

# Polyhedral techniques in combinatorial optimization

Citation for published version (APA):

Aardal, K., & van Hoesel, C. P. M. (1995). *Polyhedral techniques in combinatorial optimization*. METEOR, Maastricht University School of Business and Economics. METEOR Research Memorandum No. 014 <https://doi.org/10.26481/umamet.1995014>

## Document status and date:

Published: 01/01/1995

## DOI:

[10.26481/umamet.1995014](https://doi.org/10.26481/umamet.1995014)

## Document Version:

Publisher's PDF, also known as Version of record

## Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

## General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

[www.umlib.nl/taverne-license](http://www.umlib.nl/taverne-license)

## Take down policy

If you believe that this document breaches copyright please contact us at:

[repository@maastrichtuniversity.nl](mailto:repository@maastrichtuniversity.nl)

providing details and we will investigate your claim.

# Polyhedral Techniques in Combinatorial Optimization

Karen Aardal

Department of Econometrics  
Tilburg University  
P.O. Box 90153  
5000 LE Tilburg, The Netherlands  
aardal@kub.nl

Stan van Hoesel

Department of Quantitative Economics  
Limburg University  
P.O. Box 616  
6200 MD Maastricht, The Netherlands  
s.vanhoesel@ke.rulimburg.nl

April 28, 1995

## Abstract

Combinatorial optimization problems arise in several areas ranging from management to mathematics and graph theory. Most combinatorial optimization problems are computationally hard due to the restriction that a subset of the variables have to take integral values. During the last two decades there has been a remarkable development in polyhedral techniques leading to an increase in the size of several problem types that can be solved by a factor hundred. The basic idea behind polyhedral techniques is to derive a good linear formulation of the set of solutions by identifying linear inequalities that can be proved to be necessary in the description of the convex hull of feasible solutions. The purpose of this article is to give an overview of the developments in polyhedral theory, starting with the pioneering work by Dantzig, Fulkerson and Johnson on the traveling salesman problem, and by Gomory on integer programming. We also discuss several computational aspects and implementation issues related to the use of polyhedral methods.

Combinatorial optimization deals with maximizing or minimizing a function subject to a set of constraints and subject to the restriction that some, or all, variables should be integers. Several problems that occur in management and planning situations can be formulated as combinatorial optimization problems, such as the *lot sizing problem*, where we need to decide on which time periods to produce, and how much to produce in these periods, to satisfy customers' demand at minimal total production, storage and setup costs. Another well-known combinatorial optimization problem is the *traveling salesman problem* where we want to determine in which order a "salesman" has to visit a number of "cities" such that all cities are visited exactly once and such that the length of the tour is minimal. This problem is one of the most studied combinatorial optimization problems, not because of its importance in the planning of salesmen tours, but because of its numerous other applications, both in its own right and as a substructure of more complex models, and because it is notoriously difficult to solve. The combination of being easy to state, relatively easy to formulate as a mathematical programming problem, but computationally intractable is something a majority of combinatorial optimization problems have in common.

The computational intractability of most core combinatorial optimization problems has been theoretically indicated, i.e. it is possible to show that most of these problems belong to the class of *NP-hard problems*, see Karp (1972), and Garey and Johnson (1979). No algorithm with a worst-case running time bounded by a polynomial in the size of the input is known for any NP-hard problem, and it is strongly believed that no such algorithm exists. Therefore, to solve these problems we have to use an *enumerative algorithm*, such as dynamic programming or branch and bound, with a worst-case running time that is exponential in the size of the input. The computational hardness of most combinatorial optimization problems has inspired researchers to develop good formulations, and algorithms that are expected to reduce the size of the enumeration tree. To use information about the structure of the convex hull of feasible solutions, which is the basis for polyhedral techniques, has been one of the most successful approaches so far. The pioneering work in this direction was done by Dantzig, Fulkerson and Johnson (1954), who invented a method to solve the traveling salesman problem. They demonstrated the power of their technique on a 49-city instance, which was huge at that time.

The idea behind the Dantzig-Fulkerson-Johnson method is the following. Assume we want to solve the problem

$$\min\{cx \text{ subject to } x \in S\}, \tag{1}$$

where  $S$  is the set of feasible solutions, which in our case is the set of traveling salesman tours. Let  $S = P \cap \mathbb{Z}^n$ , where  $P = \{x \in \mathbb{R}^n : Ax \leq b\}$  and where  $Ax \leq b$  is a system of linear inequalities. Since  $S$  is difficult to characterize, we could solve the problem

$$\min\{cx \text{ subject to } x \in P\} \tag{2}$$

instead. Problem (2) is easy to solve, but since it is a relaxation of (1) it may give us a solution  $x^*$  that is not a tour. More precisely, the following two things can happen if we solve (2): either the optimal solution  $x^*$  is a tour, which means that  $x^*$  is also optimal for (1), or  $x^*$  is not a tour, in which case it is not feasible for (1). If the solution  $x^*$  is not feasible for (1) it lies outside the convex hull of  $S$  which means we can cut off  $x^*$  by identifying a hyperplane separating

$x^*$  from the convex hull of  $S$ , i.e. a hyperplane that is satisfied by all tours, but violated by  $x^*$ . An inequality that is satisfied by all feasible solutions is called a *valid inequality*. When Dantzig, Fulkerson and Johnson solved the relaxation (2) of their 49-city instance they indeed obtained a solution  $x^*$  that was not a tour. By looking at the solution they identified a valid inequality that was violated by  $x^*$ , and added this inequality to the formulation. They solved the resulting linear programming problem and obtained again a solution that was not a tour. After repeating this process a few times a tour was obtained, and since only valid inequalities were added to the relaxation, they could conclude that the solution was optimal.

Even though many theoretical questions regarding the traveling salesman problem remained unsolved, the work of Dantzig, Fulkerson and Johnson was still a breakthrough as it provided a methodology that was actually not limited to solving traveling salesman problems, but could be applied to any combinatorial optimization problem. This new area of research on how to describe the convex hull of feasible solutions by linear inequalities was called *polyhedral combinatorics*. During the last decades polyhedral techniques have been used with considerable success to solve many previously unsolved instances of hard combinatorial optimization problems, and it is still the only method available for solving large instances of the traveling salesman problem. The purpose of this paper is to describe theoretical and computational aspects of polyhedral techniques and to partially survey the results that have been obtained by applying this approach.

A natural question that arises when studying the work by Dantzig, Fulkerson and Johnson is whether it is possible to develop a general scheme for identifying valid inequalities. This question was answered by Gomory (1958), (1960), (1963) who developed a *cutting plane algorithm* for general integer linear programming, and showed that the integer programming problem  $\min\{cx \text{ subject to } x \in S\}$  can be solved by solving a finite sequence of linear programs. Chvátal (1973) proved that all inequalities necessary to describe the convex hull of integer solutions can be obtained by taking linear combinations of the original and previously generated linear inequalities and then applying a certain rounding scheme, provided that the integer solutions are bounded. Schrijver (1980) proved the more general result that it is possible to generate the convex hull of integer solutions by applying a finite set of operations on the polyhedron describing the integer solutions, if this polyhedron is rational, but not necessarily bounded. The results by Gomory, Chvátal, and Schrijver are discussed in Section 1. Here we will also address the following two questions: When can we expect to have a concise description of the convex hull of feasible solutions? How difficult is it to identify a violated inequality? These questions are strongly related to the computational complexity of the considered problem, i.e. the hardness of a problem type will catch up with us at some point, but we shall also see that certain aspects of the answers make it possible to hope that a bad situation can be turned into a rather promising one.

The results of Gomory, Chvátal and Schrijver were very important theoretically, but they did not provide direct tools for solving realistic instances within reasonable time. Researchers therefore began to develop problem specific classes of inequalities that contain inequalities that can be proved to be necessary in the description of the convex hull of feasible solutions. Based on the various classes of valid inequalities it is then necessary to develop *separation*

*algorithms*, i.e. algorithms for identifying violated inequalities given the current solution  $x^*$ . In Section 2 we begin by describing families of valid inequalities for some basic combinatorial optimization problems, and the corresponding separation problems. These inequalities are important as they are often useful when solving more complex problems as well, either directly, or as a starting point for developing new, more general families of inequalities. Moreover, they represent different arguments that can be used when developing valid inequalities. We shall also give a partial survey of polyhedral results for combinatorial optimization problems.

Next to the theoretical work of developing good classes of valid inequalities and algorithms for identifying violated inequalities, there is a whole range of implementation issues that have to be considered in order to make polyhedral methods work well. One such issue is *preprocessing*. Important elements of preprocessing are to reduce the size of the initial formulation by deleting unnecessary variables and constraints, and to reduce the size of the constraint coefficients to make the instance numerically more attractive. In the course of strengthening the relaxation by adding valid inequalities we may also want to delete some of the previously added inequalities to avoid the formulation growing too much. We may also want to work with a partial set of variables to speed up computations. Dantzig, Fulkerson and Johnson were able to find the optimal solution by adding valid inequalities only. In general however we end up in the situation where the current solution  $x^*$  is not feasible and where we are unable to identify an inequality violated by  $x^*$ . We then have to start a branch-and-bound phase. For the branch-and-bound algorithm we must decide precisely how to create new subproblems, or nodes, in the search tree, as well as a suitable search strategy. It is also possible to add inequalities in every node of the tree, in which case we need to keep track of where in the tree the various inequalities are valid. All these issues are discussed in Section 3. To illustrate the computational possibilities of polyhedral techniques we present computational results for some selected problem types in Section 4.

Even though polyhedral combinatorics has been the foremost tool for computing large instances of a vast collection of combinatorial optimization problems it is not the only technique available, and depending on the problem type it may be preferable to choose a different method. We conclude our article by briefly mentioning alternative approaches to solving integer and combinatorial optimization problems.

## Dutch Results

# 1 Theoretical background

The integer linear programming problem (ILP) is defined as

$$\min\{cx : x \in S\} \tag{3}$$

where  $S = P \cap \mathbb{Z}^n$  and  $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ . We call  $P$  the *linear formulation* of ILP. A polyhedron  $P$  is *rational* if it can be determined by a rational system  $Ax \leq b$  of linear inequalities, i.e., if all entries of  $A$  and  $b$  are rationals. The convex hull of the set  $S$  of feasible solutions, denoted  $\text{conv}(S)$ , is the smallest convex set containing  $S$ . A facet-defining valid

inequality is a valid inequality that is necessary to describe  $\text{conv}(S)$ , i.e. it is the “strongest possible” valid inequality. In Figure 1 we give an example of sets  $P$ ,  $S$  and  $\text{conv}(S)$ .

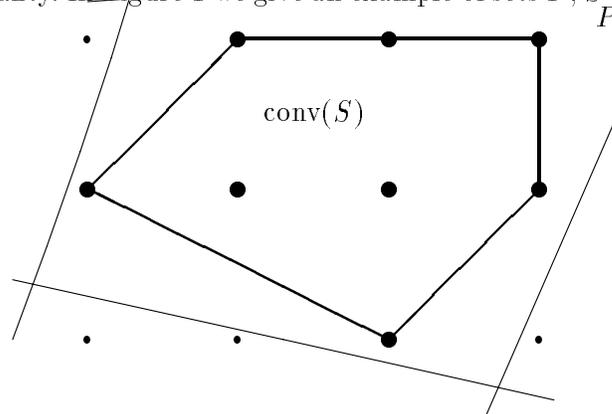


Figure 1:  $P$ ,  $S$ , and  $\text{conv}(S)$ .

If we know the linear description of  $\text{conv}(S)$  we can solve the linear programming problem  $\min\{cx : x \in \text{conv}(S)\}$  which is computationally easy. In this section we shall primarily address the issue of how difficult it is to obtain  $\text{conv}(S)$ . First we show that for rational polyhedra, and for not necessarily rational bounded polyhedra, we can generate  $\text{conv}(S)$  algorithmically in a finite number of steps. In general however, there is no upper bound on the number of steps in terms of the dimension of  $S$ . We also demonstrate that it is very unlikely that  $\text{conv}(S)$  of any NP-hard problem can be described by concise families of linear inequalities. Finally, we relate the complexity of the problem of finding a hyperplane separating a vector  $x^*$  from  $\text{conv}(S)$  or showing that  $x^*$  belongs to  $\text{conv}(S)$ , to the complexity of the optimization problem given  $S$ . In general these two problems are equally hard, but if we restrict the search of a separating hyperplane to a specific class, this problem might be polynomially solvable even if the underlying optimization problem is NP-hard.

## 1.1 Solving Integer Programming Problems by Linear Programming

### 1.1.1 Gomory’s Cutting Plane Algorithm

What was needed to transform the procedure of Dantzig, Fulkerson and Johnson (1954) into an algorithm was a systematic procedure for generating valid inequalities that are violated by the current solution. Assume that we want to solve the variant of ILP where the integer vectors in  $S$  are bounded and where all entries of the constraint matrix  $A$  and the right-hand side vector  $b$  are integers. Gomory (1958), (1960) and (1963) developed a *cutting plane algorithm* based on the simplex method, for solving integer linear problems on this form. This was the first algorithm developed for integer linear programming that could be proved to terminate in a finite number of iterations. The basic idea of Gomory’s algorithm is similar to the approach of Dantzig, Fulkerson and Johnson, i.e. instead of solving ILP directly we solve the linear programming (LP) relaxation  $\min\{cx : x \in P\}$  by the simplex method. If the optimal solution

to LP is integral, then we are done, and otherwise we need to identify a valid inequality cutting off  $x^*$ . Gomory developed a technique for automatically identifying a violated valid inequality and proved that after adding a finite number of inequalities, called *Gomory cutting planes*, the optimal solution is obtained. We shall illustrate Gomory's technique by an example. Assume we have solved the linear relaxation of an instance of ILP, as described above, by the simplex method, and that one of the rows of the tableau reads

$$x_1 - \frac{1}{11}x_3 + \frac{2}{11}x_4 = \frac{36}{11}$$

where  $x_1$  is a basic variable and variables  $x_3$  and  $x_4$  are non-basic, i.e. at the current solution  $x_1 = 36/11$  and  $x_3 = x_4 = 0$ . We now split each coefficient in an integer and a fractional part by rounding down all coefficients. The integer terms are put in the left-hand side of the equation and the fractional terms are put in the right-hand side. Since all coefficients are rounded down, the fractional part of the variable coefficients in the right-hand side becomes nonpositive,

$$x_1 - x_3 - 3 = -\frac{10}{11}x_3 - \frac{2}{11}x_4 + \frac{3}{11}.$$

In any feasible solution to ILP, the left-hand side should be integral. Moreover, all variables are nonnegative. Since the variables in the right-hand side appear with nonpositive coefficients we can conclude that

$$\frac{3}{11} - \frac{10}{11}x_3 - \frac{2}{11}x_4 \leq 0, \text{ and integer.} \quad (4)$$

We have argued that the inequality (4) is valid, i.e. it is not violated by any feasible integer solution. It is easy however to see that it does cut off the current fractional solution as  $x_3 = x_4 = 0$ . Let  $\lfloor x \rfloor$  denote the integer part of  $x$ .

### Outline of Gomory's cutting plane algorithm.

1. Solve the linear relaxation of ILP with the simplex method. The current number of variables is  $k$ . If the optimal solution  $x^*$  is integral, stop.
2. Choose a source row  $i_0$  in the optimal tableau with a fractional basic variable. Row  $i_0$  reads  $\bar{a}_{i_0,1}x_1 + \bar{a}_{i_0,2}x_2 + \dots + \bar{a}_{i_0,k}x_k = \bar{b}_{i_0}$ . Let  $a'_{ij} = \bar{a}_{ij} - \lfloor \bar{a}_{ij} \rfloor$ , and  $b'_i = \bar{b}_i - \lfloor \bar{b}_i \rfloor$ .
3. Add the equation  $-a'_{i_0,1}x_1 - a'_{i_0,2}x_2 - \dots - a'_{i_0,k}x_k + x_{k+1} = -b'_{i_0}$ , where  $x_{k+1}$  is a slack variable, to the current linear formulation, and reoptimize using the dual simplex method. If the optimal solution  $x^*$  is integral, stop, otherwise  $k \leftarrow k + 1$ , go to 2.

In the outline above we have not specified how to choose the source row. To be able to prove that the algorithm terminates in a finite number of steps we have to make sure that certain technical conditions are satisfied. The technical details are omitted here but can be found in Gomory (1963) who gives two proofs of finiteness, and in Schrijver (1986), page 357.

**Theorem 1** Gomory (1963). *There exists an implementation of Gomory's cutting plane algorithm such that after a finite number of iterations either an optimal integer solution is found, or it is proved that  $S = \emptyset$ .*

A recent discussion on Gomory cutting planes can be found in Balas et al. (1994) who incorporate the cutting plane algorithm in a branch-and-bound procedure and report on computational experience.

### 1.1.2 Chvátal's Rounding Procedure

Chvátal (1973) studied the more general version of ILP, where the integer vectors of  $S$  are bounded and where the entries of  $A$  and  $b$  are real numbers. He showed that if one takes linear combinations of the linear inequalities defining  $P$  and then applies rounding, and repeats the procedure a finite number of times,  $\text{conv}(S)$  is obtained. After each iteration of the procedure we get a new linear formulation containing more inequalities. We again illustrate the procedure by an example. Note that this example will be referred to frequently in the sequel. Let  $G = (V, E)$  be an undirected graph where  $V$  is the set of vertices and  $E$  is the set of edges. A *matching*  $M$  in a graph is a subset of edges such that each vertex is incident to at most one edge in  $M$ , see **Figure ?**. Let  $x_e$  be equal to one if edge  $e$  belongs to the matching  $M$  and zero otherwise, and let  $\delta(v) = \{e \in E : e \text{ is incident to } v\}$ . The maximum cardinality matching problem can be formulated as the following integer linear programming problem.

$$\max \sum_{e \in E} x_e \tag{5}$$

$$\text{s.t.} \quad \sum_{e \in \delta(v)} x_e \leq 1 \quad \text{for all } v \in V \tag{6}$$

$$0 \leq x_e \leq 1 \quad \text{for all } e \in E \tag{7}$$

$$x_e \text{ integer} \quad \text{for all } e \in E \tag{8}$$

**Figure of matching plus mention possible applications** Let  $U$  be any subset consisting of  $k$  vertices, where  $k \geq 3$  and odd, and let  $E(U)$  be the set of edges with both end-vertices in  $U$ . By adding inequalities (6) for all  $v \in U$  we obtain  $2 \sum_{e \in E(U)} x_e \leq |U|$ , or equivalently

$$\sum_{e \in E(U)} x_e \leq \frac{|U|}{2}. \tag{9}$$

Since each  $x_e$  is an integer, the left-hand side of (9) has to be integral. As  $|U|$  is odd, the right-hand side of (9) is fractional, and hence we can round down the right-hand side of (9) giving the valid inequality

$$\sum_{e \in E(U)} x_e \leq \left\lfloor \frac{|U|}{2} \right\rfloor \tag{10}$$

which we call an *odd-set constraint*. It is easy to show that the odd-set constraints are necessary to describe the convex hull of matchings in  $G$ . We also note that there are exponentially many odd-set constraints as there are exponentially many ways of forming subsets  $U$ . We shall now give a more formal description of Chvátal's procedure.

An inequality  $\sum_{j=1}^n a_j x_j \leq b$  is said to belong to the *elementary closure* of a set  $P$  of linear inequalities, denoted  $e^1(P)$ , if there are inequalities  $\sum_{j=1}^n a_{ij} x_j \leq b_i$   $i = 1, \dots, m$  in  $P$  and nonnegative real numbers  $\lambda_1, \lambda_2, \dots, \lambda_m$  such that

$$\sum_{i=1}^m \lambda_i a_{ij} = a_j \quad \text{with } a_j \text{ integer, } j = 1, \dots, n,$$

and

$$\left\lfloor \sum_{i=1}^m \lambda_i b_i \right\rfloor \leq b.$$

For integer values of  $k > 1$ ,  $e^k(P)$  is defined recursively as  $e^k(P) = \epsilon(P \cup e^{k-1}(P))$ . The *closure* of  $P$  is defined as  $c(P) = \cup_{k=1}^{\infty} e^k(P)$ .

**Theorem 2** Chvátal (1973). *If  $S$  is a bounded polyhedron, then  $\text{conv}(S)$  can be obtained after a finite number  $k$  of closure operations.*

An interesting question is if  $k$  can be bounded from above by a function of the dimension of  $S$ . Chvátal called the minimum number of closure operations  $k$  required to obtain  $\text{conv}(S)$ , given a linear formulation  $P$ , the *rank* of  $P$ . If we return to the matching problem (6)–(8), it was proved by Edmonds (1965) that the convex hull of the matching polytope is determined by inequalities (6), (7) and (10). As the odd-set constraints (10) can be obtained by applying one closure operation on the linear formulation, the rank of the set of inequalities (6) and (7) is one. In general however, there is no upper bound on  $k$  in terms of the dimension of  $S$  as the following two-dimensional problem illustrates.

$$\begin{array}{rcll} \max & & x_2 & \\ & x_1 & & \leq 1 \\ & x_1 & & \geq 0 \\ -tx_1 & + & x_2 & \leq 1 \\ tx_1 & + & x_2 & \leq t + 1 \\ & & x_2 & \geq 0 \\ & & x_1, x_2 & \text{integer} \end{array}$$

**Explain the outcome of the example** Only if  $S = \emptyset$  does there exist an upper bound on  $k$  that is a function of the dimension of  $P$ . This was proved by Cook et al. (1987).

There is a clear relation between Chvátal's closure operations and Gomory's cutting planes in the sense that every Gomory cutting plane can be obtained by a series of closure operations and every inequality belonging to the elementary closure can be obtained as a Gomory cutting plane. It would be possible to prove Theorem 2 using Gomory's algorithm, but then one would first need to get rid of the inequalities  $x_j \geq 0$ ,  $j = 1, \dots, n$  and the assumption that the entries of  $A$  and  $b$  have to be integer. For further details, see Chvátal (1973).

### 1.1.3 Schrijver's Rounding Procedure

Schrijver (1980) studied the version of ILP where  $S$  is not necessarily bounded, and where  $P$  is defined by a rational system of linear inequalities. The operations carried out on  $P$  to obtain

the convex hull of feasible solutions is quite different from the linear combination and rounding schemes developed by Gomory and Chvátal. The key component of Schrijver's procedure is the formulation of a *totally dual integral* (TDI) system of inequalities. A rational system  $Ax \leq b$  of linear inequalities is TDI if for all integer vectors  $c$  such that  $\max\{cx : Ax \leq b\}$  is finite, the dual,  $\min\{yb : yA = c, y \geq 0\}$ , has an integer optimal solution. Note that if  $Ax \leq b$  is TDI, and if  $b$  is integral, then  $P = \{x : Ax \leq b\}$  is an integral polyhedron, i.e. all extreme points of  $P$  are integral. TDI systems were introduced by Edmonds and Giles (1977).

Each iteration of Schrijver's procedure consists of the following two steps.

1. Given a rational polyhedron  $P$ , find a TDI system  $Ax \leq b$  defining  $P$ , with  $A$  integral.
2. Round down the right-hand side  $b$ .

It has been proved by Giles and Pulleyblank (1979) and Schrijver (1981) that there exists a TDI system as in 1. for every rational polyhedron  $P$ , and that the TDI system is unique if  $P$  is full-dimensional. Finding such a TDI system can be done in finite time. After one iteration of the above procedure we get a polyhedron  $P^{(1)}$  strictly contained in  $P$  unless  $P$  is integral. Given the polyhedron  $P^{(1)}$  we repeat the steps 1. and 2. This continues until  $\text{conv}(S)$  is obtained.

**Theorem 3** Schrijver (1980). *For each rational polyhedron  $P$ , there exists a number  $k$ , such that after  $k$  iterations of Schrijver's procedure  $\text{conv}(S)$  is obtained.*

The results presented above are of significant theoretical importance as they give algorithmic ways of generating the convex hull of feasible solutions. All three approaches are finite, but from a practical point of view finite in most cases does not imply that computations can be done within reasonable time. One apparent question is whether for some problem classes it is possible to write down the linear description of the convex hull in terms of concise families of linear inequalities. If that is possible we could apply linear programming directly. This is the topic of the following subsection.

## 1.2 Concise Linear Descriptions

We mentioned in the previous subsection that the convex hull of matchings in a general undirected graph  $G$  is given by the defining inequalities (6), (7) and the exponential class of inequalities (10). Assume now that  $G$  is *bipartite*, i.e. that we can partition the set  $V$  of vertices into two sets  $V_1, V_2$  such that all edges have one endvertex in  $V_1$  and the other endvertex in  $V_2$ . For bipartite graphs the convex hull of matchings is described by the defining inequalities (6) and (7) only, which is a polynomial system of linear inequalities. This means that for bipartite graphs the integrality condition (8) is redundant. In contrast, there is no concise linear description known for the traveling salesman problem, even if we allow for exponential families of inequalities. The reason why the bipartite matching problem is so easy is that the constraint matrix is *totally unimodular* (TU). A matrix  $A$  is TU if each subdeterminant of  $A$  is equal to 0, 1 or -1.

**Theorem 4** *If  $A$  is a TU matrix the polyhedron  $P = \{x : Ax \leq b\}$  is integral for all integer vectors  $b$  for which  $P$  is not empty.*

Seymour (1980) provided a complete characterization of TU matrices yielding a polynomial algorithm for testing whether a matrix is TU. For a thorough discussion on TU matrices we refer to Schrijver (1986), and Nemhauser and Wolsey (1988).

An observation that is interesting to make in this context is that the bipartite matching problem is polynomially solvable as its linear description is polynomial in the dimension of the problem. For the matching problem in general undirected graphs there is a polynomial combinatorial algorithm due to Edmonds (1965), but the traveling salesman problem is known to be NP-hard. The following theorem confirms that there is a natural link between the computational complexity of a class of problems and the possibility of providing concise linear descriptions of the convex hull of feasible solutions. Before stating the result we need to introduce the following decision problems:

The *lower-bound feasibility problem*. An instance is given by integers  $m, n$ , an  $m \times n$  matrix  $A$ , vectors  $b$  and  $c$ , and a scalar  $\delta$ . The question is:  $\exists x \in \mathbb{Z}^n : Ax \leq b, cx > \delta$ ?

The *facet validity problem*. An instance is given by the same input as for the lower-bound feasibility problem. The question is: Does  $cx \leq \delta$  define a facet of  $\text{conv}(\{x \in \mathbb{Z}^n : Ax \leq b\})$ ?

Note that if the lower-bound feasibility problem for a family of polyhedra is NP-complete then optimizing over the same family of polyhedra is NP-hard.

**Lemma 5** *If any NP-complete problem belongs to co-NP, then  $NP=co-NP$ .*

**Theorem 6** Karp and Papadimitriou (1980). *If lower-bound feasibility is NP-complete, and facet validity belongs to NP then  $NP=co-NP$ .*

The way to prove Theorem 6 is to show that if facet validity belongs to NP, then lower-bound feasibility belongs to co-NP. If lower-bound feasibility is NP-complete we can through Lemma 5 conclude that  $NP=co-NP$ . It is extremely unlikely that  $NP=co-NP$ , as this implies that all NP-complete problems have a compact certificate for the no-answer. Hence, if we believe that  $NP \neq co-NP$ , and if  $\min\{cx : x \in S\}$  is NP-hard, then there are classes of facets of  $\text{conv}(S)$  for which there is no short proof that they are facets.

### 1.3 Equivalence Between Optimization and Separation

We have seen that if a problem is NP-hard we cannot expect to have a concise linear description of the convex hull of feasible solutions. Moreover, for the matching problem, which is polynomially solvable and which has a concise linear description of the convex hull of feasible solutions, this description is exponential in the dimension of the problem. These observations do not necessarily have to be negative since what we primarily need is a good description of the area around the optimal solution. The question then is whether it is possible to identify a violated inequality whenever needed, i.e. if we can find a hyperplane separating a given fractional solution from the convex hull, or prove that no such hyperplane exists.

The *separation problem for a family  $FP$  of polyhedra*. Given a polyhedron  $P \in FP$ , and a solution  $x^*$ , find an inequality  $cx \leq \delta$ , valid for  $P$ , satisfying  $cx^* > \delta$ , or prove that  $x^* \in P$ .

The *optimization problem for a family  $FP$  of polyhedra*. Given is a polyhedron  $P \in FP$ . Assume that  $P \neq \emptyset$  and that  $P$  is bounded. Given a vector  $c \in \mathbb{R}^n$ , find a solution  $x^0$  such that  $cx^0 \leq cx$  for all  $x \in P$ .

**Theorem 7** Grötschel, Lovász and Schrijver (1981). *There exists a polynomial time algorithm for the separation problem for a family  $FP$  of polyhedra, if and only if there exists a polynomial time algorithm for the optimization problem for  $FP$ .*

The theorem says that separation in general is equally hard as optimization but, as we shall see in the next section, when applying the polyhedral approach we develop *specific* families of valid inequalities for a given problem type, such as the odd-set constraints (10) developed for the matching problem.

The *separation problem based on a family  $FI$  of valid inequalities*. Given a solution  $x^*$ , find an inequality  $cx \leq \delta$  belonging to  $FI$ , satisfying  $cx^* > \delta$ , or prove that no such inequality in  $FI$  exists.

The separation problem based on a family of valid inequalities may be polynomially solvable even if the underlying optimization problem is NP-hard. Moreover, even if a family of inequalities is NP-hard to separate we may still be able to separate it effectively using a heuristic. Good separation heuristics together with a good implementation of a preprocessing routine and a branch-and-bound scheme, form the basis for the success of the polyhedral approach.

## 2 Polyhedral Results for Selected Combinatorial Structures

The results presented in the previous section did provide very important theoretical answers, but no efficient computational tools. In the early seventies there was a renewed interest in developing general purpose integer programming solvers. Instead of Gomory's cutting plane method, which tended to be very time consuming, one developed facet defining inequalities and separation algorithms for various problem types and embedded the separation algorithms in a branch-and-bound framework. **B&C flowchart here?** Since the added inequalities could be proved to be necessary to describe the convex hull of feasible solutions one could expect that they would be more effective than the Gomory cutting planes. Moreover, by developing facet defining inequalities and associated separation algorithms for some basic combinatorial structures that occur frequently in more general combinatorial optimization problems, and by implementing these algorithms in commercial software, it would possibly be very useful when solving a wide range of combinatorial problems. In the late seventies and in the eighties remarkable computational progress was made. Here we shall describe some classes of facet defining valid inequalities developed for a few basic, important, combinatorial optimization problems. The main purpose is to give an impression of how inequalities and separation algorithms are developed, and how they can be used, not only for the problem for which they are developed, but also for more general structures. We conclude the section by giving a list

of different problem types for which polyhedral results are known, together with references. Since the space provided here is not enough for a complete survey, we recommend the following literature to the interested reader. The books by Schrijver (1986), and Nemhauser and Wolsey (1988) provide a broad theoretical foundation as well as many examples. The article by Jünger et al. (1994) contains a comprehensive survey of computational results obtained by using polyhedral techniques. The latest developments on solving large traveling salesman problems is found in the article by Applegate et al. (1994).

## 2.1 Preliminaries

Here we introduce basic definitions that are needed to understand the terminology used in subsequent sections.

The set of *linear combinations* of a set of vectors  $x^1 \dots x^K \subset \mathbb{R}^n$  is the *linear space*  $LS = \{\sum_{k=1}^K \alpha_k x^k : \alpha \in \mathbb{R}^K\}$ . If  $x^1 \dots x^K$  form a minimal system, i.e., none of the vectors is a linear combination of the others, then the vectors  $x^1 \dots x^K$  are called *linearly independent*. Equivalently, the vectors  $x^1 \dots x^K$  are linearly independent if  $\alpha_k = 0, \forall k$  is the unique solution to the system  $\sum_{k=1}^K \alpha_k x^k = 0$ . The dimension of a linear space  $LS$ , denoted by  $\dim(LS)$  is defined as the minimum number of linearly independent points in the space.

The set of *affine combinations* of the  $K + 1$  points  $x^0, x^1 \dots x^K \subset \mathbb{R}^n$  is called an *affine space*  $AS = \{\sum_{k=0}^K \alpha_k x^k : \alpha \in \mathbb{R}^{K+1}; \sum_{k=0}^K \alpha_k = 1\}$ . Thus, an affine space can be viewed as a linear space translated over a vector  $x^0$ :  $AS = \{x^0 + \sum_{k=1}^K \beta_k (x^k - x^0) : \beta \in \mathbb{R}^K\}$ . Hyperplanes in  $\mathbb{R}^n$  are affine spaces. If the set of points  $x^0 \dots x^K$  is a minimal system, i.e., none of the points is an affine combination of the others, then the points  $x^0 \dots x^K$  are called *affinely independent*. Equivalently, the points  $x^0 \dots x^K$  are affinely independent if  $\alpha_k = 0, \forall k$  is the unique solution to the system  $\sum_{k=0}^K \alpha_k x^k = 0; \sum_{k=0}^K \alpha_k = 0$ . The *dimension* of an affine space, denoted by  $\dim(AS)$ , is the number of affinely independent points minus 1. Thus, if the points  $x^0 \dots x^K$  are affinely independent, the affine space defined by these points has dimension  $K$ .

A *polyhedron*  $P$  is the set of points satisfying a system of finitely many linear inequalities, i.e.,  $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ . The dimension of  $P$ , denoted  $\dim(P)$ , is the dimension of the smallest affine space containing  $P$ . A bounded polyhedron is called a *polytope*.

An inequality  $\pi x \leq \pi_0$  is called *valid* for  $P$  if each point in  $P$  satisfies the inequality. The set  $F = \{x \in P : \pi x = \pi_0\}$  is called a *face* of  $P$  and the valid inequality  $\pi x \leq \pi_0$  is said to *define* the face  $F$ . A face is said to be *proper* if it is not empty and if it is properly contained in  $P$ , i.e. if  $\emptyset \neq F \neq P$ . The dimension of a proper face  $F$ ,  $\dim(F)$ , is strictly smaller than the dimension of  $P$ . If  $\dim(F) = \dim(P) - 1$ , i.e., if  $F$  is maximal, then  $F$  is called a *facet*. The facet defining inequalities are important since they are precisely the inequalities needed to define the convex hull of feasible solution in addition to the set of inequalities that are satisfied with equality by every feasible point.

## 2.2 The Vertex Packing Problem

maybe skip A *vertex packing* is a subset  $V' \subseteq V$  of vertices in an undirected graph  $G = (V, E)$ , such that no two vertices in  $V'$  are adjacent. Let  $x_v = 1$  if  $v \in V'$  and let  $x_v = 0$  otherwise. The integer programming formulation of the maximum cardinality vertex packing problem is given below.

$$\max \sum_{v \in V} x_v \quad (11)$$

$$\text{s.t.} \quad x_v + x_w \leq 1 \quad \text{for all } \{v, w\} \in E \quad (12)$$

$$x_v \in \{0, 1\} \quad \text{for all } v \in V \quad (13)$$

**possible application** The vertex packing problem is sometimes referred to as the *independent set problem* or as the *stable set problem*. Let  $X_{VPG}$  be the set of feasible solutions to the vertex packing problem in the graph  $G$  and let  $\alpha(G)$  be the maximum cardinality of a vertex packing in  $G$ . An edge is called *critical* if its removal from  $G$  produces a graph  $G'$  with  $\alpha(G') > \alpha(G)$ . Chvátal (1975) derived the following general sufficient condition for an inequality to define a facet of  $\text{conv}(X_{VPG})$ .

**Theorem 8** Chvátal (1975). *Let  $E^*$  be the set of critical edges of  $G$ . If the graph  $G^* = (V, E^*)$  is connected, then the inequality  $\sum_{j \in V} x_j \leq \alpha(G)$  defines a facet of  $\text{conv}(X_{VPG})$ .*

A *clique* in a graph  $G$  is a complete subgraph of  $G$ , see Figure 2a). Since no two vertices

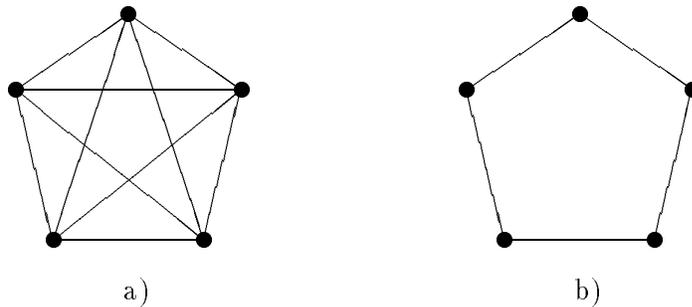


Figure 2: a) A clique. b) An odd-hole.

in  $V'$  are allowed to be adjacent we could take any clique  $C$  in  $G$  and require that at most one vertex belonging to  $C$  should belong to the vertex packing  $V'$  giving the valid inequality

$$\sum_{j \in C} x_j \leq 1. \quad (14)$$

**Theorem 9** Padberg (1973). *Let  $C$  be a clique in the graph  $G$ . The inequality (14) defines a facet of  $\text{conv}(X_{VPG})$  if and only if  $C$  is maximal.*

*Proof. Sufficiency:* The dimension of the vertex packing polytope is  $|V|$ . Hence, to prove that (14) defines a facet of  $\text{conv}(X_{VPG})$  we need to find  $|V|$  affinely independent points that are tight for (14). Let  $C$  be a maximal clique. For every  $v \in C$  we take the vertex packing that contains only  $v$ . For  $v \notin C$  we first choose a node  $w \in C$  that is not adjacent to  $v$ . Since  $C$  is maximal such a node exists. We then take the vertex packing that contains both nodes  $v$  and  $w$ . The  $|V|$  points given above are feasible and satisfy the clique inequality with equality. Thus, the inequality is facet-defining.

*Necessity:* If  $C$  is not maximal then there is a clique  $C'$  such that  $C \subset C'$ . The clique inequality defined by  $C'$  dominates the inequality defined by  $C$ . ■

Another class of valid inequalities for the vertex packing problem is the family of *odd-hole inequalities*. An odd hole  $H$  in a graph  $G$  is a chordless cycle consisting of an odd number of vertices, i.e. there are no edges of  $G$  connecting any nonconsecutive vertices in  $H$ , see Figure 2b). Since the number of vertices in  $H$  is odd, at most  $\lfloor |H|/2 \rfloor = (|H| - 1)/2$  vertices in  $H$  can belong to any vertex packing. Hence the following odd-hole inequality is valid,

$$\sum_{j \in H} x_j \leq \frac{|H| - 1}{2}. \quad (15)$$

Padberg showed that (15) defines a facet of  $\text{conv}(X_{VPG} \cap \{x_j = 0 \text{ for all } j \notin H\})$ , i.e. in general (15) defines a face of  $\text{conv}(X_{VPG})$  of dimension less than  $\dim(X_{VPG}) - 1$ . The question is whether it is possible to increase the dimension of (15) such that (15) becomes a facet for  $\text{conv}(X_{VPG})$ . One way of increasing the dimension of a face is through *sequential lifting* (Padberg (1973) and Wolsey (1976)), which is illustrated in the following example. Consider

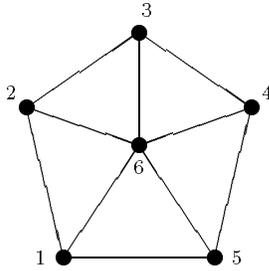


Figure 3: A wheel.

the graph in Figure 3. Such a graph is called a *wheel*. The inequality  $x_1 + x_2 + x_3 + x_4 + x_5 \leq 2$  defines a facet of  $\text{conv}(X_{VPG} \cap \{x_6 = 0\})$ . The question is whether there exists a constant  $\alpha \geq 0$  such that  $x_1 + x_2 + x_3 + x_4 + x_5 + \alpha x_6 \leq 2$  defines a facet of  $\text{conv}(X_{VPG})$ . If  $x_6 = 0$ ,  $\alpha$  can take any value, hence assume that  $x_6 = 1$ . If  $x_6 = 1$  we must have  $x_j = 0$ ,  $j = 1, \dots, 5$ , since  $x_6$  is adjacent to all other vertices. The maximal value of  $\alpha$ , such that the inequality remains valid, is  $\alpha = 2$ . In this example we had only one variable set to a fixed value, but in general we include one variable at the time, with a nonnegative coefficient, in the inequality. The following two theorems imply that if the inequality is facet defining in the reduced space, and if we “lift” in all variables sequentially with maximal coefficients, then the resulting inequality defines a facet in the full space.

**Theorem 10** Wolsey (1976). Let  $S \subseteq \{0, 1\}^n$ . Suppose

$$\sum_{j=2}^n \pi_j x_j \leq \pi_0 \quad (16)$$

is valid for  $S^0 = S \cap \{x \in \{0, 1\}^n : x_1 = 0\}$ . If  $S \cap \{x \in \{0, 1\}^n : x_1 = 1\} \neq \emptyset$ , then

$$\alpha x_1 + \sum_{j=2}^n \pi_j x_j \leq \pi_0 \quad (17)$$

is valid for  $S$  for any  $\alpha \leq \pi_0 - \max_{S \cap \{x: x_1=1\}} \{\sum_{j=2}^n \pi_j x_j\}$ . If (16) defines a face of  $\text{conv}(S^0)$  of dimension  $k$ , and if  $\alpha$  is chosen maximal, then (17) defines a face of  $\text{conv}(S)$  of dimension  $k + 1$ .

**Theorem 11** Wolsey (1976). Let  $S \subseteq \{0, 1\}^n$ . Suppose

$$\sum_{j=2}^n \pi_j x_j \leq \pi_0 \quad (18)$$

is valid for  $S^1 = S \cap \{x \in \{0, 1\}^n : x_1 = 1\}$ . If  $S \cap \{x \in \{0, 1\}^n : x_1 = 0\} \neq \emptyset$ , then

$$\beta x_1 + \sum_{j=2}^n \pi_j x_j \leq \pi_0 + \beta_1 \quad (19)$$

is valid for  $S$  for any  $\beta \geq \max_{S \cap \{x: x_1=0\}} \sum_{j=2}^n \pi_j x_j - \pi_0$ . If (18) defines a face of  $\text{conv}(S^1)$  of dimension  $k$ , and if  $\beta$  is chosen minimal, then (19) defines a face of  $\text{conv}(S)$  of dimension  $k + 1$ .

Sequential lifting is sequence dependent, i.e. different lifting sequences give rise to different inequalities. Zemel (1978) proposed an alternative lifting procedure, called *simultaneous lifting*. As the name indicates, the coefficients of all variables that are to be lifted are considered simultaneously, yielding inequalities that cannot be obtained in general by sequential lifting. For more details on lifting procedures, see Nemhauser and Wolsey (1988).

The separation problem for clique inequalities consists of finding a maximum weight clique in a graph. This problem is NP-hard, and therefore we usually turn to heuristics for finding violated clique inequalities. The separation problem for odd-hole inequalities can be solved in polynomial time by applying a shortest path algorithm to a slightly adapted graph, see Hoffman and Padberg (1993).

### 2.3 The Traveling Salesman Problem

Consider an undirected complete graph  $G = (V, E)$  with  $n = |V|$ . In the traveling salesman problem (TSP) we want to find a minimum length *Hamiltonian cycle*, i.e. a minimum length cycle containing each vertex exactly once. Let  $x_e = 1$  if edge  $e$  belongs to the Hamiltonian

cycle, and let  $x_e = 0$  otherwise. Moreover, let  $d_e$  denote the length of edge  $e \in E$ . Usually, the vertices of the graph are called *cities*, and the Hamiltonian cycle is called a *tour*.

$$\min \sum_{e \in E} d_e x_e \tag{20}$$

$$\text{s. t.} \quad \sum_{e: v \in e} x_e = 2 \quad \text{for all } v \in V \tag{21}$$

$$\sum_{e \subset S} x_e \leq |S| - 1 \quad \text{for all } S : \emptyset \neq S \neq V \tag{22}$$

$$x_e \in \{0, 1\} \quad \text{for all } e \in E \tag{23}$$

The formulation restricted to the constraints (21) and (23) is called the *2-matching relaxation* of TSP and its solutions are referred to as *2-matchings*. Such solutions may constitute disjoint cycles, or *subtours*. Constraints (22), introduced by Dantzig et al. (1954), prevent subtours, and are therefore called *subtour elimination constraints*. Edmonds (1965) studied the polyhedral structure of the 2-matching problem, and obtained a complete linear description of the convex hull of feasible solutions by adding so-called *2-matching inequalities* to constraints (21) and  $0 \leq x_e \leq 1$  for all  $e \in E$ . Since the 2-matching problem is a relaxation of TSP, the 2-matching inequalities are also valid for TSP. We illustrate these inequalities in the following example. Consider the fractional solution illustrated in Figure 4. The thick

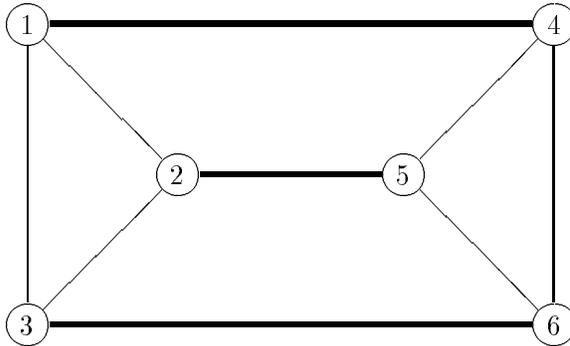


Figure 4: A fractional solution violating a 2-matching constraint.

lines correspond to variables that have value 1 and the thin lines correspond to variables with value 0.5. Clearly, this solution satisfies the degree constraints (21). To separate this solution from the convex hull of 2-matchings we introduce the following inequality. Consider the set of vertices  $H = \{1, 2, 3\}$ . Let  $E(H)$  be the set of edges with both endvertices in  $H$ , and let  $E' = \{\{1, 4\}, \{2, 5\}, \{3, 6\}\}$ , i.e., each edge in  $E'$  has exactly one endvertex in  $H$ . Furthermore, let  $x(F) = \sum_{e \in F} x_e$ . From the set of edges  $E(H) \cup E'$  at most four can belong to a 2-matching since otherwise at least one of the vertices in  $H$  will have degree 3, which violates constraints (21). The cumulative value of the variables corresponding to edges in  $E(H) \cup E'$  is 4.5. Hence,

we can conclude that the inequality  $x(E(H)) + x(E') \leq 4$  is violated by the solution described above. In general, a 2-matching constraint has the form

$$x(E(H)) + x(E') \leq |H| + \left\lfloor \frac{1}{2}|E'| \right\rfloor$$

where  $H \subset V$  and where the edges in  $E'$  have precisely one endvertex in  $H$ . Note that only 2-matching constraints with an odd number of edges in  $E'$ , can be facet-defining, since the inequalities otherwise are implied by the degree constraints.

*Comb inequalities* were introduced by Chvátal (1975) as a generalization of the 2-matching constraints. In the comb inequalities the edges in  $E'$  are replaced by an odd number  $s$  of disjoint vertex sets  $T_1, \dots, T_s$ , called teeth, each having one vertex in common with the handle  $H$ . The comb inequality is written as

$$x(E(H)) + \sum_{j=1}^s x(E(T_j)) \leq |H| + \sum_{j=1}^s (|T_j| - 1) - \frac{1}{2}(s + 1). \quad (24)$$

The fractional solution illustrated in Figure 5 satisfies the 2-matching constraints and the subtour elimination constraints, but not the comb inequality defined by  $H = \{1, 5, 6, 7\}$ , and  $T_1 = \{1, 2\}, T_2 = \{3, 4, 5, 6\}, T_3 = \{7, 8\}$ . Again, thick lines correspond to variables with value 1 and thin lines to variables with value 0.5. Chvátal's comb inequalities were generalized by

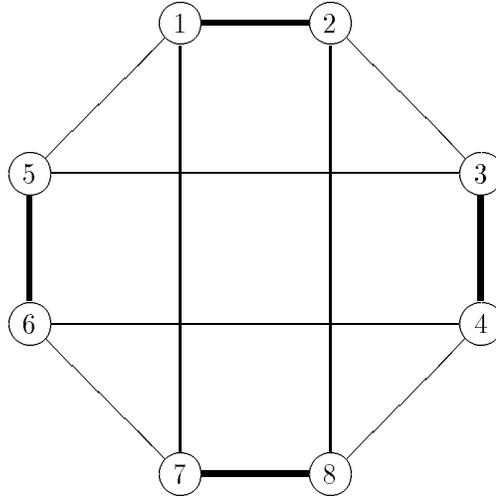


Figure 5: Fractional solution violating a comb inequality.

Grötschel and Padberg (1979) who introduced structures where each tooth can have more than one vertex in common with the handle. The *clique tree inequalities* introduced by Grötschel and Pulleyblank (1986) are further generalization of combs in the sense that clique trees contain multiple handles, which are connected through the teeth. Many more exotic classes of inequalities have been derived to date, but the search for new classes is still vivid. A good overview of the current state-of-the-art is provided by Applegate et al. (1994). Goemans (1993) considers the quality of various classes of inequalities with respect to their induced relaxations.

The separation problem based on the subtour elimination constraints can be viewed as a minimum cut problem, which is polynomially solvable using max-flow algorithms. Separation of the 2-matching constraints is also polynomial, which was showed by Padberg and Grötschel (1985). Violated 2-matching constraints are however usually identified using a heuristic, since this is still effective and faster in practice. No exact polynomial time algorithm is known to date for solving the separation problem based on the comb inequalities, but there are fast heuristic methods available that perform quite well. For clique tree inequalities, not even good heuristics are known that will perform well in general. To illustrate the effectiveness of the polyhedral approach to solve TSP we provide detailed computational results in sections 3 and 4.

## 2.4 The Knapsack Problem

Let  $N = \{1, \dots, n\}$ . The knapsack problem is formulated as

$$\max \sum_{j \in N} c_j x_j \quad (25)$$

$$\text{s.t.} \quad \sum_{j \in N} a_j x_j \leq b \quad (26)$$

$$x_j \in \{0, 1\} \quad \text{for all } j \in N. \quad (27)$$

Assume that the vectors  $c$ ,  $a$  and the right-hand side  $b$  are rational, and let  $X_K$  denote the set of feasible solutions to the knapsack problem. We call a set  $C$  a *cover* or a *dependent set* with respect to  $N$  if  $\sum_{j \in C} a_j > b$ . A cover is *minimal* if  $\sum_{j \in S} a_j \leq b$  for all  $S \subset C$ . If we choose all elements from the cover  $C$ , it is clear that the right-hand side of (26) is exceeded. Hence, the following *knapsack cover inequality* (Balas (1975), Hammer et al. (1975) and Wolsey (1975))

$$\sum_{j \in C} x_j \leq |C| - 1 \quad (28)$$

is valid. A generalization of (28) is given by the family of  $(1, k)$ -*configuration inequalities* (Padberg (1980)). Let  $\bar{C} \subseteq N$ , and  $t \in N \setminus \bar{C}$  be such that  $\sum_{j \in \bar{C}} a_j \leq b$  and such that  $Q \cup \{t\}$  is a minimal cover for all  $Q \subseteq \bar{C}$  with  $|Q| = k$ . Let  $T(r) \subseteq \bar{C}$  vary over all subsets of cardinality  $r$  of  $\bar{C}$ , where  $r$  is an integer satisfying  $k \leq r \leq |\bar{C}|$ . The  $(1, k)$ -configuration inequality

$$(r - k + 1)x_t + \sum_{j \in T(r)} x_j \leq r \quad (29)$$

is valid for  $\text{conv}(X_K)$ , and if  $k = |\bar{C}|$  the cover inequalities (28) are obtained. The  $(1, k)$ -configuration inequalities are primarily designed to deal with elements  $j$  of the knapsack having a large coefficient  $a_j$ .

In general (28) is not facet defining, but as with the odd-hole inequalities (15) they can be lifted to become facets. One special case of a lifted cover inequality, where all lifting coefficients are equal to zero or one, is obtained by considering the *extension*  $E(C)$  of a minimal cover  $C$ ,

where  $E(C) = \{k \in N \setminus C : a_k \geq a_j, \text{ for all } j \in C\}$ . The inequality  $\sum_{j \in E(C)} x_j \leq |C| - 1$  is valid for  $\text{conv}(X_K)$  and under certain conditions it also defines a facet of  $\text{conv}(X_K)$ . The most general form of the knapsack cover inequality is obtained by partitioning the set  $N$  into the sets  $(N', N \setminus N')$ . Let  $x_j = 0$  for all  $j \in N \setminus N'$ , and let  $C'$  be a minimal cover with respect to  $N'$ . Moreover, let  $x_j = 1$  for all  $j \in N' \setminus C'$ . By using the lifting results presented in Theorems 10 and 11, we can conclude that  $\text{conv}(X_K)$  has a facet of the following form

$$\sum_{j \in N \setminus N'} \alpha_j x_j + \sum_{j \in N' \setminus C'} \beta_j x_j + \sum_{j \in C'} x_j \leq |C'| - 1 + \sum_{j \in N' \setminus C'} \beta_j, \quad (30)$$

where  $\alpha_j \geq 0$  for all  $j \in N \setminus N'$  and  $\beta_j \geq 0$  for all  $j \in N' \setminus C'$ . Balas (1975) characterized the lifting coefficients  $\alpha_j$  in the case where  $N' \setminus C' = \emptyset$ .

The separation problem based on the cover inequalities can again be viewed as a knapsack problem as we show below. Assume we are given the point  $x^*$ . To find a cover inequality (28) violated by  $x^*$  we need to find a set  $C$  such that  $\sum_{j \in C} x_j^* > |C| - 1$  and  $\sum_{j \in C} a_j > b$ . Let  $z_j = 1$  if  $j \in C$ , and let  $z_j = 0$  otherwise and assume without loss of generality that  $a_j, j \in N$  and  $b$  are integral. For (28) to be violated the  $z_j$ -variables have to satisfy the constraints

$$\sum_{j \in N} x_j^* z_j > \left( \sum_{j \in N} z_j \right) - 1 \quad \text{and} \quad \sum_{j \in N} a_j z_j \geq b + 1.$$

The first of the above constraints can be rewritten as  $\sum_{j \in N} (1 - x_j^*) z_j < 1$ , leading to the following formulation of the problem of finding the most violated cover inequality (28).

$$\min \quad \eta = \sum_{j \in N} (1 - x_j^*) z_j \quad (31)$$

$$\text{s.t.} \quad \sum_{j \in N} a_j z_j \geq b + 1 \quad (32)$$

$$z_j \in \{0, 1\} \quad \text{for all } j \in N \quad (33)$$

A violated cover inequality is identified if and only if  $\eta < 1$ . To see that the separation problem (31)-(33) is equivalent to a knapsack problem we only need to complement the  $z_j$ -variables, i.e. substitute  $z_j$  by  $1 - z_j$ . Problem (31)-(33) is however often easier to solve than the original knapsack problem since, at a typical fractional solution  $x^*$ , many variables take value zero or one. If  $x_j^* = 1$  the coefficient of  $z_j$  in (31) is equal to zero and we can set  $z_j$  equal to one. Analogously, if  $x_j^* = 0$  we set  $z_j$  is equal to zero. Therefore, typically few variables remain in the separation problem. Crowder et al. (1983) developed a heuristic for the separation problem and for choosing the sets  $N'$  and  $C'$ . Once a minimal cover  $C'$  is generated it is also used in a heuristic for finding a violated  $(1, k)$ -configuration inequality. They implemented the algorithms and solved large 0-1 integer programming problems by automatically generating knapsack cover inequalities. Recent work on the knapsack polytope is done by Weismantel (1994).

## 2.5 The Single-Node Flow Problem

Consider a single node in a directed graph, and let  $N$  be the set of arcs entering the node. The outflow from the node is equal to  $b$ . Let  $x_j$  be a continuous variable denoting the flow on arc  $j$ , and let  $m_j$  be the capacity of arc  $j$ . If arc  $j$  is open, then  $y_j = 1$ , otherwise  $y_j = 0$ . The following fixed charge single-node flow structure is a relaxation of many combinatorial flow models,

$$\sum_{j \in N} x_j = b \quad (34)$$

$$0 \leq x_j \leq m_j y_j \quad \text{for all } j \in N \quad (35)$$

$$y_j \in \{0, 1\} \quad \text{for all } j \in N. \quad (36)$$

Let  $X_{FC}$  denote the set of feasible solutions to (34)–(36). A subset  $J \subseteq N$  is called a *flow cover* if  $\sum_{j \in J} m_j = b + \lambda$  with  $\lambda > 0$ . If we have a cover  $J$  and if we close one arc  $k \in J$  then  $\max\{x_j : j \in J \setminus k\} = \min\{b, \sum_{j \in J \setminus k} m_j\} = \min\{b, b - (m_k - \lambda)\} = b - (m_k - \lambda)^+$  yielding the valid inequality

$$\sum_{j \in J} x_j \leq b - \sum_{j \in J} (m_j - \lambda)^+ (1 - y_j). \quad (37)$$

**Theorem 12** Padberg, Van Roy and Wolsey (1985). *The flow cover inequality (37) defines a facet of  $\text{conv}(X_{FC})$  if and only if  $\max_{j \in J} \{m_j\} > \lambda$ .*

Let  $z_j = 1$  if  $j \in J$  and let  $z_j = 0$  otherwise, and let  $(x^*, y^*)$  denote a fractional point. For a given value of  $\lambda$ , the separation problem based on the family of flow cover inequalities (37) is formulated as follows.

$$\max \sum_{j \in N} [x_j^* + (m_j - \lambda)^+ (1 - y_j^*)] z_j \quad (38)$$

$$\text{s.t.} \quad \sum_{j \in N} m_j z_j = b + \lambda \quad (39)$$

$$z_j \in \{0, 1\} \quad \text{for all } j \in N. \quad (40)$$

Once we have a set  $J$  satisfying the condition of Theorem 12 we can extend the flow cover inequality by including flow from the arcs belonging to the set  $L \subseteq (N \setminus J)$ .

Let  $\bar{m}_l = \max\{\max_{j \in J} \{m_j\}, m_l\}$  for all  $l \in L$ . The following *extended flow cover inequality* is valid for  $\text{conv}(X_{FC})$ ,

$$\sum_{j \in J \cup L} x_j \leq b - \sum_{j \in J} (m_j - \lambda)^+ (1 - y_j) + \sum_{j \in L} (\bar{m}_j - \lambda) y_j. \quad (41)$$

Padberg et al. (1985) gave sufficient conditions for the extended flow cover inequality to define a facet of  $\text{conv}(X_{FC})$ . Aardal et al. (1993) showed that the separation problem based on the family of extended flow cover inequalities can be solved in polynomial time if  $m_j = m$  for all  $j \in N$ .

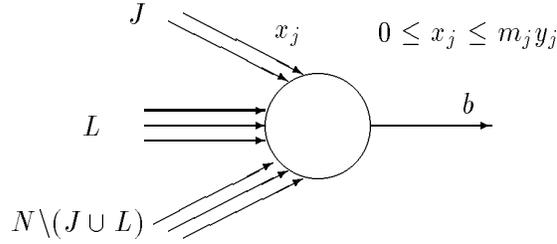


Figure 6: Single-node fixed charge flow.

Van Roy and Wolsey (1986) also studied the single-node flow model with both fixed charge inflow and outflow arcs as well as general uncapacitated fixed charge structures, for which they developed various families of facet defining valid inequalities. Separation heuristics for these inequalities are also discussed by Van Roy and Wolsey (1987).

## 2.6 An Application: The Facility Location Problem

Here we shall discuss how some of the inequalities presented above can be used, and extended, to solve facility location problems. The facility location problem is defined as follows. Let  $M = \{1, \dots, m\}$  be the set of facilities, and let  $N = \{1, \dots, n\}$  be the set of clients. Facility  $j$  has capacity  $m_j$ , and client  $k$  has demand  $d_k$ . The total demand of the clients in the set  $S \subseteq N$  is denoted by  $d(S)$ . The fixed cost of opening facility  $j$  is equal to  $f_j$  and the cost of transporting one unit of goods from facility  $j$  to client  $k$  is equal to  $c_{jk}$ . Let  $y_j = 1$  if facility  $j$  is open and let  $y_j = 0$  otherwise. The flow from facility  $j$  to client  $k$  is denoted by  $v_{jk}$ . We want to determine which facility should be opened and how the flow should be distributed between the open facilities and the clients such that the sum of the fixed costs of opening the facilities, and the transportation costs is minimized, and such that all clients are served, and all capacity restrictions are satisfied. The mathematical formulation is given below.

$$\min \sum_{j \in M} f_j y_j + \sum_{j \in M} \sum_{k \in N} c_{jk} v_{jk} \quad (42)$$

$$\text{s.t.} \quad \sum_{j \in M} v_{jk} = d_k \quad \text{for all } k \in N \quad (43)$$

$$\sum_{k \in N} v_{jk} \leq m_j y_j \quad \text{for all } j \in M \quad (44)$$

$$0 \leq v_{jk} \leq d_k y_j \quad \text{for all } j \in M, k \in N \quad (45)$$

$$y_j \in \{0, 1\} \quad \text{for all } j \in M \quad (46)$$

### 2.6.1 The Uncapacitated Case

In the uncapacitated facility location (UFL) problem we have  $m_j = d(N)$  for all  $j \in M$ . It is convenient to scale the flow by substituting the variables  $v_{jk}$  by the variables  $x_{jk} = v_{jk}/d_k$ .

The set of feasible solutions to UFL,  $X_{UFL}$ , is given by the following sets of constraints.

$$\sum_{j \in M} x_{jk} = 1 \quad \text{for all } k \in N \quad (47)$$

$$0 \leq x_{jk} \leq y_j \quad \text{for all } j \in M, k \in N \quad (48)$$

$$y_j \in \{0, 1\} \quad \text{for all } j \in M \quad (49)$$

It is possible to require explicitly that  $x_{jk} \in \{0, 1\}$  since there is at least one optimal solution of UFL having this property. Moreover, we can change the equality sign in constraint set (47) to a less-than-or-equal-to sign if we make an appropriate change in the objective function (for more details see Cho et al. (1983)). Finally, by complementing the  $y_j$ -variables, i.e. by introducing  $y'_j = 1 - y_j$ , we obtain the following *vertex packing formulation* of UFL.

$$\sum_{j \in M} x_{jk} \leq 1 \quad \text{for all } k \in N \quad (50)$$

$$x_{jk} + y'_j \leq 1 \quad \text{for all } j \in M, k \in N \quad (51)$$

$$y'_j, x_{jk} \in \{0, 1\} \quad \text{for all } j \in M, k \in N \quad (52)$$

Let  $X_{UFLVP}$  be the set of feasible solutions to (50)–(52). Given a vertex packing formulation of UFL, we can construct an associated undirected graph, called *the intersection graph* by introducing a vertex for every variable and an edge for every pair of nonorthogonal columns. To determine  $\text{conv}(X_{UFLVP})$  is equivalent to determining the convex hull of vertex packings in the associated intersection graph. Hence, we can use all results described in Section 2.2 to derive valid inequalities for UFL. Due to the construction of the intersection graph all cliques in this graph are described by inequalities (50) and (51), and all odd holes are cycles where every third vertex is a  $y'_j$ -vertex. Both Cornuéjols and Thizy (1982) and Cho et al. (1983) used the result by Chvátal given in Theorem 8 to find more general inequalities than the odd-hole inequalities. All these inequalities have a regular cyclic structure and all coefficients are equal to one for all variables except one example of a simultaneously lifted odd-hole inequality given by Cornuéjols and Thizy. Aardal and Van Hoesel (1995) discuss further use of simultaneous lifting to get new facets having different coefficients.

## 2.6.2 The Capacitated Case

By aggregating the flow from each depot we can easily see that a version of the knapsack as well as the single node flow structure form relaxations of the capacitated facility location (CFL) problem. Let  $v_j = \sum_{k \in N} v_{jk}$ . By using the aggregate flow variable  $v_j$  we can obtain the aggregate capacity and demand constraints

$$0 \leq v_j \leq m_j y_j \quad \text{for all } j \in M \quad (53)$$

$$\sum_{j \in M} v_j = d(N). \quad (54)$$

If we combine constraints (53) and (54) with constraint (46) we obtain a single-node flow polytope and a so-called surrogate knapsack polytope  $X_{SK} = \{y \in \{0, 1\} : \sum_{j \in M} m_j y_j \geq d(N)\}$ . Complementing the  $y_j$ -variables, i.e. letting  $y'_j = 1 - y_j$  for all  $j \in M$  gives the knapsack polytope  $\{y' \in \{0, 1\} : \sum_{j \in M} m_j y'_j \leq \sum_{j \in M} m_j - d(N)\}$ . Hence we can use both the knapsack cover inequalities as well as the flow cover inequalities when solving CFL. Both classes of inequalities can also be derived for subsets  $K \subseteq N$  of clients. Especially the cover inequalities have proved very useful computationally, as is illustrated further in Section 3.1.3. One way of generalizing the flow cover inequalities is by considering a subset of clients as well as subsets of arcs yielding the family of *effective capacity inequalities* (Aardal et al. (1993)). Let  $K_j \subseteq K$  for all  $j \in M$  and let  $\bar{m}_j = \min\{m_j, d(K_j)\}$ . Let  $J$  define a cover with respect to  $K$ , i.e.  $\sum_{j \in J} \bar{m}_j = d(K) + \lambda$  with  $\lambda > 0$ . The effective capacity (EC) inequality

$$\sum_{j \in J} \sum_{k \in K_j} v_{jk} \leq d(K) - \sum_{j \in J} (\bar{m}_j - \lambda)^+ (1 - y_j) \quad (55)$$

is valid for  $\text{conv}(X_{CFL})$ . The facet defining EC inequalities were completely characterized by Aardal et al. (1993). To further generalize the EC inequalities consider the function  $f(J)$  which is the maximum feasible flow from the depots in  $J$  to the clients in  $K$  on the arcs  $\{(j, k) : j \in J, k \in K_j\}$ . By using maximum flow arguments we can show that  $f(J) - f(J \setminus \{j\}) \geq (\bar{m}_j - \lambda)^+$ . Hence the valid inequality

$$\sum_{j \in J} \sum_{k \in K_j} v_{jk} \leq f(J) - \sum_{j \in J} (f(J) - f(J \setminus \{j\})) (1 - y_j) \quad (56)$$

is at least as strong as the EC inequality (55). Inequalities (56) are called *submodular* inequalities since the function  $f(J)$  is a submodular set function. Submodular inequalities were first considered by Wolsey (1989) and further developed for CFL by Aardal et al. (1993). Since there is no closed-form expression of  $f(J)$  in general, it is hard to characterize the submodular facets. Aardal et al. completely characterized two subclasses for which  $f(J) - f(J \setminus \{j\}) \geq (\bar{m}_j - \lambda)^+$  for at least one  $j \in J$ , namely the single-depot and the multi-depot inequalities. The separation problem based on the EC inequalities and the submodular inequalities are discussed by Aardal (1994).

## 2.7 A List of Polyhedral Results for Combinatorial Problems

Here we provide a list of polyhedral results that are known for combinatorial optimization problems. If a recent survey of results for a specific problem class is known, we refer to the survey and not to the individual articles. Surveys are marked with an asterisk. Due to the vast literature, we do not claim the list to be complete.

**Airline crew scheduling:** Hoffman and Padberg (1993). **Boolean quadratic polytope:** Padberg (1989), Lee and Leung (1993a). **Clique problems:** Pulleyblank and Shepherd (1993), Balas et al. (1994b). **Clustering:** Grötschel and Wakabayashi (1989). **Coloring:** Lee and Leung (1993b), Nemhauser and Park (1991). **Covering, packing and partition:**

Balas and Padberg (1972), Padberg (1973,1977,1980), Nemhauser and Trotter (1974), Trotter (1975), Wolsey (1976b), Balas and Zemel (1977), Balas and Ho (1980), Balas and Ng (1989a,b), Cornuéjols and Sassano (1989), Laurent (1989), Nobili and Sassano (1989), Sassano (1989), Grötschel and Wakabayashi (1990), Chopra and Rao (1993). **Cut polytopes:** Barahona and Mahjoub (1986), Barahona et al. (1988), Conforti et al. (1990/91a,b), De Sousa and Laurent (1991), Deza et al. (1992), Deza and Laurent (1992a,b). **Frequency assignment:** Aardal et al. (1995). **General integer and mixed 0-1 structures:** Wolsey (1976a), Peled (1977), Zemel (1978), Crowder et al. (1983), Padberg et al. (1985), Van Roy and Wolsey (1985,1986,1987), Goemans (1989), Nemhauser and Wolsey (1990). **Knapsack problems:** Balas (1975a), Hammer et al. (1975), Wolsey (1975), Balas and Zemel (1978), Padberg (1980), Nemhauser and Vance (1994), Weismantel (1994). **Layout design:** Leung (1994). **Linear ordering:** Grötschel et al. (1984,1985), Reinelt (1985), Mitchell and Borchers (1992,1993). **Location:** Cornuéjols et al. (1977), Cornuéjols and Thizy (1982), Cho et al. (1983a,b), Leung and Magnanti (1989), Aardal et al. (1993,1994), Aardal (1994), Aardal and Van Hoesel (1995). **Lot sizing:** Pochet and Wolsey (1994)\*. **Matching:** Edmonds (1965), Grötschel and Holland (1985). **Network and VLSI design:** Pochet and Wolsey (1992), Grötschel et al. (1992b,1993,1995). **Postman problems:** Grötschel and Win (1992). **Scheduling:** Queyranne and Schulz (1994)\*. **Subgraph polytopes:** Balas and Pulleyblank (1983), Barahona et al. (1985), Barahona and Mahjoub (1989,1992), Chopra (1992), Jünger and Mutzel (1993). **Tenary problems:** Chopra (1989a). **Traveling salesman problems:** Dantzig et al. (1954, 1959), Grötschel and Padberg (1979), Grötschel (1980), Padberg and Hong (1980), Cornuéjols and Pulleyblank (1982), Grötschel and Pulleyblank (1986), Padberg and Rinaldi (1987,1990,1991), Fischetti (1991a,1992), Grötschel and Holland (1991), Naddef and Rinaldi (1991,1992), Reinelt (1991), Naddef (1992), Clochard and Naddef (1993), Goemans (1993), Applegate et al. (1994), Balas et al. (1995). **Trees, forests and arborecences:** Gamble and Pulleyblank (1989), Chopra (1989b), Fischetti (1991b), Balas and Fischetti (1992), Chopra et al. (1992), Goemans (1992), Grötschel et al. (1992a), Hall and Magnanti (1992), Chopra and Rao (1994a,b). **Vehicle routing:** Araque (1989,1990), Araque et al. (1990), Cornuéjols and Harche (1993).

### 3 Computational Aspects

Once specific classes of valid inequalities for a certain version of ILP have been developed we can implement the separation algorithms for these inequalities in the following cutting plane algorithm.

#### Outline of the cutting plane algorithm.

1. Initialize the linear programming relaxation LP of ILP.
2. Solve LP and let  $x^*$  be the optimal solution. If  $x^*$  is integral, stop, otherwise go to step 3.
- 3.

3. Separation algorithms are run to identify inequalities violated by  $x^*$ . If one or more inequalities, or cuts, have been found add them to LP and go to step 2. If no violated inequality is found, stop.

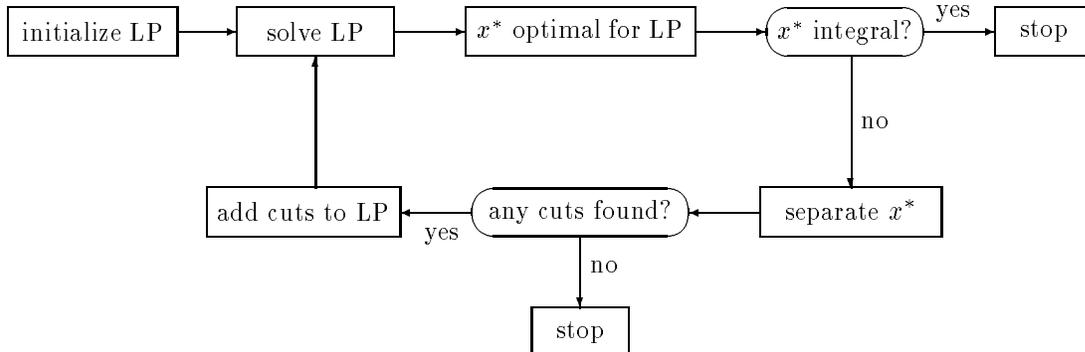


Figure 7: Basic cutting plane algorithm.

If the algorithm terminates by finding an integral solution  $x^*$ , then  $x^*$  is provably optimal. Otherwise, the final fractional solution provides a lower bound on the optimal value, if we assume that ILP is a minimization problem. Contrary to Gomory’s cutting plane algorithm we cannot guarantee that the algorithm terminates with the optimal solution to ILP since we in general consider only a subset of all classes of facet defining inequalities, and since the separation problems are often solved heuristically. Nevertheless, this technique has proved very effective for finding at least very strong lower bounds. A good lower bound decreases the expected size of a branch-and-bound tree if we need to obtain the optimal solution. To illustrate how the lower bound develops if we add valid inequalities sequentially, we consider a TSP instance of 120 cities from Grötschel (1980), which was solved to optimality after adding cutting planes only. The optimal solution was found after 13 calls to the LP-solver. The value of the LP relaxation,  $z_{LP}$ , and the number of added cuts at each iteration, are given in Table 1. In total 36 subtour elimination constraints, 25 2-matching constraints, and 35 comb constraints were added. As can be concluded from the table, it is good practice to generate and add many violated inequalities at each iteration, since in general the computing time needed to solve the linear programs increase modestly, but the lower bound converges to the optimal value much quicker.

In the remainder of this section we shall discuss how the basic cutting plane algorithm can be extended and embedded in a branch-and-bound framework. We also discuss several implementation issues. Each extension is illustrated by an example or by tables showing computational results. In the tables we use the following notation:  $z_{LP}$  denotes the value of the LP-relaxation, and  $z_{IP}$  and  $z_{MIP}$  denote the optimal value of the integer and the mixed-integer optimization problems respectively. By *% gap* we mean the percentage duality gap, i.e.  $(z_{IP} - z_{LP})/z_{IP}$ . The percentage duality gap closed, denoted *% gap closed* is calculated as  $(z_{LP}^{\text{root}} - z_{LP})/(z_{IP} - z_{LP})$ , where  $z_{LP}^{\text{root}}$  is the value of the LP-relaxation after all violated inequalities that have been identified in the root node of the branch-and-bound tree have been

iteration	$z_{LP}$	inequalities
1	6,662.5	13
2	6,883.5	15
3	6,912.5	7
4	6,918.8	9
5	6,928.0	6
6	6,935.3	9
7	6,937.2	8
8	6,939.5	5
9	6,940.4	4
10	6,940.8	12
11	6,941.2	5
12	6,941.5	3
13	6,942.0	

Table 1: A cutting plane algorithm applied to a 120-city TSP.

added. The number of branch-and-bound nodes needed to verify the optimal solution is given in the column *B&B nodes*.

### 3.1 Extending the Cutting Plane Algorithm

There are several ways to extend the basic cutting plane algorithm. We will describe the major additional techniques in the order in which they appear in an extended cutting plane algorithm.

#### 3.1.1 Preprocessing

Preprocessing integer linear programs involves removing redundant constraints, tightening the constraint coefficients and right-hand sides of the constraints, and fixing variables to certain values. This leads not only to better lower bounds provided by the linear relaxation, but also to a significant reduction in the size of the formulation, both with respect to the number of constraints and number of variables. An important factor is also that the instance becomes numerically more tractable if large coefficients are reduced. There are many preprocessing techniques described in the literature. For each technique, or combination of techniques, one needs to find the right balance between effectiveness and computing time. Here we shall present some simple methods that strengthen a linear program quickly. These methods are described by Savelsbergh (1994), and originally developed by Crowder et al. (1983) and Hoffman and Padberg (1991).

Consider the following subset of constraints from a mixed integer program, where  $N^+$  is the subset of variables with positive coefficients,  $N^-$  is the subset of variables with negative coefficients, and  $N = N^+ \cup N^-$ .

$$\sum_{j \in N^+} a_j x_j - \sum_{j \in N^-} a_j x_j \leq b \tag{57}$$

$$l_j \leq x_j \leq u_j \quad \text{for all } j \in N \quad (58)$$

A lower bound on the left-hand side of (57) is  $LB = \sum_{j \in N^+} a_j l_j - \sum_{j \in N^-} a_j u_j$ . If  $LB > b$ , then the problem is infeasible. An upper bound on the left-hand side of (57) is  $UB = \sum_{j \in N^+} a_j u_j - \sum_{j \in N^-} a_j l_j$ . If  $UB \leq b$ , then the constraint is redundant. It is also possible to tighten the bounds (58) on the variables by considering one variable at the time. Consider variable  $x_k$ ,  $k \in N^+$ , and let  $LB_k = \sum_{j \in N^+ \setminus \{k\}} a_j l_j - \sum_{j \in N^-} a_j u_j$ . Clearly, every feasible solution satisfies  $x_k \leq (b - LB_k)/a_k$ . Hence, the upper bound  $u_k$  can be decreased if  $u_k > (b - LB_k)/a_k$ . Analogous results can be obtained for the lower bound  $l_k$ .

An elegant preprocessing technique is “probing” on the variables, which means fixing variables temporarily. Probing techniques were introduced by Guignard and Spielberg (1981). By fixing a variable we may detect logical relations between variables that can be used to tighten, and reduce the size of the formulation as is demonstrated in the following example. Consider the following set of constraints with two binary variables  $x_1$  and  $x_2$ , and two nonnegative real variables  $y_1$  and  $y_2$ .

$$\begin{array}{rcll} y_1 & + & 3 y_2 & \geq & 12 \\ 2 y_1 & + & y_2 & \geq & 15 \\ y_1 & & & \leq & 10 x_1 \\ & & y_2 & \leq & 20 x_2 \end{array}$$

We probe on  $x_1$  by setting  $x_1$  equal to zero. Then, by the third constraint,  $y_1$  has to be equal to zero as well, which, due to the second and fourth constraints, implies that  $y_2 \geq 15$  and  $x_2 = 1$ . If we consider the first constraint we see that if  $x_1 = 0$  then we can increase the right-hand side to 45. If however  $x_1 = 1$  then the right-hand side has to be equal to 12. Hence, it is possible to add the term  $(45 - 12)(1 - x_1)$  to the right-hand side of the first constraint that now becomes

$$y_1 + 3y_2 \geq 12 + 33(1 - x_1).$$

Implication inequalities derived from binary variables can also be used to obtain clique constraints. In the previous example we saw that  $x_1 = 0$  implies  $x_2 = 1$ . Thus, we have  $x'_1 + x'_2 \leq 1$ , where  $x'_i$ ,  $i = 1, 2$  denotes the complement of the variable of  $x_i$ . To find such clique inequalities we can construct an auxiliary graph that has one vertex for every variable and its complement. Two vertices are connected by an edge if the corresponding variables cannot both have value one. Consider the auxiliary graph shown in Figure 8.

From the structure of the graph we conclude that  $x'_2$  has to be equal to zero. To see that this is true note that  $x'_2 = 1$  implies  $x_2 = 0$ . If  $x_2 = 0$  then either  $x_3 = 0$  or  $x_3 = 1$ . If  $x_3 = 0$ , then  $x'_3 = 1$ , which implies  $x'_1 = 0$ , which in turn implies that  $x_1 = 1$ . This is however not feasible since  $x_1$  is adjacent to  $x'_2$ . A similar contradiction is obtained if we choose  $x_3 = 1$ . This example shows that by investigating logical implication we may be able to fix variables and thereby reduce the problem size. Moreover, the cliques in the auxiliary graph do in general induce inequalities that are stronger than the inequalities in the original formulation.

The effectiveness of the various preprocessing techniques has been tested by Savelsbergh (1994) on a set of 10 mixed integer programming problems from the literature. Table 2 shows

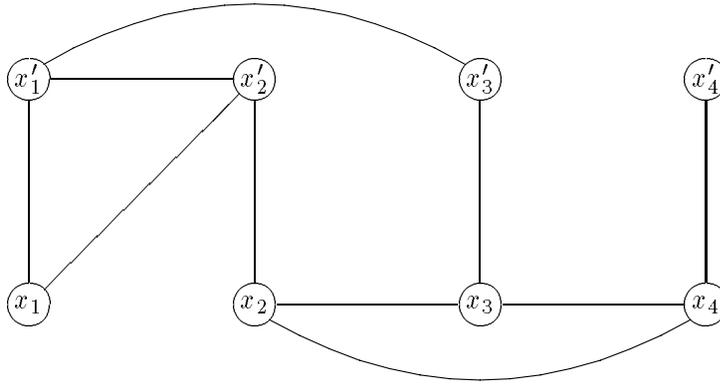


Figure 8: Auxiliary vertex packing graph.

the improvement of the lower bound after preprocessing as well as in the number of branch-and-bound nodes needed to verify the optimal solution. Observe that the linear programming bound increases substantially for all problems, and that the size of the branch-and-bound tree decreases quite a lot for most instances. For two instances however, the number of branch-and-bound nodes of the preprocessed problem is larger than for the original problem. This phenomenon is not really well understood. For more details regarding preprocessing we refer to Crowder et al. (1983), Hoffman and Padberg (1991) and Dietrich and Escudero (1990).

problem	$z_{LP}$ without preproc.	$z_{LP}$ with preproc	$z_{MIP}$	B&B nodes without prepr.	B&B nodes with prepr.
egout	149.5	562.1	568.1	553	3
fixnet3	40717.0	50414.2	51973.0	131	5
fixnet4	4257.9	7703.4	8936.0	2561	1031
fixnet6	1200.8	3192.5	3983.0	4795	4305
khb05250	95919464.0	106750366.0	106940226.0	11483	13
gen	112130.0	112271.0	112313.0	11	15
att	125.9	149.1	160.2	6459	127
sample2	247.0	290.4	375.0	336	51
p0033	2520.8	2838.5	3089.0	15	7
lseu	834.6	947.9	1120.0	297	464

Table 2: Effect of preprocessing techniques.

### 3.1.2 Postprocessing the Linear Program

After the linear program is solved, either the optimal solution is found, or, more usually, a fractional solution  $x^*$  is obtained, which provides a lower bound  $z_{LP}$  on the optimal value  $z_{IP}$ . Suppose we know a feasible solution with value  $z_F$ . The value  $z_F$  is an upper bound on  $z_{IP}$ , thus  $z_{IP}$  is guaranteed to lie in the interval  $[z_{LP}, z_F]$ . Heuristics that use the fractional solution  $x^*$  to create a feasible solution are known as *primal heuristics*. One example of a simple primal heuristic is rounding the fractional variables to feasible integer values. Besides providing a worst case distance between the lower bound and the optimal value, an upper

problem	% gap	B&B nodes	time (s)	cover ineq.	% gap closed	B&B nodes	time (s)
25081	5.9	23	8	4	100.0	1	3
25082	10.3	125	34	10	74.3	7	7
25083	7.5	79	25	6	85.5	5	8
25084	2.2	9	6	1	100.0	1	4
25085	5.2	19	7	5	86.6	3	7
50331	1.5	399	686	13	86.0	31	125
50332	1.2	691	1560	58	54.3	51	450
50333	1.5	259	556	122	54.1	89	769
50334	0.7	239	493	42	76.6	23	213
50335	1.3	685	1232	25	78.3	49	248

Table 3: Result of adding knapsack cover inequalities to CFL.

bound can also be used to fix variables by reduced cost fixing, or more involved, by parametric analysis on a single variable.

### 3.1.3 Generating Generic Inequalities

Besides the problem specific classes of valid inequalities, we can try to find violated generic inequalities. Many capacitated problems contain knapsack type constraints, in which case we may try to find violated extended knapsack cover inequalities (30). Other generic classes of valid inequalities are clique inequalities (14), obtained from the auxiliary graph of the binary variables as shown in Figure 8, and flow cover inequalities (41), obtained from variable upper bound constraints. The capacitated facility location problem provides a good insight in what these generic inequalities might offer. Table 3 shows the improvement obtained by adding extended cover inequalities to the formulation given in Section 2.6. The first five instances have 8 facilities and 25 clients, whereas the last five instances have 33 facilities and 50 clients. For more details, see Aardal (1994).

## 3.2 Embedding the Cutting Plane Algorithm in a Branch and Bound Framework

### 3.2.1 The Algorithm

In the early days of polyhedral techniques problems were solved by applying a cutting plane algorithm, followed by a straightforward branching process. In the mid-eighties Grötschel et al. (1984) used a cutting plane algorithm in every node of the branch-and-bound tree to solve the linear ordering problem. Padberg and Rinaldi (1987) called this idea *branch and cut*.

#### Outline of the branch-and-cut algorithm.

1. Initialize a list  $L$  of subproblems of the original problem. Repeat steps 2 and 3, until  $L$  is empty.

2. Select a subproblem  $S$  from  $L$ .
3. Consider the linear relaxation of  $S$  and apply a cutting plane algorithm to the relaxation. If  $S$  is not solved, create new subproblems by branching. Put the new subproblems in  $L$ .

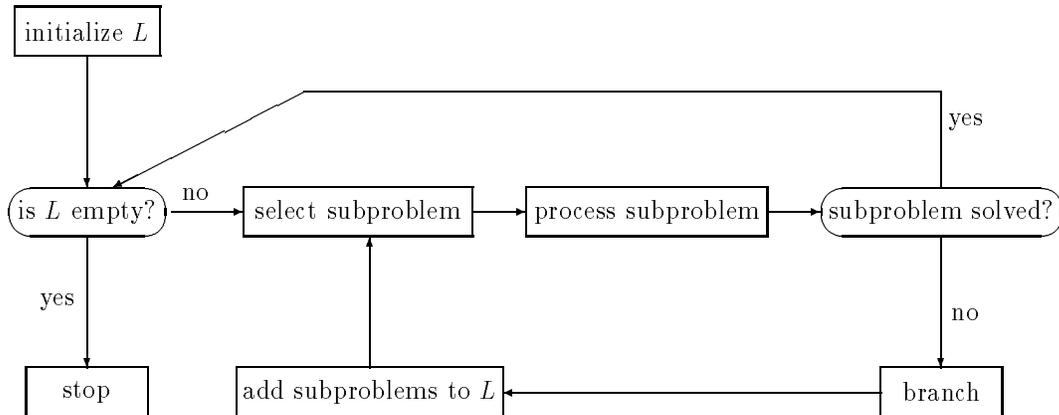


Figure 9: Branch-and-cut algorithm.

Every subproblem in  $L$  corresponds to a node in the branch-and-cut tree. The subproblems that still need to be investigated are called *active*. In order to avoid complete enumeration the search tree is *pruned* at subproblem  $j$ , i.e. no further subproblems are created at node  $j$ , if one of the following conditions hold: *a)* subproblem  $j$  is infeasible, *b)* the optimal solution to the linear relaxation of subproblem  $j$  is integral, or *c)*  $z_{LP}^j \geq \bar{z}$ , where  $\bar{z}$  is the best known upper bound.

In the branch-and-cut algorithm we need to specify a *search strategy* and a *branching strategy*, i.e. how to select a subproblem from the list  $L$ , and how to create new subproblems. The most commonly used search strategies are *depth-first search* and *breadth-first search*. In depth-first search one of the subproblems created at the current node is investigated if the current node is not pruned, whereas in breadth-first search all nodes at the current level of the tree are investigated before any node at the level below. The most frequently used branching rules are to branch on a variable according to one, or a mix, of the following four criteria. Here we assume that the variables are binary.

1. Select the variable with value closest to 0.5.
2. Select the variable with value closest to 1.
3. Select the variable with highest objective coefficient.
4. Select a set  $P$  of “promising” variables and compute for each variable in  $P$  the lower bound that is obtained at the corresponding subproblem. Select the variable that yields the smallest lower bound.

Padberg and Rinaldi (1991) suggest a combination of 1. and 3. for the traveling salesman problem. Rule 2. is surprisingly effective in combination with a depth-first strategy. Rule 4., introduced by Applegate et al. (1994), has similarities with the “steepest-edge” idea used in the simplex method for linear programming when choosing the variable to enter the basis. Other strategies have been proposed by Balas and Toth (1985). Jünger et al. (1992) report on computational experience with some combinations of these rules. When branching on a constraint, usually a clique constraint, a new branch is created for each value that the left-hand side of the constraint can obtain. Clochard and Naddef (1993) suggest such a rule for the traveling salesman problem.

### 3.2.2 Implementation Issues

The various components of the extended cutting plane algorithm may not be very effective in each node of the branch-and-cut tree. Preprocessing for instance has much effect in the root node of the tree since the original formulation of a problem usually contains a lot of redundancy. Similarly, it may be hard to find effective cutting planes in the subproblems further down in the tree. Hence, the major effort on separation is usually put in the root node. In an implementation of a branch-and-cut algorithm we can therefore introduce selection mechanisms for where in the tree certain components should be performed. Effectiveness versus computational effort should then be weighed against each other.

As mentioned above, the search tree can be pruned at a certain node if the lower bound obtained at that node exceeds the best known upper bound. In order to decrease the expected size of the search tree it is therefore crucial to compute a good upper bound by a primal heuristic before entering the branching phase.

*Branch pausing*, introduced by Padberg and Rinaldi (1991), is a strategy where subproblems with high lower bounds are temporarily ignored if the lower bounds are greater than a certain threshold value. The threshold value is an estimate of the optimal value of the problem. The advantage with branch pausing is that the expected size of the search tree gets smaller. If we choose to consider subproblems in the order of increasing value of the lower bounds the implementation however gets quite complicated since subsequently chosen subproblems are not necessarily related.

Maintaining the cutting planes is a rather difficult implementation issue. In early versions of branch-and-cut packages, one was only allowed to generate globally valid inequalities, i.e., inequalities that are valid for the original problem instance. These inequalities were maintained in a central pool, from which one could select violated inequalities for the current subproblem. The global cuts usually work well, but to use the full power of the branch-and-cut algorithm, one should also be able to generate inequalities that are locally valid. Balas et al. (1994) report on very good results using branch and cut with locally valid Gomory cuts. When solving large instances it becomes important to work with a formulation that is as small as possible. One important feature is therefore to be able to delete inequalities from the active formulation and store them in a pool. A detailed overview of general implementation ideas can be found in Jünger et al. (1994). Data structures and other implementation details specific

for the traveling salesman problem can be found in Applegate et al. (1994). To conclude this section we show in Figure 10 the branch-and-cut tree of a 532-city traveling salesman problem solved by Padberg and Rinaldi (1987). This tree gives an indication of the development of the lower bound at different levels of the tree.

## 4 Computational Results for Selected Problems

To give an idea of how polyhedral techniques perform, and how large instances can be solved, we have selected a number of problem types for which computational results are reported in the literature. For a more extensive survey we refer to Jünger et al. (1994).

### 4.1 The Vertex Packing Problem

Nemhauser and Sigismondi (1992) report on solving randomly generated instances of the maximum cardinality vertex packing problem. The sizes of the instances vary between 40 and 120 vertices, and for every size they consider different densities by changing the probability that an edge is in the graph between 0.1 and 0.9. The code used by the authors was limited in the sense that the cutting plane algorithm was run only in the root node, and that only primitive branching rules were available. In Table 4 we report the results for the 0.2 density instances.

vertices	% gap	clique ineq.	odd-hole ineq.	% gap closed	B&B nodes	LP- iterations
40	7	86	0	100.0	1	41
60	13	203	36	92.3	16	1439
80	21	369	33	80.9	97	13352
90	15	222	13	86.7	58	3649
100	29	181	19	93.1	108	6631
110	35	781	5	77.1	394	84115
120	40	903	5	72.5	251	35194

Table 4: Results for the vertex packing problem.

In general the clique inequalities closed most of the duality gap, but for low-density graphs lifted odd-holes were also important. The test instances with medium density graphs were the most difficult ones to solve. For instance, some of the medium-density 120-vertex problems were not solved within 100000 LP iterations. It seems from this study that random vertex packing problems are difficult to solve by the polyhedral approach. If we consider structured vertex packing problems however, much larger instances can be tackled as the following two applications show.

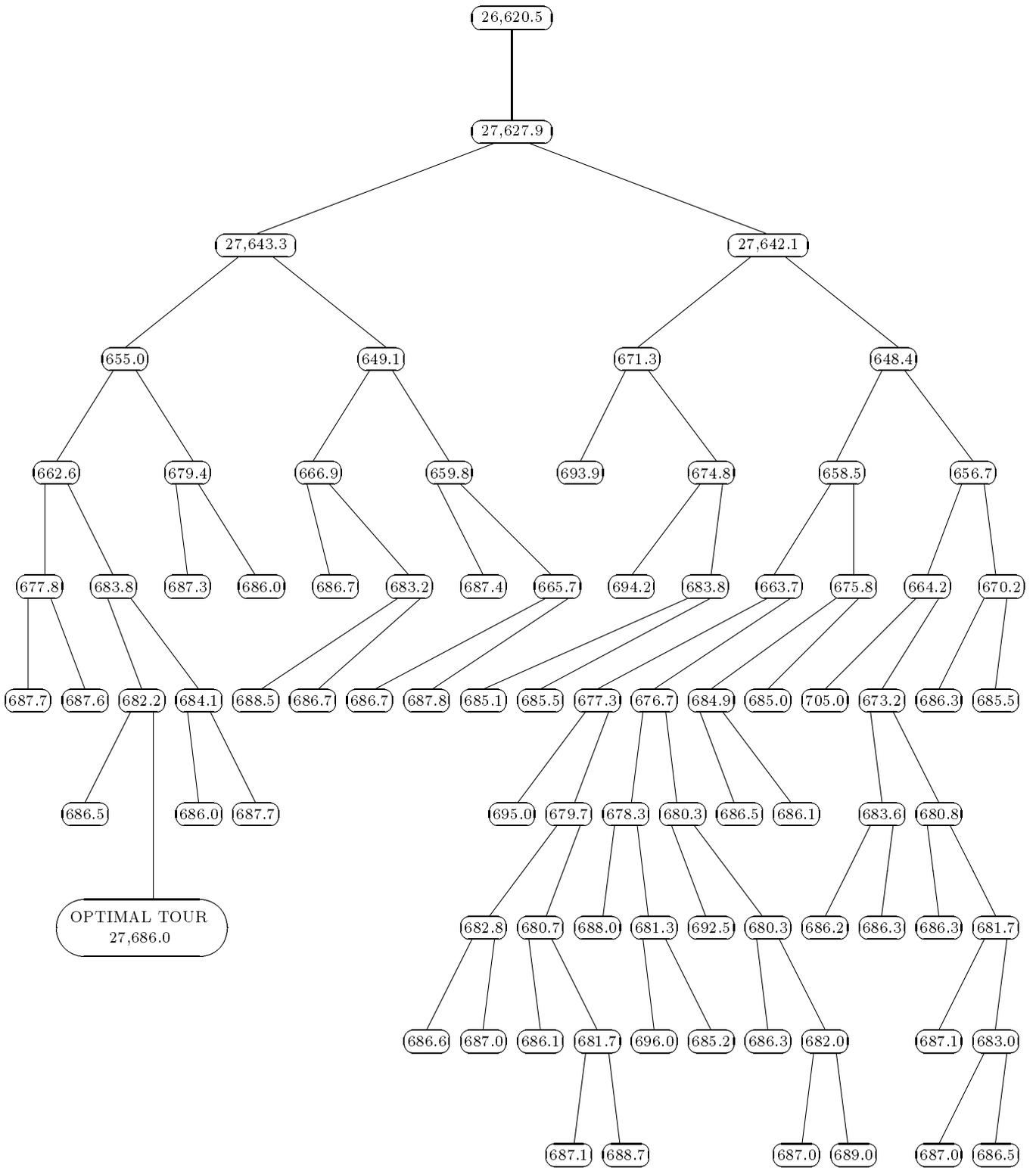


Figure 10: Branch-and-cut tree for the 532-city TSP.

### 4.1.1 Frequency Assignment

The frequency assignment problem is the problem of assigning frequencies to transmission links such that no interference occurs and such that the number of used frequencies is minimized. The frequency assigned to a specific link has to be chosen from a set depending on the link. To avoid interference we have restrictions on every pair  $(i, j)$  of links that the frequencies assigned to these links should differ by at least a certain prespecified amount. The problem is modeled as a vertex packing problem using a binary variable for each feasible combination of a link and a frequency. In Table 5 we present computational results as reported by Aardal et al. (1995). The number of variables is approximately equal to forty times the number of links giving instances of between approximately 4000 and 18000 variables. By making heavily use of preprocessing, the number of variables is reduced by at least fifty percent. The “lower bound by branch and bound” reported in the table is obtained by partial branching, and the time reported is the time needed to verify optimum, or, in the case of the last instance, the time needed to find the feasible solution of value 16. The computations were carried out on a HP90000/720 work station.

links	lower bound by clique ineq.	lower bound by B&B	best known feasible value	time (s)
100	14	14	14	46
200	14	14	14	1925
340	20	22	22	6167
458	14	14	16	400

Table 5: Results for the frequency assignment problem.

### 4.1.2 The Set Partitioning Problem: Airline Crew Scheduling

Hoffman and Padberg (1993) report on solving huge set partitioning problems arising in airline crew scheduling problems. The cutting plane phase uses preprocessing techniques, and clique and lifted odd-hole inequalities. In the branch-and-cut phase a variable branching rule is used. From the reported results we have selected the instances with the largest number of variables and constraints. These results are presented in Table 6. Of the total time needed to solve the various problems, by far the longest time is spent on getting within the last percent of the optimal value. In Table 7 we show for three instances how much time it takes to get within one and two percent of the optimal value, as well as the time needed to verify optimality.

## 4.2 The Traveling Salesman Problem

The literature on computational results for the traveling salesman problem is vast, and some of the results have already been shown in previous sections. To make the progress visual, we

original		preprocessed				B&B	
variables	constr.	variables	constr.	$z_{LP}^{\text{root}}$	ineq.	nodes	$z_{IP}$
5198	531	3846	360	30494	0	1	30494
7292	646	5862	488	26977	74	1	27040
8308	801	6235	521	53736	345	5	53839
8627	825	6694	537	49616	37	1	49649
148633	139	138951	139	1181590	0	1	1181590
288507	71	202603	71	132878	0	1	132878
1053137	145	370642	90	9950	389	1	10022

Table 6: Results for the airline crew scheduling problem.

variables	constraints	time 2% (s)	time 1% (s)	time opt (s)
87482	36	225	298	2642
8904	823	375	375	14441
7195	426	868	7443	139337

Table 7: Time needed to get within certain percentages of the optimal value.

give in Table 8 a list of “world records” with respect to the size of the instances. It should be noted that there are still some small instances unsolved, which indicates that small does not necessarily imply easy, and that large is not synonymous with difficult. The instances we report on here are all Euclidean symmetric traveling salesman problems, and they arise from applications such as finding routes through actual cities, routing of drilling machines when manufacturing printed circuit boards, and x-ray crystallography. The instances can be found in the library NETLIB, see Reinelt (1991). Table 8 contains information on the number of “cities”  $n$  of the instances. For all instances a complete graph is assumed which means that the number of variables is equal to  $\frac{1}{2}n(n-1)$ . The data is obtained from the original articles, so later techniques may perform differently. For instance, for the 532-city instance we know of three different numbers reported for the total number of branch-and-cut nodes needed. To give an idea of the solution times, the 2392-city problem was solved in approximately 6 hours on a CYBER. As can be seen from Table 8, the lower bounds in the root node are very close to the optimal value which partly explains the success of cutting plane algorithms for the symmetric traveling salesman problem. When solving large instances a very advanced implementation is necessary, see Applegate et al. (1994).

### 4.3 General Zero-One Linear Programs

Crowder et al. (1983) present the first computational results for large-scale zero-one linear programs. The test problems are real life instances without any apparent structure. On a set of 10 instances they show the effects of simple preprocessing techniques, and knapsack cover and  $(1, k)$ -configuration inequalities generated and added in the root node of the branch-and-bound tree. In the other nodes they use only reduced-cost fixing to eliminate variables. Their computational results are shown in Table 9.

cities	$x_{LP}^{root}$	$x_{IP}$	B&B nodes	application	year	reported by:
49	12345	12345	1	map USA	1954	Dantzig et al.
120	6942	6942	1	map Germany	1980	Grötschel
318	??	41349	??	drilling	1980	Crowder & Padberg
532	27628	27686	85	map USA	1987	Padberg & Rinaldi
666	294080	294358	21	worldmap	1991	Grötschel & Holland
1002	258860	259045	13	drilling	1990	Padberg & Rinaldi
2392	378027	378032	3	drilling	1990	Padberg & Rinaldi
3038	137660	137694	287	drilling	1992	Applegate et. al
4461	182528	182566	2092	??	1994	Applegate et. al
7397	23253123	23260728	2247	programmable logic arrays	1994	Applegate et. al

Table 8: Results for the traveling salesman problem.

original problem			preprocessing			cutting plane		B&B	
vars	constr.	$z_{LP}$	vars.	constr.	$z_{LP}$	ineq.	$z_{LP}$	nodes	$z_{IP}$
33	16	2520.6	33	16	2819.4	36	3065.3	113	3089.0
40	24	61796.5	40	24	61829.1	29	61862.8	11	62027.0
201	134	6875.0	195	134	7125.0	139	7125.0	1116	7615.0
282	242	176867.5	282	222	176867.5	462	255033.1	1862	258411.0
291	253	1705.1	290	206	1749.9	278	5022.7	87	5223.8
548	177	315.3	527	157	3125.9	296	8643.5	36	8691.0
1550	94	1706.5	1550	94	1706.5	94	1706.5	10	1708.0
1939	109	2051.1	1939	109	2051.1	110	2051.1	334	2066.0
2655	147	6532.1	2655	147	6532.1	149	6535.0	214	6548.0
2756	756	2688.7	2734	739	2701.1	1065	3115.3	2392	3124.0

Table 9: Results for general zero-one problems.

## 5 Alternative Techniques

In the last two decades there has been a remarkable development in polyhedral techniques leading to an increase in the size of many combinatorial problems that can be solved by a factor hundred. Most of the computational successes have occurred for zero-one combinatorial problems where the polytope is defined once the dimension is given, such as the traveling salesman problem. For more complex combinatorial optimization problems, and for general integer programming problems less progress has been made. Here we shall give a brief overview of other available solution techniques.

If the number of variables is large compared to the number of constraints *column generation* may in many cases be a good alternative. It can be viewed as a dual approach to polyhedral techniques in the sense that one aims at generating the extreme points of  $\text{conv}(S)$  rather than its facets. Instead of solving a separation problem to generate a violated inequality we need to solve the problem of finding a column, i.e. a feasible solution, that can improve the objective function. Column generation was introduced by Gilmore and Gomory (1961) to solve the cutting stock problem. Recent applications are presented by Savelsbergh (1993) and Vanderbeck and Wolsey (1994).

In *Lagrangean relaxation* we relax the problem by removing a subset of the constraints, different from the nonnegativity constraints. Violation of the relaxed constraints is penalized by including these constraints, with a nonnegative multiplier, in the objective function. The multipliers are then updated iteratively so as to maximize the lower bound obtained from the relaxed problem. To update the Lagrangean multipliers *subgradient optimization* is often used. Lagrangean relaxation was used successfully by Held and Karp (1970,1971) to solve traveling salesman problems. For further details we refer to Geoffrion (1974), Held et al. (1974) and Fisher (1981).

Lovász and Schrijver (1991) considered 0-1 integer linear programming problems and proposed a procedure of increasing — or lifting — the dimension of the problem by introducing more variables and then projecting the extended formulation back onto the original space. From the projection step strong valid inequalities are obtained for the original problem. They showed that by repeating this procedure a number of times equal to the number of variables in the original space, the convex hull of feasible solutions is obtained. At the lifting step the number of variables involved are squared and the number of constraints is increased by a factor two times the number of variables. Balas et al. (1993) developed this technique further and proved that it is sufficient to double the number of variables and constraints at the lifting step. They also related this technique to a convexification technique introduced by Balas (1979) and used this relation to develop a class of finitely converging cutting plane algorithms, called *lift-and-project* algorithms, for mixed 0-1 linear programming problems.

Cook et al. (1993) present an implementation of a *generalized basis reduction algorithm* for solving general integer programming problems. This method is based on the important theoretical result by H.W. Lenstra (1983) that the integer programming problem (3) is polynomially solvable if the number of variables is fixed. A central part of his results is to show that it is possible, in polynomial time, to find either an integral vector belonging to the

bounded polyhedron  $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ , or an integral direction  $d \in \mathbb{Z}^n \setminus \{0\}$  such that  $\max\{dx : x \in P\} - \min\{dx : x \in P\} \leq \gamma_n$  where  $\gamma_n$  depends on the dimension of  $P$  only. A direction  $d$  as described above is called *flat*. Instead of branching on variables as in conventional branch-and-bound techniques, the flat directions are used to branch on hyperplanes  $dx = t$ ,  $x \in P$ , where  $t$  is an integer varying between  $\lceil \min\{dx : x \in P\} \rceil$  and  $\lfloor \max\{dx : x \in P\} \rfloor$ . Since the direction  $d$  is flat the number of subproblems created at each level of the search tree is limited by a constant depending only on  $n$ . Moreover, we have no more than  $n$  levels in the tree. To find flat directions Cook et al. use the generalized basis reduction technique developed by Lovász and Scarf (1992).

One of the main drawbacks of polyhedral techniques, as described in Section 2, is that the separation problem based on several facet defining inequalities is hard to solve, or sometimes even hard to formulate. Boyd (1994) developed a cutting plane algorithm for general integer programming that is based on so-called *Fenchel duality*. The basic idea of Boyd's method is to prove that a certain point  $\bar{x}$  belongs to  $\text{conv}(S)$  or to find a separating hyperplane, that is as far as possible from  $\bar{x}$ . Such a separating hyperplane is referred to as a *Fenchel cut*. To find a Fenchel cut one needs to maximize a piecewise linear function on a nonlinear domain. Boyd suggests different relaxations of the nonlinear domain and reports on computational experience using these relaxations to solve the test problems of Crowder et al. (1983).

Natraj et al. (1994) used the theory of *Gröbner bases* to develop a solution method to solve a difficult scheduling problem. For a more general treatment of this technique we refer to Thomas (1992) and Sturmfels and Thomas (1994). The idea behind the approach by Natraj et al. is to walk from one integer solution to another in such a way that the objective function improves at every step. The directions used in this walk are specified by the Gröbner basis associated with the problem. A Gröbner basis can be viewed as a so-called *test set* of integral vectors  $x^1, \dots, x^N$ , depending on the constraint matrix only. These vectors have the property that a feasible solution  $x^*$  is optimal if and only if  $c(x^* + x^k) \geq cx^*$  whenever  $x^* + x^k$ ,  $k = 1, \dots, N$  is a feasible solution.

## Acknowledgment

We would like to thank David Applegate for providing data on the traveling salesman problem reported in Section 4.2. Large parts of this article was written while the first author was visiting University of California at Berkeley. Financial support provided by the late Gene Lawler, and Umesh Vazirani through grant IRI-9120074 from NSF is greatly acknowledged.

## References

- K. AARDAL (1994) "Capacitated facility location: separation algorithms and computational experience", CentER Discussion Paper 9480, Tilburg.
- K. AARDAL, A. HIPOLITO, C.P.M. VAN HOESEL, B. JANSEN, C. ROOS, AND T. TERLÍKY (1995)

- “A branch-and-cut algorithm for the frequency allocation problem”, Working Paper, Tilburg University, Tilburg.
- K. AARDAL AND C.P.M. VAN HOESEL (1995) “Uncapacitated facility location: lifting, separation and computations” (in preparation).
- K. AARDAL, M. LABBÉ, J. LEUNG AND M. QUEYRANNE (1994) “On the two-level uncapacitated facility location problem”, CentER Discussion Paper 9486, Tilburg.
- K. AARDAL, Y. POCHET AND L.A. WOLSEY (1993) “Capacitated facility location: valid inequalities and facets”, CORE Discussion Paper 9323, Louvain-la-Neuve, (to appear in *Mathematics of Operations Research*).
- J.M. VAN DEN AKKER, C.P.M. VAN HOESEL AND M.W.P. SAVELSBERGH (1993) “Facet inducing inequalities for single-machine scheduling problems”, Memorandum COSOR 93-27, Eindhoven University of Technology, Eindhoven.
- D. APPLGATE, R.E. BIXBY, V. CHVÁTAL AND W. COOK (1994) “Finding cuts in the TSP”, Preliminary report, available through anonymous ftp from netlib.att.com in the directory netlib/att/math/applegate/TSP.
- J.R. ARAQUE (1989) “Solution of a 48-city vehicle routing problem by branch-and-cut”, Working Paper, Department of Applied Mathematics and Statistics, State University of New York, Stony Brook.
- J.R. ARAQUE (1990) “Lots of combs of different sizes for vehicle routing”, CORE Discussion Paper 9074, Louvain-la-Neuve.
- J.R. ARAQUE, L.A. HALL AND T.L. MAGNANTI (1990) “Capacitated trees, capacitated routing and associated polyhedra”, CORE Discussion Paper 9061, Louvain-la-Neuve.
- E. BALAS (1975a) “Facets of the knapsack polytope”, *Mathematical Programming* **8** 146–164.
- E. BALAS (1975b) “Disjunctive programming”, *Annals of Discrete Mathematics* **5** 3–51.
- E. BALAS, S. CERIA AND G. CORNUÉJOLS (1993) “A lift-and-project cutting plane algorithm for mixed 0-1 programs”, *Mathematical Programming* **58** 295–324.
- E. BALAS, S. CERIA, G. CORNUÉJOLS AND NATRAJ (1994a) “Gomory cuts revisited”, Working Paper No. 1994-16, Carnegie Mellon University, Graduate School of Industrial Administration, Pittsburgh.
- E. BALAS, S. CERIA, G. CORNUÉJOLS AND G. PATAKI (1994b) “Polyhedral methods for the maximum clique problem”, Management Science Research Report Number 602, Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh.
- E. BALAS AND M. FISCHETTI (1992) “The fixed-outdegree 1-arborescence polytope”, *Mathematics of Operations Research* **17** 1001–1018.
- E. BALAS, M. FISCHETTI AND W.R. PULLEYBLANK (1995) “The precedence-constrained asymmetric traveling salesman polytope”, *Mathematical Programming* **68** 241–265.
- E. BALAS AND A.C. HO (1980) “Set covering algorithms using cutting planes, heuristics and sub-gradient optimization” *Mathematical Programming Study* **12** 37–60.
- E. BALAS AND S.M. NG (1989a) “On the set covering polytope: I. All the facets with coefficients in  $\{0, 1, 2\}$ ”, *Mathematical Programming* **43** 57–69.
- E. BALAS AND S.M. NG (1989b) “On the set covering polytope: II. Lifting the facets with coefficients in  $\{0, 1, 2\}$ ”, *Mathematical Programming* **45** 1–20.
- E. BALAS AND M.W. PADBERG (1972) “On the set covering problem”, *Operations Research* **20** 1152–1161.
- E. BALAS AND W.R. PULLEYBLANK (1983) “The perfectly matchable subgraph polytope of a bipartite graph”, *Networks* **13** 486–516.

- E. BALAS AND P. TOTH (1985) “Branch and bound methods”, in: *Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization* (E. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys, eds.), Wiley and Sons.
- E. BALAS AND E. ZEMEL (1977) “Critical cutsets and of graphs and canonical facets of set-packing polytopes”, *Mathematics of Operations Research* **2** 15–19.
- E. BALAS AND E. ZEMEL (1978) “Facets of the knapsack polytope from minimal covers”, *SIAM Journal on Applied Mathematics* **34** 119–148.
- F. BARAHONA, M. GRÖTSCHEL AND A.R. MAHJOUR (1985) “Facets of the bipartite subgraph polytope”, *Mathematics of Operations Research* **10** 340–358.
- F. BARAHONA, M. GRÖTSCHEL, M. JÜNGER AND G. REINELT (1988) “An application of combinatorial optimization to statistical physics and circuits layout design”, *Operations Research* **36** 493–513.
- F. BARAHONA AND A.R. MAHJOUR (1986) “On the cut polytope”, *Mathematical Programming* **36** 157–173.
- F. BARAHONA AND A.R. MAHJOUR (1989) “Facets of the balanced (acyclic) induced subgraph polytope”, *Mathematical Programming* **45** 21–33.
- F. BARAHONA AND A.R. MAHJOUR (1992) “On 2-connected subgraph polytopes”, in: *Proceedings of the Second Conference on Integer Programming and Combinatorial Optimization* (E. Balas, G. Cornuéjols and R. Kannan, eds.)
- E.A. BOYD “Fenchel cutting planes for integer programs”, *Operations Research* **42** 53–64.
- D.C. CHO, E.L. JOHNSON, M.W. PADBERG AND M.R. RAO (1983a) “On the uncapacitated plant location problem. I: valid inequalities and facets”, *Mathematics of Operations Research* **8** 579–589.
- D.C. CHO, M.W. PADBERG AND M.R. RAO (1983b) “On the uncapacitated plant location problem. II: facets and lifting theorems”, *Mathematics of Operations Research* **8** 590–612.
- S. CHOPRA (1989a) “On ternary problems”, *Mathematical Programming* **45** 35–47.
- S. CHOPRA (1989b) “On the spanning tree polyhedron”, *Operations Research Letters* **8** 25–29.
- S. CHOPRA (1992) “The k-edge connected spanning subgraph polytope” in: *Proceedings of the Second Conference on Integer Programming and Combinatorial Optimization* (E. Balas, G. Cornuéjols and R. Kannan, eds.)
- S. CHOPRA, E.R. GORRES AND M.R. RAO (1992) “Solving the Steiner tree problem on a graph using branch and cut”, *ORSA Journal on Computing* **3** 149–156.
- S. CHOPRA AND M.R. RAO (1993) “The partition problem”, *Mathematical Programming* **59** 87–115.
- S. CHOPRA AND M.R. RAO (1994a) “The Steiner tree problem I: formulations, compositions and extensions of facets”, *Mathematical Programming* **64** 209–229.
- S. CHOPRA AND M.R. RAO (1994b) “The Steiner tree problem II: properties and classes of facets”, *Mathematical Programming* **64** 231–246.
- V. CHVÁTAL (1973) “Edmonds polytopes and a hierarchy of combinatorial problems”, *Discrete Mathematics* **4** 185–224.
- V. CHVÁTAL (1975) “On certain polytopes associated with graphs” *Journal of Combinatorial Theory B* **18** 138–154.
- J.M. CLOCHARD AND D. NADDEF (1993) “Using path inequalities in a branch and cut code for the symmetric traveling salesman problem”, in: *Proceedings of the Third Conference on Integer Programming and Combinatorial Optimization* (G. Rinaldi and L.A. Wolsey, eds.)
- M. CONFORTI, M.R. RAO AND A. SASSANO (1990/91a) “The equipartition polytope I”, *Mathematical Programming* **49** 49–70.

- M. CONFORTI, M.R. RAO AND A. SASSANO (1990/91b) “The equipartition polytope II”, *Mathematical Programming* **49** 71–90.
- W. COOK, C. COULLARD AND GY. TURÁN (1987) “On the complexity of cutting-plane proofs”, *Discrete Applied Mathematics* **18** 25–38.
- W. COOK, T. RUTHERFORD, H.E. SCARF AND D. SHALLCROSS (1993) “An implementation of the generalized basis reduction algorithm for integer programming” *ORSA Journal on Computing* **5** 206–212.
- G. CORNUÉJOLS, M.L. FISHER AND G.L. NEMHAUSER (1977) “On the uncapacitated location problem”, *Annals of Discrete Mathematics* **1** 163–177.
- G. CORNUÉJOLS AND F. HARCHE (1993) “Polyhedral study of the capacitated vehicle routing problem”, *Mathematical Programming* **60** 21–52.
- G. CORNUÉJOLS AND W.R. PULLEYBLANK (1982) “The traveling salesman polytope and  $\{0, 2\}$ -matchings”, *Annals of Discrete Mathematics* **16** 25–55.
- G. CORNUÉJOLS AND A. SASSANO (1989) “On the 0,1 facets of the set covering polytope”, *Mathematical Programming* **43** 45–56.
- G. CORNUÉJOLS AND J.-M. THIZY (1982) “Some facets of the simple plant location polytope”, *Mathematical Programming* **23** (1982) 50–74.
- H.P. CROWDER, E.L. JOHNSON AND M.W. PADBERG (1983) “Solving large-scale zero-one linear programming problems”, *Operations Research* **31** 803–934.
- H. CROWDER AND M.W. PADBERG (1980) “Solving large-scale symmetric traveling salesman problems to optimality”, *Management Science* **26** 459–509.
- G.B. DANTZIG, D.R. FULKERSON AND S.M. JOHNSON (1954) “Solution of a large-scale traveling-salesman problem”, *Operations Research* **2**, 393–410.
- G.B. DANTZIG, D.R. FULKERSON AND S.M. JOHNSON (1959) “On a linear-programming, combinatorial approach to the traveling-salesman problem”, *Operations Research* **7** 58–66.
- C.C DE SOUSA AND M. LAURENT (1991) “Some new classes of facets for the equicut problem”, CORE Discussion Paper 9157, Louvain-la-Neuve.
- M. DEZA, M. GRÖTSCHEL AND M. LAURENT (1992) “Clique-web facets for multicut polytopes”, *Mathematics of Operations Research* **17** 981–1000.
- M. DEZA AND M. LAURENT (1992a) “Facets for the cut cone I”, *Mathematical Programming* **56** 121–160.
- M. DEZA AND M. LAURENT (1992b) “Facets for the cut cone II: clique-web inequalities”, *Mathematical Programming* **56** 161–188.
- B.L. DIETRICH AND L.F. ESCUDERO (1990) “Coefficient reduction for knapsack like constraints in 0-1 programs”, *Operations Research Letters* **2** 9–14.
- J. EDMONDS (1965) “Maximum matching and a polyhedron with 0,1-vertices”, *Journal of Research of the National Bureau of Standards (B)* **69** 67–72.
- J. EDMONDS AND R. GILES (1977) “A min-max relation for submodular functions on graphs”, [in: *Studies in Integer Programming* (P.L. Hammer, E.L. Johnson, B.H. Korte and G.L. Nemhauser, eds.)] *Annals of Discrete Mathematics* **1** 185–204.
- M. FISCHETTI (1991a) “Facets of the asymmetric traveling salesman polytope”, *Mathematics of Operations Research* **16** 42–56.
- M. FISCHETTI (1991b) “Facets of two Steiner arborescence polyhedra”, *Mathematical Programming* **51** 401–419.
- M. FISCHETTI (1992) “Three facet-lifting theorems for the asymmetric traveling salesman polytope”, in: *Proceedings of the Second Conference on Integer Programming and Combinatorial Optimization* (E. Balas, G. Cornuéjols and R. Kannan, eds.)

- M.L. FISHER (1981) “The Lagrangean relaxation method for solving integer programming problems”, *Management Science* **27** 1–18.
- A.B. GAMBLE AND W.R. PULLEYBLANK (1989) “Forest covers and a polyhedral intersection theorem” *Mathematical Programming* **45** 49–58.
- M.R. GAREY AND D.S. JOHNSON (1979) *Computers and Intractability: A Guide to the Theory of NP-completeness*, Freeman, San Francisco.
- A.M. GEOFFRION (1974) “Lagrangean Relaxation for Integer Programming”, *Mathematical Programming Study* **2** 82–114.
- R. GILES AND W.R. PULLEYBLANK (1979) “Total dual integrality and integer polyhedra”, *Linear Algebra and Its Applications* **25** 191–196.
- P.C. GILMORE AND R.E. GOMORY (1961) “A linear programming approach to the cutting-stock problem”, *Operations Research* **9** 849–859.
- M.X. GOEMANS (1989) “Valid inequalities and separation for mixed 0-1 constraints with variable upper bounds”, *Operations Research Letters* **8** 315–322.
- M.X. GOEMANS (1992) “Polyhedral description of trees and arborescences”, in: *Proceedings of the Second Conference on Integer Programming and Combinatorial Optimization* (E. Balas, G. Cornuéjols and R. Kannan, eds.)
- M.X. GOEMANS (1993) “Worst-case comparison of valid inequalities for the TSP”, Working paper, Department of Mathematics, Massachusetts Institute of Technology.
- R.E. GOMORY (1958) “Outline of an algorithm for integer solutions to linear programs”, *Bulletin of the American Mathematical Society* **64**, 275–278.
- R.E. GOMORY (1960) “Solving linear programming problems in integers”, in: *Combinatorial Analysis* (R. Bellman and M. Hall, Jr., eds.), Proceedings of Symposia in Applied Mathematics X, American Mathematical Society, Providence, pp. 211–215.
- R.E. GOMORY (1963) “An algorithm for integer solutions to linear programs”, in: *Recent Advances in Mathematical Programming* (R.L. Graves and P. Wolfe, eds.), McGraw-Hill, New York, pp. 269–302.
- M. GRÖTSCHEL (1980) “On the symmetric traveling salesman problem: solution of a 120 city problem”, *Mathematical Programming Studies* **12** 61–77.
- M. GRÖTSCHEL AND O. HOLLAND (1985) “Solving matching problems with linear programming”, *Mathematical Programming* **33** 243–259.
- M. GRÖTSCHEL AND O. HOLLAND (1991) “Solution of large-scale symmetric traveling salesman problems”, *Mathematical Programming* **51** 141–202.
- M. GRÖTSCHEL, M. JÜNGER AND G. REINELT (1984) “A cutting plane algorithm for the linear ordering problem”, *Operations Research* **32** 1195–1220.
- M. GRÖTSCHEL, M. JÜNGER AND G. REINELT (1985) “Facets of the linear ordering problem”, *Mathematical Programming* **33** 43–60.
- M. GRÖTSCHEL, L. LOVÁSZ AND A. SCHRIJVER (1981) “The ellipsoid method and its consequences in combinatorial optimization”, *Combinatorica* **1** 169–197 [corrigendum: **4** 291–295].
- M. GRÖTSCHEL, A. MARTIN AND R. WEISMANTEL (1992a) “Packing steiner trees: polyhedral investigations”, Preprint SC 92-8, Konrad-Zuse-Zentrum für Informationstechnik, Berlin.
- M. GRÖTSCHEL, A. MARTIN AND R. WEISMANTEL (1993) “Routing in grid graphs by cutting planes”, in: *Proceedings of the Third Conference on Integer Programming and Combinatorial Optimization* (G. Rinaldi and L.A. Wolsey, eds.)
- M. GRÖTSCHEL, C.L. MONMA AND M. STOER (1992b) “Computational results with a cutting plane algorithm for designing communication networks with low-connectivity constraints” *Operations Research* **40** 309–330.

- M. GRÖTSCHEL, C.L. MONMA AND M. STOER (1995) “Design of survivable networks”, in: M. Ball, T.L. Magnanti, C.L. Monma and G.L. Nemhauser (eds.) *Handbooks in Operations Research and Management Science: Networks* (to appear), North-Holland, Amsterdam.
- M. GRÖTSCHEL AND M.W. PADBERG (1979) “On the symmetric traveling salesman problem I: inequalities”, *Mathematical Programming* **16** 265–280.
- M. GRÖTSCHEL AND W.R. PULLEYBLANK (1986) “Clique tree inequalities and the symmetric traveling salesman problem”, *Mathematics of Operations Research* **11** 537–569.
- M. GRÖTSCHEL AND Y. WAKABAYASHI (1989) “A cutting plane algorithm for a clustering problem”, *Mathematical Programming* **45** 59–96.
- M. GRÖTSCHEL AND Y. WAKABAYASHI (1990) “Facets of the clique-partitioning polytope”, *Mathematical Programming* **47** 367–387.
- M. GRÖTSCHEL AND Z. WIN (1992) “A cutting plane algorithm for the windy postman problem”, *Mathematical Programming* **55** 339–358.
- M. GUIGNARD AND K. SPIELBERG (1981) “Logical reduction methods in zero-one programming”, *Operations Research* **29**, 49–74.
- L.A. HALL AND T.L. MAGNANTI (1992) “A polyhedral intersection theorem for capacitated spanning trees”, *Mathematics of Operations Research* **17** 390–410.
- P.L. HAMMER, E.L. JOHNSON AND U.N. PELED (1975) “Facets of regular 0-1 polytopes”, *Mathematical Programming* **8** 179–206.
- M. HELD AND R.M. KARP (1970) “The traveling-salesman problem and minimum spanning trees”, *Operations Research* **18** 1138–1162.
- M. HELD AND R.M. KARP (1971) “The traveling-salesman problem and minimum spanning trees: part II”, *Mathematical Programming* **1** 6–25.
- M. HELD P. WOLFE AND H.P. CROWDER (1974) “Validation of subgradient optimization”, *Mathematical Programming* **6** 62–88.
- K.L. HOFFMAN AND M.W. PADBERG (1991) “Improving LP-representation of zero-one linear programs for branch-and-cut”, *ORSA Journal on Computing* **3** 121–134.
- K.L. HOFFMAN AND M.W. PADBERG (1993) “Solving airline crew scheduling problems by branch and cut”, *Management Science* **39** 657–682.
- M. JÜNGER AND P. MUTZEL (1993) “Solving the maximum weight planar subgraph”, in: *Proceedings of the Third Conference on Integer Programming and Combinatorial Optimization* (G. Rinaldi and L.A. Wolsey, eds.)
- M. JÜNGER, G. REINELT AND S. THIENEL (1992) “Provably good solutions for the traveling salesman problem”, Report No. 92.114, Angewandte Mathematik und Informatik, Universität zu Köln.
- M. JÜNGER, G. REINELT AND S. THIENEL (1994) “Practical problem solving with cutting plane algorithms in combinatorial optimization”, Report No. 94.156, Angewandte Mathematik und Informatik, Universität zu Köln.
- R.M. KARP (1972) “Reducibility among combinatorial problems”, in: *Complexity of Computer Computations* (R.E. Miller and J.W. Thatcher, eds.), Plenum Press, New York, pp. 85–103.
- R.M. KARP AND C.H. PAPADIMITRIOU (1980) “On linear characterizations of combinatorial optimization problems”, *Proceedings of the 21st Annual Symposium on Foundation of Computer Science*, IEEE, New York, pp. 1–9.
- M. LAURENT (1989) “A generalization of antiwebs to independence systems and their canonical facets”, *Mathematical Programming* **45** 97–108.
- J. LEE AND J. LEUNG (1993a) “On the boolean quadratic forest polytope”, Working Paper. Department of Management Information Systems, University of Arizona, Tucson.

- J. LEE AND J. LEUNG (1993b) “A comparison between two edge-coloring formulations”, *Operations Research Letters* **13** 215–223.
- J. LEUNG (1994) “Polyhedral structure and properties of a model for layout design”, *European Journal on Operational Research* **77** 195–207.
- H.W. LENSTRA, JR. (1983) “Integer programming with a fixed number of variables”, *Mathematics of Operations Research* **8** 538–548.
- J.M.Y. LEUNG AND T.L. MAGNANTI (1989) “Valid inequalities and facets of the capacitated plant location problem” *Mathematical Programming* **44** 271–291.
- J.M.Y. LEUNG (1994) “Polyhedral structure and properties of a model for layout design”, *European Journal of Operational Research* **77** 195–207.
- L. LOVÁSZ AND H.E. SCARF (1992) “The generalized basis reduction algorithm”, *Mathematics of Operations Research* **17** 751–764.
- L. LOVÁSZ AND A. SCHRIJVER (1991) “Cones of matrices and set-functions and 0-1 optimization”, *SIAM Journal on Optimization* **1** 166–190.
- J.E. MITCHELL AND B. BORCHERS (1992) “A primal-dual interior point cutting plane method for the linear ordering problem”, Report No. 204, Rensselaer Polytechnic Institute.
- J.E. MITCHELL AND B. BORCHERS (1993) “Solving real-world linear ordering problems using a primal-dual interior point cutting plane method”, Report No. 207, Rensselaer Polytechnic Institute.
- D. NADDEF (1992) “The binested inequalities for the symmetric traveling salesman polytope”, *Mathematics of Operations Research* **17** 882–900.
- D. NADDEF AND G. RINALDI (1991) “The symmetric traveling salesman polytope and its graphical relaxation: composition of valid inequalities”, *Mathematical Programming* **51** 359–400.
- D. NADDEF AND G. RINALDI (1992) “The crown inequalities for the symmetric traveling salesman polytope”, *Mathematics of Operations Research* **17** 308–326.
- N.R. NATRAJ, R.R. THOMAS AND S.R. TAYUR (1994) “An algebraic geometry algorithm for scheduling in presence of setups and correlated demands”, *Mathematical Programming* (to appear).
- G.L. NEMHAUSER AND S. PARK (1991) “A polyhedral approach to edge coloring”, *Operations Research Letters* **10** 315–322.
- G.L. NEMHAUSER AND G. SIGISMONDI (1992) “A strong cutting plane/branch-and-bound algorithm for node packing”, *Journal of the Operational Research Society* **43** 443–457.
- G.L. NEMHAUSER AND L.E. TROTTER (1974) “Properties of vertex packings and independence system polyhedra”, *Mathematical Programming* **6** 48–61.
- G.L. NEMHAUSER AND P.H. VANCE (1994) “Lifted cover facets of the 0-1 knapsack polytope with GUB constraints”, *Operations Research Letters* **16** 255–263.
- G.L. NEMHAUSER AND L.A. WOLSEY (1988) *Integer and Combinatorial Optimization*, Wiley, New York.
- G.L. NEMHAUSER AND L.A. WOLSEY (1990) “A recursive procedure to generate all cuts for 0-1 mixed integer programs”, *Mathematical Programming* **46** 379–390.
- P. NOBILI AND A. SASSANO (1989) “Facets and lifting procedures for the set covering polytope”, *Mathematical Programming* **45** 111–147.
- M.W. PADBERG (1973) “On the facial structure of set packing polyhedra”, *Mathematical Programming* **5** 199–215.
- M.W. PADBERG (1977) “On the complexity of set packing polyhedra”, [in: *Studies in Integer Programming* (P.L. Hammer, E.L. Johnson, B.H. Korte and G.L. Nemhauser, eds.)] *Annals of Discrete Mathematics* **1** 421–434.

- M.W. PADBERG (1980) “ $(1, k)$ -configurations and facets for packing problems”, *Mathematical Programming* **18** 94–99.
- M.W. PADBERG (1989) “The boolean quadratic polytope: some characteristics, facets and relatives”, *Mathematical Programming* **45** 139–172.
- M.W. PADBERG AND M. GRÖTSCHEL (1985) “Polyhedral computations” in: *Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization* (E. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys, eds.), Wiley and Sons.
- M.W. PADBERG AND S. HONG (1980) “On the symmetric traveling salesman problem”, *Mathematical Programming Studies* **12** 78–107.
- M.W. PADBERG AND G. RINALDI (1987) “Optimization of a 532-city symmetric traveling salesman polytope by branch and cut”, *Operations Research Letters* **6** 1–7. [Addendum **9** (1990) 353.]
- M.W. PADBERG AND G. RINALDI (1990) “Facet identification for the symmetric traveling salesman problem”, *Mathematical Programming* **47** 219–257.
- M.W. PADBERG AND G. RINALDI (1991) “A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems”, *Siam Review* **33** 60–100.
- M.W. PADBERG, T.J. VAN ROY AND L.A. WOLSEY (1985) “Valid inequalities for fixed charge problems”, *Operations Research* **33** 842–861.
- U.N. PELED (1977) “Properties of facets of binary polytopes”, *Annals of Discrete Mathematics* **1** 435–455.
- Y. POCHET AND L.A. WOLSEY (1992) “Network design with divisible capacities: aggregated flow and knapsack subproblems”, in: *Proceedings of the Second Conference on Integer Programming and Combinatorial Optimization* (E. Balas, G. Cornuéjols and R. Kannan, eds.)
- Y. POCHET AND L.A. WOLSEY (1994) “Algorithms and reformulations for lot sizing problems”, CORE Discussion Paper 9427, Louvain-la-Neuve.
- W.R. PULLEYBLANK AND F.B. SHEPHERD (1993) “Formulations for the stable set polytope of a claw-free graph”, in: *Proceedings of the Third Conference on Integer Programming and Combinatorial Optimization* (G. Rinaldi and L.A. Wolsey, eds.)
- M. QUEYRANNE AND A. SCHULZ (1994) “Polyhedral approaches to machine scheduling”, Working Paper No. 408/1994, Fachbereich 3 Mathematik, Technische Universität Berlin, Berlin.
- G. REINELT (1985) *The Linear Ordering Problem: Algorithms and Applications*, Helderman, Berlin.
- G. REINELT (1991) “TSPLIB—A traveling salesman library”, *ORSA Journal on Computing* **3** 376–384.
- T.J. VAN ROY AND L.A. WOLSEY (1985) “Valid inequalities and separation for uncapacitated fixed charge networks”, *Operations Research Letters* **4** 105–112.
- T.J. VAN ROY AND L.A. WOLSEY (1986) “Valid inequalities for mixed 0-1 programs”, *Discrete Applied Mathematics* **14** 199–213.
- T.J. VAN ROY AND L.A. WOLSEY (1987) “Solving mixed integer programming problems using automatic reformulation”, *Operations Research* **35** 45–57.
- A. SASSANO (1989) “On the facial structure of the set covering problem”, *Mathematical Programming* **44** 181–202.
- M.W.P. SAVELSBERGH (1993) “A branch-and-price algorithm for the generalized assignment problem”, Report COC-93-02, Computational Optimization Center, Georgia Institute of Technology, Atlanta.
- M.W.P. SAVELSBERGH (1994) “Preprocessing and probing techniques for mixed integer programming”, *ORSA Journal on Computing* **6** 445–454.
- A. SCHRIJVER (1980) “On cutting planes”, [in: *Combinatorics 79 Part II* (M. Deza and I.G. Rosenberg, eds.)] *Annals of Discrete Mathematics* **9** 291–296.

- A. SCHRIJVER (1981) “On total dual integrality”, *Linear Algebra and Its Applications* **38** 27–32.
- A. SCHRIJVER (1986) *Theory of Linear and Integer Programming*, Wiley, Chichester.
- P.D. SEYMOUR (1980) “Decomposition of regular matroids”, *Journal of Combinatorial Theory (B)* **28** 305–359.
- B. STURMFELS AND R.R. THOMAS (1994) “Variation of cost functions in integer programming”, Manuscript, Cornell University, Ithaca.
- R.R. THOMAS (1992) “A geometric Buchberger algorithm for integer programming”, Manuscript, Cornell University, Ithaca.
- L. TROTTER (1975) “A class of facet-producing graphs for vertex packing polytopes”, *Discrete Mathematics* **12** 373–388.
- F. VANDERBECK AND L.A. WOLSEY (1994) “An exact algorithm for IP column generation”, CORE Discussion Paper 9419, Louvain-la-Neuve.
- R. WEISMANTEL (1994) “On the 0/1 knapsack polytope”, preprint SC 94-1, Konrad-Zuse-Zentrum für Informationstechnik, Berlin.
- L.A. WOLSEY (1975) “Faces for a linear inequality in 0-1 variables”, *Mathematical Programming* **8** 165–178.
- L.A. WOLSEY (1976a) “Facets and strong valid inequalities for integer programs”, *Operations Research* **24** 367–372.
- L.A. WOLSEY (1976b) “Further facet generating procedures for vertex packing polytopes”, *Mathematical Programming* **11** 158–163.
- E. ZEMEL (1978) “Lifting the facets of zero-one polytopes” *Mathematical Programming* **15** 268–277.