

Polyhedral Techniques in Combinatorial Optimization II: Computations

Karen Aardal

Department of Computer Science
Utrecht University
P.O. Box 80089
3508 TB Utrecht, The Netherlands
aardal@cs.ruu.nl

Stan van Hoesel

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

Abstract

The polyhedral approach is one of the most powerful techniques available for solving hard combinatorial optimization problems. The main idea behind the technique is to consider the linear relaxation of the integer combinatorial optimization problem, and try to iteratively strengthen the linear formulation by adding violated *strong valid inequalities*, i.e., inequalities that are violated by the current fractional solution but satisfied by all feasible solutions, and that define high-dimensional faces, preferably facets, of the convex hull of feasible solutions. If we have the complete description of the convex hull of feasible solutions all extreme points of this formulation are integral, which means that we can solve the problem as a linear programming problem. Linear programming problems are known to be computationally easy. In Part I of this article we discuss theoretical aspects of polyhedral techniques. Here we will mainly concentrate on the computational aspects. In particular we discuss how polyhedral results are used in cutting plane algorithms. We also consider a few theoretical issues not treated in Part I. We conclude the article by briefly mentioning some alternative techniques for solving combinatorial optimization problems.

1 Introduction

A *combinatorial optimization problem* is an integer linear programming (ILP) problem

$$\min\{cx : Ax \leq b, x \in \mathbb{Z}^n\} \tag{1}$$

having a combinatorial character. A well-known combinatorial optimization problem is the *traveling salesman problem*, i.e., the problem of finding the shortest tour through a number of cities such that each city is visited precisely once. The only known method for solving the traveling salesman problem is an enumerative method, such as branch-and-bound. If the lower bound obtained by solving the linear relaxation of the problem is not very close to the optimal value, we can expect that the branch-and-bound tree will grow too big to be able to solve any realistic instance. The idea behind *polyhedral combinatorics* is to obtain a strong lower bound on the optimal solution value by finding a good linear formulation of the set $X = \{x \in \mathbb{Z}^n : Ax \leq b\}$ of feasible solutions. This is done by adding linear inequalities that are necessary in the description of the convex hull of X to the original linear formulation $Ax \leq b$. The convex hull of X is the smallest convex set containing all points in X . The advantage with this approach is that if the convex hull of X is known, we can solve $\min\{cx : x \in \text{conv}(X)\}$ as a linear programming problem, which is computationally easy, but gives the same solution as optimizing over X .

As discussed in Part I of this article (Aardal and Van Hoesel (1995b)) it is hard in general to describe the convex hull of X by concise families of inequalities even if we allow for classes containing exponentially many inequalities. In a practical setting however the complete description of the convex hull of X is not needed. What is important is that we have a good description of the region close to the optimal solution, which suggests an approach where we generate linear inequalities as they are needed. Such an algorithm is usually called a *cutting plane algorithm*, and is typically embedded in a branch-and-bound procedure to produce good lower bounds. To make a cutting plane algorithm work we essentially need to consider three issues: (i) develop families of strong valid inequalities, (ii) develop separations algorithms, i.e., algorithms for identifying violated inequalities belonging to the various families, and (iii) designing an efficient implementation of the complete framework, including a branch-and-bound algorithm, a preprocessor, primal heuristics for finding good feasible solutions, and a branching strategy for the branch-and-bound phase.

In Part I we considered some important theoretical aspects of polyhedral combinatorics and cutting plane techniques. Questions that we were asking were for instance: Is there an algorithmic way to generate all inequalities necessary to describe the convex hull of feasible solutions? When can we expect to be able to describe the convex hull of feasible solutions using concise families of valid inequalities? How difficult is it to identify a violated linear inequality? In this article we shall mainly study computational aspects of polyhedral techniques, even though we study a few techniques and theoretical issues that were not treated in Part I.

When developing families of valid inequalities for a certain problem P it is useful to consider valid inequalities for relaxations of P , since an inequality that is valid for a relaxation of P is also valid for P . Once we gain insight in the facial structure of the simpler relaxation we

can try to develop more problem specific inequalities. Moreover, we can include separation algorithms for valid inequalities for several common relaxations of integer and combinatorial optimization problems, and algorithms for automatically detecting the structure of these relaxations, as a basic feature in our cutting plane algorithm. In Part I we describe families of inequalities for a few generic combinatorial structures. In Section 2 we will make a more extensive presentation, and illustrate some useful techniques. One such technique is *lifting*, which is used if we have a valid inequality for a set of solutions projected onto a subspace. By using lifting on the inequality we obtain a valid inequality for the full space. We will also discuss two ways of proving that an inequality is facet defining, and how to prove that a given set of families of valid inequalities generates the convex hull of feasible solutions. We conclude the section by giving 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 cutting plane techniques 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. Due to logical implications it may also be possible to delete some variables, which reduces the size of the problem formulation.

When applying a cutting plane algorithm we in general 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. In a 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. Preprocessing and other implementation 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 solving 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. In Section 5, we briefly mention alternative approaches to solving integer and combinatorial optimization problems.

2 Polyhedral Results for Generic Combinatorial Structures

Some combinatorial structures occur as substructures in a large number of combinatorial optimization problems. The study of these generic structures has two purposes. First, a valid inequality for a generic structure may form the theoretical starting point for developing families of inequalities for more specialized problems. Second, valid inequalities for generic structures are often effective for more specialized problems as well, so implementing separation

algorithms for generic inequalities is very useful in general-purpose cutting plane algorithms.

Here we present some well-known generic structures, extending our presentation in Part I. When discussing various families of inequalities we also take the opportunity to describe different techniques, such as facet proving techniques, techniques for proving that certain families of inequalities define the convex hull of feasible solutions, and lifting.

2.1 Preliminaries

Here we introduce basic definitions that are needed to understand the terminology used in subsequent sections. For the reader that wishes to study integer programming and polyhedral combinatorics more in detail, we refer to Schrijver (1986) and Nemhauser and Wolsey (1988).

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 $\alpha_k = 0$, for all 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* $AS = \{\sum_{k=0}^K \alpha_k x^k : \alpha \in \mathbb{R}^{K+1}; \sum_{k=0}^K \alpha_k = 1\}$. An affine space can be viewed as a linear space translated over a vector x^0 : $AS = \{x^0 + \sum_{k=1}^K \beta_k (x^k - x^0) : \beta \in \mathbb{R}^K\}$. 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 $\alpha_k = 0$, for all 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 a smallest (by inclusion) affine space containing P . A bounded polyhedron is called a *polytope*.

An inequality $\pi x \leq \pi_0$ is called *valid* for a polyhedron 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 called *proper* if $\emptyset \neq F$ and $F \neq P$. The dimension of a proper face F , $\dim(F)$, is strictly smaller than $\dim(P)$. If $\dim(F) = \dim(P) - 1$, i.e., if F is maximal, we call F a *facet*. The reason why we are interested in the facet defining inequalities is that they are precisely the inequalities that we need to describe the convex hull of feasible solutions, in addition to the set of inequalities that are satisfied with equality by every feasible point.

The *convex hull of feasible solutions*, denoted $\text{conv}(X)$, is the set of points that can be

obtained by taking convex combinations of points in X , i.e.,

$$\text{conv}(X) = \left\{ \sum_{k=1}^K \lambda_k x^k : X = \{x^k\}_{k=1}^K, \sum_{k=1}^K \lambda_k = 1, \lambda_k \geq 0, k = 1, \dots, K \right\}.$$

2.2 The Vertex Packing Problem

Here we describe two classes of valid inequalities for the vertex packing problem. We also give an example of an easy facet proof and illustrate *lifting* techniques. Lifting is an iterative technique where we start with an inequality that is valid under the condition that a subset N of the variables are fixed. At each iteration a subset $M \subseteq N$ of the fixed variables are included in the inequality with coefficients that guarantee that the resulting inequality is valid. In *sequential lifting* the set M consists of one variable at each iteration, whereas in *simultaneous lifting* there are no restrictions on the choice of M . Typically we have $M = N$.

Consider an undirected graph $G = (V, E)$ where V is the set of vertices and E the set of edges, i.e., unordered pairs of vertices. A *vertex packing* in G is a subset $V' \subseteq V$ of vertices such that no two vertices in V' are adjacent. We define variables x_v for each vertex v , and let $x_v = 1$ if $v \in V'$ and $x_v = 0$ otherwise. The integer programming formulation of the maximum cardinality vertex packing problem is given below.

$$\begin{aligned} \max \quad & \sum_{v \in V} x_v \\ \text{s.t.} \quad & x_v + x_w \leq 1 && \text{for all } \{v, w\} \in E && (2) \\ & x_v \in \{0, 1\} && \text{for all } v \in V && (3) \end{aligned}$$

The vertex packing problem is sometimes referred to as the *independent set problem* or the *stable set problem*. Let X_{VPG} be the set of incidence vectors corresponding to feasible vertex packings 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 1 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 subgraph of G where each two vertices are connected by an edge, see Figure 1a. Since no two vertices 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

$$\sum_{j \in C} x_j \leq 1. \tag{4}$$

Theorem 2 Padberg (1973). *Let C be a clique in the graph G . The clique inequality (4) defines a facet of $\text{conv}(X_{VPG})$ if and only if C is maximal.*

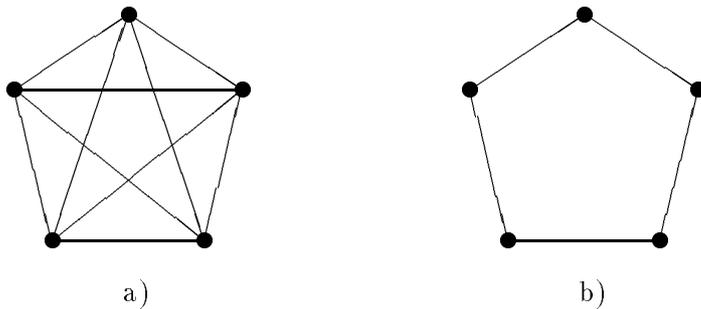


Figure 1: a) A clique. b) An odd hole.

Proof. Sufficiency: The dimension of the vertex packing polytope is $|V|$. Hence, to prove that (4) defines a facet of $\text{conv}(X_{VPG})$ we need to find $|V|$ affinely independent points that are tight for (4). 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 1b. 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.

$$\sum_{j \in H} x_j \leq \frac{|H| - 1}{2} \quad (5)$$

Padberg (1973) showed that (5) defines a facet of $\text{conv}(X_{VPG} \cap \{x \in \{0, 1\}^{|V|} : x_j = 0 \text{ for all } j \notin H\})$, i.e., in general (5) 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 (5) such that (5) becomes a facet for $\text{conv}(X_{VPG})$. One way of increasing this dimension is through *sequential lifting* (Padberg (1973), and Wolsey (1976a)), which is illustrated in the following example.

The graph in Figure 2 is an odd hole with a central vertex adjacent to all vertices of the hole. This structure, called a *wheel*, is used to illustrate the sequential lifting procedure. The inequality $x_1 + x_2 + x_3 + x_4 + x_5 \leq 2$ defines a facet of $\text{conv}(X_{VPG} \cap \{x \in \{0, 1\}^6 : x_6 = 0\})$. The problem is to determine the maximum nonnegative value of the constant α such that $x_1 + x_2 + x_3 + x_4 + x_5 + \alpha x_6 \leq 2$ is a valid inequality of $\text{conv}(X_{VPG})$. If $x_6 = 0$, α 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. Hence, the maximal value of α is $\alpha = 2$. The following two

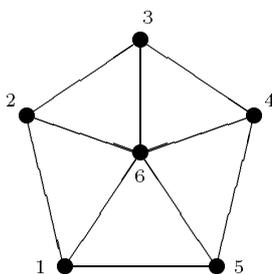


Figure 2: A wheel.

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

$$\sum_{j=2}^n \pi_j x_j \leq \pi_0 \quad (6)$$

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

$$\alpha x_1 + \sum_{j=2}^n \pi_j x_j \leq \pi_0 \quad (7)$$

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 (6) defines a face of $\text{conv}(S^0)$ of dimension k , and if α is chosen maximal, then (7) defines a face of $\text{conv}(S)$ of dimension $k + 1$.

Theorem 4 Wolsey (1976a). *Let $S \subseteq \{0, 1\}^n$. Suppose*

$$\sum_{j=2}^n \pi_j x_j \leq \pi_0 \quad (8)$$

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

$$\beta x_1 + \sum_{j=2}^n \pi_j x_j \leq \pi_0 + \beta_1 \quad (9)$$

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 (8) defines a face of $\text{conv}(S^1)$ of dimension k , and if β is chosen minimal, then (9) defines a face of $\text{conv}(S)$ of dimension $k + 1$.

Sequential lifting is sequence dependent, such that different lifting sequences give rise to different inequalities. Zemel (1978) proposed a more general lifting procedure, called *simultaneous lifting*. As the name indicates, the coefficients of several variables are determined simultaneously, yielding inequalities that in general cannot be obtained by sequential lifting.

For notational ease we consider the case where a set of variables that are all fixed to zero are simultaneously lifted. Let $S \subseteq \{0, 1\}^n$, and suppose that the inequality

$$\sum_{j=k+1}^n \pi_j x_j \leq \pi_0 \quad (10)$$

is valid for $S^0 = S \cap \{x \in \{0, 1\}^n : x_1 = x_2 = \dots = x_k = 0\}$. If $S \cap \{x \in \{0, 1\}^n : \sum_{j=1}^k x_j \geq 1\} \neq \emptyset$, then

$$\sum_{j=1}^k \alpha_j x_j + \sum_{j=k+1}^n \pi_j x_j \leq \pi_0 \quad (11)$$

is valid for S if for any $(x'_1, \dots, x'_k) \neq 0$, we have that $\sum_{j=1}^k \alpha_j x'_j \leq \pi_0 - \max_{S \cap \{x: x_1=x'_1, \dots, x_k=x'_k\}} \{\sum_{j=k+1}^n \pi_j x_j\}$. Hence, the feasible vectors $\alpha = (\alpha_1, \dots, \alpha_k)$ define a polyhedral set. The extreme points of this set yield new facet defining inequalities in general if the original inequality is facet defining. For more details, see Zemel (1978). We will give an example of a vertex packing problem where simultaneous lifting of an odd-hole inequality yields an inequality that cannot be obtained by sequential lifting. The left part of Figure 3 shows the original underlying graph in dotted lines, and the odd hole in solid lines. The right figure illustrates the structure corresponding to the inequality that is obtained by simultaneously lifting the variables corresponding to vertices 8, 14, 20, 21, and 27. The coefficients of all lifted variables are equal to 0.5. The resulting inequality is facet defining and cannot be obtained by sequential lifting.

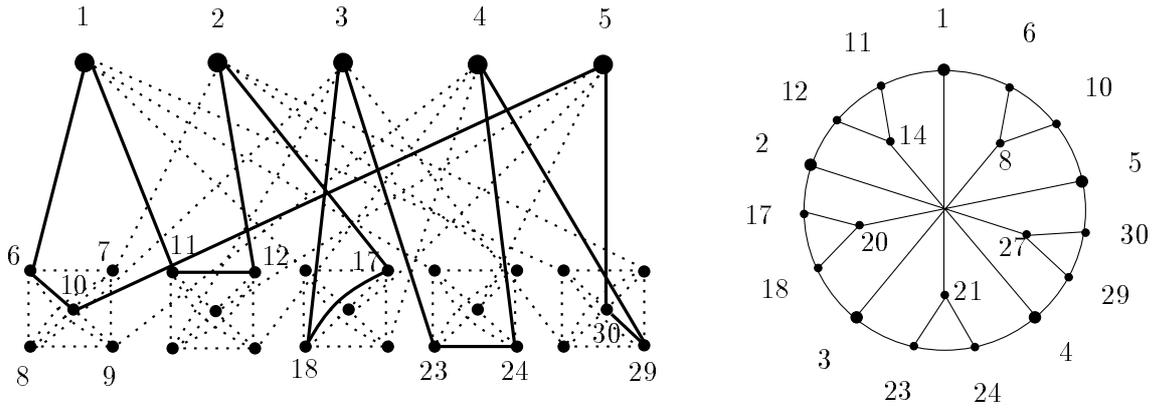


Figure 3: Simultaneous lifting of an odd-hole inequality.

If we apply sequential lifting to the same initial odd-hole inequality we obtain the lifting illustrated in Figure 4 or any of the four liftings that can be obtained by rotating the figure. For more details on lifting procedures, see Nemhauser and Wolsey (1988).

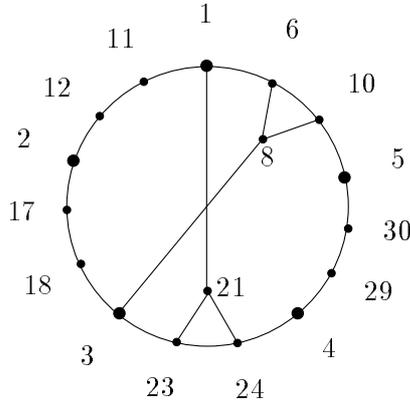


Figure 4: Sequential lifting of an odd-hole inequality.

2.3 The Knapsack Problem

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

$$\max \sum_{j \in N} c_j x_j$$

$$\text{s.t.} \quad \sum_{j \in N} a_j x_j \leq b \quad (12)$$

$$x_j \in \{0, 1\} \quad \text{for all } j \in N. \quad (13)$$

The knapsack problem occurs as a substructure of several combinatorial optimization problem having a capacity or budget constraint. Assume that the vectors c , a and the right-hand side b are rational, and let X_K be the set of incidence vectors corresponding to the feasible solutions to the knapsack problem. Let C be a subset of N such that $\sum_{j \in C} a_j > b$, and such that C is minimal with respect to this property, i.e., $\sum_{j \in S} a_j \leq b$ for all $S \subset C$. We call the set C a *minimal cover*. In Part I we described the family of *knapsack cover inequalities* (Balas (1975), Hammer et al. (1975) and Wolsey (1975))

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

The inequalities (14) are valid for X_K since, if we include all items in C in the knapsack, we exceed the right-hand side b , which means that we have to exclude at least one of the elements in C .

In Part I we discussed the special case of a lifted cover inequality that is obtained if we consider 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 lifting coefficients of the variables in $E(C) \setminus C$ are all equal to one. 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 3 and 4, we can conclude that $\text{conv}(X_K)$ has a facet of the following form

$$\sum_{j \in N \setminus N'} \alpha_j x_j + \sum_{j \in N' \setminus C'} \beta_j x_j + \sum_{j \in C'} x_j \leq |C'| - 1 + \sum_{j \in N' \setminus C'} \beta_j, \quad (15)$$

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 α_j in the case where $N' \setminus C' = \emptyset$.

The family of $(1, k)$ -*configuration inequalities* (Padberg (1980)) is defined as follows. 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

$$(r - k + 1)x_t + \sum_{j \in T(r)} x_j \leq r \quad (16)$$

is valid for $\text{conv}(X_K)$. If $k = |\bar{C}|$ the knapsack cover inequality (14) is obtained. The $(1, k)$ -configuration inequality (16) can be obtained by the following lifting procedure. We start with the inequality

$$\sum_{j \in T(r)} x_j \leq k - 1, \quad (17)$$

which is valid for $X_K \cap \{x \in \{0, 1\}^n : x_t = 1\}$. The maximal lifting coefficient of variable x_t is equal to $r - k + 1$.

2.4 The Fixed Charge Uncapacitated Flow Problem

Here we consider a general class of valid inequalities for the uncapacitated fixed-charge flow polytope. Consider a directed graph $G = (V, A)$, and let x_{ij} denote the flow on arc $(i, j) \in A$. If arc (i, j) is used we have to pay a fixed cost. Therefore, let $y_{ij} = 1$ if arc (i, j) is opened, and let $y_{ij} = 0$ otherwise. Each node i has a known outflow d_i . If d_i is negative it means that node i has an inflow. We use d_i^+ to denote $\max\{0, d_i\}$. The polytope X_{UFC} is defined as the set of vectors corresponding to the solutions satisfying the following constraints.

$$\sum_{\{j:(i,j) \in A\}} x_{ij} - \sum_{\{k:(k,i) \in A\}} x_{ki} = d_i \quad \text{for all } i \in V \quad (18)$$

$$0 \leq x_{ij} \leq M y_{ij} \quad \text{for all } (i, j) \in A \quad (19)$$

$$y_{ij} \in \{0, 1\} \quad \text{for all } (i, j) \in A \quad (20)$$

In constraint (19) M is a large enough positive number, which we need in order to enforce $y_{ij} = 1$ if $x_{ij} > 0$. Typically $M = \sum_{i \in V} d_i^+$.

Let $X \subset V$, and let $E(X)$ be the subset of arcs for which both endpoints belong to X , i.e., $E(X) = \{(i, j) \in A : i, j \in X\}$. Moreover, let $(X, \bar{X}) = (X, V \setminus X) = \{(i, j) \in A : i \in X, j \in V \setminus X\}$. Consider the subset $C \subseteq (\bar{X}, X)$, and the subset $R \subseteq E(X)$, and let $H = \{j : (i, j) \in$

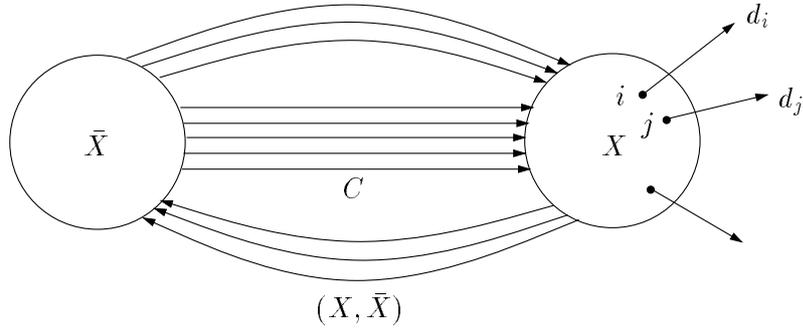


Figure 5: The fixed charge uncapacitated flow model.

C . Let $V_j^R = \{j\} \cup \{k \in X : \text{there exists a directed path from } j \text{ to } k \text{ using only arcs of } R\}$ for $j \in H$. The flow model is illustrated in Figure 5.

The following family of valid inequalities for X_{UFC} was developed by Van Roy and Wolsey (1985).

$$\sum_{(i,j) \in C} x_{ij} \leq \sum_{(i,j) \in C} \left(\sum_{k \in V_j^R} d_k^+ \right) y_{ij} + \sum_{(i,j) \in E(X) \setminus R} x_{ij} + \sum_{(i,j) \in (X, \bar{X})} x_{ij} \quad (21)$$

The intuition behind the inequalities is as follows. The flow on the arcs in the subset $C \subseteq (\bar{X}, X)$ either flows along arcs in R , or arcs in $E(X) \setminus R$ or arcs in (X, \bar{X}) . The flow that goes along arcs in R is limited by the outflow $\sum_k d_k$, where node k is a node that can be reached by a directed path from node j , such that $(i, j) \in C$. This explains the coefficient of the y_{ij} -variables. A subclass of the inequalities (21) is the family of (l, S) -inequalities for the economic lot-sizing problem presented in Section 2.6.1.

The separation problem based on the network inequalities (21) is difficult in general as we need to simultaneously choose sets X , R and C . Van Roy and Wolsey treated three special cases where it is possible to generate a violated network inequality (21), or a weakened version of it, in polynomial time. Let (x^*, y^*) denote a fractional solution. The easiest case is when the sets X and R are known and we only need to identify a set C . Here we only have to evaluate $\sum_{(i,j) \in (\bar{X}, X)} (x_{ij}^* - (\sum_{k \in V_j^R} d_k^+) y_{ij}^*)$, and let $C = \{(i, j) \in (\bar{X}, X) : x_{ij}^* > (\sum_{k \in V_j^R} d_k^+) y_{ij}^*\}$. In the second case the set C is fixed and we look for a set X assuming that $R = E(X)$. The best choice of X can be found using a maximum flow algorithm. The third case deals with a slight modification of (21) developed for the generalization of the flow model (18)-(20) in which we have arcs with an upper bound, i.e., $0 \leq x_{kl} \leq m_{kl}$ for $(k, l) \in (E(X) \setminus R) \cup (X, \bar{X})$. In the modified inequality we want to replace d_k^+ with $(d_k + m_{kl})^+$ for the subset $Q \subseteq (E(X) \setminus R) \cup (X, \bar{X})$ of the arcs. If the sets X and R are fixed, it is possible to find the best choice of Q and C in polynomial time.

2.5 The Single-Node Capacitated 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 on 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.

$$\sum_{j \in N} x_j = b \quad (22)$$

$$0 \leq x_j \leq m_j y_j \quad \text{for all } j \in N \quad (23)$$

$$y_j \in \{0, 1\} \quad \text{for all } j \in N \quad (24)$$

Let X_{FC} denote the set of vectors corresponding to feasible solutions to (22)–(24). In this section we will discuss the following topics. First, we will describe the basic *flow cover inequality* that is valid for X_{FC} , and show that this inequality is facet defining. We will use a different proof technique compared to the one used to prove Theorem 2. Next, we will discuss the separation problem based on the family of flow cover inequalities. Once we have a flow cover inequality we can extend it. We describe the result that if $m_j = m$ for all $j \in N$, then the extended flow cover inequalities, together with the defining inequalities define the convex hull of feasible solutions. Moreover, if all capacities are equal, then the extended flow cover inequalities can be separated in polynomial time. An application of the flow cover inequalities is shown in Section 2.5.

2.5.1 The Flow Cover Inequalities

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 flow 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

$$\sum_{j \in J} x_j \leq b - \sum_{j \in J} (m_j - \lambda)^+ (1 - y_j). \quad (25)$$

In Figure 6 we illustrate the single-node capacitated flow model.

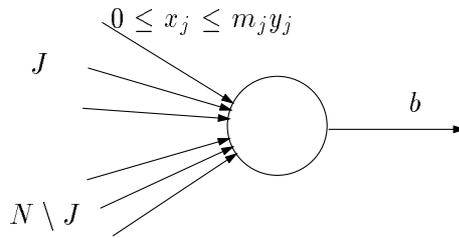


Figure 6: The single-node capacitated flow model with sets J and $N \setminus J$.

Theorem 5 Padberg, Van Roy and Wolsey (1985). *Assume that $\sum_{j \in N} m_j - m_r \geq b$ for all $r \in N$, and that $J \subset N$. The flow cover inequality (25) defines a facet of $\text{conv}(X_{FC})$ if and only if $\max_{j \in J} \{m_j\} > \lambda$.*

Proof. To prove that inequality (25) defines facets we will use a slightly different technique than the one we use to prove Theorem 2. The method used here is often referred to as the *indirect* method, see Nemhauser and Wolsey (1988), Chapter I.4, Theorem 3.6.

To show that inequality (25) is facet defining we need to show that (25) does not define an improper face, i.e., that there exists a feasible point such that inequality (25) is satisfied with strict inequality at this point. We also need to prove that the *only* inequality that is satisfied with equality by all points $(x, y) \in X_{FC}$ that lie on the hyperplane $\sum_{j \in J} x_j = b - \sum_{j \in J} (m_j - \lambda)^+(1 - y_j)$, is inequality (25) plus $\gamma(\sum_{j \in N} x_j = b)$, where γ is an arbitrarily chosen scalar. If there were more inequalities of this sort it would mean that these inequalities all define faces of lower dimension. Also, note that the polytope $\text{conv}(X_{FC})$ is not full-dimensional as all feasible points satisfy $\sum_{j \in N} x_j = b$. Therefore, we have to add $\gamma(\sum_{j \in N} x_j = b)$ to inequality (25) in the facet proof.

To see that inequality (25) does not define an improper face, we consider any feasible point for which the following holds.

$$\begin{aligned} y_j &= 1 && \text{for all } j \in N \\ \sum_{j \in J} x_j &= b - \varepsilon \\ \sum_{j \in N \setminus J} x_j &= \varepsilon \end{aligned}$$

Such a point is possible to construct since J defines a cover, and since J is a proper subset of N . Since inequality (25) is satisfied with strict inequality at this point, we have showed that inequality (25) does not define an improper face.

Next, we need to show that if all points $(x, y) \in X_{FC}$ that are tight for (25) satisfy

$$\sum_{j \in N} \alpha_j x_j + \sum_{j \in N} \beta_j y_j = \alpha_0, \quad (26)$$

then

- (i) $\beta_j = 0$ for all $j \in N \setminus J$
- (ii) $\alpha_j = \gamma$ for all $j \in N \setminus J$
- (iii) $\alpha_j = \bar{\alpha} + \gamma$ for all $j \in J$
- (iv) $\beta_j = -\bar{\alpha}(m_j - \lambda)^+$ for all $j \in J$
- (v) $\alpha_0 = \bar{\alpha}(b - \sum_{j \in J} (m_j - \lambda)^+) + \gamma b$.

To show that $\beta_j = 0$ for all $j \in N \setminus J$, we consider any feasible solution in which $\sum_{j \in J} x_j = b$, and $y_j = 1$ for all $j \in J$ and all $(N \setminus J) \setminus \{l\}$, where l is chosen arbitrarily. First, let $y_l = 0$, and then create a new solution where every variable takes the same value as in the first solution, except y_l , which now takes value one. Evaluating (26) at both solutions and comparing the two expressions gives $\beta_l = 0$. Since arc l was chosen arbitrarily in $N \setminus J$, we can conclude that $\beta_j = 0$ for all $j \in N \setminus J$.

Next, we show that $\alpha_j = \gamma$ for all $j \in N \setminus J$. Here we first consider a solution in which we close the arc $k \in J$ with largest capacity. Due to the assumptions of the theorem we know that $m_k > \lambda$. Let $y_j = 1$ for all $j \in N \setminus \{k\}$, $\sum_{j \in J \setminus \{k\}} x_j = b - (m_k - \lambda)$, and let $\sum_{j \in N \setminus J} x_j = (m_k - \lambda)$. Furthermore, assume that $x_j < m_j$ for all $j \in N \setminus J$. The second solution we consider is constructed as follows. Choose arbitrarily two arcs $j', j'' \in N \setminus J$. Let all variables take the same values as in the first point except that we increase the flow by ε on arc j' and decrease the flow by ε on arc j'' . Comparing the expressions obtained by evaluating (26) at the two solutions gives $\alpha_{j'} = \alpha_{j''}$. Since j' and j'' were arbitrarily chosen, we have $\alpha_j = \alpha^1 = \gamma$ for all $j \in N \setminus J$.

To show that $\alpha_j = \bar{\alpha} + \gamma$ for all $j \in J$ we need to consider a solution in which all arcs in N are open, and in which $\sum_{j \in J} x_j = b$ and $x_j < m_j$ for all $j \in J$. Now we can choose any two arcs $j', j'' \in J$ and make an ε -change of flow as in the previous step of the proof. This shows that $\alpha_{j'} = \alpha_{j''}$. Varying over all choices of j', j'' gives $\alpha_j = \alpha^2$. Assume that $\alpha^2 = \bar{\alpha} + \gamma$. If $\bar{\alpha} = 0$, then inequality (26) would be trivial. Hence, assume that $\bar{\alpha} > 0$.

We have now reduced equality (26) to the following expression

$$(\bar{\alpha} + \gamma) \sum_{j \in J} x_j + \gamma \sum_{j \in N \setminus J} x_j + \sum_{j \in J} \beta_j y_j = \alpha_0. \quad (27)$$

By evaluating (27) at any feasible solution where $y_j = 1$ for all $j \in N$ that is tight for (25), and any tight feasible solution where one arc $k \in J$ is closed and all other arcs are open, we get $-\bar{\alpha}(m_k - \lambda)^+ - \beta_k = 0$. Varying over all possible choices of k gives

$$\beta_j = -\bar{\alpha}(m_j - \lambda)^+ \text{ for all } j \in J. \quad (28)$$

Finally we need to determine the value of α_0 . By using the value (28) of β_j for all $j \in J$ in equation (27), and by evaluating (27) at any feasible point that is tight for (25) we obtain

$$\alpha_0 = \bar{\alpha}(b - \sum_{j \in J} (m_j - \lambda)^+) + \gamma b,$$

which completes our proof. ■

2.5.2 Separation Based on the Flow Cover Inequalities

Let (x^*, y^*) denote a fractional solution to the LP-relaxation of the single-node capacitated flow problem (22)-(24). Moreover, let $z_j = 1$ if $j \in J$ and let $z_j = 0$ otherwise. For a given value of $\lambda > 0$, the separation problem based on the family of flow cover inequalities (25) is formulated as the following knapsack type problem.

$$\max \sum_{j \in N} [x_j^* + (m_j - \lambda)^+(1 - y_j^*)] z_j \quad (29)$$

$$\text{s.t.} \quad \sum_{j \in N} m_j z_j = b + \lambda \quad (30)$$

$$z_j \in \{0, 1\} \quad \text{for all } j \in N. \quad (31)$$

Crowder et al. (1983), and Van Roy and Wolsey (1987) discuss a heuristic for problem (29)-(31).

2.5.3 The Extended Flow Cover Inequalities

Once we have a set J satisfying the conditions of Theorem 5 we can extend the flow cover inequality by including flow from the arcs belonging to the set $L \subseteq (N \setminus J)$, see Figure 7.

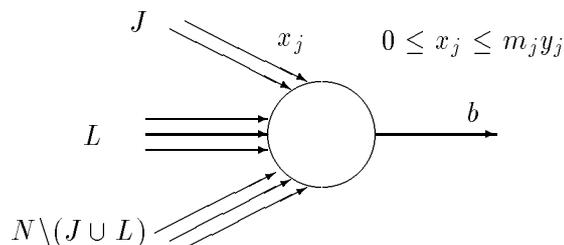


Figure 7: The single-node capacitated flow model with sets J , L , and $N \setminus (J \cup L)$.

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 $\text{conv}(X_{FC})$.

$$\sum_{j \in J \cup L} x_j \leq b - \sum_{j \in J} (m_j - \lambda)^+(1 - y_j) + \sum_{j \in L} (\bar{m}_j - \lambda)y_j \quad (32)$$

It is only interesting to include arc l in L if $m_l > \bar{m} - \lambda$, since we otherwise obtain a stronger inequality by combining (32) with $L \setminus \{l\}$ with the defining constraint $x_l \leq m_l y_l$. Padberg et al. (1985) showed that if $0 < \bar{m} - \lambda < m_l \leq \bar{m}$ then the extended flow cover inequality (32) defines a facet of $\text{conv}(X_{FC})$, and obtained the following result in the equal capacity case. Let X_{FC}^C denote the set of vectors corresponding to feasible solutions to (22)-(24) if $m_j = m$ for all $j \in N$.

Theorem 6 Padberg, Van Roy, and Wolsey (1985). *Assume that $m_j = m$ for all $j \in N$, and that b is not an integer multiple of m . Let $l = \lceil b/m \rceil$, and $\lambda = ml - b$. Let S be any flow cover with respect to N , i.e., $|S| \geq l$. The extended flow cover inequalities*

$$\sum_{j \in S} x_j \leq b + \sum_{j \in S} (m - \lambda)y_j - (m - \lambda)l \quad (33)$$

together with the defining inequalities (22), (23) with $m_j = m$ for all $j \in N$, and the inequalities $0 \leq y_j \leq 1$ for all $j \in N$ define the convex hull of X_{FC}^C .

The ideas behind the proof are as follows. First, the authors characterize the optimal solution to the problem (22)-(24), with $m_j = m$ for all $j \in N$, given an arbitrarily chosen objective function. Call this solution (\hat{x}, \hat{y}) . Next, they show that such an optimal solution can be obtained by solving an assignment problem, which we shall refer to as problem AP . The next

step is to consider the linear formulation that we assume defines the convex hull of feasible solutions, i.e., the formulation given in the theorem, and its dual. Finally, they characterize a dual feasible solution, given the same arbitrarily chosen primal objective function as above, and show that it has the same value as the optimal solution to the dual of the assignment problem AP , and hence to AP , which implies that it also has the same value as the optimal solution (\hat{x}, \hat{y}) .

Aardal et al. (1995) 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 $j \in N$. This result can be obtained by considering the following extended formulation. Let $n = |N|$.

$$\begin{aligned}
P = \{(\pi, x, y) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{Z}^n : & \sum_{j \in N} \pi_j \leq b - (m - \lambda)l \\
& \pi_j \geq x_j - (m - \lambda)y_j \quad \text{for all } j \in N \\
& \sum_{j \in N} x_j = b \\
& 0 \leq x_j \leq my_j \quad \text{for all } j \in N \\
& 0 \leq y_j \leq 1 \quad \text{for all } j \in N \\
& \pi_j \geq 0 \quad \text{for all } j \in N
\end{aligned}$$

Theorem 7 Aardal et al. (1995). *The projection of P onto the subspace $\{\pi^n : \pi = 0\}$ is equal to $\text{conv}(X_{FC}^C)$.*

Proof. By eliminating all π_j -variables we obtain inequalities (33), and the defining constraints. ■

From the separation point of view we note that the set S consists of all arcs j for which $x_j - (m - \lambda)y_j > 0$ at the current solution, which implies that the separation problem can be solved in polynomial time.

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 (Van Roy and Wolsey (1985)), 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 Applications

2.6.1 The Economic Lot-Sizing Problem

As seen in the previous subsection it is sometimes possible to describe the convex hull of feasible solutions by concise families of valid inequalities. For some problems, like the bipartite matching problem, these families contain polynomially many inequalities. In general, however, the families contain exponentially many inequalities as for instance the class of extended flow cover inequalities (33). The first complete convex hull descriptions are due to Edmonds, who characterized the matching polytope (1965) and the polymatroid polytope (1970). In

this subsection we briefly discuss such proof techniques and give a proof of the complete characterization of the convex hull of feasible solutions to the economic lot-sizing problem.

In the *primal method* we start by considering an arbitrary valid and facet defining inequality $ax \leq b$. We then determine all relations between the coefficients $(a|b)$ of this inequality that can be obtained under the assumptions that the inequality is valid and facet defining. Finally, it is shown that all feasible combinations of a and b precisely yield the facet defining inequalities that were assumed to define the convex hull. This technique usually involves a lot of technicalities, and is therefore used less frequently.

In the *dual proof technique* we take an arbitrary objective function c , and solve the dual problem of $\min\{cx : x \in P = \text{conv}(X)\}$. We then try to identify a primal solution $x \in X$ that satisfies the complementary slackness conditions given c . Since there is an objective function for each extreme point x of P , such that x is the unique optimum, this proves that each extreme point of P is in X , which in turn proves that P is the convex hull of X . Lovász (1983) uses this technique to characterize the convex hull of the polymatroid polytope. The proof by Padberg et al. (1985) that the extended flow cover inequalities (33) describe the convex hull of the single-node flow model with equal capacities is also of this type, see Section 2.5.3. An elegant variant of the dual technique has been used by Lovász (1979) on the matching polytope. He considers an arbitrary objective function c , and the face of optimal solutions F of P with respect to c . It is then shown that this face is contained in one of the faces defined by the families of valid inequalities and the defining constraints that are assumed to describe the convex hull.

We illustrate the technique of Lovász (1979) on the economic lot-sizing (ELS) problem. In ELS we are given T time periods constituting the planning horizon. In each period there is a nonnegative demand d_t which has to be satisfied with production in one of the periods $\{1, \dots, t\}$. We have a per unit production cost c_t in each period, and a set-up cost, f_t , that is incurred whenever there is positive production in period t . Let $d_{i,j}$ denote the cumulative demand of the periods $\{i, \dots, j\}$. The standard formulation of ELS involves nonnegative production variables x_t and binary set-up variables y_t .

$$\begin{aligned} \min \quad & \sum_{t=1}^T (f_t y_t + c_t x_t) \\ \text{s.t.} \quad & \sum_{t=1}^T x_t = d_{1,T} \end{aligned} \tag{34}$$

$$\sum_{\tau=1}^t x_\tau \geq d_{1,t} \quad \text{for all } t = 1, \dots, T-1 \tag{35}$$

$$x_t \leq d_{t,T} y_t \quad \text{for all } t = 1, \dots, T \tag{36}$$

$$y_t \leq 1 \quad \text{for all } t = 1, \dots, T \tag{37}$$

$$x_t, y_t \geq 0 \quad \text{for all } t = 1, \dots, T \tag{38}$$

$$y_t \text{ integral} \qquad \qquad \qquad \text{for all } t = 1, \dots, T \qquad (39)$$

Equation (34) models the restriction that there is no inventory at the beginning and the end of the planning horizon. The constraints (35) ensure that the inventory at the end of each period is nonnegative, and that all demand is met. Finally, the inequalities (36) force a set-up in each period that has positive production. We assume that $d_1 > 0$, which implies that the production in period 1 is positive, and thus the corresponding set-up variable $y_1 = 1$. We denote the vectors corresponding to the set of solutions satisfying (34)-(39) by X_{ELS} . Due to the equations (34) and $y_1 = 1$, the dimension of $\text{conv}(X_{ELS})$ is at most $2T - 2$. In fact, $\dim(\text{conv}(X_{ELS}))$ is precisely equal to $2T - 2$.

In Part I we described the following so-called (l, S) -inequalities introduced by Barany et al. (1984) for ELS.

$$\sum_{t \in \{1, \dots, l\} \setminus S} x_t + \sum_{t \in S} d_t y_t \geq d_{1,l} \quad \text{for all } l \in \{1, \dots, T\}, \text{ and all } S \subseteq \{1, \dots, l\} \quad (40)$$

The (l, S) -inequalities constitute a subfamily of the fixed charge uncapacitated network inequalities (21), which is seen in the following way, c.f. Figure 8. Let $E = \{(0, t)_{t=1}^T, (t, t+1)_{t=1}^{T-1}\}$, and let x_t denote the flow along arcs $(0, t)$, s_t de flow along arcs $(t, t+1)$, and d_t the outflow from node t . Moreover, let $X = \{1, 2, \dots, l\}$, $R = \{E(X)\}$ and $C = \{(0, i_1), \dots, (0, i_r)\}$, with $i_j \in \{1, \dots, l\}$ for $j = 1, \dots, r$, such that $H = \{i_1, \dots, i_r\} \subseteq \{1, \dots, l\}$. We also note that the inequalities (35) and (36) are special cases of the (l, S) -inequalities. Inequalities (35) can be obtained by taking $S = \emptyset$, and inequalities (36) are obtained by taking $S = \{k\}$. Barany et al. proved that $\text{conv}(X_{ELS})$ is defined by the (l, S) -inequalities together with the constraints $y_1 = 1$, (34), (37), and (38). We will prove this result by using Lovász's technique (see Lovász (1979)).

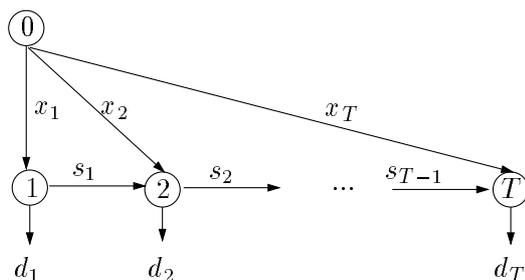


Figure 8: The economic lot-sizing problem represented as a network.

Let F be the set of optimal solutions of $\text{conv}(X_{ELS})$ with respect to the objective function $fy + cx$. To simplify the analysis we adjust the objective function in the following way. Let $c_{\min} = \min_{t \in \{1, \dots, T\}} \{c_t\}$. We first add the constant $c_{\min} d_{1,T} = c_{\min} \sum_{t=1}^T x_t$ to the objective function, and then subtract c_{\min} from every per unit production cost c_t . This ensures that the minimum per unit production cost among all periods is zero, and does not change the optimal solution since $\sum_t x_t = d_{1,T}$. Similarly, we add $f_1 y_1 = f_1$ to the objective function, and set f_1 to zero.

Case 1. Suppose that $f_t < 0$ for some $t \in \{2, \dots, T\}$.

Then all solutions in F satisfy $y_t = 1$, since a solution with $y_t = 0$ can be improved by setting $y_t = 1$. Hence, the face defined by $y_t = 1$ contains F .

From now on we may assume that $f_t \geq 0$ for all $t \in \{1, \dots, T\}$. Define $l = \max\{t : c_\tau > 0 \text{ or } f_\tau > 0 \text{ for all } \tau \leq t\}$. Hence, if $l < T$, then $f_{l+1} = c_{l+1} = 0$.

Case 2. Suppose that there is a period $t \in \{l + 2, \dots, T\}$ such that $f_t > 0$.

Then all solutions in F satisfy $y_t = 0$, since a solution with $y_t = 1$ can be improved by setting $y_t = 0$, and, if necessary, moving production in t to period $l + 1$ at a cost reduction. Hence, the face defined by $y_t = 0$ contains F . We deal with periods $t > l + 1$ for which $c_t > 0$ in a similar way.

From now on, we may assume that for all $t > l$ we have $f_t = c_t = 0$. Moreover, we may assume that $l \geq 1$, otherwise the objective function has zero coefficients only. Define $S = \{t \leq l : c_t = 0\}$.

If $l = T$ then S is not empty, since the minimum unit production cost is zero. Hence, the (l, S) -inequality based on l and S as defined above, is not implied by equation (34). Moreover, if $l = 1$, then $c_1 > 0$, which ensures that the (l, S) -inequality is not implied by $y_1 = 1$ either.

Case 3. We prove that all solutions of F satisfy the (l, S) -inequality at equality. Suppose that there is a solution (\bar{y}, \bar{x}) in F for which this is not true, i.e.,

$$\sum_{t \in \{1, \dots, l\} \setminus S} \bar{x}_t + \sum_{t \in S} d_{t,l} \bar{y}_t > d_{1,l} \quad (41)$$

Let u be minimal in S with $\bar{y}_u = 1$ if it exists, otherwise set $u = l + 1$. Production that takes place in $\{u + 1, \dots, l\}$ can be moved to u at a cost reduction, since these periods have positive unit production costs, or positive set-up costs. Therefore, all production in $\{u, \dots, l\}$ takes place in u . It follows from (41) that there must be overproduction in the periods $\{1, \dots, u - 1\} \setminus S$, which can be moved to u at a further cost reduction. Hence, we can conclude that any solution satisfying (41) is not cost optimal. This finishes the proof. ■

In Van Hoesel et al. (1994) a similar proof is given for the more general economic lot-sizing problem with additional start-up costs.

2.6.2 The Facility Location Problem

Here we shall discuss how odd-hole inequalities and flow cover inequalities can be used, and extended, when solving 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.

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

$$\text{s.t.} \quad \sum_{j \in M} v_{jk} = d_k \quad \text{for all } k \in N \quad (42)$$

$$\sum_{k \in N} v_{jk} \leq m_j y_j \quad \text{for all } j \in M \quad (43)$$

$$0 \leq v_{jk} \leq d_k y_j \quad \text{for all } j \in M, k \in N \quad (44)$$

$$y_j \in \{0, 1\} \quad \text{for all } j \in M \quad (45)$$

Inequalities (44) are redundant with respect to the integer formulation, but they do strengthen the linear programming relaxation of the facility location problem quite substantially.

The Uncapacitated Case

In the uncapacitated facility location (UFL) problem we have $m_j \geq d(N)$ 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.

$$\sum_{j \in M} x_{jk} = 1 \quad \text{for all } k \in N \quad (46)$$

$$0 \leq x_{jk} \leq y_j \quad \text{for all } j \in M, k \in N \quad (47)$$

$$y_j \in \{0, 1\} \quad \text{for all } j \in M \quad (48)$$

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 (46) 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. (1983a)). 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.

$$\sum_{j \in M} x_{jk} \leq 1 \quad \text{for all } k \in N \quad (49)$$

$$x_{jk} + y'_j \leq 1 \quad \text{for all } j \in M, k \in N \quad (50)$$

$$y'_j, x_{jk} \in \{0, 1\} \quad \text{for all } j \in M, k \in N \quad (51)$$

Let X_{UFLVP} be the set of feasible solutions to (49)–(51). 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,

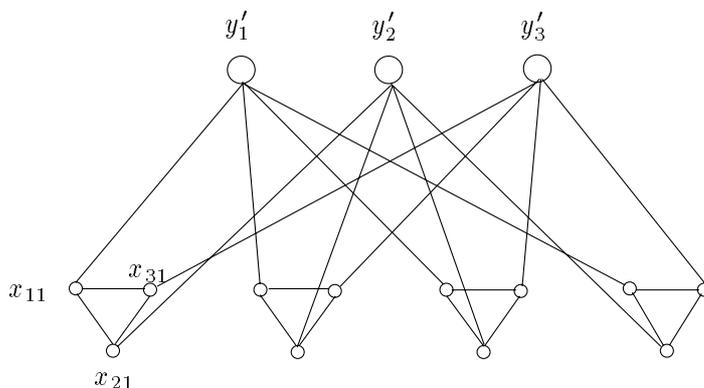


Figure 9: The intersection graph for $|M| = 3$, and $|N| = 4$.

see Figure 9. 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 (49) and (50), and all odd holes are cycles where every third vertex is a y'_j -vertex, see Figure 10. Both Cornuéjols and

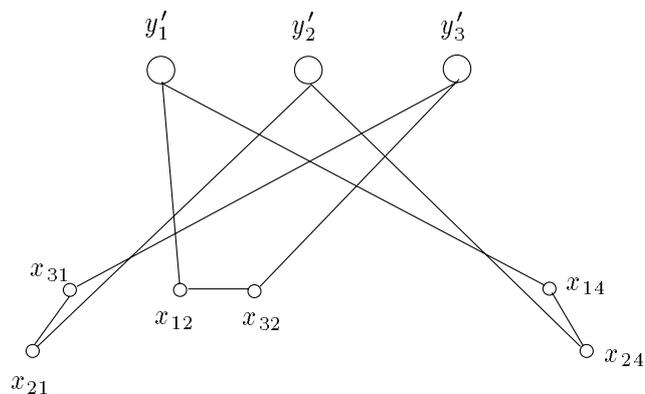


Figure 10: An odd hole in the intersection graph.

Thizy (1982) and Cho et al. (1983a,b) used the result by Chvatál given in Theorem 1 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. This lifted odd-hole inequality is precisely the inequality illustrated in Figure 3 in Section 2.2. Aardal and Van Hoesel (1995a) discuss further use of simultaneous lifting to get new facets having different coefficients.

The Capacitated Case

Here we show how flow cover inequalities can be generated based on aggregate information

from the formulation.

Let $v_j = \sum_{k \in N} v_{jk}$, and consider the following valid, but redundant constraints.

$$0 \leq v_j \leq m_j y_j \quad \text{for all } j \in M \quad (52)$$

$$\sum_{j \in M} v_j = d(N) \quad (53)$$

In Section 2.3 of Part I of this article, we described how we can combine constraints (52) and (53) with constraint (45) to obtain the knapsack polytope $X_K^{CFL} = \{y' \in \{0, 1\}^{|M|} : \sum_{j \in M} m_j y'_j \leq \sum_{j \in M} m_j - d(N)\}$, where $y'_j = 1 - y_j$. Knowing that the knapsack polytope X_K^{CFL} forms a relaxation of the capacitated facility location (CFL) problem, we can conclude that the knapsack cover inequalities generated from X_K^{CFL} are valid for CFL. If we again combine the aggregate constraints with constraint (45) we can obtain the single-node flow polytope $\{(v, y) \in \mathbb{R}^{|M|} \times \{0, 1\}^{|M|} : \sum_{j \in M} v_j = d(N), 0 \leq v_j \leq m_j y_j, j \in M\}$, see Figure 11. Hence we can use the flow cover inequalities when solving CFL. The first step in

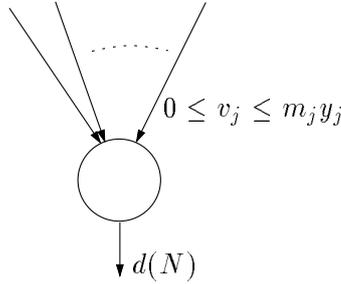


Figure 11: The single-node flow relaxation of CFL.

generalizing the flow cover inequality is made by considering inequalities based on subsets $K \subseteq N$ of clients. One way of generalizing the flow cover inequalities further is by considering a subset of clients as well as subsets of arcs yielding the family of *effective capacity inequalities* (Aardal et al. (1995)). 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 flow 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

$$\sum_{j \in J} \sum_{k \in K_j} v_{jk} \leq d(K) - \sum_{j \in J} (\bar{m}_j - \lambda)^+ (1 - y_j) \quad (54)$$

is valid for $\text{conv}(X_{CFL})$. The facet defining EC inequalities were completely characterized by Aardal et al. (1995).

Example 1 Consider the CFL structure given in Figure 12. Let $J = \{1, 2, 3\}$, $K = \{1, 2, 3, 4\}$, $K_1 = \{3, 4\}$, $K_2 = \{1\}$, and $K_3 = K$. We have $\lambda = \sum_{j \in J} \bar{m}_j - d(K) = 9$. The facet defining EC inequality based on this structure is

$$v_{13} + v_{14} + v_{21} + v_{31} + v_{32} + v_{33} + v_{34} \leq 39 - 14(1 - y_1) - (1 - y_2) - 6(1 - y_3).$$

■

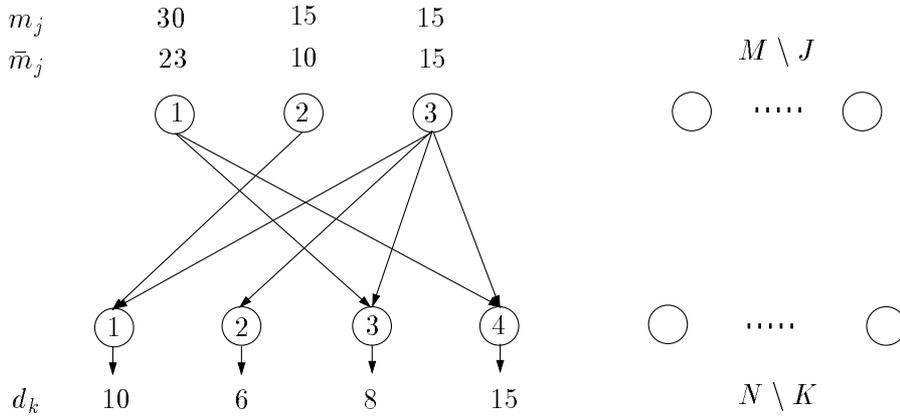


Figure 12: A CFL structure.

An even more general inequality is obtained by considering the function $f(J)$ which is the maximum feasible flow from the depots in J to the clients in K on the arcs $\{(j, k) : j \in J, k \in K_j\}$. The function $f(J)$ is submodular on M . The *submodular inequality*

$$\sum_{j \in J} \sum_{k \in K_j} v_{jk} \leq f(J) - \sum_{j \in J} (f(J) - f(J \setminus \{j\}))(1 - y_j) \quad (55)$$

is valid for CFL, see Part I. By using maximum flow arguments we can show that $f(J) - f(J \setminus \{j\}) \geq (\bar{m}_j - \lambda)^+$. Hence the submodular inequality (55) is at least as strong as the EC inequality (54) when $f(J) = d(K)$. Submodular inequalities were first considered by Wolsey (1989) and further developed for CFL by Aardal et al. (1995). 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 $f(J) - f(J \setminus \{j\}) > (\bar{m}_j - \lambda)^+$ 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, as well as computational experience from using these 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 fleet and crew scheduling: Hoffman and Padberg (1993), Hane et al. (1995). **Boolean quadratic polytope:** Padberg (1989), Lee and Leung (1993a). **Clique problems:** Grötschel and Wakabayashi (1989, 1990). **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), 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), Pulleyblank and Shepherd (1993), Balas et al. (1994b). **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 (1975), Hammer et al. (1975), Wolsey (1975), Balas and Zemel (1978), Padberg (1980), Nemhauser and Vance (1994), Weismantel (1994), Gu et al. (1995). **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 (1994), Aardal et al. (1994,1995), Aardal and Van Hoesel (1995a). **Lot sizing:** Pochet and Wolsey (1995)*. **Matching:** Edmonds (1965), Grötschel and Holland (1985). **Network and VLSI design:** Pochet and Wolsey (1992), Grötschel et al. (1992b,1993,1995), Bienstock and Günlük (1994), Bienstock et al. (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), Crowder and Padberg (1980), 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), Applegate et al. (1994), Balas et al. (1995), Goemans (1995). **Trees, forests and arborescences:** 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 (1) have been developed we can implement the separation algorithms for these inequalities in the following cutting plane algorithm, see Figure 13.

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. 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.

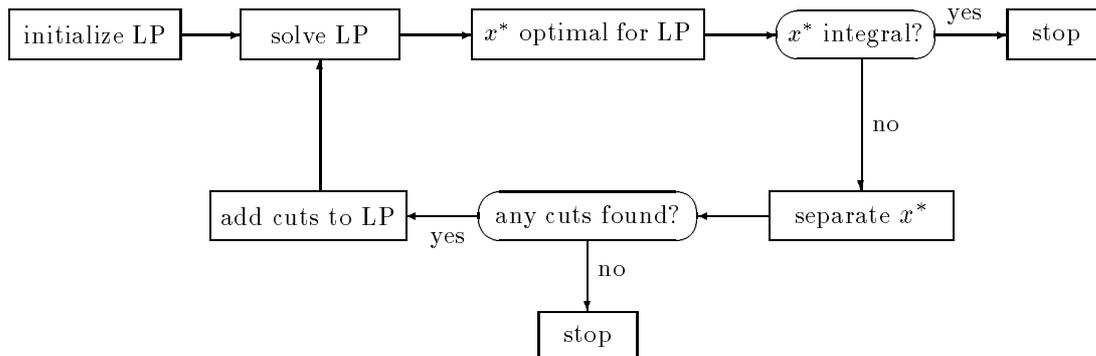


Figure 13: Basic cutting plane algorithm.

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, (Gomory (1958), see Part I) 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, the cutting plane 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. As can be concluded from the table,

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.

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 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, $(z_{IP} - z_{LP})/z_{IP}$. The percentage duality gap closed, denoted *% gap closed* is calculated as $(z_{LP}^{\text{root}} - z_{LP})/(z_{IP} - z_{LP})$, where z_{LP}^{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 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 $N = N^+ \cup N^-$.

$$\sum_{j \in N^+} a_j x_j - \sum_{j \in N^-} a_j x_j \leq b \quad (56)$$

$$l_j \leq x_j \leq u_j \quad \text{for all } j \in N \quad (57)$$

A lower bound on the left-hand side of (56) 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 (56) 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 (57) 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 14.

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 the improvement of the lower bound after preprocessing as well as the reduction in the number of branch-and-bound nodes needed to verify the optimal solution. Observe that the linear

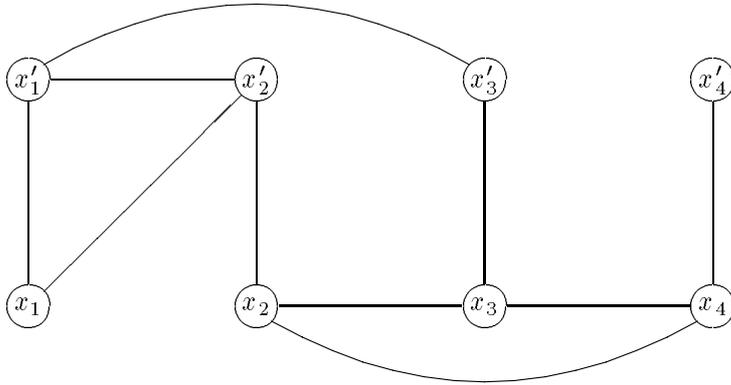


Figure 14: Auxiliary vertex packing graph.

programming bound increases substantially for all problems, and that the size of the branch-and-bound tree decreases by a fair amount 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 has to do with the relative lack of robustness of branch-and-bound, and 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 prepr.	z_{LP} with prepr.	z_{MIP}	B&B nodes without prepr.	B&B nodes with prepr.
egout	149.5	562.1	568.1	553	3
fixnet3	40,717.0	50,414.2	51,973.0	131	5
fixnet4	4,257.9	7,703.4	8,936.0	2,561	1,031
fixnet6	1,200.8	3,192.5	3,983.0	4,795	4,305
khb05250	95,919,464.0	106,750,366.0	106,940,226.0	11,483	13
gen	112,130.0	112,271.0	112,313.0	11	15
att	125.9	149.1	160.2	6,459	127
sample2	247.0	290.4	375.0	336	51
p0033	2,520.8	2,838.5	3,089.0	15	7
lseu	834.6	947.9	1,120.0	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

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 lifted knapsack cover inequalities (15). Other generic classes of valid inequalities are clique inequalities (4), obtained from the auxiliary graph of the binary variables as shown in Figure 14, odd-hole inequalities (5), network inequalities (21), and extended flow cover inequalities (32). The capacitated facility location problem provides a good insight in what these generic inequalities might offer. Table 3 shows the improvement obtained by adding lifted cover inequalities to the formulation given in Section 2.6.2. The first five instances have 33 facilities and 50 clients, whereas the last five instances have 75 facilities and 100 clients. For more details, see Aardal (1994).

problem	% gap	B&B nodes	time (s)	cover ineq.	% gap closed	B&B nodes	time (s)
50331	1.5	399	686	13	86.0	31	125
50332	1.2	691	1,560	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	1,232	25	78.3	49	248
100751	0.7	4,077	22,977	295	40.9	611	10,560
100752	0.6	15,419	74,351	648	55.4	1,243	20,055
100753	0.1	183	761	48	12.5	59	844
100754	0.3	6,687	40,604	228	9.1	537	11,076
100755	0.1	117	621	9	34.6	23	406

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

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 in the root node of the branch-and-bound tree only, since the LP-solvers were not designed to handle row management in a practical way, which made the implementation quite involved. 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 .

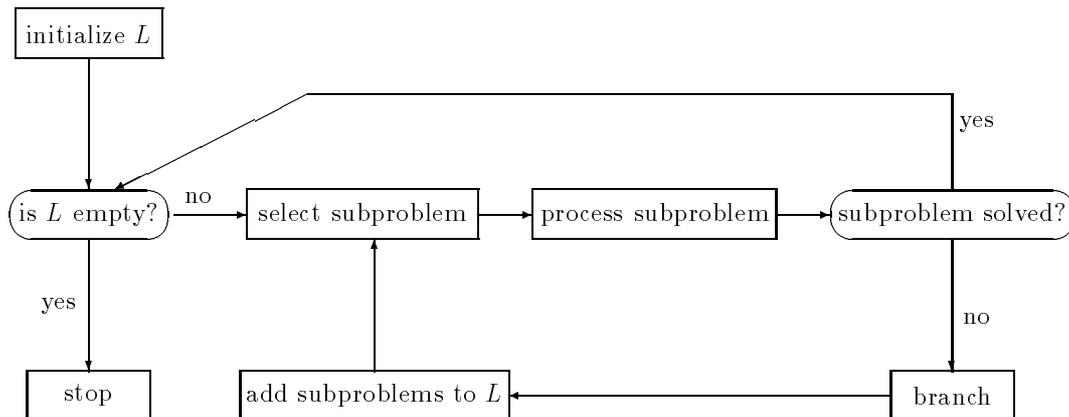


Figure 15: Branch-and-cut algorithm.

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: (i) subproblem j is infeasible, (ii) the optimal solution to the linear relaxation of subproblem j is integral, or (iii) $z_{LP}^j \geq \bar{z}$, 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 various 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 is usually weak at the same time as it 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 indicating 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 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. (1994a) report on 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. (1995). 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 16 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. At the first node we give the starting LP-value, and at the second node we give the LP-value after all cuts generated in the root node have been added. Note that after the second level of the tree all values are of the order 27,000, so we only give the digits as of the hundreds.

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. (1995).

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	1,439
80	21	369	33	80.9	97	13,352
90	15	222	13	86.7	58	3,649
100	29	181	19	93.1	108	6,631
110	35	781	5	77.1	394	84,115
120	40	903	5	72.5	251	35,194

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 100,000 LP iterations. It seems from this study that random vertex

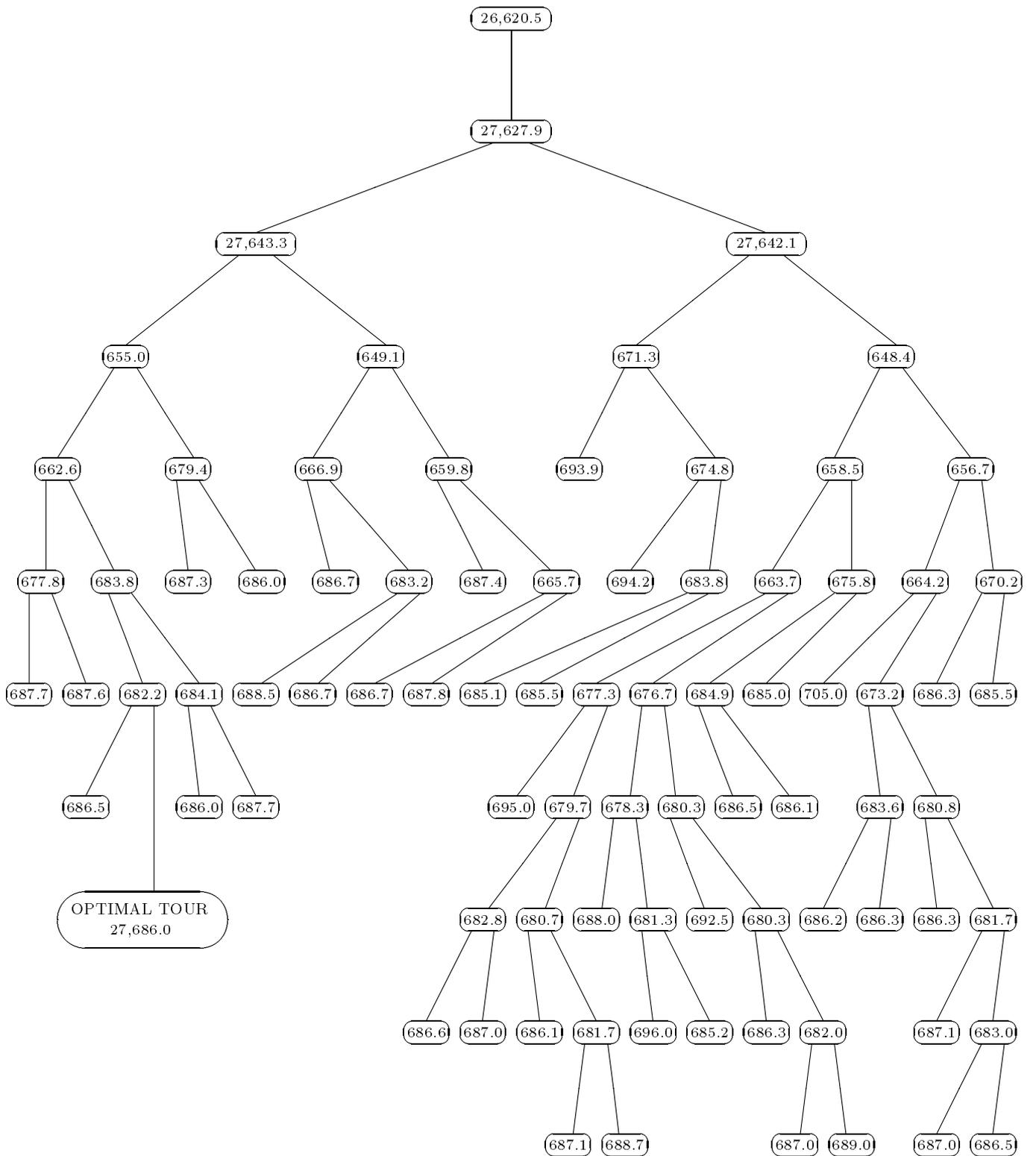


Figure 16: Branch-and-cut tree for the 532-city TSP.

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.

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 workstation.

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	1,925
340	20	22	22	6,167
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.

original		preprocessed		z_{LP}^{root}	ineq.	B&B	
variables	constr.	variables	constr.			nodes	z_{IP}
5,198	531	3,846	360	30,494	0	1	30,494
7,292	646	5,862	488	26,977	74	1	27,040
8,308	801	6,235	521	53,736	345	5	53,839
8,627	825	6,694	537	49,616	37	1	49,649
148,633	139	138,951	139	1,181,590	0	1	1,181,590
288,507	71	202,603	71	132,878	0	1	132,878
1,053,137	145	370,642	90	9,950	389	1	10,022

Table 6: Results for the airline crew scheduling problem.

variables	constraints	time 2% (s)	time 1% (s)	time opt (s)
87,482	36	225	298	2,642
8,904	823	375	375	14,441
7,195	426	868	7,443	139,337

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

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 Section 3. To make the progress visual, we 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 TSPLIB, 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-

cities	z_{LP}^{root}		B&B		application	year	reported by:
	z_{LP}	z_{IP}	nodes				
49	12,345	12,345	1		map USA	1954	Dantzig et al.
120	6,942	6,942	1		map Germany	1980	Grötschel
318	38,765	41,345	35		drilling	1980	Crowder & Padberg
532	27,628	27,686	85		map USA	1987	Padberg & Rinaldi
666	294,080	294,358	21		world map	1991	Grötschel & Holland
1,002	258,860	259,045	13		drilling	1990	Padberg & Rinaldi
2,392	378,027	378,032	3		drilling	1990	Padberg & Rinaldi
3,038	137,660	137,694	287		drilling	1992	Applegate et. al
4,461	182,528	182,566	2,092		map Germany	1994	Applegate et. al
7,397	23,253,123	23,260,728	2,247		programmable logic arrays	1994	Applegate et. al

Table 8: Computational results for the traveling salesman problem.

bound tree. In the other nodes they use only reduced-cost fixing to eliminate variables. Their computational results are shown in Table 9.

original problem			preprocessing			cutting plane		B&B	
vars.	constr.	z_{LP}	vars.	constr.	z_{LP}	ineq.	z_{LP}	nodes	z_{IP}
33	16	2,520.6	33	16	2,819.4	36	3,065.3	113	3,089.0
40	24	61,796.5	40	24	61,829.1	29	61,862.8	11	62,027.0
201	134	6,875.0	195	134	7,125.0	139	7,125.0	1,116	7,615.0
282	242	176,867.5	282	222	176,867.5	462	255,033.1	1,862	258,411.0
291	253	1,705.1	290	206	1,749.9	278	5,022.7	87	5,223.8
548	177	315.3	527	157	3,125.9	296	8,643.5	36	8,691.0
1,550	94	1,706.5	1,550	94	1,706.5	94	1,706.5	10	1,708.0
1,939	109	2,051.1	1,939	109	2,051.1	110	2,051.1	334	2,066.0
2,655	147	6,532.1	2,655	147	6,532.1	149	6,535.0	214	6,548.0
2,756	756	2,688.7	2,734	739	2,701.1	1,065	3,115.3	2,392	3,124.0

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 $\min\{cx : Ax \leq b, x \in \mathbb{Z}^n\}$ is polynomially solvable if the number of variables is fixed. A central part of his result 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 γ_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).

Tayur et al. (1995) 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 Tayur 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 and the objective function 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.

Acknowledgement

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. HIPPOLITO, C.P.M. VAN HOESEL AND B. JANSEN (1995) “A branch-and-cut algorithm for the frequency allocation problem”, Working Paper, Tilburg University, Tilburg.
- K. AARDAL AND C.P.M. VAN HOESEL (1995a) “Uncapacitated facility location: lifting, separation and computations” (in preparation).
- K. AARDAL AND C.P.M. VAN HOESEL (1995b) “Polyhedral techniques in combinatorial optimization I: theory” (to appear in The Golden Jubilee Issue of *Statistica Neerlandica*).
- K. AARDAL, M. LABBÉ, J. LEUNG AND M. QUEYRANNE (1994) “On the two-level uncapacitated facility location problem”, CentER Discussion Paper 9486, Tilburg (to appear in *ORSA Journal on Computing*).
- K. AARDAL, Y. POCHEM AND L.A. WOLSEY (1995) “Capacitated facility location: valid inequalities and facets”, *Mathematics of Operations Research* **20** 562–582.
- 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 (1975) "Facets of the knapsack polytope", *Mathematical Programming* **8** 146–164.
- E. BALAS (1979) "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. GRÖTSCHHEL AND A.R. MAHJOUB (1985) "Facets of the bipartite subgraph polytope", *Mathematics of Operations Research* **10** 340–358.
- F. BARAHONA, M. GRÖTSCHHEL, 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. MAHJOUB (1986) "On the cut polytope", *Mathematical Programming* **36** 157–173.
- F. BARAHONA AND A.R. MAHJOUB (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.)
- I. BARANY, T.J. VAN ROY AND L.A. WOLSEY (1984) “Uncapacitated lot-sizing: the convex hull of solutions”, *Mathematical Programming Study* **22** 32–43.
- D. BIENSTOCK, S. CHOPRA, O. GÜNLÜK AND C.-Y. TSAI (1995) “Minimum capacity installation for multicommodity network flows”, Working Paper, Department of IEOR, Columbia University, New York.
- D. BIENSTOCK AND O. GÜNLÜK (1994) “Capacitated network design – polyhedral structure and computation”, Working Paper, Department of IEOR, Columbia University, New York.
- 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 (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, 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) “Cliques-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 (1970) “Submodular functions, matroids and certain polyhedra” in: *Combinatorial Structures and Their Applications* (R. Guy, et al., eds) Gordon and Breach, 69–87.
- 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.
- A.M. GEOFFRION (1974) “Lagrangean Relaxation for Integer Programming”, *Mathematical Programming Study* **2** 82–114.
- 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 (1995) “Worst-case comparison of valid inequalities for the TSP”, *Mathematical Programming* **69** 335–349.
- R.E. GOMORY (1958) “Outline of an algorithm for integer solutions to linear programs”, *Bulletin of the American Mathematical Society* **64**, 275–278.
- 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, 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. Z. GU, G.L. NEMHAUSER AND M.W.P. SAVELSBERGH (1995) “Sequence independent lifting of cover inequalities”, in: *Proceedings of the Fourth Conference on Integer Programming and Combinatorial Optimization. Lecture Notes in Computer Science vol 920* (E. Balas and J. Clausen, eds.) Springer, 452–461.
- 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.
- C.A. HANE, C. BARNHART, E.L. JOHNSON, R.E. MARSTEN, G.L. NEMHAUSER AND G. SIGISMONDI (1995) “The fleet assignment problem: solving a large-scale integer program”, *Mathematical Programming* **70** 211–232.

- 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.
- C.P.M. VAN HOESEL, A.P.M. WAGELMANS AND L.A. WOLSEY (1994) “Polyhedral characterization of the economic lot-sizing problem with start-up costs”, *SIAM Journal on Discrete Mathematics* **7** 141–151.
- 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 (1995) “Practical problem solving with cutting plane algorithms in combinatorial optimization”, in: *Combinatorial Optimization. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol 20* (W. Cook, L. Lovász, and P. Seymour, eds.) AMS, 111–152.
- 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.
- H.W. LENSTRA, JR. (1983) “Integer programming with a fixed number of variables”, *Mathematics of Operations Research* **8** 538–548.
- J. LEUNG (1994) “Polyhedral structure and properties of a model for layout design”, *European Journal on Operational Research* **77** 195–207.
- J.M.Y LEUNG AND T.L. MAGNANTI (1989) “Valid inequalities and facets of the capacitated plant location problem” *Mathematical Programming* **44** 271–291.
- L. LOVÁSZ (1979) “Graph theory and integer programming”, *Annals of Discrete Mathematics* **4** 141–158.
- L. LOVÁSZ (1983) “Submodular functions and convexity”, in: *Mathematical Programming: The State of the Art* (A. Bachem, M. Grötschel, and B. Korte, eds.), Springer.
- 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 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.
- 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 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 (1995) “Algorithms and reformulations for lot sizing problems”, in: *Combinatorial Optimization. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol 20* (W. Cook, L. Lovász, and P. Seymour, eds.) 245–293.

- 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 (1986) *Theory of Linear and Integer Programming*, Wiley, Chichester.
- 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 (to appear in *Mathematics of Operations Research*).
- S.R. TAYUR, R.R. THOMAS AND N.R. NATRAJ (1995) “An algebraic geometry algorithm for scheduling in presence of setups and correlated demands”, *Mathematical Programming* **69** 369–401.
- 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.