3.32)\quad \begin{aligned}
&(1)\ 0 \leqq x_a \leqq 1 && \text{for all } a \in A\\
&(2)\ \sum_{a\in C} x_a \leqq |C|-1 && \text{for all directed cycles } C \subseteq A\\
&(3)\ \sum_{a\in F} x_a \leqq |F|-k+1 && \text{for all } k\text{-fences } F \subseteq A\\
&(4)\ \sum_{a\in M} x_a \leqq |M|-\frac{k+1}{2} && \text{for all Möbius ladders } M \subseteq A
\end{aligned}$$

and that (almost) all of these inequalities define facets of $P_{AC}(D)$. Instead of solving (3.31) we try to optimize $c^T x$ over (3.32). Using the ellipsoid method we know that we can optimize over the polytope $P_C(D)$ defined by (1) and (2) of (3.32) in polynomial time. But the ellipsoid method is very inefficient in practice, so we replace it by (the nonpolynomial) simplex algorithm. Practical experience has shown that the simplex algorithm works extremely fast on the average. (Recent work of Borgwardt (1982) has established a theoretical explanation of this.) Moreover, numerical experiments indicate that one can indeed optimize over $P_C(D)$ in this way for quite large digraphs.

For this we have to use the separation algorithm for $P_C(D)$ described in Section 3.4. By choosing data structures carefully one can implement this separation algorithm (based on shortest path techniques) so that it runs in $0(|V(D)|^3)$ time. For large digraphs this is quite a lot, so one often does a preprocessing by running fast problem specific heuristics that try to find violated inequalities. The design of such heuristics is guided by a careful analysis of fractional solutions that come up during practical experience with such an algorithm. It turned out empirically that such heuristics can significantly speed up the actual performance.

Now it may happen that a solution x^* satisfies all inequalities (1), (2), i.e. x^* optimizes $c^T x$ over $P_C(D)$, but that it is not integral. Then we try to cut off x^* using the inequalities (3) and (4) of (3.32). These inequalities are best possible cutting planes, since they define facets of $P_{AC}(D)$, but we do not know any (nontrivial) algorithm that checks whether x^* satisfies the inequalities (3), (4) or not. (This situation usually occurs in $\mathcal{NP}$-hard problems. There are "recognizable classes" of facets and some, which are not "well-behaved".) In such a case, we again design heuristics, hoping that they will find some violated inequalities of type (3) or (4). It might happen (and in practice it usually does) that by iterating this cutting plane recognition procedure (using separation heuristics combined with exact separation algorithms) one ends up with an integral solution. In this case we have solved the acyclic subdigraph problem.

If, however, the last optimum solution x^* is not integral and no inequality of type (1),...,(4) can be found that is violated by x^*, then we go to our last resort: branch & bound, cf. (4.6). Practical experience shows that in most cases only a few branching steps are necessary to get to the optimum integral solution.

There are quite a number of further tactical issues involved. One has to decide how many cutting planes to add in each step in order to keep the LP small, whether inequalities which are nonbinding in the present optimum solution should be removed etc. Our practical experience with this type of algorithms shows that there is no general answer. Each problem has to be studied individually. But there is a good chance that such investigations result in practically quite efficient methods.

Computational experience with cutting plane methods for hard problems as described above is reported for instance in Crowder & Padberg (1980), Barahona & Maccioni (1982), Grötschel, Jünger & Reinelt (1983), Crowder, Johnson & Padberg (1983). Such an algorithm has been implemented for the polynomially solvable matching problem as well. A surprising outcome of the computational experiments reported in Grötschel & Holland (1984) is the empirically observed fact that this (theoretically nonpolynomial) algorithm is as fast as the best combinatorial matching algorithms. Thus, it seems that the type of cutting plane algorithms described above deserves further attention, even in the case of polynomially solvable problems.

4. Future Developments

It is impossible to survey — given bounded space — all flourishing branches of combinatorial optimization and to discuss all significant recent results. In Section 3 I have made an attempt to outline the developments in polyhedral combinatorics, the subject closest to my own research interest, and I have already pointed out various directions of future research in this area.

I will now present a (nonsystematic and probably unbalanced) collection of further topics and problems which I think have future potential from a theoretical or practical viewpoint and which are worth studying. I will give only few comments and quote only very few references in order to keep within my page limits. I am sure that my opinion is biased but I hope that some readers may find something of interest.

4.1 *Relations to other Branches of Mathematics*

In the first two sections (4.1) and (4.2) I would like to point out some of my views about possible general future developments in the relations of combinatorial optimization to other mathematical fields, and I will give a few remarkable examples.

(4.1) Integer Programming and Number Theory

It seems that integer programming and number theory (in particular the geometry of numbers) study the same objects, but from very different viewpoints and using quite unrelated methods. There should be a way to make the deep results collected in number theory in the last centuries profitable for integer programming. And vice versa, number theory might benefit from some of the concepts and algorithms developed in integer programming.

For instance, ways to compute the Smith or Hermite normal form of a matrix are known for decades, but only recently Kannan & Bachem (1979) found an algorithm — using some nice combinatorial "tricks" — that calculates these normal forms in polynomial time. On the other hand, H. W. Lenstra jun. (1983), as mentioned in Section 3.4, proved — using number theoretic arguments — that integer programming problems are solvable in polynomial time in fixed dimension. In Grötschel, Lovász & Schrijver (1984) a (still minor) attempt is made to build a bridge between these two disciplines. In particular, a number of algorithmic versions of various results known in the geometry of numbers are proved.

It seems to me that the algorithm of Lenstra, Lenstra & Lovász (1982) to find a reduced basis in a lattice might be a first good tie between these two areas (and algebra in addition). The algorithm was developed to derive certain results in polyhedral theory (in particular about polytopes associated with combinatorial optimization problems) from the ellipsoid method. It turned out that it can

also be utilized, cf. Lenstra et al. (1982), to factor polynomials over the rationals in polynomial time. Kannan and Lovász observed that one can derive from the basis reduction algorithm a method which, given an algebraic number and a bound on the degree of its minimal polynomial, computes the minimal polynomial in polynomial time. The basis reduction algorithm has recently been applied by H. te Riele and A. Odlyzko, see te Riele (1983), to disprove the long standing Mertens' Conjecture. Moreover, the basis reduction algorithm is currently being used by several people to break cryptosystems, see e. g. Adleman (1983) and Lagarias & Odlyzko (1983).

Cryptography, anyway, seems to be one of the reasons that has attracted number theoretists to study complexity questions. The exciting developments in prime testing and factoring of integers clearly show that nontrivial number theory is able to contribute substantially to a better understanding of very down-to-earth complexity or integer programming problems.

(4.2) Relations of Combinatorial Optimization to Other Mathematical Disciplines

I have already mentioned before various contacts and fruitful cooperations of the theory of combinatorial optimization with other branches of mathematics. There is no way to be complete, but I would like to mention a few more of these which I think are worth investigating.

Polyhedral combinatorics should be able to benefit significantly from the developments in *convex geometry*, in particular the (general) *theory of polyhedra*. Although the objects of study in these two areas are more or less the same there have been very few applications of the "general theory" (as for instance described in the book Grünbaum (1967)) to the study of concrete polyhedra (as described in Sections 3.1 and 3.2). Mainly due to the work of V. Klee and his collaborators the contacts get closer and more of the geometers get interested in "real-world polyhedra" like matching polytopes and travelling salesman polytopes. Maybe the new techniques developed in the theory of polyhedra will lead to a proof of the Hirsch conjecture, or at least to proofs of this conjecture for more classes of interesting combinatorial polyhedra (see for instance Klee & Kleinschmidt (1984)). These lines of research may also produce a polynomial time version of the simplex method, something of real practical interest.

A striking example of a fertile application of *commutative algebra* are the results of Stanley concerning the enumeration of faces of various dimensions of polytopes. Stanley extended and applied the theory of graded algebras and their Hilbert functions, and Cohen-Macauly rings to give (among others) tight upper bounds on the number of faces in each dimension in terms of the number of vertices. Subsequently, he combined these methods with some recent results in *algebraic geometry* to complete the proof of McMullen's conjectured characterization of the face-counting vectors of simplicial polytopes. An excellent account

of these developments is given in Billera (1983). Billera also describes the first attempts of an application of these methods to the study of integer solutions to systems of linear inequalities. I am sure that the power of these methods has not fully been recognized yet, and that we may expect further interesting results from this approach.

I also believe that the methods of *algebra* have not been exploited enough yet. Of course, algebra is so vast a field that in the concrete situation of a combinatorial optimization problem it is almost impossible to guess which of the algebraic techniques might fruitfully apply. Algebraic concepts certainly have influenced the field. There are matching polynomials, chromatic polynomials, chain groups and the like. But as Stanley's approach shows, in certain special cases the use of more sophisticated groups, rings etc. associated with a combinatorial object may result in deep new insights. This area of research is almost untouched.

In general, the question we address here is "What is a 'good' representation of a combinatorial optimization problem?". The survey in Section 3 shows how combinatorial optimization problems can be represented by means of polyhedra, and it also proves that this method has lead to theoretically and practically exciting new developments. But there is a host of further possibilities.

The paper Lovász (1982), for instance, describes two interesting further approaches. First Lovász discusses a method introduced by W. W. and S. R. Li to associate a certain polynomial with a graph. The ideas of this and the proof techniques come from *algebraic geometry*. Using Hilbert's Nullstellensatz it is possible to determine the degrees of these polynomials and to obtain estimations from these degrees for the chromatic number and the stability number of a graph. In the second approach *topology* is used. With each graph G a neighbourhood complex $\mathcal{N}(G)$ is associated whose connectivity gives a lower bound on the chromatic number. Both applications use nontrivial theory to obtain quite surprising connections between algebraical resp. topological invariants and combinatorial parameters. Applications of this kind, however, are quite sporadic. This is probably due to the fact that there are only very few people who know enough from either area to see nontrivial connections.

The relations of combinatorial optimization to other branches of mathematical programming are manifold. Nonlinear programming, stochastic programming, and integer programming borrow from and stimulate each other. Two significant contributions of nonlinear programming to combinatorial optimization are the development of the ellipsoid method (see Section 3.4) and its extension to a powerful tool in combinatorial optimization, and moreover, the technique of Lagrangean relaxation of (hard) combinatorial optimization problems together with the design of subgradient algorithms (incorporated in branch & bound schemes) for the solution of these Lagrangean relaxations (cf. Fisher (1981) and Geoffrion (1974) for surveys).

4.2 *General Areas of Future Research*

The next problem areas concern developments in combinatorial optimization which, I think, need further study and which may contribute significantly to a better theoretical understanding of combinatorial optimization or to a better use of the theory in practice.

(4.3) Generalizations of Min-Max Results

The presently best survey of min-max results in combinatorial optimization theory is Schrijver (1983). Schrijver has collected all results known to date, classified them with respect to area and generality and described their relation. I have already mentioned in Section 3.2 (for the special case of submodular functions) that major research projects are carried out to unify these results and find a general setting which allows a better understanding of the fact that certain combinatorial objects stand in a min-max relation to others (cf. König's Theorem in Section 3.2) while other quite similar ones do not.

Major contributions to this area are, for instance, the results of Mader (1978 a) (1978 b) on edge resp. vertex disjoint S-paths and Seymour's results on flows in matroids, cf. Seymour (1977) (1981).

Further progress in this min-max theory is highly desirable not only from a theoretical point of view. Min-max relations usually are good optimality criteria and therefore often form the backbone of polynomial time algorithms.

(4.4) Speed-Up and Lower Bounds for Easy Problems

Algorithmic research with respect to problems solvable in polynomial time mainly concentrates on the two subjects mentioned in the heading. There has been significant progress with respect to the first in the recent years, while almost no nontrivial results can be reported about the second.

There are two ways to get better algorithms for easy problems. Either one finds a new method with better time or space complexity or one modifies one of the existing algorithms in some way. The first case — of course — is rather rare. The research efforts on the second in the recent years were quite successful and have not only brought up a list of new "tricks" with which such speed-ups can be obtained, but also a general theory of algorithmic techniques and data handling procedures which provides a powerful tool for algorithm improvements. A very nice account of this theory can be found in Tarjan (1978). Gabow (1983) reports about the success of the scaling technique.

For example, the speed-up techniques were particularly successful with respect to calculating maximum flows in networks and shortest paths in digraphs. The original (nonpolynomial) Ford-Fulkerson algorithm for network flows has been modified and remodified in various ways and quite substantial

running time improvements were obtained, see Tarjan (1983a) for a survey of this. There are a number of competing shortest path algorithms each of which has been subject to various modifications. Recent improvements on shortest paths methods in planar graphs are reported in Frederickson (1983). A new variant of Dantzig's algorithm with good expected running time is presented in Bloniarz (1983).

A surprising example of speed improvement by a new method was the $0(n^{\log_2 7})$ matrix multiplication algorithm of Strassen (1969). Stimulated by this, there has been further progress in the meantime, see for instance Coppersmith & Winograd (1982) where a matrix multiplication algorithm with running time less than $0(n^{2.495548})$ is described. These authors give a speed-up theorem and by this they also show that there is no best matrix multiplication algorithm. A similarly striking case is the planarity algorithm of Hopcropft & Tarjan (1972) which proved that for a graph $G=(V,E)$ planarity can be tested in $0(|V|)$ time.

Whenever such improvements are obtained a question that arises is whether or not this new algorithm is best possible with respect to time (or space) complexity. This leads to the task of finding lower bounds for the computational complexity of a combinatorial optimization problem (with respect to some machine model, like RAM or Turing machines). So the question is, can one prove that for every instance of a combinatorial optimization problem of input length n at least $p(n)$ steps are necessary for its solution, where p: $\mathbb{N} \to \mathbb{N}$ is some function (e. g. a polynomial).

There are some trivial bounds. For instance if two (n,n)-matrices have to be multiplied then each entry of the two matrices has to be touched at least once. Thus, at least $2n^2$ steps are necessary to compute the product of two (n,n)-matrices. Similarly, for the determination of certain graph parameters all nodes or all edges have to be examined at least once. So one gets lower bounds $|V|$ or $|E|$ for the number of steps necessary to calculate this parameter, see Rivest & Vuillemin (1978).

It is somewhat astonishing that such lower bounds are often the only ones available, and for most easy combinatorial optimization problems the gap between the complexity of the best known algorithm and the best lower bound for its solution is considerable. A discussion of methods to establish lower bounds can be found in Weide (1977).

(4.5) Heuristics for Hard Problems

Most of the combinatorial optimization problems that come up in the real world are $\mathcal{NP}$-complete. But solutions have to be found, e. g. for the routing of garbage collection trucks or the layout of a computer chip, and mathematicians cannot hide behind an intractability proof and leave the problem to the imagination of economists or engineers. Practice demands the design of heuristic algorithms which produce a "good" solution of the problem.

Heuristic methods for hard problems are widely used by practitioners with (more or less) satisfactory success. For many years the judgement of the quality and effectiveness of heuristic methods was largely based on empirical computational experience. That is, some test runs on "representative" real world and some "representative" randomly generated problem instances were performed and the method which yielded the "best result on the average" was chosen to be used.

The recent years have seen an increasing mathematical interest in the performance analysis of heuristics. In particular, two new tools — "worst-case-analysis" and "average case analysis" — have been developed and provide reasonable means to judge the quality of a heuristic algorithm.

In *worst-case analysis* one tries to prove a performance guarantee for a heuristic, i.e. to show that for every instance of an optimization problem a certain algorithm produces a solution whose value differs from the optimum value by no more than, say, p per cent. Of course, one would like to have a very fast algorithm with best possible performance guarantee.

Results of this type are surveyed in Fisher (1980), Garey & Johnson (1979), Grötschel (1982) and Korte (1979). It turns out that hard problems differ very much with respect to their "approximability".

For instance, no performance guarantee for polynomial time heuristics can be given at all for the symmetric travelling salesman problem (1.1), unless $\mathscr{P} = \mathscr{NP}$. But if the intercity distances c_{ij} satisfy the triangle inequality $c_{ij} + c_{jk} \geqq c_{ik}$, $1 \leqq i < j < k \leqq n$, then the $0(n^3)$ algorithm of Christofides (1976) produces a tour which is at most 50% longer than the optimal tour. It is unknown whether a smaller bound can be achieved in polynomial time.

There are hard problems for which heuristics with provably best possible polynomial time performance guarantee exist. For example, Hochbaum & Shmoys (1984) describe a polynomial time heuristic for the k-center problem (with triangle inequality) which gives a solution whose value is at most twice as large as the optimum value, and they prove that the existence of a polynomial time approximation algorithm with better performance guarantee would imply $\mathscr{P} = \mathscr{NP}$.

Very few hard problems can be approximated up to any given accuracy. One such example is the knapsack problem. (Find a 0/1-vector maximizing $c^T x$ over all 0/1-vectors satisfying $a^T x \leqq b$.) For this problem Ibarra & Kim (1975) have designed a so-called *fully polynomial approximation scheme* which is an algorithm that, given an instance of the knapsack problem and a rational $\varepsilon > 0$, produces a solution S such that the error of the value $c(S)$ of S relative to the optimum value is at most ε, and which has a running time which is polynomial in the input length of the instance and ε^{-1}.

Striking progress has been made with respect to the bin packing problem (pack n items into as few bins as possible). The First Fit heuristic (Pick any item and put it into the first bin into which it fits!) was shown by Garey et al. (1976) to give no more than $\frac{17}{10}$th of the optimum number of bins, D. Johnson (1973)

proved that a variant of this, First Fit Decreasing, produces $\frac{11}{9}$th of the optimum value, Yao (1980) improved this to $\frac{11}{9} - \varepsilon$ for some very small $\varepsilon > 0$. Finally Fernandez de la Vega & Lueker (1981) gave a linear time algorithm — based on a linear programming relaxation — which for any fixed $\varepsilon > 0$ gives no more than $(1 + \varepsilon)$ times the optimum number of bins. But the running time of this algorithm increases very quickly with ε getting small. Karmarkar & Karp (1982) considered the dual of the Fernandes de la Vega & Lueker linear program and applied the ellipsoid method to obtain a fully polynomial approximation scheme, see Coffman, Garey & Johnson (1983) for a survey.

For many hard combinatorial optimization problems no polynomial time heuristic algorithms with good (or any) performance guarantee are known. For example, for the acyclic subdigraph problem, see Section 3.2, a trivial algorithm gives a 50% relative error (Take any linear ordering of the nodes of $D = (V, A)$, let B be the arcs of D which are consistent with this ordering and let $B' := A \setminus B$. Then clearly B and B' are acyclic and the value of B or B' is at least 1/2 of the optimum value!), but nothing better is available. For the asymmetric travelling salesman problem (with triangle inequality) no constant bound is known. All known performance guarantees depend on the number of cities.

For problems, like the TSP, where no performance guarantee can be given at all, worst-case analysis is not an appropriate tool to judge a heuristic algorithm. More promising in such (but not only in these) cases is a *probabilistic analysis* of the performance of a heuristic. The idea here is the following. Given a combinatorial optimization problem, then with each $n \in \mathbb{N}$ a probability distribution over the instances of input size n is associated. Then one tries to prove that with probability tending to 1 (with growing input size) a certain heuristic produces a solution whose value is ε-close to the optimum solution.

Algorithms with such a good *average-case* behaviour have been designed for various hard combinatorial optimization problems, (e. g. the TSP and the acyclic subdigraph problem), but this area is still in its infancy. Most of the algorithms that have been analyzed are extremely simple (and it has empirically been observed that they perform rather poorly (within the practically relevant problem sizes) compared with other widely used heuristics). It seems, however, to be a very difficult problem to make the stochastic machinery work for more sophisticated algorithms.

Surveys of this approach are Karp (1976), Lueker (1979) and Weide (1980). Two interesting papers, for example, are Halton & Terada (1982) and Burkard & Finke (1984). An annotated bibliography of the probabilistic analysis of algorithms is Karp (1984).

(4.6) Exact Optimization Algorithms for Hard Problems

In many real world situations it is necessary (or desirable or profitable) to know the true optimum solution of a problem instance and not only a "good"

solution. Thus, algorithms have to be designed for hard problems which empirically show good running time performance. It would of course be even better to be able to prove that such algorithms run fast on the average (within some probability model), but to my knowledge no result of this type has been obtained so far for $\mathcal{NP}$-complete problems.

There are two principle methods available for solving hard problems exactly. One is the cutting plane technique described in Sections 3.3 and 3.5. The other is the branch & bound method.

The guiding idea of the branch & bound technique is to enumerate all feasible solutions in an "intelligent manner". The enumeration is organized in such a way that at every step the universe of feasible solutions is partitioned into disjoint subsets and that at every step lower and/or upper bounds for the value of the best solution within these subsets are computed. If (in case of a maximization problem) this upper bound for a certain subset is smaller than the present best known feasible solution or the best known lower bound, all solutions contained in this subset can be omitted from further considerations. Otherwise the subset is split into smaller pieces to obtain a finer partition of the set of all feasible solutions, and the procedure is continued.

It is apparent that the relative success of a branch & bound method heavily depends on the partitioning strategy and the quality of the bounds that are computed. The determination of bounds is the most important feature of such algorithms and a large part of the research effort in the last years has gone into finding methods for computing good bounds with reasonable computational effort.

It is impossible to compare these two approaches in general with respect to their quality. Historically the cutting plane methods came first. They were then superseded in the sixties by the branch & bound algorithms. In particular the technique of Lagrangean relaxation developed in the early seventies considerably improved the performance of these algorithms. Now there is a revival of cutting plane methods as described in Section 3.5. But still all such quality judgements are extremely problem specific. An approach, working well for some problem, might fail in another. This also shows that we do not know enough about the "character" of hard problems.

The current trend is to combine the two methods and enrich them with various heuristic features. These approaches and their combination are described in Grötschel (1982). Most of these techniques have in fact been developed in order to solve travelling salesman problems. This problem seems to have become a standard hard problem for which everybody tries to show the success of his new ideas. The branch & bound algorithms existing for the travelling salesman problem are surveyed in Balas & Toth (1983) and the cutting plane methods for this problem in Padberg & Grötschel (1984).

(4.7) Adaptation to Technical Progress

Without the rapid development of computers the explosive growth of combinatorial optimization in the last thirty years is unimaginable. It is absolutely hopeless to try to solve 15-city travelling salesman problems by hand. However, even the biggest computer cannot handle 30-city problems using brute force enumeration only. But the joint progress in mathematics, algorithm design techniques and computer hardware makes it possible to solve 150-city problems routinely, and even problems with more than 300 cities have been solved, cf. Crowder & Padberg (1980).

The last years brought two new technical developments: microcomputers and parallel computers. Both these two technologies have already influenced the research in combinatorial optimization.

With respect to microcomputers an attempt is made to design algorithms for combinatorial optimization problems which are particularly suited for these types of machines, especially to give small companies which have no access to large computing facilities the possibility to benefit from the results obtained in mathematical programming. The public interest in these developments is, for instance, shown by the growing number of technical sessions on this subject at meetings of the Operations Research Society of America. I have, however, the feeling that due to further technical progress this problem area will disappear within the next ten years, say. I believe that in a few years we shall have cheap desk calculators with 10 Megabyte or more central memory, and so there will be no need any more for special purpose algorithms for computers with small central or peripheral memory.

More significant is parallelization. Most of the existing algorithms cannot be parallelized, and so really new methods have to be developed to exploit the power of these new computers. Very interesting progress has been made in this area in the recent years. It is impossible to survey all the theoretical and practical aspects of parallelism here. We recommend the very up-to-date annotated bibliography Kindervater & Lenstra (1984) on this subject, and the $\mathcal{NP}$-completeness columns seven and eight by D. S. Johnson (1983) in the Journal of Algorithms.

4.3 *Some Concrete Open Problems*

Before, I have outlined general developments in combinatorial optimization which I expect or hope for. Now I would like to mention a few concrete open problems which I am interested in. Most of the problems are probably nontrivial. I do not consider this problem list representative for the field. The problems reflect my own research interests.

The first group of problems could be called "*combinatorialization of ellipsoidal results*". What I mean by this is the following. A host of combinatorial optimization problems was shown to be solvable in polynomial time using the ellipsoid method, cf. Section 3.4. The ellipsoid method is — for various reasons

— not a really good algorithm from the practical point of view. But for quite a number of combinatorial problems these ellipsoidal algorithms are the only polynomial ones known to date. Thus, I would like to see good combinatorial algorithms for these problems. The following problems are of particular interest.

(4.8) **Problem.** *Find a polynomial time combinatorial algorithm to solve the weighted stable set, clique, clique covering and colouring problem on perfect graphs.*

There has been some progress recently by the Grenoble group (Burlet, Fonlupt, Uhry et al.) who gave good combinatorial algorithms for these problems for large classes of perfect graphs, but the general case seems to be hard. A side remark on perfect graphs! Maybe a proof of

(4.9) **The Perfect Graph Conjecture.** *A graph is perfect if and only if it contains neither an odd chordless cycle of length at least five nor the complement of such an odd chordless cycle as an induced subgraph.*

could shed new light on the structure of perfect graphs and provide the tools for a good algorithm, see the books Golumbic (1980) and Berge & Chvátal (1984) for perfect graphs and conjecture (4.9).

The second problem of this ellipsoidal group is:

(4.10) **Problem.** *Find a polynomial time combinatorial algorithm to minimize submodular functions.*

Problem (4.10) has already been mentioned in Section (3.2), see Theorem (3.12). Such an algorithm could be derived from a positive solution of

(4.11) **Problem.** (Polymatroid Separation Problem). *Given a submodular function* $f: 2^E \to \mathbb{Q}$ *satisfying* $f(\emptyset) = 0$ *and* $S \subseteq T \subseteq E \Rightarrow f(S) \leqq f(T)$. *Set*

$$P_f := \{x \in \mathbb{R}^E \mid \sum_{e \in F} x_e \leqq f(F) \quad \text{for all } F \subseteq E$$
$$x_e \geqq 0 \qquad \text{for all } e \in E\}.$$

Find a combinatorial polynomial time separation algorithm for P_f. *(In fact it suffices to find an algorithm which checks whether the point* $(1,1,\ldots,1) \in \mathbb{R}^E$ *is contained in* P_f.*)*

Cunningham (1984) has a good algorithm (based on network flow techniques) which solves (4.10) in case f satisfies in addition $f(S) \leqq |S|$ for all $S \subseteq E$.

In order to get efficient cutting plane algorithms for hard combinatorial optimization problems it is necessary to have polynomial time separation algorithms for large classes of facet defining inequalities of the associated polyhedra, see Sections 3.1, 3.2, 3.4 and 3.5. With respect to the travelling salesman polytope Q_T^n the following problem is unsolved, cf. Theorem (3.16).

(4.12) **Problem.** *Find polynomial time algorithms which check whether a given point $y \in \mathbb{Q}^E$ satisfies the comb inequalities (3.16) (4) resp. the clique tree inequalities (3.16) (5) and if not provide a violated inequality of this type.*

Padberg & Rao (1982) can handle a special case of the comb inequalities (the so-called 2-matching inequalities), but not more is known. Similarly for the acyclic subdigraph problem we have (cf. Theorem (3.17)).

(4.13) **Problem.** *Find polynomial time algorithms that check whether a given point $y \in \mathbb{Q}^A$ satisfies all k-fence inequalities (3.17) (3) resp. all Möbius ladder inequalities (3.17) (4) and if not yield a violated inequality of this type.*

It follows from results of Grötschel, Lovász & Schrijver (1981) that the facets of polyhedra associated with easy problems are algorithmically well-characterized in the following sense.

(4.14) **Theorem.** *If $P_\mathcal{I} \subseteq \mathbb{R}^n$ is a polytope associated with a combinatorial optimization problem (as described in 3.1) which is solvable in polynomial time and if $c^T x$ is an objective function such that $\gamma = \max\{c^T x \mid x \in P_\mathcal{I}\}$, then one can find in polynomial time n facet defining inequalities $a_i^T x \leqq \alpha_i$ and nonnegative rationals λ_i, $i = 1, \ldots, n$ such that*

$$c = \lambda_1 a_1 + \ldots + \lambda_n a_n \text{ and } \gamma = \lambda_1 \alpha_1 + \ldots + \lambda_n \alpha_n. \quad \square$$

Theorem (4.14) shows that in some way one can "get his hand on the facets" of $P_\mathcal{I}$, which suggests, that it should be possible to find an explicit linear characterization of $P_\mathcal{I}$. However, there are some problems whose associated polytopes have resisted all characterization attacks so far.

A graph is called *claw-free* if it does not contain the graph $K_{1,3}$ as an induced subgraph.

(4.15) **Problem.** *Find a complete linear characterization for the convex hull of the incidence vectors of stable sets in claw-free graphs.*

Minty (1980) and Sbihi (1978) have shown that maximum stable sets in claw-free graphs can be found in polynomial time. Giles & Trotter (1981) discovered some "wild" facets of stable set polytopes of claw free graphs, and Chvátal (unpublished) showed that the Chvátal rank of this class of polyhedra is unbounded. So this is probably a tough problem.

For graphs with positive edge weights it is easy to find not only shortest paths but also shortest paths and cycles of even or odd lengths using matching techniques or modifications of Dijkstra's method. Similary, in digraphs shortest odd dicycles are easy to compute. The polytope $P_S(D)$ of shortest (s,t)-paths in a digraph D has the nice description given in Theorem (3.13). Nothing similar is known if a parity condition is added.

Let $G=(V,E)$ be a graph, $D=(V,A)$ be a digraph and let $s,t \in V$ be different nodes. Define

$$Q_1(G) := \text{conv}\{\chi^P \in \mathbb{R}^E \mid P \subseteq E \text{ contains an } (s,t)\text{-path of even length}\}$$
$$Q_2(G) := \text{conv}\{\chi^P \in \mathbb{R}^E \mid P \subseteq E \text{ contains an } (s,t)\text{-path of odd length}\}$$
$$Q_3(G) := \text{conv}\{\chi^C \in \mathbb{R}^E \mid C \subseteq E \text{ contains an even cycle}\}$$
$$Q_4(G) := \text{conv}\{\chi^C \in \mathbb{R}^E \mid C \subseteq E \text{ contains an odd cycle}\}$$

and let $P_1(D),\ldots,P_4(D)$ be defined analogously (replacing (s,t)-path by directed (s,t)-path and cycle by directed cycle).

(4.16) **Problem.** *Find complete linear characterizations of the polytopes* $Q_1(G),\ldots,Q_4(G)$ *and* $P_1(G),\ldots,P_4(G)$. □

Finally I would like to repeat a very interesting and difficult problem mentioned in Section 3.2, whose solution probably also depends on a good characterization of a certain polyhedron.

(4.17) **Problem.** *Design a polynomial time algorithm for the weighted matroid matching problem for matric matroids.*

4.4 *Further Information and Conclusion*

The number of books on combinatorial optimization (and related areas) is not too large. A short list of relatively new books — each with a different emphasis — is: Christofides (1975), Garey & Johnson (1979), Garfinkel & Nemhauser (1972), Golumbic (1982), Grötschel, Lovász & Schrijver (1985), Lawler (1976), Lawler, Lenstra & Rinnooy Kan (1984), Lovász (1979), Lovász & Plummer (1984), Papadimitriou & Steiglitz (1982), Recski (1985), Schrijver (1984a), Tarjan (1983). Soon the book O'hEigeartaigh, Lenstra & Rinnooy Kan (1984) will appear that contains annotated bibliographies on various branches of combinatorial optimization. In these books more detailed information can be found about the concepts and problems whose developments have been surveyed in this paper.

In the thirty years of its existence combinatorial optimization has developed into a fertile and rapidly expanding field. It has many relations to other mathematical disciplines (I hope I could point this out), manifold nontrivial applications to areas like Physics, Management, Economics, Engineering etc. Fortunately, combinatorial optimization has not split into a pure, a computational, and an applied branch (yet), and a large part of the attraction of this field (at least to me) originates from the fact that still many new problems come into the field from (quite varying) issues of the real world which cannot be solved routinely.

References

Adleman L. M. (1983), "*On Breaking Generalized Knapsack Public Key Crypto-Systems*", Proceedings of the 15th Annual ACM Symposium on Theory of Computing,ACM, 1983, 402—412.

Bachem A., M. Grötschel & B. Korte (eds.) (1983), "*Mathematical Programming — The State of the Art, Bonn* 1982", Springer Verlag, Heidelberg, 1983.

Baker T., J. Gill & R. Solovay (1975), "*Relativizations of the P =ωNP Question*", SIAM Journal on Computing **4** (1975) 431—442.

Balas E. & M. W. Padberg (1975), "*Set partitioning*" in: B. Roy (ed.), "*Combinatorial Programming: Methods and Applications*", Reidel, Dordrecht, 1975, 205—258.

Balas E. & P. Toth (1983), "*Branch and Bound Methods for the Traveling Salesman Problem*", Management Science, Research Report No. MSRR **488**, Carnegie-Mellon University, Pittsburgh, March 1983.

Barahona F. & E. Maccioni (1982), "*On the Exact Ground States of Three-Dimensional Ising Spin Glasses*", Journal of Physics A: Math. Gen. **15** (1982) L611—L615.

Berge C. & V. Chvátal (1984), "*Topics in Perfect Graphs*", North-Holland, Amsterdam, 1984, to appear.

Billera L. (1983), "*Polyhedral Theory and Commutative Algebra*" in: A. Bachem, M. Grötschel & B. Korte (eds.), "*Mathematical Programming — The State of the Art, Bonn* 1982", Springer Verlag, Heidelberg, 1983, 57—77.

Birkhoff G. (1946), "*Tres observaciones sobre el algebra lineal*", Rev. Univ. Nac. Tucuman, Ser. A,**5** (1946) 147—148.

Bloniarz P. A. (1983), "*A Shortest-Path Algorithm with Expected Time 0 ($n^2 \log n \log^* n$)*", SIAM Journal on Computing **12** (1983) 588—600.

Borgwardt K.-H. (1982), "*The average number of pivot steps required by the simplex-method is polynomial*", Zeitschrift für Operations Research **26** (1982) 157—177.

Burkard R. E. & U. Finke (1984), "*Probabilistic Asymptotic Properties of Some Combinatorial Optimization Problems*", Discrete Appl. Math. (1984) to appear.

Christofides N. (1975), "*Graph Theory, an Algorithmic Approach*", Academic Press, New York, 1975.

Christofides N. (1976), "*Worst-Case Analysis of a New Heuristic for the Travelling Salesman Problem*", Tech. Report, Grad. School Industrial Administration, Carnegie-Mellon University. Abstract in "*Algorithms and Complexity — New Directions and Recent Results*", J. F. Traub (ed.), (1976) 441. Academic Press, New York.

Chvátal V. (1973), "*Edmonds Polytopes and a Hierarchy of Combinatorial Problems*", Discrete Mathematics **4** (1973) 305—337.

Cobham A. (1965), "*The Intrinsic Computational Difficulty of Functions*", Proc. 1964 International Congress for Logic, Methodology and Philosophy of Science, Y. Bar-Hillel (ed.), North-Holland, 1965, 24—30.

Coffman E. G., M. R. Garey & D. S. Johnson (1983), "*Approximation Algorithms for Bin-Packing — An Updated Survey*", Preprint, Bell Laboratories, Murray Hill, 1983.

Cook S. A. (1971), "*The Complexity of Theorem-Proving Procedures*", Proc. ACM Symp. Theory of Computing **3** (1971) 151—158.

Coppersmith D. & S. Winograd (1982), "*On the Asymptotic Complexity of Matrix Multiplication*", SIAM Journal on Computing **11** (1982) 472—492.

Crowder H., E. L. Johnson & M. W. Padberg (1983), "*Solving Large-Scale Zero-One Linear Programming Problems*", Operations Research **31** (1983) 803—834.

Crowder H. P. & M. W. Padberg (1980), "*Solving Large-Scale Symmetric Traveling Salesman Problems to Optimality*", Management Science **26** (1980) 495—509.

Cunningham W. (1984), "*Testing Membership in Matroid Polyhedra*", Journal of Combinatorial Theory (B) (1984) to appear.

Dantzig, G. B., D. R. Fulkerson & S. M. Johnson (1954), "*Solution of a Large-Scale Traveling Salesman Problem*", Operations Research **2** (1954) 393—410.
Dijkstra E. W. (1959), "*A Note on two Problems in Connection with Graphs*", Numerische Mathematik **1** (1959) 269—271.
Edmonds J. (1965a), "*Paths, Trees and Flowers*", Canad. J. Math. **17** (1965) 449—467.
Edmonds J. (1965b), "*Maximum Matchings and a Polyhedron with* 0 – 1 *Vertices*". J. Res. Nat. Bur. Standards Sect. B **69** (1965) 125—130.
Edmonds J. (1967), "*Optimum Branchings*", J. Res. Nat. Bur. Standards Sect. B **71** (1967) 233—240.
Edmonds J. (1970), "*Submodular Functions, Matroids and Certain Polyhedra*", in: R. Guy (ed.), "*Combinatorial Structures and Their Applications*", Proc. Calgary International Conference. Gordon and Breach, New York, (1970) 69—87.
Edmonds J. (1971), "*Matroids and the Greedy Algorithm*", Mathematical Programming **1** (1971) 127—136.
Edmonds J. & R. Giles (1977), "*A Min-Max Relation for Submodular Functions on Graphs*", Annals of Discrete Mathematics **1** (1977) 185—204.
Edmonds J. & R. Giles (1984), "*Total Integrality of Linear Inequality Systems*", Proceedings of the Silver Jubilee Conference on Combinatorics, held at the University of Waterloo, Waterloo, Canada, 1982 to appear.
Edmonds J. & R. M. Karp (1972), "*Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems*", J. ACM **19** (1972) 248—264.
Elias, P., A. Feinstein & C. E. Shannon (1956), "*A Note on the Maximum Flow through a Network*", IRE Trans. Information Theory IT **2** (1956) 117—119.
Fernandez de la Vega W. & G. S. Lueker (1981), "*Bin Packing can be Solved within* $1+\varepsilon$ *in Linear Time*", Combinatorica **1** (1981) 349—355.
Fisher M. L. (1980), "*Worst-Case Analysis of Heuristic Algorithms*", Management Science **26** (1980) 1—17.
Fisher M. L. (1981), "*Lagrangean Relaxation Methods for Combinatorial Optimization*", Management Science **27** (1981) 1—18.
Ford L. R. & D. R. Fulkerson (1956), "*Maximum Flow through a Network*", Canad. J. Math. **8** (1956) 399—404.
Frank A. (1979), "*Kernel System of Directed Graphs*", Acta Sci. Math., Szeged, **41** (1979) 63—76.
Frank A. (1984), "*Generalized Polymatroids*", Proceedings of the 6th Hungarian Combinatorial Colloquium, in: A. Hajnal. L. Lovász & V. T. Sôs (eds.) "*Finite and Infinite Sets*", North-Holland, Amsterdam, 1984, to appear.
Frederickson G. N. (1983), "*Shortest Path Problems in Planar Graphs*", 24th Annual Symposium on Foundations of Computer Science, IEEE, 1983, 242—247.
Fujishige S. (1983), "*A Characterization of a Base Polyhedron Associated with a Submodular System*", Working Paper No **83283**-OR, Institut für Ökonometrie und Operations Research, Universität Bonn, 1983.
Fulkerson D. R. (1971), "*Blocking and Anti-Blocking Pairs of Polyhedra*", Mathematical Programming **1** (1971) 168—194.
Gabow H. N. (1983), "*Scaling Algorithms for Network Problems*", 24th Annual Symposium on Foundations of Computer Science, IEEE, 1983, 248—257.
Garey M. R., R. L. Graham, D. S. Johnson & A. C. Yao (1976), "*Resource Constrained Scheduling as Generalized Bin Packing*", Journal of Combinatorial Theory A **21** (1976) 257—298.
Garey M. R. & D. S. Johnson (1979), "*Computers and Intractability: A Guide to the Theory of NP-Completeness*", Freeman, San Francisco, 1979.
Garfinkel R. S. & G. L. Nemhauser (1972), "*Integer Programming*", Wiley, London, 1972.
Geoffrion A. M. (1974), "*Lagrangean Relaxation for Integer Programming*", Mathematical Programming Studies **2** (1974) 82—114.

Giles R. & L. E. Trotter (1981), "*On Stable Set Polyhedra of $K_{1,3}$-Free Graphs*", Journal of Combinatorial Theory B **31** (1981) 313—326.
Golumbic M. C. (1980), "*Algorithmic Graph Theory and Perfect Graphs*", Academic Press, New York, 1980.
Gomory R. E. (1958), "*Outline of an Algorithm for Integer Solutions to Linear Programs*", Bull. Amer. Math. Soc. **64** (1958) 275—278.
Gomory R. E. (1960), "*Solving Linear Programming Problems in Integers*", in: R. Bellmann & M. Hall (eds.), "*Combinatorial Analysis*", American Mathematical Society, Providence, 1960, 211—216.
Grötschel M. (1982), "*Approaches to Hard Combinatorial Optimization Problems*" in: B. Korte (ed.), "*Modern applied mathematics: Optimization and Operations Research*", North-Holland, Amsterdam, 1982, 437—515.
Grötschel M. & O. Holland (1984), "*A cutting Plane Algorithm for the Matching Problem*", 1984, to appear.
Grötschel M., M. Jünger & G. Reinelt (1982), "*On The Acyclic Subgraph Polytope*", Working Paper No. **82215**-OR, Institut für Okonometrie und Operations Research, Universität Bonn, 1982.
Grötschel M., M. Jünger & G. Reinelt (1983), "*Optimal Triangulation of Large Real World Input-Output Matrices*", Preprint No. **9**, Mathematisches Institut, Universität Augsburg, 1983.
Grötschel M., L. Lovász & A. Schrijver (1981), "*The Ellipsoid Method and its Consequences in Combinatorial Optimization*", Combinatorica **1** (1981) 169—197.
Grötschel M., L. Lovász & A. Schrijver (1985), "*The Ellipsoid Method and Combinatorial Optimization*", Springer, Berlin, 1985, to appear.
Grötschel M. & M. W. Padberg (1984), "*Polyhedral Aspects of the Traveling Salesman Problem I: Theory*" in: E. L. Lawler, J. K. Lenstra & A. H. G. Rinnooy Kan, (eds.), "*The Traveling Salesman Problem*", Wiley, 1984, to appear.
Grünbaum B. (1967), "*Convex Polytopes*", Wiley, London, 1967.
Halton J. H. & R. Terada (1982), "*A Fast Algorithm for the Euclidean Traveling Salesman Problem, Optimal with Probability One*", SIAM Journal on Computing **11** (1982) 28—46.
Hartmanis J. & J. E. Hopcroft (1976), "*Independence Results in Computer Science*", SIGACT News **8,4** (1976) 13—24.
Hassin R. (1978), "*On Network Flows*", Ph. D. Thesis, Yale University, Boston, 1978.
Hausmann D. (1978), "*Integer Programming and Related Areas: A Classified Bibliography* 1976—1978", Lecture Notes in Economics and Mathematical Systems **160**, Springer, Berlin, 1978.
Hochbaum D. & D. Shmoys (1984), "*A Best Possible Heuristic for the k-Center Problem*", Mathematics of Operations Research, 1984, to appear.
Hoffman A. J. & J. B. Kruskal (1956), "*Integral Boundary Points of Convex Polyhedra*", in: H. W. Kuhn & A. W. Trucker (eds.), "*Linear Inequalities and Related Systems*", Ann. of Math. Studies **38**, Princeton Univ. Press, Princeton, N. J., 1956, 233—246.
Hoffman A. J. & D. E. Schwartz (1978), "*On Lattice Polyhedra*", in: A. Hajnal & V. T. Sôs (eds.), "*Combinatorics*", North-Holland, Amsterdam, 1978, 593—598.
Hopcroft J. E. & R. E. Tarjan (1972), "*Efficient Planarity Testing*", SIAM Journal on Computing **2** (1973) 225—231.
Ibarra O. H. & C. E. Kim (1975), "*Fast Approximation Algorithms for the Knapsack and Subset Sum Problems*", J. Assoc. Comput. Mach. **22** (1975) 463—468.
Iri M. (1983), "*Applications of Matroid Theory*", in: A. Bachem, M. Grötschel & B. Korte (eds.), "*Mathematical Programming — The State of the Art, Bonn* 1982", Springer, Heidelberg, 1983, 158—201.
Johnson D. S. (1973), "*Near-Optimal Bin Packing Algorithms*", Ph. D. Thesis, Department of Mathematics, MIT, Cambridge, Massachussetts, 1973.
Johnson D. S. (1983), "The $\mathcal{NP}$-Completeness Column: An Ongoing Guide", Journal of Algorithms **4** (1983) 189—203 and 286—300.

Johnson D. S., A. Demers, J. D. Ullman, M. R. Garey & R. L. Graham (1974), "*Worst-Case Performance Bounds for Simple One-Dimensional Packing Algorithms*", SIAM Journal on Computing **3** (1974) 299—325.

Kannan R. & A. Bachem (1979), "*Polynomial Algorithms for Computing the Smith and Hermite Normal Forms of an Integer Matrix*", SIAM Journal on Computing **8** (1979) 499—507.

Karmarkar N. & R. M. Karp (1982), "*An Efficient Approximation Scheme for the One-Dimensional Bin Packing Problem*", Proc. 23rd Annual Symposium on Foundations of Computer Science, 1982, 212—230.

Karp R. M. (1972), "*Reducibility Among Combinatorial Problems*", in: R. E. Miller & J. W. Thatcher (eds.), "*Complexity of Computer Computations*", Plenum Press, New York, 1972, 85—103.

Karp R. M. (1976), "*The Probabilistic Analysis of Some Combinatorial Search Algorithms*", Memorandum No. ERL-M **581**, University of California, Berkeley 1976.

Karp R. M. (1984), "*Probabilistic Analysis of Deterministic and Probabilistic Algorithms: An Annotated Bibliography*", in: O'hEigeartaigh et al. (eds.) (1984) to appear.

Karp R. M. & Ch. H. Papadimitriou (1982), "*On Linear Characterization of Combinatorial Optimization Problems*", SIAM Journal on Computing **11** (1982) 620—632.

Kastning C. (1976), "*Integer Programming and Related Areas: A Classified Bibliography*", Lecture Notes in Economics and Mathematical Systems **128**, Springer, Berlin, 1976.

Khachiyan L. G. (1979), "*A Polynomial Algorithm in Linear Programming*", Soviet Math. Dokl. **20** (1979) 191—194.

Kindervater G. A. P. & J. K. Lenstra (1984), "*Parallel Algorithms in Combinatorial Optimization: An Annotated Bibliography*", in: O'hEigeartaigh et al. (eds.) (1984) to appear.

Klee V. (1980), "*Combinatorial Optimization: What is the State of the Art*", Mathematics of Operations Research **5** (1980) 1—26.

Klee V. & P. Kleinschmidt (1984), "*The d-Step Conjecture and its Relatives*", SIAM, Philadelphia, 1984, to appear.

Korte B. (1979), "*Approximative Algorithms for Discrete Optimization Problems*", Annals of Discrete Mathematics **4** (1979) 85—120.

Korte B. & L. Lovász (1983), "*Structural Properties of Greedoids*", *Combinatorica* **3** (1983) 359—374.

Lagarias J. C. & A. M. Odlyzko (1983), "*Solving Low-Density Subset Sum Problems*", 24th Annual Symposium on Foundations of Computer Science, IEEE, 1983, 1—10.

Lawler E. L. (1976), "*Combinatorial Optimization: Networks and Matroids*", Holt, Rinehart & Winston, New York, 1976.

Lawler, E. L., J. K. Lenstra & A. H. G. Rinnooy Kan (1984), "*The Traveling Salesman Problem*", Wiley, New York, 1984, to appear.

Lawler E. L. & C. U. Martel (1982), "*Flow Network Formulations of Polymatroid Optimization Problems*", Annals of Discrete Mathematics **16** (1982) 189—200.

Lenstra A. K., H. W. Lenstra jun. & L. Lovász (1982) "*Factoring Polynomials with Rational Coefficients*", Math. Ann. **261** (1982) 515—534.

Lenstra jun. H. W. (1983), "*Integer Programming with a Fixed Number of Variables*", Mathematics of Operations Research **8** (1983) 538—548.

Lovász L. (1979), "*Combinatorial Problems and Exercises*", North-Holland, Amsterdam, 1979.

Lovász L. (1981), "*The Matroid Matching Problem*", in: L. Lovász & V. T. Sôs (eds.) "*Algebraic Methods in Graph Theory*", North-Holland, Amsterdam, 1981, 495—517.

Lovász L. (1982), "*Bounding the Independence Number of a Graph*", Annals of Discrete Mathematics **16** (1982) 213—223.

Lovász L. (1983), "*Submodular Functions and Convexity*", in: A. Bachem, M. Grötschel & B. Korte (eds.), "*Mathematical Programming — The State of the Art, Bonn* 1982", Springer, Heidelberg, 1983, 235—257.

Lovász L. & M. Plummer (1984), "*The Matching Structure of Graphs*", Akademia Kiado, Budapest, 1984, to appear.

Lueker G. S. (1979), "*Maximization Problems on Graphs with Edge Weights Chosen from a Normal Distribution*", Proc. Xth Annual ACM Symp. on Theory of Computing 1978.
Mader W. (1978a), "*Über die Maximalzahl kantendisjunkter A-Wege*", Arch. Math., Basel, **30** (1978) 325—336.
Mader W. (1978b), "*Über die Maximalzahl kreuzungsfreier H-Wege*", Arch. Math., Basel, **31** (1978) 387—402.
Minty G. J. (1980), "*On Maximal Independent Sets of Vertices in a Claw-Free Graph*", J. Combinatorial Theory B **28** (1980) 284—304.
von Neumann J. (1953), "*A Certain Zero-Sum Two-Person Game Equivalent to the Optimum Assignment Problem*", in: W. Tucker & H. W. Kuhn (eds.), "*Contributions to the Theory of Games II*", Annals of Math. Studies **38**, Princeton Univ. Press, Princeton, N. J., 1953, 5—12.
O'hEigeartaigh M., J. K. Lenstra & A. H. G. Rinnooy Kan (1984), "*Combinatorial Optimization: Annotated Bibliographies*", Wiley, New York, 1984, to appear.
Padberg, M. W. & M. Grötschel (1984), "*Polyhedral Aspects of the Traveling Salesman Problem II: Computation*", in: E. L. Lawler, J. K. Lenstra & A. H. G. Rinnooy Kan (eds.), "*The Traveling Salesman Problem*", Wiley, 1984, to appear.
Padberg M. W. & M. R. Rao (1982), "*Odd Minimum Cut-Sets and b-Matchings*", Mathematics of Operations Research 7 (1982) 67—80.
Padberg M. W. & M. R. Rao (1984), "*The Russian Method for Linear Inequalities III: Bounded Integer Programming*", Mathematical Programming Studies, 1984, to appear.
Papadimitriou Ch. H. (1984), "*Polytopes and Complexity*", in:: Proceedings of the Silver Jubilee Conference on Combinatorics, held at the University of Waterloo, Waterloo, Canada, 1982, 1984, to appear.
Papadimitriou Ch. H. & K. Steiglitz (1982), "*Combinatorial Optimization: Algorithms and Complexity*", Prentice-Hall, Englewood Cliffs, 1982.
Pulleybank W. R. (1983), "*Polyhedral Combinatorics*", in: A. Bachem, M. Grötschel & B. Korte (eds.), "*Mathematical Programming — The State of the Art, Bonn* 1982", Springer, Heidelberg, (1983), 312—345.
von Randow R. (1982), "*Integer Programming and Related Areas: A Classified Bibliography* 1978—1981", Lecture Notes in Economics and Mathematical Systems **197**, Springer, Berlin 1982.
Recski A. (1985), "*Matroid Theory and its Applications*", Springer, Berlin 1985, to appear.
te Riele H. (1983), "*Mertens' Conjecture Disproved*", Research Announcement, CWI Newsletter, Amsterdam **1** (1983) 23—24.
Rivest R. & S. Vuillemin (1978), "*On Recognizing Graph Properties from Adjacency Matrices*", Theor. Comp. Sci. **3** (1978) 371—384.
Roberts F. S. (1978), "*Graph Theory and its Applications to Problems of Society*", SIAM, Philadelphia, 1978.
Sbihi N. (1978), "*Etudes des stables dans les graphes sans étoile*", M. Sc. Thesis, Univ. Sci. et Méd. Grenoble, 1978.
Schrijver A. (1980), "*On Cutting Planes*", Annals of Discrete Mathematics **9** (1980) 291—296.
Schrijver A. (1983), "*Min-Max Results in Combinatorial Optimization*", in:A. Bachem, M. Grötschel & B. Korte (eds.), "*Mathematical Programming — The State of the Art, Bonn* 1982", Springer, Heidelberg, 1983, 439—500.
Schrijver A. (1984a), "*Polyhedral Combinatorics*", Wiley, New York, 1984, to appear.
Schrijver A. (1984b), "*Total Dual Integrality from Cross-Free Families — a General Framework*", Mathematical Programming (1984) to appear.
Schrijver A. (1984c), "*Total Dual Integrality from Graphs, Crossing Families, and Sub- and Supermodular Functions*", Proceedings of the Silver Jubilee Conference on Combinatorics, Univ. of Waterloo, Academic Press, 1984, to appear.
Seymour P. D. (1977), "*The Matroids with the Max-Flow Min-Cut Property*", Journal of Combinatorial Theory B **23** (1977) 189—222.

Seymour P. D. (1980), "*Decomposition of Regular Matroids*", Journal of Combinatorial Theory B, **28** (1980) 305—359.
Seymour P. D. (1981), "*Matroids and Multicommodity Flows*", Europ. J. Comb. **2** (1981) 257—290.
Shor N. Z. (1970), "*Convergence Rate of the Gradient Descent Method with Dilatation of the Space*", Cybernetics **6** (1970) 102—108.
Shor N. Z. (1977), "*Cut-Off Method with Space Extension in Convex Programming Problems*", Cybernetics **13** (1977) 94—96.
Strassen V. (1969), "*Gaussian Elimination is Not Optimal*", Numer. Math. **13** (1969) 354—356.
Tarjan R. E. (1978), "*Complexity of Combinatorial Algorithms*", SIAM Rev. **20** (1978) 457—491.
Tarjan R. E. (1983), "*Data Structures and Network Algorithms*", Society for Industrial and Applied Mathematics, Philadelphia, 1983.
Tarjan R. E. (1983a), "*Algorithms for Maximum Network Flow*", Paper presented at the NETFLOW **83** Workshop, Pisa, Italy, 1983.
Tutte W. T. (1965), "*Lectures on Matroids*", J. Res. Nat. Bur. Standards **69** B (1965) 1—47.
Weide B. (1977), "*A Survey of Analysis Techniques for Discrete Algorithms*", Computing Surveys **9** (1977) 291—313.
Weide B. (1980), "*Random Graphs and Graph Optimization Problems*", SIAM J. Comput. **9** (1980) 552—557.
Welsh D. J. A. (1976), "*Matroid Theory*", Academic Press, New York, 1976.
Yamnitsky B. & L. A. Levin (1982), "*An Old Linear Programming Algorithm Runs in Polynomial Time*", Paper presented at the Silver Jubilee Conference, Waterloo, Ontario, Canada, 1982.
Yao A. C. (1980), "*New Algorithms for Bin Packing*", J. ACM **27** (1980) 207—227.