On Combinatorial Optimization Problems Arising in Computer Systems Design

Carlos E. Ferreira

November 3, 1997

Contents

List of Figures

List of Tables

Introduction

There is no doubt that computer aided design is a major research area of our time. In this thesis we face a very interesting practical problem in computer system design, proposed by a computer manufacturing company. We will illustrate on that particular application, that practice gives rise to interesting (and difficult!!) mathematical problems and, at the same time, that mathematical methods can provide important contributions to solving real problems. In practical applications, different questions can be proposed to a problem solver. Sometimes one is interested in finding in reasonable time an optimum solution for the problem. Sometimes, time constraints are too restrictive, and one is forced to satisfy oneself with "good" solutions found very fast, or possibly very good solutions at the expense of some more time. The methods of Combinatorial Optimization provide us with tools to contribute to all these questions.

Combinatorial Optimization problems have been a challenge for mathematicians, computer scientists and engineers for many decades. In general, they can be defined as follows: *Given a finite set S, and a cost (or profit) function defined on this set, find an element of S that minimizes (maximizes) the given function.* Usually S is defined implicitly, through, for example, the properties that the elements of S must satisfy. Another interesting characteristic of such problems is that often it is not difficult to generate the elements of S, or to decide whether a given element is in S or not. Though finite, S can be very large, and a solution method like "list all possible solutions and choose the best one" leads eventually to some years of computations, even on the most modern computer systems.

For some Combinatorial Optimization problems there are efficient algorithms that solve them to optimality, that is, algorithms that in reasonable time find an element of S that maximizes (minimizes) the profit (cost) function. This is the case, for example, for the problem of finding a spanning tree with minimum weight in a given graph with a weight function on the edges, or the problem of finding a matching with maximum weight in a graph with a weight function on the edges. For other Combinatorial Optimization problems, like the Traveling Salesman Problem, Knapsack Problem, and so on, however, no efficient algorithm is known. Many people believe that these problems are, in some sense, harder than the ones mentioned above. In fact, these problems are proven to be "difficult" ($\mathcal{NP}$-hard). The question: "Does there exists a polynomial time (or *efficient*) algorithm for a $\mathcal{NP}$-hard problem?", abbreviated $\mathcal{P} = \mathcal{NP}$?, has been the major theme in theoretical computer science for the last 20 years. It turns to be an interesting phenomenon (or even a conspiracy) that most interesting problems (as the discussed in this thesis) are $\mathcal{NP}$-hard. In this thesis, we study two Combinatorial Optimization problems arising from a practical application: the Multiple Knapsack Problem and the Multicut Problem on Hypergraphs. Let us introduce them.

The Knapsack Problem is probably one of the most studied problem arising in the field of Operations Research. It can be defined as follows. Given a set N of items, vectors $f, p \in \mathbb{N}^N$, where f contains the weights of the items and p the profits, and a natural $F \in \mathbb{N}$ the knapsack capacity, one wants to find a subset $S \subseteq N$ such that $\sum_{i \in S} f_i \leq F$ and $\sum_{i \in S} p_i$ is maximum. The fascination of the problem arises in the fact that, though the problem is known to be *difficult* ($\mathcal{NP}$-hard), there are very good approximation algorithms for it, that run in polynomial time and yield solutions that are arbitrarily close to the optimum value. In this case, although one cannot give an efficient algorithm to solve the problem, it is possible to provide "good" solutions quickly. In some sense the problem seems to be "easier" than other difficult problems, such as the Traveling Salesman Problem. In this thesis we study the Multiple Knapsack Problem, a generalization of the Knapsack Problem in which more than one knapsack are available. This problem has also been proven difficult.

In general, one can define a Multicut Problem in the following way. Given a graph (or a hypergraph) $G = (V, E)$, the Multicut Problem consists in finding a partition of V into k subsets that maximizes (or minimizes) some function. Two examples of problems of this type are the so called Mincut and Maxcut Problems. In the first case, a graph $G = (V, E)$ is given with a nonnegative function w defined on E and it is desired to find a partition of V into two parts, such that the sum over the weights of the edges with one endnode in each of these sets is minimum. In the Maxcut Problem, the function w has arbitrary values, and one looks for a partition of V into two parts, such that the sum over the weights of the edges with one endnode in each part is maximum. These two examples are especially interesting, because the Maxcut Problem is $\mathcal{NP}$-hard, while there is a polynomial time algorithm for the Mincut Problem. The Multicut Problem we are interested in is the following. Given a hypergraph $G = (V, E)$, a set M, functions $f : V \longrightarrow \mathbb{N}$, $g : E \longrightarrow \mathbb{N}$, cost functions $c : V \times M \longrightarrow \mathbb{N}$ and $d : E \longrightarrow \mathbb{N}$, and vectors $F, S \in \mathbb{N}^M$. The problem consists in finding a partition $\mathcal{V} = (V_1, \ldots, V_j)$ of V such that $j \leq |M|$, $\sum_{v \in V_k} f(v) \leq F_k$ for all $k \in M$, $\sum_{e \in \delta(V_k)} g(e) \leq S_k$ for all $k \in M$, where $\delta(V_k)$ is the set of edges with exactly one endnode in V_k, and this partition has minimum cost, that is, minimizes $\sum_{k \in M} \sum_{e \in \delta(V_k)} d(e) + \sum_{k \in M} \sum_{v \in V_k} c(v, k)$.

As mentioned above, our research was motivated by some practical application in computer systems design. This topic arises in the phase of the global design where certain components have been already defined, the networks connecting components are also defined and we want to decide how to group the components and how to integrate them on a given number of multichip modules or printed circuit boards. We present now a rough description of this application.

Suppose a computer system with several components and some modules (multichip modules or printed circuit boards) to which these components must be assigned. There are technological constraints that make certain assignments invalid. One example of such a technological constraint is that the components must be placed on a module without overlapping. Some components are linked by several nets. If one separates the components of some net into two modules, a wire must link these modules. The number of wires leaving a module is limited by technological reasons. This problem can be modelled as a 0/1 integer program. The mathematical problem that arises from this model turns out to be "hopeless". One possible idea to overcome this difficulty is to consider simplifications of the model, trying to obtain an approximate solution as close as possible to the original problem. The Multiple Knapsack Problem arises as a first "difficult" subproblem of this model. Let N be the set of components, $f = (f_i)$ the area of these components, M the set of modules, $F = (F_k)$ their area, and $c = (c_{ik})$ a cost function associated to assigning component i to module k. The solution of the Multiple Knapsack Problem gives a first bound to the solution of the original problem, where the net constraints are, in some way, hidden in the objective function. Another simplification of the model closer to the original application is described in the second part of this thesis, where we address a Multicut Problem on Hypergraphs. In the case of our application, besides the above sets and vectors we are given a set Z of nets, a vector $g = (g_t)$ with the multiplicity of the nets, and the vector $S = (S_k)$ which expresses the maximum number of external wires allowed to leave each module. The solution of the multicut problem gives us a closer bound to the value of the optimum solution of the real problem.

Our main objective in this thesis is to study both problems derived from the application in computer systems design, and present solution approaches to them. Solution in the meaning of the questions we mentioned in the very beginning of this introduction: our aim is to provide optimal solutions, or, at least, provable "good" solutions for large scale instances in reasonable computer times. The approach suggested in this thesis is a Branch and Cut Algorithm. Using this approach, very good results in solving large scale Combinatorial Optimization problems (e.g. the Traveling Salesman Problem) have been obtained in the last years.

This thesis is structured as follows. In Chapter 1 we state our terminology and present some notation and basic results in graph theory, linear algebra and polyhedral theory. In Chapter 2 we present a description of the application and the model we propose using 0/1 variables. Moreover, in this chapter the problems we study in the rest of the thesis are presented: the Multiple Knapsack Problem and the Multicut Problem on Hypergraphs.

Chapters 3 to 6 address the Multiple Knapsack Problem. Chapters 3 and 4 are mainly dedicated to a literature review of the problem and related ones. Chapter 3 addresses a very important special case of the Multiple Knapsack Problem, namely, the Single Knapsack Problem. In this chapter we survey some of the results of the literature on this problem, discussing, for example, polynomial and fully polynomial approximation schemes. Further, some important definitions and results on facet defining inequalities for the Single Knapsack Polytope are presented. In Chapter 4 we present a literature survey on the Multiple Knapsack Problem. Moreover, we present some other related problems of the literature: the Generalized Assignment Problem and the General Zero-One Programming Problem. We address especially polyhedral investigations to these problems.

We finish our discussion about the Multiple Knapsack Problem studying a polytope associated with it, and by presenting a Branch and Cut Method for its solution. This is the subject of Chapters 5 and 6. More precisely, in Chapter 5 we associate the elements of the solution set of the problem with the vertices of a polytope and investigate its properties. We derive, besides some basic results (we show, for example, that every facet defining inequality of the polytope associated with a single knapsack constraint constitutes a facet of the polytope associated with the Multiple Knapsack Problem), new classes of facet defining joint inequalities of this polytope. In Chapter 6 we present an implementation of a Branch and Cut Algorithm for the Multiple Knapsack Problem. In this chapter we use the knowledge of the polyhedral characteristics of the polytope described in Chapter 5. Further, implementation details and computational results of the algorithm applied to small instances and practical problems are also presented.

Chapters 7 to 9 are dedicated to Multicut Problems on graphs and hypergraphs. In Chapter 7 we state the problem that we study and present some related problems and a brief literature survey on the subject. In Chapter 8 polyhedral investigations of the polytope associated with the Multicut Problem on Hypergraphs are presented. We show, for example, which results from the subproblems can be inherited, and present new classes of facet defining inequalities combining node and edge variables. These results are used as a basis for a Branch and Cut Algorithm, which is described in Chapter 9.

This thesis has been made under the excellent supervision of Prof. Dr. Martin Grötschel. His critics and suggestions were indispensable. Most of all, his encouragement and enthusiasm motivated me a lot during these years. I am very grateful to his support. Most of the work presented in this thesis has been made jointly with Dr. A. Martin and Dr. R. Weismantel. My work in Germany began in the University of Augsburg and has been concluded in the Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB). My best thanks to both institutions that provided the facilities during my research. In particular, thanks are due to Prof. Dr. Karl-Heinz Borgwardt (Augsburg), Prof. Dr. Robert E. Bixby (Rice University and – in 1992 – TU Berlin) and Prof. Dr. Rolf Möhring (TU Berlin).

I am also grateful to the financial support given by the Deutscher Akademischer Austauschdienst (DAAD), whose assistance (especially personified in *Frau* Helga Wahre) was very important during my study. Further, special thanks also to the Computer Science Department of the Institute of Mathematics and Statistics of the University of São Paulo, whose program of sending students abroad to complete their PhD. studies made it possible to me being in Germany for four years with financial support.

There are several people to them I would like to say my *muito obrigado* and my *Dankeschön*. My *Dankeschön* goes to Ralf Borndörfer, Dr. Alexander Martin and Dr. Robert Weismantel with whom I had almost daily interesting discussions about the themes here discussed. Their careful revisions of the first versions of this thesis have avoided several errors. Of course the remaining errors in this final version are only my responsibility. Moreover, and, for me, even more important, their encouragement and friendship (including Alex's and Robert's families) were really decisive during these years. I am also very grateful to the other colleagues in the combinatorial optimization department of ZIB, that showed me how professional and cooperative a team can work. *Dankeschön!!*

Muito obrigado to all my colleagues in São Paulo. Especially, my best thanks to Yoshiko Wakabayashi, who introduced me in Combinatorial Optimization. *Muito obrigado* also to the moral support given by my (in these years) "e-mail" friends Dilma, Cris, Coelho and Claudio. Last, I would like to thank my family for everything they have done, encouraging me all the time and being patient during these years. *Muito obrigado!!*

Carlos Eduardo Ferreira Berlin, October 1993.

Chapter 1

The Mathematical Model of the Application

This thesis was motivated by an application concerning some aspects that arise in the design of main frame computers. We do not present an exact mathematical model involving all possible aspects in the design of a main frame computer. This would be a very difficult job. The topic considered here comes up in the phase of the global design. We consider the problem of distributing the components of a computer system (memory, IO devices, CPU's, etc.) onto the boards (or modules) that constitute this system. There are several technological constraints that make some possible distributions invalid. Moreover, also due to technological reasons, some distributions may be realizable in a cheaper manner. The task we would like to model is: *Find a valid assignment of the computer components to the modules that causes minimum realization costs.* As we are going to see is this section, the mathematical problem arising from this practical application turns out to be very complicated. Exact approaches to solve these problems would require hundreds of thousands of variables. A possible approach to solve these real problems approximately is to consider simplifications of the exact model. In such simplifications some constraints are relaxed or substituted in the objective function using penalties, or some variables are left out. We suggest two simplifications of the original model in this chapter giving rise to treatable (but still difficult) mathematical problems: the Multiple Knapsack Problem and the Multicut Problem on Hypergraphs. In the rest of this thesis we suggest an approach to solve these problems. The results in this chapter has been obtained in a joint work with Siemens Nixdorf and are published in [FGKKMW93].

In this chapter we go into details, presenting more precisely the application we are interested in. This is done in Section 2.1. Section 2.2 is devoted to the presentation of a model of the problem using 0/1 variables. The last section addresses simplifications of the model presented in 2.2. A first simplification motivates the study of the Multiple Knapsack Problem (object of investigation of the Chapters 3 to 6). Another simplification of the model proposed in 2.2 is also presented, that gives rise to the Multicut Problem on Hypergraphs. A study of this problem is presented in Chapters 7 to 9.

1.1 Description of the problem

In the application we want to solve, we are given a list of electronic components. For our problem, it is usual to view these components as two-dimensional rectangles, but there may also be a few differently shaped units or "dummy components" to reserve space on certain modules. The most important property of the electronic circuits – for our purposes – is the area that these components cover. We abstract from the technical details by saying that a set N of *items* (the components) is given where each item $i \in N$ has a *weight* $f_i \in \mathbb{N}$.

The electronic components have to be integrated on printed circuit boards, multi chip modules or other devices. There may be various types of printed circuit boards. Each of these devices is defined by several technical properties that we do not intend to describe here. We call these devices *modules* from now on and denote the set of modules that are available by M. Three properties of modules are important for us. Every module $k \in M$ has a *capacity* F_k, representing its "area" or the weight it can hold, a *cut capacity* S_k, describing the number of wires that can be connected to this module, and a (generic) *cost* K_k, representing the corresponding fabrication cost of that particular module.

The electronic components have certain contact points, called pins, from which wires can extend to pins of other components. In the logical design phase it is determined which pins of which components have to be connected by a wire to ensure certain functional properties. Usually, a collection of pins that have to be connected is called a net. We simplify the situation by essentially disregarding the pins (their number will only enter the objective function, see below). We define a *net* to be a subset of the set of items and we ignore which of the pins of the components are to be connected. The list of nets is denoted by $\mathcal{Z} := \{T_1, \ldots, T_z\}$. We set $Z := \{1, \ldots, z\}$ and, for simplicity, we will often speak of net $t \in Z$ instead of net $T_t \in \mathcal{Z}$.

For some nets t, it is necessary to partition the set T_t into two subsets S_t and R_t (with $S_t \cup R_t = T_t,\ S_t \cap R_t = \emptyset$). The items of S_t are called *drivers*. They transmit information over the wire to the items in R_t. R_t consists of so-called "receivers" and "termination resistors".

Our task is to assign the items (electronic components) N to the modules (printed circuit boards, ...) M in such a way that a certain objective function is minimized and a number of technical side constraints is satisfied. Our way of approaching the practical problem reduces the technical side constraints to three essential requirements. Let us describe these now.

Suppose an assignment of items to modules $a : N \to M$ is given. For each module $k \in M$, let $B(k)$ denote the set of items that are assigned to module k.

For an assignment to be *feasible*, the following conditions must hold:

- *Knapsack constraints:*
  For each module k, the weight of the items that are assigned to that module must not exceed the capacity F_k of the module, i.e., $\sum_{i \in B(k)} f_i \leq F_k$.

- *Cut constraints:*
  The number of nets t with $T_t \cap B(k) \neq \emptyset$ and $T_t \not\subseteq B(k)$ must not exceed the cut capacity S_k of module k for all $k \in M$.

- *Net constraints:*
  There are two special classes of nets, the *one-module nets* and the *s-r nets*. These special nets must satisfy respectively the following rules.

(R1) one-module nets: a net of this type must have all its components assigned to the same module.

(R2) s-r nets: either all items of a s-r net t must be assigned to the same module, or all items of S_t must be assigned to the same module, say k, but none of the items of R_t may be assigned to k.

Clearly, the knapsack constraints are meant to ensure that the items assigned to some module fit onto that module. Note, however, that the 2-dimensional problem of packing components onto devices is approximated in our model by a 1-dimensional problem. It may, in fact, be possible that the components of a feasible solution in the latter sense do not fit onto the board when the problem is considered in its (real) 2-dimensional version. The reason for considering the 1-dimensional simplification is that, at the time when the present model is solved (repeatedly), the exact design of the components is usually not completed. There exist good estimates for the component areas and there is some flexibility with respect to giving the components their final shape. Reasonably sized "dummy" items produce empty spaces on the modules that help to finally place the components. Thus, the 1-dimensional simplification is – at this stage of the process – a reasonably good model of what the designer has in mind.

For each module k, there exists a so-called "connector" which contains a certain number of pins. These pins can be employed to connect items placed on k to items on other modules. The number of pins of the connector that can be used for inter module wiring from k is the cut capacity S_k. Since every net that has an item on k and at least one other not on k uses exactly one of these pins we obtain the cut constraint.

The reason for introducing the (significantly complicating) net constraints is of very technical nature. We omit a detailed discussion of this aspect.

Let us now explain the objective function we came up with. The objective function is of the form

$$(*) \qquad \min \sum_{k \in M} K_k \cdot I_k(a) + \lambda \cdot C(a).$$

We first describe the second term of the objective function, the so-called *external cost* of the assignment a. The external cost $C(a)$ depends only on the number of nets whose items are assigned to different modules. In order to explain the function $C(a)$ exactly we must describe some technical issues of the problem in more detail. Of course, the design of a main frame computer is not finished after assigning the items to the modules. Thereafter, the items must be physically placed onto each module and the nets must be physically routed, i.e., connected via wires. Routing of a net t whose items are assigned to different modules is done as follows (see Figure 1.1). For each module k with $B(k) \cap T_t \neq \emptyset$ and $T_t \not\subseteq B(k)$ an additional pin at the border of the module, a so-called *external pin*, is introduced (see the black rectangles in Figure 1.1). A routing for net t is obtained by connecting the items of $B(k) \cap T_t$ within each module with the corresponding external pin (see dotted lines in Figure 1.1) and by connecting the external pins via a so-called *external wire* (see dashed lines in Figure 1.1).

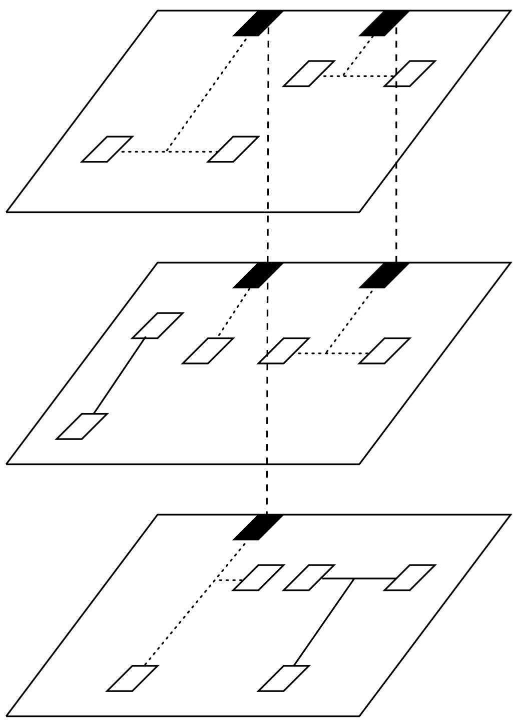


Figure 1.1: Routing nets with external pins.

The cost of an external wire is approximated in the objective function by the number of external pins. If $p(t)$ denotes the number of external pins for some net

t (where $p(t) := 0$, if all items of net t are assigned to the same module), we define $C(a) := \sum_{t \in Z} p(t)$. The factor λ in the second term of the objective function is a penalty parameter that weighs the external cost in relation to the first term of the objective function, the so-called *internal cost*.

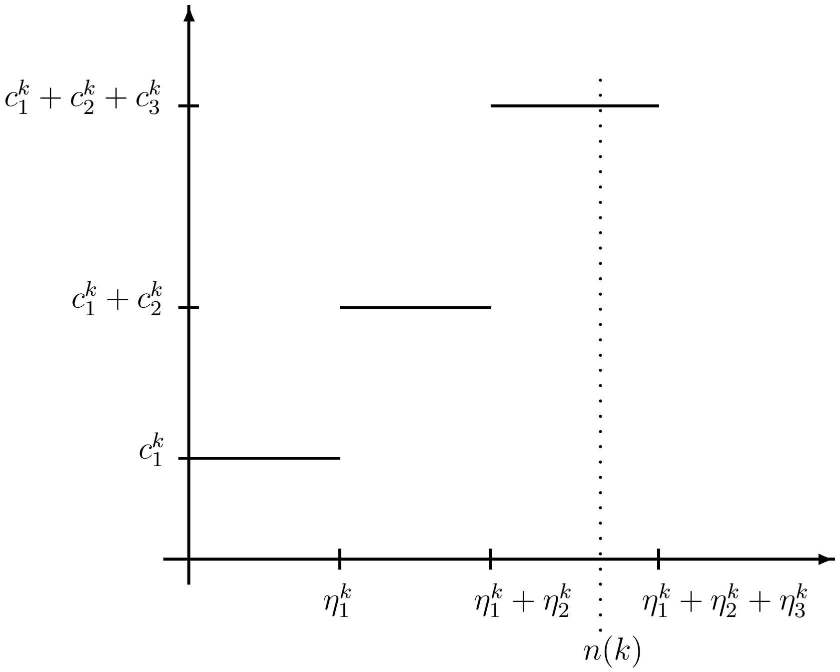


Figure 1.2: The internal cost function I_k of some module k

The internal costs consist of the sum of the internal costs for each module. Consider some module $k \in M$. The routing of the nets inside module k is performed on so-called *layers*. On each layer only a certain number of nets can be connected. The number of layers necessary to do the complete routing strongly depends on the technology used to produce the printed circuit board or the multi chip module. In our case it is estimated as follows. For each net $t \in Z$, we set

$$w_k(t) := \begin{cases} |T_t| - 1, & \text{if } T_t \subseteq B(k), \\ |T_t \cap B(k)|, & \text{if } T_t \cap B(k) \neq \emptyset, T_t \not\subseteq B(k), \\ 0, & \text{else,} \end{cases}$$

and we define an auxiliary number $n(k)$ by $n(k) := \sum_{t \in Z} w_k(t)$. The number of layers necessary to assign all items of B_k grows with the number $n(k)$. The production cost of one module mainly depends on the number of layers that are necessary. Each installation of a layer costs a certain amount, but the total cost of a module grows superlinearly with the number of layers, since production faults in a later stage usually destroy successful work on the initial layers. This cost function can be expressed by a staircase function (denoted in the objective function by $I_k(a)$). Let us now explain this function in more detail (see also

Figure 1.2). For each layer l, we denote by c_l^k the installation cost for layer l and by η_l^k the "capacity" of layer l. Let l^* denote the smallest integer such that $\sum_{l=1}^{l^*} \eta_l^k \geq n(k)$, i.e., l^* is the smallest number such that the system of nets that connects the components and external pins on module k can (probably) be routed when module k is designed as a device with l^* layers. Then, $I_k(a)$ is set to $\sum_{l=1}^{l^*} c_l^k$.

Summing up our previous discussions we can formulate the module design problem for a main frame computer as follows:

(Module Design Problem)

Given data:

- A set N of items. Each item $i \in N$ has some weight f_i.
- A set M of modules. With each module $k \in M$ a capacity F_k, a cut capacity S_k and a cost factor K_k are associated.
- A list of nets $\mathcal{Z} = \{T_1, \ldots, T_z\}$ with $T_t \subseteq N$ for $t \in Z = \{1, \ldots, z\}$.
- A list of one-module nets.
- a list of s-r nets, and for each net t, sets R_t and S_t containing the sender and receiver items.

Problem:

Find an assignment of the items to the modules such that the knapsack constraints, the cut constraints and the net constraints are satisfied and such that the objective function $(*)$ is minimized.

Our problem analysis has revealed that some of the technical requirements can be treated easily. We do this in a preprocessing stage and describe here two such cases concerning the net constraints.

Preprocessing

Suppose t is a one module net (described in (R1)). In this case we simply define a new item i' with weight $f_{i'} := \sum_{i \in T_t} f_i$. The new set of items is $N' := (N \backslash T_t) \cup \{i'\}$. Thus, the net constraint (R1) is automatically satisfied if we assign the items of N' to the modules of M by taking all other constraints into account.

In the same manner we can simplify net constraint (R2). Suppose t is a s-r net. Let S_t be the set of senders and R_t the set of receivers or termination resistors, respectively. Again, we introduce a new item i' with weight $f_{i'} := \sum_{i \in S_t} f_i$ and set $N' := (N \setminus S_t) \cup \{i'\}$. After doing this iteratively we can assume that each s-r net t has exactly one sender, i.e., $|S_t| = 1$.

These changes, of course, imply an obvious redefinition of the nets and an adjustment of the objective function.

1.2 A formulation using 0/1 variables

In this subsection we provide a 0/1 programming formulation of the module design problem. For that purpose we introduce the following four sets of 0/1 variables.

For all items $i \in N$ and all modules $k \in M$, we introduce a variable x_{ik} with the interpretation

$$x_{ik} := \begin{cases} 1, & \text{if item } i \text{ is assigned to module } k, \\ 0, & \text{else.} \end{cases}$$

For every net $t \in Z$ and every module $k \in M$, we introduce three variables y_{tk}, y^1_{tk} and y^2_{tk} with the following interpretation.

$$y_{tk} := \begin{cases} 1, & \text{if some items of } T_t \text{ are assigned to module } k \\ & \text{but not all of them,} \\ 0, & \text{else.} \end{cases}$$

$$y^1_{tk} := \begin{cases} 1, & \text{if at least one item of } T_t \text{ is not assigned to module } k, \\ 0, & \text{else.} \end{cases}$$

$$y^2_{tk} := \begin{cases} 1, & \text{if at least one item of } T_t \text{ is assigned to module } k, \\ 0, & \text{else.} \end{cases}$$

Note that there are dependencies between the variables y_{tk}, y^1_{tk} and y^2_{tk}. However, for ease of exposition of the constraints, it is convenient to introduce all three sets of variables. For every module k, there is an upper bound l_k, say, of layers available. This integer depends on the production technology used. In order to model the staircase function I_k, we introduce a variable v^k_l for each layer $l = 1, \ldots, l_k$ and each module $k \in M$.

These variables have the following meaning:

$$v^k_l := \begin{cases} 1, & \text{if } n(k) \geq \sum_{r=1}^{l} \eta^k_r, \\ 0, & \text{else.} \end{cases}$$

With these four sets of variables we are able to model the side constraints of the module design problem, i.e., the knapsack constraints, the cut constraints and the net constraints.

$$\sum_{k \in M} x_{ik} = 1, \qquad \text{for all } i \in N, \tag{2.1}$$

i.e., each item is assigned to exactly one module.

$$\sum_{i \in N} f_i x_{ik} \leq F_k, \qquad \text{for all } k \in M, \tag{2.2}$$

i.e., the knapsack constraints must be satisfied.

$$\sum_{t\in Z} y_{tk} \le S_k, \qquad \text{for all } k \in M, \tag{2.3}$$

i.e., the cut constraints must be met.

$$x_{sk} + x_{ik} \le 2 - y^1_{tk}, \quad \text{for all s-r nets } t \in Z \text{ and for all } i \in R_t \text{ (where } \{s\} = S_t). \tag{2.4}$$

These inequalities are derived from the following reasoning. If $y^1_{tk} = 0$, that is all items of net t are assigned to module k, inequality (2.4) is obviously valid. On the other hand, if $y^1_{tk} = 1$, which means that at least one item of net t is not assigned to module k , the sender s and a receiver (resp. termination resistor) $i \in R_t$ cannot both be assigned to module k. Thus, (2.4) ensures that the net constraints (R2) are satisfied. Note that net constraints (R1) are handled in the preprocessing phase.

The following constraints (2.5) to (2.9) are necessary to logically connect the involved variables.

$$\sum_{i\in T_t} x_{ik} + |T_t| y^1_{tk} \ge |T_t|, \qquad \text{for all } k \in M,\ t \in Z, \tag{2.5}$$

i.e., if $\sum_{i\in T_t} x_{ik} < |T_t|$, which means that not all items of net T_t are assigned to module k, y^1_{tk} must be one.

$$\sum_{i\in T_t} x_{ik} + y^1_{tk} \le |T_t|, \qquad \text{for all } k \in M,\ t \in Z, \tag{2.6}$$

i.e., if $\sum_{i\in T_t} x_{ik} = |T_t|$, which means that all items of net T_t are assigned to module k, y^1_{tk} must be zero.

$$\sum_{i\in T_t} x_{ik} - |T_t| y^2_{tk} \le 0, \qquad \text{for all } k \in M,\ t \in Z, \tag{2.7}$$

i.e., if $\sum_{i\in T_t} x_{ik} \ge 1$, which says that at least one item of net T_t is assigned to module k, y^2_{tk} must be one.

$$\sum_{i\in T_t} x_{ik} - y^2_{tk} \ge 0, \qquad \text{for all } k \in M,\ t \in Z, \tag{2.8}$$

i.e., if $\sum_{i\in T_t} x_{ik} = 0$, that is, no item of net T_t is assigned to module k, y^2_{tk} must be zero.

$$y^1_{tk} + y^2_{tk} = 1 + y_{tk}, \qquad \text{for all } k \in M,\ t \in Z, \tag{2.9}$$

i.e., if $y_{tk} = 0$ either y^1_{tk} or y^2_{tk} must be zero. On the other hand, if $y_{tk} = 1$ both y^1_{tk} and y^2_{tk} must be one. From equation (2.9) we conclude that, for each pair tk, one of the variables y^1_{tk}, y^2_{tk} or y_{tk} is redundant. However, we have introduced all three types of variables here to simplify the explanation of the model.

In order to obtain a correct formulation of the objective function, the following constraints are introduced.

$$\begin{aligned}(2.10)\quad & \sum_{t\in Z}\Big(\sum_{i\in T_t} x_{ik} + y^1_{tk} - 1\Big) - \sum_{l=1}^{l_k} \eta^k_l v^k_l \le 0, \\ & v^k_1 \ge v^k_2 \ge \ldots \ge v^k_{l_k}, \quad \text{for all } k \in M.\end{aligned}$$

It is easy to see that for a given $k \in M$, $w_k(t) = \sum_{i\in T_t} x_{ik} + y^1_{tk} - 1$. This implies that $n(k) = \sum_{t\in Z}(\sum_{i\in T_t} x_{ik} + y^1_{tk} - 1)$. Hence, this set of inequalities models that $v^k_l = 1$ for all $l = 1, \ldots, l^*$, where l^* is the smallest integer such that $\sum_{l=1}^{l^*} \eta^k_l \ge n(k)$. All other variables v^k_l, $l \in \{l^*+1, \ldots, l_k\}$, are equal to zero, since the (positive) internal costs are minimized in the objective function.

Finally, we require that every variable is either zero or one.

$$(2.11)\quad \begin{aligned} x_{ik} &\in \{0,1\}, && \text{for all } i \in N,\ k \in M, \\ y_{tk}, y^1_{tk}, y^2_{tk} &\in \{0,1\}, && \text{for all } t \in Z,\ k \in M, \\ v^k_l &\in \{0,1\}, && \text{for all } l = 1, \ldots, l_k,\ k \in M. \end{aligned}$$

These eleven sets of inequalities model all technical side constraints considered in our version of the real task. The objective function expressed in terms of the 0/1 variables is of the form

$$(2.12)\qquad \min \sum_{k\in M} K_k \sum_{l=1}^{l_k} c^k_l v^k_l \; + \; \lambda \sum_{t\in Z}\sum_{k\in M} y_{tk}.$$

This objective function corresponds to the one described in (∗). This follows from the fact that constraints (2.10) ensure that, for every $k \in M$, the expression $\sum_{l=1}^{l_k} c^k_l v^k_l$ models the staircase function I_k. Moreover, for each net $t \in Z$, the term $\sum_{k\in M} y_{tk}$ corresponds to the number of external pins $p(t)$.

1.3 Simplifications of the exact model

As pointed out in the introduction of this chapter, to solve the mathematical problem arising from the formulation given in the last section exactly seems to be (at the current standards of integer programming) hopeless. One idea to attack the problem is to consider simplifications of it. In this section we present two simplifications to this problem, giving rise to the problems studied in this thesis: the Multiple Knapsack Problem and the Multicut Problem on Hypergraphs.

1.3.1 First simplification: the Multiple Knapsack Problem

In this simplification we leave out completely the variables associated with the nets, considering only the packing aspects of the problem. Our objective is to develop a model using only the variables (x_{ik}) defined in the last section. To this end we approximate the staircase function I_k by an average value D_k, defined as follows:

$$D_k := \frac{K_k}{l_k}\left(\sum_{r=1}^{l_k} \frac{c_r^k}{\eta_r^k}\right).$$

We approximate the value $n(k)$ by $\sum_{t\in Z}\sum_{i\in T_t} x_{ik}$. In the last section we see that the exact value of $n(k)$ is given by $\sum_{t\in Z}(\sum_{i\in T_t} x_{ik} + y_{tk}^1 - 1)$. It means that our approximation does not coincide with the exact value when some nets are totally assigned to a single module.

The approximate value of the objective function is, then, given by

$$\min \sum_{k\in M} D_k \sum_{t\in Z}\sum_{i\in T_t} x_{ik}.$$

Observe that this is a rough approximation of the original objective function, since the external costs involved in an assignment are not taken in account. The first simplification can be written:

$$\begin{array}{ll} \min & \sum_{k\in M} D_k \sum_{t\in Z}\sum_{i\in T_t} x_{ik} \\ s.t. & (2.1), (2.2), \\ & x_{ik} \in \{0,1\}, \text{ for all } i\in N, k\in M. \end{array}$$

We call this special case of the Generalized Assignment Problem (see Section 4.2) *Knapsack Assignment Problem* KAP. The equations (2.1) result, in the approach we intend to utilize to solve instances of this problem, in some technical difficulties. In order to avoid such technicalities, we introduce the following modification in the problem described above. For each equation in (2.1) we introduce a slack variable z_i, $i \in N$, and a suitably chosen penalty parameter Q_i associated with z_i. The idea is to choose Q_i such that the variables z_i must be equal to zero in all optimal solutions of the modified problem. Then, KAP is clearly equivalent to

$$\begin{array}{ll} \min & \sum_{k\in M} D_k \sum_{t\in Z}\sum_{i\in T_t} x_{ik} + \sum_{i\in N} Q_i z_i \\ s.t. & \sum_{k\in M} x_{ik} + z_i = 1, \text{ for all } i\in N, \\ & (2.2), \\ & x_{ik}\in\{0,1\}, \text{ for all } i\in N, k\in M, \\ & z_i \in \{0,1\}, \text{ for all } i \in N. \end{array}$$

Now, we can eliminate the slack variables of the problem, substituting $z_i = 1 - \sum_{k\in M} x_{ik}$, for all $i \in N$ in the objective function, obtaining,

$$\begin{array}{rl} & \min \sum_{k\in M} D_k \sum_{t\in Z}\sum_{i\in T_t} x_{ik} + \sum_{i\in N} Q_i(1 - \sum_{k\in M} x_{ik}) \\ = \sum_{i\in N} Q_i \quad + & \min \sum_{k\in M}(D_k \sum_{t\in Z}\sum_{i\in T_t} x_{ik} - Q_i \sum_{i\in N} x_{ik}) \end{array}$$

The latter problem, namely,

$$\begin{array}{lll} \min & \sum_{i\in N}\sum_{k\in M} c_{ik}x_{ik} & \\ s.t. & \sum_{k\in M} x_{ik} \le 1 & \text{for all } i\in N, \\ & \sum_{i\in N} f_i x_{ik} \le F_k & \text{for all } k \in M, \\ & x_{ik}\in\{0,1\} & \text{for all } i\in N, k\in M. \end{array}$$

is well known in the literature (usually in its maximization version) as the *Multiple Knapsack Problem*, and it is the object of study in the first part of this thesis (see Chapters 3 to 6).

1.3.2 Second simplification: the Multicut Problem on Hypergraphs

In the simplification described in the last section the nets are not taken in account at all. We cannot expect to obtain very good solutions of the real problem we are interested in, by solving this simplification. Even, it is also not true that an optimal solution of the Multiple Knapsack Problem always leads to a feasible solution of the Module Design Problem, since the cut constraints are neglected.

The second simplification is closer to the original problem. We consider both variable sets (x_{ik}) and (y_{tk}) described in Section 2.2. We use the same approximation to the internal costs as in the first simplification. The external costs can be calculated exactly with variables (y_{tk}).

For the definition of the (y_{tk}) variables we use a different set of constraints as given in Section 2.2.

(3.1) $x_{ik} + x_{jl} - y_{tk} \leq 1$ for all $t \in Z, i, j \in T_t, i \neq j$ and $k, l \in M, k \neq l$,

i.e., if some items of T_t are assigned to different modules k and l, the corresponding y_{tk} and y_{tl} variables must be one.

(3.2) $\sum_{i \in T_t} x_{ik} + y_{tk} \leq |T_t|$ for all $t \in Z, k \in M$,

i.e., if all items of T_t are assigned to the same module k, variable y_{tk} must be zero.

(3.3) $y_{tk} - \sum_{i \in T_t} x_{ik} \leq 0$ for all $t \in Z, k \in M$,

i.e., if no item of T_t is assigned to module k, the variable y_{tk} must be equal to zero.

Then, the second simplification can be stated.

$$\begin{array}{ll}
\min & \sum_{k \in M} D_k \sum_{t \in Z} x_{ik} + \lambda \sum_{t \in Z} \sum_{k \in M} y_{tk} \\
s.t. & (2.1), (2.2), \\
 & (3.1), (3.2), (3.3) \\
 & (2.3), \\
 & x_{ik} \in \{0,1\}, \text{ for all } i \in N, k \in M, \\
 & y_{tk} \in \{0,1\}, \text{ for all } t \in Z, k \in M.
\end{array}$$

The formulation for the y_{tk} variables, although equivalent to the one given in Section 2.2, has the disadvantage that the number of constraints necessary to describe the problem is given by $\sum_{t \in Z} \frac{1}{2}|T_t|(|T_t|-1)|M|(|M|-1)$, while in the model given in Section 2.2, $5|Z||M|$ constraints suffice. There are two reasons to prefer this formulation. First, this formulation has less variables (only $(|Z| +$

$|N|)|M|$). On the other hand, our computational results (see Section 9.3) show that we do not need all these constraints in the solution process of the Branch and Cut Algorithm to solve the problem instances we are interested.

Now, we perform the same transformation as in the last section to avoid equations (2.1), obtaining an equivalent problem with inequalties instead of the equations in (2.1), and a modified objective function. In our application it occurs that several nets link the same set of items. In order to reduce the number of variables involved, we introduce for each net $t \in Z$ a parameter $g_t \in \mathbb{N}$ representating the weight of net t ("number of copies"). The problem can be, then, equivalently written:

$$
\begin{array}{lll}
\min & \sum_{i \in N} \sum_{k \in M} c_{ik} x_{ik} + & \lambda \sum_{t \in Z} \sum_{k \in M} g_t y_{tk} \\
s.t. & \sum_{k \in M} x_{ik} \leq 1, & \text{for all } i \in N, \\
 & (2.2), & \\
 & (3.1), (3.2), (3.3) & \\
 & \sum_{t \in Z} g_t y_{tk} \leq S_k, & \text{for all } k \in M, \\
 & x_{ik} \in \{0,1\}, & \text{for all } i \in N, k \in M, \\
 & y_{tk} \in \{0,1\}, & \text{for all } t \in Z, k \in M.
\end{array}
$$

We call the problem formulated above *Multicut Problem on Hypergraphs* MPH. This problem is investigated in Chapters 7 to 9.

Chapter 2

The 0/1 Single Knapsack Problem

The 0/1 Single Knapsack Problem (short SKP) is among the most intensively studied problems in the Operations Research literature. It can be stated as follows.

> Given is a set N of items, each item i having a profit c_i and a weight f_i. It is also given a capacity F of the knapsack. We look for a subset $B \subseteq N$, whose total weight does not exceed the knapsack capacity, and with maximum profit.

This problem appears as a subproblem in many complex applications, and this fact has been motivating several different approaches for solving it. The aim of this chapter is to present some of these approaches, and state some important definitions used in the following chapters. It is not our intention to provide a complete overview of the results of the literature.

We begin the discussion about the SKP in Section 3.1 with a formal presentation and a formulation using 0/1 variables. We present Dantzig's algorithm for the solution of the linear programming relaxation of this formulation. In Section 3.2 we present a dynamic programming algorithm for the problem. Section 3.3 focusses on approximative algorithms. Polynomial and fully polynomial approximation schemes are described. Finally, Section 3.4 addresses the Single Knapsack Polytope. In this section we summarize some of the main results of the literature on facet defining inequalities for the Single Knapsack Polytope. Also some important definitions used in the rest of this thesis are presented.

We would like to finish this introduction to the SKP with some historical notes. In Schrijver's excellent survey on the history of integer linear programming [Sc87], one can see that the SKP has been an interesting question for many centuries. Of course, after the "invention" of linear programming, a substantial interest arose in the solution of this type of problems. The SKP is especially interesting from a historical point of view, because one can use it to show, during the last decades, the "trends" in Combinatorial Optimization. In the 50's enumeration techniques were of much interest. In this context, Bellman suggests in [Be54] a dynamic programming algorithm for the SKP. Dantzig in [D57] proposed a fast algorithm for solving the linear relaxation of the SKP, which we present in the next section, and which is used until today to obtain fast bounds for the optimal solution of the problem. It was a decisive step for the presentation of branch and bound algorithms for the problem. They dominate the literature about the SKP during the last years of the 60's and the early 70's. With the development of the theory of $\mathcal{NP}$-completeness, approximative algorithms has become more interesting. In 1975, Sahni presented a polynomial approximation scheme for the SKP. In the same year, Ibarra and Kim [IK75] suggested a fully polynomial approximations scheme. In the late 70's, the first polyhedral investigations to the problem are made by Balas [B75], Hammer, Johnson and Peled [HJP75] and Wolsey [W75]. In the 80's large scale instances have been attacked, and several different generalizations of the problem (e.g. Multiple Choice Knapsack Problem, Knapsack Problem with GUB Constraints, and so on) appear in the literature.

2.1 Presentation of the Problem

The 0/1 Single Knapsack Problem SKP can be stated as follows.

> **Problem:** SKP.
> **Instance:** A finite set N, $f \in \mathbb{N}^N$, $F \in \mathbb{N}$ and $c \in \mathbb{N}^N$.
> **Question:** Find a subset $B \subseteq N$, such that $\sum_{i \in B} f_i \leq F$ and $\sum_{i \in B} c_i$ is maximum.

The problem has been proved to be $\mathcal{NP}$-hard by Karp in [K72]. We present a proof for this fact in Lemma 3.1.1, where we prove that the decision problem associated with SKP is $\mathcal{NP}$-complete. The decision problem associated with SKP (short SKP_d) is defined as follows.

> **Problem:** SKP_d.
> **Instance:** A finite set N, $f \in \mathbb{N}^N$, $F \in \mathbb{N}$, $c \in \mathbb{N}^N$ and $k \in \mathbb{N}$.
> **Question:** Is there a subset $B \subseteq N$, such that $\sum_{i \in B} f_i \leq F$ and $\sum_{i \in B} c_i \geq k$?

Lemma 2.1.1 *The problem SKP_d is $\mathcal{NP}$-complete.*

Proof. First, given a set $B \subseteq N$ one can verify in $O(|N|)$ if it is a solution of SKP_d. Then, SKP_d is in $\mathcal{NP}$.

In order to complete the proof, we show that Partition can be reduced to SKP_d in polynomial time. The problem Partition is $\mathcal{NP}$-complete (cf. [K72]) and can be stated as follows.

Problem: Partition.
Instance: A finite set A and $s \in \mathbb{N}^A$.
Question: Is there a subset $A' \subseteq A$, such that $\sum_{a \in A'} s_a = \sum_{a \in A \setminus A'} s_a$

Now, given an instance $\mathcal{I}_P$ of Partition, we construct an instance $\mathcal{I}_K$ of SKP$_d$ in the following way: set $N := A$, $f_a = c_a := s_a$ for all $a \in A$ and $F = k := \frac{\sum_{a \in A} s_a}{2}$. We prove that a set $A' \subseteq A$ is a solution of $\mathcal{I}_P$ if and only if A' is a solution of $\mathcal{I}_K$.

Let $A' \subseteq A$ a solution of $\mathcal{I}_P$. Then, $\sum_{a \in A'} s_a = \sum_{a \in A \setminus A'} s_a$. Thus, $\sum_{a \in A'} f_a = F$ and $\sum_{a \in A'} c_a = k$, which guarantees that A' is a solution of $\mathcal{I}_K$.

Conversely, suppose that A' is a solution of $\mathcal{I}_K$. Then, $\sum_{a \in A'} f_a \leq F$ and $\sum_{a \in A'} c_a \geq k$. Substituting according to the transformation above we obtain $\sum_{a \in A'} s_a = \frac{\sum_{a \in A} s_a}{2}$. Therefore, $\sum_{a \in A'} s_a = \sum_{a \in A \setminus A'} s_a$. ■

In order to model this problem using integer programming, let us introduce the variables x_i with the interpretation, $x_i = 1$ if item i is in the solution and 0, otherwise. An integer programming formulation for the SKP is given via

$$
\begin{array}{lll}
\max & \sum_{i \in N} c_i x_i & \\
s.t. & \sum_{i \in N} f_i x_i & \leq F, \\
 & x_i \in \{0,1\} & \text{for all } i \in N.
\end{array} \tag{3.1}
$$

Dantzig's bound

Dantzig suggests in [D57] a fast polynomial algorithm for solving the linear relaxation of the problem given above, i.e., we substitute the constraints $x_i \in \{0,1\}$ by $0 \leq x_i \leq 1$ for all $i \in N$. This procedure makes possible the appearance of several branch and bound strategies for the problem. In the implementation of our branch and cut approach, we need to solve some knapsack problems in heuristic procedures. We use also the strategy described in this section to solve the linear relaxation of these problems. We refer to it as Dantzig's procedure.

Consider a SKP given by (3.1). Let $n = |N|$ and suppose that the items are sorted such that

$$\frac{c_1}{f_1} \geq \frac{c_2}{f_2} \geq \ldots \geq \frac{c_n}{f_n}.$$

Let us also define the *critical element* s as $s := \min\{i \mid \sum_{j=1}^{i} f_j > F\}$. Then, the solution of the linear relaxation of (3.1) can be stated as follows.

Theorem 2.1.2 *The optimal solution x^* of the linear relaxation of (3.1) is*

$$x_i^* := \begin{cases} 1, & \textit{if } 1 \leq i \leq s-1; \\ 0, & \textit{if } s+1 \leq i \leq n; \\ \frac{F - \sum_{j=1}^{s-1} f_j}{f_s}, & \textit{if } i = s. \end{cases}$$

Proof. Let x' be an optimal solution of the linear relaxation of (3.1), and suppose $\sum_{i \in N} c_i x'_i > \sum_{i \in N} c_i x^*_i$. Suppose, now, that there is an index $i_1 < s$ for which $x^*_{i_1} = 1$ and $x'_{i_1} < 1$. In this case, there must be an index $i_2 \geq s$ such that $x'_{i_2} > x^*_{i_2}$. Choose the biggest $i_2 \in N$ with this property.

Now, define $\epsilon = \min\{x'_{i_2} - x^*_{i_2}, \frac{f_{i_2}}{f_{i_1}}(x^*_{i_1} - x'_{i_1})$, and consider the vector x'' as bellow.

$$x''_i := \begin{cases} x'_i + \frac{f_{i_2}\epsilon}{f_{i_1}}, & \text{if } i = i_1; \\ x'_i - \epsilon, & \text{if } i = i_2; \\ x'_i, & \text{otherwise.} \end{cases}$$

The vector x'' is feasible, and $\sum_{i \in N} c_i x''_i = \sum_{i \in N} c_i x'_i + c_{i_1}\frac{f_{i_2}}{f_{i_1}}\epsilon - c_{i_2}\epsilon$. But, by the choice of i_2, $\frac{c_{i_1}}{f_{i_1}} > \frac{c_{i_2}}{f_{i_2}}$, and then, x'' has a better objective function value than x', contradicting our assumption that x' was an optimal solution.

Analogously, one can prove that $x'_{i_2} = 0$ for all $i_2 > s$. ■

The procedure is clearly linear, if the items are already sorted as assumed. This initial sorting is considered the bottleneck for solving the problem by many approaches. Balas and Zemel prove in [BZ80], however, that the critical element can be found in linear time even if the items are not sorted.

Let z^* be the optimal solution value of the linear programming relaxation. We call the value $\lfloor z^* \rfloor$ Dantzig's bound. This bound was the best known for the SKP until 1977. Upper bounds dominating Dantzig's bound are first suggested by Martello and Toth in [MT77]. The idea of the authors is to consider the two problems obtained by setting the critical variable (the variable associated with the critical element) to zero and to one. Then, Dantzig's bound is calculated for both subproblems, and the maximum among these values is an upper bound for SKP that dominates Dantzig's bound. The reader interested in branch and bound approaches, using Dantzig's bound and improved ones, can find some papers on the subject in, for example, [HS74] and [MT77]. A good survey on the subject is given by [MT91].

2.2 Dynamic Programming Algorithm

In this section we present an algorithm for the SKP based on dynamic programming techniques. The algorithm has been first suggested by Bellman in [Be54] and is the classical example for introducing dynamic programming in almost all text books.

Let us first define some symbols. Suppose that an instance (N, f, F, c) of the SKP is given. For $I \subseteq N$, $\hat{F} \leq F$, set

$$g(I, \hat{F}) := \max\{\sum_{j \in I} c_j x_j \mid \sum_{j \in I} f_j x_j \leq \hat{F}, x_j \in \{0,1\} \text{ for all } j \in I\}.$$

The value $g(I, \hat{F})$ is the optimal solution value of a subinstance of the problem we are interested in. The idea in dynamic programming is to solve small instances of the SKP and combine the solutions to obtain a solution of the original problem. It can be summarized in the following recursion formula:

$$g(I, \hat{F}) = \begin{cases} -\infty, & \text{if } I = \emptyset; \\ g(I \setminus \{i\}, \hat{F}), & \text{for } i \in I, 1 \leq \hat{F} \leq f_i - 1; \\ \max(g(I \setminus \{i\}, \hat{F}), g(I \setminus \{i\}, \hat{F} - f_i) + c_i), & \text{for } i \in I, f_i \leq \hat{F} \leq F. \end{cases}$$

The optimal solution value of the original problem is given by $g(N, F)$. In order to avoid calculating any subproblem more than once, one can implement the dynamic programming algorithm as filling out an array with F elements. The algorithm has time complexity $O(|N|F)$ and space complexity $O(F)$. Observe that this algorithm is not polynomially bounded in the encoding length ($|N|$). We say that this algorithm is *pseudo polynomial*. An algorithm is called a *pseudo polynomial time algorithm* if its time complexity function is bounded above by a polynomial function depending on the encoding length of the instance and on the magnitude of the largest number of the instance.

Although simple, the algorithm can be useful from both, a theoretical and a practical point of view. In Section 3.4.2 we observe that the lifted coefficients of the Minimal Cover Inequalities can be calculated using the dynamic programming approach, and, since in that case the knapsack capacity is polynomially bounded in the instance length, the procedure has polynomial time complexity. Another application of the dynamic programming approach arises in the next section, in the presentation of Ibarra and Kim's fully polynomial approximation scheme to the SKP. In order to solve large scale instances of the SKP some authors use this approach combined with branch and bound techniques to solve small instances exactly (see [MT91] for details on such approaches).

2.3 Approximative algorithms

Once again the SKP is largely used in introductory books on Combinatorial Optimization for introducing important concepts such as *polynomial* and *fully polynomial approximation schemes*. In this section we will present these ideas. In the remainder of this section we present some simple, but useful heuristic procedures to SKP. Suppose that an instance (N, f, F, c) of the SKP is given and let z^* be the value of an optimal solution of SKP.

A first simple heuristic algorithm for SKP is the *profit greedy heuristic*. Let $n := |N|$ and suppose that the items are sorted such that $c_1 \geq \ldots \geq c_n$. Now, initialize $F' := F$ and $z^P := 0$. F' contains the rest capacity of the knapsack and z^P the value of the heuristic solution. For each item $i \in N$ considered in the order above, if $f_i \geq F'$ insert i in the heuristic solution and update $F' := F' - f_i$ and

$z^P := z^P + c_i$. No universal guarantee can be given on the quality of the solution produced by this heuristic, as we show in Example 2.3.1.

Example 2.3.1 Let $k \in \mathbb{N}$, and define $N = \{1, \ldots, k+1\}$, $f_1 = k$, $f_i = 1$ for $i = 2, \ldots, k+1$, $F = k$, $c_1 = 2$ and $c_i = 1$ for all $i = 2, \ldots, k+1$. Then, $z^P = 2$ and $z^* = k$. Thus no guarantee for the quality of the value can be given. ■

Another simple approximation algorithm for SKP works as follows. Let us introduce for each item $i \in N$ the *weight density* $\frac{c_i}{f_i}$. Suppose now that the items are sorted in decreasing order according to the weight densities:

$$\frac{c_1}{f_1} \geq \frac{c_2}{f_2} \geq \ldots \geq \frac{c_n}{f_n}.$$

Now, determine $t := \max\{i \in N \mid \sum_{j=1}^{i} f_j \leq F\}$. The value of the *weight density greedy heuristic* z^W is given by $z^W := \sum_{i=1}^{t} c_i$. As for the profit greedy heuristic, no universal guarantee can be given on the quality of the solution, as we show in Example 2.3.2.

Example 2.3.2 Let $n = 2$, $f_1 = c_1 = 1$, $f_2 = c_2 = K$, $F = K$. Then, $z^W = 1$, and $z^* = K$. Therefore, $\frac{z^* - z^W}{z^*} = \frac{K-1}{K}$ and no ϵ guarantee can be given. ■

However, if we combine both heuristic procedures showed above in the following way, a guarantee can be given. Let $z^G = \max(z^P, z^W)$. The following lemma shows that we can give a quality guarantee for the value z^G. We call this combination as *greedy heuristic*.

Lemma 2.3.3 $z^G \geq \frac{z^*}{2}$.

Proof. Let z_{LP} be the optimum value of the linear relaxation of the knapsack problem. Observe that $z^* \leq z_{LP}$. Moreover, if s is the critical element, then

$$\begin{array}{rcl} z_{LP} & < & \sum_{i=1}^{s} c_i \\ & = & \sum_{i=1}^{s-1} c_i + c_s \\ & \leq & z^W + z^P \\ & \leq & 2z^G. \end{array}$$

Thus, $z^G > \frac{z^*}{2}$. ■

2.3.1 Polynomial time approximation scheme

An approximative algorithm A is called a *polynomial time approximation scheme* if, given an instance of the problem and a "accuracy parameter" $\epsilon > 0$, A produces in polynomial time in the encoding length of the instance and ϵ a solution z^{A_ϵ}, such that $z^{A_\epsilon} \geq (1-\epsilon)z^*$, where z^* is the value of an optimal solution. The first approximation scheme suggested in the literature for the SKP was given by Sahni in [Sa75].

The idea is to use the greedy heuristic given in the last section combined with partial enumeration to obtain the desired bound.

Given $\epsilon = \frac{1}{k+1}$, consider the following scheme $S(k)$. All subsets $M \subseteq N$, with $|M| \leq k$ and $\sum_{i \in M} f_i \leq F$, are considered as candidates. For each candidate, the reduced problem in which the items in M are in the solution is considered. For the remaining variables, apply then the greedy heuristic. This task is executed for each candidate. The best solution found in all iterations is chosen as result $z^{S(k)}$. It is not difficult to see that Sahni's procedure is polynomial in the encoding length of the instance but not in the accuracy parameter. Supposing that the items are sorted according to the order assumed in the greedy heuristic, the procedure has time complexity $O(|N|)$. Since it is executed, at most, for all subset of N with less than or equal to k elements, the total time complexity is $O(|N|^{k+1})$. The following theorem due to Sahni [Sa75] states that the solution given by the scheme has the desired guarantee.

Theorem 2.3.4 *Let $z^{S(k)}$ be the solution value given by the scheme $S(k)$. Then,*

$$z^{S(k)} \geq (1 - \frac{1}{k+1}) z^*.$$

Proof. Let B be an optimal solution. If $|B| \leq k$, then the optimal solution would be found by the procedure. Suppose, then, that $|B| = k + r$, $r > 0$. Let $M' \subseteq B$ the subset with $|M'| = k$ and containing the elements of B with maximum profit. Observe that M' is one of the candidates analyzed in the heuristic approach. Consider the iteration when set M' is analyzed. Let z_r^G be the value of the greedy heuristic applied to the reduced problem. Now, let $j_1, \ldots, j_r$ be the items in $B \setminus M'$, and suppose they are sorted such that $\frac{c_{j_1}}{f_{j_1}} \geq \ldots \geq \frac{c_{j_r}}{f_{j_r}}$. Let j_m be the first item in the optimal solution not presented in the heuristic solution.

We derive some observations. First, $c_{j_i} \leq \frac{z^*}{k+1}$ for all $i = 1, \ldots, r$. Since j_m was not selected by the greedy heuristic, we can conclude that the residual capacity is less than f_{j_m}. In this case, since the items are sorted, no more than c_{j_m} as profit can be achieved. Moreover, until item j_m the heuristic solution selected only items present in the optimal solution. Thus,

$$\begin{aligned} z^* &= \textstyle\sum_{i \in M'} c_i + \sum_{i=1}^{m-1} c_{j_i} + \sum_{i=m}^{r} c_{j_i} \\ &\leq \textstyle\sum_{i \in M'} c_i + z_r^G + c_{j_m} \\ &\leq z^{S(k)} + c_{j_m} \\ &\leq z^{S(k)} + \frac{z^*}{k+1} \end{aligned}$$

Therefore, $z^{S(k)} \geq (1 - \frac{1}{k+1}) z^*$. ■

The bound given by the theorem above is also tight as the following Example 2.3.5 shows.

Example 2.3.5 Given k, let $n = k+2$, $c_1 = 2$, $f_1 = 1$ and $c_i = f_i = K > 2$, for $i = 2, \ldots, n+2$ and $F = (k+1)K$. The optimal solution value is $z^* = (k+1)K$, but the solution value given by the scheme is $z^{S(k)} = kK + 2$. ■

2.3.2 Fully polynomial approximation schemes

Sahni's algorithm produces a solution as good as desired, but the time complexity of the scheme increases with the desired guarantee. An algorithm A is called a *fully* polynomial approximation scheme if A takes as input both an instance of the problem and an "accuracy requirement" ϵ and produces in polynomial time in the input length and $\frac{1}{\epsilon}$ a solution z^{A_ϵ} such that $z^{A_\epsilon} \geq (1-\epsilon)z^*$.

Ibarra and Kim present in [IK75] a fully polynomial approximation scheme for the SKP. In the remainder of this section we present the basic ideas of the algorithm.

Let (N, f, F, c) be an instance of SKP and suppose the items sorted such that

$$\frac{c_1}{f_1} \geq \ldots \geq \frac{c_n}{f_n}.$$

Calculate now the critical element s. The bound $z_{est} := \sum_{i=1}^{s} c_i$ will be utilized to determine two parameters: s, a scaling parameter, and t, a threshold parameter. Ibarra and Kim use $s = \frac{z_{est}\epsilon^2}{9}$ and $t = \frac{z_{est}\epsilon}{3}$.

With t, divide the index set into two parts, the set S containing the small items (items with $c_i < t$) and the set L with the large ones (items with $c_i \geq t$). The idea is to solve exactly the problem for the large items and enlarge this solution using the small ones, in a greedy like way.

Now, consider the following problem

$$\begin{array}{rll} z(d) = & \min & \sum_{i \in L} f_i x_i \\ & s.t. & \sum_{i \in L} \lfloor \frac{c_i}{s} \rfloor x_i = d, \\ & & x_i \in \{0,1\} \text{ for all } i \in L. \end{array}$$

Observe that $z(d)$ can be calculated using the dynamic programming algorithm discussed in Section 3.2. Calculate it for $d = 0, \ldots, \lfloor \frac{z_{est}}{s} \rfloor$. Actually, the implementation suggested in Section 3.2 provides the solution for all these problems with only one call to the routine. Now, with each of the optimal solutions of the scaled knapsacks on the large instances, try to enlarge it, in a greedy way, using the small items in S for it. The heuristic solution given by the procedure is the best one found during the process.

It is not difficult to see that the algorithm suggested is polynomial in the encoding length of the instance and $\frac{1}{\epsilon}$. First, the items must be sorted, requiring $O(|N| \log |N|)$ time. Then, the critical element is found and the estimate for the optimal value is determined. It consumes time $O(|N|)$. To divide the items into S and L one needs time of order $O(|N|)$. Actually, it is not necessary to sort the items to find the critical element and divide the items set into two parts. Lawler applies median finding techniques to prove in [La79] that it can be done in $O(|N|\log\frac{1}{\epsilon})$. Then, for solving exactly the knapsack problem to find the value $z(d)$ one needs, cf. Section 3.2, $O(|N|d)$ or $O(|N|\frac{z_{est}}{s})$. Finally, we execute $\lfloor\frac{z_{est}}{s}\rfloor+1$ times the greedy procedure (that has time complexity $O(|N|)$) to enlarge each optimal solution on the large items to a heuristic solution. So, the total time complexity can be given by $O(|N|(\log|N|+\frac{z_{est}}{s}))$. Using the choice of the authors for the parameter s, we obtain the time complexity $O(|N|(\log|N|+\frac{1}{\epsilon^2}))$ (or, in Lawler's improved version, $O(|N|\log\frac{1}{\epsilon}+\frac{1}{\epsilon^4}))$. In the next theorem, we show the quality guarantee that can be achieved by the heuristic solution found.

Theorem 2.3.6 *Let z^{IK} be the value given by the heuristic procedure. Then,*

$$z^{IK}\geq z^*-(\frac{s}{t}z^*+t).$$

Proof. Let x^* be an optimal solution of SKP. Let $d:=\sum_{i\in L}\lfloor\frac{c_i}{s}\rfloor x_i^*$. Observe that $d\leq\lfloor\frac{z_{est}}{s}\rfloor$, therefore, it was considered during the execution of the procedure. Let x' be the best solution found by the algorithm in this step, and compare the optimal solution with this heuristic solution.

Note that

$$\sum_{i\in L}c_ix_i^*-\sum_{i\in L}c_ix_i' \leq s\sum_{i\in L}\lfloor\tfrac{c_i}{s}\rfloor x_i^*+\sum_{i\in L}(c_i-s\lfloor\tfrac{c_i}{s}\rfloor x_i^*)-s\sum_{i\in L}\lfloor\tfrac{c_i}{s}\rfloor x_i',$$

observing that x' is optimal in L

$$\begin{aligned} &= \sum_{i\in L}(c_i-s\lfloor\tfrac{c_i}{s}\rfloor x_i^*)\\ &\leq s\sum_{i\in L}x_i^*\\ &\leq \tfrac{s}{t}\sum_{i\in L}c_ix_i^*\\ &\leq \tfrac{s}{t}z^*. \qquad \text{(I)}\end{aligned}$$

On the other hand, x' was enlarged using the greedy heuristic on the small elements. Observe, also, that in x^* is a optimal solution of this reduced problem (otherwise a better one would exist). Moreover, in Lemma 3.3.2 we observed that the difference between the value of the greedy solution and the optimal value is bounded by the maximum profit $c_{i_{max}}$. Then,

$$\sum_{i\in S}c_ix_i^*-\sum_{i\in S}c_ix_i'\leq c_{i_{max}}\leq t. \quad \text{(II)}$$

Finally, observe that

$$\begin{aligned} z^{IK}-z^* &\geq \sum_{i\in N}c_ix_i'-\sum_{i\in N}c_ix_i^*\\ &= \sum_{i\in L}c_ix_i'-\sum_{i\in L}c_ix_i^*+\sum_{i\in S}c_ix_i'-\sum_{i\in S}c_ix_i^*,\end{aligned}$$

using (I) and (II)

$$\geq -\tfrac{s}{t}z^*-t.$$

Therefore, $z^{IK} \geq z^* - (\frac{s}{t}z^* + t)$. ■

The following corollary proves that Ibarra and Kim's procedure has the desired quality guarantee.

Corollary 2.3.7 $z^{IK} \geq (1-\epsilon)z^*$.

Proof. Setting $s = \frac{z_{est}\epsilon^2}{9}$ and $t = \frac{z_{est}\epsilon}{3}$ in the formula given in Theorem 3.3.5, one obtains

$$\begin{array}{rcl} z^{IK} & \geq & z^* - (\frac{\frac{z_{est}\epsilon^2}{9}}{\frac{z_{est}\epsilon}{3}} z^* + \frac{z_{est}\epsilon}{3}) \\ & = & z^* - (\frac{z^*\epsilon}{3} + \frac{z_{est}\epsilon}{3}) \end{array}$$

since $z_{est} \leq 2z^*$

$$\begin{array}{rcl} & \geq & z^* - (\frac{z^*\epsilon}{3} + \frac{2z^*\epsilon}{3}) \\ & = & (1-\epsilon)z^* \end{array}$$

■

2.4 The Single Knapsack Polytope

In this section we present some of the main results in the literature about the Single Knapsack Polytope. The first investigations on this polytope go back to 1975, with the papers by Balas [B75], Hammer, Johnson and Peled [HJP75] and Wolsey [W75].

For an instance (N, f, F) of the SKP, the Single Knapsack Polytope can be defined as follows

$$SKP(N, f, F) := conv\{x \in \mathbb{R}^N \mid \sum_{i \in N} f_i x_i \leq F \text{ and } x_i \in \{0,1\} \text{ for all } i \in N\}.$$

Some trivial results are proven in the next lemmas.

Lemma 2.4.1 *Given an instance (N, f, F) of the SKP. The dimension of SKP (N, f, F) is equal to $|N| - |\{i \in N \mid f_i > F\}|$.*

Proof. Let $I := \{i \in N \mid f_i > F\}$. First, observe that $SKP(N, f, F) \subseteq \{x \in \mathbb{R}^N \mid x_i = 0$ for all $i \in I\}$. Then, $\dim(SKP(N, f, F)) \leq \dim(N \setminus I) = |N| - |I|$.

Now, consider that $e_i \in SKP(N, f, F)$ for all $i \in N$ such that $f_i \leq F$. Moreover, $\mathbf{0} \in SKP(N, f, F)$ and all these vectors are affinely independent. Then, $\dim(SKP(N, f, F)) = |N| - |I|$. ■

In the remainder of this chapter we suppose that $f_i \leq F$ for all $i \in N$, and therefore, the polytope is full dimensional.

Lemma 2.4.2 *The inequality $x_i \geq 0$ defines a facet of $SKP(N, f, F)$ for all $i \in N$.*

Proof. The inequality is clearly valid. Suppose that there is a facet defining inequality $bx \leq \beta$ such that $EQ(x_i \geq 0) \subseteq EQ(bx \leq \beta)$. Now, observe that $\mathbf{0}$ and e_j are elements of $EQ(x_i \geq 0)$, for all $j \in N \setminus \{i\}$. Then, $b_j = 0$ for all $j \in N \setminus \{i\}$. Then, $bx \leq \beta$ is a scalar multiple of $x_i \geq 0$, and, thus, it defines a facet of $SKP(N, f, F)$. ■

In the following section, we investigate facet defining inequalities for polytopes associated with subinstances of the SKP. Observe that our definition of the polytope SKP allows us to consider only some subset $V \subseteq N$. It is clear that, if $SKP(N, f, F)$ is full dimensional, then for $V \subseteq N$, $\dim(SKP(V, f, F)) = |V|$. For ease of notation we use throughout this chapter the symbol f instead of $f|_V$.

In the next section we present some of the main results on facet defining inequalities for the Single Knapsack Polytope. In Section 3.4.2 we consider the problem of *lifting* facet defining inequalities of polytopes associated with subinstances of $SKP(N, f, F)$. We finish this discussion about the Single Knapsack Polytope with an overview about other theoretical results on $SKP(N, f, F)$. Moreover, we prove that the separation problem for the Minimal Cover Inequalities is $\mathcal{NP}$-complete.

2.4.1 Facet defining inequalities for the Single Knapsack Polytope

In the paper of Balas [B75], Hammer, Johnson and Peled [HJP75] and Wolsey [W75] we can find a very important class of facet defining inequalities: the *Minimal Cover Inequalities.*

A set $S \subseteq N$ is a *cover* if $\sum_{i \in S} f_i > F$. The cover is *minimal* if $\sum_{i \in S \setminus \{s\}} f_i \leq F$ for all $s \in S$. Given a minimal cover S, we call the inequality

$$\sum_{i \in S} x_i \leq |S| - 1$$

the *Minimal Cover Inequality* corresponding to S.

Proposition 2.4.3 *Let $S \subseteq N$ be a minimal cover. Then the Minimal Cover Inequality defines a facet of the polytope $SKP(S, f, F)$.*

Proof. The inequality is clearly valid. Suppose, now, that there is a facet defining inequality $bx \leq \beta$, such that $EQ(\sum_{i \in S} x_i \leq |S| - 1) \subseteq EQ(bx \leq \beta)$. Let $s \in S$ and define $x^s \in \mathbb{R}^{|S|}$ as follows

$$x_i^s := \begin{cases} 1, & i \in S \setminus \{s\}; \\ 0, & \text{otherwise.} \end{cases}$$

Observe that $x^s \in EQ(\sum_{i\in S} x_i \leq |S|-1)$ for all $s \in S$. Then, given $s_1, s_2 \in S$, $bx^{s_1} = bx^{s_2}$, and thus, $b_{s_2} = b_{s_1}$. So, $bx \leq \beta$ is a scalar multiple of the Minimal Cover Inequality. ■

Actually, the three papers cited above consider different versions of this inequality. Balas considers the original polytope $SKP(N, f, F)$ and looks for necessary and sufficient conditions for the Minimal Cover Inequality (or a lifted version of it) to define a facet of this polytope. As we see in the next section, these inequalities can be efficiently generated from Minimal Cover Inequalities using the lifting procedure. Moreover, the author suggests an exponential time procedure for generating valid and facet defining inequalities based on minimal covers. Hammer, Johnson and Peled obtain essentially the same results as does Balas in the case of the Single Knapsack Polytope. The authors study a more general problem in their paper.

Wolsey uses in [W75] the same approach discussed above. Further, he observes that the lifting procedure due to Padberg [P75] provides a way to generate inequalities defining faces of higher dimensions. Another interesting observation of the author is that different inequalities can be obtained from minimal covers applying the operation of complementing (substituting the variable x_i by $1 - x_i$) and afterwards applying the lifting procedure. In Section 3.4.3 we stress the theoretical importance of this operation, and in the Chapter 6, in the presentation of our implementation, we show some results on generating violated inequalities to the Single Knapsack Problem using the same operation.

Another class of facet defining inequalities of $SKP(N, f, F)$ has been introduced by Padberg in [P80]: the (1,k)-Configuration Inequalities.

Suppose that an instance (N, f, F) of the SKP is given. A set $N' \cup \{z\}$, with $N' \subseteq N$ and $z \in N \setminus N'$ is called a *(1,k)-configuration* if

(i) $\sum_{j\in N'} f_j \leq F$;

(ii) $K \cup \{z\}$ is a minimal cover, for all $K \subset N'$, with $|K| = k$.

Given a (1,k)-configuration $N' \cup \{z\}$, and $T \subseteq N'$, $k \leq |T| = r \leq |N'|$, we call the inequality

$$(r-k+1)x_z + \sum_{i\in T} x_i \leq r$$

the *(1,k)-Configuration Inequality* corresponding to $N' \cup \{z\}$ and $T \subseteq N'$.

Proposition 2.4.4 *Suppose that an instance (N, f, F) of the SKP is given, and let $N' \cup \{z\}$ be a (1,k)-configuration. Then, for all $T \subseteq N'$, $|T| = r$, where $k \leq r \leq |N'|$, the corresponding (1,k)-Configuration Inequality defines a facet of $SKP(N' \cup \{z\}, f, F)$.*

Proof. Let $T \subseteq N'$, $|T| = r$, and suppose $ax \leq \alpha$ is the corresponding (1,k)-Configuration Inequality. First we prove that the inequality is valid for the polytope $SKP(N' \cup \{z\}, f, F)$. Suppose this is not the case and let $x' \in SKP(N' \cup \{z\}, f, F)$ with $ax' > \alpha$. Since $|T| = r$, we must have $x'_z = 1$. Then, we know that $\sum_{i \in T} x'_i \leq k - 1$ (as for all $K \subseteq N$, $|K| = k$, $K \cup \{z\}$ is a minimal cover). Then,

$$\sum_{i \in T} x'_i + (r - k + 1)x'_z \leq k - 1 + r - k + 1 \leq r,$$

a contradiction.

To prove that the inequality defines a facet of SK, suppose that there exists a facet defining inequality $bx \leq \beta$ such that $EQ(ax \leq \alpha) \subseteq EQ(bx \leq \beta)$. Observe that $x^1 := \chi^T$, and $x^1 + e_i$ are in $EQ(ax \leq \alpha)$ for all $i \in N' \setminus (T \cup \{z\})$. Then, $bx^1 = b(x^1 + e_i)$, and, therefore, $b_i = 0$ for all $i \in N' \setminus (T \cup \{z\})$.

Now, let $Q \subseteq T$ with $|Q| = k - 1$, and define $x^2 := \chi^{Q \cup \{z\}}$. Note that $x^2 \in EQ(ax \leq \alpha)$. Observe that for all $i \in T \setminus (Q \cup \{z\})$, and $j \in Q$, $x^2 - e_j + e_i \in EQ(ax \leq \alpha)$. Then, $bx^2 = b(x^2 - e_j + e_i)$, and, therefore, $b_j = b_i$. Since we can repeat it to each element in $T \setminus (Q \cup \{z\})$ and Q, we can conclude that there exists a constant b_t such that for all $i, j \in T$, $b_i = b_j = b_t$.

Observing, now, that x^1 and x^2 are in $EQ(ax \leq \alpha)$, we have $bx^1 = bx^2$, then, $rb_t = b_z + (k-1)b_t$, and therefore, $b_z = (r-k+1)b_t$. So, the inequality $bx \leq \beta$ is a scalar multiple of $ax \leq \alpha$, and, therefore, it defines a facet of $SKP(N' \cup \{z\}, f, F)$.

■

The Minimal Cover Inequality is a special case of the $(1, k)$-Configuration Inequality, since for all minimal cover S and $s \in S$, the set $S \setminus \{s\} \cup \{s\}$ is a (1,$|S| - 1$)-configuration.

Example 2.4.5 Consider the instance of the SKP with $n = 4$, $f_1 = f_2 = f_3 = 3$, $f_4 = 7$ and $F = 9$. Then, $N' = \{1, 2, 3\} \cup \{4\}$ is a (1,1)-configuration. Thus, the inequalities,

$$\begin{array}{lll} x_1 + x_4 \leq 1 & x_2 + x_4 \leq 1 & x_3 + x_4 \leq 1 \\ x_1 + x_2 + 2x_4 \leq 2 & x_1 + x_3 + 2x_4 \leq 2 & x_2 + x_3 + 2x_4 \leq 2 \\ x_1 + x_2 + x_3 + 3x_4 \leq 3 & & \end{array}$$

define facets of the polytope $SKP(N' \cup \{4\}, f, F)$. ■

Observe that for a given (1,k)-configuration $N' \cup \{z\}$, we have

$$\sum_{r=k}^{|N'|} \binom{|N'|}{r}$$

distinct facets of the polytope $SKP(N' \cup \{z\}, f, F)$ induced by (1,k)-Configuration Inequalities.

In Section 3.4.3 we mention some more classes of facet defining inequalities for the $SKP(N, f, F)$.

2.4.2 The lifting procedure

As we see in Section 3.4.1, it usually makes our life easier to study subinstances of the problem and determine facet defining inequalities for the polytopes associated with these subinstances. But, we are interested in solving the original problem. To this end we must be able to obtain from these inequalities, facet defining inequalities of the original polytope. This idea is called *lifting*.

The origin of the lifting procedure goes back to Padberg, in [P75]. Let us define formally the concept of "lifting an inequality". For the ease of presentation, we introduce the definitions and procedures for the case of the Single Knapsack Polytope. The concept of lifting and the lifting procedure, however, apply in general.

Suppose we are given an instance (N, f, F) of SKP. Also a set $V \subseteq N$ and an inequality $ax \leq \alpha$ that is valid for $SKP(V, f, F)$ are also given. An inequality $a'x \leq \alpha$, $a' \in \mathbb{R}^N$, is called a *lifting* of $ax \leq \alpha$ if $a'_i = a_i$ for all $i \in V$. We are interested in the following problem.

Problem 2.4.6 *Given a set $V \subseteq N$ and a valid (resp. facet defining) inequality $ax \leq \alpha$ for $SKP(V, f, F)$. Find a lifting (or even all liftings) of this inequality that is valid (resp. facet defining) for $SKP(N, f, F)$.*

Padberg [P75] suggests a general procedure to calculate the coefficients of a'.

Sequential lifting procedure - Padberg [P75].
Input: An instance (N, f, F) of the SKP, $V \subseteq N$ and $ax \leq \alpha$ valid for $SKP(V, f, F)$
Output: $a' \in \mathbb{N}^n$ such that $a'x \leq \alpha$ is a lifting of $ax \leq \alpha$ and is valid for $SKP(N, f, F)$.

Initialize $a'_i := a_i$ for all $i \in V$.
Choose a sequence of the coefficients in $N \setminus V$.
Initialize $K := N \setminus V$.
While there is some item $k \in K$ calculate.

$$\begin{array}{rll} z_k := & \max & \sum_{i \in N \setminus K} a'_i x_i \\ & s.t. & \sum_{i \in N \setminus K} f_i x_i \leq F - f_k, \\ & & x_i \in \{0, 1\}, \text{ for all } i \in N \setminus K; \\ a_k := & \alpha - z_k. & \\ K := & K \setminus \{k\}. & \end{array}$$

Lemma 2.4.7 *Let (N, f, F) be an instance of SKP. Given $V \subseteq N$ and $ax \leq \alpha$ a valid inequality for $SKP(V, f, F)$, then $a'x \leq \alpha$ calculated as described above is valid for $SKP(N, f, F)$. Moreover, if $ax \leq \alpha$ defines a facet of $SKP(V, f, F)$, then $a'x \leq \alpha$ defines a facet of $SKP(N, f, F)$.*

Proof. First we prove that the inequality $a'x \leq \alpha$ is valid for $SKP(N, f, F)$. Let $K := N \setminus V$ as in the procedure above. Given a set $K \subseteq N \setminus V$ we denote by $a^K \in \mathbb{R}^N$ the vector obtained by applying the procedure with the instance $(V \cup K, f, F)$ and the inequality $ax \leq \alpha$ ($a^{N \setminus V} = a'$). We prove by induction in $|K|$ that the inequality $a^K x \leq \alpha$ is valid for $SKP(V \cup K, f, F)$.

If $|K| = 0$, then the statement is clearly true. Consider now that $|K| \geq 1$, and let $k \in K$. Observe that, using the induction hypothesis, the inequality $a^{K \setminus \{k\}} x \leq \alpha$ is valid for $SKP(V \cup K \setminus \{k\}, f, F)$. Suppose that there exists a vector $x' \in SKP(V \cup K, f, F)$ such that $a^K x' > \alpha$. Then, $x'_k = 1$. But, by the algorithm described above, the coefficient $a^K_k = \alpha - z_k$, where $z_k = \max\{a^{K \setminus \{k\}} x \mid x \in SKP(V \cup K \setminus \{k\}, f, F - f_k)$. A contradiction.

Now we prove by induction in $|K|$ that if the inequality $ax \leq \alpha$ defines a facet of $SKP(V, f, F)$, then $\dim(EQ(SKP(V \cup K, f, F), a^K x \leq \alpha)) = |V| + |K|$.

If $|K| = 0$, then the statement is clearly true. Consider now that $|K| \geq 1$ and let $k \in K$. Using induction hypothesis, $\dim(EQ(SKP(V \cup K \setminus \{k\}, f, F), a^{K \setminus \{k\}} x \leq \alpha)) = |V| + |K| - 1$. Let B be a basis of this set. By the algorithm described above, there exists a $x^* \in SKP(V \cup K \setminus \{k\}, f, F - f_k)$ such that $a^{K \setminus \{k\}} x^* = z_k$. Then, $x^* + e_k$ is an element of $EQ(SKP(V \cup K, f, F), a^K x \leq \alpha)$.

Now, for all vectors $x \in B$ extend it to a vector x' in $\mathbb{R}^{V \cup K}$ by setting $x'_k = 0$ and $x'_i = x_i$ for all $i \in V \cup K \setminus \{k\}$. Let B' the set containing all extended vectors. Note that $x' \in EQ(SKP(V \cup K, f, F), a^K x \leq \alpha)$ for all $x' \in B'$. Moreover, B' and $x^* + e_k$ are affinely independent. Thus, $\dim(EQ(SKP(V \cup K, f, F), a^K x \leq \alpha)) = |V| + |K|$. ■

Example 2.4.8 Let us consider the following instance of the SKP:

$$4x_1 + 5x_2 + 5x_3 + 6x_4 + 6x_5 + 7x_6 \leq 14.$$

The set $S = \{3, 4, 5\}$ is a minimal cover, and therefore the inequality $x_3 + x_4 + x_5 \leq 2$ defines a facet of $SKP(\{3, 4, 5\}, f, F)$. Applying Padberg's procedure to the sequence $(1, 2, 6)$ we obtain the facet defining inequality $x_1 + x_3 + x_4 + x_5 + x_6 \leq 2$. If we apply the procedure to the sequence $(2, 1, 6)$ we obtain a different lifted inequality $x_2 + x_3 + x_4 + x_5 + x_6 \leq 2$. ■

In order to obtain all lifted inequalities that this procedure can yield, all possible sequences must be tried. Another problem arises in Padberg's procedure. In step 3. an integer program must be solved for all $k \in N \setminus V$ exactly, and we know that no efficient algorithm is known to execute this job in general. However, Zemel could prove in [Ze89] that for the facet defining inequalities of the Single Knapsack Polytope, this task can be accomplished in polynomial time. Consider the following "dual" problem:

$$\begin{array}{rll} d_k(z) := & \min & \sum_{j \in N \setminus K} f_j x_j \\ & s.t. & \sum_{j \in N \setminus K} a'_j x_j \quad \geq z \\ & & x_j \in \{0, 1\} \quad \text{for all } j \in N \setminus K. \end{array}$$

It is easy to see that $z_k = \max\{z \mid d_k(z) \leq F - f_k\}$. So, consider the following procedure to calculate the lifting coefficient:

Lifting procedure - Zemel [Ze89].
Input: An instance (N, f, F) of the SKP, $V \subseteq N$ and $ax \leq \alpha$ ($\alpha \in \mathbb{N}$) valid for $SKP(V, f, F)$.
Output: $a' \in \mathbb{R}^N$ such that $a'x \leq \alpha$ is a lifting of $ax \leq \alpha$ and is valid for $SKP(N, f, F)$.

Initialize $a'_i := a_i$ for all $i \in V$.
Choose a sequence of the coefficients in $N \setminus V$.
Intialize $K := N \setminus V$.
Calculate for $z = 0, \ldots, \alpha$

$$\begin{array}{rll} d_1(z) := & \min & \sum_{j \in V} f_j x_j \\ & s.t. & \sum_{j \in V} a_j x_j \geq z, \\ & & x_j \in \{0,1\}, \text{ for all } j \in V. \end{array}$$

Set $i := 1$.
While there exists a $k \in K$ calculate.

$$\begin{array}{rl} z_k := & \max\{z \mid d_i(z) \leq F - f_k\}. \\ a_k := & \alpha - z_k. \\ d_{i+1}(z) := & \begin{cases} d_i(z), & z = 0, \ldots, a_k - 1 \\ \min\{d_i(z), d_i(z - a_k) + f_k\} & z = a_k, \ldots, \alpha. \end{cases} \\ K := & K \setminus \{k\}. \\ i := & i + 1. \end{array}$$

Essentially, the idea is to apply the dynamic programming approach to solve the "dual" problem. The computational complexity of the dynamic programming procedure to calculate d_1 is given by $O(|N|\alpha)$ (see Section 3.2). In both, Minimal Cover or (1,k)-Configuration Inequalities, $\alpha = O(|N|)$. So, this calculation can be accomplished in $O(|N|^2)$. We execute the while $|N \setminus V|$ times. In each iteration j, a maximum is performed over a set of z elements (and, therefore, requires $O(|N|)$), and d_{j+1} is calculated for the next iteration, consuming also $O(|N|)$. Then this loop has time complexity $O(|N|^2)$, and this is also the total time complexity of the whole procedure. Although the description above suggests that $O(|N|^2)$ space is necessary, one only need the value of d in the current iteration. So, the space complexity is $O(|N|)$.

Padberg's approach described above is often called *sequential lifting*.

Some improvements can also be made for the case of lifting Minimal Cover Inequalities. Balas and Zemel [BZ78] could prove some lower and upper bounds to the value of the lifting coefficient of a variable. They observe also that either the bounds are equal (and, therefore, no effort is necessary to calculate the coefficient) or differ by one.

The sequential lifting procedure, however, does not provide all lifted inequalities from an inequality $ax \leq \alpha$ of $SKP(V, f, F)$. This is shown in Example 2.4.9.

Example 2.4.9 Consider the following instance of the SKP:

$$2x_1 + 2x_2 + 2x_3 + 2x_4 + 3x_5 + 3x_6 + 3x_7 + 5x_8 \leq 6.$$

Observe that $S = \{1, 2, 3, 4\}$ defines a minimal cover, and therefore, $x_1+x_2+x_3+x_4 \leq 3$ is a facet defining inequality of $SKP(\{1, 2, 3, 4\}, f, F)$. The inequality

$$x_1 + x_2 + x_3 + x_4 + \frac{3}{2}x_5 + \frac{3}{2}x_6 + \frac{3}{2}x_7 + 3x_8 \leq 3$$

is a lifted facet defining inequality of $SKP(N, f, F)$ but cannot be obtained by sequential lifting. ■

To this end Balas and Zemel provide in [BZ78] the following *simultaneous lifting* procedure. Let $\mathcal{M} := \{M \subseteq N \setminus V \mid \sum_{i \in M} f_i \leq F\}$. Now, for each $M \in \mathcal{M}$, let $a_M := \alpha - z_m$, where

$$\begin{array}{rll} z_M := & max & \sum_{i \in V} a_i x_i \\ & & \sum_{i \in V} f_i x_i \leq F - \sum_{i \in M} f_i \\ & & x_i \in \{0, 1\} \text{ for all } i \in V. \end{array}$$

Then, the inequality $\sum_{i \in V} a_i x_i + \sum_{i \in N \setminus V} b_i x_i \leq \alpha$ defines a facet of $SKP(N, f, F)$ if and only if b is a vertex of the polyhedron

$$T = \{t \in \mathbb{R}^{N \setminus V} \mid \sum_{i \in M} q_i \leq a_M, \text{for all } M \in \mathcal{M}\}.$$

It is clear that it is not trivial to implement such a procedure efficiently in general.

2.4.3 An overview on the Single Knapsack Polytope

In this section we give an overview on interesting theoretical questions about the Single Knapsack Polytope that have already been answered, or still remain open. It is not our objective to go into details in all results presented here, but to provide an overview on the results and give to the interested reader the corresponding references.

Some results on facet defining inequalities

Most of the work on facet defining inequalities for the Single Knapsack Polytope was made during the 70's. However, new classes appeared recently in a paper by Gottlieb and Rao [GR88], where the authors investigate overlapping and disjoint minimal covers and (1,k)-configurations, giving necessary and sufficient conditions such that such a combination defines a facet of the Single Knapsack Polytope.

Balas and Zemel prove in [BZ84] that using the operations of lifting and complementing (see a next subsection) all facet defining inequalities of positive 0-1 polytopes can be obtained from minimal covers.

Complete descriptions for polytopes associated with SKP with small values of the right hand side are investigated by Hammer and Peled [HP82] and Abdel Hamid [AH94]. Complete descriptions are given for polytopes associated with SKP with right hand side less than or equal to 7.

The computational complexity of several problems related to recognizing and lifting an inequality was studied by Hartvigsen and Zemel [HZ92]. Some examples of results they provide in this paper are:

i. Recognize whether an inequality with integer coefficients is the lifting of a minimal cover S.
Complexity class: $\mathcal{P}$;

ii. Given a knapsack constraint and an integer s, is there a minimal cover with cardinality s?
Complexity class: $\mathcal{NP}$-complete;

iii. Recognize whether a given inequality is valid for $SKP(N, f, F)$.
Complexity class: co-$\mathcal{NP}$-complete;

iv. Recognize whether a given inequality is a facet of $SKP(N, f, F)$.
Complexity class: $\mathcal{D}^p$.

The separation problem

Crowder, Johnson and Padberg conjecture in [CJP83] that the Separation Problem for the Minimal Cover Inequalities can be solved in polynomial time. Unfortunately, this is not the case unless $\mathcal{P} = \mathcal{NP}$, since we could prove that the separation problem for this inequality class is $\mathcal{NP}$-hard.

We will prove that the decision version of the separation problem for the Minimal Cover Inequalities SMC$_d$ is $\mathcal{NP}$-complete showing that the decision problem associated with th Single Knapsack Problem SKP$_d$, which is known to be $\mathcal{NP}$-complete (see Section 3.1).

Let us introduce the problems:

Problem: SMC$_d$.
Instance: A finite set N, $x \in [0,1]^N$, $a \in \mathbb{N}^N$ and $b \in \mathbb{N}$.
Question: Is there a set $S \subseteq N$ such that $\sum_{i \in S} a_i \geq b+1$, $\sum_{i \in S \setminus \{j\}} a_i \leq b$ for all $j \in S$ and $\sum_{i \in S} x_i > |S| - 1$?

Problem: SKP$_d$.
Instance: A finite set M, $f \in \mathbb{N}^m$, $F \in \mathbb{N}$, $c \in \mathbb{N}^M$ and $k \in \mathbb{N}$.
Question: Is there a set $S' \subseteq M$ such that $\sum_{i \in S'} f_i \leq F$ and $\sum_{i \in S'} c_i \geq k$?

Theorem 2.4.10 *SMC$_d$ is $\mathcal{NP}$-complete.*

Proof. We observe that SMC$_d$ belongs to $\mathcal{NP}$, since it can be verified whether a given set S satisfies the constraints in polynomial time ($O(|N|)$).

Let us show that SKP$_d$ can be reduced to SMC$_d$ in polynomial time.

Given an instance $\mathcal{I}_{SKP}$ of SKP$_d$, construct the following instance $\mathcal{I}_{SMC}$ of the SMC$_d$:

$$\begin{array}{lll} N & := M. & \\ x_i & := 1 - \frac{f_i}{F} + \epsilon & \text{for all } i \in N. \\ a_i & := c_i & \text{for all } i \in N. \\ b & := k - 1. & \end{array}$$

where $0 < \epsilon < \frac{1}{|N|F}$.

Now we will prove that $\mathcal{I}_{SKP}$ has a solution if and only if $\mathcal{I}_{SMC}$ has one.

Let S be a solution of $\mathcal{I}_{SMC}$. Then,

$$\sum_{i \in S} a_i \geq b + 1 \text{ and } \sum_{i \in S} x_i > |S| - 1.$$

By substitution,

$$\textstyle\sum_{i \in S} c_i \geq k, \text{ and } \sum_{i \in S}(1 - \frac{f_i}{F} + \epsilon) > |S| - 1,$$

and thus,

$$\begin{array}{rcl} |S|(1+\epsilon) - \frac{1}{F}\sum_{i \in S} f_i & > & |S| - 1, \\ \frac{1}{F}\sum_{i \in S} f_i & < & 1 + \epsilon|S|. \end{array}$$

Since $\epsilon < \frac{1}{|N|F}$, and $f_i \in \mathbb{N}$,

$$\textstyle\sum_{i \in S} f_i \leq F.$$

So, S is a solution of $\mathcal{I}_{SKP}$.

The converse can be proved as follows. Let S' be a solution of $\mathcal{I}_{SKP}$. Without loss of generality we can choose a minimal solution. Then,

$$\sum_{i \in S'} c_i \geq k \text{ and } \sum_{i \in S'} f_i \leq F.$$

Substituting c and k, we obtain,

$$\sum_{i \in S'} a_i \geq b + 1.$$

Since the solution is minimal, $\sum_{i \in S' \setminus \{j\}} a_i \leq b$ for all $j \in S$. Dividing the second inequality by $-F$, and adding $|S'|$ in both sides, we obtain

$$\begin{array}{rl} |S'| - \frac{1}{F} \sum_{i \in S'} f_i & \geq |S'| - 1, \\ \sum_{i \in S'} (1 - \frac{f_i}{F}) & \geq |S'| - 1. \end{array}$$

But, since $\epsilon > 0$,

$$\begin{array}{rl} \sum_{i \in S'} (1 - \frac{f_i}{F} + \epsilon) & > |S'| - 1, \\ \sum_{i \in S'} x_i & > |S'| - 1. \end{array}$$

So, S' is a solution of $\mathcal{I}_{SMC}$. ■

Boyd presents in [Bo92] a pseudo polynomial algorithm to solve this problem. The algorithm constructs a network flow problem with a pseudo polynomial number of vertices and edges. For the (1,k)-Configuration Inequality, it remains open whether the corresponding separation problem can be solved in polynomial time or whether it is $\mathcal{NP}$-hard.

Complementing

Complementing an inequality can be viewed as the converse operation of lifting. To our knowledge the idea of complemented variables was first suggested by Wolsey in [W75]. In the following we present a general procedure to calculate the value of the coefficient of the complemented variables in the resulting inequality in order to guarantee that it is valid or facet-defining. At this point, we would like to mention that the word *lifting* is reserved in this thesis for the operation studied in Section 3.4.2. Of course the ideas are very similar: in lifting we consider a variable not part of the problem, and investigate what happens when this variable is introduced. Here, the subproblem in which the item is selected is studied, and, when returning to the original problem, we must investigate what happens when this variable is excluded. We present the idea of complementing an inequality for the Single Knapsack Polytope, but, as in the case of lifting, it is applicable for general polytopes.

Suppose we are given an instance (N, f, F) of SKP. Given is a set $V \subseteq N$ and an inequality $ax \leq \alpha$ that is valid for $SKP(V, f, F - \sum_{i \in N \setminus V} f_i)$. An inequality $\overline{a}x \leq \overline{\alpha}$, $\overline{a} \in \mathbb{R}^N$ is called a *complementing* of $ax \leq \alpha$ if $\overline{a}_i = a_i$ for all $i \in V$ and $\overline{\alpha} \geq \alpha$.

Sequential complementing procedure
Input: An instance (N, f, F) of the SKP, $V \subseteq N$ and an inequality $ax \leq \alpha$ ($\alpha \in \mathbb{N}$) valid for $SKP(V, f, F - \sum_{i \in N \setminus V} f_i)$.
Output: $\overline{a} \in \mathbb{R}^n$, $\overline{\alpha} \in \mathbb{N}$ such that $\overline{a}x \leq \overline{\alpha}$ is a complementing of $ax \leq \alpha$ and is valid for $SKP(N, f, F)$.

Initialize $\overline{a}_i = a_i$ for all $i \in V$ and $\overline{\alpha} := \alpha$.

Choose a sequence of the coefficients in $N \setminus V$.
Initialize $K := N \setminus V$.
While there is some item $k \in K$ calculate.

$$\begin{array}{rcl} \mu_k & := & \max \ \ \overline{a}x \\ & & s.t. \ \ x \in SKP(N \setminus K, f, F - \sum_{i \in K \setminus \{k\}} f_i), \\ & & \ \ x_i \in \{0,1\}, \text{ for all } i \in N \setminus K; \\ \overline{a}_k & := & \mu_k - \overline{\alpha}. \\ \overline{\alpha} & := & \mu_k. \\ K & := & K \setminus \{k\}. \end{array}$$

Lemma 2.4.11 *Let (N, f, F) be an instance of SKP. Given $V \subseteq N$ and $ax \le \alpha$ a valid inequality for $SKP(V, f, F - \sum_{i \in N \setminus V} f_i)$, then $\overline{a}x \le \overline{\alpha}$ calculated as described in the procedure above is valid for $SKP(N, f, F)$. Moreover, if $ax \le \alpha$ defines a facet of $SKP(V, f, F - \sum_{i \in N \setminus V} f_i)$, then $\overline{a}x \le \overline{\alpha}$ defines a facet of $SKP(N, f, F)$.*

Proof. First we prove that the inequality $\overline{a}x \le \overline{\alpha}$ is valid for $SKP(N, f, F)$. Let $K := N \setminus V$ as in the procedure above. Given a set $K \subseteq N \setminus V$ we denote by $a^K \in \mathbb{R}^N$ and α^K the output obtained by applying the procedure with the instance $(V \cup K, f, F - \sum_{i \in N \setminus (V \cup K)} f_i)$ and the inequality $ax \le \alpha$ ($a^{N \setminus V} = \overline{a}$ and $\alpha^{N \setminus V} = \overline{\alpha}$). We prove by induction in $|K|$ that the inequality $a^K x \le \alpha^K$ is valid for $SKP(V \cup K, f, F - \sum_{i \in N \setminus (V \cup K)} f_i)$.

If $|K| = 0$, then the statement is clearly true. Consider now that $|K| \ge 1$, and let $k \in K$. Observe that, using the induction hypothesis, the inequality $a^{K \setminus \{k\}} x \le \alpha^{K \setminus \{k\}}$ is valid for $SKP(V \cup K \setminus \{k\}, f, F - \sum_{i \in N \setminus (V \cup K \setminus \{k\})} f_i)$. Suppose that there exists a vector $x' \in SKP(V \cup K, f, F - \sum_{i \in N \setminus (V \cup K)} f_i)$ such that $a^K x' > \alpha^K$. Since $\alpha^K \ge \alpha^{K \setminus \{k\}}$, then $x'_k = 1$. But, by the algorithm described above, the coefficient $a^K_k = \mu_k - \alpha^{K \setminus \{k\}}$, where $\mu_k = \max\{a^{K \setminus \{k\}} x \mid x \in SKP(V \cup K \setminus \{k\}, f, F - \sum_{i \in N \setminus (V \cup K)} f_i)\}$. A contradiction.

Now we prove by induction in $|K|$ that if the inequality $ax \le \alpha$ defines a facet of $SKP(V, f, F - \sum_{i \in N \setminus V} f_i)$, then $\dim(EQ(SKP(V \cup K, f, F - \sum_{i \in N \setminus (V \cup K)} f_i), a^K x \le \alpha^K)) = |V| + |K| - 1$.

If $|K| = 0$, then the statement is clearly true. Consider now that $|K| \ge 1$ and let $k \in K$. Using induction hypothesis, $\dim(EQ(SKP(V \cup K \setminus \{k\}, f, F - \sum_{i \in N \setminus (V \cup K \setminus \{k\})} f_i), a^{K \setminus \{k\}} x \le \alpha^{K \setminus \{k\}})) = |V| + |K| - 2$. Let B be a basis of this set. By the algorithm described above, there exists a $x^* \in SKP(V \cup K \setminus \{k\}, f, F - \sum_{i \in N \setminus (V \cup K \setminus \{k\}} f_i)$ such that $a^{K \setminus \{k\}} x^* = \mu_k = \alpha^K$. Then, x^* is an element of $EQ(SKP(V \cup K, f, F - \sum_{i \in N \setminus (V \cup K)} f_i), a^K x \le \alpha^K)$.

Now, for all vectors $x \in B$ extend it to a vector x' in $\mathbb{R}^{V \cup K}$ by setting $x'_k = 1$ and $x'_i = x_i$ for all $i \in V \cup K \setminus \{k\}$. Let B' the set containing all extended vectors. Note that $x' \in EQ(SKP(V \cup K, f, F - \sum_{i \in N \setminus (V \cup K)} f_i), a^K x \le \alpha^K))$ for all $x' \in B'$. Moreover, B' and x^* are affinely independent. Thus, $\dim(EQ(SKP(V \cup K, f, F - \sum_{i \in N \setminus (V \cup K)} f_i), a^K x \le \alpha^K)) = |V| + |K| - 1$. ∎

As in the case of lifting, the generated inequality depends on the order in which the elements are considered, and a 0/1 program must be solved.

By using similar techniques as described in [Ze89] we could prove that for the particular case of the single knapsack polytope, the computation of the complemented coefficients for the Minimal Cover and (1,k)-Configuration Inequalities can be made in polynomial time. We describe now the procedure to calculate these coefficients.

Let $K := N \setminus V$ and $k \in K$. To solve the integer programming problem given in the procedure efficiently, we use a similar idea as suggested by Zemel in [Ze89] for the lifting procedure. Observe that "dualizing" the problem above as follows, we obtain an equivalent one. Consider $d(\mu)$ the solution of the following problem:

$$\begin{array}{rll} d(\mu) = & \min & \sum_{j \in N \setminus K} f_j x_j \\ & s.t. & \overline{a}x \geq \mu, \\ & & x_j \in \{0,1\}, \text{ for all } j \in N \setminus K, \end{array}$$

and, observe that $\mu_k = \max\{\mu \mid d(\mu) \leq F - \sum_{i \in K \setminus \{k\}} f_i\}$. Using dynamic programming techniques the problem above can be solved in time complexity $O(|N|\mu_k)$. So, if we prove that there exists a polynomial upper bound for μ_k, depending on the input length, then the procedure can be accomplished in polynomial time.

Lemma 2.4.12 $\mu_k \leq k(\sum_{i \in V} a_i - \alpha)$.

Proof. A trivial upper bound for μ_k is given by

$$\mu_k \leq \sum_{i \in V} \overline{a}_i + \sum_{j \in K \setminus \{k\}} \overline{a}_j.$$

Consider a given order of the set K and observe that $\overline{a}_j = \mu_j - \mu_{j-1}$ for all $j \in K \setminus \{k\}$ (with $\mu_0 = \alpha$) and $\overline{a}_i = a_i$ for all $i \in V$. Then, we obtain

$$\mu_k \leq \sum_{i \in V} a_i + \mu_{k-1} - \alpha.$$

Solving the recursion formula we obtain $\mu_k \leq k(\sum_{i \in V} a_i - \alpha)$. ∎

For the case of Minimal Cover and (1,k)-Configuration Inequalities, $\sum_{i \in V} a_i - \alpha$ is bounded by $|N|$. Since $k \leq |N|$, a polynomial upper bound for μ_k is given by $|N|^2$. Then, the coefficient of the complemented variables can be calculated in polynomial time.

Example 2.4.13 Consider the instance of the SKP with $n = 4$, $f_1 = f_2 = f_3 = 3$, $f_4 = 7$ and $F = 9$. Now, suppose that variable x_4 is complemented. Then, we analyze the polytope $SKP(\{1,2,3\}, f, 2)$. The inequality $x_1 + x_2 + x_3 \leq 0$ is a facet defining inequality of this polytope. Solving the problem above, one obtains, $z_4 = 3$. Then, $x_1 + x_2 + x_3 + 3x_4 \leq 3$ is a facet defining inequality of $SKP(\{1,2,3,4\}, f, F)$. ∎

An interesting observation in Example 2.4.13 is that the resulting inequality is a (1,k)-configuration of the original problem. In Chapter 6 we show that this procedure can be used to generate many violated inequalities in the branch and cut process.

Chapter 3

A literature overview

In this chapter we survey the results of the literature on the Multiple Knapsack Problem and two other related problems: the Generalized Assignment Problem (short GAP) and the General Zero-One Programming Problem (short GZOP). It is not our intention to provide a complete overview on all approaches used to solve these problems. Instead, we choose some approaches that we think are representative. In Section 4.1 we present the Multiple Knapsack Problem (short MKP) and several different methods proposed in the literature for its solution. Section 4.2 addresses the GAP. In this section, we restrict our attention mainly to polyhedral investigations of the problem. Finally, in Section 4.3 we present the GZOP. We decided to include this problem in this overview, because it is usually called "Multiple Knapsack" in the literature. In this section, we present briefly some attempts to attack this problem.

3.1 The Multiple Knapsack Problem

In the literature there are several problems called "Multiple Knapsack Problem". We are interested in the following one:

> Given a set N of items with weights f_i, $i \in N$, a set M of knapsacks with capacities F_k, $k \in M$, and a profit function $c = (c_{ik})$ associated with assigning item i to knapsack k. We want to find an assignment of the items to the knapsacks such that the sum of the weights of the items assigned to each knapsack is less than or equal to the corresponding capacity and this assignment has maximum profit.

This problem arises in an application in the design of computer systems. Imagine a computer system with some modules and several hundreds of components that must be distributed onto these modules. One can model this problem as stated above, where f_i is associated with the area of the component i and F_k with the area of the module k. The function c must model the profit associated with assigning

i to k, which can be related to several technological constraints, for example, the number of nets linking components assigned to different modules, the occupation density of the modules and so on. The complete model and its relation with the Multiple Knapsack Problem is presented in Chapter 2.

This section addresses the Multiple Knapsack Problem. We begin, in Section 4.1.1 with a formal definition of the problem and a model using 0-1 variables. Moreover, a complexity analysis is presented. A brief overview on the exact approaches of the literature to the Multiple Knapsack Problem is presented in Section 4.1.2. Approximative algorithms are surveyed in Section 4.1.3.

3.1.1 Presentation of the problem

The Multiple Knapsack Problem MKP is the following:

> **Problem:** MKP
> **Instance:** Finite sets N and M, $f \in \mathbb{N}^N$, $F \in \mathbb{N}^M$ and $c \in \mathbb{N}^{N \times M}$.
> **Question:** Find a partition $\mathcal{B} = (B_1, \ldots, B_j, B_o)$ of N, such that $j \leq |M|$, $\sum_{i \in B_k} f_i \leq F_k$, for all $k \leq j$ and $\sum_{k \in M} \sum_{i \in B_k} c_{ik}$ is maximum.

When, for the profit vector $c_{ik} = c_i$ for all $i \in N$, $k \in M$ holds, we call the problem MKP *with uniform profit*. Let $n := |N|$ and $m := |M|$. The problem is clearly $\mathcal{NP}$-hard, since, when $m = 1$, we have the Single Knapsack Problem, which is known to be $\mathcal{NP}$-hard (see a proof for it in Section 3.1).

In order to model the problem stated above using an integer programming formulation, let us introduce variables $x_{ik} \in \{0,1\}^{mn}$ with the interpretation $x_{ik} = 1$, if item i is assigned to knapsack k and 0, otherwise. An integer programming formulation of the MKP is given via

$$
(4.1.1)\quad
\begin{array}{llll}
\max \sum_{i \in N} \sum_{k \in M} c_{ik} x_{ik} & & & \\
& & & \\
s.t. \ \sum_{i \in N} f_i x_{ik} & \leq F_k, & \text{for all } k \in M; & (1) \\
\quad\ \ \sum_{k \in M} x_{ik} & \leq 1, & \text{for all } i \in N; & (2) \\
\quad\ \ x_{ik} \in \{0,1\}, & & \text{for all } i \in N, k \in M. & (3)
\end{array}
$$

The constraints (1) in (4.1.1) are called *knapsack* constraints and the constraints (2) SOS (Special Ordered Sets) constraints. We have already observed that the problem is $\mathcal{NP}$-hard for $m = 1$. An enumeration algorithm runs in $O((m+1)^n)$ steps, thus, if n is constant in polynomial time. Some instances of the problem can be solved efficiently. If $f_i = f$ for all $i \in N$, we can divide all knapsack constraints by f and we obtain a transportation problem, that can be solved in polynomial time.

We can also make some reductions in the size of the problem instance. For example, if $f_i > F_k$ for an $i \in N$, $k \in M$, then the corresponding variable x_{ik} can be fixed to zero.

We would like to end this section with some historical notes on the MKP. The problem has been studied since the 60's, under several different names. The first reference we found in the literature goes back to Gilmore and Gomory's paper [GG66] of 1966. In the 70's some special cases of the problem were studied under the name *Loading Problem* (see, for example, Eilon and Christofides [EC72]). The name "Multiple Knapsack" for the problem here studied seems to appear for the first time in [IKo75] (though the authors still refer to the problem in the title as "loading problem") and was finally stated by Martello and Toth in their book [MT91]. Observe that the problem presented in this section is a generalization of the problem in Martello and Toth's book, who consider only the problem with uniform profit. We could not find any paper in the literature on this general formulation.

3.1.2 Exact algorithms for the MKP

In the case of the MKP, the dynamic programming approach does not yield a pseudo polynomial algorithm (as it is the case for the Single Knapsack Problem – see Section 3.2). The exact approaches of the literature are essentially based on enumeration techniques. In the remainder of this section we present some of them.

Ingargiola and Korsh (1975)

Ingargiola and Korsh present in [IKo75] an enumeration algorithm for the MKP with uniform profit, based on reduction techniques. Define the following relation R in $N \times N$:

$$\begin{array}{rl} R^0: & iR^0j \text{ if and only if} \\ & f_i \geq f_j \text{ and } c_i \leq c_j \text{ (with both equalities not holding), or} \\ & f_i = f_j, c_i = c_j \text{ and } i < j; \\ R^{h+1}: & iR^{h+1}j \text{ if and only if} \\ & iR^hj \text{ or} \\ & \text{there exists a set } S, \text{ with } sR^hj \text{ for all } s \in S \text{ and} \\ & f_j + \sum_{s \in S} f_s \leq f_i \text{ and } c_j + \sum_{s \in S} c_s \geq c_i; \\ R = & \lim_{h \longrightarrow \infty} R^h \end{array}$$

The idea of the relation is that if for the items i, j iRj and i is included in an optimal solution, then there exists also an optimal solution that includes j (observe that if $f_i = f_j$ and $c_i = c_j$ then the item with smaller index is taken). On the other hand, if j is not in any optimal solution, then no item i with iRj can be in an optimal solution.

The remarks above provide some bounds that can be used in order to restrict the search for an optimal solution. First, for each item i an upper bound on the profit can be obtained for a solution excluding i, namely, $\sum_{j \in N} c_j - c_i - \sum_{j:jRi} c_j$.

If we have a heuristic solution including item i with a better value, then the item must be in an optimal solution. On the other hand, we obtain a bound on the necessary area in the knapsacks for a solution including i (the authors call it *extended weight*), namely, $f_i + \sum_{j:iRj} f_j$. If the total knapsack capacity is less than the extended weight of an item, then it can be excluded from the optimal solution.

The algorithm suggested has two phases. The first phase explores the properties given by the relation R, and determines which items can be excluded and which items can be in an optimal solution. The second phase enumerates all possibilities in a branch and bound way. For each item $m+1$ subproblems are generated: in the first the item is excluded, in the second it is assigned to the first knapsack, and so on (see Figure 4.1).

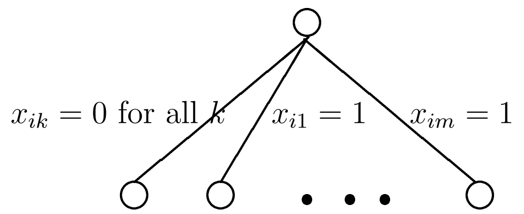


Figure 3.1: Branching strategy of Ingargiola and Korsh

The authors report on computational results on randomly generated instances with n varying from 15 to 25 and $m \leq 6$. The bottleneck of the approach is the determination of the relation R. The authors do not present the algorithm for it, and mention that the time needed for this calculation make it difficult to compare their results with previous papers of the literature.

Hung and Fisk (1978)

Hung and Fisk present in [HF78] a branch and bound approach for the MKP with uniform profit. They present two different implementations of the algorithm, differing in the calculation of the upper bound. The branching strategy is the following. They proceed in a depth first way, and each item generates $m+1$ new branches like in the procedure from Ingargiola and Korsh (see Figure 4.1), one corresponding to excluding this item of the solution and m ones for assigning the item to each knapsack.

Let us present the upper bounds used in the procedure. A first upper bound is the value of the *surrogate relaxation*. The idea is to combine the knapsacks using an array $\pi = (\pi_1, \ldots, \pi_m)$ of nonzero multipliers. Then, the value of the surrogate relaxation $S(\pi)$ is given by

$$\begin{array}{rlll} S(\pi) = & \max & \sum_{i=1}^{n} c_i y_i & \\ & s.t. & \sum_{k=1}^{m} \pi_k \sum_{i=1}^{n} f_i y_i & \leq \sum_{k=1}^{m} \pi_k F_k; \\ & & y_i \in \{0,1\} & \text{for all } i \in N. \end{array}$$

The authors use as vector π the dual multipliers of the knapsack constraints in the linear relaxation of the MKP. The resulting Single Knapsack Problem is solved exactly, yielding an upper bound for the value of the solution of the MKP. In Example 3.1.1 we show an instance in which the bound given is greater than the optimal solution of the original problem.

Example 3.1.1 Consider the instance of the MKP given by $n = 5$, $m = 2$, $f = (2, 2, 2, 2, 2)$ and $F = (5, 5)$. Let $\pi = (1, 1)$ and consider the corresponding surrogate relaxation:

$$\begin{array}{ll} \max & \sum_{i=1}^{n} c_i y_i \\ s.t. & 2y_1 + 2y_2 + 2y_3 + 2y_4 + 2y_5 \leq 10. \end{array}$$

The optimal solution of the problem above is clearly $y_i = 1$ for $i = 1, \ldots, 5$, but there is no solution of the original problem with this objective function value.

■

The second relaxation used in [HF78] is the *Lagrangean relaxation*. Given a vector $\lambda = (\lambda_1, \ldots, \lambda_n)$ of Lagrangean multipliers, the value of the Lagrangean relaxation $L(\lambda)$ is given by

$$\begin{array}{rlll} L(\lambda) = & \max & \sum_{i=1}^{n} c_i x_{ik} - \sum_{i=1}^{n} \lambda_i (\sum_{k=1}^{m} x_{ik} - 1) & \\ & s.t. & \sum_{i=1}^{n} f_i x_{ik} \leq F_k, & \text{for all } k \in M; \\ & & x_{ik} \in \{0, 1\}, & \text{for all } i \in N, k \in M. \end{array}$$

Observe that the problem described above can be divided into m independent Single Knapsack Problems, with the same weight and profit functions. The only difference between the problems is the capacity of each knapsack.

As Lagrangean multipliers the authors use the dual multipliers of the SOS constraints in the linear relaxation of the MKP. The branching variable used depends on the upper bound. In the case of the surrogate relaxation, the authors use the item with lowest index. For the Lagrangean relaxation they consider the item with maximum conflict, i.e., the item that appears most times in the solution of the knapsack problems.

The authors report on the solution of randomly generated problems with uniform distribution with $n = 50$, 100, 150 and 200, and $m = 2$, 4 and 6. They observe that the Lagrangean relaxation is superior when one attempts to solve large problems ($n \geq 60$). They restrict the maximum running time to 250 seconds. The surrogate relaxation proceeds the exact solution of a knapsack problem, which is time consuming. This is the reason for the surrogate relaxation fails in some problems.

Martello and Toth (1980)

The approach suggested by Martello and Toth in [MT80] differs from the one discussed in the last section in the choice of the Lagrangean multipliers and in the branching strategy. Moreover, the authors suggest a list of improving heuristics, used to get better bounds.

For the Lagrangean relaxation, Martello and Toth use the multipliers $\lambda = (\lambda_i) = 0$ for all $i \in N$. They observe also that the exact solution of the surrogate relaxation consumes too much time, and observe also empirically that the quality of the bound decreases when we are deep in the branch and bound tree. Therefore, the solution of the surrogate relaxation is calculated heuristically after some level in the tree.

A significant difference from this approach and the previous described is the branching strategy. In each iteration of the algorithm a current solution is considered. For the branching step, an item i is selected with $\sum_{k \in M} x_{ik} > 1$. If none exists, a feasible solution was found, and the node is fathomed. Let K_i be the set of knapsacks where this item is currently assigned. Then, $|K_i| + 1$ new nodes are considered, in the first $|K_i|$, a variable x_{ik} is set to one, and in the last one all variables x_{ik} for $k \in K_i$ are set to zero.

The authors present a careful comparison with the two previous papers on the MKP, yielding better computer times. The instances are all randomly generated with uniform distribution, $n = 25$, 50 and 100 and $m = 2$, 3 and 4.

Martello and Toth (1981)

Martello and Toth present in [MT81] a so called “bound and bound” algorithm for the MKP with uniform profit. The method is a slight modification of the branch and bound method discussed in the last sections. In this approach, it is supposed that a heuristic procedure for the MKP is given that always finds a feasible solution (if there exists one) such that no other item can be assigned without violating the knapsack constraints.

The algorithm proceeds in the following depth-first way. At each iteration a set S of variables are set, either to zero or to one. In the first iteration no variable is in S. Then a heuristic solution is obtained, and an upper bound for the subtree rooted in this node is calculated. If there is no solution for the current problem the node is fathomed and a backtrack step is performed. If the heuristic solution is better than the current best solution, then the value of the best solution is updated. We verify if the value of the heuristic solution equals the upper bound in the node. If it is the case, the node is also fathomed. Now, a branch step is performed. As branching variable x_{ik} the authors choose an item i and knapsack k such that i is assigned to k in the heuristic solution and variable x_{ik} is not set. This variable is included in S and set to one, and a new iteration is performed. Finally, in the backtrack step, the last variable set must be considered. If it was set to one, then the procedure sets it to zero and performs a new iteration. Otherwise the variable is discarded from S.

The main difference of the procedure from the other one developed by the authors is that a heuristic solution is present. This heuristic solution is used to provide a lower bound to the value of the solution, what can cut some branches of the branch and bound tree. Moreover, the choice of the branching variable is made in the "direction" given by the heuristic solution. Martello and Toth use as heuristic procedure the following idea. First, reduce the problem subtracting, for the variables in S set to one, their weight from the capacity of the corresponding knapsack. Let $N' = N \setminus S_i$, where $S_i := \{i \in N \mid x_{ik}$ is set to one in any knapsack $k \in M\}$. Then, they solve for each knapsack $k \in M$ exactly the Single Knapsack Problem in the remaining variables $N' \setminus S_k$, where $S_k := \{i \in N \mid x_{ik}$ is set to zero$\}$. The items assigned to knapsack k are included into S_i in order to avoid assigning an item two more than one knapsack, and N' is updated for a new iteration. As upper bound for the node, the surrogate and the Lagrangean relaxations are used, as discussed in the last section.

The authors compare this new approach with the one of the last section, and report better computer times. The instances are randomly generated with a uniform distribution, for the instances considered, $n = 25$, 50, 100 and 200, and $m =$ 2, 3 and 4.

3.1.3 Heuristic algorithms for the MKP

In this section we present some approximative algorithms for the Multiple Knapsack Problem with uniform profit. The problem is proven to be $\mathcal{NP}$-*complete in strong sense* (cf. [MT91]), since the 3-Partition Problem can be polynomially transformed to it. Then, no *fully polynomial approximation scheme* can exist for the problem, unless $\mathcal{P} = \mathcal{NP}$. No polynomial approximation scheme is known for the MKP in the general case (Frieze and Clarke present in [FC84] a polynomial approximation scheme for the GZOP when m is fixed - see Section 4.3). In the remainder of this section we present some heuristic algorithms for the Multiple Knapsack Problem with uniform profit.

Fisk and Hung (1979)

Fisk and Hung present in [FH79] a heuristic procedure based on the solution of the surrogate relaxation of the MKP with uniform profit.

Consider the surrogate relaxation $S(\pi)$ of the MKP. Let T be the set of items included in the optimal solution of the Single Knapsack Problem corresponding to the relaxation. The heuristic looks for a feasible assignment of the items in T to the knapsacks in the original problem. If an assignment is found in which all items of T are assigned, then it is clearly an optimal solution.

In order to find a feasible assignment of T, the authors suggest an "exchange routine". In the beginning the items are assigned to the knapsacks in a random way, until there is no sufficient space to assign an item. In this case, a "rearrangement" of the items in the knapsacks is performed, where exchanges are made in order to increase the maximum available capacity in some knapsack. The authors suggest first a one-to-one exchange, and, if no such exchange is possible, one-to-two and two-to-one exchanges. The procedure looks for such exchanges for each pair of knapsacks. It goes on in this way until sufficient space for the item is achieved. If after trying all exchanges, no sufficient space is achieved, then the item considered is excluded from the current solution.

The authors report on solving randomly generated problems with up to 1000 items and 6 knapsacks. In only one case from 100 problem instances the exchange procedure did not find a solution with the same value of the surrogate relaxation. For all other 99 instances the heuristic solution had the value of the surrogate relaxation, and, therefore, was provably optimal.

Martello and Toth (1981)

Martello and Toth present in [MT81a] a series of heuristic procedures to the MKP with uniform profit. The heuristic procedure suggested in [FH79] has the disadvantage that in each iteration a knapsack problem must be exactly solved. The idea presented in the paper is to compare the behavior of simple heuristics with the one suggested in [FH79].

The authors suggest very simple heuristic ideas to find an initial solution and improve a solution given. For an initial solution they suggest three procedures:

- MK1: for each $k \in M$, a greedy solution of the single knapsack problem is calculated. The items in the solution are excluded from the set of the items to avoid an item being assigned to more than one knapsack;
- MK2: the idea is to assign items to knapsacks in a cyclic way. Items that cannot be assigned to any knapsack are excluded;
- MK3: first a greedy solution is given for all knapsacks $k \in M$. In this case, items are allowed to be assigned to more than one knapsack. Then, the conflicts are treated. To this end, the item is excluded from all knapsacks and the greedy solution is recalculated for the items with index greater than the index of the conflict item. The item in conflict is assigned to the knapsack where the difference of the two greedy solutions is maximum (or the loss of profit excluding the item would be the greatest). The procedure is iterated until no conflict is present.

They compare the heuristics empirically in a set of problem instances, and the heuristic MK3 has the best bounds but the worst running times.

As improving heuristics two ideas are presented. In the first one, two exchange between items assigned to different knapsacks are checked. The exchange is performed if it allows an item excluded in the current solution to be assigned to a knapsack. The second procedure works in the following way. In each iteration an item assigned to a knapsack is substituted by one or more items excluded in the current solution.

The authors report on solving large instances of randomly generated problems with correlated and uncorrelated item weights and knapsack capacities. In the instances considered $n \leq 1000$ and $m = 10$, 20, 50 and 100. The average error reported is always under 7%. Moreover, since the procedures do not consume too much time or memory, better results are provided in comparison with the approach from Fisk and Hung [FH79].

3.2 The Generalized Assignment Problem

The GAP can be defined as follows.

> Given a set N of jobs, a set M of machines with resource availability F_k, $k \in M$, an array of resource demands f_{ik}, $i \in N$, $k \in M$ corresponding to assigning job i to machine k and a cost function c associated with assigning the job i to machine k. We want to find an assignment of all jobs to the machines such that resource availability constraints are satisfied and the assignment has minimum cost.

In the literature there are some applications of this problem. Fisher and Jaikumar in [FJ81] present a formulation for the Vehicle Routing Problem using the GAP, where the machines correspond to trucks and the jobs to items to be delivered. A special case of the problem is studied by de Maio and Roveda [MR71], where the jobs have uniform demands for all machines $f_{ik} = f_i$ for all $k \in M$, $i \in N$. They report on a practical application concerning a transportation problem, where items have to be distributed at warehouses in a way that the demands of the shops are satisfied and minimizing the delivery costs. Fisher, Jaikumar and van Wassenhove refer in [FJW86] to some other applications like assigning software development tasks to programmers, assigning jobs to computers in computer networks and designing communication networks with node capacity constraints.

In the next section we present a formulation for this problem using 0-1 variables. Some results of the literature for the GAP are listed in Section 4.2.2. Polyhedral approaches to GAP and special cases are presented in Section 4.2.3.

3.2.1 A formulation of GAP using 0-1 variables

The GAP is the following problem.

> **Problem:** GAP
> **Instance:** Finite sets N and M, $f \in \mathbb{N}^{N \times M}$, $F \in \mathbb{N}^M$ and $c \in \mathbb{N}^{N \times M}$.
> **Question:** Find a partition $\mathcal{B} = (B_1, \ldots, B_j)$ of N, such that $j \leq |M|$, $\sum_{i \in B_k} f_i \leq F_k$, for all $k \in M$ and $\sum_{k \in M} \sum_{i \in B_k} c_{ik}$ is minimum.

We introduce 0-1 variables in order to model the problem. Let $x_{ik} \in \{0, 1\}^{N \times M}$ with the interpretation $x_{ik} = 1$, when job i is assigned to machine k and 0, otherwise. An integer programming formulation of GAP is given via

$$
\begin{array}{llllll}
 & \min \sum_{i\in N} \sum_{k\in M} c_{ik} x_{ik} & & & & \\
 & & & & & \\
 & s.t. \ \sum_{i\in N} f_{ik} x_{ik} & \leq F_k, & \text{for all } k \in M; & & (1) \\
(4.2.1) & \qquad \sum_{k\in M} x_{ik} & = 1, & \text{for all } i \in N; & & (2) \\
 & \qquad x_{ik} \in \{0,1\}, & & \text{for all } i \in N, k \in M. & & (3)
\end{array}
$$

The problem is proven to be $\mathcal{NP}$-hard [MT91], even for $m = 2$. The feasibility problem is also $\mathcal{NP}$-hard, since Partition can be reduced to GAP in polynomial time [MT91].

3.2.2 A brief literature survey on the GAP

In this section we list some results of the literature on GAP in the general form and some special cases. It is not our objective to provide a complete survey, but give the reader a feeling on the methods used to solve the problem.

In [MR71], de Maio and Roveda consider the special case of the GAP with uniform demands for all machines, $f_{ik} = f_i$ for all $k \in M$, $i \in N$. They suggest an enumeration algorithm in which all n-tuples of M are considered in increasing order of cost. The first feasible tuple generated is, thus, also optimal.

Srinivasan and Thompson study in [ST73] the same problem. They suggest a branch and bound algorithm. First, they transform the problem given above, taking $y_{ik} = f_i x_{ik}$. Then, the problem can be rewritten as the following transportation problem

$$
\begin{array}{llll}
\max \sum_{i\in N} \sum_{k\in M} \frac{c_{ik}}{f_i} y_{ik} & & & \\
 & & & \\
s.t. \ \sum_{i\in N} y_{ik} & \leq F_k, & \text{for all } k \in M; & \\
\qquad \sum_{k\in M} y_{ik} & = f_i, & \text{for all } i \in N; & \\
\qquad y_{ik} \in \{0, f_i\} & & \text{for all } i \in N, k \in M. &
\end{array}
$$

The linear relaxation of the problem above can be efficiently calculated. Then, in each node of the branch and bound tree, the optimal solution of the LP relaxation is calculated. If all y_{ik} variables are in $\{0, f_i\}$ then a feasible solution is found and the node is fathomed. Otherwise, a "fractional" variable is chosen and two subproblems are generated, in one with this variable fixed to zero and in the other one to f_i.

Some branch and bound strategies have been developed for the GAP. Ross and Soland use in [RS75] as bound a Lagrangean relaxation on the constraints (2) of (4.2.1), where the Lagrangean multipliers $\lambda = (\lambda_i)$ are chosen as the maximal c_{ik} for all $k \in M$ such that the job i can be assigned to this machine ($f_{ik} \leq F_k$). The branching variable is selected as the one with maximal "conflict" in the resulting independent knapsack problems.

Martello and Toth present in [MT81b] a better bound to the problem, taking the minimum from the bound analyzed by Ross and Soland and the value of the Lagrangean relaxation using **0** as Lagrangean multipliers. The strategy used to select the branching variable is similar to the one used by Ross and Soland.

Fisher, Jaikumar and van Wassenhove develop in [FJW86] a technique to obtain better Lagrangean multipliers called *multiplier adjustment method*. The idea is, at each iteration, find jobs violating the relaxed constraints and increase in a suitable way, the corresponding penalty (or Lagrangean) multiplier. The multipliers found by the procedure are used to generate bounds for a branch and bound method.

Crema study in [Cr90] a generalization of the GAP, where different resources are needed by the jobs. In this case, the jobs demands become a three dimensional vector and the machines capacities two dimensional. The author compares the value of different relaxations to the problem.

3.2.3 Polyhedral investigations

In this section we survey a polyhedral investigation to a polytope associated with GAP, studied by Gottlieb and Rao [GR90]. Further, a special case of the problem has been also studied from a polyhedral point of view by Abdel Hamid [AH94].

Gottlieb and Rao (1990)

In [GR90] and [GR90a] Gottlieb and Rao present a polyhedral investigation of a polytope associated with the GAP. First, they transform the problem into an equivalent one, where penalty parameters are associated with the equalities (2) in (4.2.1). This transformation is made in order to obtain a full dimensional polytope. So, the polytope is then defined as follows

$$GAP(N,M,f,F) := conv\{x \in \mathbb{R}^{mn} \mid \begin{array}{lll} \sum_{i\in N} f_{ik}x_{ik} \leq F_k, & \text{for all} & k \in M; \\ \sum_{k\in M} x_{ik} \leq 1, & \text{for all} & i \in N; \\ x_{ik} \in \{0,1\}, & \text{for all} & i \in N, \\ & & k \in M\}. \end{array}$$

The authors usually consider polytopes associated with subproblems of the original problem. Let $H \subseteq N$, $W \subseteq M$ and $Y \subseteq H \times W$, and define $GAP^Y(H,W,f,F)$ as the polytope in $\mathbb{R}^Y$ where only the pairs in Y are considered as feasible. In order to simplify the presentation, the authors represent the problem as a bipartite graph $G = (V,E)$ where $V = H \cup W$ and $e = ik \in E$ if $(i,k) \in Y$.

Now, we describe some of the contributions from Gottlieb and Rao. In [GR90] they present some classes of valid inequalities for the polytope. In this summary we will show only the first one, the so called *cycle inequality*. The interested reader can find other classes in [GR90] and [GR90a].

Theorem 3.2.1 (Cycle Inequalities) *Suppose that an instance (N, M, f, F) of the GAP is given. Let $H \subseteq N$, $W \subseteq M$ and $Y \subseteq H \times W$ such that the graph defined as above has exactly one cycle C, and this cycle contains all nodes of W. Suppose $W = P \cup Q$, $P \cap Q = \emptyset$, $|Q| \geq 1$ and $|P| = 2p + 1$. Denote for $k \in W$, $N_k := \{i \in N \mid (i, k) \in Y\}$. Further, let $i_k, j_k \in H$ be the nodes adjacents to k in the cycle C ($f_{i_k} \leq f_{j_k}$). If,*

(i) for all $k \in Q$, N_k is a cover for k,

(ii) for all $k \in P$, $N_k \setminus \{j_k\}$ is a cover for k,

then

$$\sum_{(i,k)\in Y} x_{ik} \leq |H| - p - 1$$

is valid for $GAP^Y(H, W, f, F)$.

Proof. Observe that

$$\begin{aligned} 2\textstyle\sum_{(i,k)\in Y} x_{ik} &= \textstyle\sum_{k\in Q}\sum_{i\in N_k} x_{ik} + \sum_{k\in P}\sum_{i\in N_k\setminus\{i_k\}} x_{ik} + \\ &\quad \textstyle\sum_{k\in P}\sum_{i\in N_k\setminus\{j_k\}} x_{ik} + \sum_{ik\in C} x_{ik} + \sum_{k\in Q}\sum_{i\in N_k\setminus\{i_k,j_k\}} x_{ik}, \end{aligned}$$

due to (i), (ii) and SOS constraints,

$$\begin{aligned} &\leq \textstyle\sum_{k\in Q}(|N_k| - 1) + 2\sum_{k\in P}(|N_k| - 2) + |W| + \sum_{k\in Q}(|N_k| - 2), \\ &= \textstyle 2\sum_{k\in Q}|N_k| + 2\sum_{k\in P}|N_k| - 3|Q| - 4|P| + |W|, \\ &= 2|Y| - 3|Q| - 4|P| + |W|, \end{aligned}$$

since $|Y| = |H| + |W|$ and $|W| = |P| + |Q|$,

$$\begin{aligned} &= 2|H| - |P|, \\ &= 2|H| - 2p - 1. \end{aligned}$$

Therefore, $\sum_{(i,k)\in Y} x_{ik} \leq |H| - p - \frac{1}{2}$, or, since the variables are integers,

$$\sum_{(i,k)\in Y} x_{ik} \leq |H| - p - 1.$$

■

Gottlieb and Rao provide also in [GR90] some necessary conditions that an inequality must satisfy in order to define a nontrivial joint facet of $GAP(N, M, f, F)$. We present in the following one of their results.

Theorem 3.2.2 *Let $ax \leq \alpha$ be a nontrivial joint facet defining inequality of $GAP^Y(H, W, f, F)$. Then, for all $k \in W$,*

$$|N_k \cap N^{(k)}| \geq 2,$$

where, $N^{(k)} := \cup_{l\in W\setminus\{k\}} N_l$.

Proof. Let $k \in W$ and suppose that $|N_k \cap N^{(k)}| < 2$. Let us analyze the two cases. First, let $|N_k \cap N^{(k)}| = 0$. Then, there exists α^1 and α^2, $\alpha = \alpha^1 + \alpha^2$, such that for all $x \in EQ(ax \leq \alpha)$, $\sum_{i \in N_k} a_{ik} x_{ik} = \alpha^1$ and $\sum_{l \in W \setminus \{k\}} \sum_{i \in N_l} a_{il} x_{il} = \alpha^2$. Then, $\sum_{i \in N_k} a_{ik} x_{ik} \leq \alpha^1$ and $\sum_{l \in W \setminus \{k\}} \sum_{i \in N_l} a_{il} x_{il} \leq \alpha^2$ are valid inequalities for $GAP^Y(H, W, f, F)$, contradicting the assumption that $ax \leq \alpha$ is a facet defining inequality.

Suppose, now, that $N_k \cap N^{(k)} = \{i\}$. Let $x^1 \in EQ(ax \leq b)$ with $x^1_{il} = 0$ for all $l \in W$ (there is one, otherwise $EQ(ax \leq b) \subseteq EQ(\sum_{l \in W} x_{il} \leq 1)$). Define,

$$\alpha^1 := \sum_{j \in N_k} a_{jk} x^1_{jk} \text{ and } \alpha^2 := \sum_{l \in W \setminus \{k\}} \sum_{j \in N_l} a_{jl} x^1_{jl}.$$

Consider another solution $x^2 \in EQ(ax \leq \alpha)$ with $x^2_{ik} = 1$ (there is one, otherwise $EQ(ax \leq b) \subseteq \{x \in \mathbb{R}^Y \mid \sum_{l \in W} x_{il} = 0\}$). In this case, $\sum_{j \in N_k} a_{jk} x^2_{jk}$ must also be less than or equal α^1, otherwise we could combine x^1 and x^2 giving a point violating the inequality. Then, $\sum_{j \in N_k} a_{jk} x_{jk} \leq \alpha^1$ is a valid inequality for $GAP^Y(H, W, f, F)$. The same argument holds for $\sum_{l \in W \setminus \{k\}} \sum_{j \in N_l} a_{jl} x_{jl} \leq \alpha^2$. But, it contradicts the fact that $ax \leq \alpha$ defines a facet. ■

Moreover, they could prove that nontrivial facet defining individual inequalities are also facet defining inequalities of $GAP(N, M, f, F)$ (in Section 5.2 we present a proof for this fact for the Multiple Knapsack Polytope). Another interesting result due to Gottlieb and Rao is that the fractional solution of the LP Relaxation of problem (4.2.1) violates a nontrivial individual inequality.

3.3 General Zero One Programming

In this section we present the General Zero One Programming Problem. This problem is included in this survey about related problems to the MKP because it is usually called "Multiple Knapsack" in the literature. The problem can be defined as follows.

> **Problem:** GZOP
> **Instance:** $n, m \in \mathbb{N}$, a (m, n)-matrix A with $a_{ij} \in \mathbb{N}$ for all $i \in M$, $b \in \mathbb{N}^m$ and $c \in \mathbb{N}^n$.
> **Question:** Find $\max cx$ such that $Ax \leq b$ and $x \in \{0, 1\}^n$.

The problem is $\mathcal{NP}$-hard, even if all elements of A, b and c are required to belong to $\{0, 1\}$ (cf. [GJ79]). If m is fixed, Frieze and Clarke [FC84] provide a polynomial approximation scheme for the problem. Korte and Schrader prove in [KS80] that no fully polynomial approximation scheme can exist for the problem, even in the case of fixed m, unless $\mathcal{P} = \mathcal{NP}$. Let N be the set of the columns and M the set of rows. The idea of Frieze's approach is similar to Sahni's procedure to the Knapsack Problem (see Section 3.3.1). Consider all subsets $S \subseteq N$, $|S| \leq k$, and

$\sum_{j \in S} a_{ij} \leq b_i$ for all $i \in M$. The variables in S are supposed to be in this solution and are set to one. Now, solve the linear relaxation of the "reduced problem", i.e., the problem obtained by considering only the variables of $N \setminus S$ and the reduced capacities b. Like in Sahni's algorithm, the idea is to solve exactly instances with less than or equal k elements, and complete, in a greedy like way, the solution with the others. They prove that the solution given by this procedure x' is a ϵ-approximation of the optimal solution, when $k = \min(n, \lceil \frac{m(1-\epsilon)}{\epsilon} \rceil)$.

The GZOP has been attacked by Crowder, Johnson and Padberg [CJP83], and Hoffman and Padberg [HP91] where algorithms based on the LP relaxation of the problem, with cutting planes from individual knapsack inequalities are used to solve large instances.

Crowder, Johnson and Padberg consider the GZOP in the case that the matrix A is sparse and has no apparent structure. The idea used to solve this problem is to consider the linear relaxation of it (that is, substitute the constraints $x \in \{0,1\}^n$ by $0 \leq x_i \leq 1$ for all $i \in N$) and try to obtain better bounds using valid inequalities of the polytope

$$GZOP(m, n, A, b) := conv\{x \in \mathbb{R}^n \mid Ax \leq b, x_i \in \{0,1\} \text{ for all } i \in N\}.$$

They observe that the facet defining inequalities of the polytopes associated with each individual knapsack constraint are valid for $GZOP(m, n, A, b)$. In [CJP83] heuristic procedures for generation of violated Minimal Cover and (1,k)-Configuration Inequalities are suggested (we present these procedures in the Chapter 6). With the inequalities generated by these routines, the authors obtain an improvement of up to 187% on the value of the linear relaxation. No more inequality is added to the LP if the improvement obtained is too small. All problems considered are solved to optimality, using a final branch and bound phase. The instances considered have up to 2756 variables, and are solved in reasonable computer times (less than one hour).

Hoffman and Padberg improve on the results of [HP91] using different strategies for generating violated inequalities from individual knapsack constraints. They use, for example, the idea of complementing as described in Section 3.4. Moreover, several new tools, coming from the Branch and Cut method, like eliminating redundant inequalities, LP based heuristics, and so on, (see Chapter 6 for more details on the Branch and Cut approach) are presented in the new approach. Instances with up to 6000 variables are considered, and solved to optimality.

Chapter 4

A Polyhedral Investigation of the Multiple Knapsack Problem

In this chapter we investigate MKP from a polyhedral point of view. The idea is to associate with every feasible solution of the problem a vertex of the Multiple Knapsack Polytope, and study its properties. In particular, our main aim is to provide classes of valid and facet defining inequalities for this polytope. These inequalities are of fundamental importance in the method we use to solve the problem (presented in Chapter 6). One can classify the valid or facet defining inequalities for the Multiple Knapsack Polytope into two types:

- inequalities whose all nonzero coefficients belong to the same knapsack. These are called *individual* inequalities;
- inequalities that combine nonzero coefficients of at least two different knapsacks. These are called *joint* inequalities.

In Chapter 3 we describe some valid and facet defining inequalities from the literature for the Single Knapsack Polytope. These inequalities coincide, as we show in this chapter, with the individual inequalities of the Multiple Knapsack Polytope. In this chapter we present also several new classes of valid and facet defining joint inequalities for the Multiple Knapsack Polytope. Polyhedral investigations to related polytopes are described in Chapter 4.

This chapter is organized as follows. In Section 5.1 we define the Multiple Knapsack Polytope and show some results about its dimension. We present in Section 5.2 some initial results about facet defining inequalities. We show, for example, that all facet defining inequalities of polytopes associated with single knapsack constraints define also individual facets of the Multiple Knapsack Polytope. Sections 5.3 to 5.8 are dedicated to joint inequalities. We present, in these sections, new classes of valid and facet defining joint inequalities of the Multiple Knapsack Polytope. Finally, in Section 5.9 a procedure is presented to extend facet defining inequalities of the polytope to higher dimensions. Some of the results in this chapter are presented in [FMW93].

4.1 The Multiple Knapsack Polytope

In Chapter 4 we give a formulation of the MKP using 0-1 variables. Let us repeat this formulation. Given an instance (N, M, f, F) of the MKP, consider the variables $x_{ik} \in \{0,1\}^{mn}$ with the interpretation $x_{ik} = 1$, when item i is assigned to knapsack k and $x_{ik} = 0$, otherwise. An integer programming formulation of the MKP is given via

$$\max \textstyle\sum_{i \in N} \sum_{k \in M} c_{ik} x_{ik}$$

$$\begin{array}{llll} \sum_{i \in N} f_i x_{ik} & \leq F_k & \text{for all } k \in M & (1) \\ \sum_{k \in M} x_{ik} & \leq 1 & \text{for all } i \in N & (2) \\ x_{ik} \in \{0,1\} & & \text{for all } i \in N, k \in M & (3) \end{array}$$

The constraints (1) are called *knapsack* constraints and the constraints (2) SOS (Special Ordered Sets) constraints.

It is usual, in this approach, to investigate subinstances of the original problem. The inequalities found for these subinstances are, then, extended to the original problem using the so called Lifting Procedure.

Let $A_i \subseteq N$ and $B_i \subseteq M$ for $i = 1, \ldots, t$. Define $T := \bigcup_{i=1}^{t} A_i \times B_i$. T can be interpreted as the set of possible assignments, i.e. some item $i \in N$ can be assigned only to the knapsacks k such that $(i, k) \in T$. The Multiple Knapsack Polytope is defined as follows.

$$MKP(T, f, F) := conv\{x \in \mathbb{R}^T \mid \begin{array}{ll} \sum_{i:(i,k) \in T} f_i x_{ik} \leq F_k, & k \in \bigcup_{l=1}^{t} B_l, \\ \sum_{k:(i,k) \in T} x_{ik} \leq 1, & i \in \bigcup_{j=1}^{t} A_j, \\ x_{ik} \in \{0,1\}, & (i,k) \in T. \end{array}$$

In this notation the polytope corresponding to the original problem coincides with $MKP(N \times M, f, F)$. For ease of notation we use f and F instead of $f|_{\bigcup_{j=1}^{t} A_j}$ and $F|_{\bigcup_{l=1}^{t} B_l}$ in the definition of the polytope. We will also use $MKP(N, M, f, F)$ as a synonym for $MKP(N \times M, f, F)$.

Lemma 4.1.1 *$MKP(T, f, F)$ is full dimensional if and only if $f_i \leq F_k$ for all $(i, k) \in T$.*

Proof. If there is a pair $(i, k) \in T$ with $f_i > F_k$, then $MKP(T, f, F) \subset \{x \in \mathbb{R}^T \mid x_{ik} = 0\}$, and, therefore, the polytope is not full dimensional. Conversely, it suffices to observe that $\mathbf{0}$ and the unit vectors e_{ik} are in $MKP(T, f, F)$ for all $(i, k) \in T$, and are affinely independent. ■

In particular, $MKP(N \times M, f, F)$ is full dimensional if and only if $f_i \leq F_k$ for all $i \in N$, $k \in M$. We assume throughout this chapter that it is the case.

4.2 Some initial results on facet defining inequalities

In this section we show some initial results on facet defining inequalities of the Multiple Knapsack Polytope. In the next lemmas we present two classes of facet defining inequalities of the $MKP(T,f,F)$. We refer to them as *trivial* inequalities.

Lemma 4.2.1 *The inequality*

$$x_{ik} \geq 0$$

defines a facet of $MKP(T,f,F)$ for all $(i,k) \in T$.

Proof. The inequality is clearly valid. Let $(i',k') \in T$ and suppose that there exists a facet defining inequality $bx \leq \beta$ such that $EQ(x_{i'k'} \geq 0) \subseteq EQ(bx \leq \beta)$. Observe now that $\mathbf{0}$ and e_{ik} are in $EQ(x_{ik} \geq 0)$ for all $(i,k) \in T \setminus \{(i',k')\}$. Then, $b_{ik} = 0$ for all $(i,k) \in T \setminus \{(i',k')\}$. Thus, $bx \leq \beta$ is a scalar multiple of the inequality, and, therefore, it defines a facet. ■

Observe that these are the only facet defining inequalities of the $MKP(T,f,F)$ with negative coefficients (in the form $ax \leq \alpha$). Suppose that there exists another facet defining inequality $ax \leq \alpha$ with $a_{ik} < 0$. Then, there must exists a vector $x' \in EQ(ax \leq \alpha)$ with $x'_{ik} = 1$ (otherwise $EQ(ax \leq \alpha) \subseteq \{x \in \mathbb{R}^T \mid x_{ik} = 0\}$, contradicting the assumption that $ax \leq \alpha$ is facet defining). But then, $x' - e_{ik}$ is also a feasible point of the polytope and $a(x'-e_{ik}) = ax' - a_{ik} > \alpha$, a contradiction.

Lemma 4.2.2 *Given $i \in \bigcup_{j=1}^{t} A_j$, define $K_i := \{k \in M \mid (i,k) \in T\}$. The inequality*

$$\sum_{k \in K_i} x_{ik} \leq 1$$

defines a facet of $MKP(T,f,F)$ if and only if $|K_i| \geq 2$ or $K_i = \{k\}$ and $f_i + f_j \leq F_k$ for all $j \in \{j' \in N \setminus \{i\} \mid (j',k) \in T\}$.

Proof. First let us prove that the conditions are sufficient. The inequality is clearly valid. Let $i \in \bigcup_{j=1}^{t} A_j$ and suppose that there exists a facet defining inequality $bx \leq \beta$ such that $EQ(\sum_{k \in K_i} x_{ik} \leq 1) \subseteq EQ(bx \leq \beta)$. Note that e_{ik} is an element of $EQ(\sum_{k \in K_i} x_{ik} \leq 1)$ for all $k \in K_i$. Then, $b_{ik} = b_{ik'}$ for all $k, k' \in K_i$. Now, due to the condition in the theorem, we have two cases to analyze.

- $|K_i| \geq 2$.
  Let $i' \in \cup_{j=1}^{t} A_j \setminus \{i\}$ and $k' \in K_{i'}$. Let also $k \in K_i \setminus \{k'\}$ (there exists some, since $|K_i| \geq 2$). Then, $e_{ik} + e_{i'k'}$ is an element of $EQ(\sum_{k \in K_i} x_{ik} \leq 1)$, and therefore, $b_{i'k'} = 0$ for all $i' \in \cup_{j=1}^{t} A_j \setminus \{i\}$ and $k' \in K_{i'}$.

- $K_i = \{k\}$ and $f_i + f_j \leq F_k$ for all $j \in \{j' \in N \setminus \{i\} \mid (j', k) \in T\}$.
  Let $i' \in \cup_{j=1}^{t} A_j \setminus \{i\}$ and $k' \in K_{i'}$. If $k' \neq k$, $e_{ik} + e_{i'k'}$ is an element of $EQ(\sum_{k \in K_i} x_{ik} \leq 1)$, and therefore, $b_{i'k'} = 0$. In the other case, since $f_i + f_{i'} \leq F_k$, $e_{ik} + e_{i'k}$ is also in $EQ(\sum_{k \in K_i} x_{ik} \leq 1)$, and therefore, $b_{i'k} = 0$.

Summing up all cases above, $bx \leq \beta$ is a scalar multiple of the inequality, and therefore, it defines a facet of $MKP(T, f, F)$.

Now we prove that the conditions are also necessary. Suppose that the conditions are not satisfied, i.e., $K_i = \{k\}$ and there exists a $j \in \{j' \in N \setminus \{i\} \mid (j', k) \in T\}$ such that $f_i + f_j > F_k$. In this case, the inequality $x_{ik} + x_{jk} \leq 1$ is clearly valid. Then, the inequality $\sum_{k \in K_i} x_{ik} \leq 1$ is the sum of the inequality above and $x_{jk} \geq 0$, and therefore cannot be facet defining. ■

In the next Lemma we prove that all facet defining inequalities of the polytopes associated with the knapsack constraints define also facets of the $MKP(N \times M, f, F)$. This result (in a more general version) is due to Gottlieb and Rao [GR90].

Lemma 4.2.3 *Let $k \in M$ and consider $SKP(N, f, F_k)$, the Single Knapsack Polytope associated with the k-th knapsack constraint. Suppose $ay \leq \alpha$ is a nontrivial facet defining inequality of $SKP(N, f, F_k)$. Then, $a'x \leq \alpha$ defines a facet of $MKP(N \times M, f, F)$, where $a' \in \mathbb{N}^{mn}$ and*

$$a'_{ij} := \begin{cases} a_i, & \text{if } j = k, \\ 0, & \text{otherwise.} \end{cases}$$

Proof. Since the inequality $ay \leq \alpha$ defines a facet of $SKP(N, f, F_k)$, the inequality $a'x \leq \alpha$ is clearly valid for $MKP(N, f, F)$. Now suppose that there exists a facet defining inequality $bx \leq \beta$ of $MKP(N, M, f, F)$ such that $EQ(MKP(N, M, f, F), a'x \leq \alpha) \subseteq EQ(MKP(N, M, f, F), bx \leq \beta)$. Let $i \in N$ and $l' \in M \setminus \{k\}$. Since $ay \leq \alpha$ is a nontrivial facet of $SKP(N, f, F_k)$, there exists a vector $y' \in EQ(SKP(N, f, F_k), ay \leq \alpha)$ such that $y'_i = 0$ (otherwise, $EQ(SKP(N, f, F_k), ay \leq \alpha) \subseteq \{y \in \mathbb{R}^n \mid y_i = 1\}$, and $y_i \leq 1$ is a trivial valid inequality). Then, the vector $x' \in \mathbb{R}^{mn}$ defined by:

$$x'_{jl} := \begin{cases} y'_j, & \text{for all } j \in N, l = k, \\ 0, & \text{otherwise} \end{cases}$$

is an element of $EQ(MKP(N \times M, f, F), a'x \leq \alpha)$. Moreover, the vector $x' + e_{il} \in \mathbb{R}^{mn}$ is also an element of $EQ(MKP(N \times M, f, F), a'x \leq \alpha)$ for all $l \in M \setminus \{k\}$. Then, $bx' = b(x' + e_{il})$, and, thus, $b_{il} = 0$ for all $l \in M \setminus \{k\}$. The same argument holds for all $i \in N$. The inequality $bx \leq b$ is then an individual inequality. Since $ay \leq \alpha$ defines a facet of $SKP(N, f, F_k)$, $bx \leq b$ is a scalar multiple of $a'x \leq \alpha$, and, therefore, $a'x \leq \alpha$ defines a facet of $MKP(N \times M, f, F)$. ■

The result above provides several facet defining inequalities of the polytope we study. As described in the Chapter 3, much work has been spent on investigations of the Single Knapsack Polytope, and many valid and facet defining inequalities for it are known. In the remainder of this chapter we investigate joint inequalities, i.e., inequalities that combine the coefficients of at least two knapsacks.

The individual inequalities are not sufficient to provide a complete knowledge about $MKP(N \times M, f, F)$. In Example 4.2.4 we show an instance of the MKP and a fractional solution that cannot be cut off by any individual inequality.

Example 4.2.4 Consider the following instance of the Multiple Knapsack Problem. Let $m = 3$, $n = 3$,

$$\begin{aligned} 3x_{1,1} + 4x_{2,1} + 5x_{3,1} &\leq 5, \\ 3x_{1,2} + 4x_{2,2} + 5x_{3,2} &\leq 7, \\ 3x_{1,3} + 4x_{2,3} + 5x_{3,3} &\leq 5. \end{aligned}$$

The complete description of the polytope $MKP(N, M, f, F)$ (obtained by an enumeration algorithm) is the following:

$$\begin{aligned} x_{ik} &\geq 0 \quad \text{for all } i \in N, k \in M, \\ \textstyle\sum_{k \in M} x_{ik} &\leq 1 \quad \text{for all } i \in N, \\ x_{1,1} + x_{2,1} + x_{3,1} &\leq 1, \\ x_{1,3} + x_{2,3} + x_{3,3} &\leq 1, \\ x_{2,2} + x_{3,2} &\leq 1, \\ x_{1,2} + x_{3,2} &\leq 1, \\ x_{1,2} + x_{2,2} + x_{3,2} + x_{1,3} + x_{2,3} &\leq 2, \\ x_{1,1} + x_{2,1} + x_{1,2} + x_{2,2} + x_{3,2} &\leq 2. \end{aligned}$$

Observe now that $x' \in \mathbb{R}^{3 \times 3}$ given by:

$$x' := \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} \end{pmatrix}$$

cannot be cut off by any individual inequality. Moreover, x' violates the inequality $x_{1,1} + x_{2,1} + x_{1,2} + x_{2,2} + x_{3,2} \leq 2$. ■

4.3 Extended Cover Inequalities

In this section we present a class of facet defining inequalities for $MKP(N \times M, f, F)$: the Extended Cover Inequalities. We discuss in the following, conditions on the inequality to be valid and facet defining. Afterwards, a generalization is given. This class of inequalities can be obtained as an application of the procedure presented in Section 5.9. Another interesting observation is that the conditions required for an inequality be facet defining are local in each knapsack. Then, it is not difficult to see that the same ideas apply to the polytope associated with the GAP.

Suppose we are given an instance (N, M, f, F) of the MKP. Let $k, l \in M$, $S \subseteq N$ a cover for k and $T \subseteq N \setminus S$ such that $\sum_{i \in T} f_i \leq F_l$ and $T \cup \{s\}$ is a cover for knapsack l for all $s \in S$. Then, we call the inequality

$$\sum_{i \in S} x_{ik} + \sum_{i \in S \cup T} x_{il} \leq |S| + |T| - 1$$

Extended Cover Inequality corresponding to S, T, k and l.

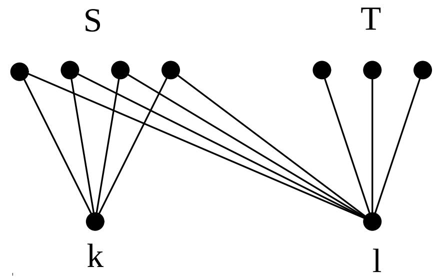


Figure 4.1: Extended Cover Inequality.

Lemma 4.3.1 *The Extended Cover Inequality is valid for $MKP(N \times M,\ f, F)$.*

Proof. Suppose it is not the case, and let $x' \in MKP(N \times M, f, F)$ that violates the inequality. Then, due to the SOS constraints,

$$\sum_{i \in S} x'_{ik} + \sum_{i \in S \cup T} x'_{il} = |S| + |T|.$$

Since $T \cup \{s\}$ is a cover for all $s \in S$, $\sum_{i \in S} x'_{il} = 0$. But, S is a cover for k, and therefore, $\sum_{i \in S} x'_{ik} \leq |S| - 1$, a contradiction. ■

In the following theorem we present necessary and sufficient conditions for the Extended Cover Inequality to define a facet.

Theorem 4.3.2 *Suppose we are given an instance (N, M, f, F) of the MKP. Let $k, l \in M$, $S \subseteq N$ a cover for k and $T \subseteq N \setminus S$ such that $\sum_{i \in T} f_i \leq F_l$ and $T \cup \{s\}$ is a cover for knapsack l for all $s \in S$. Define $A := (S \times M) \cup (T \times M \setminus \{k\})$. The Extended Cover Inequality corresponding to S, T, k and l defines a facet of $MKP(A, f, F)$ if and only if*

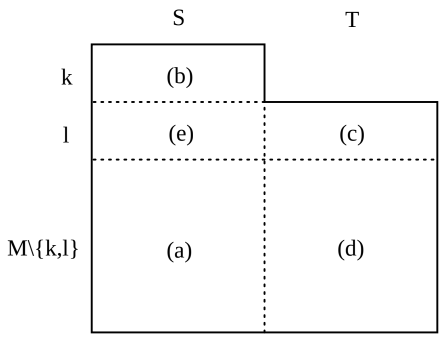
i) S is a minimal cover for k;

ii) *There exists a* $s_1 \in S$ *such that* $T \cup \{s_1\}$ *is a minimal cover for* l*;*

iii) *For all* $s \in S$*, there exists a* $t \in T$ *such that* $\sum_{i \in T \setminus \{t\}} f_i + f_s \leq F_l$*;*

iv) *There exist* $t_0 \in T$ *and* $S_0 \subseteq S$*,* $|S_0| = 2$ *such that* $\sum_{i \in T \setminus \{t_0\}} f_i + \sum_{i \in S_0} f_i \leq F_l$*.*

Proof. The inequality is valid due to Theorem 5.3.1. Define $a \in \mathbb{R}^{mn}$ by $a := \sum_{i \in S} e_{ik} + \sum_{i \in S \cup T} e_{il}$, and set $\alpha := |S| + |T| - 1$. Let us first prove that if the conditions *i)* to *iv)* are satisfied, then the inequality $ax \leq \alpha$ defines a facet of $MKP(A, f, F)$. Suppose that there exists a facet defining inequality $bx \leq \beta$ with $EQ(ax \leq \alpha) \subseteq EQ(bx \leq \beta)$.



- (a) $b_{sj} = 0$ for all $s \in S$, $j \in M \setminus \{k, l\}$.

Let $s \in S$, and observe that the following vector $x^s = (x^s_{ij})$

$$x^s_{ij} := \begin{cases} 1, & i \in S \setminus \{s\}, j = k; \\ 1, & i \in T, j = l; \\ 0, & \text{otherwise} \end{cases}$$

is an element of $EQ(ax \leq \alpha)$. Furthermore, for all $j \in M \setminus \{k, l\}$, $x^s + e_{sj}$ is also in $EQ(ax \leq \alpha)$. Thus, $bx^s = b(x^s + e_{sj})$ implies $b_{sj} = 0$ for all $j \in M \setminus \{k, l\}$, $s \in S$.

- (b) $b_{sk} = c_1$ for all $s \in S$.

Let now $s' \in S \setminus \{s\}$ (there exist some, since $f_i \leq F_k$ for all $i \in N$, $k \in M$), and observe that $x^s + e_{sk} - e_{s'k}$ is an element of $EQ(ax \leq \alpha)$, since S is a minimal cover for k. Then, $bx^s = b(x^s + e_{sk} - e_{s'k})$, and, therefore, $b_{sk} = b_{s'k}$. Since the same argument holds for all $s' \in S$, there exists a constant c_1 such that $b_{sk} = c_1$ for all $s \in S$.

- (c) $b_{tl} = c_2$ for all $t \in T$.

Let $s_1 \in S$ as required in condition *ii)* and define x^{s_1} as above. Then, for all $t \in T$, $x^2 := x^{s_1} - e_{tl} + e_{s_1 l}$ is in $EQ(ax \leq \alpha)$. Then, $bx^{s_1} = bx^2$, which implies $b_{tl} = b_{s_1 l}$ for all $t \in T$. Thus, there exists a constant c_2 such that $b_{tl} = c_2$ for all $t \in T$.

- (d) $b_{tj} = 0$ for all $t \in T$, $j \in M \setminus \{k, l\}$.

Consider x^2 as defined above, and observe that $x^2 + e_{tj}$ is in $EQ(ax \leq \alpha)$ for all $j \in M \setminus \{k, l\}$. So, $bx^2 = b(x^2 + e_{tj})$, which implies $b_{tj} = 0$ for all $t \in T, j \in M \setminus \{k, l\}$.

• (e) $b_{sl} = c_2$ for all $s \in S$.

Observe now that due to condition *iii)*, for all $s \in S$ there exists a $t \in T$ such that $x^s - e_{tl} + e_{sl}$ is an element of $EQ(ax \leq \alpha)$. Then, $bx^s = b(x^s - e_{tl} + e_{sl})$, and thus, $b_{sl} = b_{tl} = c_2$, and the same argument holds for all $s \in S$.

• $c_1 = c_2$.

Finally, let $t_0 \in T$ and $S_0 \in S$ satisfying $|S_0| = 2$ and $\sum_{i \in T \setminus t_0} f_i + \sum_{i \in S_0} f_i \leq F_l$ as required in *iv)*. Let $s_2 \in S_0$ and define $x^3 = (x^3_{ij})$ and $x^4 = (x^4_{ij})$ via

$$x^3_{ij} := \begin{cases} 1, & i \in T \setminus t_0 \cup S_0, j = l; \\ 1, & i \in S \setminus S_0, j = k; \\ 0, & \text{otherwise.} \end{cases}$$

$$x^4_{ij} := \begin{cases} 1, & i \in T \setminus t_0 \cup (S_0 \setminus \{s_2\}), j = l; \\ 1, & i \in (S \setminus S_0) \cup \{s_2\}, j = k; \\ 0, & \text{otherwise.} \end{cases}$$

Observe that x^3 and x^4 are elements of $EQ(ax \leq \alpha)$. Then, $bx^3 = bx^4$, implying that, $b_{s_2 l} = b_{s_2 k}$, and therefore, $c_1 = c_2$.

So, we conclude that $bx \leq \beta$ is a scalar multiple of $ax \leq \alpha$. Thus, the Extended Cover Inequality defines a facet of $MKP(A, f, F)$.

Now, we prove that the conditions are also necessary. First suppose that S is not a minimal cover. Then, there exists a set $S' \subset S$ such that S' is also a cover for k. Then, the Extended Cover Inequality corresponding to S', T, k and l is valid. The inequality $\sum_{i \in S \setminus S'} (x_{ik} + x_{il}) \leq |S| - |S'|$ is also valid. Now, observe that the Extended Cover Inequality corresponding to S, T, k and l is the sum of the inequalities given above, and, therefore, it cannot define a facet of $MKP(A, f, F)$.

Suppose, now, that condition *ii)* is not satisfied. Let t_0 be an element of T with minimum weight. Then, $T \setminus \{t_0\} \cup \{s\}$ is a cover for all $s \in S$. Then, the Extended Cover Inequality corresponding to S, $T \setminus \{t_0\}$, k and l is valid. Moreover, $x_{t_0 l} \leq 1$ is also valid. In this case $ax \leq \alpha$ is the sum of the inequalities given above, and, therefore, cannot define a facet of $MKP(A, f, F)$.

If condition *iii)* is not satisfied, there exists an item $s \in S$ such that $T \setminus \{t\} \cup \{s\}$ is a cover for all $t \in T$. Then, if s is assigned to l, at most $|T| - 2$ elements of T can be assigned there, and then $ax \leq |S| - 1 + 1 + |T| - 2 \leq |S| + |T| - 2$. Then, $EQ(ax \leq \alpha) \subseteq \{x \in \mathbb{R}^A \mid x_{sl} = 0\}$, and the inequality cannot define a facet.

Finally, if condition *iv)* is not satisfied, $EQ(ax \leq \alpha) \subseteq \{x \in \mathbb{R}^A \mid \sum_{i \in S \cup T} x_{il} = |T|\}$. Observe that $\sum_{i \in S \cup T} x_{il} \geq |T|$ is valid for all $x \in EQ(ax \leq \alpha)$. Now, suppose that $x' \in EQ(ax \leq \alpha)$ and $\sum_{i \in S \cup T} x'_{il} > |T|$. Then, define $S_l := \{i \in S \mid x'_{il} = 1\}$ and $T_l := \{i \in T \mid x'_{il} = 1\}$. Due to our assumption, $|S_l| > |T| - |T_l|$.

Since $T \cup \{s\}$ is a cover for all $s \in S$ and condition *iv.* is not satisfied, then $|S_l| \geq 3$ and $|T_l| \leq |T| - 2$. Summing up,

$$\sum_{i \in S} x'_{ik} + \sum_{i \in S \cup T} x'_{il} = |S| - |S_l| + |S_l| + |T_l| \leq |S| + |T| - 2,$$

contradicting our assumption that $x' \in EQ(ax \leq \alpha)$. ■

Example 4.3.3 Consider the following instance of the MKP. Let $m = 3$, $n = 3$,

$$\begin{aligned} 3x_{1,1} + 4x_{2,1} + 5x_{3,1} &\leq 5, \\ 3x_{1,2} + 4x_{2,2} + 5x_{3,2} &\leq 7, \\ 3x_{1,3} + 4x_{2,3} + 5x_{3,3} &\leq 5. \end{aligned}$$

The inequality $x_{1,1} + x_{2,1} + x_{1,2} + x_{2,2} + x_{3,2} \leq 2$ is the Extended Cover Inequality corresponding to $S = \{1, 2\}$, $T = \{3\}$, $k = 1$ and $l = 2$. This inequality defines a facet of $MKP(N \times M, f, F)$. ■

We present now a generalization of the Extended Cover Inequality to more than one knapsack. Let $k, l_1, l_2, \ldots, l_t \in M$ be distinct knapsacks. Let also $S \subseteq N$ be a cover for k and $T_1, \ldots, T_t \subseteq N \setminus S$, $T_i \cap T_j = \emptyset$, for $i, j = 1, \ldots, t$. Moreover, suppose for all $j = 1, \ldots, t$ that $\sum_{i \in T_j} f_i \leq F_{l_j}$ and $T_j \cup \{s\}$ is a cover for l_j for all $s \in S$. We call the inequality:

$$\sum_{i \in S} x_{ik} + \sum_{j=1}^{t} \sum_{i \in S \cup T_j} x_{il_j} \leq |S| + \sum_{j=1}^{t} |T_j| - 1$$

the *Extended Cover Inequality* corresponding to $S, T_1, \ldots, T_t$, k and $l_1, \ldots, l_t$. The inequality is valid for the polytope $MKP((S \times M) \cup \bigcup_{j=1}^{t} (T_j \times M' \cup \{l_j\}), f, F)$, where $M' := M \setminus \{k, l_1, \ldots, l_t\}$. The proof for this fact is essentially identical to the one presented in Lemma 5.3.1.

Theorem 4.3.4 *Let $M' := M \setminus \{k, l_1, \ldots, l_t\}$ and $A := (S \times M) \cup \bigcup_{j=1}^{t} (T_j \times M' \cup \{l_j\})$. The Extended Cover Inequality corresponding to S, $T_1, \ldots, T_t$, k and $l_1, \ldots, l_t$ defines a facet of $MKP(A, f, F)$ if and only if:*

i) S is a minimal cover for k;

ii) For all $j = 1, \ldots, t$, there exists a $s_j \in S$ such that $T_j \cup \{s_j\}$ is a minimal cover for l_j;

iii) For all $j = 1, \ldots, t$, and for all $s \in S$, there exists a $t_j \in T_j$ such that $\sum_{i \in T_j \setminus \{t_j\}} f_i + f_s \leq F_{l_j}$;

iv) For all $j = 1, \ldots, t$, there exists $t_0^j \in T_j$ and $S_0^j \subseteq S$, $|S_0^j| = 2$ such that $\sum_{i \in T \setminus \{t_0^j\}} f_i + \sum_{i \in S_0^j} f_i \leq F_{l_j}$.

■

The proof for Theorem 5.3.4 is essentially the same as for Theorem 5.3.2, one must just consider each knapsack l_i and corresponding set T_i independently.

Observe that this inequality class can be obtained from Minimal Cover Inequalities applying the procedure suggested in Section 5.9.

4.4 Extended (1,d)-Configuration Inequalities

In this section we present a similar facet defining inequality of $MKP(N \times M, f, F)$. Before describing the inequality let us introduce a definition. A set S is a *t-cover* for a knapsack k if for all $T \subseteq S$, with $|T| = t-1$, $\sum_{i \in S \setminus T} f_i > F_k$ and $\sum_{i \in S \setminus (T \cup \{s\})} f_i \leq F_k$ for all $s \in S \setminus T$. Observe that a minimal cover is a 1-cover.

Suppose we are given an instance (N, M, f, F) of the MKP. Let $k, l \in M$, $Q \cup \{s\}$ be a (1,d)-configuration for the knapsack k, and $T \subseteq N \setminus (Q \cup \{s\})$ such that $\sum_{i \in T} f_i \leq F_l$. Moreover, let $Q' \subseteq Q$, with $d \leq |Q'| \leq |Q|$. The inequality,

$$\sum_{i \in Q'} x_{ik} + (|Q'| - d + 1) x_{sk} + \sum_{i \in T \cup (Q' \cup \{s\})} x_{il} \leq |Q'| + |T|$$

is called *Extended (1,d)-Configuration Inequality* corresponding to Q', $Q \cup \{s\}$, T, k and l.

Lemma 4.4.1 *The Extended (1,d)-Configuration Inequality is valid for the polytope $MKP(N \times M, f, F)$ if and only if*

i) *$T \cup \{s\}$ is a cover for l;*

ii) *for all $Q'_k \subseteq Q'$ with $|Q'_k| = d - t$, $t \geq 1$, no more than $|T| + t - 1$ items from $(Q' \setminus Q'_k) \cup T$ fits into knapsack l.*

Proof. Let us prove that the conditions are sufficient. Suppose that the inequality is not valid, and let $x' \in MKP(N \times M, f, F)$ violating the inequality. There are two cases to be analyzed.

- Case 1: $x'_{sk} = 0$.
  In this case, since x' violates the inequality, we must have $\sum_{i \in Q'} x'_{ik} = |Q'|$ and $\sum_{T \cup \{s\}} x'_{jl} = |T| + 1$. This is not possible due to condition *i)*

- Case 2: $x'_{sk} = 1$.
  Since x' violates the inequality,

$$\sum_{i \in Q'} x'_{ik} + \sum_{j \in T \cup Q'} x'_{jl} > |T| + d - 1.$$

$Q \cup \{s\}$ is a (1,d)-configuration for k, then $\sum_{i\in Q'} x'_{ik} \le d-1$. Define $Q'_k := \{i \in Q' \mid x'_{ik} = 1\}$ and $|Q'_k| = d-t$. By the condition *ii)*, $\sum_{i\in(Q'\setminus Q'_k)\cup T} x'_{il} \le |T|+t-1$, what contradicts our assumption that the inequality is violated by x'.

Now we prove that the conditions are also necessary. Suppose that the inequality is valid. If the condition *i)* does not hold, consider the following vector $x^1 = (x^1_{ij}) \in MKP(N \times M, f, F)$.

$$x^1_{ij} := \begin{cases} 1, & i \in Q', j = k; \\ 1, & i \in T \cup \{s\}, j = l; \\ 0, & \text{otherwise} \end{cases}$$

The vector x^1 violates the inequality which is a contradiction to the assumption that the inequality is valid.

Now, suppose that condition *ii)* is not satisfied. Then, there exists a set $Q'_k \subseteq Q'$ such that $|Q'_k| = d-t$, $t \ge 1$, and a set $R \subseteq (Q' \setminus Q'_k) \cup T$, such that $|R| \ge |T|+t$ and $\sum_{i\in R} f_i \le F_l$. Then, the assignment $x^2 = (x^2_{ij})$ is in $MKP(N \times M, f, F)$, where x^2 is defined as follows.

$$x^2_{ij} := \begin{cases} 1, & i \in Q'_k \cup \{s\}, j = k; \\ 1, & i \in R, j = l; \\ 0, & \text{otherwise.} \end{cases}$$

But, this contradicts the assumption that the inequality is valid, since

$$\begin{aligned} \sum_{i\in Q'} x^2_{ik} + (|Q'|-d+1)x^2_{sk} + \sum_{i\in T\cup Q'\cup\{s\}} x^2_{il} &\ge d-t+|Q'|-d+1+|T|+t \\ &= |Q'|+|T|+1. \end{aligned}$$

■

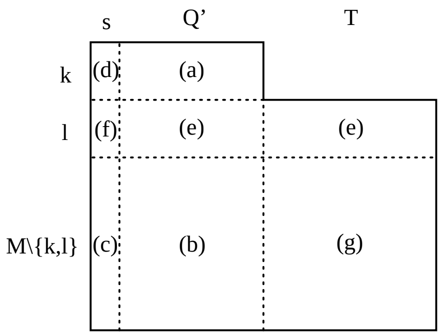
In the next theorem we present necessary and sufficient conditions for the Extended (1,d)-Configuration Inequality to define a facet of the Multiple Knapsack Polytope.

Theorem 4.4.2 *Suppose we are given an instance $N, M, f, F)$ of the MKP. Let $k, l \in M$, $Q \cup \{s\}$ be a (1,d)-configuration for knapsack k. Moreover let $Q' \subseteq Q$ with $d \le |Q'| \le |Q|$, and $T \subseteq N \setminus (Q \cup \{s\})$ satisfying $\sum_{i\in T} f_i \le F_l$. Further, suppose that $T \cup \{s\}$ is a minimal cover for l and for all $Q'' \subseteq Q'$, with $|Q''| \le |Q'|-d+1$, $T \cup Q''$ is a $|Q''|$-cover for knapsack l. Let, also, $A := (Q' \cup \{s\} \times M) \cup (T \times (M \setminus \{k\}))$. Then, the corresponding Extended (1,d)-Configuration Inequality defines a facet of $MKP(A, f, F)$ if and only if the inequality is valid, and*

- *i) for some $Q'_k \subseteq Q'$, $|Q'_k| = d-t$, $t > 1$, there exists a set $R \subseteq T \cup (Q' \setminus Q'_k)$ such that $|R| = |T|+t-1$ and $\sum_{i\in R} f_i \le F_l$, or;*

ii) there exists some $i \in Q$ and $t \in T$ such that $\sum_{i \in T\setminus\{t\}\cup\{s,i\}} f_i \leq F_l$.

Proof. Let $ax \leq \alpha$ be the Extended (1,d)-Configuration Inequality corresponding to Q', $Q \cup \{s\}$, T, k and l. First we prove that under the conditions given in the theorem, the inequality defines a facet of $MKP(A, f, F)$. In order to prove that it defines a facet of $MKP(A, f, F)$, suppose that there exists a facet defining inequality $bx \leq \beta$ such that $EQ(ax \leq \alpha) \subseteq EQ(bx \leq \beta)$.



• (a) $b_{ik} = c_1$ for all $i \in Q'$.
Let $Q'_k \subseteq Q'$, $|Q'_k| = d - 1$ and define $x^1 = (x^1_{ij})$ as follows.

$$x^1_{ij} := \begin{cases} 1, & i \in Q'_k \cup \{s\}, j = k; \\ 1, & i \in T, j = l; \\ 0, & \text{otherwise} \end{cases}$$

Observe that x^1 is in $EQ(ax \leq \alpha)$. Since $Q \cup \{s\}$ is a (1,d)-configuration, for all $i_1 \in Q'_k$ and $i_2 \in Q' \setminus Q'_k$, $x^1 - e_{i_1k} + e_{i_2k}$ is also in $EQ(ax \leq \alpha)$. Then, $bx^1 = b(x^1 - e_{i_1k} + e_{i_2k})$, and therefore, $b_{i_1k} = b_{i_2k}$ for all $i_1 \in Q'_k$ and $i_2 \in Q' \setminus Q'_k$. Thus, there is a constant c_1 such that $b_{ik} = c_1$ for all $i \in Q'$.
• (b) $b_{ij} = 0$ for all $i \in Q'$, $j \in M \setminus \{k, l\}$.
Consider x^1 as defined above, and observe that $x^1 + e_{ij}$ is also an element of $EQ(ax \leq \alpha)$ for all $i \in Q' \setminus Q'_k$ and $j \in M \setminus \{k, l\}$. Then, $bx^1 = b(x^1 + e_{ij})$, and therefore, $b_{ij} = 0$ for all $i \in Q' \setminus Q'_k$ and $j \in M \setminus \{k, l\}$. Using the same argument for different choices of Q'_k, one can conclude that $b_{ij} = 0$ for all $i \in Q'$, $j \in M \setminus \{k, l\}$.
• (c) $b_{sj} = 0$ for all $j \in M \setminus \{k, l\}$.
Consider, now, the following vector $x^2 = (x^2_{ij})$.

$$x^2_{ij} := \begin{cases} 1, & i \in Q', j = k; \\ 1, & i \in T, j = l; \\ 0, & \text{otherwise} \end{cases}$$

The vector x^2 is also an element of $EQ(ax \leq \alpha)$. Since $x^2 + e_{sj}$ is also in $EQ(ax \leq \alpha)$ for all $j \in M \setminus \{k, l\}$, $bx^2 = b(x^2 + e_{sj})$, and thus, $b_{sj} = 0$ for all $j \in M \setminus \{k, l\}$.

• (d) $b_{sk} = (|Q'| - d + 1)c_1$.

Consider x^1 and x^2 as defined above, and observe that $bx^1 = bx^2$ implies $(d-1)b_{qk} + b_{sk} = |Q'|b_{qk}$. Then, $b_{sk} = (|Q'| - d + 1)b_{qk}$.

• (e) $b_{il} = b_{tl} = c_2$ for all $i \in Q'$, $t \in T$.

Consider x^1 as defined above. Observe that $x^1 - e_{tl} + e_{il}$ is an element of $EQ(ax \leq \alpha)$ for all $t \in T$, $i \in Q'_k$, since $T \cup \{i\}$ is a 1-cover for all $i \in Q'_k$. Then, $bx^1 = b(x^1 - e_{tl} + e_{il})$, and therefore, $b_{tl} = b_{il}$ for all $t \in T$, $i \in Q'_k$. Repeating the same argument for different choices of Q'_k, one can prove that there exists a constant c_2 such that $b_{tl} = b_{il} = c_2$ for all $t \in T$, $i \in Q'$.

• (f) $b_{sl} = c_2$.

Let $t \in T$ and consider $x^3 := x^2 - e_{tl} + e_{sl}$. The vector x^3 is also in $EQ(ax \leq \alpha)$, then $bx^2 = bx^3$, and, therefore, $b_{sl} = b_{tl} = c_2$.

• (g) $b_{tj} = 0$ for all $t \in T$, $j \in M \setminus \{k, l\}$.

Observe that $x^3 + e_{tj}$ is also in $EQ(ax \leq \alpha)$ for all $j \in M \subseteq \{k, l\}$. Then, $bx^3 = b(x^3 + e_{tj})$, and therefore, $b_{tj} = 0$ for all $t \in T$ and $j \in M \subseteq \{k, l\}$.

• $c_1 = c_2$.

We know that at least one of the two conditions given in the theorem must hold. Let us analyze the two cases.

Case 1. For some $Q'_k \subseteq Q'$, $|Q'_k| = d - t$ and $t > 1$ there exists $R \subseteq T \cup (Q' \setminus Q'_k)$ such that $|R| = |T| + t - 1$ and $\sum_{i \in R} f_i \leq F_l$. In this case, consider the following vector $x^4 = (x^4_{ij})$.

$$x^4_{ij} := \begin{cases} 1, & i \in Q'_k \cup \{s\}, j = k; \\ 1, & i \in R, j = l; \\ 0, & \text{otherwise} \end{cases}$$

The vector x^4 is an element of $EQ(ax \leq \alpha)$. Since $|R| > |T|$, there exists a $i \in R \cap Q'$. Then, $x^4 - e_{il} + e_{ik}$ is an element of $EQ(ax \leq \alpha)$ (since $t > 1$). Then $bx^4 = b(x^4 - e_{il} + e_{ik})$, and therefore, $b_{ik} = b_{il}$. In this case, we observe that the inequality $bx \leq \beta$ is a scalar multiple of $ax \leq \alpha$.

Case 2. There exist some $t \in T$, $i \in Q$ such that $\sum_{i \in T \setminus \{t\} \cup \{s,i\}} f_i \leq F_l$. Consider the following vector $x^5 = (x^5_{ij})$.

$$x^5_{ij} := \begin{cases} 1, & i \in Q' \setminus \{i\}, j = k; \\ 1, & i \in T \setminus \{t\} \cup \{s, i\}, j = l; \\ 0, & \text{otherwise} \end{cases}$$

The vector x^5 is an element of $EQ(ax \leq \alpha)$. Moreover, $x^5 - e_{il} + e_{ik}$ is also in $EQ(ax \leq \alpha)$. Thus, $bx^5 = b(x^5 - e_{il} + e_{ik})$, and therefore, $b_{il} = b_{ik}$. The inequality $bx \leq \beta$ is a scalar multiple of $ax \leq \alpha$.

In both cases the inequality $bx \leq \beta$ is a scalar multiple of $ax \leq \alpha$, and thus, it defines a facet of $MKP(A, f, F)$.

Let us now prove that the conditions are also necessary. Suppose that $ax \leq \alpha$ is a facet defining inequality. Then, it must be valid. Suppose, now, that both conditions given in the theorem do not hold. Let $x' \in EQ(ax \leq \alpha)$ and consider the following two cases.

Case 1. $x'_{sk} = 0$.
In this case, since the condition *ii)* is not satisfied, x'_{sl} must also be zero. Then, $\sum_{i \in Q' \cup T \cup \{s\}} x'_{il} = |T|$.

Case 2. $x'_{sk} = 1$.
In this case, since the condition *i)* does not hold, $\sum_{i \in Q'} x'_{ik} = d - 1$, and, therefore $\sum_{i \in Q' \cup T \cup \{s\}} x'_{il} = |T|$.

In both cases the vector x' satisfies the equality $\sum_{i \in Q' \cup T \cup \{s\}} x'_{il} = |T|$. Then, $EQ(ax \leq \alpha) \subseteq \{x \in \mathbb{R}^A \mid \sum_{i \in Q' \cup T \cup \{s\}} x_{il} = |T|\}$, contradicting the assumption that $ax \leq \alpha$ defines a facet of $MKP(A, f, F)$. ■

4.5 Combined Covers Inequalities

In this section we present a facet defining inequality of the $MKP(N \times M, f, F)$: the Combined Covers Inequality. We discuss, in the following, conditions for the inequality be valid and facet defining. The conditions imposed in order that the inequality is facet defining are local in each knapsack. Then, similar ideas lead to facet defining inequalities of the polytope associated with the GAP.

Suppose we are given an instance (N, M, f, F) of the MKP. Let $k_1, k_2, k_3 \in M$ be three different knapsacks. Let $S_1 \subseteq N$ be a cover for k_1, $S_2 \subseteq N$ be a cover for knapsack k_2, with $S_1 \cap S_2 \neq \emptyset$, and $T \subseteq N \setminus (S_1 \cup S_2)$ such that $\sum_{i \in T} f_i \leq F_3$ and $\sum_{i \in T \cup \{i\}} f_i > F_3$ for all $i \in S_1 \cup S_2$. Then, we call the inequality

$$\sum_{i \in S_1} x_{ik_1} + \sum_{i \in S_2} x_{ik_2} + \sum_{i \in S_1 \cup S_2 \cup T} x_{ik_3} \leq |S_1| + |S_2| + |T| - 2,$$

the *Combined Covers Inequality* corresponding to S_1, S_2, T, k_1, k_2 and k_3.

Lemma 4.5.1 *The Combined Covers Inequality is valid for $MKP(N \times M, f, F)$.*

Proof. Suppose it is not the case, and let x' be a vector in $MKP(N \times M, f, F)$ violating the inequality. Due to the SOS constraints, $\sum_{i \in S_1} x'_{ik_1} + \sum_{i \in S_2} x'_{ik_2} + \sum_{i \in S_1 \cup S_2 \cup T} x'_{ik_3} \leq |S_1| + |S_2| + |T| - |S_1 \cap S_2|$. If $|S_1 \cap S_2| > 1$ the inequality cannot be violated. Then assume $|S_1 \cap S_2| = 1$. All items in $S_1 \cup S_2 \cup T$ must be assigned to the three knapsacks. Then, in k_3 only the items of T are present (since $T \cup \{i\}$ is a cover for all $i \in S_1 \cup S_2$). But then, S_1 and S_2 must be completely assigned to k_1 and k_2, what is not possible, since they are covers for the corresponding knapsacks. ■

In the following theorem we present necessary and sufficient conditions for the Combined Covers Inequality to define a facet.

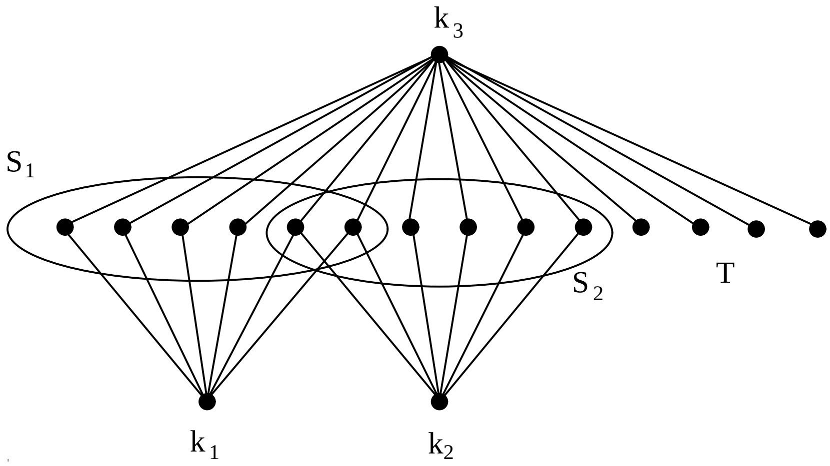
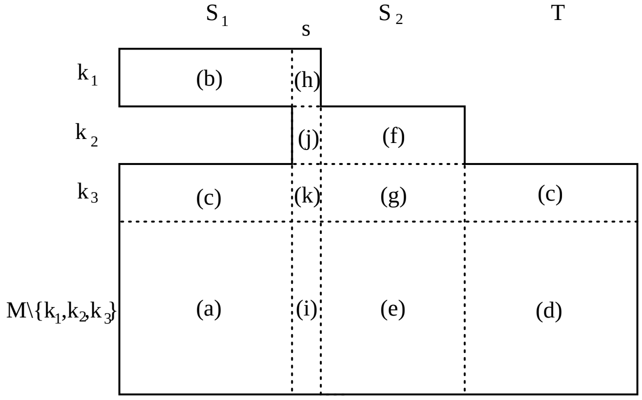


Figure 4.2: Combined Covers Inequality.

Theorem 4.5.2 *Suppose we are given an instance (N, M, f, F) of the MKP. Let $k_1,\ k_2, k_3 \in M,\ S_1 \subseteq N$ a minimal cover for $k_1,\ S_2 \subseteq N$ a minimal cover for k_2 with $S_1 \cap S_2 = \{s\}$, and $T \subseteq N \setminus (S_1 \cup S_2)$, such that $T \cup \{i\}$ is a minimal cover with respect to k_3 for all $i \in S_1 \cup S_2$. Further, suppose that s_1 is an element in $S_1 \setminus \{s\}$ with minimum weight and $f_{s_1} \leq f_i$ for all $i \in S_2 \setminus \{s\}$. Define $A := (S_1 \times M \setminus \{k_2\}) \cup (S_2 \times M \setminus \{k_1\}) \cup (T \times M \setminus \{k_1, k_2\})$. Then, the corresponding Combined Covers Inequality defines a facet of $MKP(A, f, F)$ if and only if there exists a set $R \subseteq S_1 \cup S_2 \cup T$ with $|R \cap T| = |T| - 1$, $\sum_{i \in R} f_i \leq F_{k_3}$ and R has nonempty intersection with both S_1 and S_2 and $R \cap S_2 \neq \{s\}$.*

Proof. Let us represent the Combined Covers Inequality as $ax \leq \alpha$. We prove that the condition is sufficient. To prove that the inequality defines a facet of the polytope, let $bx \leq \beta$ be a facet defining inequality with $EQ(ax \leq \alpha) \subseteq EQ(bx \leq \beta)$.



- (a) $b_{ij} = 0$ for all $i \in S_1 \setminus \{s\},\ j \in M \setminus \{k_1, k_2, k_3\}$.

Let $i \in S_1 \setminus \{s\}$ and observe that $x^1 = (x^1_{ij})$ defined as follows.

$$x^1_{ij} := \begin{cases} 1, & i \in S_1 \setminus \{i\}, j = k_1; \\ 1, & i \in S_2 \setminus \{s\}, j = k_2; \\ 1, & i \in T, j = k_3; \\ 0, & \text{otherwise.} \end{cases}$$

The vector x^1 is an element of $EQ(ax \leq \alpha)$. Observing that $x^1 + e_{ij}$ for all $j \in M \setminus \{k_1, k_2, k_3\}$ is in $EQ(ax \leq \alpha)$, we can conclude that $bx^1 = b(x^1 + e_{ij})$, and, therefore, $b_{ij} = 0$, for all $i \in S_1 \setminus \{s\}$ and $j \in M \setminus \{k_1, k_2, k_3\}$.

• (b) $b_{ik_1} = c_1$ for all $i \in S_1 \setminus \{s\}$

Consider x^1 as defined above, and observe that $x^1 - e_{jk_1} + e_{ik_1}$ is in $EQ(ax \leq \alpha)$ for $j \in S_1 \setminus \{s\}$, $j \neq i$, we can conclude that there exists a constant c_1 such that $b_{jk_1} = c_1$ for all $j \in S_1 \setminus \{s\}$.

• (c) $b_{ik_3} = b_{tk_3} = c_3$ for all $i \in S_1 \setminus \{s\}$, $t \in T$.

Observe that $x^{1'} := x^1 - e_{tk_3} + e_{ik_3}$ is in $EQ(ax \leq \alpha)$ for all $t \in T$, since $T \cup \{i\}$ is a minimal cover with respect to k_3 for all $i \in S_1 \cup S_2$. Thus, $bx^1 = b(x^1 - e_{tk_3} + e_{ik_3})$, and, therefore, there exists a constant c_3 such that $b_{tk_3} = b_{ik_3} = c_3$ for all $i \in S_1 \setminus \{s\}$ and $t \in T$.

• (d) $b_{tj} = 0$ for all $t \in T$, $j \in M \setminus \{k_1, k_2, k_3\}$.

Consider again $x^{1'}$ as defined above and observe that $x^{1'} + e_{tj}$ is in $EQ(ax \leq \alpha)$ for all $j \in M \setminus \{k_1, k_2, k_3\}$. Then, $bx^{1'} = b(x^{1'} + e_{tj})$ and therefore, $b_{tj} = 0$ for all $t \in T$, $j \in M \setminus \{k_1, k_2, k_3\}$.

• (e) $b_{s_2 j} = 0$ for all $s_2 \in S_2 \setminus \{s\}$, $j \in M \setminus \{k_1, k_2, k_3\}$.

Analogously, let $s_2 \in S_2 \setminus \{s\}$ and define $x^2 = (x^2_{ij})$.

$$x^2_{ij} := \begin{cases} 1, & i \in S_1 \setminus \{s\}, j = k_1; \\ 1, & i \in S_2 \setminus \{s_2\}, j = k_2; \\ 1, & i \in T, j = k_3; \\ 0, & \text{otherwise.} \end{cases}$$

As in the case outlined above, observe that x^2 is contained in $EQ(ax \leq \alpha)$, and for all $j \in M \setminus \{k_1, k_2, k_3\}$, $x^2 + e_{s_2 j}$ is in $EQ(ax \leq \alpha)$. Then, $bx^2 = b(x^2 + e_{s_2 j})$, and therefore, $b_{s_2 j} = 0$, for all $s_2 \in S_2 \setminus \{s\}$ and $j \in M \setminus \{k_1, k_2, k_3\}$.

• (f) $b_{s_2 k_2} = c_2$ for all $s_2 \in S_2 \setminus \{s\}$.

Observe that $x^2 - e_{jk_2} + e_{s_2 k_2}$ is in $EQ(ax \leq \alpha)$ for $j \in S_2 \setminus \{s\}$, $j \neq s_2$. Then, we can conclude that there exists a constant c_2 such that $b_{s_2 k_2} = c_2$ for all $s_2 \in S_2 \setminus \{s\}$.

• (g) $b_{s_2 k_3} = c_3$ for all $s_2 \in S_2 \setminus \{s\}$.

Consider x^2 as defined above and observe that $x^2 - e_{tk_3} + e_{s_2 k_3}$ is in $EQ(ax \leq \alpha)$. So, $bx^2 = b(x^2 - e_{tk_3} + e_{s_2 k_3})$, and, therefore, $b_{s_2 k_3} = b_{tk_3} = c_3$ for all $s_2 \in S_2 \setminus \{s\}$.

• (h) $b_{sk_1} = c_1$.

Now, observe the following vector $x^3 = (x^3_{ij})$.

$$x^3_{ij} := \begin{cases} 1, & i \in S_1 \setminus \{s\}, j = k_1; \\ 1, & i \in S_2 \setminus \{s\}, j = k_2; \\ 1, & i \in T, j = k_3; \\ 0, & \text{otherwise.} \end{cases}$$

The vector x^3 is in $EQ(ax \leq \alpha)$. Then, $bx^1 = bx^3$, and, therefore, $b_{sk_1} = c_1$.

• (i) $b_{sj} = 0$ for all $j \in M \setminus \{k_1, k_2, k_3\}$.

Since $x^3 + e_{sj}$ is an element of $EQ(ax \leq \alpha)$ for all $j \in M \setminus \{k_1, k_2, k_3\}$ we can conclude that $bx^3 = b(x^3 + e_{sj})$ and therefore, $b_{sj} = 0$ for all $j \in M \setminus \{k_1, k_2, k_3\}$.

• (j) $b_{sk_2} = c_2$.

Considering that x^2 and x^3 are in $EQ(ax \leq \alpha)$, $bx^2 = bx^3$, and, then, $b_{sk_2} = c_2$.

• (k) $b_{sk_3} = c_3$.

Further, for all $t \in T$, $x^3 - e_{tk_3} + e_{sk_3}$ is in $EQ(ax \leq \alpha)$. Thus, $bx^3 = b(x^3 - e_{tk_3} + e_{sk_3})$, and, therefore, $b_{sk_3} = b_{tk_3} = c_3$.

• $c_2 = c_3$.

Finally, let $R \subseteq S_1 \cup S_2 \cup T$ with $|R \cap T| = |T| - 1$, $\sum_{i \in R} f_i \leq F_{k_3}$, $\emptyset \neq R \cap S_1$ and $R \cap S_2 \neq \{s\} \neq \emptyset$ as required in the theorem. Let $R_1 := R \cap S_1$ and $R_2 := R \cap S_2$. We distinguish two cases:

Case 1. $s \notin R$.

Define, then, the vector $x^4 = (x^4_{ij})$ as follows.

$$x^4_{ij} := \begin{cases} 1, & i \in S_1 \setminus R_1, j = k_1; \\ 1, & i \in S_2 \setminus (R_2 \cup \{s\}), j = k_2; \\ 1, & i \in R, j = k_3; \\ 0, & \text{otherwise.} \end{cases}$$

The vector x^4 is an element of $EQ(ax \leq \alpha)$. Let $r_2 \in R_2$, and observe that $x^4 - e_{r_2k_3} + e_{r_2k_2}$ is also in $EQ(ax \leq \alpha)$. Then, $bx^4 = b(x^4 - e_{r_2k_3} + e_{r_2k_2})$, and therefore, $c_2 = c_3$.

Case 2. $s \in R$.

In this case, define the vector $x^{4'} = ({x^{4'}}_{ij})$ as follows.

$$ {x^{4'}}_{ij} := \begin{cases} 1, & i \in S_1 \setminus R_1, j = k_1; \\ 1, & i \in S_2 \setminus R_2, j = k_2; \\ 1, & i \in R, j = k_3; \\ 0, & \text{otherwise.} \end{cases}$$

The vector $x^{4'}$ is an element of $EQ(ax \leq \alpha)$. Again, Let $r_2 \in R_2 \setminus \{s\}$, and observe that $x^{4'} - e_{r_2k_3} + e_{r_2k_2}$ is also in $EQ(ax \leq \alpha)$. Then, $bx^{4'} = b(x^{4'} - e_{r_2k_3} + e_{r_2k_2})$, and therefore, $c_2 = c_3$.

- $c_1 = c_3$.
  Analogously, we distinguish into two cases.

Case 1. $s \notin R$.
Define $x^5 = (x^5_{ij})$ as follows.

$$x^5_{ij} := \begin{cases} 1, & i \in S_1 \setminus (R_1 \cup \{s\}), j = k_1; \\ 1, & i \in S_2 \setminus R_2, j = k_2; \\ 1, & i \in R, j = k_3; \\ 0, & \text{otherwise.} \end{cases}$$

The vector x^5 is an element of $EQ(ax \leq \alpha)$. Let $r_1 \in R_1$, and observe that $x^5 - e_{r_1k_3} + e_{r_1k_1}$ is also in $EQ(ax \leq \alpha)$. Then, $bx^5 = b(x^5 - e_{r_1k_3} + e_{r_1k_1})$, and therefore, $c_1 = c_3$.

Case 2. $s \in R$.
In this case, we must consider two possibilities.

Case 2.1. $R \cap S_1 \neq \{s\}$.
Define $x^{5'} = ({x^{5'}}_{ij})$ as follows.

$$ {x^{5'}}_{ij} := \begin{cases} 1, & i \in S_1 \setminus R_1, j = k_1; \\ 1, & i \in S_2 \setminus R_2, j = k_2; \\ 1, & i \in R, j = k_3; \\ 0, & \text{otherwise.} \end{cases}$$

The vector $x^{5'}$ is in $EQ(ax \leq \alpha)$. Let $r_1 \in R_1 \setminus \{s\}$, and observe that $x^{5'} - e_{r_1k_3} + e_{r_1k_1}$ is also in $EQ(ax \leq \alpha)$. Then, $bx^{5'} = b(x^{5'} - e_{r_1k_3} + e_{r_1k_1})$, and therefore, $c_1 = c_3$.

Case 2.2. $R \cap S_1 = \{s\}$.
In this case, let $s_2 \in R \cap S_2 \setminus \{s\}$ and define $R' := R \setminus \{s_2\} \cup \{s_1\}$, $R'_1 := R' \cap S_1$ and $R'_2 := R' \cap S_2$.
Define $x^6 = (x^6_{ij})$ as follows.

$$x^6_{ij} := \begin{cases} 1, & i \in S_1 \setminus R'_1, j = k_1; \\ 1, & i \in S_2 \setminus R'_2, j = k_2; \\ 1, & i \in R', j = k_3; \\ 0, & \text{otherwise.} \end{cases}$$

The vector x^6 is also an element of $EQ(ax \leq \alpha)$. Moreover, $x^6 - e_{s_1k_3} + e_{s_1k_1}$ is also in $EQ(ax \leq \alpha)$. Thus, $b(x^6) = b(x^6 - e_{s_1k_3} + e_{s_1k_1})$ and then, $c_1 = c_3$.

We conclude that the inequality $bx \leq \beta$ is a scalar multiple of $ax \leq \alpha$, and then, it defines a facet of $MKP(A, f, F)$.

Now, we prove that the condition is also necessary. Suppose that $ax \leq \alpha$ defines a facet of $MKP(A, f, F)$ and the condition in the theorem is not satisfied, i.e., for all $R \subseteq S_1 \cup S_2 \cup T$ with $|R \cap T| = |T| - 1$ and $\sum_{i \in R} f_i \leq F_{k_3}$, R has empty intersection with S_1 or S_2, or $R \cap S_2 = \{s\}$.

Consider $x' \in EQ(ax \leq \alpha)$ such that $\sum_{i \in S_1 \cup S_2 \cup T} x'_{ik_3} \geq |T| + 1$ (there must be some, otherwise $EQ(ax \leq \alpha) \subseteq \{x \in \mathbb{R}^A \mid \sum_{i \in S_1} x_{ik_1} + \sum_{i \in S_2} x_{ik_2} = |S_1| + |S_2| - 2\}$, and the inequality cannot be facet defining). Let $R(x') := \{i \in S_1 \cup S_2 \cup T \mid x'_{ik_3} = 1\}$. Observe that $|R(x') \cap T| \leq |T| - 1$. If $|R(x') \cap T| \leq |T| - 2$, then due to the SOS constraints, $\sum_{i \in S_1} x_{ik_1} + \sum_{i \in S_1} x_{ik_2} + \sum_{i \in S_1 \cup S_2 \cup (R(x') \cap T)} x_{ik_3} \leq |S_1| + |S_2| + |T| - 2 - |S_1 \cap S_2|$, contradicting the assumption that $x' \in EQ(ax \leq \alpha)$.

Then, $|R(x') \cap T| = |T|_1$. Since the condition in the theorem is not satisfied, then $R(x') \cap S_1 = \emptyset$ or $R(x') \cap S_2 = \emptyset$ or $R(x') \cap S_2 = \{s\}$. This fact holds for all $x' \in EQ(ax \leq \alpha)$ with $\sum_{i \in S_1 \cup S_2 \cup T} x'_{ik_3} \geq |T| + 1$.

Observe that if there exists $x^1 \in EQ(ax \leq \alpha)$ with $\sum_{i \in S_1 \cup S_2 \cup T} x^1_{ik_3} \geq |T| + 1$ and $R(x^1) \cap S_1 = \emptyset$, then we can construct a point $x^3 \in EQ(ax \leq \alpha)$ in the following way. W.l.o.g. we can suppose that $x^1_{s_1 k_1} = 0$ (otherwise we can exchange the items in k_1). Let $s_2 \in R(x^1) \cap S_2$ and define $x^3 := x^1 - e_{s_2 k_3} + e_{s_1 k_3}$. Observe that $x^3 \in EQ(ax \leq \alpha)$, $\sum_{i \in S_1 \cup S_2 \cup T} x^3_{ik_3} = \sum_{i \in S_1 \cup S_2 \cup T} x^1_{ik_3} \geq |T| + 1$, and $R(x^3)$ satisfies the condition in the theorem. This contradicts our assumption.

Then, we can suppose that for all $x' \in EQ(ax \leq \alpha)$ with $\sum_{i \in S_1 \cup S_2 \cup T} x'_{ik_3} \geq |T| + 1$, either $R(x') \cap S_2 = \emptyset$ or $R(x') \cap S_2 = \{s\}$. But, in this case we prove that $EQ(ax \leq \alpha) \subseteq \{x \in \mathbb{R}^A \mid \sum_{i \in S_2} x_{ik_2} = |S_2| - 1\}$, a contradiction to our assumption that $ax \leq \alpha$ is facet defining.

In order to prove that $EQ(ax \leq \alpha) \subseteq \{x \in \mathbb{R}^A \mid \sum_{i \in S_2} x_{ik_2} = |S_2| - 1\}$, assume that it is not the case, i.e., there exists a $x^* \in EQ(ax \leq \alpha)$ with $\sum_{i \in S_2} x^*_{ik_2} \leq |S_2| - 2$. In this case, observe that $\sum_{i \in S_1 \cup S_2 \cup T} x^*_{ik_3} \geq |T| + 1$, and therefore, either $R(x^*) \cap S_2 = \emptyset$ or $R(x^*) \cap S_2 = \{s\}$. Thus, if $|R(x^*)| = |T| + t$, then $|R(x^*) \cap S_1| = t + 1$. Summing up,

$$\begin{aligned} ax^* &= |S_1| - |R(x^*) \cap S_1| + |S_2| - 2 + |R(x^*)| \\ &\leq |S_1| - (t+1) + |S_2| - 2 + |T| + t = |S_1| + |S_2| + |T| - 3. \end{aligned}$$

This contradicts the assumption that $x^* \in EQ(ax \leq \alpha)$. ■

Example 4.5.3 Consider the following instance of the MKP: $n = 4$, $m = 3$,

$$\begin{aligned} 3x_{1,1} + 5x_{2,1} + 3x_{3,1} + 6x_{4,1} &\leq 7 \\ 3x_{1,2} + 5x_{2,2} + 3x_{3,2} + 6x_{4,2} &\leq 7 \\ 3x_{1,3} + 5x_{2,3} + 3x_{3,3} + 6x_{4,3} &\leq 6 \end{aligned}$$

The Combined Covers Inequality corresponding to $S_1 = \{1, 2\}$, $k_1 = 1$, $S_2 = \{2, 3\}$, $k_2 = 2$, $T = \{4\}$ and $k_3 = 3$ is $x_{1,1} + x_{2,1} + x_{2,2} + x_{3,2} + x_{1,3} + x_{2,3} + x_{3,3} + x_{4,3} \leq 3$ defines a facet of $MKP(N \times M, f, F)$. ■

4.6 Heterogeneous Two Cover Inequalities

In this section we present an inequality that involves two covers and two knapsacks. Observe also that the coefficients of the inequality are not all equal to zero or one. As in the previous sections, similar ideas can be used to provide facet defining inequalities for the polytope associated with GAP, since all properties required to the inequality be valid or facet defining are local in each knapsack.

Suppose we are given an instance (N, M, f, F) of the MKP. Let $k, l \in M, k \neq l$, S be a cover with respect to k and let $G \subseteq N \setminus S$. Then, we call the inequality

$$\sum_{i \in S} x_{ik} + \sum_{i \in S \cup G} (|S| - 1)x_{il} \leq |S|(|S| - 1),$$

the *Heterogeneous Two Cover Inequality* corresponding to S, G, k and l. In the following lemma we characterize when the inequality is valid for the polytope $MKP((S \cup G) \times M, f, F)$.

Lemma 4.6.1 *The Heterogeneous Two Cover Inequality is valid for $MKP((S \cup G) \times M, f, F)$ if and only if for all $R \subseteq S \cup G$, $|R| \geq |S|$, $R \cap G \neq \emptyset$, then R is a cover for knapsack l.*

Proof. Let $x' \in MKP((S \cup G) \times M, f, F)$ violating the Heterogeneous Two Cover Inequality. It implies, since S is a cover for knapsack k, that $\sum_{i \in S \cup G} x'_{il} \geq |S|$. Define $R := \{i \in S \cup G \mid x'_{il} = 1\}$. The condition of the theorem above implies that $R = S$. But, in this case, x' does not violate the inequality, which contradicts our assumption.

Now we prove that the condition is also necessary. Suppose that the inequality is valid but the condition does not hold, i.e., there exists a $R \subseteq S \cup G$, $|R| \geq |S|$, $R \cap G \neq \emptyset$ with R fitting into knapsack l. We analyze two cases.

- $S \not\subseteq R$.
  Define the following vector $x^1 = (x^1_{ij})$,

$$x^1_{ij} := \begin{cases} 1, & i \in S \setminus R, j = k; \\ 1, & i \in R, j = l; \\ 0, & \text{otherwise.} \end{cases}$$

  The vector x^1 is an element of $MKP((S \cup G) \times M, f, F)$. But, $\sum_{i \in S} x^1_{ik} + \sum_{i \in S \cup G}(|S| - 1)x^1_{il} \geq 1 + (|S| - 1)|S|$. It contradicts the assumption that the inequality is valid.

- $S \subseteq R$.
  In this case, define $x^2 = (x^2_{ij})$ as follows.

$$x^2_{ij} := \begin{cases} 1, & i \in R, j = l; \\ 0, & \text{otherwise.} \end{cases}$$

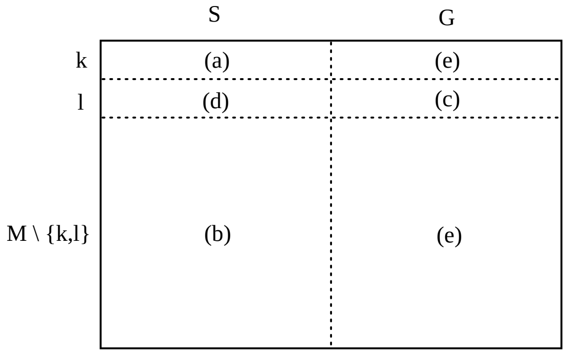
Since $S \subseteq R$ and $R \cap G \neq \emptyset$, then $|R| \geq |S| + 1$. Then, $\sum_{i \in S} x_{ik}^2 + \sum_{i \in S \cup G}(|S| - 1)x_{il}^2 \geq (|S| - 1)(|S| + 1)$, contradicting the assumption that the inequality is valid.

■

We investigate now the instances for which the Heterogeneous Two Cover Inequality defines a facet.

Theorem 4.6.2 *Given an instance (N, M, f, F) of the MKP. Let $k, l \in M$, $S \subseteq N$ be a minimal cover for $k \in M$ with $\sum_{i \in S} f_i \leq F_l$. Moreover, let any $G \subseteq N \setminus S, |G| \geq |S|$ define a $(|G| - |S| + 1)$-cover with respect to l. The corresponding Heterogeneous Two Cover Inequality defines a facet for the polytope $MKP((S \cup G) \times M, f, F)$ if and only if it is valid for $MKP((S \cup G) \times M, f, F)$ and for every $s \in S$ there exists a subset $G' \subseteq G$, $|G'| = |S| - 2$ satisfying $\sum_{i \in G'} f_i + f_s \leq F_l$.*

Proof. Let us prove that if the conditions given above are satisfied, then the inequality defines a facet of $MKP((S \cup G) \times M, f, F)$. We represent the Heterogeneous Two Cover Inequality as $ax \leq \alpha$. In order to prove that it defines a facet of $MKP((S \cup G) \times M, f, F)$ we suppose that there exists a facet defining inequality $bx \leq \beta$ such that $EQ(ax \leq \alpha) \subseteq EQ(bx \leq \beta)$.



• (a) $b_{sk} = c_1$ for all $s \in S$.

Let $s \in S$ and $\tilde{G} \subseteq G$, $|\tilde{G}| = |S| - 1$ and consider the following vector $x^s = (x_{ij}^s)$.

$$x_{ij}^s := \begin{cases} 1, & i \in S \setminus \{s\}, j = k; \\ 1, & i \in \tilde{G}, j = l; \\ 0, & \text{otherwise.} \end{cases}$$

Observe that x^s is an element of $EQ(ax \leq \alpha)$. For all $s' \in S \setminus \{s\}$, $x^s - e_{s'k} + e_{sk}$ is also in $EQ(ax \leq \alpha)$, since S is a minimal cover for k. Thus, $bx^s = b(x^s - e_{s'k} + e_{sk})$, and therefore, $b_{s'k} = b_{sk}$ for all $s' \in S$, and therefore, there exists a constant c_1 such that $b_{sk} = c_1$ for all $s \in S$.

• (b) $b_{sj} = 0$ for all $s \in S$, $j \in M \setminus \{k, l\}$.

Consider x^s as defined above and observe that $x^s + e_{sj}$ is in $EQ(ax \leq \alpha)$ for all $j \in M \setminus \{k, l\}$. Then, $bx^s = b(x^s + e_{sj})$, and therefore, $b_{sj} = 0$ for all $j \in M \setminus \{k, l\}$.

Since that the same argument can be repeated for all $s \in S$, $b_{sj} = 0$ for all $s \in S$, $j \in M \setminus \{k, l\}$.

• (c) $b_{gl} = c_2$ for all $g \in G$.

Now, let $g \in \tilde{G}$ and $h \in G \setminus \tilde{G}$ (there exists some, since $|\tilde{G}| < |G|$). Observe that $x^s - e_{gl} + e_{hl}$ is in $EQ(ax \le \alpha)$, since G is a $(|G| - |S| + 1)$-cover. Thus, $bx^s = b(x^s - e_{gl} + e_{hl})$, and therefore, $b_{gl} = b_{hl}$. Using the same argument for different choices of g and h one can conclude that there exists some constant c_2 such that $b_{gl} = c_2$ for all $g \in G$.

• (d) $b_{sl} = c_2$ for all $s \in S$.

By the second condition of the theorem, for all $s \in S$ there exists some $G' \subseteq G$, $|G'| = |S| - 2$ such that $\sum_{i \in G'} f_i + f_s \le F_l$. Define the following vector $x^1 = (x^1_{ij})$.

$$x^1_{ij} := \begin{cases} 1, & i \in S \setminus \{s\}, j = k; \\ 1, & i \in G' \cup \{s\}, j = l; \\ 0, & \text{otherwise.} \end{cases}$$

The vector x^1 is an element of $EQ(ax \le \alpha)$. Now, let $g \in G \setminus G'$ and observe that $x^1 - e_{sl} + e_{gl}$ is also in $EQ(ax \le \alpha)$ (since G is a $(|G| - |S| + 1)$-cover). Thus, $bx^1 = b(x^1 - e_{sl} + e_{gl})$, and therefore, $b_{sl} = b_{gl}$ for all $g \in G$. Since the same construction can be repeated for all $s \in S$, we conclude that $b_{sl} = b_{gl} = c_2$ for all $s \in S$, $g \in G$.

• (e) $b_{gj} = 0$ for all $j \in M \setminus \{l\}$.

Consider now the following vector x^2.

$$x^2_{ij} := \begin{cases} 1, & i \in S, j = l; \\ 0, & \text{otherwise.} \end{cases}$$

The vector $x^2 \in EQ(ax \le \alpha)$. Moreover, for all $g \in G$, $x^2 + e_{gj}$ is also in $EQ(ax \le \alpha)$ for all $j \in M \setminus \{l\}$. Thus, $bx^2 = b(x^2 + e_{gj})$, and then, $b_{gj} = 0$ for all $g \in G$, $j \in M \setminus \{l\}$.

• $c_2 = (|S| - 1)c_1$

Since x^s and x^2 are elements of $EQ(ax \le \alpha)$, $bx^s = bx^2$, and then, $(|S| - 1)b_{sk} + (|S| - 1)b_{gl} = |S|b_{sl}$. Combining with the observations derived above we conclude $c_2 = (|S| - 1)c_1$.

Then, we conclude that the inequality $bx \le \beta$ is a scalar multiple of $ax \le \alpha$, and thus it defines a facet of $MKP((S \cup G) \times M, f, F)$.

We prove now that the conditions are also necessary. To this end we assume that $EQ(ax \le \alpha)$ is a facet defining inequality for $MKP((S \cup G) \times M, f, F)$. Then it is clearly valid. Now, suppose that there exists an item $s \in S$ such that for all subsets $\tilde{G} \subseteq G$, $|\tilde{G}| = |S| - 2$, $\sum_{i \in \tilde{G}} f_i + f_s > F_l$ holds. Under these assumption we prove that $\sum_{i \in G \cup S \setminus \{s\}} x'_{il} = |S| - 1$ for all $x' \in EQ(ax \le \alpha)$. Let x' be an element of $EQ(ax \le \alpha)$. First, observe that the inequality $\sum_{i \in G \cup S \setminus \{s\}} x'_{il} \le |S| - 1$ is valid for all $x' \in MKP((S \cup G) \times M, f, F)$. Now suppose that x' satisfies $\sum_{i \in G \cup S \setminus \{s\}} x'_{il} < |S| - 1$. Let us analyze the two cases.

Case 1. $x'_{sl} = 0$.
In this case, $\sum_{i\in S} x'_{ik} + \sum_{i\in G\cup S}(|S|-1)x'_{il} \leq |S|-1+(|S|-1)(|S|-2) < \alpha$. It contradicts our assumption that $x' \in EQ(ax \leq \alpha)$.

Case 2. $x'_{sl} = 1$.
We have again two cases to analyze.

Case 2.1. $\sum_{i\in S\setminus\{s\}} x'_{il} = t > 0$.
In this case, if the inequality $\sum_{i\in G\cup S\setminus\{s\}} x'_{il} < |S|-1$ holds, then $\sum_{i\in S} x'_{ik}+\sum_{i\in G\cup S}(|S|-1)x'_{il} \leq |S|-(t+1)+(|S|-1)(|S|-1) = |S|(|S|-1)-t < \alpha$. It contradicts our assumption that $x' \in EQ(ax \leq \alpha)$.

Case 2.2. $\sum_{i\in S\setminus\{s\}} x'_{il} = 0$.
Then, due to our assumption, $\sum_{i\in G} x'_{il} < |S|-2$. Then, $\sum_{i\in S} x'_{ik} + \sum_{i\in G\cup S}(|S|-1)x'_{il} \leq |S|-1+(|S|-1)(|S|-2) < \alpha$, contradicting our assumption that $x' \in EQ(ax \leq \alpha)$.

Observing all cases above, we can conclude that $EQ(ax \leq \alpha)$ must be included in $\{x \in \mathbb{R}^A \mid \sum_{i\in G\cup S\setminus\{s\}} x_{il} = |S|-1\}$, where $A = (S \cup G) \times M$. Thus, the inequality cannot define a facet of the polytope. ■

Example 4.6.3 Consider the following instance of the MKP: $n = 7$, $m = 2$,

$$\begin{aligned} 4x_{1,1} + 5x_{2,1} + 7x_{3,1} + 8x_{4,1} + 8x_{5,1} + 8x_{6,1} + 8x_{7,1} &\leq 16 \\ 4x_{1,2} + 5x_{2,2} + 7x_{3,2} + 8x_{4,2} + 8x_{5,2} + 8x_{6,2} + 8x_{7,2} &\leq 14. \end{aligned}$$

The Heterogeneous Two Cover Inequality corresponding to $S = \{1,2,3\}$, $G = \{4,5,6,7\}$, $k = 2$ and $l = 1$ is $x_{1,2} + x_{2,2} + x_{3,2} + 2x_{1,1} + 2x_{2,1} + 2x_{3,1} + 2x_{4,1} + 2x_{5,1} + 2x_{6,1} + 2x_{7,1} \leq 6$ and it defines a facet of $MKP(N, M, f, F)$.

If we change the weight of item 3 to $f_3 = 9$, the inequality remains valid, but it is not facet defining. By changing the weight of item 4 to $f_4 = 7$ the corresponding inequality is not even valid any more. ■

4.7 Multiple Cover Inequalities

In [W90], it was observed that given a set $S \subseteq N$ and a set $J \subseteq M$ with $\sum_{i\in S} f_i > \sum_{k\in J} F_k$, the inequality,

$$\sum_{j\in J}\sum_{i\in S} x_{ij} \leq |S|-1$$

is valid for the polytope $MKP(N \times M, f, F)$. If $|J| \geq 2$, a set of items S with the property $\sum_{i\in S} f_i > \sum_{k\in J} F_k$ is called a *multiple cover* with respect to J and the *Multiple Cover Inequality*, as defined above, corresponding to S and J. A set of items S is called a *minimal multiple cover* with respect to J, if $\sum_{i\in S} f_i > \sum_{k\in J} F_k$

and for all $s \in S$ there exists a valid assignment of $S \setminus \{s\}$ to the knapsacks in J. The Multiple Cover Inequality does not always define a facet of $MKP(S \times J, f, F)$, as the following example shows.

Example 4.7.1 Given the following instance of the MKP, where $n = 5$, $m = 2$,

$$\begin{aligned} 3x_{1,1} + 4x_{2,1} + 5x_{3,1} + 5x_{4,1} + 7x_{5,1} &\leq 8 \\ 3x_{1,2} + 4x_{2,2} + 5x_{3,2} + 5x_{4,2} + 7x_{5,2} &\leq 7 \end{aligned}$$

Let $S = \{2, 4, 5\}$ and $J = \{1, 2\}$. S is a minimal multiple cover for J. The corresponding inequality $x_{2,1} + x_{2,2} + x_{4,1} + x_{4,2} + x_{5,1} + x_{5,2} \leq 2$ is clearly valid for the $MKP(N \times M, f, F)$, but does not define a facet, since it is the sum of the two valid inequalities $x_{2,1} + x_{4,1} + x_{5,1} \leq 1$ and $x_{2,2} + x_{4,2} + x_{5,2} \leq 1$. ■

In the remainder of this section we focus on necessary and sufficient conditions such that the Multiple Cover Inequality is facet defining. Before treating the general case we elucidate the conditions for the special case of the MKP where the knapsack capacity values are all equal. More formally, let (MUKP) (*Multiple Uniform Knapsack Problem*) denote all instances of MKP such that $F_k = F$ for all $k \in M$. Given $A \subseteq N$ and $B \subseteq M$ we define $MUK(A \times B, f, F)$ as the corresponding polytope. Given $x \in MUK(A \times B, f, F)$, define $B_k(x) := \{i \in N \mid x_{ik} = 1\}$.

Lemma 4.7.2 *Given an instance (N, M, f, F) of the (MUKP). Let $J \subseteq M$ and S be a minimal multiple cover for J. Then, the Multiple Cover Inequality defines a facet of $MUK(S \times J, f, F)$ if and only if there exists an item $i \in S$ and a valid assignment x' of all the items in $S \setminus \{i\}$ to the knapsacks in J such that $|B_k(x')| \neq |B_l(x')|$ for some $k, l \in J, k \neq l$.*

Proof. We will first prove that the condition is sufficient. We represent the Multiple Cover Inequality corresponding to S and J as $ax \leq \alpha$. The inequality $ax \leq \alpha$ is clearly valid. Let us prove that it defines a facet of $MUK(S \times J, f, F)$. Suppose that $bx \leq \beta$ defines a facet of $MUK(S \times J, f, F)$ such that $EQ(ax \leq \alpha) \subseteq EQ(bx \leq \beta)$. Let $i_0 \in N$ be such that $f_{i_0} = \min\{f_i \mid i \in N\}$, and let x^1 denote a valid assignment of all the items in $S \setminus \{i_0\}$ to the knapsacks in J. Obviously, x^1 is in $EQ(ax \leq \alpha)$. Also, notice that for all $k \in J$ and $i \in B_k$, the vector $x - e_{ik} + e_{i_0k}$ is an element of $EQ(ax \leq \alpha)$. Thus, $bx^1 = b(x^1 - e_{ik} + e_{i_0k})$, providing $b_{ik} = b_{i_0k}$ for all $i \in B_k$. Moreover, since the capacities of the knapsacks are all equal, we can exchange the items of every pair of knapsacks and repeat the same arguments as above. Summing up, we conclude that for every $k \in J$ there exists a constant c_k such that $b_{ik} = c_k$ for all $i \in S$.

In order to prove that $c_k = c_l$ for all $k, l \in J$, let i be an item in S and let x' denote a valid assignment of all items in $S \setminus \{i\}$ to the knapsacks in J such that $|B_k(x')| \neq |B_l(x')|$ for some $k, l \in J, k \neq l$ as required in the condition of the theorem. Since all knapsacks have the same capacity, we can construct a valid assignment $x'' = (x''_{ij})$ via:

$$x''_{ij} := \begin{cases} x'_{il}, & \text{for all } i \in S, j = k; \\ x'_{ik}, & \text{for all } i \in S, j = l; \\ x'_{ij}, & \text{otherwise.} \end{cases}$$

Clearly, x' and x'' belong to the face $EQ(ax \leq \alpha)$. Thus, $bx' = bx''$, yielding

$$|B_k(x')|c_k + |B_l(x')|c_l = |B_k(x')|c_l + |B_l(x')|c_k.$$

This implies $c_k = c_l$. Due to the uniform knapsack capacities, we can apply this construction for all others knapsacks and, finally obtain that $bx \leq \beta$ is a scalar multiple of $ax \leq \alpha$ which completes the first part of the proof.

It remains to be shown that the condition is also necessary. Suppose it is not satisfied, i.e., for all $x' \in EQ(ax \leq \alpha)$ and $k, l \in J, k \neq l$, $|B_k(x')| = |B_l(x')|$ holds. In this case, all $x \in EQ(ax \leq \alpha)$ satisfy the equation $\sum_{i \in S} x_{ik} = \frac{|S|-1}{|J|}$, for all $k \in J$. Thus, the inequality cannot be facet defining. ■

In the remaining part of this subsection we will treat the general case where arbitrary knapsack capacities are given. Unfortunately, it will turn out that necessary and sufficient conditions for the Multiple Cover Inequality to define a facet are rather complicated and involve many (probably) unavoidable technicalities.

First, let us assume that for every $i \in N$ and $k \in M$ there exists a vector $x \in EQ(\sum_{i \in S} \sum_{j \in J} x_{ij} \leq |S| - 1)$ such that $x_{ik} = 1$. Otherwise $EQ(\sum_{i \in S} \sum_{j \in J} x_{ij} \leq |S| - 1)$ is a subset of $\{x \in \mathbb{R}^{S \times J} \mid x_{ik} = 0\}$, which implies that $\sum_{i \in S} \sum_{j \in J} x_{ij} \leq |S| - 1$ cannot define a facet.

For the exposition of the next lemma let us further assume that the items are ordered such that $f_1 \leq \ldots \leq f_{|S|}$. In the following we use some more definitions. Given a subset X of some vector space, define

$$\text{diff}(X) := \{x - y \mid x, y \in X\}.$$

to be the difference set of X. For a valid assignment $x' \in MKP(S \times J, f, F)$ we define

$$j(x', k) := \max\{j \mid j \in B_k(x')\}.$$

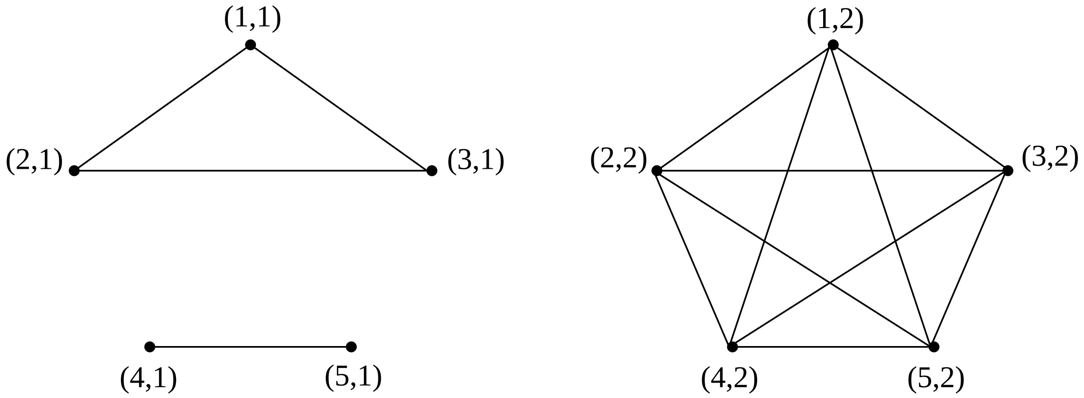
For a given $i \in S$ we set $j_i^k := \max\{j(x', k) \mid x'$ is a valid assignment of all elements in $S \setminus \{i\}$ to the knapsacks in $J\}$.

For $S \subseteq N$, $J \subseteq M$, let us define the *exchange-graph* $G = (V, E)$ in the following way. The node set V corresponds to the set $S \times J$ and two nodes (i_1, k_1) and (i_2, k_2), $i_1 < i_2$, are adjacent when $k_1 = k_2$ and $j_{i_1}^{k_1} \geq i_2$. Let d denote the number of connected components of G (note that $d \geq |J|$). Let (V_l, E_l), $l = 1, \ldots, d$ denote these components.

Example 4.7.3 Consider the following instance of the MKP. The set $S = \{1, 2, 3, 4, 5\}$ is a minimal multiple cover for $J = \{1, 2\}$. The corresponding exchange-graph has 3 components and is shown below.

$$\begin{aligned} 5x_{1,1} + 5x_{2,1} + 5x_{3,1} + 8x_{4,1} + 9x_{5,1} \leq & \ 10 \\ 5x_{1,2} + 5x_{2,2} + 5x_{3,2} + 8x_{4,2} + 9x_{5,2} \leq & \ 17 \end{aligned}$$

i	1	2	3	4	5
j_i^1	3	3	2	5	4
j_i^2	5	5	5	5	4



Lemma 4.7.4 *For every component l in the exchange graph G, there exist $|V_l| - 1$ linearly independent vectors belonging to* $\text{lin}(\text{diff}(EQ(\sum_{i \in S} \sum_{j \in J} x_{ij} \leq |S| - 1)))$. *Moreover, every nonzero component of any of these vectors is a nonzero component of χ^{V_l}.*

Proof. Let $l \in \{1, \dots, d\}$ and T_l be a spanning tree in $G_l = (V_l, E_l)$. Let $\{(i_1, k), (i_2, k)\}$ be an edge in the tree T_l. W. l. o. g. we assume that $i_1 < i_2$. Thus, $j_{i_1}^k \geq i_2$. Let x' be a valid assignment of $S \setminus \{i_1\}$ to the knapsacks in J such that $\sum_{i \in S} \sum_{j \in J} x'_{ij} = |S| - 1$ and $j_{i_1}^k \in B_k$ (there exists some by definition of $j_{i_1}^k$). We distinguish the following two cases.

- Case 1. $i_2 \in B_k(x')$.
  Since $f_{i_1} \leq f_{i_2}$, the vector $x'' := x' - e_{i_2 k} + e_{i_1 k}$ also satisfies $\sum_{i \in S} \sum_{j \in J} x''_{ij} = |S| - 1$. Thus, $x'' - x' = e_{i_1 k} - e_{i_2 k}$ is an element of $\text{lin}(\text{diff}(EQ(\sum_{i \in S} \sum_{j \in J} x_{ij} \leq |S| - 1)))$.

- Case 2. $i_2 \notin B_k(x')$.
  Suppose i_2 is assigned to knapsack l. Since $f_{i_1} \leq f_{i_2} \leq f_{j_{i_1}^k}$ we obtain that $x^2 = x' - e_{j_{i_1}^k k} + e_{i_1 k}$, $x^3 = x' - e_{i_2 l} + e_{i_2 k} + e_{i_1 l} - e_{j_{i_1}^k k}$ and $x^4 = x' - e_{i_2 l} + e_{i_1 l}$ are valid assignments with $x^2, x^3, x^4 \in EQ(\sum_{i \in S} \sum_{j \in J} x_{ij} \leq |S| - 1)$. Thus, the vector $(x^2 - x') - ((x^3 - x') - (x^4 - x')) = -e_{j_{i_1}^k k} + e_{i_1 k} + e_{i_2 l} - e_{i_2 k} - e_{i_1 l} + e_{j_{i_1}^k k} - e_{i_2 l} + e_{i_1 l} = e_{i_1 k} - e_{i_2 k}$ is an element of $\text{lin}(\text{diff}(EQ(\sum_{i \in S} \sum_{j \in J} x_{ij} \leq |S| - 1)))$.

So, we conclude that for each edge $\{(i_1,k),(i_2,k)\}$ of the tree the vector $e_{i_1k} - e_{i_2k}$ is in $\mathrm{lin}(\mathrm{diff}(EQ(\sum_{i\in S}\sum_{j\in J} x_{ij} \leq |S|-1)))$. Obviously, the nonzero components of these vectors are nonzero components of χ^{V_l}. Since T_l is a tree, it is not difficult to see that this set of vectors is linearly independent. ■

Consider now any of the components $G_l = (V_l, E_l)$, $l \in \{1,\dots,d\}$ and let T_l be the spanning tree in G_l used in the proof of Lemma 4.7.4. Furthermore, let t_l be any node of V_l. From the proof of Lemma 4.7.4 we know that for every edge $e = uv$ of the tree the vector $e_u - e_v$ is an element of $\mathrm{lin}(\mathrm{diff}(EQ(\sum_{i\in S}\sum_{j\in J} x_{ij} \leq |S|-1)))$. For a node $w \in V_l$ denote by $P(w)$ the unique path in T_l from t_l to w. For $w \in V_l \setminus \{t_l\}$ set $\mu_w = \sum_{uv\in P(w)} \sigma(uv)\,(e_u - e_v)$ where $\sigma(uv) = 1$, if $|P(u)| < |P(v)|$, and $\sigma(uv) = -1$, otherwise. Obviously, μ_w is an element of $\mathrm{lin}(\mathrm{diff}(EQ(\sum_{i\in S}\sum_{j\in J} x_{ij} \leq |S|-1)))$ and $(\mu_w)_{ik} = 1$, if $(i,k) = t_l$, $(\mu_w)_{ik} = -1$, if $ik = w$, and $(\mu_w)_{ik} = 0$, otherwise. Denote by $D = \{\mu_w \mid w \in V_l \setminus \{t_l\}$ for some $l \in \{1,\dots,d\}$ $\}$. Lemma 4.7.4 implies that D is a set of linearly independent vectors.

For a vector $x \in \mathbb{R}^{S\times J}$ we will frequently use the abbreviation $(x)^l \in \mathbb{R}^{V_l}$ to denote the subvector of x corresponding to the components $(i,k) \in V_l$. For an assignment x' we define the *cardinality vector* $g(x') \in \mathbb{N}^d$ by

$$g_l(x') := |\{(i,k) \in V_l \mid x'_{ik} = 1\}|, \text{ for } l = 1,\dots,d.$$

Let $\{x^1,\dots,x^c\}$ be a maximal set of valid assignments such that $x^i \in EQ(\sum_{i\in S}\sum_{j\in J} x_{ij} \leq |S|-1)$ for $i = 1,\dots,c$ and $\{g(x^i) \mid i = 1,\dots,c\}$ is linearly independent. Set $D' := D \cup \{x^i - x^1 \mid i = 2,\dots,c\}$. Obviously, each vector $x^i - x^1$, $i \in \{2,\dots,c\}$ is linearly independent from D, since $g(\mu) = 0$ for each $\mu \in D$. Thus, D' is a set of $|S||J| - d + c - 1$ linearly independent vectors. In the next theorem we show that the set D' is a basis of $\mathrm{lin}(\mathrm{diff}(EQ(\sum_{i\in S}\sum_{j\in J} x_{ij} \leq |S|-1)))$. As a corollary we obtain that the multiple cover inequality defines a facet if and only if $c = d$.

Theorem 4.7.5 *D' is a basis of* $\mathrm{lin}(\mathrm{diff}(EQ(\sum_{i\in S}\sum_{j\in J} x_{ij} \leq |S|-1)))$.

Proof. Let $x, y \in EQ(\sum_{i\in S}\sum_{j\in J} x_{ij} \leq |S|-1)$ be given. Let $T := \{t_i \mid i = 1,\dots,l\}$. Set

$$z^i = x^i - x^1 - \sum_{\{v\in V\setminus T \mid (x^i - x^1)_v = -1\}} \mu_v + \sum_{\{v\in V\setminus T \mid (x^i-x^1)_v = 1\}} \mu_v.$$

for $i = 2,\dots,c$. It easy to see that each of the vectors z^i is of the form

$$(z^i)^l_w = \begin{cases} g_l(x^i) - g_l(x^1), & \text{if } w = t_l, \\ 0, & \text{otherwise.} \end{cases}$$

In the same way we obtain a vector τ by setting

$$\tau := x - y - \sum_{\{v\in V\setminus T \mid (x-y)_v = -1\}} \mu_v + \sum_{\{v\in V\setminus T \mid (x-y)_v = 1\}} \mu_v$$

where τ is of the form

$$\tau_w = \begin{cases} g_l(x) - g_l(y), & \text{if } w = t_l, \\ 0, & \text{otherwise.} \end{cases}$$

Due to the choice of $x^1, \dots, x^c$ there exist $\lambda_1, \dots, \lambda_c$ such that

$$\tau = \sum_{i=1}^{c} \lambda_i z^i.$$

This implies that D' is a basis of $\text{lin}(\text{diff}(EQ(\sum_{i \in S} \sum_{j \in J} x_{ij} \leq |S| - 1)))$. ■

Corollary 4.7.6 *Given an instance (N, M, f, F) of the MKP. Let $J \subseteq M$ and $S \subseteq N$ be a minimal multiple cover for J. The corresponding Multiple Cover Inequality defines a facet of $MKP(S \times J, f, F)$ if and only if there exist valid assignments $x^1, \dots, x^d$ such that $\{g(x^i) \mid i = 1, \dots, d\}$ is linearly independent, where d is the number of components of the exchange graph.* ■

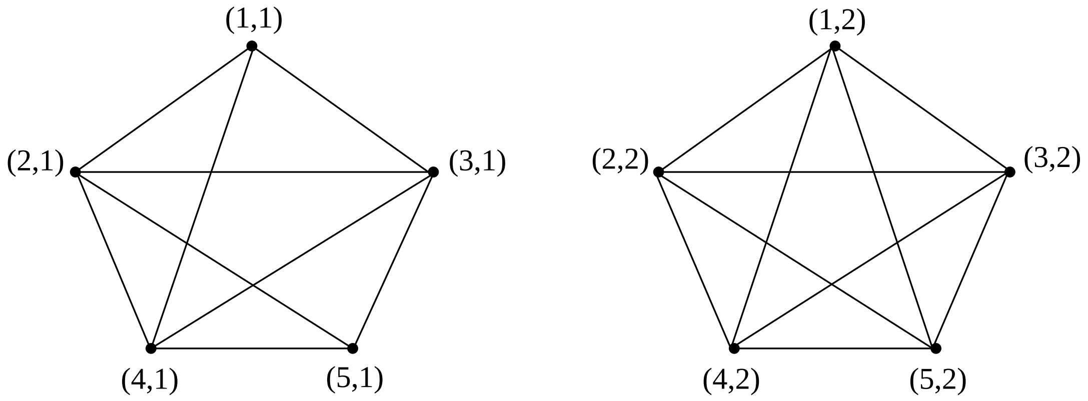
Let us now comment the previous results. Lemma 4.7.4 states that the dimension of the face $EQ(\sum_{i \in S} \sum_{j \in J} x_{ij})$ is at least $|S||J| - d$. Theorem 5.7.5 guarantees that, if we are able to find b valid assignments of $EQ(\sum_{i \in S} \sum_{j \in J} x_{ij})$ whose cardinality vectors are linearly independent, the dimension of the face is at least $|S||J| - d + b$. Hence, d valid assignments with linearly independent cardinality vectors suffice to show that the inequality is facet defining. The task (to find these vectors) is still nontrivial, yet "easier" than the original one, namely, finding $S||J| + 1$ affinely independent vectors on the face.

Example 4.7.7 In Example 4.7.3 one can check that there do not exist d linearly independent cardinality vectors. By applying Corollary 5.7.6 we can conclude that the multiple cover inequality does not define a facet of the corresponding polytope. ■

Example 4.7.8 Consider the following instance of the multiple knapsack problem. The set $S = \{1, 2, 3, 4, 5\}$ is a minimal multiple cover for $J = \{1, 2\}$.

$$\begin{aligned} x_{1,1} + 2x_{2,1} + 2x_{3,1} + 2x_{4,1} + 3x_{5,1} &\leq 4 \\ x_{1,2} + 2x_{2,2} + 2x_{3,2} + 2x_{4,2} + 3x_{5,2} &\leq 5 \end{aligned}$$

i	1	2	3	4	5
j_i^1	4	5	5	5	4
j_i^2	5	5	5	5	4



In this example, the cardinality vectors $(2,2)$ and $(1,3)$ are linearly independent. By applying Corollary 5.7.6 we can conclude that the inequality $\sum_{i=1}^{5}\sum_{j=1}^{2} x_{ij} \leq 4$ defines a facet of the polytope $MKP(S \times J, f, F)$. ■

4.8 Further Joint Inequalities

We were able to find some more classes of facet defining inequalities for the Multiple Knapsack Polytope, especially in the case where there are many items with the same weight. These inequality classes, however, seem to be very particular, and we could not provide exact or even heuristic separation procedures for them. Nevertheless, in this section we present some of our results. We omit the proofs in this section, since they are involved and these classes are very particular.

Theorem 4.8.1 *Given an instance of the MKP. Let $j_1, j_2, j_3, j_4 \in N$, such that $f_{j_1} \leq f_{j_2} \leq f_{j_3} \leq f_{j_4}$, and $k_1, k_2, k_3 \in M$. Furthermore, define $H := I_{j_1} \cup I_{j_2} \cup I_{j_3} \cup I_{j_4}$, and suppose that $|I_{j_3}| \geq 2$ and $|I_{j_4}| \geq 2$. Then the inequality,*

$$\sum_{l=1}^{3} x_{j_1 k_l} + \sum_{l=1}^{3} x_{j_2 k_l} + \sum_{l=1}^{2} \sum_{i \in I_{j_3}} x_{i k_l} + x_{j_3 k_3} + \sum_{l=1}^{3} \sum_{i \in I_{j_4}} x_{i k_l} \leq 5$$

defines a facet of $MKP(H \times M, f, F)$ if, for the first knapsack

$f_{j_1} + f_{j_2} \leq F_{k_1}$,
$f_{j_2} + f_{j_3} > F_{k_1}$,

for the second

$f_{j_2} + f_{j_3} \leq F_{k_2}$,
$f_{j_1} + 2f_{j_2} \leq F_{k_2}$,
$f_{j_2} + f_{j_4} > F_{k_2}$,
$2f_{j_3} > F_{k_2}$,

and for the third one

$f_{j_3} + f_{j_4} \leq F_{k_3}$,
$f_{j_2} + 2f_{j_3} \leq F_{k_3}$,
$2f_{j_4} > F_{k_3}$.

■

Example 4.8.2 Consider the following instance of the MKP. Let $m = 3$, $n = 6$,

$$\begin{aligned} 2x_{1,1} + 3x_{2,1} + 9x_{3,1} + 9x_{4,1} + 11x_{5,1} + 11x_{6,1} &\leq 11, \\ 2x_{1,2} + 3x_{2,2} + 9x_{3,2} + 9x_{4,2} + 11x_{5,2} + 11x_{6,2} &\leq 13, \\ 2x_{1,3} + 3x_{2,3} + 9x_{3,3} + 9x_{4,3} + 11x_{5,3} + 11x_{6,3} &\leq 21. \end{aligned}$$

Let $j_1 = 1$, $j_2 = 2$, $I_{j_3} = \{3, 4\}$ and $I_{j_4} = \{5, 6\}$. The inequalities

$$\sum_{k=1}^{3} x_{1,k} + \sum_{i=1}^{3} x_{2,k} + \sum_{k=1}^{2} (x_{3,k} + x_{4,k}) + x_{3,3} + \sum_{k=1}^{3} (x_{5,k} + x_{6,k}) \leq 5,$$

$$\sum_{k=1}^{3} x_{1,k} + \sum_{i=1}^{3} x_{2,k} + \sum_{k=1}^{2} (x_{3,k} + x_{4,k}) + x_{4,3} + \sum_{k=1}^{3} (x_{5,k} + x_{6,k}) \leq 5,$$

define facets of $MKP(N \times M, f, F)$. ■

Theorem 4.8.3 *Suppose given an instance (N, M, f, F) of the MKP. Let $S \subseteq N$, $j \in N \setminus S$ and $k_1, k_2 \in M$. Define $H := S \cup I_j$, $W := \{k_1, k_2\}$.*

Then the inequalities,

$$\sum_{i \in S \cup I_j} (|S| - 2) x_{ik_1} + \sum_{i \in S} x_{ik_2} + x_{jk_2} \leq (|S| - 1)^2$$

define facets of $MKP(H \times M, f, F)$ for all $j \in I_j$ if, for the first knapsack

$\sum_{i \in S} f_i \leq F_1$,
$(|S|-1)f_j \leq F_1$,
$f_j + \sum_{i \in S \setminus \{s\}} f_i > F_1$, *for all* $s \in S$;

and for the second

$\sum_{i \in S} x_i > F_{k_2}$,
$\sum_{i \in S \setminus \{s_1, s_2\}} x_i + f_j \leq F_{k_2}$, *for all* $s_1, s_2 \in S$,
$2f_j \leq F_{k_2}$.

■

Example 4.8.4 Consider the following instance of the MKP. Let $m = 2$, $n = 5$,

$$\begin{aligned} 3x_{1,1} + 5x_{2,1} + 5x_{3,1} + 6x_{4,1} + 6x_{5,1} &\leq 13, \\ 3x_{1,2} + 5x_{2,2} + 5x_{3,2} + 6x_{4,2} + 6x_{5,2} &\leq 12. \end{aligned}$$

Let $S = \{1, 2, 3\}$ and $I_j = \{4, 5\}$. The inequalities

$$\sum_{i=1}^{5} x_{i,1} + \sum_{i \in S} x_{i,2} + x_{4,2} \leq 4,$$

$$\sum_{i=1}^{5} x_{i,1} + \sum_{i \in S} x_{i,2} + x_{5,2} \leq 4$$

define facets of $MKP(N \times M, f, F)$. ■

Theorem 4.8.5 *Suppose we are given an instance* (N, M, f, F) *of the MKP. Let* $S \subseteq N$, $j \in N \setminus S$, $|S| = |I_j| = a$ *and* $k_1, k_2 \in M$. *Define* $H := S \cup I_j$, $W := \{k_1, k_2\}$.

Then the inequalities,

$$\sum_{i \in S} (a-2) x_{ik_1} + \sum_{i \in I_j} (a-1) x_{ik_1} + \sum_{i \in I_j} x_{ik_2} \leq (a-1)^2 + 1$$

defines a facet of $MKP(H \times M, f, F)$ *if, for the first knapsack*

$\sum_{i \in S} f_i \leq F_1$,
$(|S|-1)f_j \leq F_1$,
$f_j + \sum_{i \in S \setminus s} f_i > F_1$, *for all* $s \in S$;

and for the second

$2f_j + f_s \leq F_{k_2}$, *for all* $s \in S$,
$3f_j > F_{k_2}$.

■

Example 4.8.6 Consider the following instance of the MKP. Let $m = 2$, $n = 8$,

$$\begin{array}{ll} 4x_{1,1} + 5x_{2,1} + 6x_{3,1} + 6x_{4,1} + 7x_{5,1} + 7x_{6,1} + 7x_{7,1} + 7x_{8,1} & \leq 21, \\ 4x_{1,2} + 5x_{2,2} + 6x_{3,2} + 6x_{4,2} + 7x_{5,2} + 7x_{6,2} + 7x_{7,2} + 7x_{8,2} & \leq 20. \end{array}$$

Let $S = \{1, 2, 3, 4\}$ and $I_j = \{5, 6, 7, 8\}$. The inequality

$$\sum_{i=1}^{4} 2x_{i,1} + \sum_{i=5}^{8} 3x_{i,1} + \sum_{i=5}^{8} x_{i,2} \leq 10$$

is facet defining for the polytope $MKP(N \times M, f, F)$. ■

4.9 Extending facet defining inequalities

In the previous sections we deal with particular classes of inequalities and focus on conditions for the inequalities to be valid or facet defining for $MKP(N \times M, f, F)$. Let us now present a general procedure that allows the extension of particular classes of inequalities. In this section we first state the theorems in full generality. To illustrate the procedure it is subsequently applied to particular examples. We observe that some of the classes presented in the last sections can be obtained as applications of this procedure. A further observation is that the conditions required for the application of the procedure are local in each knapsack. Then, it is not difficult to see that the same ideas apply for extending inequalities of the polytope associated with the GAP.

Let us define what we mean with *extending a facet defining inequality*. Suppose we are given an instance (N, M, f, F) of the MKP, sets $A_u \subseteq N, B_u \subseteq M$ for $u = 1, \ldots, t$, and suppose $ay \leq \alpha$ is a facet defining inequality for $MKP(\bigcup_{u=1}^{t} A_u \times B_u, f, F)$. We now choose sets $T_1, \ldots, T_r$ of mutually disjoint items of $N \setminus \bigcup_{u=1}^{t} A_u$ and $k_1, \ldots, k_r$ pairwise disjoint elements of $M \setminus \bigcup_{u=1}^{t} B_u$. Moreover, we define $\beta_v := \max\{|G| \mid G \subseteq T_v, \sum_{i \in G} f_i \leq F_{k_v}\}, v = 1, \ldots, r$. We call the inequality,

$$ax + \sum_{v=1}^{r} \sum_{i \in A \cup T_v} x_{ik_v} \leq \alpha + \sum_{v=1}^{r} \beta_v$$

the *Extension* of the inequality $ax \leq \alpha$ corresponding to T_{v_i} and k_{v_i}, $i = 1, \ldots, r$.

Theorem 4.9.1 *Let be given an instance (N, M, f, F) of the MKP, sets $A_u \subseteq N$, $B_u \subseteq M$ for $u = 1, \ldots, t$ (we set $A := \bigcup_{u=1}^{t} A_u \subseteq N$, $B := \bigcup_{u=1}^{t} B_u \subseteq M$) and an inequality $ay \leq \alpha$ that is facet defining for $MKP(\bigcup_{u=1}^{t} A_u \times B_u, f, F)$ and satisfies the following additional requirement:*

($\star$) For all $\tilde{A} \subseteq A, |\tilde{A}| \geq 2$ the following holds: every vector $y \in MKP(\bigcup_{u=1}^{t} A_u \times B_u, f, F)$ such that $y_{ik} = 0$ for all $i \in \tilde{A}$, $k \in \{l \in B \mid (i,l) \in \bigcup_{u=1}^{t} A_u \times B_u\}$, satisfies $ay \leq \alpha - |\tilde{A}| + 1$.

We choose a positive integer $r \leq \min\{|N \setminus A|, |M \setminus B|\}$, sets $T_1, \ldots, T_r$ that are mutually disjoint subsets of $N \setminus A$ and a subset $\{k_1, \ldots, k_r\}$ of $M \setminus B$.

Under these assumptions, the corresponding extension of $ay \leq \alpha$, namely,

$$ax + \sum_{v=1}^{r} \sum_{i \in A \cup T_v} x_{ik_v} \leq \alpha + \sum_{v=1}^{r} \beta_v$$

is valid for the polytope $MKP((\bigcup_{u=1}^{t} A_u \times B_u) \cup (\bigcup_{v=1}^{r} (T_v \cup A) \times \{k_v\}), f, F)$ if and only if $\tilde{T}_v \cup \{i\}$ is a cover with respect to k_v for all $i \in A$, $\tilde{T}_v \subseteq T_v$, $|\tilde{T}_v| = \beta_v$ and $v = 1, \ldots, r$.

Proof. Let us denote the Extension of $ay \leq \alpha$ by $bx \leq \beta$ and set $Q := \bigcup_{u=1}^{t} A_u \times B_u$.

Let us first prove that the condition is necessary. Suppose, there exists an index $v \in \{1, \ldots, r\}$, an item $i_0 \in A$ and a set $\tilde{T}_v \subseteq T_v$, $|\tilde{T}_v| = \beta_v$ such that $\tilde{T}_v \cup \{i_0\}$ is not a cover with respect to k_v. Let $\tilde{T}_w \subseteq T_w, w \in \{1, \ldots, r\} \setminus \{v\}$, $|\tilde{T}_w| = \beta_w$, $\sum_{i \in \tilde{T}_w} f_i \leq F_{k_w}$.

Since $ay \leq \alpha$ defines a facet of $MKP(Q, f, F)$ there exists a vector $y' \in EQ(MKP(Q, f, F), ax \leq \alpha)$ with $y'_{i_0 k} = 0$ for all $(i_0, k) \in Q$. Set $x' = (x'_{ik})$ as follows.

$$x'_{ik} := \begin{cases} y'_{ik}, & \text{for all } (i,k) \in Q; \\ 1, & \text{for all } i \in \tilde{T}_w, k = k_w, w \in \{1, \ldots, r\} \setminus \{v\}; \\ 1, & \text{for all } i \in \tilde{T}_v, k = k_v; \\ 0, & \text{otherwise.} \end{cases}$$

$\tilde{T}_v \cup \{i_0\}$ does not define a cover with respect to knapsack k_v. Thus, $x' + e_{i_0 k_v}$ is a valid assignment yielding $b(x' + e_{i_0 k_v}) = ay' + \sum_{w \in \{1, \ldots, r\} \setminus \{v\}} \beta_w + \beta_v + 1 > \alpha + \sum_{v=1}^{r} \beta_v$. This implies that the condition is necessary.

In order to prove the converse direction, let us assume that the inequality is not valid for the polytope $MKP(Q \cup \bigcup_{v=1}^{r} (T_v \cup A) \times \{k_v\}, f, F)$, i.e., there exists a vector $x'' \in MKP(Q \cup \bigcup_{v=1}^{r} (T_v \cup A) \times \{k_v\}, f, F)$ with $bx'' > \beta$. Let $y'' := x''|_Q$. Set $\tilde{T}_v := \{i \in T_v \mid x_{ik_v} = 1\}$ and $A_v := \{i \in A \mid x_{ik_v} = 1\}, v = 1, \ldots, r$. Since $y'' \in MKP(Q, f, F)$, then $ay'' \leq \alpha$, and therefore, there must exist a $v \in \{1, \ldots, r\}$ satisfying $|A_v \cup \tilde{T}_v| > \beta_v$. Let $V \subseteq \{1, \ldots, r\}$ denote the maximal subset of knapsacks with $|A_v \cup \tilde{T}_v| > \beta_v$ for all $v \in V$. Due to the condition, every subset of T_v of cardinality β_v and one element from A defines a cover with respect to k_v ($v \in V$). This implies, that $|\tilde{T}_v| \leq \beta_v - 1$. Thus $|A_v|$ must be greater than or equal to two, for all $v \in V$. Due to requirement ($\star$), we know that

$ay'' \leq \alpha - \sum_{v \in V} |A_v| + 1$. Summing up, we obtain the following relations:

$$\begin{aligned} bx &\leq \alpha - \textstyle\sum_{v \in V} |A_v| + 1 + \sum_{v \in \{1,\dots,r\} \setminus V} \beta_v + \sum_{v \in V} (|\tilde{T}_v| + |A_v|) \\ &\leq \alpha - \textstyle\sum_{v \in V} |A_v| + 1 + \sum_{v \in \{1,\dots,r\} \setminus V} \beta_v + \sum_{v \in V} (\beta_v + |A_v|) - |V| \\ &= \beta + 1 - |V| \leq \beta, \end{aligned}$$

contradicting the assumption that x' is a point violating the inequality. Thus, the inequality is valid, which completes the proof. ■

In the subsequent theorem necessary and sufficient conditions are given for the extension of an inequality to be facet defining.

Theorem 4.9.2 *Let be given an instance (N, M, f, F) of the MKP, sets $A_u \subseteq N$, $B_u \subseteq M$ for $u = 1, \dots, t$ (we set $A := \bigcup_{u=1}^t A_u \subseteq N$, $B := \bigcup_{u=1}^t B_u \subseteq M$) and an inequality $ay \leq \alpha$ that defines a facet of $MKP(\bigcup_{u=1}^t A_u \times B_u, f, F)$ and satisfies the following additional requirement:*

($\star$) *For all $\tilde{A} \subseteq A$, $|\tilde{A}| \geq 2$ the following holds: every vector $y \in MKP((\bigcup_{u=1}^t A_u \times B_u, f, F)$ such that $y_{ik} = 0$ for all $i \in \tilde{A}$, $k \in \{l \in B \mid (i,l) \in \cup_{u=1}^t A_u \times B_u\}$, satisfies $ay \leq \alpha - |\tilde{A}| + 1$.*

We choose a positive integer $r \leq \min\{|N \setminus A|, |M \setminus B|\}$, sets $T_1, \dots, T_r$ that are mutually disjoint subsets of $N \setminus A$ and a subset $\{k_1, \dots, k_r\}$ of $M \setminus B$. We require that $T_v \cup \{i\}$ is a minimal cover with respect to knapsack k_v for all $i \in A$.

Under these assumptions, the corresponding extension of the inequality $ay \leq \alpha$, namely,

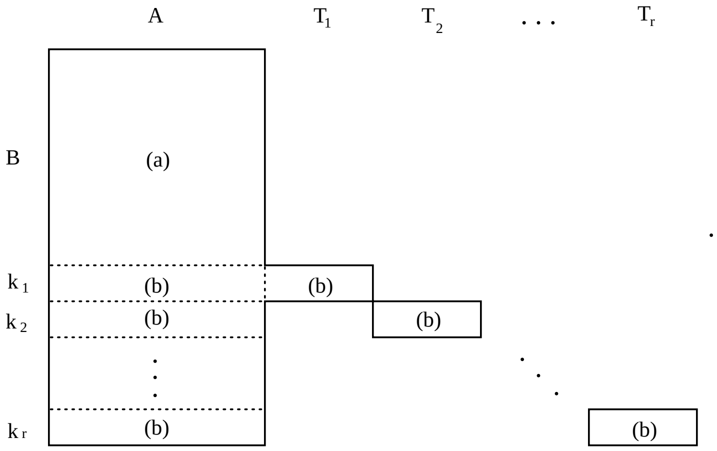
$$ax + \sum_{v=1}^r \sum_{i \in A \cup T_v} x_{ik_v} \leq \alpha + \sum_{v=1}^r |T_v|$$

defines a facet of $MKP((\bigcup_{u=1}^t A_u \times B_u) \cup (\bigcup_{u=1}^r (T_u \times A) \times \{k_v\}), f, F)$ if and only if for every $v \in \{1, \dots, r\}$ there exist a set $\tilde{A} \subseteq A$, $|\tilde{A}| \geq 2$, an item $t_v \in T_v$ and an assignment $y \in MKP(\bigcup_{i=1}^t A_i \times B_i, f, F)$ with $y_{ik} = 0$ for all $i \in \tilde{A}$, $k \in \{l \in B \mid (i,l) \in \bigcup_{u=1}^t A_u \times B_u\}$ such that $ay = \alpha - |\tilde{A}| + 1$ and $\sum_{i \in T_v \setminus \{t_v\} \cup \tilde{A}} f_i \leq F_{k_v}$.

Proof. We represent the extension of $ay \leq \alpha$ by $bx \leq \beta$ and set $Q := \bigcup_{u=1}^t A_u \times B_u$ and $\tilde{Q} := Q \cup \bigcup_{v=1}^r T_v \times \{k_1, \dots, k_r\}$. Due to Theorem 4.9.1, the inequality is valid if and only if $T_v \cup \{i\}$ defines a cover with respect to k_v, $v = 1, \dots, r$. Clearly, this requirement is satisfied and we can conclude that inequality $bx \leq \beta$ is valid. In the following we will show, that the inequality defines a facet of $MKP(\tilde{Q}, f, F)$ if and only if the conditions above are satisfied.

We start by proving that the conditions are sufficient. Suppose that there exists a facet defining inequality $cx \leq \gamma$ such that $EQ(MKP(\tilde{Q}, f, F), bx \leq \beta) \subseteq$

$EQ(MKP(\tilde{Q}, f, F), cx \le \gamma)$.



• (a) $c_{ik} = \lambda b_{ik}$ for all $(i, k) \in Q$.

Let $y' \in EQ(MKP(Q, f, F), ay \le \alpha)$. Consider the vector $x' = (x'_{ij})$ defined as follows.

$$x'_{ik} := \begin{cases} y'_{ik}, & \text{for all } (i, k) \in Q; \\ 1, & \text{for all } i \in T_w, k = k_w, w \in \{1, \ldots, r\}; \\ 0, & \text{otherwise.} \end{cases}$$

Obviously, x' is an element of $EQ(MKP(\tilde{Q}, f, F), bx \le \beta)$. Since the inequality $ay \le \alpha$ defines a facet of the polytope $MKP(Q, f, F)$, we conclude that $c_{ik} = \lambda b_{ik}$ for all $(i, k) \in Q$ for some $\lambda > 0$.

• (b) $c_{uk_v} = c_v$ for all $v = 1, \ldots, r$, $u \in A \cup T_v$.

Next, let $i_0 \in A$ and $y'' \in EQ(MKP(Q, f, F), ay \le \alpha)$ such that $y''_{i_0 k} = 0$ for all $k \in B$ (such a solution does exist, otherwise $EQ(MKP(Q, f, F), ay \le \alpha) \subseteq \{y \in \mathbb{R}^Q \mid \sum_{k \in B} y_{i_0 k} = 1\}$). Let $v \in \{1, \ldots, r\}$ be given. Consider the vector $x'' = (x''_{ik})$ defined as follows.

$$x''_{ik} := \begin{cases} y''_{ik}, & \text{for all } (i, k) \in Q; \\ 1, & \text{for all } i \in T_w, k = k_w, w \in \{1, \ldots, r\} \setminus \{v\}; \\ 1, & \text{for all } i \in T_v \setminus t \cup \{i_0\}, k = k_v; \\ 0, & \text{otherwise.} \end{cases}$$

Since $T_v \cup \{i_0\}$ is a minimal cover for k_v, x'' belongs to $EQ(MKP(\tilde{Q}, f, F), bx \le \beta)$ for all $t \in T_v$. Moreover, $x'' - e_{i_0 k_v} + e_{tk_v}$ is also in $EQ(MKP(\tilde{Q}, f, F), bx \le \beta)$. Thus, $cx'' = c(x'' - e_{i_0 k_v} + e_{tk_v})$ for all $t \in T_v$. Since the same argument can be repeated for all $i_0 \in A$ and all $v \in \{1, \ldots, r\}$, we conclude that for all $v = 1, \ldots, r$, there exists a constant c_v such that $c_{uk_v} = c_v$ for all $u \in A \cup T_v$.

• $c_v = \lambda$ for all $v = 1, \ldots, r$.

Choose any $v \in \{1, \ldots, r\}$. We know that, as required in the conditions of the theorem, there exist a set $\tilde{A} \subseteq A$, $|\tilde{A}| \ge 2$, an item $t_v \in T_v$ and a valid assignment

$y \in MKP(Q, f, F)$ with $y_{ik} = 0$ for all $i \in \tilde{A}$, $k \in B$ such that $ay = \alpha - |\tilde{A}| + 1$ and $\sum_{i \in T_v \setminus \{t_v\} \cup \tilde{A}} f_i \leq F_{k_v}$. We now define the vector $x = (x_{ik})$ as follows.

$$x_{ik} := \begin{cases} y_{ik}, & \text{for all } (i,k) \in Q; \\ 1, & \text{for all } i \in T_w, k = k_w, w \in \{1, \ldots, r\} \setminus \{v\}; \\ 1, & \text{for all } i \in T_v \setminus \{t_v\} \cup \tilde{A}, k = k_v; \\ 0, & \text{otherwise.} \end{cases}$$

Obviously,

$$bx = ay - |\tilde{A}| + 1 + \sum_{w \in \{1,\ldots,r\} \setminus \{v\}} |T_w| + |T_v| - 1 + |\tilde{A}| = \beta,$$

i.e., $x \in EQ(MKP(\tilde{Q}, f, F), bx \leq \beta)$. Now, observing that x' as defined above is also an element of $EQ(MKP(\tilde{Q}, f, F), bx \leq \beta)$, we obtain $cx = cx'$, and therefore,

$$\lambda(\alpha - |\tilde{A}| + 1) + \sum_{w \in \{1,\ldots,r\} \setminus \{v\}} c_w |T_w| + (|T_v| - 1 + |\tilde{A}|) c_v = \lambda\alpha + \sum_{w=1}^{r} c_w |T_w|,$$

or

$$0 = (c_v - \lambda)(|\tilde{A}| - 1).$$

Since $|\tilde{A}| \geq 2$, then it follows that $c_v = \lambda$. Thus, the inequalities $bx \leq \beta$ and $cx \leq \gamma$ are equal up to multiplication with a scalar.

Now we prove that the conditions are also necessary. Assume that the inequality is facet defining and the conditions are violated, i.e., there exist some $v \in \{1, \ldots, r\}$ such that, for all $\tilde{A} \subseteq A$, $|\tilde{A}| \geq 2$, the following holds: every $t_v \in T_v$ satisfies $\sum_{i \in T_v \setminus \{t_v\} \cup \tilde{A}} f_i > F_{k_v}$, or, for every assignment $y \in MKP(Q, f, F)$ with $y_{ik} = 0$ for all $i \in \tilde{A}$, $k \in B$ satisfies $ay < \alpha - |\tilde{A}| + 1$. In this case, we claim that $EQ(MKP(\tilde{Q}, f, F), bx \leq \beta) \subseteq \{x \in \mathbb{R}^{\tilde{Q}} \mid \sum_{i \in T_v \cup A} x_{ik_v} = |T_v|\}$, contradicting our assumption that the inequality is facet defining.

First we prove that under the assumptions above, $EQ(MKP(\tilde{Q}, f, F), bx \leq \beta) \subseteq \{x \in \mathbb{R}^{\tilde{Q}} \mid \sum_{i \in T_v \cup A} x_{ik_v} = |T_v|\}$. First, suppose that there exists some assignment $x \in EQ(MKP(\tilde{Q}, f, F), bx \leq \beta)$ with $\sum_{i \in T_v \cup A} x_{ik_v} < |T_v|$. Let $W := \{w \in \{1, \ldots, r\} \setminus \{v\} \mid |\{i \in T_w \cup A \mid x_{ik_w} = 1\}| > |T_w|\}$ and $W' := \{1, \ldots, r\} \setminus W$. Obviously, $W \neq \emptyset$, since $\sum_{i \in T_v \cup A} x_{ik_v} < |T_v|$. Set $\tilde{T_w} := \{i \in T_w \mid x_{ik_w} = 1\}$ and $\tilde{A_w} := \{i \in A \mid x_{ik_w} = 1\}$ for all $w \in W$. Since $T_w \cup \{i\}$ is a cover for all $i \in A$, we conclude that $|\tilde{A_w}| \geq 2$ and $\tilde{T_w} \leq |T_w| - 1$. Thus, we obtain that

$$\begin{aligned} bx &< \alpha - \textstyle\sum_{w \in W} |\tilde{A_w}| + 1 + \sum_{w \in W} (|\tilde{A_w}| + |\tilde{T_w}|) + \sum_{w' \in W' \setminus \{v\}} |T_{w'}| + |T_v| \\ &\leq \alpha + \textstyle\sum_{t=1}^{r} |T_t| + 1 - |W| \leq \beta. \end{aligned}$$

It contradicts the assumption that $x \in EQ(MKP(\tilde{Q}, f, F), bx \leq \beta)$. Thus, $\sum_{i \in T_v \cup A} x_{ik_v} \geq |T_v|$.

Now, suppose there exists an assignment $x \in EQ(MKP(\tilde{Q}, f, F), bx \leq \beta$ with $\sum_{i \in T_v \cup A} x_{ik_v} > |T_v|$. Again, define $W := \{w \in \{1, \ldots, r\} \setminus \{v\} \mid |\{i \in T_w \cup A \mid x_{ik_w} = 1\}| > |T_w|\}$, $W' := \{1, \ldots, r\} \setminus W$, $\tilde{T}_w := \{i \in T_w \mid x_{ik_w} = 1\}$ and $\tilde{A}_w := \{i \in A \mid x_{ik_w} = 1\}$ for all $w \in W$. Clearly, $v \in W$, yielding $|W| \geq 1$. If $|W| \geq 2$, we obtain

$$\begin{array}{rcl} bx & \leq & \alpha - |\cup_{w \in W} \tilde{A}_w| + 1 + \sum_{w \in W}(|\tilde{A}_w| + |\tilde{T}_w|) + \sum_{w' \in W'} |T_{w'}| \\ & \leq & \alpha + \sum_{w=1}^{r} |T_w| - |W| + 1 < \beta. \end{array}$$

It remains the case $W = \{v\}$. Since $T_v \cup \{i\}$ is a cover with respect to knapsack k_v for all $i \in A$, we have $|\tilde{A}_v| \geq 2$ and $|\tilde{T}_v| \leq |T_v| - 1$. Define $y \in MKP(Q, f, F)$ by setting $y := x|_Q$. Due to our assumption, we have $ay < \alpha - |\tilde{A}_v| + 1$ or $\sum_{i \in T_v \setminus \{t_v\} \cup \tilde{A}_v} f_i > F_{k_v}$ for all $t_v \in T_v$. In the first case, we obtain

$$\begin{array}{rcl} bx & < & \alpha - |\tilde{A}_v| + 1 + |\tilde{A}_v| + |\tilde{T}_v| + \sum_{w \in \{1, \ldots, r\} \setminus \{v\}} |T_w| \\ & \leq & \beta. \end{array}$$

This contradicts the assumption that $x \in EQ(MKP(\tilde{Q}, f, F), bx \leq \beta)$. In the second case, $\sum_{i \in T_v \setminus \{t_v\} \cup \tilde{A}_v} > F_{k_v}$ for all $t_v \in T_v$. We conclude that $|\tilde{A}_v| \geq 3$ and $|\tilde{T}_v| \leq |T_v| - 2$. Then,

$$\begin{array}{rcl} bx & \leq & \alpha - |\tilde{A}_v| + 1 + |\tilde{A}_v| + |\tilde{T}_v| + \sum_{w \in \{1, \ldots, r\} \setminus \{v\}} |T_w| \\ & \leq & \alpha + \sum_{w \in \{1, \ldots, r\} \setminus \{v\}} |T_w| + |T_v| - 2 + 1 \\ & < & \beta. \end{array}$$

This is also a contradiction to our assumption that $x \in EQ(MKP(\tilde{Q}, f, F), bx \leq \beta)$. It completes the proof. ■

In the following, we illustrate the application of the extension procedure on some examples.

Example 4.9.3 Given an instance (N, M, f, F) of the MKP. Let $S \subseteq N$ be a minimal cover with respect to some knapsack $k \in N$ and let $M' \subseteq M$ be a subset of knapsacks with $k \in M'$. Then, the inequality

$$\sum_{i \in S} x_{ik} \leq |S| - 1$$

defines a facet for the polytope $MKP(S \times M', f, F)$. Define $a := \sum_{i \in S} e_{ik}$ and $\alpha := |S| - 1$. Obviously, the inequality $ax \leq \alpha$ meets the requirement $(\star)$ of Theorem 5.9.2, where $\cup_{u=1}^{r}(A_u \times B_u) = S \times M$. Let us choose a positive integer $r \leq \min\{|N \setminus S|, |M \setminus M'|\}$, sets $T_1, \ldots, T_r$ that are mutually disjoint subsets of $N \setminus S$ and a subset $\{k_1, \ldots, k_r\}$ of $M \setminus M'$. Moreover, we require that $T_v \cup \{i\}$ is a minimal cover with respect to knapsack k_v for all $i \in A$. Due to Theorem 5.9.2, the corresponding extension of the Minimal Cover Inequality

$$\sum_{i \in S} x_{ik} + \sum_{v=1}^{r} \sum_{i \in S \cup T_v} x_{ik_v} \leq |S| - 1 + \sum_{v=1}^{r} |T_v|$$

defines a facet of the polytope $MKP((S \times M' \cup \bigcup_{v=1}^{r}(S \cup T_v) \times \{k_v\}, f, F)$ if and only if for every $v \in \{1, \ldots, r\}$ there exist $i, j \in S, i \neq j$ and an item $t \in T_v$ such that $\sum_{s \in T_v \setminus \{t\}} f_s + f_i + f_j \leq F_{k_v}$.

Moreover, it can be easily checked that the extended inequality still satisfies the requirement $(\star)$ of Theorem 5.9.2. Thus, a repeated extension in the "spirit" of Theorem 5.9.2 is possible. Observe also that the Extended Cover Inequality, presented in Section 5.3 can be obtained as an Extension of a Minimal Cover Inequality for some knapsack k corresponding to a set T and knapsack l. ■

Example 4.9.4 Given an instance (N, M, f, F) of the MKP. Let $\{i_1, \ldots, i_{n'}\} = N' \subseteq N$ be a set of items. Without loss of generality we assume that $f_{i_1} \leq \ldots \leq f_{i_{n'}}$. Set $I_{i_{n'}} = \{j \in N \mid f_j = f_{i_{n'}}\}$ and define $r_1 := \max\{\lambda \in \mathbb{N} \mid \lambda f_{i_{n'}} \leq F_1\}$. Assume, $|I_{i_{n'}}| \geq r_1$ and let T_1 be a subset of $I_{i_{n'}}$ of cardinality $|T_1| \geq r_1$. If we assume $T_1 \cup \{i_1, \ldots, i_{n'-1}\}$ to define a minimal cover with respect to knapsack 1, the inequality

$$\sum_{v=1}^{n'-1} x_{i_v 1} + \sum_{i \in T_1} x_{i1} \leq r_1 + n' - 2$$

defines a facet for the polytope $MKP((\{i_1, \ldots, i_{n'-1}\} \cup T_1) \times \{1\}, f, F)$.

Now, let us extend this inequality in the following way: for the ease of notation, set

$$r_l := \max\{\lambda \in \mathbb{N} \mid \lambda f_{i_{n'}} \leq F_l\}, l \in M \setminus \{1\}$$

and suppose, $|I_{i_{n'}}| \geq \sum_{l \in M} r_l$ holds.

Set $r := |M|$ and let $T_1, \ldots, T_r$ be a partition of $I_{i_{n'}}$ such that $|T_l| = r_l$ for $l = 2, \ldots, r$ and $T_1 = I_{i_{n'}} \setminus \bigcup_{l=2}^{|M|} T_l$. Using this notation, we obtain mutually disjoint sets T_l, $l = 1, \ldots, |M|$ such that $f(T_l) \leq F_l$ for $l = 2, \ldots, |M|$.

Under these assumptions, the inequality

$$\sum_{k \in M} (\sum_{v=1}^{n'-1} x_{i_v k} + \sum_{i \in T_1} x_{ik}) + \sum_{l=2}^{|M|} \sum_{i \in T_l} x_{il} \leq \sum_{k \in M} r_k + n' - 2$$

is valid for $MKP(N' \times M \cup (\bigcup_{u=2}^{r}(N' \setminus I_{i_{n'}} \times \{l\}), f, F)$ if and only if for every $l \in M \setminus \{1\}$, $T \cup \{i_1\}$ is a cover with respect to knapsack l, where $T \subseteq I_{i_{n'}}, |T| = r_l$ (note that i_1 is the element with smallest weight).

The inequality defines a facet of the polytope $MKP(N' \times M \cup (\bigcup_{u=2}^{r}(N' \setminus I_{i_{n'}} \times \{l\}), f, F)$ if and only if it is valid (note: $T \cup \{i_1\}$ is automatically a minimal cover with respect to knapsack k_l) and for every $l \in M \setminus \{1\}$, the condition $(r_l - 1) f_{i'_n} + f_{i_1} + f_{i_{n'-1}} \leq F_l$ holds (note, this condition is stronger than the one given in Theorem 4.9.2).

■

Example 4.9.5 Given an instance (N, M, f, F) of the MKP. Let $S \subseteq N$ be a minimal multiple cover with respect to a given set $J \subseteq M$ and suppose, the inequality

$$\sum_{j \in J} \sum_{i \in S} x_{ij} \leq |S| - 1$$

defines a facet for the polytope $MKP(S \times J, f, F)$ (cf. Section 5.7). Define $a := \sum_{j \in J} \sum_{i \in S} e_{ik}$ and $\alpha := |S| - 1$. Then, it is easy to verify that the inequality $ax \leq \alpha$ meets the requirement $(\star)$ of Theorem 5.9.2. Again, let us choose a positive integer $r \leq \min\{|N \setminus S|, |M \setminus J|\}$, sets $T_1, \ldots, T_r$ that are mutually disjoint subsets of $N \setminus S$ and a subset $\{k_1, \ldots, k_r\}$ of $M \setminus J$. Moreover, we require for all $v = 1, \ldots, r$ that $T_v \cup \{i\}$ is a minimal cover with respect to knapsack k_v for all $i \in A$.

By applying Theorem 5.9.2, we can conclude that the extension of the Minimal Multiple Cover Inequality,

$$\sum_{j \in J} \sum_{i \in S} x_{ij} + \sum_{v=1}^{r} \sum_{i \in S \cup T_v} x_{ik_v} \leq |S| - 1 + \sum_{v=1}^{r} |T_v|$$

defines a facet of the polytope $MKP((S \times J \cup \bigcup_{v=1}^{r} (S \cup T_v) \times \{k_v\}, f, F)$ if and only if for every $v \in \{1, \ldots, r\}$ there exist $i, j \in S, i \neq j$ and an item $t \in T_v$ such that $\sum_{s \in T_v \setminus \{t\}} f_s + f_i + f_j \leq F_{k_v}$.

In addition, the extended minimal multiple cover inequality still satisfies the requirements of Theorem 5.9.2. Thus an iterative extension of the original inequality is possible. ■

4.10 Concluding remarks

In this chapter we investigated the polytope associated with MKP. The only other attempt to study a similar problem from this point of view are the papers by Gottlieb and Rao [GR90] and [GR90a], where the authors investigate the polytope associated with GAP.

In this chapter, we presented several new classes of valid joint inequalities and discussed necessary and sufficient conditions under which these inequalities define facets of the Multiple Knapsack Polytope. Although some of these conditions involve many technicalities (e.g. in Section 5.7), we were able to derive separation heuristics for most classes of inequalities discussed here. Moreover, in Section 5.9 a procedure is presented to extend iteratively facet defining inequalities for the Multiple Knapsack Polytope. This procedure gives rise to a huge class of joint inequalities, which generalize, for instance, the Extended Cover Inequalities presented in Section 5.3.

The inequalities discussed in this chapter are used as a starting point for the development of a Branch and Cut Algorithm for the MKP, which is discussed in the next chapter.

Chapter 5

A Branch and Cut Algorithm for the Multiple Knapsack Problem

In this chapter we present an implementation of a Branch and Cut Algorithm for the Multiple Knapsack Problem. The general method is described in Section 6.1. Section 6.2 addresses the core of the method: the Separation Problem. In this section we show the separation strategies used in our approach for finding valid inequalities for the $MKP(N, M, f, F)$. Different separation procedures for several classes of valid and facet defining inequalities are suggested and a comparison of them is provided. Further, some details on the lifting procedure, and the so called complementing heuristic are given. In Section 6.3 we present some heuristic procedures based on LP techniques to the MKP. A comparison of the behavior of all these heuristics for a set of "difficult" problems is given. Section 6.4 shows the behavior of different branching strategies applied to some examples. Further implementation details are given in Section 6.5. Finally, in Section 6.6 we describe our computational results in solving small and large scale real world problems. The results in this chapter are also presented in [FMW93a].

We describe our approach to the minimization version of the MKP. Moreover, we ignore feasible solutions of the MKP that do not assign all the items. It is due to the fact that these solutions are not feasible to our original application. For more details on the original problem and its relationship with the MKP see Chapter 2.

We finish this introduction with some notes about the Branch and Cut method. The method derives from a Cutting Plane algorithm in combination with branch and bound and heuristic procedures. The name "Branch and Cut" appears the first time in [PR87], applied to solving large scale instances of the Travelling Salesman Problem. The method, as presented in this chapter, was introduced by Grötschel, Jünger and Reinelt in [GJR85] for solving Linear Order problems. The Cutting Plane method was developed at the end of the 50's by Gomory [Go58] to solve integer linear programs. Although theoretical interesting, the method proposed by Gomory turns out to be time and memory consuming, since the calculations should be made very precisely in order to guarantee correctness. Actually, the idea of adding inequalities to obtain a better bound of the linear relaxation of an integer program is older. Dantzig, Fulkerson and Johnson [DFJ54] have already used it to solve instances of the Travelling Salesman Problem. A substantial improvement on the results of this method came with the utilization of violated facet defining inequalities of a polytope associated with the problem to obtain better cutting planes. This approach was developed by Padberg and Grötschel in [PG85] to solve large instances of the Traveling Salesman Problem. In the last years, the Cutting Plane and the Branch and Cut methods based on facet defining inequalities have been used successfully to solve large scale instances of real world combinatorial optimization problems.

5.1 The Branch and Cut Method

In this section we describe the main idea of the approach we apply to the Multiple Knapsack Problem. In the last chapter we associate each solution of the MKP to a vertex of the corresponding polytope $MKP(N, M, f, F)$. So, the problem can be equivalently described as

$$\begin{array}{ll} \min & cx \\ s.t. & x \in MKP(N, M, f, F). \end{array}$$

In order to apply Linear Programming techniques to solve this problem, we need a description of this polytope by means of equalities and inequalities. Since the number of facet defining inequalities is exponential in the size of the input, we use a cutting plane approach to solve this problem. One cannot expect, however, to obtain a complete description of the polytope, since MKP is $\mathcal{NP}$-hard. Nevertheless, the results of the last chapter provide several facet defining and valid inequalities that can be useful to solve the problem. These inequalities might be not sufficient to solve the problem. We embed the cutting plane phase into a branch and bound procedure by fixing some variables.

Let us describe more precisely the Branch and Cut method. In each iteration we consider a polytope P' which is described by some of the valid or facet defining inequalities of the last chapter and with some variables fixed to the lower or upper bounds. It is obvious that P' contains $MKP(N, M, f, F)$. In the first LP, for example, we start with the trivial inequalities only, and no variable fixed. Then, we determine an optimal solution x' of this LP. There are two possibilities:

- $x' \in MKP(N, M, f, F)$. Then, x' is also an optimal solution for MKP;

- $x' \notin MKP(N, M, f, F)$. In this case there is a facet defining inequality $ax \leq \alpha$ of $MKP(N, M, f, F)$ that cuts x' off, i.e., $ax' > \alpha$.

The inequality $ax \leq \alpha$ is called a *cutting plane*, and the problem of finding such an inequality is known as the *Separation Problem*. We discuss in Section 6.2 this problem in detail. If we are able to find some of those inequalities, we add them to the LP, obtaining a new polytope P'' that includes $MKP(N, M, f, F)$ and is included in P'. Therefore, P'' is a "better" relaxation of $MKP(N, M, f, F)$. With this new LP we begin a new iteration. However, it can be the case that we find no such inequality in the separation step, because our knowledge about the polytope is not sufficient or our separation routines are not good enough. In this case, in order to continue the process, we choose a fractional variable and create two subproblems: in the first one we fix the value of this variable to 1, and in the second one, we fix the value to 0. In both cases a new LP is generated, and we begin a new iteration. If, during the branching process, we do not find any fractional variable, a backtrack step is performed. Of course the choice of the branching variable is also important for the performance of the method. We discuss some branching strategies in Section 6.4.

Observe that, since the polytope P' contains $MKP(N, M, f, F)$, the optimal value of the LP (if no variable is fixed) is a lower bound on the optimal solution value. On the other hand, using primal heuristic procedures, we can find upper bounds on the optimal solution value. A bound on the quality of these heuristic solutions can be calculated using the lower bound given by the LP. In Section 6.3 we show some heuristic procedures based on the information given by the LP solution. It is also possible that, if some variables are fixed, the value of the LP solution is greater than the current upper bound, or the current LP is infeasible. In both cases, a backtrack in the branch and bound tree is performed.

In Figure 5.1 we give a simplified flow chart of the method.

5.2 Separation strategies

In this section we address the *Separation Problem* for some classes of inequalities described in the last chapter. Suppose that we are given an optimal solution x' of the current LP. We know that there exists an inequality $ax \leq \alpha$ valid for the polytope $MKP(N, M, f, F)$ such that $ax' > \alpha$. Our job is to find such inequalities.

In Section 3.4.3 we present a proof that the Separation Problem for the Minimal Cover Inequalities of the Single Knapsack Polytope is $\mathcal{NP}$-hard. In all classes of inequalities discussed in the last chapter, the concept of a cover is always involved. Of course, this does not give a proof that there does not exist efficient algorithms for the separation of all these classes. We concentrated, however, in efficient heuristic strategies for separation of the classes discussed. In this section, we present the ideas we use to find violated inequalities in our Branch and Cut Algorithm. We describe, for the various classes of inequalities, the heuristic procedures we implemented. We use the set of problems cb described in Section 6.6.1 to test the various heuristics. The parameter we used to test this "efficiency" is

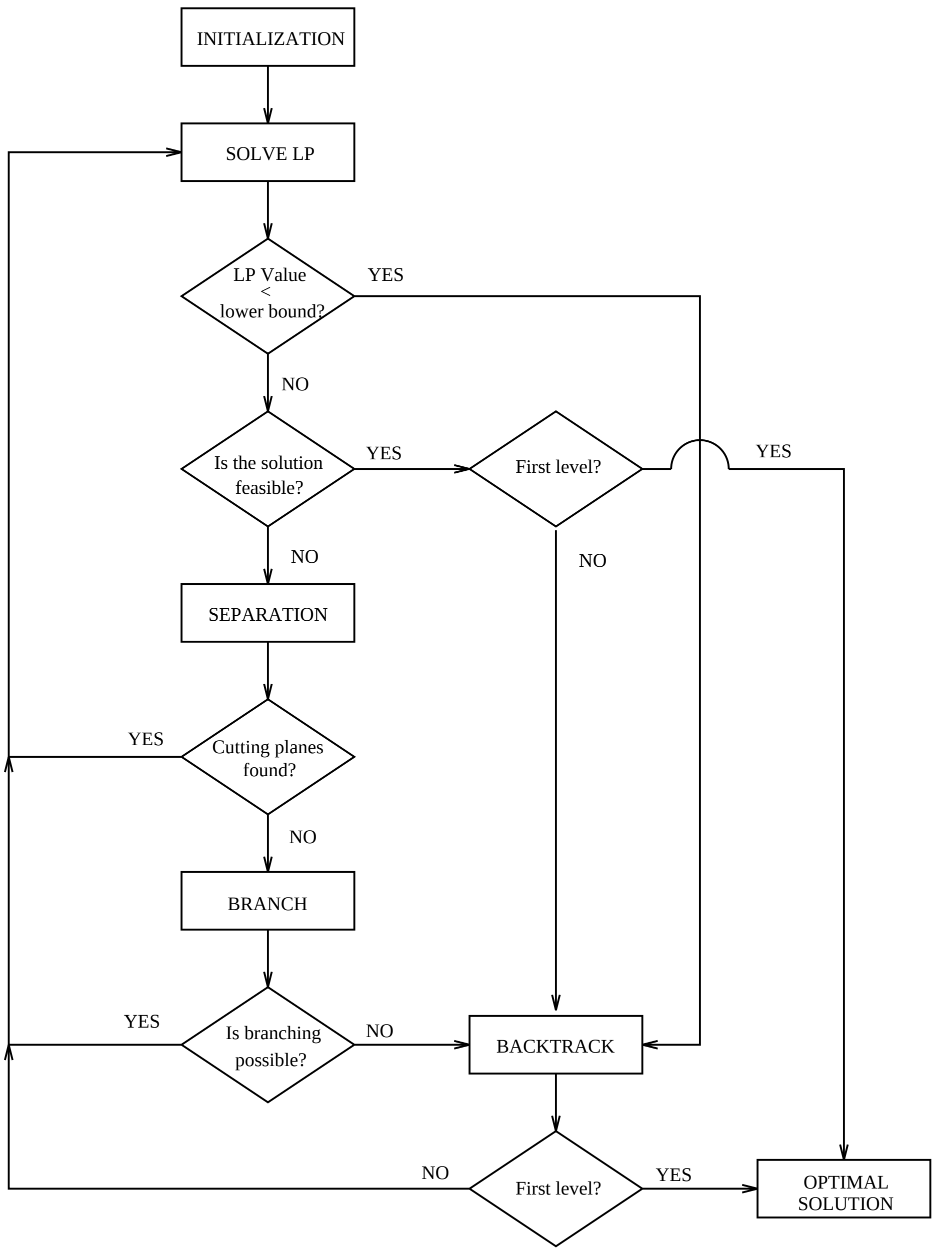


Figure 5.1: Simplified flow chart of the Branch and cut method.

the increase in the value of the LP relaxation caused by the inequalities found by the procedure. To this end we execute our algorithm until the procedure finds no more violated inequality. The last lower bound we obtain is our performance parameter. We compare this value with the value of the lower bound after the first LP (we call it *first lower bound* in the rest of this chapter). In Section 6.5 we describe the inequalities of the first LP.

5.2.1 Separating Minimal Cover Inequalities

Crowder, Johnson and Padberg suggest in [CJP83] the following heuristic procedure for finding violated Minimal Cover Inequalities. The idea is, given a fractional solution x', to solve the following integer program for each knapsack $k \in M$:

$$\begin{array}{rl} \min & \sum_{i \in N} (1 - x'_{ik}) s_i \\ s.t. & \sum_{i \in N} f_i s_i > F_k; \\ & s_i \in \{0, 1\}. \end{array}$$

If the optimum solution value is less than 1.0, then s is the characteristic vector of a cover S, which is violated by x'. Of course we are not able to solve this problem exactly. Thus, we use Dantzig's procedure (see Section 3.1) to find the optimal solution of the relaxation, and round up the value of the critical item. Afterwards, some elements of the cover S are eliminated in order to obtain a minimal one, and the inequality is lifted to a facet of the polytope $MKP(N, M, f, F)$. Let us describe the procedure algorithmically.

> **Sep. Procedure: CJP Heuristic for Minimal Cover Ineq.**
> **Input:** An instance (N, f, F) of the SKP and $x' \in \mathbb{R}^n$, $0 \leq x'_i \leq 1$ for all $i \in N$.
> **Output:** A violated Minimal Cover Inequality or the procedure fails.
>
> Apply Dantzig's procedure to solve:
>
> $$\begin{array}{rl} \min & \sum_{i \in N} (1 - x'_i) s_i \\ s.t. & \sum_{i \in N} f_i s_i > F, \\ & 0 \leq s_i \leq 1 \text{ for all } i \in N. \end{array}$$
>
> Let $S := \{i \in N \mid s_i > 0\}$.
> Transform S in a minimal cover.
> Lift the corresponding inequality to a facet of $SKP(N, f, F)$.
> If the resulting inequality is violated add it to the list of violated inequalities.

We have tested the procedure outlined above on problems `cb`. The results obtained are shown in Table 5.1. The first column shows the value of the first

Problem	1. LB	z^{CJP}
cb1	26649	27621
cb2	27985	28535
cb3	29237	30087
cb4	30607	31279
cb5	39829	40115
cb6	33169	33458
cb7	32027	32991
cb8	32806	33711
cb9	29825	30976
cb10	36612	36953

Table 5.1: Minimal cover separation CJP-heuristic.

lower bound, and the second column, the lower bound z^{CJP} obtained after no more violated inequalities are found by this procedure.

We implemented another idea for the minimal cover separation. In our application, though the number of items is very large, it turns out that many of them have the same weight. Let us say that these items are of the same *type*. Then, when the number of types is small, one can enumerate all possible minimal covers, based on the types of the items.

Example 5.2.1 Suppose an instance (N, f, F) of the SKP in which we have $F = 100$ and only four different item types: 20, 35, 40 and 50. Then, the complete list of possible minimal covers is:

type 20	type 35	type 40	type 50
6	0	0	0
4	1	0	0
4	0	1	0
3	2	0	0
3	0	0	1
2	2	0	0
2	1	1	0
2	0	2	0
1	1	0	1
0	3	0	0
0	2	1	0
0	2	0	1
0	1	2	0

type 20	type 35	type 40	type 50
0	1	1	1
0	0	3	0
0	0	2	1
0	0	0	3

■

The number of items of each type is bounded by n. Let T be the number of different types. Then, it is easy to see that, n^T is an upper bound for the number of different *minimal cover types*. This bound can, indeed, be improved. Let $t \in \mathbb{N}^T$ be an array containing the number of items with each item. An upper bound for the number of different minimal cover types is given by $\prod_{i=1}^{t} \min(t_i, \lceil \frac{F+1}{f_i} \rceil)$. In Example 5.2.1, an upper bound is given by $\min(6, t_{20}) * \min(3, t_{35}) * \min(3, t_{40}) * \min(3, t_{50})$. If the number of different types is small, one can enumerate all possibilities. We can suppose that the items are sorted such that items with the same weight have contiguous indices. Moreover, we can construct in the beginning of the process the array $(t_1, t_2, \dots, t_T)$ containing the number of items with the same weight for each type. Now, given a knapsack k and the current LP solution x', we can sort the items of each type, such that

$$\begin{array}{lll} x'_{1k} \geq & \dots & \geq x'_{t_1 k} \\ x'_{(t_1+1)k} \geq & \dots & \geq x'_{(t_1+t_2)k} \\ \vdots & & \\ x'_{(t_1+\dots+t_{T-1}+1)k} \geq & \dots & \geq x'_{(t_1+\dots+t_T)k} \end{array}$$

Then, we can check for each cover type (enumerated as in Example 5.2.1) if there is a violated Minimal Cover Inequality of this type. To this end, we must only check the best possible inequality of each type, that we obtain with the first items of the type in the order above. Note that it provides a polynomial exact separation procedure if the number of item types is constant.

Actually, in our implementation we do not enumerate all possible cover types. The number of possibilities is still too large. Then, we generate in the beginning of the process a set of cover types, and test, in each iteration, for violated inequalities of these types. Let us describe more precisely the idea of the heuristic procedure.

Sep. Procedure: CT Heuristic for Minimal Cover Ineq.
Input: An instance (N, f, F) of the SKP, T the number of item types, an array $(t_1, \dots, t_T)$ with the number of items of each type, $x' \in \mathbb{R}^n$, $0 \leq x'_i \leq 1$ for all $i \in N$.
Output: Violated Minimal Cover Inequalities, or the procedure fails.

Generate the cover types.
Sort x' according to the following order.

$$\begin{array}{lll} x'_1 \geq & \ldots & \geq x'_{t_1} \\ x'_{t_1+1} \geq & \ldots & \geq x'_{t_1+t_2} \\ \vdots & & \\ x'_{t_1+\ldots+t_{T-1}+1} \geq & \ldots & \geq x'_{t_1+\ldots+t_T} \end{array}$$

For each generated cover type $c = (c_1, \ldots, c_T)$.
    Select from each item type i the c_i best items according to the above order.
    Lift the corresponding inequality.
    If the inequality is violated, append it to the list of violated inequalities.

In Table 5.2 we report on the results obtained by applying this procedure. Let z^{CT} be the value of the last LP, before the procedure fails in finding a violated inequality. The results presented in Table 5.2 show that for these test problems

Problem	1. LB	z^{CT}
cb1	26649	26649
cb2	27985	27985
cb3	29237	29237
cb4	30607	30607
cb5	39829	39829
cb6	33169	33169
cb7	32027	32027
cb8	32806	32806
cb9	29825	29825
cb10	36612	36612

Table 5.2: Minimal cover separation CT-heuristic.

the heuristic procedure fails completely. It is important to say that the fact that the number of different types must be not too large is decisive for the performance of the heuristic. In the next Table 5.3 we test both procedures for minimal cover separation for problems mk (described in Section 6.6.1). The results in bold face indicate a better bound obtained. No improvement on the CJP heuristic was observed. This heuristic procedure is, therefore, not used in the rest of the thesis.

Problem	1. LB	z^{CJP}	z^{CT}
mk1	1168	**1220**	1206
mk2	3813	3845	3845
mk3	351	366	366
mk4	864	880	880
mk5	844	946	946
mk6	693	700	700
mk7	892	964	**967**

Table 5.3: Comparison: CJP and CT heuristics.

5.2.2 Separating (1,k)-Configuration Inequalities

In [CJP83] Crowder, Johnson and Padberg suggest a heuristic procedure to find violated (1,k)-configurations. The idea is to enlarge minimal covers. Take a minimal cover S (found, for example, by applying the procedure outlined above) and let z be the element with maximal weight. Set $N' := S \setminus \{z\}$. Let $i \in N \setminus S$, and verify if $(N' \cup \{i\}) \cup \{z\}$ is a (1,k)-configuration (see Section 3.4.2). If it is the case, the element is inserted into N' and the corresponding inequality is lifted, and, if violated, added to the LP.

In our implementation, we maintain a pool of covers found during the execution of the algorithm. We try to enlarge the covers stored in this pool to (1,k)-configurations. In the following we present the heuristic in algorithmic form.

Sep. Procedure: Heuristic CJP for (1,k)-Configuration Ineq.
Input: An instance (N, f, F) of the SKP, $x' \in \mathbb{R}^n$, $0 \leq x'_i \leq 1$ for all $i \in N$.
Output: Violated (1,k)-Configuration Inequalities or the procedure fails.

Take a minimal cover S from the pool.
Let $z \in S$ be an item with maximum weight.
$N' := S \setminus \{z\}$, $k := |N'|$.
For each item $i \in N \setminus S$ such that $\sum_{j \in N' \cup \{i\}} f_j \leq F$.
    If for all $K \subseteq N' \cup \{i\}$, with $|K| = k$, $K \cup \{z\}$ is a cover.
        $N' := N' \cup \{i\}$.
        Lift the corresponding (1,k)-Configuration Inequality.
        If the inequality is violated, append it to the list of violated inequalities.

In Table 5.4 we show a comparison of the lower bound obtained by this heuristic procedure with the one obtained by our approach described in the following.

In addition, we implemented another heuristic procedure to find violated (1,k)-configurations. The idea is to partition the items set into two subsets: L which contains the large items, and S which contains the small ones. A threshold parameter controls this partition. Then, both sets are sorted by decreasing order of the value of the current LP solution. Let z be the item in L with maximum value of the current LP solution, and $N' \subseteq S$ such that $N' \cup \{z\}$ is a (1,k)-configuration. We choose N' in order to obtain also big values of the current LP solution. In the following we present a detailed description of this heuristic procedure.

Sep. Procedure: Heuristic LS for (1,k)-Configuration Ineq.
Input: An instance (N, f, F) of the SKP, $x' \in \mathbb{R}^n$, $0 \leq x'_i \leq 1$ for all $i \in N$ and a threshold parameter t.
Output: Violated (1,k)-Configuration Inequality or the procedure fails.

Let $L := \{i \in N \mid f_i > t\}$ and $S := \{i \in N \mid f_i \leq t\}$.
Let $z \in L$ be an item such that $x'_z = \max\{x'_i \mid i \in L\}$.
Solve the following problem by applying Dantzig's procedure:

$$\begin{array}{rl} \max & \sum_{i \in S} x'_i t_i \\ s.t. & \sum_{i \in S} f_i t_i \leq F, \\ & 0 \leq t_i \leq 1, \text{ for all } i \in S. \end{array}$$

Set $N' := \{i \in S \mid t_i = 1\}$.
If $N' \cup \{z\}$ is a (1,k)-Configuration.
  Lift the corresponding inequality.
  If the inequality is violated, append it to the list of violated inequalities.

This heuristic procedure has a very good performance compared to our implementation of the heuristic suggested by Crowder, Johnson and Padberg. In Table 5.4 we present the lower bound provided by both heuristic procedures on the same set of examples. Let z^{CJP1k} be the lower bound provided by the procedure suggested by Crowder, Johnson and Padberg and z^{LS} be the lower bound provided by our heuristic. We use in both cases, the CJP-heuristic to find violated minimal covers. In the remainder of this section, we will refer to the combination of the heuristics CJP for minimal covers and LS for (1,k)-configurations, as procedures to find violated *individual* inequalities.

5.2.3 Separating Multiple Cover Inequalities

In order to find violated Multiple Cover Inequalities, we choose a subset M' of knapsacks and determine $\sum_{k \in M'} F_k$. Then, we apply the separation procedures for Minimal Covers and (1,k)-Configuration Inequalities for this "summarized" knapsack. Of course one cannot test all $2^{|M|} - (|M| + 2)$ possible different choices

Problem	1. LB	z^{CJP1k}	z^{LS}
cb1	26649	27627	**27800**
cb2	27985	28587	**28789**
cb3	29237	30087	**30492**
cb4	30607	31279	**31846**
cb5	39829	40125	**40400**
cb6	33169	33458	**33824**
cb7	32027	32991	**33441**
cb8	32806	33713	**33905**
cb9	29825	30976	**31275**
cb10	36612	36953	**37038**

Table 5.4: (1,k)-configuration separation: comparison of heuristics.

for M'. We use the following heuristic idea to reduce the number of tests, and, at the same time, try to obtain many violated inequalities. For each item i we determine $J_i := \{k \in M \mid x_{ik} > 0\}$. If $2 \leq |J_i| \leq |M| - 1$, the set is tested in the heuristic procedure. Finally, if $|J_i| = |M|$, we apply the procedure for all subsets of M with $|M| - 1$ elements.

The behavior of the separation procedures for joint inequalities is shown in Table 5.5.

5.2.4 Separating Heterogeneous Two Cover Inequalities

We use basically the same idea presented for finding violated Minimal Cover Inequalities to find promising sets G and S for a Heterogeneous Two Cover Inequality corresponding to S, G and some knapsacks k and l. First we find a cover S with respect to some knapsack k maximizing the value of the current LP solution in both knapsacks k and l. Then, we construct G by selecting in $N \setminus S$ a maximal set such that for all subset $T \subseteq S \cup G$, $|T| = |S|$ and $T \cap G \neq \emptyset, \sum_{i \in T} f_i > F_l$. In the following we present the procedure algorithmically.

Sep. Procedure: Heuristic for Het. Two Cover Ineq.
Input: An instance (N, M, f, F) of the MKP, $x' \in \mathbb{R}^{mn}$, $0 \leq x'_{ik} \leq 1$ for all $i \in N$. $k \in M$.
Output: Violated Heterogeneous Two Cover Inequalities or the procedure fails.

For all pairs $k, l \in M$, $k \neq l$.
  Solve the following linear program:

$$\begin{array}{rl} \min & \sum_{i \in N}(1.0 - x'_{ik} - x'_{il})s_i \\ s.t. & \sum_{i \in N} f_i s_i > F, \\ & 0 \leq s_i \leq 1 \text{ for all } i \in N. \end{array}$$

Set $S := \{i \in N \mid s_i > 0\}$.
Initialize $G := \emptyset$.
Let s_{max} be such that $f_{s_{max}} = \max\{f_i \mid i \in S\}$.
Initialize $T := S \setminus \{s_{max}\}$; $f_T := \sum_{i \in T} f_i$.
For each element i in $N \setminus S$.
  If $f_T + f_i > F_l$.
    $G := G \cup \{i\}$.
    Update T and f_T conveniently.
Lift the Heterogeneous Two Cover Inequality.
If it is violated, append to the list of violated inequalities.

We present in Table 5.5 some results on applying this heuristic procedure to obtain violated inequalities.

5.2.5 Separating Extended Cover Inequalities

In the last chapter we define some valid and facet defining joint inequalities of the $MKP(N, M, f, F)$, i.e., inequalities that combine the coefficients of more than one knapsack. One of them is the Extended Cover Inequality. In this section we present two heuristic separation procedures for this class of inequalities.

A first heuristic idea is to use the covers stored in our pool to try to extend them to other knapsacks. The procedure works as follows. Let S be a cover for some knapsack k taken from the covers pool, and let $T := N \setminus S$. Then, for all $l \in M \setminus \{k\}$, try to construct an Extended Cover Inequality corresponding to S, T, k and l (see Section 5.2). By considering the definition of the Extended Cover Inequality, one can see that we need a set T that fits into l and $T \cup \{s\}$ is a cover for l for all $s \in S$. The corresponding inequality will be valid. So, we eliminate the items of T until it fits into l, and then test the other condition. The inequality is finally lifted and, if it is violated, appended to the LP. In the following, we present a more precise description of the procedure.

Sep. Procedure: Heuristic EC1 for Extended Cover Ineq.
Input: An instance (N, M, f, F) of the MKP and $x' \in \mathbb{R}^{mn}$, $0 \leq x'_{ik} \leq 1$ for all $i \in N$, $k \in M$.
Output: Violated Extended Cover Inequalities or the procedure fails.

For all covers stored in the covers' pool.
  Let S be a cover for knapsack k taken from the covers' pool.
  Let s_{min} be an item in S with minimum weight.
  Set $T := N \setminus S$.

For all $l \in M \setminus \{k\}$.
Set $T_l := T$.
Sort the elements in T such that $x'_{il} \geq \ldots \geq x'_{|T|l}$.
Determine $t_l := \max\{t \in \{1, \ldots, |N \setminus S|\} \mid \sum_{i=1}^{t} f_i \leq F_l\}$.
Set T'_l as the first t_l elements of T_l.
If $T'_l \cup \{s_{min}\}$ is a cover for l.
Lift the corresponding Extended Cover Inequality.
If the resulting inequality is violated, append it to the list of violated inequalities.

The second heuristic procedure for separation of Extended Cover Inequalities is based on the observation that the coefficients of S appear in the left hand side of the inequality in both knapsacks k and l. We use, then, the same idea given for finding Minimal Cover Inequalities in order to search for promising sets S and T. The procedure looks for violated Extended Cover Inequalities corresponding to knapsacks k and M'. For the choice of the knapsacks to run this heuristic procedure we use the same idea described for the separation of Multiple Cover Inequalities. We describe the procedure in the following.

Sep. Procedure: Heuristic EC2 for Extended Cover Ineq.
Input: An instance (N, M, f, F) of the MKP, $x' \in \mathbb{R}^{mn}$, $0 \leq x'_{ik} \leq 1$ for all $i \in N$, $k \in M$, a knapsack $k \in M$ and $M' \subseteq M \setminus \{k\}$.
Output: Violated Extended Cover Inequalities or the procedure fails.

Apply Dantzig's procedure to solve the problem:

$$\begin{array}{ll} \min & \sum_{i \in N}(1.0 - \sum_{l \in M' \cup \{k\}} x'_{il})s_i \\ s.t. & \sum_{i \in N} f_i s_i > F_k \\ & 0 \leq s_i \leq 1 \text{ for all } i \in N. \end{array}$$

Set $S := \{i \in N \mid s_i > 0\}$.
Transform S in a minimal cover for k.
Set s_{min} be an item in S with minimum weight.
Apply Dantzig's procedure to solve the problem:

$$\begin{array}{ll} \min & \sum_{i \in N \setminus S}(1.0 - \sum_{l \in M'} x'_{il})t_i \\ s.t. & \sum_{i \in N \setminus S} f_i t_i > \sum_{l \in M'} F_l - f_{s_{min}} \\ & 0 \leq t_i \leq 1 \text{ for all } i \in N \setminus S. \end{array}$$

Set $T := \{i \in N \setminus S \mid t_i > 0\}$.
Transform T in a minimal cover for $\sum_{l \in M'} F_l - f_{s_{min}}$.
Lift the corresponding Extended Cover Inequality.
If it is violated, append it to the LP.

In Table 5.5 we show the importance of the joint inequalities in order to solve the MKP. In the second column we show the value of the lower bound obtained by the appending only individual inequalities, and, in the third one, the value of the lower bound obtained by introducing also the joint inequalities presented in the last three sections. As we see in Table 5.5, the combination of individual and

Problem	1. LB	ind. ineq.	ind. and joint ineq.
cb1	26649	27800	**27857**
cb2	27985	28789	**28793**
cb3	29237	30492	**30514**
cb4	30607	31846	31846
cb5	39829	**40400**	40378
cb6	33169	33824	33824
cb7	32027	33441	33441
cb8	32806	33905	**33935**
cb9	29825	**31275**	31224
cb10	36612	37038	37038

Table 5.5: Comparison of lower bound obtained using only individual and individual and joint inequalities.

joint inequalities provide a better bound than the individual inequalities alone in most of the examples tested.

5.2.6 Lifting and Complementing

In Section 3.4.2 we present the lifting procedure suggested by Zemel in [Ze89] to calculate efficiently a lifting of facet defining inequalities of the Single Knapsack Polytope. In our Branch and Cut approach we observed that the Lifting Procedure plays a very important role in the generation of violated inequalities. Sometimes, we find a non violated Minimal Cover Inequality that, after lifting, gives rise to a violated one. We proceed, then, like described in 3.4.2 in order to calculate the lifting coefficients for the individual inequalities.

For the joint inequalities, however, no efficient general procedure is known for the lifting. We have implemented particular heuristic liftings for each separation procedure described in this section. In the case of the Extended Cover and Heterogeneous Two Cover Inequalities, instead of calculating exactly the value of the lifting coefficient by Padberg's Procedure, as described in Section 3.4.2, we calculate the value of the linear relaxation of the integer program presented there and round it down. Let z_i be the value of the integer program and z_i' the bound calculated as outlined above. It is clear that $z_i \leq z_i'$, and therefore, the lifting coefficient calculated by this heuristic is less than or equal to the exact coefficient. It guarantees that the lifted inequality calculated is, at least, valid, but not necessarily facet defining.

For the Multiple Cover Inequality we proceed in a different way. We consider the joint inequality as an individual inequality for the "summarized" knapsack. We apply, then, Zemel's procedure to calculate the value of the lifting coefficient of the "summarized variables". This value is used for the lifting coefficients of the original inequality. This coefficient is surely a lower bound for the exact coefficient of these variables, and therefore, the generated inequality is valid.

Now, we would like to give some more details about the operation of complementing. The idea is to reduce the problem, by fixing some variables to one. In our implementation we complement all variables whose value of the current LP solution are close to one. We use 0.05 as a threshold parameter. Let us describe the procedure more precisely.

We are given an instance (N, M, f, F) of the MKP, the solution x' of the current LP and a threshold parameter ϵ. Let $N^r := N$ and $F_k^r := F_k$ for all $k \in M$. For each variable (i, k), such that $x'_{ik} > 1.0 - \epsilon$, we set $N^r := N^r \setminus \{i\}$ and $F_k^r := F_k^r - f_i$. We apply, then, the separation strategies discussed in the last sections to this reduced problem. If we find a violated inequality in the reduced problem, we calculate the value of the coefficient of the complemented variables, in order to obtain a valid inequality of the original problem. Observe that we complement variables whose value is not exactly one, and, therefore, it can be that a violated inequality in the subproblem is not violated in the original problem any more.

In Section 3.4.3 we present an efficient procedure to calculate the coefficient of the complemented variable in the case of the individual inequalities. No efficient algorithm, however, is known for the joint inequalities. We implemented, again, particular procedures for each inequality class. We proceed, in the case of the Multiple Cover Inequality, like we do for lifting. We consider the joint inequality as a individual one for the "summarized knapsack", and calculate exactly the coefficients in the new "summarized" knapsack. These values are an upper bound for the exact values, and, thus, the generated inequality is valid.

For the case of the Heterogeneous Two Cover Inequality we solve the LP relaxation of the integer program presented in Section 3.4.3. This value is, then, rounded up, guaranteeing that the generate inequality is valid.

For the case of the Extended Cover inequality we can obtain good bounds to the corresponding coefficients by a more careful investigation. Let $k, l \in M$, $S \subseteq N$, $T \subseteq N \setminus S$ and $i \in N \setminus (S \cup T)$ the complemented variable. We have two possibilities for this Extended Cover Inequality, depending on in which knapsack the item i was complemented.

- Case 1. The item i is complemented in knapsack k. Then, $\sum_{j \in S} f_j > F_k - f_i$. It is clear that $S \cup \{i\}$ is a cover for knapsack k. If $T \cup \{i\}$ is a cover for knapsack l, we can insert i into S and the corresponding Extended Cover

Inequality is valid. Otherwise, the inequality

$$\sum_{j \in S \cup \{i\}} x_{jk} + \sum_{j \in S \cup T} x_{jl} \leq |S| + |T| - 1$$

is also valid.

- Case 2. The item i is complemented in knapsack l. Then, $\sum_{j \in T} f_j \leq F_l - f_i$ and $T \cup \{s\}$ is a cover for all $s \in S$. In this case, insert i in T, and observe that the corresponding Extended Cover Inequality is valid.

In Table 5.6 we present the lower bound obtained with the operation of complementing. In the second column we show the first lower bound. The third column shows the lower bound obtained with all separation procedures presented in the last sections, and in the fourth column we see the effect obtained by applying additionally the complementing operation. We observe that we obtain a better

Problem	1. LB	all sep. proc.	all sep. proc. and compl.
cb1	26649	27857	27857
cb2	27985	28793	**28880**
cb3	29237	30514	**30520**
cb4	30607	31846	31846
cb5	39829	40378	**40396**
cb6	33169	33824	33824
cb7	32027	33441	**33451**
cb8	32806	**33935**	33905
cb9	29825	**31224**	31219
cb10	36612	37038	37038

Table 5.6: Comparison of LB obtained by using complementing heuristic.

lower bound using the complementing heuristic for most of these tests.

Final remarks

We have used random problems to test the quality of the inequalities generated by our separation routines. In Section 6.6.1 we describe how we generate these problems. Table 5.7 shows typical results for random problems. We generated for each value of $|N|$ and $|M|$ four different instances and show in the table the average obtained over all tests made. The third, fourth and fifth columns show the average value of the gap between lower and upper bounds after the first LP, after no more violated individual inequalities are found and immediately before branching, respectively.

$\|N\|$	$\|M\|$	Average gap after first LP	Average gap after ind. ineq.	Average gap after joint ineq.
50	4	561.0	231.4 (41.2%)	36.6 (6.5%)
100	4	265.4	164.0 (61.8%)	78.6 (29.6%)
150	4	355.2	149.2 (42.0%)	114.2 (32.3%)
200	4	335.4	117.0 (34.9%)	75.2 (22.4%)
300	4	332.0	112.4 (33.9%)	100.6 (30.3%)
400	4	210.5	154.8 (73.5%)	90.0 (42.8%)
500	4	171.0	41.5 (24.3%)	38.2 (22.4%)

Table 5.7: Random examples.

We can observe in Table 5.7 that although we cannot solve most of the problems without branching, the results confirm that the gap between lower and upper bounds is substantially decreased by using the inequalities generated by our procedures. These problems can be solved using Branch and Cut in reasonable time, as showed in Section 6.6.2.

We have tested several possibilities on the order in which these separation procedures are called, and compared them taking in account both the CPU time used and the quality of the produced bound. In our experiments the following strategy showed the best performance. We use first only the separation routines for individual inequalities with complementing and add afterwards also the routines for separation of joint inequalities.

5.3 Heuristic procedures

In this section we describe two different LP based heuristics which we implemented for the MKP. Moreover, an improvement heuristic is presented and its behavior is tested on some bench mark examples. Observe that for our application (see Chapter 2) we are interested only in feasible solutions assigning all the items. A heuristic solution that does not assign all items would provide, due to the transformation in the objective function presented in Chapter 2, a very poor upper bound. So, throughout this section we will restrict our attention to feasible solutions of the MKP assigning all items. If the heuristic procedure does not assign all items, we will say that it *fails* in producing a feasible solution.

5.3.1 Rounding heuristics

Given the optimal solution x' of the current LP, the idea of the *rounding heuristic* is to try to obtain a feasible solution for the MKP by assigning each item i to the knapsack k such that x'_{ik} is maximum and this assignment does not violate the corresponding knapsack constraint.

Rounding Heuristic.
Input: An instance (N, M, f, F) of the MKP and a vector $x' \in \mathbb{R}^{mn}$, $0 \leq x'_{ik} \leq 1$ for all $i \in N$, $k \in M$.
Output: A feasible solution $y \in M^n$ of MKP assigning all items or the procedure fails (y_i contains the knapsack where the item i is assigned).

Choose a sequence of the items to consider.
Initialize $F'_k := F_k$ for all $k \in M$.
For each item i according to the order chosen.
  Let k^* such that $x'_{ik^*} := \max\{x'_{ik} \mid k \in M \text{ and } f_i \leq F'_k\}$.
  If there is no such k^*, the procedure fails and stop.
  else $y_i := k^*$.
    $F'_{k^*} := F'_{k^*} - f_i$.
Return y.

Observe in Example 5.3.1 that depending on the order in which the items are considered the rounding heuristic provides different results.

Example 5.3.1 Let $n = 3$, $m = 2$, $f = (3, 4, 5)$, $F = (5, 8)$ and the vector

$$x' := \begin{pmatrix} 0.5 & 0.8 & 1.0 \\ 0.5 & 0.2 & 0.0 \end{pmatrix}$$

Then, applying the algorithm above to the sequence $\{1, 2, 3\}$ the procedure fails, because it assigns the first item to the first knapsack, and then there is no possibility to assign the other two items. But, if we apply the procedure in the order $\{2, 3, 1\}$ a feasible solution is obtained. ■

In our implementation, three different orders have been tested: sequential order of the indices, a randomly generated order and the so called "best" order. In this third case we determine for every item $i \in N$, $k^*(i)$ such that $x'_{ik^*(i)} = \max\{x'_{ik} \mid k \in M\}$. The "best" order is, then, the one obtained by sorting the items in decreasing order of the value $x'_{ik^*(i)}$. We applied these variants of the heuristic for examples cb. The results are summarized in Table 5.8, where we used for all examples a fixed separation strategy. The value in the third, fourth and fifth columns correspond to the best heuristic solution found in the first node of the branch and cut tree using the rounding heuristic in sequential, random and "best" order, respectively. The values can be compared with the optimal solution value of the examples, in the second column.

In Table 5.8 we observe that a fixed order does not work well, while the random and "best" order produce similar results.

Problem	Opt. Sol.	Seq. order	Random order	"Best" order
cb1	28035	29143	**28733**	**28733**
cb2	28880	28880	28880	28880
cb3	30541	30541	30541	30541
cb4	32217	fails	**43885**	fails
cb5	40493	40957	**40609**	**40609**
cb6	33824	33824	33824	33824
cb7	33679	38045	**34904**	**34904**
cb8	34009	36741	35036	**34550**
cb9	31533	**37085**	40241	40241
cb10	37143	44786	**38190**	**38190**

Table 5.8: Rounding Heuristic: comparison between different orders.

5.3.2 Randomized rounding

The idea of this heuristic procedure is to consider the solution x' of the current LP as a probability distribution, where x'_{ik} is the probability of the item i to be assigned to knapsack k. A number of experiments is realized, and the best feasible solution found is selected. Let us describe the procedure more precisely.

Randomized Rounding.
Input: An instance (N, M, f, F) of the MKP and a vector $x' \in \mathbb{R}^{mn}$, $0 \leq x'_{ik} \leq 1$ for all $i \in N$, $k \in M$.
Output: A feasible solution y of MKP assigning all items or the procedure fails.

For $t := 1$ to *number-of-trials*
  For each item $i \in N$
    Assign $y_i^t := k$ with probability x'_{ik}.
  If y^t is better than the best solution found so far.
    $y := y^t$.
If some solution was found.
  Return y.
Else procedure fails.

This procedure has been used to provide good heuristic solutions to the Unconstrained Global Routing Problem (see [Le90]). In the Table 5.9 we show the results obtained by this heuristic on the same examples as in Table 5.8. The number of trials is chosen depending on the number of fractional values in the x'. We observed that the efficiency of the procedure decreases if the number of fractional variables is too small.

Problem	Opt. Sol.	Randomized Rounding
cb1	28035	fails
cb2	28880	28880
cb3	30541	30541
cb4	32217	fails
cb5	40493	**40493**
cb6	33824	33824
cb7	33679	**34290**
cb8	34009	**34009**
cb9	31533	fails
cb10	37143	**37375**

Table 5.9: Randomized Rounding.

Although the procedure does not provide a feasible solution in the examples cb1, cb4 and cb9, in all other examples it gives a solution at least as good as the other strategies. We show in Table 5.10 how much time it takes to find an optimal solution of the problem by applying each heuristic. In this case we use fixed separation and branching strategies for all examples.

Problem	Seq. Order	Random Order	"Best" Order	Rand. Rounding
cb1	50.13	**39.55**	**39.38**	57.28
cb2	20.58	20.55	20.53	**17.53**
cb3	22.52	22.48	22.28	22.58
cb4	1:50.33	**1:49.27**	1:50.33	1:54.98
cb5	1:17.00	1.20.73	1:17.15	**44.90**
cb6	4.82	4.83	4.07	4.87
cb7	10:16.85	8:14.00	8:17.25	**2:15.70**
cb8	9:20.07	10:01.73	4:13.18	**13.02**
cb9	15:36.37	12:48.30	12:46.03	**3:55.04**
cb10	9:46.60	11:41.92	11:52.08	**5:28.72**

Table 5.10: Time needed to find an optimal solution (min:sec).

In Table 5.10 we observe that, in general, the Randomized Rounding needs less time to find an optimal solution in most of the examples considered.

5.3.3 Improvement heuristic

The idea of the improvement heuristic is the following: Given a feasible solution y assigning all items, try to find a better solution by one exchanges, i.e., by moving one item from a knapsack to a different one. It is based on a heuristic procedure suggested by Fidducia and Mattheysis in [FM82] for improving graph partitions. Let us describe the procedure more precisely.

> **Improvement heuristic.**
> **Input:** An instance (N, M, f, F) of the MKP and a feasible solution y assigning all items.
> **Output:** A feasible solution y' of the MKP assigning all items.
>
> Initialize y' with y.
> For $p := 1$ to *number-of-passes*.
>   Initialize y^p with y'.
>   Set $F'_k := F_k - \sum_{i:y'_i=k} f_i$ for all $k \in M$.
>   All items are supposed to be unlocked.
>   While $P := \{(i,k) \in N \times M \mid i$ unlocked, $f_i \leq F'_k$ and $y^p_i \neq k\} \neq \emptyset$.
>   Determine i^* and k^* such that $c_{i^*k^*} = \min\{c_{ik} \mid (i,k) \in P\}$.
>   Assign i^* to k^*.
>   Update current solution y^p and F'.
>   Lock item i^* (to avoid cycling).
>   Determine y^* as the best solution found in this pass.
>   If y^* has a better value than y', let $y' := y^*$.
>   Else return y' and stop.
> Return y'.

One of the ideas of the procedure above is to allow in the internal *while*, "up hill" moves. It is a standard trick used in local search procedures to avoid getting stuck in local optima. In the end of the pass, all solutions found during it are analyzed, and the best one is selected. If the initial solution of the pass is the best one, this solution is returned. Otherwise a new pass is performed. Observe that the procedure is time consuming, since $O(nm)$ is required in each pass to test possible moves.

We applied the improvement heuristic to the same examples showed in the last section. In Table 5.11 we show in bold face the values were some improvement was achieved. For problems cb2, cb3 and cb6 no improvement is possible since an optimal solution is found before branching. Therefore these examples do not appear in this table.

In the next Table 5.12 we give the time needed to each heuristic procedure to find an optimal solution. It is shown in bold face when some substancial reduction in the time was obtained.

Prob.	Opt. Sol.	Seq. order	Rand. order	"Best" order	Rand. Round.
cb1	28035	**28952**	**28701**	**28701**	–
cb4	32217	–	**41010**	**41010**	–
cb5	40493	**40942**	**40493**	**40541**	40493
cb7	33679	**34400**	**33855**	34904	**34290**
cb8	34009	**34580**	**34401**	**34045**	34009
cb9	31533	**34603**	**33976**	**34138**	–
cb10	37143	**37747**	**37425**	**37824**	**37326**

Table 5.11: Improvement heuristic.

Problem	Seq. Order	Rand. Order	"Best" Order	Rand. Rounding
cb1	53.78	39.27	39.20	1:01.78
cb2	20.55	**19.35**	19.80	17.52
cb3	**14.90**	22.32	19.03	**15.60**
cb4	1:50.43	1:50.10	1:50.63	**1:52.63**
cb5	1:21.37	**1:02.88**	1:30.70	45.02
cb6	4.82	4.73	5:12	4.93
cb7	**3:19.15**	**1:29.63**	**3:42.67**	**1:54.03**
cb8	**3:20.63**	**3:02.08**	**1:01.37**	13.20
cb9	**13:14.10**	**10:26.17**	**10:25.67**	3:58.11
cb10	15:32.42	**3:01.80**	**3:17.83**	5:31.32

Table 5.12: Time to find an opt. sol. with improvement heuristic (min:sec).

In both Tables 5.11 and 5.12 we observe that the improvement heuristic has a better behavior when applied to the solutions coming from the Rounding heuristics than with the ones coming from Randomized Rounding.

We use the combination of Randomized Rounding, and Rounding in random order with improvement as primal heuristic for most problems.

5.4 Branching strategies

In the beginning of this chapter we describe the Branch and Cut method, and mention the importance of the choice of the branching variable to solve instances of the MKP. Our aim is to maintain the number of levels in the branch and bound tree small, otherwise the method turns out to be impracticable. In this section we present four different branching strategies, and compare them empirically on a set of examples. Moreover, for each different strategy we implemented two variants, in the first one the branching variable is set to one and afterwards to zero, and in the other one it is made in the opposite order.

The four branching strategies we tested are the following:

- *max*: choose a fractional variable (i, k) such that x'_{ik} is maximum over all fractional variables;
- *min*: choose a fractional variable (i, k) such that x'_{ik} is minimum over all fractional variables;
- *rand*: choose randomly a fractional variable;
- *half*: choose a fractional variable (i, k) such that $|x'_{ik} - 0.5|$ is minimum.

The idea behind the first two heuristics is to try to use the information given by the LP solution to find a better solution. In the *rand* strategy, we avoid using a fixed strategy. The idea of the *half* strategy is that when a branching step is performed, the variables near from 0.5 do not give enough information yet and, in some sense, are the "conflict variables". This is the strategy suggested in most papers on Branch and Cut Algorithms. In the next Table 5.13 we show the total CPU time used to solve some of the examples `cb` using each of the eight strategies described. In all examples we use fixed separation and heuristic strategies. In Table 5.13 we observe that the strategy *half* has a better performance, especially in the cases where no good upper bound is known before branching (example `cb7`, for instance). We use this strategy whenever we have to proceed a branching step.

5.5 Further implementation details

In Figure 6.1 we present a flow chart of the Branch and Cut Method for the MKP. In the last sections we discuss extensively three steps of the method: the

Problem	*min*		*max*		*rand*		*half*	
	zero	one	zero	one	zero	one	zero	one
cb1	0:24	1:57	0.24	3:06	0:32	0:43	**0:20**	0:41
cb2	0:54	0:54	0:54	0:54	0:49	**0:41**	0:43	0:44
cb3	1:40	1:38	1:27	1:50	1:51	1:40	1:06	**0:59**
cb7	24:40	26:36	17:14	15:17	11:28	6:38	3:41	**2:07**

Table 5.13: Solution time using different branching strategies (min:sec).

separation, calculation of upper bounds and branching strategy. In this section we describe briefly our implementation of the other steps given in the flowchart.

In Section 6.1 we mention that first LP can be constituted by some set of valid inequalities for $MKP(N, M, f, F)$. In our implementation we use the knapsack constraints, the SOS constraints and the inequalities $0 \leq x_{ik} \leq 1$, for all $i \in N$, $k \in M$ as the first LP.

A fundamental step of the method is the solution of the linear relaxation of the current LP. In this step we use the CPLEX Callable Library [CPLEX92] a very fast and robust LP solver written and supported by R. E. Bixby. As usual in cutting plane based algorithms we use the dual version of CPLEX.

Solving large LP's is a very hard job. Although CPLEX is one of the best LP solvers on the market, we try to maintain the number of constraints of the LP "acceptable". To this end, we consider the value of the slack variables of each constraint. If this value is less than zero, the inequality is deleted from the LP. The inequalities leaving the LP are stored for some time in a "pool" of inequalities. In each iteration every inequality in this pool is considered and, if it is violated, added to the LP again. Periodically the inequalities in the pool are examined and the oldest ones are eliminated by a garbage disposer. The "age" of an inequality is the number of LP's since it was eliminated.

Another idea we have implemented to simplify the structure of the LP to be solved is the fixing of variables by reduced costs, as suggested by Crowder, Johnson and Padberg in [CJP83]. Let z' be the value of the current LP solution and $d = (d_{ik})$ the corresponding reduced costs. Moreover, let z^* be an upper bound to the optimum value. Then, for each non basic variable (i, k), we can verify two cases.

- if $x'_{ik} = 0$ and $z^* - z' < d_{ik}$ then the variable can be fixed to zero;
- if $x'_{ik} = 1$ and $z^* - z' < -d_{ik}$ then the variable can be fixed to one.

In Table 5.14 we show the number of variables we can fix by reduced costs (and consequent logical implications) in the solution process of some examples.

Problem	# variable fixed	
cb1	148	(82.2%)
cb2	320	(88.9%)
cb3	341	(90.7%)
cb4	360	(85.1%)
cb5	491	(89.4%)
cb6	0	(0.0%)
cb7	279	(88.6%)
cb8	284	(86.3%)
cb9	316	(84.0%)
cb10	455	(93.2%)

Table 5.14: Number of variables fixed by reduced costs.

In the last section we describe several separation heuristics. The total number of inequalities appended to the LP in each iteration is a parameter of our algorithm. If we find more violated inequalities than is specified by this parameter, we select the best inequalities. As selection parameter the current "slack" (or violation) of the inequality is used.

Another important aspect of the selection of cuts to the LP is the redundancy checking. It can be the case that the separation heuristics generate the same inequality several times. Before appending a new cut to the LP, one must verify if it is not already added. This checking is a time consuming operation, and we try to avoid comparing all coefficients. Then, before doing that, we compare the slacks of the cuts, the values of the right hand sides and the number of coefficients. We proceed the coefficient comparison only when all this parameters are equal. Moreover, all inequalities we generate in our separation procedures are appended to a candidates list where each inequality is sorted in decreasing order of the coefficients. It provides also some gain if the comparison is necessary.

5.6 Computational results

We present in this section the computational results of the algorithm described in the last sections. The idea is to show the behavior of our implementation of a Branch and Cut algorithm for several different classes of examples. In Section 6.6.1 we describe these examples, providing, in the case of the real world problems, a brief discussion of the application that they come from. In Section 6.6.2 we show the results obtained by solving some small and random examples. These examples have been used as tests for our program. Section 6.6.3 addresses the examples coming from the practical applications that motivated this investigation.

Problem	n	m	$\sum_{i\in N} f_i$	$\sum_{k\in M} F_k$
mk1	11	4	44	45
mk2	34	4	143	143
mk3	5	3	30	33
mk4	7	3	25	25
mk5	7	3	36	38
mk6	7	3	25	27
mk7	7	3	33	37

Table 5.15: Description of the examples mk.

Problem	n	m	$\sum_{i\in N} f_i$	$\sum_{k\in M} F_k$
cb1	30	6	2497	2700
cb2	45	8	3325	3600
cb3	47	8	3425	3600
cb4	47	9	3890	4050
cb5	61	9	3704	4050
cb6	30	5	2497	2560
cb7	45	7	3325	3584
cb8	47	7	3425	3584
cb9	47	8	3890	4096
cb10	61	8	3704	4096

Table 5.16: Description of the examples cb.

5.6.1 Description of the problem instances

We tested the performance of our implementation of a Branch and Cut Algorithm on four different types of examples. Let us call them mk, cb, cl and pd. In Tables 5.15, 5.16, 5.17 and 5.18 we present the characteristics of the examples.

The examples mk have been created by ourselves, in order to test our algorithm with small instances and investigate the polytope. We include them in this summary, in some sense, only for "historical" reasons.

The examples cb come from an application in compiler design in distributed systems, investigated by Johnson, Mehrotra and Nemhauser in [JMN91]. Each item represents a module of the program, and the knapsacks are computers, where its capacity is the available memory space measured in KBytes. The capacities are equal for all computers in this series of examples. In Johnson, Mehrotra and Nemhauser's model, a graph is given, where the modules constitute the set

Problem	n	m	$\sum_{i\in N} f_i$	$\sum_{k\in M} F_k$
cl1	2292	16	9522	10000
cl2	681	16	2571	2704
cl3	2669	16	6762	7104
cl4	1021	16	4031	4240
cl5	68	16	260	288
cl6	6112	16	25392	26672

Table 5.17: Description of the examples cl.

Problem	n	m	$\sum_{i\in N} f_i$	$\sum_{k\in M} F_k$
pd1	257	4	83827	132704
pd2	772	6	284608	423972

Table 5.18: Description of the examples pd.

of nodes and there is an edge linking two nodes if the corresponding modules communicate. In our application, we ignore the edges, and consider the problem of distributing the modules into the computers.

Examples cl come from an application in circuit layout, investigated by Jünger, Martin, Reinelt and Weismantel [JMRW92] and Weismantel [We92]. In this case, the items are computer cells and they must be distributed in a chip. One heuristic idea proposed by the authors is to divide the total area of the chip in a number of different subregions with the same area. In this case, the subregions correspond to the knapsacks. All knapsacks have the same capacity. The affinity between the cells, modelled in [JMRW92] as a 0/1 quadratic program, is approximated by a linear objective function.

The examples pd come from the application described in Chapter 2. As we point out in the description of the problem, it is a simplification of the real problem, where the edges are also simulated, in some way, in the objective function.

We implemented also three possibilities for the objective function considered: we calculate (as described in Chapter 2) an objective function (we run examples pd and mk with this objective function), read a given objective function or generate an objective function randomly (examples cb). Moreover, we implemented a generator of random instances. In these problems, the weight of the items are randomly chosen in $[5, 300]$, the objective function coefficients are random numbers in the interval $[1, 1000]$ and the knapsack capacities are randomly computed such that $\sum_{k\in M} F_k \leq \alpha \sum_{i\in N} f_i$ where α is a random number chosen in the interval $[1.05, 1.3]$.

Problem	1^{st} LB	Opt. Sol.	# Ind. cuts	# Joint cuts	CPU Time
mk1	1167	1316	98	10	1.18
mk2	3813	4036	73	21	2.43
mk3	351	366	117	9	0.63
mk4	864	880	147	12	0.67
mk5	844	989	205	10	0.72
mk6	693	727	114	8	0.73
mk7	892	967	78	8	0.68

Table 5.19: Solution of the examples mk.

5.6.2 Small and random instances

In this section we report on our results solving the examples mk and the random instances. As we observe in the description of the problems, the examples mk are constructed very tight, i.e., the total weight of the items is close to the total available knapsack capacity. We were able to solve all examples to optimality, without branching. In Table 5.19 we show the value of the first lower bound, the value of an optimal solution, the number of individual and joint inequalities found during the execution of the algorithm, and the total CPU time (seconds) used to solve the problem (running in a Sun Sparc IPX).

In Table 5.20 we show some results on randomly generated problems. Observe that we need to branch to solve the random problems to optimality, but the computation times (indicate in min:sec running in a Sun Sparc IPX) are acceptable. We use the notation mk.n.m for these problems, where n is the number of items and m the number of knapsacks involved. In the next table we show the number of individual and joint inequalities used to solve the problems. Since we need to branch to prove optimality for all problems, we show also the number of branch and cut nodes visited.

5.6.3 Practical problems

In this section we present the tests we performed using real data, coming from the different applications.

Examples cb

As described in the last section, these examples come from an application in compiler design. We tested the program with the original objective function, but the examples turned out to be trivial. In the original problem, the objective function coefficients corresponding to variables x_{ik} is zero, since it makes no difference to

Problem	1^{st} LB	Opt. Sol.	# Ind.	# Joint	# BC	CPU Time
mk.50.4	12198	12558	1156	22	3	47.10
mk.100.4	20640	20702	2312	145	27	12:06.88
mk.150.4	41804	41954	515	532	1853	16:55.22
mk.200.4	44936	45082	1536	1331	3543	47:45.20
mk.300.4	88715	88870	1179	307	577	39:54.73
mk.400.4	94766	94796	4830	3833	5383	92:38.37
mk.500.4	107076	107092	284	228	201	22:37.90

Table 5.20: Solution of random examples.

Prob.	1^{st} LB	Opt. Sol.	# Ind. cuts	# Joint cuts	# BC	CPU Time
cb1	26649	28035	2942	90	5	20.67
cb2	27985	28880	1445	147	1	41.95
cb3	29237	30541	2387	210	3	59.40
cb4	30607	32217	50891	592	43	8:23.02
cb5	39829	40493	6068	153	7	2:07.40
cb6	33169	33824	681	10	1	4.55
cb7	32027	33679	13069	182	21	2:15.85
cb8	32806	34009	6416	145	5	1:06.82
cb9	29825	31533	43378	1012	77	14:47.60
cb10	36612	37143	4413	134	23	3:40.08

Table 5.21: Solution of the examples cb.

which computer a module is assigned. We used, thus, in our tests, a randomly generated objective function. Moreover, in the case studied by Johnson, Mehrotra and Nemhauser no restriction is made on the maximum number of computers (knapsacks). We assumed that only the minimum necessary was available. We need to branch in all except two problem to solve them to optimality. In Table 5.21 we show the value of the first lower bound, the value of an optimal solution, the total number of individual and joint inequalities found during the execution of the algorithm, the number of nodes in the branch and cut tree and the total CPU time (min:sec) used to solve the problems (on a Sun Sparc IPX).

It is interesting to compare the results above with the ones in Tables 5.10 and 5.12 where we show the time needed to find an optimal solution for these examples. For most problems, we use about 80% from the CPU Time given in Table 5.21 in order to prove optimality.

Problem	1^{st} LB	Opt. Solution	# Ind. cuts	# Joint cuts	CPU Time
cl1	2292	2292	0	0	3:56.30
cl2	932.99	939.99	145	60	23:52.40
cl3	2669	2669	0	0	3:25.18
cl4	1021	1021	0	0	4:16.65
cl5	472	472	0	0	1.80
cl6	6112	6112	0	0	27:03.75

Table 5.22: Solution of the examples cl.

Examples cl

In the examples cl, as we observe in Table 5.17, the ratio between the total weight of the items and the total available knapsack capacity is close to one. One could expect that these examples are difficult to solve. However, the first lower bound provides already the value of the optimal solution in almost all examples. A possible explanation for this fact is that in these examples the weights of the items are very similar and that there are many items with small weights and identical objective function value for all knapsacks. In the cases when the value of the first LP is already equal to the value of an optimal solution our primal heuristic procedures provide an optimal solution in the first iteration of the algorithm. It occurs for all examples except cl2. In Table 5.22 we show the value of the first lower bound, optimal solution value, number of individual and joint inequalities found during the solution of the problems and total CPU Time (min:sec) used to solve the examples (in a Sun Sparc IPX).

A desired issue for the circuit designers is to determine the minimum area for which the problem still has feasible solutions. Of course the problem turns to be infeasible if, after some area reduction, the total weight of the items exceeds the total available knapsack capacity. We reduce, then, the knapsack capacities for the problems above, and the result was the same. We solve all reduced examples, except the reduced instances of cl2, in the first iteration. We could also solve all reduced instances of cl2 to optimality without branching. We show the results in Table 5.23. Observe that a reduction quote of 5% leads to infeasibility, since the total available knapsack capacity is then less than the total weight of the items.

For most examples in Table 5.23, provably good solutions are found in acceptable CPU times. In Table 5.24 we show how much time is spent to obtain solutions with maximal deviation under 5% from the optimum (using the lower bound provided by the LP when this solution was found). We give additionally the time to find an optimal solution.

Problem	1^{st} LB	Opt. Sol.	# Ind. cuts	# Joint cuts	CPU Time
cl2 1% red.	939.99	946.99	318	143	21:12.10
cl2 2% red.	946.99	946.99	126	135	22:12.25
cl2 3% red.	953.99	960.99	162	146	22:56.90
cl2 4% red.	957.49	967.99	196	185	16:06.03

Table 5.23: Solution of the reduced instances of cl2.

Problem	max. 5% dev.	opt. sol.
cl2 0% red.	1:45.60	23:52.22
cl2 1% red.	3:44.67	3:44.67
cl2 2% red.	13:48.18	22:09.37
cl2 3% red.	1:18.28	20:09.12
cl2 4% red.	2:44.08	2:44.08

Table 5.24: Time for solution under 5%.

Examples pd

The examples pd come from the application that motivated this thesis and is described in Chapter 2. There we describe also the objective function used in the tests. In Table 5.18 we observe that the total available capacity of the knapsacks is much greater than the total weight of the items. It is clear, because after assigning the items there must be enough space for the wires on the plates. The original instances are, therefore, trivial. We show the results on solving this instances in Table 5.25.

As for the cl examples, it is interesting to the practical application to investigate instances with reduced knapsack capacities. We could find "difficult" instances of the example pd1 with a reduction rate around 36.7%, and for pd2 around 30%. Observe that a reduction of 36.85% in example pd1 and 33% in example pd2 leads to infeasibility, since the total available knapsack capacity is

Problem	Opt. Sol.	CPU Time
pd1	236250	1.97
pd2	81120	6.50

Table 5.25: Solution of the instances of pd.

Problem	1^{st} LB	Opt. Sol.	# Ind.	# Joint	#BC	CPU
pd1 36.75% red.	236250	236250	18	15	1	9.72
pd1 36.8% red.	236250	236250	18	15	1	9.42
pd2 27% red.	81130	81134	9	16	1	40.75
pd2 28% red.	81164	81176	42	41	1	5:22.25
pd2 29% red.	81200	81204	25	29	1	1:08.30
pd2 30% red.	81293	81302	33	15	1	36.41.92
pd2 31% red.	81482	81484	2547	3308	105	985:22.03
pd2 32% red.	81728	81736	44	62	5	56:35.20

Table 5.26: Solution of the reduced instances of pd examples.

less than the total weight of the items.

In Table 5.26 we show the examples and the area reduction, the value of the first lower bound, the value of an optimal solution, the number of individual and joint inequalities found during the solution of the examples, the number of BC nodes and the total CPU time (min:sec) used to solve the problems to optimality (on a Sun Sparc IPX).

Problem	max. 5% dev.	opt. sol.
pd1 36.75% red.	9.72	9.72
pd1 36.8% red.	9.38	9.38
pd2 27% red.	20.82	20.82
pd2 28% red.	7.05	5:22.23
pd2 29% red.	21.33	21.33
pd2 30% red.	21.00	36.41.90
pd2 31% red.	19.27	980:21.20
pd2 32% red.	22.05	56:22.95

Table 5.27: Time for solution under 5%.

In Table 5.27 we present the time needed to provide a solution with maximal deviation from the optimum value under 5% (using the lower bound given by the current LP when this solution was found), and the time for finding an optimal solution.

5.7 Final remarks

In this chapter we presented a Branch and Cut Algorithm for MKP. We were able to solve to optimality both random generated and instances coming from practical applications.

For all practical problems we could increase the lower bound to the optimal solution value. Most of the time spent to solve these problems was needed to find a feasible solution with this value. This fact shows that our primal heuristics must be improved.

On the other hand, these procedures provided quite good solutions (with maximum deviation from the optimum under 5%) for the real problems in acceptable computer times (see Tables 6.24 and 6.27).

Our experiments with randomly generated problems show that, although we are not able to solve most of these problems to optimality without branching, the lower bound provided by our separation routines and the upper bound given by the primal heuristic procedures are, in most cases, quite close.

Summarizing, the Branch and Cut Method showed promising results on solving instances of MKP with up to 4000 variables.

Chapter 6

Multicut Problems

Multicut problems have been receiving considerable attention in the last years. The reason for this fact is that many interesting real world problems can be formulated as multicut problems on graphs or hypergraphs. Several heuristic approaches are suggested in the literature to attack these problems. Some methods like local search, simulated annealing, tabu search and others are much studied applied to multicut problems. We will not go into these approaches in this summary. Rather, we focus on polyhedral investigations. More precisely, we provide in the following an overview on some papers of the literature that solve real world problems using methods based on polyhedral theory.

In Section 7.1 we present a general formulation of multicut problems. We focus on problems in graphs, since we do not know any paper on hypergraphs in the literature. In the other sections we present in detail some applications for some well known special cases. In Section 7.2 we present the Maxcut Problem. Section 7.3 is devoted to the Equicut Problem. Other multicuts problems studied in the literature are treated in Section 7.4. In Section 7.5 we discuss the Node-Weighted Multicut Problem.

6.1 Classifying multicut problems

We present, in this section, a general formulation for the various problems. The treatment is based on the paper by Deza, Grötschel and Laurent [DGL91].

A multicut problem in a graph can be formulated informally in the following way. We are given a graph (with edge weights) and want to partition its node set in such a way that certain constraints are satisfied and an objective function is minimized. For instance, we could be interested in partitioning the nodes of a graph into two parts such that the sum of the weights of the edges with an endnode in each part of the partition is minimal. Additionally, we could require each part of this partition to have approximately the same cardinality, or, if the nodes of the graph have weights, restrict the sum of the weights of the nodes in a partition set by some capacities. In the remainder of this section we present the problems more precisely.

Suppose we are given a graph $G = (V, E)$. Moreover, let $f \in \mathbb{N}^V$ be the node weights, m be an integer number ($M = \{1, \ldots, m\}$) representing the maximum cardinality of the partition and $F \in \mathbb{N}^M$ a vector of partition capacities. We say that a partition $(S_1, \ldots, S_j)$ of V is *valid* if $j \leq m$ and $\sum_{v \in S_k} f_v \leq F_k$ for all $k = 1, \ldots, j$. Given a partition $(S_1, \ldots, S_j)$ of V, we call the set $\delta(S_1, \ldots, S_j)$ a *multicut* of G, m, f and F. A multicut is *valid* if the corresponding partition is valid. If we want to emphasize that the partition has cardinality exactly equal to m we call the set $\delta(S_1, \ldots, S_m)$ a m-multicut.

Suppose now we are given cost functions $c : E \longrightarrow \mathbb{R}$ and $d : V \times M \longrightarrow \mathbb{R}$. With the definitions given above we can provide a general formulation for multicut problems in graphs.

> **Problem:** Multicut Problem on Graphs $\mathrm{MC}^{\leq}$.
> **Instance:** A Graph $G = (V, E)$, an integer m ($M = \{1, \ldots, m\}$), vectors $f \in \mathbb{N}^V$ and $F \in \mathbb{N}^M$, and cost functions $c : E \longrightarrow \mathbb{R}$ and $d : V \times M \longrightarrow \mathbb{R}$.
> **Question:** Find a valid multicut $Z = \delta(S_1, \ldots, S_j)$ of G, m, f and F that minimizes $\sum_{e \in Z} c(e) + \sum_{k=1}^{j} \sum_{v \in S_k} d(v, k)$.

If we want to emphasize that the cardinality of the partition must be equal to m, we denote the problem as $\mathrm{MC}^{=}$.

Using the notation given above, we can denote the Maxcut Problem as $\mathrm{MC}^{=}$ where $m = 2$, $f \equiv \mathbf{0}$, $F \equiv \mathbf{0}$ and $d \equiv \mathbf{0}$. The Equicut Problem is, in this notation, $\mathrm{MC}^{=}$ where $m = 2$, $f \equiv \mathbf{1}$, $F_1 = \lfloor \frac{|V|}{2} \rfloor$, $F_2 = \lceil \frac{|V|}{2} \rceil$ and $d \equiv \mathbf{0}$. The so called Clique Partitioning Problem (that can also be seen as a multicut problem, see [GW89]) is $\mathrm{MC}^{\leq}$ where $m = |V|$, $f \equiv \mathbf{0}$, $F \equiv \mathbf{0}$ and $d \equiv \mathbf{0}$ (in this case all partitions of V are valid). In the next sections we go into more detail with respect to the most studied multicut problems in the literature.

6.2 The Maxcut Problem

In the notation presented in the last section, the Maxcut Problem is $\mathrm{MC}^{=}$ where $m = 2$, $f \equiv \mathbf{0}$, $F \equiv \mathbf{0}$ and $d \equiv \mathbf{0}$. The problem is, in general, $\mathcal{NP}$-hard (cf. [K72]) but it is solvable in polynomial time if the graph is planar (see [OD72]). It remains, however, $\mathcal{NP}$-hard even for almost planar graphs, i.e., graphs G that contain a node v such that $G - v$ is planar (cf. [Ba83]).

In [BGJR88] Barahona, Grötschel, Jünger and Reinelt present two real world applications of the Maxcut problem. The first application arises in the field of statistical physics. The problem studied is the determination of so called ground states of spin glasses with exterior magnetic field. A spin glass is an alloy of magnetic impurities diluted in a non magnetic metal. At 0 K, the spin glass system attains a minimum energy configuration. This configuration is called a ground state. The behavior of the energy of such system has been receiving much attention in the literature. There exists many models to describe this behavior. The authors present a reduction of the problem of finding the energy configuration of such a ground state to a Maxcut Problem in graphs. Using an algorithm based on cutting planes the authors solve problems on toroidal grid graphs of up to 40×40 nodes. A new implementation of a Cutting Plane Algorithm for this problem is given by Saigal [Sai91], where problems up to 70×70 could be solved.

The second application mentioned in [BGJR88] arises in VLSI design. The design of a chip is divided in several phases. After determining where the cells of a chip must be placed, and after routing all the nets of a chip, one must provide an assignment of the wire segments to the layers. This assignment must guarantee that two wire segments do not cross on a layer. The connections between different layers are provided by the so called vias. These vias use considerable space and are an obstacle to the reduction of the size of the chip. Due to this fact, it is desirable to assign the wires to the layers in such a way that the number of vias is minimized. For the case of two layers the problem can be reduced to the Maxcut Problem in a planar graph, and, therefore, it can be solved efficiently. For the case where some pin preassignments and layer preferences must be satisfied, the Maxcut Problem resulting from the reduction is $\mathcal{NP}$-hard (as pointed out in [BGJR88]). Grötschel, Jünger and Reinelt present in [GJR89] an exact cutting plane algorithm for the problem of via minimization. They solve real world examples obtaining improvements from 7.14% to 37.29% in the total number of the vias compared to the original solution used in the production process. Let us discuss some polyhedral results of the literature on the Maxcut problem.

First we define a polytope corresponding to the problem. Suppose we are given an instance of the Maxcut Problem, i.e., a graph $G = (V, E)$ and a cost function c on the edges of G. We introduce a variable $x \in \{0,1\}^E$ corresponding to the characteristic vectors of 2-multicuts in G. We define the Maxcut Polytope by

$$MC^{=}(G, 2, \mathbf{0}, \mathbf{0}) := \operatorname{conv}\{x \in \mathbb{R}^E \mid \begin{array}{l} x = \chi^{\delta(S_1,S_2)} \text{ for some 2-multicut} \\ \delta(S_1, S_2) \text{ of } G \text{ or } x = \mathbf{0}\}. \end{array}$$

The facial structure of the polytope $MC^{=}(G, 2, \mathbf{0}, \mathbf{0})$ has been extensively studied. The paper [DL91] of over 80 pages length surveys its state-of-the-art. The next theorems state some known results on facet defining inequalities of this polytope (for more details on the proofs for these results see [BM86] and [BGM85]).

Theorem 6.2.1 *Let $C \subseteq E$ be a cycle in G and $F \subseteq C$ with $|F|$ odd. Then, the inequality*

$$\sum_{e \in F} x_e - \sum_{e \in C \setminus F} x_e \leq |F| - 1$$

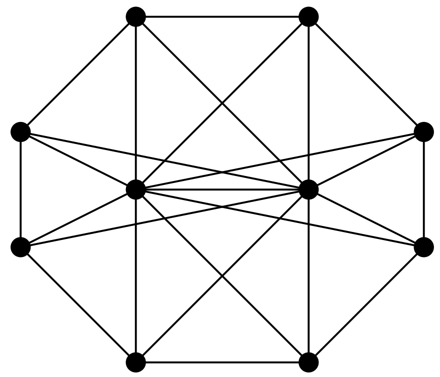


Figure 6.1: A bicycle 8-wheel.

is called Odd Cycle Inequality *and defines a facet of* $MC^=(G, 2, \mathbf{0}, \mathbf{0})$ *if and only if* C *has no chord.* ■

The Odd Cycle Inequalities provide (with the integrality constraints) an integer programming formulation for the Maxcut Problem. Barahona and Mahjoub show that these inequalities suffice to provide a complete description of the polytope $MC^=(G, 2, \mathbf{0}, \mathbf{0})$ if and only if G is not contractible to the graph K_5. Moreover, they show that the separation problem corresponding to this class of inequalities can be solved in polynomial time.

Another class of facet defining inequalities of the polytope $MC^=(G, 2, \mathbf{0}, \mathbf{0})$ is the *Bicycle p-Wheel Inequalities*. A graph is called a *bicycle p-wheel* if it consists of a cycle of length p and two nodes adjacents to each other and to all nodes in the cycle (see Figure 6.1).

Theorem 6.2.2 *Let* $G' = (W, F)$ *be a bicycle (2k+1)-wheel,* $k \geq 1$*, contained in* G*. Then, the inequality*

$$\sum_{e \in F} x_e \leq 2(2k+1)$$

defines a facet of $MC^=(G, 2, \mathbf{0}, \mathbf{0})$*.* ■

This class of inequalities can be also separated in polynomial time (cf. [Ge85]).

Other results on facet defining inequalities and the computational complexity of the corresponding separation problems can be found in the papers cited above.

6.3 The Equicut Problem

The Equicut Problem is MC$^=$ where $m = 2$, $f \equiv \mathbf{1}$, $F = (\lfloor \frac{|V|}{2} \rfloor, \lceil \frac{|V|}{2} \rceil)$ and $d \equiv \mathbf{0}$. The problem is $\mathcal{NP}$-hard for general graphs (cf. [GJS76]). An application of this problem arises in the calculation of the frontwidth in finite element computations

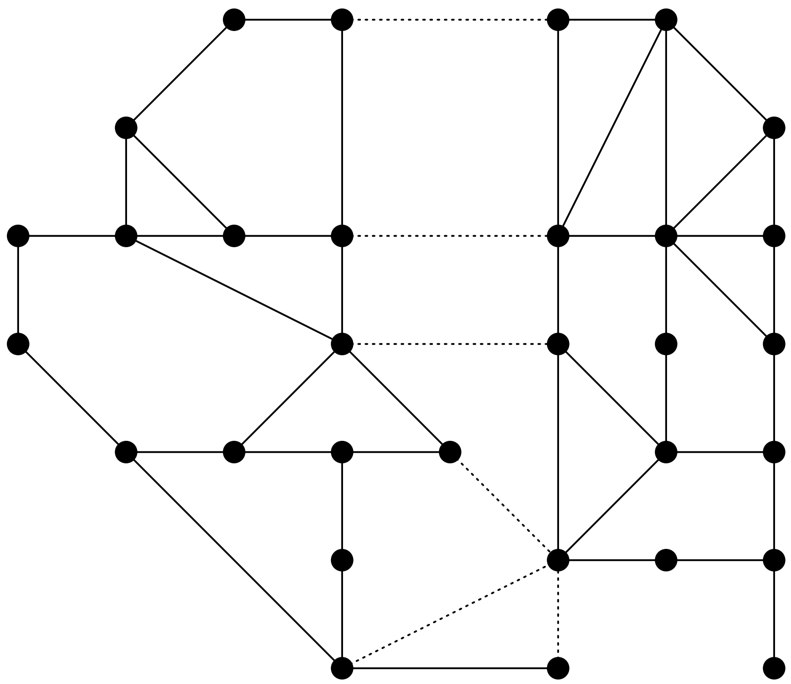


Figure 6.2: A mesh with 31 nodes and 50 edges.

and is studied by Souza, Keunings, Wolsey and Zone [SKWZ92]. One popular method for the solution of finite elements equations is the frontal technique. Given a finite element mesh, the complexity of the frontal method is proportional to the square of the so called frontwidth. Thus, it is important to find an ordering of the elements minimizing the frontwidth. Souza, Keunings, Wolsey and Zone suggest in [SKWZ92] an algorithmic approach to finding good orderings based on divide-and-conquer strategy that defines a series of Equicut Problems.

In Figure 6.2 we show an example of a graph derived from a mesh of finite elements. The dotted lines show an optimal solution of the Equicut Problem with $c \equiv \mathbf{1}$.

Let us define a polytope associated with the Equicut Problem. Given an instance (G) of the Equicut Problem, we introduce the variable $x \in \{0,1\}^E$ corresponding to characteristic vectors of 2-multicuts of (G, f, F). The Equicut Polytope is then defined as

$$MC^{=}(G, 2, \mathbf{1}, (\lfloor \tfrac{|V|}{2} \rfloor, \lceil \tfrac{|V|}{2} \rceil)) := \operatorname{conv}\{x \in \mathbb{R}^E \mid \begin{array}{l} x = \chi^{\delta(S_1,S_2)} \text{ for some valid} \\ \text{2-multicut } \delta(S_1,S_2) \text{ of} \\ G, 2, \mathbf{1} \text{ and } (\lfloor \tfrac{|V|}{2} \rfloor, \lceil \tfrac{|V|}{2} \rceil)\}. \end{array}$$

Conforti, Rao and Sassano investigate in [CRS90] and [CRS90a] the dimension of the polytope $MC^{=}(G, 2, \mathbf{1}, (\lfloor \frac{|V|}{2} \rfloor, \lceil \frac{|V|}{2} \rceil))$ and show that some facet defining

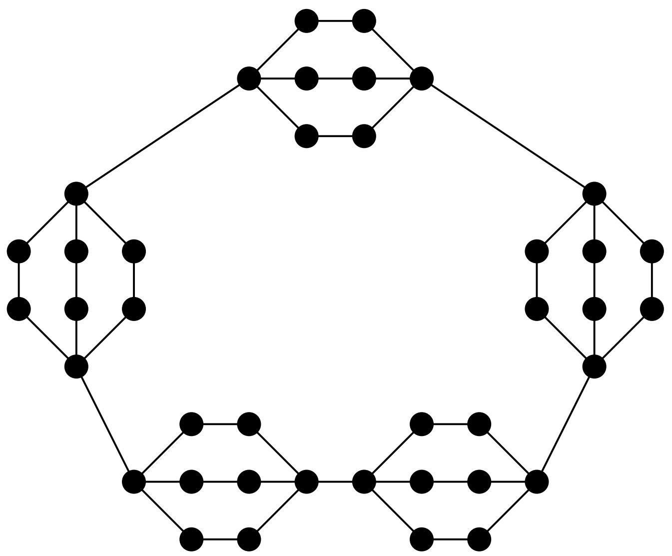


Figure 6.3: A (5,3,3)-path-block-cycle.

inequalities for the Maxcut polytope also define facets of $MC^{=}(G, 2, \mathbf{1}, (\lfloor\frac{|V|}{2}\rfloor, \lceil\frac{|V|}{2}\rceil))$. Moreover, they prove that facet defining inequalities for the complete graph remain facet inducing for polytopes obtained by removing some edges not present in the support of the inequality. An important class of facet defining inequality of the $MC^{=}(G, 2, \mathbf{1}, (\lfloor\frac{|V|}{2}\rfloor, \lceil\frac{|V|}{2}\rceil))$ is the class of *Cycle Inequalities*.

Theorem 6.3.1 *Let C be a cycle of the complete graph K_{2p+1} with $|V(C)| = p+2$ and $p \geq 3$. Then, the inequality*

$$\sum_{e \in C} x_e \geq 2$$

defines a facet of $MC^{=}(K_{2p+1}, 2, \mathbf{1}, (p, p+1))$. ■

Souza and Laurent present in [SL91] some other classes of facet defining inequalities of $MC^{=}(G, 2, \mathbf{1}, (\lfloor\frac{|V|}{2}\rfloor, \lceil\frac{|V|}{2}\rceil))$ such as the *Path-Block-Cycle Inequalities* and the *Suspended Trees Inequalities.* Now, we introduce the Path-Block-Cycle Inequalities.

A (r, q)*-path-block* is a graph with two special nodes s and d (for source and destination) joined by r node disjoint paths of length q. A (t, r, q)*-path-block-cycle* is a graph composed by t pairwise node disjoints (r, q)-path-blocks, where for all $i = 1, \ldots, t-1$, the i-th destination is adjacent to the $(i+1)$-th source and the destination of block t is adjacent to the source of the first block. In Figure 6.3 we show an example of a path-block-cycle.

Theorem 6.3.2 *Let $E_1 \cup E_2$ be a (t, r, q)-path-block-cycle, where E_1 is the set of edges internal to the blocks and E_2 the edges linking nodes of different path-blocks. Then, the inequality*

$$\sum_{e \in E_1} x_e + r \sum_{e \in E_2} x_e \geq 2r$$

defines a facet of $MC^{=}(K_{2p+1}, 2, \mathbf{1}, (\lfloor \frac{|V|}{2} \rfloor, \lceil \frac{|V|}{2} \rceil))$. ■

Computational results on solving problems arising from the application described in [SKWZ92] using a Branch and Cut Algorithm are described in [So93].

6.4 Other Multicut Problems

Grötschel and Wakabayashi investigate in [GW89] several applications of the multicut problem MC with $m = |V|$, $f \equiv \mathbf{0}$, $F \equiv \mathbf{0}$ and $d \equiv \mathbf{0}$. The problem is $\mathcal{NP}$-hard for complete graphs (cf. [Wa86]). The applications arise in several areas such as zoology, economics and sociology. In all applications it is desired to divide a set of objects into clusters (or cliques, in the case presented in the paper, since the ground graph is complete) such that "similar" objects are in the same cluster. No restriction on the number of clusters or on the number of elements in each set of the partition is made. Some examples solved by Grötschel and Wakabayashi are classification of cetacea and wild cats, classification of companies and microcomputers and votes of all member states of the UNO.

In [GW90] several classes of facet defining inequalities of a polytope associated with the problem can be found. Let us define the variable $x \in \{0, 1\}^E$ associated with multicuts of G. Then, we define the polytope

$$MC^{\leq}(G, m, \mathbf{0}, \mathbf{0}) := \{x \in \mathbb{R}^E \mid \; x = \chi^{\delta(S_1, \ldots, S_k)} \text{ for some multicut } \delta(S_1, \ldots, S_k) \text{ of } G\}.$$

Theorem 6.4.1 *(a) For every three different nodes $i, j, k \in V$ the* Triangle Inequality

$$x_{ij} + x_{jk} - x_{ik} \leq 1$$

defines a facet of $MC^{\leq}(K_n, n, \mathbf{0}, \mathbf{0})$.

(b) For every two disjoint nonempty subsets S and T of V the 2-Partition Inequality *induced by S and T,*

$$\sum_{e \in [S:T]} x_e - \sum_{e \in E(S)} x_e - \sum_{e \in E(T)} x_e \leq \min\{|S|, |T|\}$$

is valid for $MC^{\leq}(K_n, n, \mathbf{0}, \mathbf{0})$. It is facet defining if and only if $|S| \neq |T|$.

(c) *For every cycle C of length at least 5, and its set $\overline{C}$ of 2-chords, the* 2-Chorded Cycle Inequality

$$\sum_{e \in C} x_e - \sum_{e \in \overline{C}} x_e \leq \lfloor \frac{|C|}{2} \rfloor$$

is valid for $MC^{\leq}(K_n, n, \mathbf{0}, \mathbf{0})$. It is facet defining if and only if $|C|$ is odd. ■

In [GW89] and [GW90] other classes of inequalities and some separation strategies are provided. Instances of complete graphs up to 158 nodes of real world applications are solved to optimality.

Another investigation of the polytope $MC^{\leq}(G, |V|, \mathbf{0}, \mathbf{0})$ is given by Chopra and Rao in [CR91]. Actually, the authors investigate three variants of the multicut polytope, namely, given an integer $k \leq |V|$, $MC^{\leq}(G, k, \mathbf{0}, \mathbf{0})$ (as defined above), $MC^{=}(G, k, \mathbf{0}, \mathbf{0})$ and $MC^{\geq}(G, k, \mathbf{0}, \mathbf{0})$, where

$$\begin{aligned} MC^{=}(G, k, \mathbf{0}, \mathbf{0}) := \text{conv}\{x \in \mathbb{R}^E \mid \ & x = \chi^{\delta(S_1, \ldots, S_k)} \text{ for some } k\text{-multicut} \\ & \delta(S_1, \ldots, S_k) \text{ of } G\}. \\ MC^{\geq}(G, k, \mathbf{0}, \mathbf{0}) := \text{conv}\{x \in \mathbb{R}^E \mid \ & x = \chi^{\delta(S_1, \ldots, S_j)} \text{ for some multicut} \\ & \delta(S_1, \ldots, S_j) \text{ of } G, j \geq k\}. \end{aligned}$$

Some generalizations of the inequalities presented by Grötschel and Wakabayashi in [GW90] for complete graphs are provided for general graphs. Moreover, new classes of valid and facet defining inequalities for the three polytopes are presented.

An interesting investigation of small dimension multicut polytopes is presented by Deza, Grötschel and Laurent in [DGL91]. Complete and non redundant descriptions for the polytopes associated with complete graphs of 4 and 5 nodes are given. Several new classes of facet defining inequalities arise, some of them the authors generalize for higher dimension polytopes. An example are the so called *Casserole Inequalities*. Using a computer program, the authors show that the inequality

$$2x_{12} + x_{23} + x_{24} + x_{25} + x_{34} + x_{35} + x_{45} \geq 5$$

is facet defining for $MC^{=}(K_5, 3, \mathbf{0}, \mathbf{0})$ and for $MC^{\geq}(K_5, 3, \mathbf{0}, \mathbf{0})$. Figure 6.4 shows this casserole inequality.

A generalization of this inequality is given in the following theorem.

Theorem 6.4.2 *The Casserole Inequality*

$$\sum_{1 \leq i < j \leq n-1} x_{ij} + 2x_{1n} \geq \binom{n-1}{2} - 1$$

defines a facet of the polytope $MC^{\geq}(K_n, n-2, \mathbf{0}, \mathbf{0})$. ■

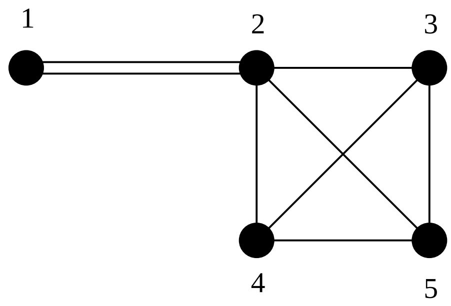


Figure 6.4: A Casserole Inequality.

6.5 The Node Weighted Multicut Problem

In some applications discussed in the literature some restrictions on the size of the partition or on the number of elements in each partition set must be satisfied. Moreover, some partition sets can be more attractive for some items, motivating costs on the node variables also. We describe now two applications of a multicut problem involving weights on the nodes and constrained partition capacities.

Johnson, Mehrotra and Nemhauser present in [JMN91] an investigation for the following Multicut Problem. We are given a graph $G = (V, E)$, nonnegative weights $f \in \mathbb{N}^V$, edge costs c_e, $e \in E$ and integers m (define $M := \{1, \ldots, m\}$) and K. The problem they consider is $\text{MC}^{\leq}$ where $F_k = K$ for all $k \in M$ and $d \equiv \mathbf{0}$. This problem arises in an application concerning compiler design. A compiler consists of several modules, where each module is a set of procedures or subroutines with its corresponding memory requirement. The modules must be combined to form clusters of restricted storage capacity. Modules assigned to different clusters cause high communication costs, because it can involve swapping of memory. The objective of the compiler construction is to assign the modules to clusters so that the storage bound is satisfied and the total communication cost between modules in different clusters is minimized. They model this problem representing the modules as nodes of a graph and the communication between modules as edges of the graph. The solution of the corresponding Multicut Problem provides an optimal assignment from the modules to the clusters. Johnson, Mehrotra and Nemhauser use column generation to solve problems with up to 61 nodes and 187 edges. For most problems they are able to find an optimal solution.

Another application for this problem appears in [We92]. In his thesis, Weismantel investigates the Placement Problem in VLSI. We are given a set of components of a chip and a master divided into base cells, where these components must be placed. Moreover a list of nets is given. We want to find a placement of the components on the master without overlapping and guaranteeing minimal realization costs (e.g. minimum routing length, minimum number of vias, and so on). The author suggests decomposition approaches to treat the problem. The base cells are grouped in a small number of clusters and the problem is then reduced to decide to which cluster of base cells a component must be assigned. This problem can be modelled as a Multicut Problem on Graphs, where f represents the area of the components, F the area of the clusters and the graph $G = (V, E)$ represents the affinity between the components. Weismantel presents a polyhedral investigation for a polytope associated with this multicut problem.

Given an instance (G, m, f, F) of the Multicut Problem on Graphs we introduce the variables $x_{ik} \in \{0,1\}^{V \times M}$ with the interpretation $x_{ik} = 1$ if and only if component i is assigned to cluster k. Moreover, we introduce variables $y_e \in \{0,1\}^E$ with the interpretation $y_{ij} = 1$ if components i and j are assigned to different components. Let us define $B_k(x) := \{i \in V \mid x_{ik} = 1\}$. The Multicut Polytope can then be defined as follows.

$$MC^{\leq}(G, m, f, F) := \{(x, y) \in \mathbb{R}^{V \times M} \times \mathbb{R}^E \mid y = \chi^{\delta(B_1(x), \ldots, B_m(x))} \text{ for some valid multicut } \delta(B_1(x), \ldots, B_m(x))\}.$$

We show in the next theorem a result on facet defining inequalities for the polytope $MC^{\leq}(G, n, f, F)$.

Theorem 6.5.1 *Suppose we are given an instance (G, m, f, F) of the Multicut Problem on Graphs $(G = (V, E), m, f, F)$, such that V defines a minimal cover with respect to F_{k_0} for some $k_0 \in M$. Moreover, suppose that E induces a tree spanning $D \cup \{d\} \subseteq V$, $|D| \geq 2$, and $\sum_{i \in D \cup \{d\}} f_i \leq F_k$ for all $k \in M \setminus \{k_0\}$. Then, the* Cover Tree Inequality

$$\sum_{i \in V} x_{ik_0} + \sum_{i \in D} \sum_{k \in M \setminus \{k_0\}} x_{ik} - \sum_{e \in E} y_e \leq |V| - 1$$

defines a facet of $MC^{\leq}(G, m, f, F)$. ■

Ferreira, Martin, Souza, Weismantel and Wolsey investigate in [FMSWW93] the same problems discussed in [JMN91] and [We92] using a cutting plane based algorithm. This research is taking place in parallel with the preparation of this thesis, and many results we achieved for the Multicut Problem on Hypergraphs presented in the next chapters are inspired by the discussions with the others authors. In [FMSWW93] the authors present several classes of valid and facet defining inequalities of $MC^{\leq}(G, m, f, F)$. In the next theorem we show one of these results. We formulate the inequality using only the y variables as defined above.

Theorem 6.5.2 *Let be given an instance $(G = (V, E), m, f, F)$ of the Multicut Problem on Graphs. Let $S \subseteq V$ define a minimal cover with respect to all $F_k, k \in M$, and denote as λ the cardinality of a minimum cut in the subgraph $G(V') = (V', E')$. The* Minimum Cut Inequality *is defined as*

$$\sum_{e \in E'} y_e \geq \lambda.$$

The inequality is valid for $MC^{\leq}(G, m, f, F)$. *We distinguish the following cases:*

- $\lambda = 1$.
  The Minimum Cut Inequality defines a facet of $MC^{\leq}(G', m, f, F)$ *if and only if* E' *is a spanning tree.*

- $\lambda = 2$.
  The Minimum Cut Inequality defines a facet of $MC^{\leq}(G', m, f, F)$ *if and only if for every cycles* $C_1, C_2 \subseteq E'$, *such that* $C_1 \cap C_2 \neq \emptyset$ *then* $|C_1 \cap C_2| \geq 3$.

- $\lambda \geq 3$.
  The Minimum Cut Inequality does not define a facet of $MC^{\leq}(G', m, f, F)$.

■

In [FMSWW93a] the authors apply a cutting plane based algorithm for solving problem instances coming from both applications described in this section.

In the next chapter we generalize some inequalities presented in [FMSWW93] for the case of the Multicut Problem on Hypergraphs.

Chapter 7

A Polyhedral Investigation of the Multicut Problem on Hypergraphs

In Chapter 2 we presented the Multicut Problem on Hypergraphs (short MPH) and how it arises in the application we investigate in this thesis. The aim of this chapter is to study a polytope associated with this problem. The inequalities described in this chapter are used by the Branch and Cut Algorithm we implemented for the problem. We discuss in Chapter 9 the implementation details of this algorithm.

This chapter is organized as follows. In Section 8.1 we introduce the Multicut Polytope on Hypergraphs and determine its dimension. We discuss in Section 8.2 some initial results about facet defining inequalities. In particular, we show which facet defining inequalities of the Multiple Knapsack Polytope are inherited. Sections 8.3 to 8.6 are devoted to valid and facet defining inequalities involving edge variables. In each section we discuss necessary and sufficient conditions such that the inequalities are valid or facet defining for the polytope.

7.1 The Multicut Polytope on Hypergraphs

We recall here the 0/1 programming formulation of the MPH from Chapter 2.

Suppose we are given a hypergraph $G = (N, \mathcal{Z})$, where $\mathcal{Z} = \{T_1, \ldots, T_z\}$. We set $Z = \{1, \ldots, z\}$ and use, for simplicity, the notation $t \in Z$ for an edge (edges are we also called *nets*) instead of $T_t \in \mathcal{Z}$. Moreover, we are given an integer m ($M = \{1, \ldots, m\}$) and vectors $f \in \mathbb{N}^N$, $g \in \mathbb{N}^Z$ and $F, S \in \mathbb{N}^M$. We say that a partition $(B_1, \ldots, B_j)$ of N is *valid* if $j \leq m$, $\sum_{v \in B_k} f_v \leq F_k$ for all $k = 1, \ldots, j$ and $\sum_{e \in \delta(B_k)} g_e \leq S_k$ for all $k = 1, \ldots, j$. Given a partition $(B_1, \ldots, B_j)$ of V, we call the set $\delta(B_1, \ldots, B_j)$ a *multicut*. Each element B_i is called a *shore*. If the partition $(B_1, \ldots, B_j)$ is valid, then the multicut $\delta(B_1, \ldots, B_j)$ is called a *valid* *multicut.*

The problem we are interested in is the following. Given is a hypergraph G with node weights f and edge weights g. It is also given a natural m corresponding to the maximum cardinality of a partition, a vector F with node-capacities and a vector S with edge-capacities corresponding to each part. The problem is to find a valid multicut $\delta(B_1, \ldots, B_j)$, i.e., the sum of the weights of the nodes in B_k is less than or equal to the corresponding node capacity for all k and the sum of the weights of the edges in $\delta(B_k)$ is less than or equal to the edge-capacity of the shore for all k. Furthermore, we want to find a valid multicut that minimizes a given cost function.

Formally, the problem is the following.

> **Problem**: Multicut Problem on Hypergraphs MPH.
> **Instance**: A hypergraph $G = (N, Z)$, an integer m ($M = \{1, \ldots, m\}$), vectors $f \in \mathbb{N}^N, F \in \mathbb{N}^M, g \in \mathbb{N}^Z, S \in \mathbb{N}^M$ and cost functions $c : Z \times M \longrightarrow \mathbb{R}$ and $d : N \times M \longrightarrow \mathbb{R}$.
> **Question**: Find a valid multicut $\delta(B_1, \ldots, B_j)$ that minimizes $\sum_{k=1}^{j} \sum_{t\in\delta(B_k)} c(t, k) + \sum_{k=1}^{j} \sum_{i\in B_k} d(i, k)$.

In the following we present a formulation of MPH using 0/1 variables. To this end, let us introduce the 0/1 variables $x = (x_{ik})$ and $y = (y_{tk})$ with the following interpretation. Given a multicut $\delta(B_1, \ldots, B_j)$,

$$x_{ik} = \begin{cases} 1 & \text{if } i \in B_k, \\ 0 & \text{otherwise.} \end{cases}$$

$$y_{tk} = \begin{cases} 1 & \text{if } T_t \cap B_k \neq \emptyset \text{ and } T_t \not\subseteq B_k, \\ 0 & \text{otherwise.} \end{cases}$$

Using these variables, the 0/1 integer programming is given as follows.

$$\begin{array}{lllll}
 & \min & \sum_{i\in N} \sum_{k\in M} d_{ik}x_{ik} + \sum_{t\in Z} \sum_{k\in M} c_{tk}y_{tk} & & \\
 & s.t. & \sum_{k\in M} x_{ik} = 1 & \text{for all } i \in N, & (1)\\
 & & \sum_{i\in N} f_i x_{ik} \leq F_k & \text{for all } k \in M, & (2)\\
 & & \sum_{t\in Z} g_t y_{tk} \leq S_k & \text{for all } k \in M, & (3)\\
(8.1.1) & & x_{ik} + x_{jl} - y_{tl} \leq 1 & \text{for all } t \in Z, i, j \in T_t (i \neq j) \text{ and} & \\
 & & & \text{for all } k, l \in M (k \neq l), & (4)\\
 & & \sum_{i\in T_t} x_{ik} + y_{tk} \leq |T_t| & \text{for all } t \in Z, k \in M, & (5)\\
 & & y_{tk} - \sum_{i\in T_t} x_{ik} \leq 0 & \text{for all } t \in Z, k \in M, & (6)\\
 & & x_{ik} \in \{0, 1\} & \text{for all } i \in N, k \in M, & (7)\\
 & & y_{tk} \in \{0, 1\} & \text{for all } t \in Z, k \in M. & (8)
\end{array}$$

For more details about this formulation, see Chapter 2. We apply, as pointed out in Chapter 2, a transformation on the objective function in order to eliminate the equations (1) (we denote by d' the modified objective function). Then, we obtain an equivalent formulation where (1) is substituted by the inequalities

$$\sum_{k\in M} x_{ik} \le 1 \text{ for all } i \in N. \tag{1'}$$

In the application we describe in Chapter 2 the objective function values for the edge variables y_{tk} are all non negative. In this case, it is easy to see that we can drop out constraints (5) and (6), since all optimal solutions from (8.1.1) are also optimal solutions of the problem where these constraints are eliminated. In (8.1.2) we give a formulation of the problem we investigate.

$$(8.1.2)\quad \begin{array}{ll} \min & \sum_{i\in N}\sum_{k\in M} d'_{ik}x_{ik} + \sum_{t\in Z}\sum_{k\in M} c_{tk}y_{tk} \\ s.t. & (1'), (2), (3), (4), (7) \text{ and } (8). \end{array}$$

With problem (8.1.2) we associate the Multicut Polytope on Hypergraphs. This is denoted by MCH and defined as follows.

$$MCH(N\times M, Z\times M, f, F, g, S) := \text{conv}\{ \begin{array}{l} (x,y) \in \mathbb{R}^{N\times M} \times \mathbb{R}^{Z\times M} \mid \\ (x,y) \text{ satisfies} \\ (1'), (2), (3), (4), (7) \text{ and } (8) \quad \}. \end{array}$$

Analogously to our definition of the polytope associated with the MKP, we allow the definition of subpolytopes associated with subinstances of MPH. Given $T \subseteq N\times M$ and $R \subseteq Z \times M$ define

$$MCH(T, R, f, F, g, S) := \text{conv}\{ \begin{array}{ll} (x,y) \in \mathbb{R}^T \times \mathbb{R}^R \mid & \\ \sum_{k:(i,k)\in T} x_{ik} \le 1 & \text{for all } i \in N, \\ \sum_{i:(i,k)\in T} f_i x_{ik} \le F_k & \text{for all } k \in M, \\ \sum_{t:(t,k)\in R} g_t y_{tk} \le S_k & \text{for all } k \in M, \\ x_{ik} + y_{jl} - y_{tk} \le 1 & \text{for all } t \in Z, i, j \in T_t \\ & (i \ne j), k, l \in M (k \ne l) \\ & \text{and } (i,k), (j,l) \in T, \\ & (t,k) \in R, \\ x_{ik} \in \{0,1\} & \text{for all } (i,k) \in T, \\ y_{tk} \in \{0,1\} & \text{for all } (t,k) \in R\}. \end{array}$$

Lemma 7.1.1 *The polytope $MCH(N\times M, Z\times M, f, F, g, S)$ is full dimensional if and only if $f_i \le F_k$ for all $i \in N$, $k \in M$ and $g_t \le S_k$ for all $t \in Z$, $k \in M$.*

Proof. Let us prove that if the conditions of the lemma are satisfied then the polytope is full dimensional. Suppose that an equation $ax + by = \alpha$ exists such that $MCH(N\times M, Z\times M, f, F, g, S) \subseteq \{(x,y) \in \mathbb{R}^{N\times M} \times \mathbb{R}^{Z\times M} \mid ax + by = \alpha\}$. We show that $a \equiv \mathbf{0}$ and $b \equiv \mathbf{0}$.

First note that $(\mathbf{0}, \mathbf{0})$ and $(e_{ik}, \mathbf{0})$ for all $i \in N$, $k \in M$ are elements of $MCH(N\times M, Z\times M, f, F, g, S)$ since $f_i \le F_k$ for all $i \in N$, $k \in M$. Hence,

$a_{ik} = 0$ for all $i \in N$, $k \in M$. Moreover, $(\mathbf{0}, e_{tk})$ for all $t \in Z$, $k \in M$ are also in $MCH(N \times M, Z \times M, f, F, g, S)$ since $g_t \leq S_k$ for all $t \in Z$, $k \in M$. Thus, $b_{tk} = 0$ for all $t \in Z$, $k \in M$.

Conversely, suppose that there exists some $i \in N$, $k \in M$ such that $f_i > F_k$. Then, $MCH(N \times M, Z \times M, f, F, g, S) \subseteq \{(x, y) \in \mathbb{R}^{N \times M} \times \mathbb{R}^{Z \times M} \mid x_{ik} = 0\}$. Analogously, if $g_t > S_k$ for some $t \in Z$, $k \in M$, then $MCH(N \times M, Z \times M, f, F, g, S) \subseteq \{(x, y) \in \mathbb{R}^{N \times M} \times \mathbb{R}^{Z \times M} \mid y_{tk} = 0\}$. This completes the proof. ■

We suppose throughout this chapter that $f_i \leq F_k$ for all $i \in N$, $k \in M$ and $g_t \leq S_k$ for all $t \in Z$, $k \in M$.

7.2 Some initial results

In this section we present some basic results for the polytope $MCH(N \times M, Z \times M, f, F, g, S)$. In the next lemma we show that the nonnegativity constraints for variables x and y define facets of the polytope.

Lemma 7.2.1 *(a) The inequality*

$$x_{ik} \geq 0$$

defines a facet of $MCH(N \times M, Z \times M, f, F, g, S)$ for all $i \in N$, $k \in M$.

(b) The inequality

$$y_{tk} \geq 0$$

defines a facet of $MCH(N \times M, Z \times M, f, F, g, S)$ for all $t \in Z$, $k \in M$.

Proof. (a) Let $i \in N$, $k \in M$. Suppose now that there exists an inequality $ax + by \leq \alpha$ that defines a facet of $MCH(N \times M, Z \times M, f, F, g, S)$ and $EQ(x_{ik} \geq 0) \subseteq EQ(ax + by \leq \alpha)$. Observe that, for all $j \in N$, $l \in M$ ($(j, l) \neq (i, k)$) the vectors $(\mathbf{0}, \mathbf{0})$ and $(e_{jl}, \mathbf{0})$ are elements of $EQ(x_{ik} \geq 0)$. Then, $a_{jl} = 0$ for all $j \in N$, $l \in M$ ($(j, l) \neq (i, k)$). Moreover, $(\mathbf{0}, e_{tl})$ is also contained in $EQ(x_{ik} \geq 0)$ for all $t \in Z$, $l \in M$. Thus, $b_{tl} = 0$ for all $t \in Z$, $l \in M$. The inequality $ax + by \leq \alpha$ is, then, a scalar multiple of $x_{ik} \geq 0$, and, therefore, it defines a facet of $MCH(N \times M, Z \times M, f, F, g, S)$.
(b) The proof is analogous to the case (a). ■

The constraints (1') from (8.1.2) also are facet defining in some cases. This issue is discussed next.

Theorem 7.2.2 *Given $i \in N$, the inequality*

$$\sum_{k \in M} x_{ik} \leq 1$$

defines a facet of $MCH(N \times M, Z \times M, f, F, g, S)$ if and only if for all $j \in N \setminus \{i\}$ and all $l \in M$ at least one of the conditions is satisfied:

- $\sum_{t \in T'} g_t \leq S_l$ *and there exists some* $l' \neq l$ *such that* $\sum_{t \in T'} g_t \leq S_{l'}$, *where* $T' := \{t \in Z \mid \{i,j\} \subseteq T_t\}$, *or*

- $f_i + f_j \leq F_l$.

Proof. Let us prove that under the conditions given in the theorem the inequality is facet defining. The inequality is clearly valid. Now suppose that there exist an inequality $ax + by \leq \alpha$ such that $EQ(\sum_{k \in M} x_{ik} \leq 1) \subseteq EQ(ax + by \leq \alpha)$.

Firstly, notice that for all $l \in M$, $(e_{il}, \mathbf{0}) \in EQ(\sum_{k \in M} x_{ik} \leq 1)$. Therefore, there exists a constant c such that $a_{il} = c$ for all $l \in M$.

Let $l \in M$ and observe that $(e_{il}, \mathbf{0}) \in EQ(\sum_{k \in M} x_{ik} \leq 1)$. Moreover, for all $t \in Z$, $l' \in M$, $(e_{il}, e_{tl'})$ is also an element of $EQ(\sum_{k \in M} x_{ik} \leq 1)$. Thus, $c = c + b_{tl'}$, and therefore, $b_{tl'} = 0$ for all $t \in Z$, $l' \in M$.

We prove now that $a_{jl} = 0$ for all $j \in N \setminus \{i\}$, $l \in M$. Given a pair (j, l), we know that at least one of the conditions given in the theorem is satisfied. Let us analyze the two cases.

- $\sum_{t \in T'} g_t \leq S_l$ and there exists some $l' \neq l$ such that $\sum_{t \in T'} g_t \leq S_{l'}$.
  Observe that $(e_{il'}, \mathbf{0})$ and $(e_{il'} + e_{jl}, \sum_{t \in T'} (e_{tl} + e_{tl'}))$ are elements of $EQ(\sum_{k \in M} x_{ik} \leq 1)$. Then, $c = c + a_{jl} + \sum_{t \in T'} (b_{tk} + b_{tl})$. Combining with the observations above, we have $a_{jl} = 0$.

- $f_i + f_j \leq F_l$.
  In this case, observe that $(e_{il}, \mathbf{0})$ and $(e_{il} + e_{jl}, \mathbf{0})$ are elements of $EQ(\sum_{k \in M} x_{ik} \leq 1)$. Thus, $c = c + a_{jl}$, and therefore, $a_{jl} = 0$.

Then, the inequality $ax + by \leq \alpha$ is a scalar multiple of the inequality $\sum_{k \in M} x_{ik} \leq 1$, and, therefore, it defines a facet of $MCH(N \times M, Z \times M, f, F, g, S)$.

We prove now the converse direction. Suppose that there exists a pair (j, l) such that $\sum_{t \in T'} g_t > S_l$ or $\sum_{t \in T'} g_t > S_{l'}$ for all $l' \in M \setminus \{l\}$ and $f_i + f_j > F_l$. Then, it is easy to see that the inequality $\sum_{k \in M} x_{ik} + x_{jl} \leq 1$ is valid (since $x_{jl} = 1$ implies $\sum_{k \in M} x_{ik} = 0$). But then, the original inequality is the sum of the inequality above and the constraint $-x_{jl} \leq 0$, and thus cannot define a facet of $MCH(N \times M, Z \times M, f, F, g, S)$. ■

Observe that the conditions given in Theorem 8.2.2 are easy to check and satisfied for practically relevant problem instances.

If we look at the problem formulation (8.1.2) more carefully, we observe that the problem is a combination of a Multiple Knapsack Problem on variables x_{ik} (constraints (1') and (2)), m independent Single Knapsack Problems (constraints (3)) and some linking constraints (constraints (4)). The first question one can ask is which results of the subproblems can be inherited, i.e., if we consider polytopes associated with these subproblems, which results on valid or facet defining inequalities carry over? Since the polytope we investigate is contained in such polytopes, we know that every valid inequality for some polytope associated with a subproblem is valid for MCH as well. In this section we show which facet defining inequalities maintain this property. Before doing that, let us introduce some notation. We call constraints (1') in (8.1.2) *SOS (Special Ordered Sets) constraints*, constraints (2) *node-knapsack constraints* and (3) *net-knapsack constraints*. Inequalities inherited from a Single Knapsack Polytope are called *individual*. An inequality is called *joint* if it comes from the MKP, and combines coefficients from at least two different knapsacks. If an inequality combines also coefficients of net variables, it is called *linking inequality*.

In the next theorem we characterize when the individual inequalities corresponding to Single Knapsack Polytopes associated with the node-knapsack constraints define facets of the Multicut Polytope on Hypergraphs.

Theorem 7.2.3 *Let $az \leq \alpha$ be an individual facet defining inequality of the polytope $SKP(N, f, F_k)$. Define $a'_{ik} = a_i$ for all $i \in N$ and $a'_{il} = 0$ for all $i \in N$, $l \in M \setminus \{k\}$. Then, the inequality $a'x \leq \alpha$ defines a facet of $MCH(N \times M, Z \times M, f, F, g, S)$ if and only if for all $i \in N$, $l \in M \setminus \{k\}$ there exists a solution $z' \in EQ(SKP(N, f, F_k), az \leq \alpha)$, such that $z'_i = 0$, $\sum_{t \in T(i,z')} g_t \leq S_k$ and $\sum_{t \in T(i,z')} g_t \leq S_l$, where $T(i, z') := \{t \in Z \mid \{i, j\} \subseteq T_t$ for some $j \in N$ with $z'_j = 1\}$.*

Proof. The inequality $a'x \leq \alpha$ is clearly valid. Let us prove that it is facet defining. Suppose that there exists an inequality $bx+cy \leq \beta$ that defines a facet of $MCH(N\times M, Z\times M, f, F, g, S)$ such that $EQ(MCH, a'x \leq \alpha) \subseteq EQ(MCH, bx+cy \leq \beta)$.

Let $z^* \in EQ(SKP, az \leq \alpha)$ and construct $x^* = (x^*_{il})$ in the following way.

$$x^*_{il} := \begin{cases} z^*_i & \text{if } l = k, \\ 0 & \text{otherwise.} \end{cases}$$

Observe that $(x^*, \mathbf{0})$ and (x^*, e_{tl}) are elements of $EQ(MCH,\ a'x \leq \alpha)$ for all $t \in Z$, $l \in M$. Then, we conclude that $c_{tl} = 0$ for all $t \in Z$, $l \in M$.

Now, let $i \in N$ and $l \in M \setminus \{k\}$. Let also $z' \in EQ(SKP(N, f, F_k), az \leq \alpha)$, such that $z'_i = 0$, $\sum_{t \in T(i,z')} g_t \leq S_k$ and $\sum_{t \in T(i,z')} g_t \leq S_l$ as required in the condition of the theorem. Define $x' = (x'_{jr})$ as follows.

$$x'_{jr} := \begin{cases} z'_j & \text{if } r = k, \\ 0 & \text{otherwise.} \end{cases}$$

Observe that $(x', \mathbf{0})$ and $(x' + e_{il}, \sum_{t \in T(i,z')}(e_{tk} + e_{tl}))$ are elements of $EQ(MCH,\ a'x \leq \alpha)$. In combination with the observations above, we can conclude that $b_{il} = 0$. The same argument holds for all $i \in N$, $l \in M\setminus\{k\}$. Finally, observe that $az \leq \alpha$ is facet defining for $SKP(N, f, F)$, and hence the inequality $bx + cy \leq \beta$ is a scalar multiple of $a'x \leq \alpha$, which completes the first part of the proof.

Conversely, suppose that the condition of the theorem is not satisfied, i.e., for some $i \in N$, $l \in M \setminus \{k\}$, $\sum_{t \in T(i,z')} g_t > S_k$ or $\sum_{t \in T(i,z')} g_t > S_l$ for all

$z \in EQ(SKP(N, f, F), az \leq \alpha)$ with $z_i = 0$. In this case, observe that the inequality $a'x + x_{il} \leq \alpha$ is valid, since $x_{il} = 1$ implies that $ax' < \alpha$. Thus, $a'x \leq \alpha$ is obtained by summing the inequality above and $-x_{il} \leq 0$ and, therefore, cannot define a facet of $MCH(N \times M, Z \times M, f, F, g, S)$. ■

It is not clear how to verify if the condition in the previous theorem is satisfied. However, for the special case in which for all $i \in N$, $\sum_{t:i \in T_t} g_t \leq S_k$ for all $k \in M$, the condition is trivially satisfied. This case occurs for most practical problems we considered.

The next theorem states that facet defining inequalities corresponding to Single Knapsack Polytopes associated with net-knapsack constraints define facets of $MCH(N \times M, Z \times M, f, F, g, S)$.

Theorem 7.2.4 *Let $az \leq \alpha$ be an individual facet defining inequality of the polytope $SKP(Z, g, S_k)$. Then, the inequality $a'y \leq \alpha$ defines a facet of $MCH(N \times M, Z \times M, f, F, g, S)$, where $a'_{tk} = a_t$ for all $t \in Z$ and $a'_{tl} = 0$ for all $t \in Z$, $l \in M \setminus \{k\}$.*

Proof. The inequality $a'y \leq \alpha$ is clearly valid. Let us prove that it is facet defining. Suppose that there exists an inequality $bx + cy \leq \beta$ that defines a facet of $MCH(N \times M, Z \times M, f, F, g, S)$ such that $EQ(MCH, a'y \leq \alpha) \subseteq EQ(MCH, bx + cy \leq \beta)$. Let $z^* \in EQ(SKP, az \leq \alpha)$, and construct $y^* = (y^*_{tl})$ as follows.

$$y^*_{tl} := \begin{cases} z^*_t & \text{if } l = k, \\ 0 & \text{otherwise.} \end{cases}$$

Observe that the vectors $(\mathbf{0}, y^*)$ and (e_{ik}, y^*) are elements of $EQ(MCH, a'y \leq \alpha)$ for all $i \in N$, $k \in M$. Then, $b_{ik} = 0$ for all $i \in N$, $k \in M$.

Moreover, $(\mathbf{0}, y^* + e_{tl})$ is also in $EQ(MCH, a'y \leq \alpha)$ for all $t \in Z$, $l \in M \setminus \{k\}$. Thus, $c_{tl} = 0$ for all $t \in Z$, $l \in M$. Considering all observations above, and since $az \leq \alpha$ is facet defining of $SKP(Z, g, S)$, we can conclude that $bx + cy \leq \beta$ is a scalar multiple of $a'y \leq \alpha$. This completes the proof. ■

Due to Theorems 8.2.3 and 8.2.4, there are exponentially many facet defining inequalities for the polytope $MCH(N \times M, Z \times M, f, F, g, S)$. In Section 3.4 we presented some of these inequalities. Moreover, in Sections 6.2.1 and 6.2.2 heuristic procedures to separate violated inequalities of the Single Knapsack Polytope are discussed. The next question is, "Do joint facets of the polytope $MKP(N, M, f, F)$ define facets for $MCH(N \times M, Z \times M, f, F, g, S)$ in general?" Unfortunately, the answer is no, as we show in Example 8.2.5.

Example 7.2.5 Consider the following instance of the MPH. $N = \{1, \ldots, 5\}$, $\mathcal{Z} = \{\{1, 2, 3, 4, 5\}\}$, $m = 2$, $f = (4, 4, 3, 3, 6)$, $F = (10, 10)$, $g = (1)$ and $S = (1, 1)$. The Extended Cover Inequality corresponding to $S = \{1, 2, 3\}$, $T = \{4, 5\}$, $k = 1$ and $l = 2$ (see Section 5.4),

$$x_{1,1} + x_{2,1} + x_{3,1} + x_{1,2} + x_{2,2} + x_{3,2} + x_{4,2} + x_{5,2} \leq 4$$

defines a facet of the polytope $MKP(N, M, f, F)$. It does not define a facet of $MCH(N \times M, Z \times M, f, F, g, S)$ since all points (x, y) satisfying the inequality at equality also satisfy $y_{1,1} = 1$ and $y_{1,2} = 1$. However, the two inequalities

$$x_{1,1} + x_{2,1} + x_{3,1} + x_{1,2} + x_{2,2} + x_{3,2} + x_{4,2} + x_{5,2} - y_{1,1} \leq 3,$$

$$x_{1,1} + x_{2,1} + x_{3,1} + x_{1,2} + x_{2,2} + x_{3,2} + x_{4,2} + x_{5,2} - y_{1,2} \leq 3$$

define facets of $MCH(N \times M, Z \times M, f, F, g, S)$. ■

A natural question that arises in Example 8.2.5 is how to "lift" the joint inequalities to facet defining inequalities of the polytope $MCH(N \times M, Z \times M, f, F, g, S)$. This is an interesting open question for which we could not obtain a satisfactory answer.

We concentrate, in the next sections, on valid and facet defining linking inequalities for $MCH(N \times M, Z \times M, f, F, g, S)$.

7.3 Net Inequalities

The first linking inequalities we investigate come from the constraints (4) in (8.1.2). We call these inequalities by *Net Inequalities*. Given a net $t \in Z$, $k \in M$ and a pair $i, j \in T_t$, $i \neq j$, we call the inequality

$$x_{ik} + \sum_{l \in M \setminus \{k\}} x_{jl} - y_{tk} \leq 1$$

the *Net Inequality* corresponding to t, k, i and j. We characterize in the next theorem when this inequality defines a facet of $MCH(N \times M, \{t\} \times M, f, F, g, S)$.

Theorem 7.3.1 *Given an instance $(G = (N, Z), M, f, F, g, S)$ of the MPH. Let $t \in Z$, $k \in M$ and a pair $i, j \in T_t$, $i \neq j$. The corresponding Net Inequality*

$$x_{ik} + \sum_{l \in M \setminus \{k\}} x_{jl} - y_{tk} \leq 1 \tag{8.3.1}$$

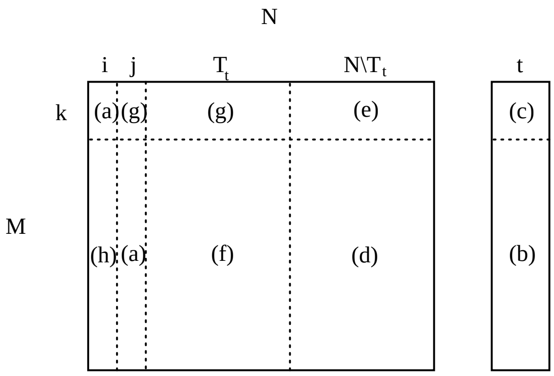
defines a facet of $MCH(N \times M, \{t\} \times M, f, F, g, S)$ if and only if the following conditions are satisfied

(i) $m \geq 3$ or $f_{i'} + f_j \leq F_l$ for all $i' \in T_t \setminus \{i, j\}$, $l \in M \setminus \{k\}$,

(ii) $f_{i'} + f_i \leq F_k$ for all $i' \in T_t \setminus \{i\}$,

(iii) $m \geq 3$ or $f_i + f_j \leq F_l$ for all $l \in M \setminus \{k\}$.

Proof. The inequality (8.3.1) is clearly valid. Suppose that there exists a facet defining inequality $ax + by \leq \alpha$ such that $EQ((8.3.1)) \subseteq EQ(ax + by \leq \alpha)$. We now prove that $ax+by \leq \alpha$ is a scalar multiple of (8.3.1) and, therefore, it defines a facet of the polytope $MCH(N \times M, \{t\} \times M, f, F, g, S)$.



• (a) $a_{ik} = a_{jl}$ for all $l \in M \setminus \{k\}$.

Firstly, observe that $(e_{ik}, \mathbf{0})$ and $(e_{jl}, \mathbf{0})$ are elements of $EQ((8.3.1))$ for all $l \in M \setminus \{k\}$. Thus, $a_{ik} = a_{jl}$ for all $l \in M \setminus \{k\}$.

• (b) $b_{tl} = 0$ for all $l \in M \setminus \{k\}$.

Now, observe that (e_{ik}, e_{tl}) is in $EQ((8.3.1))$ for all $l \in M \setminus \{k\}$. Then, $b_{tl} = 0$ for all $l \in M \setminus \{k\}$.

• (c) $b_{tk} = -a_{ik}$.

Moreover, given an $l \in M \setminus \{k\}$, $(e_{ik} + e_{jl}, e_{tk} + e_{tl})$ is also an element of $EQ((8.3.1))$. Combining the observations above we conclude that $a_{ik} = 2a_{ik} + b_{tk}$, and then, $b_{tk} = -a_{ik}$.

• (d) $a_{i'l} = 0$ for all $i' \in N \setminus T_t$, $l \in M \setminus \{k\}$.

Now, let $i' \in N \setminus T_t$, $l \in M \setminus \{k\}$. Since $i' \notin T_t$, $(e_{ik} + e_{i'l}, \mathbf{0})$ is in $EQ((8.3.1))$. Then, $a_{i'l} = 0$ for all $i' \in N \setminus T_t$, $l \in M \setminus \{k\}$.

• (e) $a_{i'k} = 0$ for all $i' \in N \setminus T_t$.

Let $l \in M \setminus \{k\}$ and consider that $(e_{i'k} + e_{jl}, \mathbf{0})$ is also in $EQ((8.3.1))$. Thus, $a_{i'k} = 0$ for all $i' \in N \setminus T_t$.

• (f) $a_{i'l} = 0$ for all $i' \in T_t \setminus \{i, j\}$ and $l \in M \setminus \{k\}$.

We analyze now the case where $i' \in T_t \setminus \{i, j\}$ and $l \in M \setminus \{k\}$. In this case, we use condition (i) of the theorem. Let us distinguish the two cases:

- $m \geq 3$.
  In this case, let $l' \in M \setminus \{k, l\}$ and observe that $(e_{ik} + e_{jl'} + e_{i'l}, e_{tk} + e_{tl} + e_{tl'})$ is an element of $EQ((8.3.1))$. Combining with the observations above, we conclude that $a_{i'l} = 0$.

- $f_{i'} + f_j \leq F_l$.
  In this case, consider that $(e_{jl} + e_{i'l}, \mathbf{0})$ is in $EQ((8.3.1))$, yielding $a_{i'l} = 0$.

Since at least one of the cases occurs for all $i' \in T_t \setminus \{i, j\}$, $l \in M \setminus \{k\}$, we conclude that $a_{i'l} = 0$ for all $i' \in T_t \setminus \{i, j\}$, $l \in M \setminus \{k\}$.
• (g) $a_{i'k} = 0$ for all $i' \in T_t \setminus \{i\}$.
Now, we consider $i' \in T_t \setminus \{i\}$ and prove that $a_{i'k} = 0$. It is due to condition (ii) It guarantees that $(e_{ik} + e_{i'k}, \mathbf{0})$ is an element of $EQ((8.3.1))$, and then, $a_{i'k} = 0$. It holds for all $i' \in T_t \setminus \{i\}$.
• (h) $a_{il} = 0$ for all $l \in M \setminus \{k\}$.
Finally, let $l \in M \setminus \{k\}$. We use condition (iii) of the theorem to prove that $a_{il} = 0$. Let us analyze the two cases:

- $m \geq 3$.
  In this case, let $l' \in M \setminus \{k, l\}$ and observe that $(e_{il} + e_{jl'}, e_{tl} + e_{tl'})$ is in $EQ((8.3.1))$. Combining with the observations above we conclude that $a_{il} = 0$. It holds for all $l \in M \setminus \{k\}$.

- $f_i + f_j \leq F_l$.
  In this case, observe that $(e_{il} + e_{jl}, \mathbf{0})$ is an element of $EQ((8.3.1))$. Thus, $a_{il} = 0$ for all $l \in M \setminus \{k\}$.

Considering all the cases described above we conclude that the inequality $ax + by \leq \alpha$ is a scalar multiple of (8.3.1), and therefore, it defines a facet of $MCH(N \times M, \{t\} \times M, f, F, g, S)$.
Now, we prove that the conditions given in the theorem are also necessary. We analyze the three cases in the following.

- $m = 2$ and there exists an $i' \in T_t \setminus \{i, j\}$ with $f_{i'} + f_j > F_l$ $(l \neq k)$.
  In this case, we prove that $EQ((8.3.1)) \subseteq \{(x, y) \in \mathbb{R}^{N \times M} \times \mathbb{R}^{\{t\} \times M} \mid x_{i'l} = 0\}$. Suppose it is not the case and let $(x', y') \in EQ((8.3.1))$ with $x'_{i'l} = 1$. Since condition (i) is not satisfied, we conclude that $x'_{jl} = 0$. But, then, if $x'_{ik} = 1$, then $y'_{tk} = 1$, contradicting the assumption that $(x', y') \in EQ((8.3.1))$.

- there exists some $i' \in T_t \setminus \{i\}$ with $f_{i'} + f_i > F_k$.
  In this case, we prove that $EQ((8.3.1)) \subseteq \{(x, y) \in \mathbb{R}^{N \times M} \times \mathbb{R}^{\{t\} \times M} \mid x_{i'k} = 0\}$. Suppose it is not the case and let $(x', y') \in EQ((8.3.1))$ with $x'_{i'k} = 1$. Since (ii) is not satisfied, $x'_{ik} = 0$. But, then, if $i' \neq j$, $x'_{jl} = 1$ for some $l \in M \setminus \{k\}$ implies $y_{tk} = 1$, contradicting the assumption that $(x', y') \in EQ((8.3.1))$. In the other case $(i' = j)$, (x', y') cannot be in $EQ((8.3.1))$.

- $m = 2$ and $f_i + f_j > F_l$ $(l \neq k)$.
  In this case, we prove that $EQ((8.3.1)) \subseteq \{(x, y) \in \mathbb{R}^{N \times M} \times \mathbb{R}^{\{t\} \times M} \mid x_{il} = 0\}$. Suppose it is not the case and let $(x', y') \in EQ((8.3.1))$ with $x'_{il} = 1$. Since the condition is not satisfied, $x'_{jl} = 0$, contradicting the assumption that $x' \in EQ((8.3.1))$

This completes the proof. ■

Observe that, for practically relevant problem instances, the conditions in the theorem will be always satisfied. Moreover, it is easy to verify whether it is the case or not.

7.4 Tree Inequalities

In this section we present another class of valid linking inequalities for the Multicut Polytope on Hypergraphs: the *Tree Inequalities*. We present the general idea of the inequality and, thereafter, characterize when these inequalities are valid and facet inducing of some polytopes.

Suppose we are given an instance $(G = (N, Z), M, f, F, g, S)$ of the MPH. Let $M' \subseteq M$, $l \in M'$ and $G' = (N', Z')$ be a subgraph of hypergraph G, such that N' is a minimal cover with respect to all knapsacks in M'. We call the inequality

$$\sum_{i \in N'} \sum_{k \in M'} x_{ik} - \sum_{t \in Z'} \sum_{k \in M' \setminus \{l\}} y_{tk} \leq |N'| - 1$$

the *Tree Inequality* corresponding to G', M' and l. In the next theorem we characterize when this inequality is valid for $MCH(N \times M, Z \times M, f, F, g, S)$.

Theorem 7.4.1 *Let an instance $(G = (N, Z), M, f, F, g, S)$ of the MPH be given. Let $Z' \subseteq Z$, $M' \subseteq M$, $l \in M'$ and $N' \subseteq N$ such that N' is a minimal cover with respect to all knapsacks in M'. Then, the inequality*

$$\sum_{i \in N'} \sum_{k \in M'} x_{ik} - \sum_{t \in Z'} \sum_{k \in M' \setminus \{l\}} y_{tk} \leq |N'| - 1 \tag{8.4.1}$$

is valid for the polytope $MCH(N' \times M', Z' \times M', f, F, g, S)$ if and only if the hypergraph $G' = (N', Z')$ is connected.

Proof. Let us prove that if G' is connected, then (8.4.1) is valid. Suppose it is not the case and let (x', y') violate (8.4.1). Due to the SOS constraints, we can conclude that every item of N' is assigned to some knapsack in M'. Let $M^* := \{k \in M' \mid x'_{ik} = 1 \text{ for some } i \in N'\}$. Since N' is a cover with respect to all knapsack of M', $|M^*| \geq 2$. But, since G' is connected, there exists some net $t \in Z'$ such that t contains two items $i, j (i \neq j)$ assigned to two different knapsacks $k_1, k_2 \in M'$. Due to the net constraints, $y'_{tk_1} = 1$ and $y'_{tk_2} = 1$, contradicting the assumption that (x', y') violates (8.4.1).

Conversely, suppose that G' is not connected. Then, let $G'_1, \ldots, G'_v$ $(v \geq 2)$ be the components of G'. Set $N_1 := \{i \in N' \mid i \text{ is in component } G'_1\}$ and $N_2 := N' \setminus N_1$. Since G' is not connected, $(\sum_{i \in N_1} e_{ik} + \sum_{i \in N_2} e_{il}, \mathbf{0})$ is in $MCH(N' \times M', Z' \times M', f, F, g, S)$ and violates (8.4.1). ■

Now, we characterize when the Tree Inequality is facet defining. To this end we need a definition. A subgraph $G' = (N', Z')$ of the hypergraph G is called a

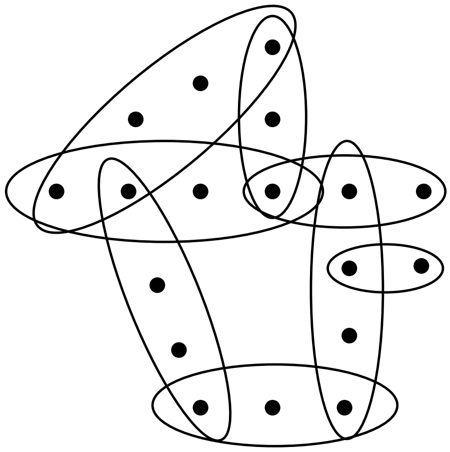


Figure 7.1: A tree.

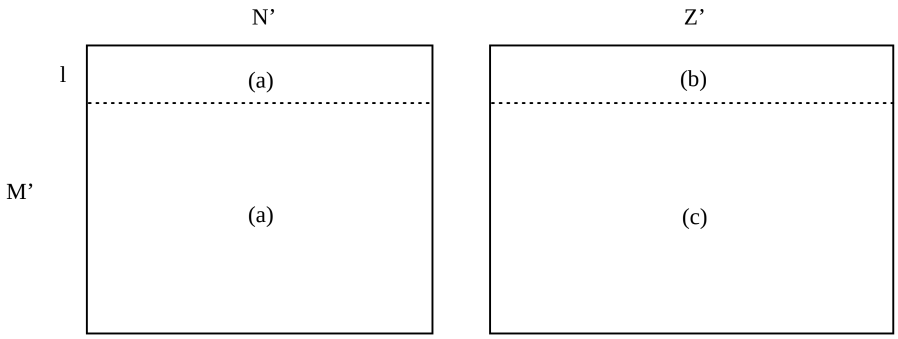
tree if it is connected and $G' - t$ is disconnected for all $t \in Z'$. In Figure 7.1 we show an example of a tree in a hypergraph. In the next theorem we prove that the Tree Inequalities are facet defining if and only if the underlying hypergraph is a tree, which justifies our choice for the name of the inequality.

Theorem 7.4.2 *Let be given an instance $(G = (N, Z), M, f, F, g, S)$ of the MPH. Let $Z' \subseteq Z$, $M' \subseteq M$, $l \in M'$ and $N' \subseteq N$ such that N' is a minimal cover with respect to all knapsacks in M'. Then, the inequality*

$$\sum_{i \in N'} \sum_{k \in M'} x_{ik} - \sum_{t \in Z'} \sum_{k \in M' \setminus \{l\}} y_{tk} \leq |N'| - 1 \tag{8.4.2}$$

defines a facet of the polytope $MCH(N' \times M', Z' \times M', f, F, g, S)$ if and only if the hypergraph $G' = (N', Z')$ is a tree.

Proof. Suppose that there exists a facet defining inequality $ax + by \leq \alpha$ such that $EQ((8.4.2)) \subseteq EQ(ax + by \leq \alpha)$. We show that $ax + by \leq \alpha$ is a scalar multiple of (8.4.2), and, therefore, (8.4.2) defines a facet of $MCH(N' \times M', Z' \times M', f, F, g, S)$.



- (a) $a_{ik} = c$ for all $i \in N'$, $k \in M'$.

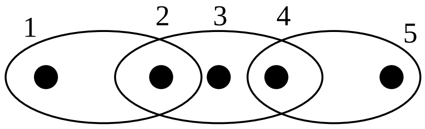


Figure 7.2: Example of a Tree Inequality.

Let $k \in M'$ and note that for all $i \in N'$, $(\sum_{j \in N' \setminus \{i\}} e_{jk}, \mathbf{0})$ is an element of $EQ((8.3.1))$. Then, there exists some constant c_k such that $a_{ik} = c_k$ for all $i \in N'$. Applying the same argument for some $k' \in M \setminus \{k\}$ we obtain $(|N'| - 1)c_k = (|N'| - 1)c_{k'}$, and therefore, $c_k = c_{k'}$. Thus, there exists some constant c such that $a_{ik} = c$ for all $i \in N'$, $k \in M'$.

• (b) $b_{tl} = 0$ for all $t \in Z'$.

Let $i \in N'$ and observe that $(\sum_{j \in N' \setminus \{i\}} e_{jl}, \mathbf{0})$ and $(\sum_{j \in N' \setminus \{i\}} e_{jl}, e_{tl})$ are elements of $EQ((8.4.2))$ for all $t \in Z'$. Thus, $(|N'| - 1)c = (|N'| - 1)c + b_{tl}$, and therefore, $b_{tl} = 0$ for all $t \in Z'$.

• (c) $b_{tk} = -c$ for all $t \in Z'$, $k \in M' \setminus \{l\}$.

Now, let $t \in Z'$. Since G' is a tree, $G' - t$ has at least two connected components $G'_1, \ldots, G'_b$ $(b \geq 2)$. Let $N_1 := \{i \in N' \mid i \text{ is in the component } G'_1\}$ and $N_2 := N \setminus N_1$. Since N_1 and N_2 are nonempty, and N' is a minimal cover with respect to all knapsack in M', we can conclude that for all $k \in M' \setminus \{l\}$, $(\sum_{i \in N_1} e_{ik} + \sum_{i \in N_2} e_{il}, e_{tk} + e_{tl})$ is an element of $EQ((8.4.2))$. Then, $(|N'| - 1)c = |N'|c + b_{tk}$, and thus, $b_{tk} = -c$ for all $k \in M' \setminus \{l\}$. The same argument applies to all $t \in Z'$.

The inequality $ax + by \leq \alpha$ is then a scalar multiple of (8.4.2), and, therefore, it defines a facet of $MCH(N' \times M', Z' \times M', f, F, g, S)$.

Conversely, we prove that if $G' = (N', Z')$ is not a tree, then the inequality (8.4.2) is not facet defining.

Suppose that G' is not a tree. If G' is not connected, then, due to Theorem 8.4.1 the inequality is not valid. Then, there must exist an edge $t \in Z'$ such that $G' - t$ is connected. But then, according to Theorem 8.4.1, the Tree Inequality corresponding to $G' - t$, M' and l is valid. Inequality (8.4.2) is the sum of the inequality above and $\sum_{k \in M' \setminus \{l\}} y_{tk} \geq 0$, and therefore, cannot be facet defining. ■

Example 7.4.3 Consider the following instance of the MPH, where $N = \{1, 2, 3, 4, 5\}$, $M = \{1, 2\}$, $f = (2, 2, 3, 3, 5)$, $F = (13, 14)$, $\mathcal{Z} = \{\{1, 2\}, \{2, 3, 4\}, \{4, 5\}\}$, $g = (1, 1, 1)$ and $S = (3, 3)$ as shown in Figure 7.2.

Then, the Tree Inequalities

$$\sum_{i=1}^{5} \sum_{k=1}^{2} x_{ik} - y_{1,1} - y_{2,1} - y_{3,1} \leq 4,$$

$$\sum_{i=1}^{5} \sum_{k=1}^{2} x_{ik} - y_{1,2} - y_{2,2} - y_{3,2} \leq 4,$$

define facets of the polytope $MCH(N \times M, Z \times M, f, F, g, S)$. ■

7.5 Cycle Inequalities

In this section we present another class of valid inequalities of the MPH: the *Cycle Inequalities*. In the case of graphs, cycles are 2-edge connected subgraphs, which is no longer true for hypergraphs. In this section we show a facet defining inequality that involves, in some extent, 2-connectivity.

Suppose we are given an instance $(G = (N, Z), M, f, F, g, S)$ of the MPH. Let $M' \subseteq M$, $l \in M'$ and $G' = (N', Z')$ be a subgraph of hypergraph G such that N' is a minimal cover with respect to all knapsacks in M'. Moreover, let Z_1 and Z_2 such that $Z' = Z_1 \cup Z_2$ and $Z_1 \cap Z_2 = \emptyset$. We call the inequality

$$2 \sum_{i \in N'} \sum_{k \in M'} x_{ik} - \sum_{t \in Z_1} \sum_{k \in M' \setminus \{l\}} y_{tk} - 2 \sum_{t \in Z_2} \sum_{k \in M' \setminus \{l\}} y_{tk} \leq 2(|N'| - 1)$$

the *Cycle Inequality* corresponding to G', M', l, Z_1 and Z_2. In the next theorem we investigate when this inequality is valid.

Theorem 7.5.1 *Let an instance $(G = (N, Z), M, f, F, g, S)$ of the MPH be given. Let $Z' = Z_1 \cup Z_2 \subseteq Z$ $(Z_1 \cap Z_2 = \emptyset)$, $M' \subseteq M$, $l \in M'$ and $N' \subseteq N$ such that N' is a minimal cover with respect to all knapsacks in M'. The corresponding Cycle Inequality,*

$$2 \sum_{i \in N'} \sum_{k \in M'} x_{ik} - \sum_{t \in Z_1} \sum_{k \in M' \setminus \{l\}} y_{tk} - 2 \sum_{t \in Z_2} \sum_{k \in M' \setminus \{l\}} y_{tk} \leq 2(|N'| - 1) \qquad (8.5.1)$$

is valid for the polytope $MCH(N' \times M', Z' \times M', f, F, g, S)$ if and only if $G' = (N', Z')$ is connected and $G' - t$ is connected for all $t \in Z_1$.

Proof. First we prove that if the condition is satisfied, then inequality (8.5.1) is valid. Suppose that there exists a $(x', y') \in MCH(N' \times M', Z' \times M', f, F, g, S)$ violating (8.5.1). Then, due to the SOS constraints, every element of N' is assigned to some knapsack in M'. Since N' is a cover for all knapsacks in M', and G' is connected, there exists an edge $t \in Z'$, such that $\{i, j\} \subseteq T_t$ and $x'_{ik_1} = 1$ and $x'_{jk_2} = 1$ for $k_1, k_2 \in M'$, $k_1 \neq k_2$. Due to the net constraints, $y'_{tk_1} = 1$ and $y'_{tk_2} = 1$. We distinguish the two possibilities for the edge t.

- $t \in Z_2$.
  In this case, (x', y') does not violate (8.5.1), contradicting our assumption.

- $t \in Z_1$.
  In this case, $G' - t$ is connected. Then, there must exist an edge $t' \in Z' \setminus \{t\}$ such that $\{i', j'\} \subseteq T_{t'}$ and $x'_{i'k'_1} = 1$ and $x'_{j'k'_2} = 1$ for $k'_1, k'_2 \in M'$, $k'_1 \neq k'_2$. Due to the net constraints, $y'_{t'k'_1} = 1$ and $y'_{t'k'_2} = 1$ contradicting our assumption that (x', y') violates (8.5.1).

Conversely, suppose that the condition is not satisfied. If G' is not connected, let N_1 be the nodes of some component of G' and $N_2 := N' \setminus N_1$. But, since N' is a minimal cover with respect to all knapsacks in M', and N_1 and N_2 are both nonempty, $(\sum_{i\in N_1} e_{ik} + \sum_{i\in N_2} e_{il}, \mathbf{0})$ is in $MCH(N' \times M', Z' \times M', f, F, g, S)$ for all $k \in M \setminus \{l\}$ and violates (8.5.1).

Now, suppose that there exists some edge $t \in Z_1$ such that $G' - t$ is not connected. Let N_1 be the nodes of some component of G' and $N_2 := N' \setminus N_1$. Then, since N' is a minimal cover with respect to all knapsacks in M', and N_1 and N_2 are nonempty, $(\sum_{i\in N_1} e_{ik} + \sum_{i\in N_2} e_{il}, e_{tk} + e_{tl})$ is an element of $MCH(N' \times M', Z' \times M', f, F, g, S)$ for all $k \in M \setminus \{l\}$ and violates (8.5.1). This completes the proof. ■

The next theorem characterizes when this inequality is facet defining. Before that, suppose we are given a hypergraph $G' = (N', Z')$, $M' \subseteq M$, $l \in M'$ and sets Z_1, Z_2 with $Z' = Z_1 \cup Z_2$ and $Z_1 \cap Z_2 = \emptyset$, such that the corresponding Cycle Inequality is valid. Observe that if there exist different three edges $t, t', t'' \in Z_1$ such that $G' - \{t, t'\}$ and $G' - \{t, t''\}$ are not connected, then $G' - \{t', t''\}$ is also not connected. To prove this statement, consider the components $G'_1 = (N'_1, Z'_1)$ and $G'_2 = (N'_2, Z'_2)$ of $G' - \{t, t'\}$ (w.l.o.g we can suppose that the hypergraph has only 2 components) and suppose that the edge $t'' \in Z_1$. Then, let $G''_1 = (N''_1, Z''_1)$ and $G''_2 = (N''_2, Z''_2)$ be the components of $G' - \{t, t''\}$ (again, we can suppose w.l.o.g. that there is only two components). Now, we can conclude that $N''_1 \subseteq N'_1$ and, therefore $N''_2 = N'_2 \cup N'_1 \setminus N''_1$. Then, t' is the only edge between N'_2 and $N'_1 \setminus N''_1$ and t'' is the only edge between N''_1 and $N'_1 \setminus N''_1$. So, $G' - \{t', t''\}$ is not connected.

Theorem 7.5.2 *Let an instance $(G = (N, Z), M, f, F, g, S)$ of the MPH be given. Let $Z' = Z_1 \cup Z_2 \subseteq Z$ $(Z_1 \cap Z_2 = \emptyset)$, $M' \subseteq M$, $l \in M'$ and $N' \subseteq N$ such that N' is a minimal cover with respect to all knapsacks in M'. The corresponding Cycle Inequality,*

$$2\sum_{i\in N'}\sum_{k\in M'} x_{ik} - \sum_{t\in Z_1}\sum_{k\in M'\setminus\{l\}} y_{tk} - 2\sum_{t\in Z_2}\sum_{k\in M'\setminus\{l\}} y_{tk} \le 2(|N'| - 1) \qquad (8.5.2)$$

defines a facet of the polytope $MCH(N' \times M', Z' \times M', f, F, g, S)$ if and only if it is valid and the following two conditions are satisfied:

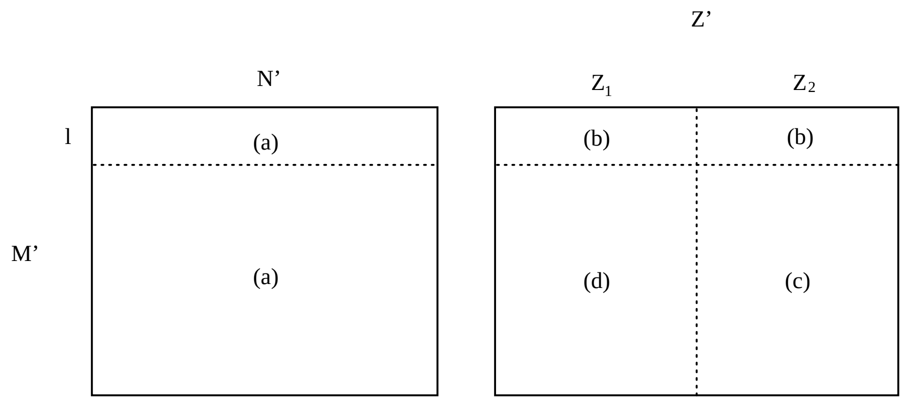
(i) for each $t \in Z_1$ there exist two edges $t', t'' \in Z_1 \setminus \{t\}$, $(t' \neq t'')$ such that the hypergraphs $G' - \{t, t'\}$ and $G' - \{t, t''\}$ are not connected (and consequently, $G' - \{t', t''\}$ is also not connected).

(ii) for each $t \in Z_2$, $G' - t$ is not connected.

Proof. We prove that if the conditions in the theorem are satisfied, then the inequality defines a facet of $MCH(N' \times M', Z' \times M', f, F, g, S)$. Suppose that there exists a facet defining inequality $ax + by \leq \alpha$ such that $EQ((8.5.2)) \subseteq EQ(ax + by \leq \alpha)$. We prove that this inequality is a scalar multiple of (8.5.2), and therefore, it is facet defining.

• (a) $a_{ik} = c$ for all $i \in N'$, $k \in M'$.

Let $i \in N'$, $k \in M'$ and notice that $(\sum_{j \in N' \setminus \{i\}} e_{jk}, \mathbf{0})$ is an element of $EQ((8.5.2))$, since N' is a minimal cover with respect to k. Moreover, for all $i' \in N' \setminus \{i\}$, $(\sum_{j \in N' \setminus \{i'\}} e_{jk}, \mathbf{0})$ is also in $EQ((8.5.2))$. Thus, $\sum_{j \in N' \setminus \{i\}} a_{jk} = \sum_{j \in N' \setminus \{i\}} a_{jk}$, and hence $a_{ik} = a_{i'k}$ for all $i' \in N' \setminus \{i\}$. The same argument applies to all $k \in M'$. Then, for all $k' \in M' \setminus \{k\}$, $(|N'| - 1)a_{ik} = (|N'| - 1)a_{ik'}$, and therefore, there exists some constant c such that $a_{ik} = c$ for all $i \in N'$, $k \in M'$.



• (b) $b_{tl} = 0$ for all $t \in Z'$.

Now observe that given $i \in N'$, $(\sum_{j \in N' \setminus \{i\}} e_{jl}, \mathbf{0})$ and $(\sum_{j \in N' \setminus \{i\}} e_{jl}, e_{tl})$ are elements of $EQ((8.5.2))$ for all $t \in Z'$. Thus, $(|N'| - 1)c = (|N'| - 1)c + b_{tl}$, and $b_{tl} = 0$ for all $t \in Z'$.

• (c) $b_{tk} = -c$ for all $t \in Z_2$, $k \in M' \setminus \{l\}$.

Let $t \in Z_2$. Due to condition (ii) $G' - t$ is not connected. Let N_1 be the node set of some component of $G' - t$ and $N_2 := N' \setminus N_1$. Since N' is a minimal cover for all knapsacks in M' and N_1 and N_2 are nonempty, $(\sum_{i \in N_1} e_{ik} + \sum_{i \in N_2} e_{il}, e_{tk} + e_{tl})$ is an element of $EQ((8.5.2))$ for all $k \in M' \setminus \{l\}$. Thus, $(|N'| - 1)c = |N'|c + b_{tk} + b_{tl}$, and together with the observations above, $b_{tk} = -c$ for all $k \in M' \setminus \{l\}$.

• (d) $b_{tk} = -\frac{c}{2}$ for all $t \in Z_1$, $k \in M \setminus \{l\}$.

Now, let $t \in Z_1$. Condition (i) guarantees that there exist two edges $t', t'' \in Z' \setminus \{t\}$ $(t' \neq t'')$, such that the hypergraphs $G' - \{t, t'\}$, $G' - \{t, t''\}$ and $G' - \{t', t''\}$ are not connected. Consider, then, the three following partitions of N':

- (N_1, N_2), where N_1 is the node set of some component of $G' - \{t, t'\}$ and $N_2 := N' \setminus N_1$. Observe that $(\sum_{i \in N_1} e_{ik} + \sum_{i \in N_2} e_{il}, e_{tk} + e_{t'k} + e_{tl} + e_{t'l})$ is an element of $EQ((8.5.2))$ for all $k \in M' \setminus \{l\}$, since N' is a minimal cover for all knapsacks in M' and N_1 and N_2 are nonempty. Combining with the observations above, $(|N'| - 1)c = |N'|c + b_{tk} + b_{t'k}$, and therefore, $b_{tk} + b_{t'k} = -c$ for all $k \in M \setminus \{l\}$.

- (N'_1, N'_2), where N'_1 is the node set of some component of $G' - \{t', t''\}$ and $N'_2 := N' \setminus N'_1$. Observe that $(\sum_{i \in N'_1} e_{ik} + \sum_{i \in N'_2} e_{il}, e_{t'k} + e_{t''k} + e_{t'l} + e_{t''l})$ is an element of $EQ((8.5.2))$ for all $k \in M' \setminus \{l\}$, since N' is a minimal cover for all knapsacks in M' and N'_1 and N'_2 are nonempty. Combining with the observations above, $(|N'| - 1)c = |N'|c + b_{t'k} + b_{t''k}$, and therefore, $b_{t'k} + b_{t''k} = -c = b_{tk} + b_{t'k}$. Then, $b_{tk} = b_{t''k}$ for all $k \in M \setminus \{l\}$.

- (N''_1, N''_2), where N''_1 is the node set of some component of $G' - \{t, t''\}$ and $N''_2 := N' \setminus N''_1$. Observe that $(\sum_{i \in N''_1} e_{ik} + \sum_{i \in N''_2} e_{il}, e_{tk} + e_{t''k} + e_{tl} + e_{t''l})$ is an element of $EQ((8.5.2))$ for all $k \in M' \setminus \{l\}$, since N' is a minimal cover for all knapsacks in M' and N''_1 and N''_2 are nonempty. Combining with the observations above, $(|N'| - 1)c = |N'|c + b_{tk} + b_{t''k}$, and therefore, $b_{tk} + b_{t''k} = -c$. Then, $b_{tk} = b_{t'k} = b_{t''k} = -\frac{c}{2}$ for all $k \in M \setminus \{l\}$.

Taking all these observations into account we conclude that the inequality $ax + by \leq \alpha$ is a scalar multiple of (8.5.2), and therefore, it defines a facet of $MCH(N' \times M', Z' \times M', f, F, g, S)$.

Now we prove the converse direction. If the inequality is not valid, it cannot be facet defining. Suppose now that (i) is not satisfied, and let $t \in Z_1$ be the edge for which the condition does not hold. Then, there are two cases to be considered.

- $G - \{t, t'\}$ is connected for all $t' \in Z'$. But then, according to Theorem 8.5.1, the Cycle Inequality corresponding to $G' - t$, M', l, $Z'_1 = Z_1 \setminus \{t\}$ and $Z'_2 = Z_2$ is also valid. The inequality (8.5.2) is the sum of the inequality above and the valid inequality $\sum_{k \in M' \setminus \{l\}} y_{tk} \geq 0$, and therefore cannot be facet defining.

- there exists some $t' \in Z_1$ such that $G' - \{t, t'\}$ is not connected, but for all $t'' \in Z_1 \setminus \{t, t'\}$, $G' - \{t, t''\}$ is connected. Observe that it implies that $G' - \{t', t''\}$ is also connected by the argument pointed out before. Consider that the following inequality

$$2 \sum_{i \in N'} \sum_{k \in M'} x_{ik} - \sum_{u \in Z_1 \setminus \{t, t'\}} \sum_{k \in M' \setminus \{l\}} y_{uk} - 2 \sum_{u \in Z_2 \cup \{t'\}} \sum_{k \in M' \setminus \{l\}} y_{uk} \leq 2(|N'| - 1)$$

is the Cycle Inequality corresponding to $G' - t$, M', l, $Z'_1 = Z_1 \setminus \{t, t'\}$ and $Z'_2 = Z_2 \cup \{t'\}$, and since the conditions of Theorem 8.5.1 are satisfied, the inequality is valid. Analogously,

$$2 \sum_{i \in N'} \sum_{k \in M'} x_{ik} - \sum_{u \in Z_1 \setminus \{t, t'\}} \sum_{k \in M' \setminus \{l\}} y_{uk} - 2 \sum_{u \in Z_2 \cup \{t\}} \sum_{k \in M' \setminus \{l\}} y_{uk} \leq 2(|N'| - 1)$$

is the Cycle Inequality corresponding to $G' - t'$, M', l, $Z'_1 = Z_1 \setminus \{t, t'\}$ and $Z'_2 = Z_2 \cup \{t\}$ and is also valid. But, (8.5.2) is the sum of the two inequalities above divided by 2.

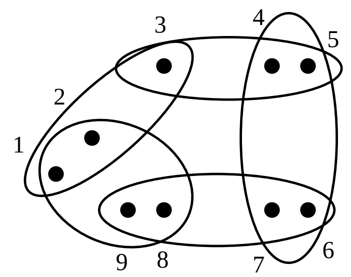


Figure 7.3: Example 8.5.3.

■

Observe that if $Z_1 = \emptyset$, then the Cycle Inequality is a scalar multiple of the Tree Inequality. We show some examples of facet defining Cycle Inequalities.

Example 7.5.3 Consider the following instance of the MPH. $N' = \{1, \ldots, 9\}$, $M' = \{1, 2\}$, $f = (2, 2, 3, 3, 4, 3, 4, 2, 3)$, $F = (24, 25)$, $\mathcal{Z} = \{\{1, 2, 3\}, \{3, 4, 5\}, \{4, 5, 6, 7\}, \{6, 7, 8, 9\}, \{1, 2, 8, 9\}\}$, $g = (1, 1, 1, 1, 1)$ and $S = (5, 5)$ (see Figure 7.3). In this case, $Z_2 = \emptyset$, and the Cycle Inequalities

$$2\sum_{i=1}^{9}\sum_{k=1}^{2} x_{ik} - \sum_{t=1}^{5} y_{t,1} \leq 16,$$

$$2\sum_{i=1}^{9}\sum_{k=1}^{2} x_{ik} - \sum_{t=1}^{5} y_{t,2} \leq 16$$

define facets of the polytope $MCH(N \times M, Z \times M, f, F, g, S)$. ■

Example 7.5.4 Consider the following instance of the MPH. $N' = \{1, \ldots, 9\}$, $M' = \{1, 2\}$, $f = (2, 2, 3, 3, 4, 3, 4, 2, 3)$, $F = (24, 25)$, $\mathcal{Z} = \{\{1, 2, 3\}, \{3, 4, 5\}, \{5, 6\}, \{6, 7, 8\}, \{8, 9\}, \{1, 9\}\}$, $g = (1, 1, 1, 1, 1, 1)$ and $S = (6, 6)$ (see Figure 7.4). In this case, $Z_1 = \{3, 5, 6\}$ and $Z_2 = \{1, 2, 4\}$. The Cycle Inequalities

$$2\sum_{i=1}^{9}\sum_{k=1}^{2} x_{ik} - \sum_{t \in Z_1} y_{t,1} - 2\sum_{t \in Z_2} y_{t,1} \leq 16,$$

$$2\sum_{i=1}^{9}\sum_{k=1}^{2} x_{ik} - \sum_{t \in Z_1} y_{t,2} - 2\sum_{t \in Z_2} y_{t,2} \leq 16$$

define facets of the polytope $MCH(N \times M, Z \times M, f, F, g, S)$. ■

Example 7.5.5 Consider the following instance of the MPH. $N' = \{1, \ldots, 10\}$, $M' = \{1, 2\}$, $f = (2, 2, 3, 3, 4, 3, 4, 2, 3, 2)$, $F = (26, 27)$, $\mathcal{Z} = \{\{1, 2, 3\}, \{3, 4\},$

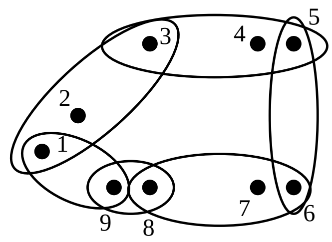


Figure 7.4: Example 8.5.4.

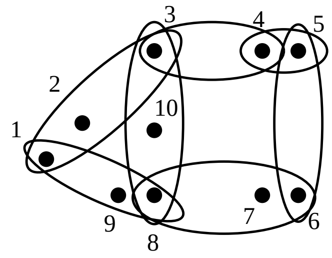


Figure 7.5: Example 8.5.5.

$\{4,5\},\{5,6\},\{6,7,8\},\{1,8,9\},\{3,8,10\}\}$, $g = (1,1,1,1,1,1,1)$ and $S = (7,7)$ (see Figure 7.5). In this case, $Z_1 = \{2,3,4\}$ and $Z_2 = \{1,5,6,7\}$.

The Cycle Inequalities

$$2\sum_{i=1}^{10}\sum_{k=1}^{2} x_{ik} - \sum_{t\in Z_1} y_{t,1} - 2\sum_{t\in Z_2} y_{t,1} \leq 18,$$

$$2\sum_{i=1}^{10}\sum_{k=1}^{2} x_{ik} - \sum_{t\in Z_1} y_{t,2} - 2\sum_{t\in Z_2} y_{t,2} \leq 18$$

define facets of the polytope $MCH(N \times M, Z \times M, f, F, g, S)$. ■

7.6 Further facet defining inequalities

In this section we show some more valid and facet defining inequalities for the polytope $MCH(N \times M, Z \times M, f, F, g, S)$. All inequalities presented in this section involve only one net of Z.

Theorem 7.6.1 *Let an instance $(G = (N,Z), M, f, F, g, S)$ of the MPH be given. Let $M' \subseteq M$, $k \in M'$ and suppose that $S \subseteq N$ is a minimal cover with respect to*

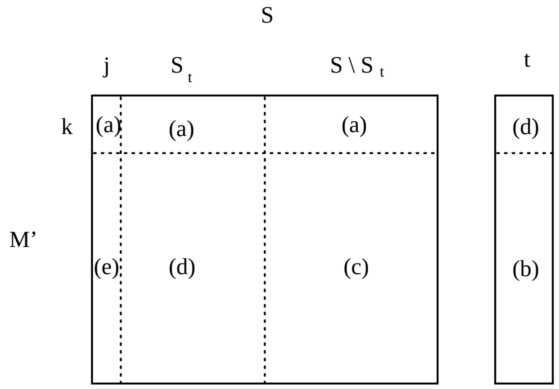
k. Let $t \in Z$ with $S \cap T_t \neq \emptyset$ and $j \in S \cap T_t$ and define $S_t := S \cap T_t$. Finally, suppose that $\sum_{i \in S_t} f_i \leq F_l$ for all $l \in M' \setminus \{k\}$. Then, the inequality,

$$\sum_{i \in S} x_{ik} + \sum_{i \in S_t \setminus \{j\}} \sum_{l \in M' \setminus \{k\}} x_{il} - y_{tk} \leq |S| - 1 \tag{8.6.1}$$

defines a facet of $MCH(S \times M', \{t\} \times M', f, F, g, S)$.

Proof. First we prove that the inequality is valid for $MCH(S \times M', \{t\} \times M', f, F, g, S)$. Suppose that there exists a vector $(x', y') \in MCH(S \times M', \{t\} \times M', f, F, g, S)$ violating (8.6.1). Then, every element of S must be assigned to some knapsack in M'. Since S is a cover for k, there exists some $i \in (S \cap T_t) \setminus \{j\}$ and $l' \in M' \setminus \{k\}$ such that $x'_{il'} = 1$. But then, $\sum_{i \in T_t} x'_{ik} = 0$, otherwise, the net constraint would imply $y'_{tk} = 1$, contradicting our assumption that (x', y') violates (8.6.1). In this case, however, the item j is not assigned to any knapsack, and it contradicts our assumption that the vector violates (8.6.1).

Now we prove that the inequality is facet defining. Suppose that there exists a facet defining inequality $ax + by \leq \alpha$ such that $EQ((8.6.1)) \subseteq EQ(ax + by \leq \alpha)$ we prove in the following that this inequality is a scalar multiple of (8.6.1).



• (a) $a_{ik} = c$ for all $i \in S$.

Let $i' \in S$ and notice that $(\sum_{i \in S \setminus \{i'\}} e_{ik}, \mathbf{0})$ is in $EQ((8.6.1))$. Since this is true for all $i' \in S$, we can conclude that there exists a constant c such that $a_{ik} = c$ for all $i \in S$.

• (b) $b_{tl} = 0$ for all $l \in M' \setminus \{k\}$.

Now, let $i' \in S$, $l \in M' \setminus \{k\}$ and observe that $(\sum_{i \in S \setminus \{i'\}} e_{ik}, e_{tl})$ is also in $EQ((8.6.1))$. Thus, $(|S| - 1)c = (|S| - 1)c + b_{tl}$, and therefore, $b_{tl} = 0$ for all $l \in M' \setminus \{k\}$.

• (c) $a_{i'l} = 0$ for all $i' \in S \setminus S_t$, $l \in M' \setminus \{k\}$.

Consider now some $i' \in S \setminus S_t$. Since $(\sum_{i \in S \setminus \{i'\}} + e_{i'l}, \mathbf{0})$ is in $EQ((8.6.1))$ for all $l \in M' \setminus \{k\}$, we can conclude $(|S| - 1)c = (|S| - 1)c + a_{i'l}$ and therefore $a_{i'l} = 0$ for all $l \in M' \setminus \{k\}$, $i' \in S \setminus S_t$.

• (d) $-b_{tk} = a_{il} = d$ for all $i \in S_t \setminus \{j\}$, $l \in M' \setminus \{k\}$.

Let $i' \in S_t \setminus \{j\}$. Observe that $(\sum_{i \in S \setminus \{i'\}} e_{ik} + e_{i'l}, e_{tk} + e_{tl})$ is in $EQ((8.6.1))$ for all $l \in M \setminus \{k\}$. Then, combining with the observations above, $(|S|-1)c = (|S|-1)c + a_{i'l} + b_{tk}$, and therefore, $a_{i'l} = -b_{tk}$ for all $l \in M' \setminus \{k\}$. Since the same argument holds for all $i' \in S_t \setminus \{j\}$, we can conclude that there exists some constant d such that $a_{il} = d$ for all $i \in S_t \setminus \{j\}$, $l \in M' \setminus \{k\}$, and $b_{tk} = -d$.

- $c = d$.

Now, observe that $(\sum_{i \in S \setminus S_t} e_{ik} + \sum_{i \in S_t \setminus \{j\}} e_{il}, \mathbf{0})$ is in $EQ((8.6.1))$ for all $l \in M' \setminus \{k\}$. Then, $(|S|-1)c = (|S| - |S_t|)c + (|S_t| - 1)d$, and therefore $c = d$.

- (e) $e_{jl} = 0$ for all $l \in M'$.

Since $(\sum_{i \in S \setminus S_t} e_{ik} + \sum_{i \in S_t} e_{il}, \mathbf{0})$ is also an element of $EQ((8.6.1))$, we conclude that $e_{jl} = 0$ for all $l \in M'$.

Considering all observations above we conclude that $ax + by \leq \alpha$ is a scalar multiple of (8.6.1), and therefore it defines a facet of $MCH(S \times M', \{t\} \times M', f, F, g, S)$.
■

The next inequality has nonzero coefficients whose values are different from 1 and -1.

Theorem 7.6.2 *Let an instance $(G = (N, Z), M, f, F, g, S)$ of the MPH be given. Let $M' \subseteq M$, $k \in M'$ and $S \subseteq N$ such that S is a minimal cover with respect to k. Let $t \in Z$ such that $S_t := S \cap T_t \neq \emptyset$ and suppose that $\sum_{i \in S_t} f_i \leq F_l$ for all $l \in M'$. Then, the inequality*

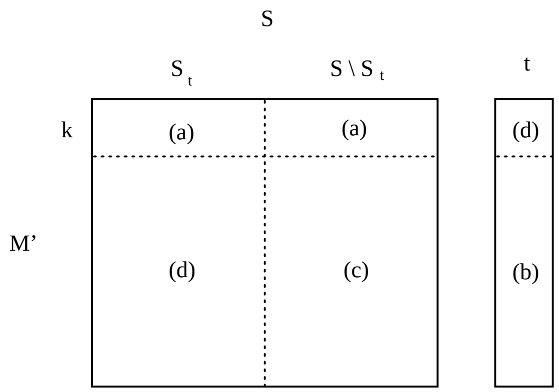
$$|S_t| \sum_{i \in S} x_{ik} + (|S_t| - 1) \sum_{i \in S_t} \sum_{l \in M' \setminus \{k\}} x_{il} - (|S_t| - 1) y_{tk} \leq |S_t|(|S| - 1) \qquad (8.6.2)$$

defines a facet of $MCH(S \times M', \{t\} \times M', f, F, g, S)$.

Proof. We first prove that the inequality is valid. Suppose that there exists some vector (x', y') violating (8.6.2). Then, every element of S must be assigned to some knapsack in M'. Since S is a cover for k, some item of $S \cap T$ is assigned to some knapsack $l \neq k$. Then, we can conclude that $\sum_{i \in S_t} x'_{ik} = 0$, otherwise, due to the net constraints, $y'_{tk} = 1$. In this case, however, (x', y') does not violate (8.6.2).

Now we prove that the inequality is facet defining. Suppose that there exists a facet defining inequality $ax + by \leq \alpha$ such that $EQ((8.6.2)) \subseteq EQ(ax + by \leq \alpha)$.

We prove that the inequality is a scalar multiple of (8.6.2).



• (a) $a_{ik} = c$ for all $i \in S$.

First, let $i' \in S$, and observe that $(\sum_{i \in S \setminus \{i'\}} e_{ik}, \mathbf{0})$ is an element of $EQ((8.6.2))$. Since the same argument holds for all $i' \in S$, we can conclude that there exists some constant c such that $a_{ik} = c$ for all $i \in S$.

• (b) $b_{tl} = 0$ for all $l \in M' \setminus \{k\}$.

Consider some $i' \in S$ and observe also that $(\sum_{i \in S \setminus \{i'\}} e_{ik}, e_{tl})$ is in $EQ((8.6.2))$ for all $l \in M' \setminus \{k\}$. Thus, $(|S|-1)c = (|S|-1)c + b_{tl}$, and therefore, $b_{tl} = 0$ for all $l \in M' \setminus \{k\}$.

• (c) $a_{i'l} = 0$ for all $i' \in S \setminus S_t$, $l \in M' \setminus \{k\}$.

Suppose that $i' \in S \setminus S_t$. Then, $(\sum_{i \in S \setminus \{i'\}} e_{ik} + e_{i'l}, \mathbf{0})$ is an element of $EQ((8.6.2))$ for all $l \in M' \setminus \{k\}$. We can, then, conclude that $(|S|-1)c = (|S|-1)c + a_{i'l}$, and therefore, $a_{i'l} = 0$ for all $l \in M' \setminus \{k\}$.

• (d) $-b_{tk} = a_{il} = d$ for all $i \in S_t$, $l \in M' \setminus \{k\}$.

Now, consider $i' \in S_t$. Then, $(\sum_{i \in S \setminus \{i'\}} e_{ik} + e_{i'l}, e_{tk} + e_{tl})$ is in $EQ((8.6.2))$ for all $l \in M' \setminus \{k\}$. Combining with the observations above we conclude $(|S|-1)c = (|S|-1)c + a_{i'l} + b_{tk}$, and thus, $a_{i'l} = -b_{tk}$ for all $l \in M' \setminus \{k\}$. Since the same argument holds for all $i' \in S_t$, we can conclude that there exists some constant d such that $a_{il} = d$ for all $i \in S_t$, $l \in M' \setminus \{k\}$, and $b_{tk} = -d$.

• $c = \frac{|S_t|}{|S_t|-1} d$.

Finally, consider that $(\sum_{i \in S \setminus S_t} e_{ik} + \sum_{i \in S_t} e_{il}, \mathbf{0})$ is an element of $EQ((8.6.2))$ for all $l \in M' \setminus \{k\}$. Thus, $(|S|-1)c = (|S| - |S_t|)c + (|S_t|)d$, and therefore $c = \frac{|S_t|}{|S_t|-1} d$.

This yields that $ax + by \leq \alpha$ is a scalar multiple of $EQ((8.6.2))$, and therefore, it defines a facet of $MCH(S \times M', \{t\} \times M', f, F, g, S)$. ■

Chapter 8

A Branch and Cut Algorithm for the Multicut Problem on Hypergraphs

In this chapter we report on our implementation of a Branch and Cut Algorithm for the MPH using the results on valid and facet defining inequalities we describe in Chapter 8. The algorithm is applied to instances coming from real world problems.

As in Chapter 6, we present our approach to the minimization version of the MPH, and consider only the feasible solutions assigning all the items. This is due to the fact that solutions that not assign all the items are not feasible in the practical application. For more details on the modelling and motivation, see Chapter 2.

In Chapter 6 we described in detail an implementation of a Branch and Cut Algorithm for the MKP. In this chapter we focus on the parts of the algorithm that are problem dependent, namely: separation strategies and primal heuristics. These are described in Sections 9.1 and 9.2, respectively. In Section 9.3 we show the computational results of the algorithm.

8.1 Separation procedures

In this section we present procedures for the separation of some classes of inequalities discussed in the last chapter. We show, for each inequality class, the number of inequalities of this type used in the solution process of the instances solved.

8.1.1 Separating Net Inequalities

The number of Net Inequalities is polynomial in the input length. Then, an algorithm that enumerate all those inequalities solves the corresponding Separation Problem trivially in polynomial time. The procedure we present here finds, for the instances arising from the practical application, many violated inequalities in each iteration. We use a parameter MAX to avoid generating too many inequalities of this type. Moreover, if more than MAX Net Inequalities are violated, we try to avoid selecting inequalities whose support have large intersection. This is done in the following way. In the case that more than MAX inequalities are found, given some net $t \in Z$ and some knapsack $k \in M$, at most one violated Net Inequality corresponding to t and k is added to the list of violated inequalities. In the following we describe precisely the procedure.

Separation Procedure for Net Inequalities
Input: An instance $(G = (N, Z), M, f, F, g, S)$ of the MPH, an integer MAX and a fractional point $(x'_{ik}, y'_{tl}) \in \mathbb{R}^{N \times M} \times \mathbb{R}^{Z \times M}$, with $0 \leq x'_{ik} \leq 1$ and $0 \leq y'_{tl} \leq 1$ for all $i \in N$, $k, l \in M$, $t \in Z$.
Output: Violated Net Inequalities if there exists any.

For every net $t \in Z$.
  For all pairs $i, j \in T_t$.
    For all $k \in M$.
      If the Net Inequality corresponding to t, i, j and k is violated.
        Store this inequality in the list of violated inequalities.
If the number of violated inequalities found is greater than MAX.
  Rearrange the set of violated inequalities such that the inequalities most violated inequalities come first (not necessarily sorted).
  Consider each inequality in this sequence, avoiding selecting more than one inequality corresponding to the same net t and knapsack k.
Return the list of selected violated inequalities.

Since we avoid sorting the whole set of violated inequalities (which would require $O(zn^2m\log(znm))$ time complexity), the procedure proposed above has time complexity limited by $O(zn^2m)$ which is also the number of different Net Inequalities of the instance. Although several thousands of violated inequalities are found by the procedure, it is not necessary to have all of them in the LP. In Table 8.1 we show the number of Net Inequalities used to solve the 18 instances described in detail in Section 9.3. For these instances we have calculated the total number of Net Inequalities, which is given in the second column. The third column show how many of them are used in the whole solution process.

Problem	Total of Net Ineq.	# Net Ineq. used
pd1.50.1511	16192	15666 (96.75%)
pd1.100.1511	19472	10808 (55.51%)
pd1.150.1511	34912	4833 (13.84%)
pd1.200.1511	46672	6768 (14.50%)
pd1.253.200	13792	12817 (92.93%)
pd1.253.500	30160	6940 (23.01%)
pd1.253.750	37488	7975 (21.27%)
pd1.253.1000	46568	4310 (9.26%)
pd1.253.1200	48168	7450 (15.34%)
pd1	59162	5380 (9.09%)
pd2.150.3326	89196	9840 (11.03%)
pd2.300.3326	145032	5831 (4.02%)
pd2.450.3326	179232	18903 (10.52%)
pd2.600.3326	229656	9759 (4.25%)
pd2.772.750	64920	4249 (6.54%)
pd2.772.1200	96420	11753 (12.19%)
pd2.772.1600	117348	10820 (9.22%)
pd2.772.2000	173796	17245 (9.92%)
pd2.772.2500	248256	30450 (12.26%)
pd2	308580	10801 (3.50%)

Table 8.1: Net Inequalities.

8.1.2 Separating Tree Inequalities

The Separation Problem for the Tree Inequalities is $\mathcal{NP}$-hard. It is due to the fact that in the particular case that the hypergraph considered is a graph, the corresponding Separation Problem is $\mathcal{NP}$-hard (cf. [FMSWW93]). We have implemented two different heuristic procedures to find violated Tree Inequalities. In the following we present the two procedures.

Weighted Distance Heuristic

In this procedure we try to find, for each item $i \in N$, a violated Tree Inequality rooted at i, corresponding to all knapsacks in M. At each iteration of the procedure we have a subtree. In the first iteration it is constituted by the isolated node i. We select, in each iteration, the most promising edge t incident to the current subtree. Then, we enlarge it, adding to the subtree all nodes incident to t. The criterion used to choose the promising edge t is described in the following. First, we want to obtain a tree that defines a cover with respect to all knapsacks in M. Thus, we try to select edges that add items to the tree with large weights. Second, the value of the current fractional solution corresponding to the selected edges must be as small as possible. In order to tackle both targets we define the following "weighted distance function" $w : Z \longrightarrow \mathbb{R}$, for the nets in Z corresponding to a given subtree $T = (VT, ET)$, where

$$w_t := \begin{cases} \frac{\sum_{k \in M} y_{tk}}{\sum_{j \in T_t \setminus VT} f_j} & \text{if } T_t \cap VT \neq \emptyset \text{ and } T_t \nsubseteq VT, \\ B & \text{if } T_t \cap VT = \emptyset, \\ 2B & \text{if } T_t \subseteq VT, \end{cases}$$

where B is a large number. We select, in each iteration of the algorithm, the edge t with minimum weighted distance. We describe precisely the heuristic in the following.

Sep. Procedure: Weighted Distance Heur. for Tree Ineq.
Input: An instance $(G = (N, Z), M, f, F, g, S)$ of the MPH, a fractional point $(x'_{ik}, y'_{tl}) \in \mathbb{R}^{N \times M} \times \mathbb{R}^{Z \times M}$, with $0 \leq x'_{ik} \leq 1$ and $0 \leq y'_{tl} \leq 1$ for all $i \in N$, $k, l \in M$, $t \in Z$.
Output: Violated Tree Inequalities, or the procedure fails.

For each item $i \in N$.
  Initialize $w_t := \begin{cases} B & \text{if } i \notin T_t, \\ \frac{\sum_{k \in M} y'_{tk}}{\sum_{j \in T_t \setminus \{i\}} f_j} & \text{otherwise.} \end{cases}$
  Initialize $VT := \{i\}$; $ET := \emptyset$.
  While $|VT| \leq n - 1$ and VT is not a cover for all knapsacks.
    Let $t \in Z \setminus ET$ be such that $w_t = \min\{w_u \mid u \in Z \setminus ET\}$.
    Update $VT := VT \cup T_t$.
    Update $ET := ET \cup \{t\}$.
    Update w conveniently.
  If VT is a cover for all knapsacks and the Tree Inequality corresponding to $T = (VT, ET)$ is violated.
    Append this inequality to the list of violated inequalities.

Spanning Tree Heuristic

The idea used in this heuristic procedure is mainly to proceed as in Kruskal's Algorithm for computing a minimum spanning tree. We interpret, at the beginning, each item i as an isolated node. We sort, then, the edges t of the graph in increasing order according to $\sum_{k \in M} y_{tk}$. We consider each edge in this sequence and add it to the tree unless it links items from only one component, or it forms a component with total sum of the net variables bigger than a given parameter r. After all nets have been considered, we obtain a set of connected components.

If the node set of some of these components defines a cover with respect to all knapsacks in M, we can verify if the corresponding Tree Inequality is violated. We describe the procedure precisely.

Sep. Procedure: Spanning Tree Heur. for Tree Inequalities.
Input: An instance $(G = (N, Z), M, f, F, g, S)$ of the MPH, a fractional point $(x'_{ik}, y'_{tl}) \in \mathbb{R}^{N\times M} \times \mathbb{R}^{Z\times M}$, with $0 \leq x'_{ik} \leq 1$ and $0 \leq y'_{tl} \leq 1$ for all $i \in N$, $k, l \in M$, $t \in Z$ and a threshold parameter $r \in \mathbb{R}$.
Output: Violated Tree Inequalities, or the procedure fails.

Initialize $C_i = (VC_i, EC_i)$ with $VC_i := \{i\}$ and $EC_i := \emptyset$ for all $i \in N$.
Sort Z such that $\sum_{k\in M} y_{t_1 k} \leq \ldots \leq \sum_{k\in M} y_{t_z k}$.
For $e \in Z$.
  If t_e links at least two different components.
    Let $C_{i_1}, \ldots, C_{i_w}$ the components linked by t_e.
    Define $E' := \cup_{j=1}^{w} EC_{i_j}$.
    If $\sum_{t\in E'\cup\{t_e\}} \sum_{k\in M} y_{tk} \leq r$
      Merge the components $C_{i_1}, \ldots, C_{i_w}$.
      Update VC_i and EC_i conveniently.
For each component C_i in the final subgraph.
  If VC_i is a cover for all knapsacks.
    If the Tree Ineq. corresponding to the component C_i is violated.
      Add this inequality to the list of violated inequalities.

This second heuristic produces less inequalities than the first one in general, but, in some sense, more global information is used in the construction of the components. In our algorithm we use both separation routines to find violated tree inequalities. In Table 8.2 we show the number of violated Tree Inequalities found by both heuristic procedures presented in this section in the whole solution process of the problem instances described in Section 9.3.

8.1.3 Separating Cycle Inequalities

Up to our knowledge, it is an open problem to decide if the Separation Problem for the Cycle Inequalities can be solved in polynomial time. We conjecture that this problem cannot be solved in polynomial time, and concentrate on developing heuristic procedures to solve it. We have implemented two heuristic procedures to separate Cycle Inequalities. They are presented in the following.

Shortest Path Heuristic

In this procedure we try to obtain cycles in the hypergraph using shortest paths. We look for violated inequalities for each net t in Z. In each iteration a subgraph

Problem	# Tree Ineq. used
pd1.50.1511	4340
pd1.100.1511	5679
pd1.150.1511	3926
pd1.200.1511	3015
pd1.253.200	6498
pd1.253.500	2758
pd1.253.750	4826
pd1.253.1000	1016
pd1.253.1200	3302
pd1	762
pd2.150.3326	5400
pd2.300.3326	2871
pd2.450.3326	1263
pd2.600.3326	1086
pd2.772.750	0
pd2.772.1200	1576
pd2.772.1600	1135
pd2.772.2000	1169
pd2.772.2500	2070
pd2	1444

Table 8.2: Tree Inequalities.

C is considered. In the first iteration C is composed by the single net t, and this net is deleted from the graph. We choose some node s in the node set of C as the source, and set all other items in C as the destinations. We determine, then, among all paths between s and the destinations, the one with minimum total cost corresponding to the value $\sum_{k \in M} y_{tk}$ associated with each edge t. This shortest path is then added to C and, afterwards, the edges of this path are deleted from the graph. We proceed in this way until the node set of C defines a cover with respect to all knapsacks in M. Then, we determine the sets Z_1 and Z_2 as defined in Section 8.5 and verify if the corresponding Cycle Inequality is violated. In the following we present precisely the procedure.

Sep. Procedure: Shortest Path Heur. for Cycle Inequalities.
Input: An instance $(G = (N, Z), M, f, F, g, S)$ of the MPH, a fractional point $(x'_{ik}, y'_{tl}) \in \mathbb{R}^{N \times M} \times \mathbb{R}^{Z \times M}$, with $0 \leq x'_{ik} \leq 1$ and $0 \leq y'_{tl} \leq 1$ for all $i \in N$, $k, l \in M$, $t \in Z$.
Output: Violated Cycle Inequalities, or the procedure fails.

For each $t \in Z$.
    Initialize $C = (VC, EC)$ where $VC := T_t$ and $EC := \{t\}$.
    Set $G' = G - t$.
    While VC is not a cover for all knapsacks.
        Choose an item $s \in VC$ as the source.
        Let $D := VC \setminus \{s\}$ be the destinations.
        Find the shortest path $P = (VP, EP)$ in G' between the source and some destination, using the value $\sum_{k \in M} y_{tk}$ as cost associated with edge t.
        If no such path exists.
            The procedure fails and stop.
        Otherwise set $G' := G' - EP$.
    Determine the sets Z_1 and Z_2 as defined in Section 8.5:
    Set $Z_1 := \emptyset$.
    For each $e \in EC$.
        If $C' = (VC, EC \setminus \{e\})$ is connected.
            $Z_1 := Z_1 \cup \{e\}$.
    $Z_2 := Z \setminus Z_1$.
    If the corresponding Cycle Inequality is violated.
        Append it to the list of violated inequalities.

Spanning Tree Heuristic

The idea in this heuristic is very similar to the one used to find violated Tree Inequalities. We proceed like in Kruskal's Algorithm for computing a minimum spanning tree. We sort the edges of the graph in increasing order according to the value $\sum_{k \in M} y_{tk}$. Then, each edge is considered and is added to the subgraph unless it links only items from one component, or the total weight of the component becomes too large. After all edges are considered, we obtain a set of connected components. We consider those components that define a cover with respect to all knapsacks and insert edges that are contained in those component, in order to obtain a highly connected subgraph. The edge set is then divided into sets Z_1 and Z_2 and we verify if the corresponding inequality is violated. We present the idea precisely in the next.

Sep. Procedure: Spanning Tree Heur. for Cycle Ineq.
Input: An instance $(G = (N, Z), M, f, F, g, S)$ of the MPH, a fractional point $(x'_{ik}, y'_{tl}) \in \mathbb{R}^{N \times M} \times \mathbb{R}^{Z \times M}$, with $0 \leq x'_{ik} \leq 1$ and $0 \leq y'_{tl} \leq 1$ for all $i \in N$, $k, l \in M$, $t \in Z$ and a threshold parameter $r \in \mathbb{R}$.
Output: Violated Cycle Inequalities, or the procedure fails.

Initialize $C_i = (VC_i, EC_i)$ with $VC_i := \{i\}$ and $EC_i := \emptyset$ for all $i \in N$.

Sort Z such that $\sum_{k\in M} y'_{t_1 k} \leq \ldots \leq \sum_{k\in M} y'_{t_z k}$.
For $e \in Z$.
  If t_e links at least two different components
    Let $C_{i_1}, \ldots, C_{i_w}$ the components linked by t_e.
    Define $E' := \cup_{j=1}^{w} EC_{i_j}$.
    If $\sum_{t\in E'\cup\{t_e\}} \sum_{k\in M} y'_{tk} \leq r$
      Merge the components $C_{i_1}, \ldots, C_{i_w}$.
      Update VC_i and EC_i conveniently.
For each component C_i in the final subgraph.
  If VC_i is a cover for all knapsacks.
    Let $NC := \{t \in Z \setminus EC_i \mid T_t \subseteq VC_i\}$.
    Sort NC such that $\sum_{k\in M} y'_{t_1 k} \leq \ldots \leq \sum_{k\in M} y'_{t_{|NC|} k}$.
    For $e \in NC$.
      If $\sum_{t\in EC_i\cup\{t_e\}} \sum_{k\in M} y_{tk} \leq r$.
        $EC_i := EC_i \cup \{t_e\}$.
    Determine the sets Z_1 and Z_2 as defined in Section 8.5:
    Set $Z_1 := \emptyset$.
    For each $e \in EC$.
      If $C' = (VC, EC \setminus \{e\})$ is connected.
        $Z_1 := Z_1 \cup \{e\}$.
    $Z_2 := Z \setminus Z_1$.
    If the corresponding Cycle Inequality is violated.
      Append it to the list of violated inequalities.

We use both procedures to find violated Cycle Inequalities. In Table 8.3 we show the number of violated Cycle Inequalities found in the whole solution process of the instances described in Section 9.3.

8.2 Primal heuristics

In Section 6.3 we present some LP based primal heuristics for the MKP. These heuristics can also be used to provide solutions for the MPH. Suppose we are given a partition $(B_1, \ldots, B_j)$ of N, such that $j \leq m$ and $\sum_{i\in B_k} f_i \leq F_k$ for all $k = 1, \ldots, j$, found by some of the heuristics presented in Section 6.3. We must verify if this partition defines a valid multicut, i.e., if all net knapsack constraints are satisfied: $\sum_{e\in\delta(B_k)} g_e \leq S_k$ for all $k \in M$. If it is the case, we must calculate the value of this solution.

In this section we present a primal heuristic procedure that is based on the values of the current LP solution correspondent to the net variables. This is the topic of Section 9.2.1. An improvement heuristic based on one exchange is presented in Section 9.2.2.

Problem	# Cycle Ineq. used
pd1.50.1511	3144
pd1.100.1511	2756
pd1.150.1511	3678
pd1.200.1511	2868
pd1.253.200	12
pd1.253.500	64
pd1.253.750	3726
pd1.253.1000	787
pd1.253.1200	2757
pd1	646
pd2.150.3326	4122
pd2.300.3326	2795
pd2.450.3326	1035
pd2.600.3326	748
pd2.772.750	0
pd2.772.1200	3
pd2.772.1600	46
pd2.772.2000	772
pd2.772.2500	1114
pd2	1143

Table 8.3: Cycle Inequalities.

8.2.1 Bin-Packing heuristic

In this heuristic we use the value of the current LP solution corresponding to the net variables to construct partitions. We interpret a small value of the net variable as a hint that the items in this net should be assigned to the same knapsack. The idea is, then, to partition the items set into clusters that, by the criterion above, should be together, taking into account also that the knapsack constraints are not violated. If the number of clusters is already less than or equal to the number of knapsacks, a feasible solution is found. Otherwise, we assign the clusters to the knapsacks in a first fit decreasing way. Now we describe the heuristic procedure more precisely.

Bin-Packing Heuristic
Input: An instance $(G = (N, Z), M, f, F, g, S)$ of the MPH, a fractional point $(x'_{ik}, y'_{tl}) \in \mathbb{R}^{N \times M} \times \mathbb{R}^{Z \times M}$, with $0 \leq x'_{ik} \leq 1$ and $0 \leq y'_{tl} \leq 1$ for all $i \in N$, $k, l \in M$, $t \in Z$ and a threshold parameter $p \in [0, 1]$.

Output: A solution $a \in M^N$ of the MPH, where a_i is the knapsack where item i is assigned to.

Initialize $C_i = (VC_i, EC_i)$ with $VC_i = \{i\}$ and $EC_i := \emptyset$ for all $i \in N$.
Sort Z such that $\sum_{k \in M} y'_{t_1 k} \leq \ldots \leq \sum_{k \in M} y'_{t_z k}$.
For $j \in Z$.
  If $\sum_{k \in M} y_{t_j k} \leq p$.
    Let $C_{i_1}, \ldots, C_{i_w}$ be the components linked by t_j.
    If $\sum_{v \in \cup_{u=1}^{w} VC_{i_u}} f_v \leq F_k$ for all $k \in M$.
      Merge the components $C_{i_1}, \ldots, C_{i_w}$.
      Update VC_i and EC_i conveniently.
Let c be the final number of components.
Define $w_{dk} := \frac{\sum_{i \in VC_d} x'_{ik}}{|VC_d|}$ for all $d = 1, \ldots, c$, $k \in M$.
Set $I_k := \emptyset$ for all $k \in M$.
If $c \leq m$
  For $d = 1, \ldots, c$.
    Find k^* such that
    $w_{dk^*} = \max\{w_{dk} \mid k \in M \text{ and } \sum_{i \in I_k \cup VC_d} f_i \leq F_k\}$.
    Set $a_i := k^*$ for all $i \in VC_d$.
    Set $I_{k^*} := I_{k^*} \cup VC_d$.
Otherwise
  Sort the components such that $\sum_{i \in VC_1} f_i \geq \ldots \geq \sum_{i \in VC_c} f_i$.
  For $d = 1, \ldots, c$.
    Select k^* such that
    $w_{dk^*} = \max\{w_{dk} \mid k \in M \text{ and } \sum_{i \in I_k \cup VC_d} f_i \leq F_k\}$.
    If such k^* exists
      Set $a_i := k^*$ for all $i \in VC_d$.
      Set $I_{k^*} := I_{k^*} \cup VC_d$.
    Otherwise the procedure fails and stop.
Return a.

In Table 8.4 we show the behavior of this heuristic procedure for some problem instances. We let our program run with the separation routine for the Net Inequalities until no more violated inequality is found, and use the LP solution of each iteration as input for the Bin-Packing Heuristic. In the second column we show the best solution we know for these instances. As we can see in Table 8.4 the heuristic fails in many cases. A possible reason for that is the fact that the quality of the solution produced depends strongly on the "amount of information" given by the net variables.

Problem	Bin-Packing Heur.	Best Solution
pd1.50.1511	fails	109702
pd1.100.1511	124063	122875
pd1.253.200	51246	51019
pd2.150.3326	fails	26824
pd2.300.3326	fails	39191
pd2.772.750	22680	22680

Table 8.4: Bin-Packing Heuristic.

8.2.2 Improvement heuristic

The idea used in this heuristic is very similar to the improvement heuristic suggested in Section 6.3.3. The difference here is in the evaluation of the gain in each pass. In the case of the heuristic described in this section, we calculate the difference of the number of nets in cut caused by moving some item from a knapsack to another one. This number is used as the gain of each move. In the following we present the heuristic. Before that we introduce following notation. Given a vector $y' \in M^N$, that represents a partition of the set N, we denote $\delta(y) := \{t \in Z \mid$ there exits two knapsacks k_1 and k_2 such that $i, j \in T_t$ and $y'_i = k_1$ and $y'_j = k_2\}$.

Improvement Heuristic
Input: An instance $(G = (N, Z), M, f, F, g, S)$ of the MPH, a feasible solution $y \in M^N$.
Output: A solution $a' \in M^N$.

Initialize a' with a.
for $p := 1$ to *number-of-passes*.
  Initialize a^p with a'.
  Set $F'_k := F_k - \sum_{i:a'_i=k} f_i$ for all $k \in M$.
  All items are supposed to be unlocked.
  While $P := \{(i, k) \in N \times M \mid i$ unlocked , $f_i \leq F'_k$ and $a^p_i \neq k\} \neq \emptyset$.
    Set $a^n_j(i, k) := \begin{cases} a^p_j & i \neq j, \\ k & i = j. \end{cases}$
    Define $gain(i, k) := |\delta(a^n(i, k))| - |\delta(a^p)|$ for all $(i, k) \in P$.
    Determine i^* and k^* such that
    $gain(i^*, k^*) = \max\{gain(i, k) \mid (i, k) \in P\}$.
    Assign i^* to k^*.
    Update current solution a^p and F'.
    Lock item i^* (to avoid cycling).
  Determine a^* as the best solution found in this pass.

If a^* has a better value than a', let $a' := a^*$.
Else return a' and stop.
Return a'.

In Table 8.5 we show the improvement achieved by applying this heuristic to the examples of Table 8.4. In Table 8.5 we observe that the combination of

Problem	Bin-Packing Heur. + Improvement Heur.	Best Solution
pd1.50.1511	**110387**	109702
pd1.100.1511	**123717**	122875
pd1.253.200	**51243**	51019
pd2.150.3326	**26869**	26824
pd2.300.3326	**39387**	39191
pd2.772.750	22680	22680

Table 8.5: Improvement Heuristic.

the two heuristic procedures find acceptable solutions. The values where some improvement was achieved are depicted in bold face. This combination is used for the solution of the instances presented in Section 9.3, though the improvement heuristic is time consuming. This effect can be observed in Table 8.9.

8.3 Computational results

In this section we report on the results of our Branch and Cut Algorithm applied to some practical instances of the problem described in Chapter 2. Table 8.6 summarizes the data of the practical problem instances. We have obtained these instances in a joint project with Siemens Nixdorf and the data is classified. In

Problem	$\|N\|$	$\|M\|$	$\|Z\|$	# variables
pd1	253	4	1511	7056
pd2	772	6	3326	24588

Table 8.6: Description of problem instances.

Table 8.7 we show the number of nets in each example with the given cardinality, i. e., in the second column is the number of nets with cardinality 1 (these nets are eliminated in the preprocessing), in third column the number of nets with cardinality 2, and so on. We can observe that most nets have small cardinalities.

Cardinality	pd1	pd2
1	3	266
2	543	549
3	395	1030
4	349	440
5	92	464
6	107	292
7	1	106
8	0	88
9	1	31
10	11	37
11	9	3
12	0	1
13	0	1
14	0	4
15	0	1
16	0	7
17	0	1
19	0	1
21	0	2
25	0	2
Total	1511	3326

Table 8.7: Distribution of the nets.

In order to test the performance of our algorithm to more instances, we have generated from these examples 18 further examples where a subset of the items set or of the net set is taken. The subset is taken with the first items (resp. nets) of the original problems. We use the notation pd$i.n.z$ to denote the reduced problems, where i denote if it is a reduced instance of the example pd1 or pd2, n denote the number of items and z the number of nets involved in the reduced instance. We use the same objective function described in Chapter 2 for the reduced examples. In the instances where a subset of the items set is chosen we have reduced also the total available capacity of the knapsacks accordingly.

In problem pd1 all knapsacks are equal, i.e., all knapsacks have the same area and cut capacities and also the same costs. As described in Chapter 2, the cost associated with external wires is equal for all knapsacks. In this case, the problem reduces to minimizing the number of external wires caused by the partition. This is not true for problem pd2. In this case, there are two types of knapsacks, k_1 and k_2 say. Knapsack k_1 has a smaller area capacity ($F_{k_1} = 0.2F_{k_2}$) and bigger cut capacity ($S_{k_1} = 3S_{k_2}$). Moreover, k_1 is "more expensive" ($K_{k_1} = 5K_{k_2}$). One can expect that problem pd2 is much harder than problem pd1.

In Table 8.8 we summarize the results we have obtained for these examples. In all examples we run the Net Inequalities separator until no more inequalities of this type are violated, and, afterwards, use the procedures for separation of individual, joint, Tree and Cycle Inequalities as described in Chapter 6 and Section 9.1. In this second step we restrict the CPU Time used to two hours.

Problem	lower bound	upper bound	gap
pd1.50.1511	108664	109702	1038 (0.96%)
pd1.100.1511	122151	122875	724 (0.59%)
pd1.150.1511	184251	184745	494 (0.27%)
pd1.200.1511	209273	209810	537 (0.26%)
pd1.253.200	50990	51019	29 (0.06%)
pd1.253.500	100987	101224	237 (0.23%)
pd1.253.750	134440	134903	463 (0.34%)
pd1.253.1000	173267	173822	555 (0.32%)
pd1.253.1200	191266	191818	522 (0.29%)
pd1	236128	236879	751 (0.32%)
pd2.150.3326	26465	26824	359 (1.36%)
pd2.300.3326	38387	39191	804 (2.09%)
pd2.450.3326	47057	47824	767 (1.63%)
pd2.600.3326	60296	60774	478 (0.79%)
pd2.772.750	22680	22680	0 (0.00%)
pd2.772.1200	30693	31400	707 (2.30%)
pd2.772.1600	37248	38313	1065 (2.86%)
pd2.772.2000	48168	49047	879 (1.82%)
pd2.772.2500	62161	63201	1040 (1.67%)
pd2	79544	80754	1210 (1.52%)

Table 8.8: Computational results.

The results obtained provide acceptable solutions (with maximum deviation under 3% from the optimum solution) for all examples tested. As expected, the results for examples from pd1 are better than for pd2. For the examples coming from pd1 we are able to guarantee a maximal deviation from the solution given and the optimal one under 1%. In Table 8.9 we show the distribution of the total CPU Time in the time for LP solution, heuristic procedures and separation procedures.

Problem	LP Time	Heur. Time	Sep. Time
pd1.50.1511	56.2%	15.5%	26.1%
pd1.100.1511	36.1%	34.2%	27.7%
pd1.150.1511	21.7%	52.0%	24.2%
pd1.200.1511	25.9%	43.3%	28.0%
pd1.253.200	46.3%	22.0%	29.2%
pd1.253.500	41.8%	33.7%	22.3%
pd1.253.750	28.4%	47.8%	21.7%
pd1.253.1000	36.0%	44.2%	16.8%
pd1.253.1200	28.2%	47.0%	21.5%
pd1	18.2%	56.4%	21.3%
pd2.150.3326	49.6%	32.8%	15.2%
pd2.300.3326	31.4%	55.0%	10.4%
pd2.450.3326	60.3%	24.1%	11.5%
pd2.600.3326	22.4%	60.1%	11.7%
pd2.772.750	21.3%	72.1%	2.3%
pd2.772.1200	6.3%	81.3%	8.0%
pd2.772.1600	11.7%	75.2%	8.9%
pd2.772.2000	14.0%	80.1%	4.4%
pd2.772.2500	8.0%	87.9%	2.9%
pd2	15.0%	69.2%	11.0%

Table 8.9: Distribution of computer time.

In Table 8.9 we can verify that most of the CPU Time is used in the LP solution and heuristic procedures. This performance is justified by the time consuming improvement heuristic we used for all examples (described in Section 9.2.2).

We have also run our algorithm with another choice of the objective function, namely, 0 for all node variables and 1 for the net variables, trying to obtain lower and upper bounds to the number of external wires in a valid partition. The results for some examples are shown in Table 8.10.

Table 8.10 shows that we were not able to repeat the good results for this different choice of the objective function. One explanation is the fact that the cost associated with a external wire is relatively small comparing to the costs associated with different knapsacks in the objective function of interest for the practical application. This shows that it remains still very much to be done for this type of problems, not only from a theoretical point of view but also from the aspect of casting the theory in an efficient code.

Problem	lb	ub	gap
pd1.50.1511	226	581	255
pd1.100.1511	170	395	225
pd1.253.200	13	17	4
pd2.150.3326	48	217	169
pd2.772.750	0	0	0
pd2.772.1200	25	59	34

Table 8.10: Bounds on external wires.

Summary

In this section we give an overview on our thesis and how our approaches help in the solution of the real world problem we have investigated.

In this thesis we present two difficult combinatorial optimization problems arising in some practical application in computer system design: the Multiple Knapsack Problem MKP and the Multicut Problem on Hypergraphs MPH. The practical application that motivated this thesis is described in Chapter 2.

In order to solve the problems arising in this practical application we suggest Branch and Cut Algorithms. We investigate the properties of polytopes associated with the problems and derive several results on valid and facet defining inequalities for these polytopes. In Chapters 5 and 8 we present many classes of valid and facet defining inequalities for these polytopes.

The corresponding Separation Problems have also been investigated. In Chapter 3, we show that the Separation Problem for the Minimal Cover Inequalities is $\mathcal{NP}$-hard. For the inequality classes we introduce, the corresponding Separation Problems remain open. In Chapters 6 and 9 we describe several heuristic routines for the separation of such inequalities.

In the case of the MKP, we were able to solve to optimality all instances coming from practical applications in acceptable computer times. Experimenting with the objective function coefficients and random generated problems, we could verify that our separation routines produce a good lower bound for most cases tested.

We have obtained similar results for the instances of MPH arising from the practical applications. Nevertheless, experiments with different objective functions show that there is still very much to be done in the theoretical investigation of the polytope and the practical implementation of the algorithm for this problem.

Finally, we make some comments on the original problem that motivated this thesis. As described in Chapter 2, the idea was to investigate simplifications of the exact model, and how the corresponding solutions apply for it. In general, we cannot guarantee that an optimal solution of the MKP provides a feasible solution of the original problem, since the constraints involving external wires are completely neglected. The solution of the MPH, however, can be used by the computer designers as a starting point. It is not necessarily an optimal solution for the original problem, since, for example, we have approximated the original piecewise

linear objective function by a linear one. Further, the conditions corresponding to the treatment of r-s nets (see Chapter 2) are ignored. These conditions can be checked in a post optimization phase and one can try to obtain a feasible solution by exchanging some items. It can occur that the computer designer is not satisfied with the solution given since the approximation we use for the objective function is too rough. In this case, we can change the objective function and use our algorithm iteratively. This is an usual approach in computer aided design. In order to obtain a best possible approximation to the real world problem we should optimize over the exact model given in Chapter 2. To attack this model, however, is still hopeless for the current stand of the theory of 0/1 programming.

Bibliography

[AH94] A. Abdel Hamid, "Combinatorial Optimization Problems Arising in the Design and Management of an Automatic Storage System", *Ph.D. Dissertation*, Technische Universität Berlin (1994).

[B75] E. Balas, "Facets of the Knapsack Polytope", *Mathematical Programming* 8, 146 - 164 (1975).

[BZ80] E. Balas and E. Zemel, "An Algorithm for Large Zero-One Knapsack Problems", *Operations Research* 28, 1130 - 1154 (1980).

[BZ78] E. Balas and E. Zemel, "Facets of the Knapsack Polytope from Minimal Covers", *SIAM Journal on Applied Mathematics* 34, 119 - 148 (1978).

[BZ84] E. Balas and E. Zemel, "Lifting and Complementing Yields All the Facets of Positive Zero-One Programming Polytopes", *Mathematical Programming*, R.W. Cottle eds., Elsevier Science Publishers (1984).

[Ba83] F. Barahona, "The Max Cut Problem in Graphs Not Contractible to K_5", *Operations Research Letters* 2, 107 - 111 (1983).

[BGJR88] F. Barahona, M. Grötschel, M. Jünger and G. Reinelt, "An Application of Combinatorial Optimization to Statistical Physics and Circuit Layout Design", *Operations Research* 36, 493 - 513 (1988).

[BGM85] F. Barahona, M. Grötschel and R. Mahjoub, "Facets of the Bipartite Subgraph Polytope", *Math. Operations Research* 10, 340 - 358 (1985).

[BM86] F. Barahona, R. Mahjoub, "On the Cut Polytope", *Math. Programming* 36, 157 - 173 (1986).

[Be54] R. Bellman, "Some Applications of the Theory of Dynamic Programming – a Review", *Operations Research* 2, 275 - 288 (1954).

[Ber76] C. Berge, *Graphs and Hypergraphs*, North-Holland, Amsterdam (1976).

[Bo92] E. A. Boyd, "A Pseudo-Polynomial Network Flow Formulation for Exact Knapsack Separation", *Networks* 22, 503 - 514 (1992).

[CR91] S. Chopra and M. R. Rao, "On the multiway cut polyhedron", *Networks* 21, 51-89 (1991).

[CRS90] M. Conforti, M. R. Rao and A. Sassano, "The Equipartition Polytope I: Formulations, Dimension and Basic Facets", *Mathematical Programming* 49, 49 - 70 (1990).

[CRS90a] M. Conforti, M. R. Rao and A. Sassano, "The Equipartition Polytope II: Valid Inequalities and Facets", *Mathematical Programming* 49, 71 - 90 (1990).

[CPLEX92] *Using the CPLEX Callable Library and CPLEX Mixed Integer Library*, CPLEX Optimization (1992).

[Cr90] A. Crema, "Un Algoritmo para el Problema de Asignación Generalizado con Multiples Recursos", manuscript (1990) (in Spanish).

[CJP83] H. Crowder, E. L. Johnson and M. W. Padberg, "Solving Large-Scale Zero-One Linear Programming Problems", *Operations Research* 31, 803 - 834 (1983).

[D57] G. B. Dantzig, "Discrete Variables Extremum Problems", *Operations Research* 5, 266 - 277 (1957).

[DFJ54] G. B. Dantzig, R. Fulkerson and S. M. Johnson, "Solution of a Large Scale Traveling Salesman Problem", *Operations Research* 2, 393 - 410 (1954).

[DGL91] M. Deza, M. Grötschel and M. Laurent, "Complete Descriptions of Small Multicut Polytopes", *Applied geometry and discrete mathematics, Festschr. 65th Birthday Victor Klee*, DIMACS, Ser. Discret. Math. Theor. Comput. Sci. 4, 221-252 (1991).

[DL91] M. Deza and M. Laurent, "A Survey of the Known Facets of the Cut Cone", *Technical Report* 91722, Institut für Ökonometrie und Operations Research, Universität Bonn (1991).

[EC72] S. Eilon and N. Christofides, "The Loading Problem", *Management Science* 18, 432 - 433 (1972).

[FGKKMW93] C. E. Ferreira, M. Grötschel, S. Kiefl, C. Krispenz, A. Martin and R. Weismantel, "Some Integer Programs Arising in the Design of Main Frame Computers", *Zeitschrift für Operations Research* 38, 77 - 100 (1993).

[FMSWW93] C. E. Ferreira, A. Martin, C. C. de Souza, R. Weismantel and L. A. Wolsey, "Formulations and Valid Inequalities for Node Capacitated Graph Partitioning", in preparation (1993).

[FMSWW93a] C. E. Ferreira, A. Martin, C. C. de Souza, R. Weismantel and L. A. Wolsey, "A Branch and Cut Algorithm for Node Capacitated Graph Partitioning", in preparation (1993).

[FMW93] C. E. Ferreira, A. Martin and R. Weismantel, "Facets for the Multiple Knapsack Polytope", Konrad-Zuse-Zentrum für Informationstechnik Berlin *Preprint* SC 93-04 (1993), submitted.

[FMW93a] C. E. Ferreira, A. Martin and R. Weismantel,"A Cutting Plane Based Algorithm for the Multiple Knapsack Problem", Konrad-Zuse-Zentrum für Informationstechnik Berlin, *Preprint* SC 93-07 (1993), submitted.

[FM82] C. M. Fiduccia and R. M. Mattheyses, "A Linear-Time Heuristic for Improving Network Partitions", *Proc. 19th. DAC*, 175 - 181 (1982).

[FJ81] M. L. Fisher and R. Jaikumar, "A Generalized Assignment Heuristic for Vehicle Routing", *Networks* 11, 109 - 124 (1981).

[FJW86] M. L. Fisher, R. Jaikumar and L. N. van Wassenhove, "A Multiplier Adjustment Method for the Generalized Assignment Problem", *Management Science* 12, 1095 - 1103 (1986).

[FH79] J. C. Fisk and M. S. Hung, "A Heuristic Routine for Solving Large Loading Problems", *Naval Research Logistics Quarterly* 26, 643 - 650 (1979).

[FC84] A. M. Frieze and M. R. B. Clarke, "Approximation Algorithms for the m-Dimensional 0-1 Knapsack Problem: Worst Case and Probabilistic Analyses", *European Journal of Operational Research* 15, 100 - 109 (1984).

[GJ79] M. R. Garey and D. S. Johnson, *Computer and Intractability - A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, New York (1979).

[GJS76] M. R. Garey, D. S. Johnson and L. Stockmeyer, "Some Simplified $\mathcal{NP}$-complete Graph Problems", *Theoretical Computer Science* 1, 237-267 (1976).

[Ge85] A. M. H. Gerards, "Testing the Odd Bicycle Wheel Inequalities for the Bipartite Subgraph Polytope", *Math. of Operations Research* 10, 359 - 360 (1985).

[GG66] P. C. Gilmore and R. E. Gomory, "The Theory and Computation of Knapsack Functions", *Operations Research* 14, 1045 - 1074 (1966).

[Go58] R. E. Gomory, "Outline of an Algorithm for Integer Solution to Linear Programs", *Bulletin of the American Mathematical Society* 64, 275 - 278 (1958).

[GR88] E. S. Gottlieb and M. R. Rao, "Facets of the Knapsack Polytope Derived from Disjoint and Overlapping Index Configurations", *Operations Research Letters* 7, 95 - 100 (1988).

[GR90] E. S. Gottlieb and M. R. Rao, "The Generalized Assignment Problem: Valid Inequalities and Facets", *Mathematical Programming* 46, 31 - 52 (1990).

[GR90a] E. S. Gottlieb and M. R. Rao, "(1,k)-Configuration Facets for the Generalized Assignment Problem", *Mathematical Programming* 46, 53 - 60 (1990).

[GJR85] M. Grötschel, M. Jünger and G. Reinelt, "A Cutting Plane Algorithm for the Linear Ordering Problem", *Operations Research* Volume 32 no 6, 1195 - 1220 (1985).

[GJR89] M. Grötschel, M. Jünger and G. Reinelt, "Via Minimization with Pin Preassignments and Layer Preference", *Zeitschrift für Angewandte Mathematik und Mechanik* 69, 393 - 399 (1989).

[GW89] M. Grötschel and Y. Wakabayashi,"A Cutting Plane Algorithm for a Clustering Problem", *Mathematical Programming B* 45, 59 - 96 (1989).

[GW90] M. Grötschel and Y. Wakabayashi,"Facets for the Clique Partitioning Polytope", *Mathematical Programming A* 47, 367 - 388 (1990).

[HJP75] P. L. Hammer, E. L. Johnson and U. N. Peled, "Facets of Regular 0-1 Polytopes", *Mathematical Programming* 8, 179 - 206 (1975).

[HP82] P. L. Hammer and U. N. Peled, "Computing Low Capacity 0-1 Knapsack Polytopes", *Zeitschrift für Operations Research* 26, 243 - 249 (1982).

[HZ92] D. Hartvigsen and E. Zemel, "The Complexity of Lifted Inequalities for the Knapsack Problem", *Discrete Applied Mathematics* 39, 113 - 123 (1992).

[HP91] K. L. Hoffman and M. Padberg, "Improving LP-Representations of Zero-One Linear Programs and Branch and Cut", *ORSA Journal on Computing* 3, 121 - 134 (1991).

[HS74] E. Horowitz and S. Sahni, "Computing Partitions with Applications to the Knapsack Problem", *Journal of the ACM* 21, 277 - 292 (1974).

[HF78] M. S. Hung and J. C. Fisk, "An Algorithm for Zero-One Multiple Knapsack Problems", *Naval Research Logistics Quarterly* 24, 571 - 579 (1978).

[IK75] O. H. Ibarra and C. E. Kim, "Fast Approximation Algorithms for the Knapsack and Sum of Subset Problems", *Journal of ACM* 22, 463 - 468 (1975).

[IKo75] G. Ingargiola and J. F. Korsh, "An Algorithm for the Solution of 0-1 Loading Problems", *Operations Research* 23, 1110 - 1119 (1975).

[JMN91] E. L. Johnson, A. Mehrotra and G. L. Nemhauser, "Min-Cut Clustering", *Mathematical Programming B/* 62, 133 - 151 (1993).

[JMRW92] M. Jünger, A. Martin, G. Reinelt and R. Weismantel, "Quadratic 0/1 Optimization and a Decomposition Approach for the Placement of Eletronic Circuits", *Preprint SC 92-10*, Konrad-Zuse-Zentrum für Informationstechnik Berlin (1992) appears in Math. Programming.

[K72] R. M. Karp, "Reducibility among Combinatorial Problems", in R. E. Miller and J. W. Thatcher eds., *Complexity of Computer computations*, Plenum Press, New York, 85 - 103 (1972).

[KS80] B. Korte and R. Schrader, "On the Existence of Fast Approximation Schemes", *Technical Report 80163-OR*, Institut für Ökonometrie und Operations Resarch, Universität Bonn (1980).

[La79] E. L. Lawler, "Fast Approximation Algorithms for Knapsack Problems", *Mathematics of Operations Research* 4, 339 - 356 (1979).

[Le90] T. Lengauer, *Combinatorial Algorithms for Integrated Circuit Layout*, John Wiley and Sons, Chichester (1990).

[MR71] A. O. de Maio and C. A. Roveda, "An All 0-1 Algorithm for a Certain Class of Transportation Problems", *Operations Research* 19, 1406 - 1418 (1971).

[MT77] S. Martello and P. Toth, "An Upper Bound for the Zero One Knapsack Problem and a Branch and Bound Algorithm", *European Journal of Operational Research* 1, 169 - 175 (1977).

[MT80] S. Martello and P. Toth, "Solution of the Zero-One Multiple Knapsack Problem", *European Journal of Operational Research* 4, 322 - 329 (1980).

[MT81] S. Martello and P. Toth, "A Bound and Bound Algorithm for the Zero-One Multiple Knapsack Problem", *Discrete Applied Mathematics* 3, 275 - 288 (1981).

[MT81a] S. Martello and P. Toth, "Heuristic Algorithms for the Multiple Knapsack Problem", *Computing* 27, 93 - 112 (1981).

[MT81b] S. Martello and P. Toth, "An Algorithm for the Generalized Assignment Problem", in J.P. Brans eds., *Operational Research 81*, North-Holland, Amsterdam, 589 - 603 (1981).

[MT91] S. Martello and P. Toth, *Knapsack Problems - Algorithms and Computer Implementations*, John Wiley and Sons, New York (1991).

[OD72] G. I. Orlova and Y. G. Dorfman, "Finding the Maximal Cut in a Graph", *Engrg. Cybernet.* 10, 502 - 506 (1972).

[P75] M. W. Padberg, "A Note on 0-1 Programming", *Operations Research* 23, 833 - 837 (1975).

[P80] M. W. Padberg, "(1,k)-Configurations and Facets for Packing Problems", *Mathematical Programming* 18, 94 - 99 (1980).

[PR87] M. W. Padberg and G. Rinaldi, "Optimization of a 532-city Symmetric Traveling Salesman Problem by Branch and Cut", *Operations Research Letters* 6, 1 - 7 (1987).

[PG85] M. W. Padberg and M. Grötschel, "Polyhedral Aspects of the Travelling Salesman Problem II : Computations", in E. L. Lawler, J. K. Lenstra and A. H. G. Rinooy Kan eds., *The Travelling Salesman Problem*, John Wiley and Sons, London/New York (1985).

[RS75] G. T. Ross and R. M. Soland, "A Branch and Bound Algorithm for the Generalized Assignment Problem", *Mathematical Programming* 8, 91 - 103 (1975).

[Sa75] S. Sahni, "Approximative Algorithms for the 0-1 Knapsack Problem", *Journal of the ACM* 22, 115 - 124 (1975).

[Sai91] S. Saigal, "Optimizing Over the Cut Cone: a New Polyhedral Algorithm for the Maximum-Weight Cut Problem", *Ph.D. Dissertation*, Rice University, Texas (1991).

[Sc87] A. Schrijver, *Theory of Linear and Integer Programming*, John Wiley & Sons, Essex (1987).

[So93] C. C. de Souza, "The Graph Equipartition Problem: Optimal Solutions, Extensions and Applications", *Ph.D. Dissertation*, Universitè Catholique de Louvain la Neuve (1993).

[SKWZ92] C. C. de Souza, R. Keunings, L.A. Wolsey and O. Zone, "A New Approach to Minimising the Frontwidth in Finite Element Calculations", *Discussion paper* 9255, Universitè Catholique de Louvain la Neuve (1992).

[SL91] C. C. de Souza and M. Laurent, "Some New Classes of Facets for the Equicut Polytope", *Technical Report* 91738, Forschungsinstitut für Diskrete Mathematik, Universität Bonn (1991).

[ST73] V. Srinivasan and G. L. Thompson, "An Algorithm for Assigning Uses to Sources in a Special Class of Transportation Problems", *Operations Research* 21, 284 - 295 (1973).

[Wa86] Y. Wakabayashi, "Aggregation of Binary Relations: Algorithmic and Polyhedral Investigations", *Ph.D. Dissertation*, Universität Augsburg (1986).

[We92] R. Weismantel, "Plazieren von Zellen: Analyse und Lösung eines quadratischen 0/1-Optimierungsproblems", *Ph.D. Dissertation*, Technische Universität Berlin (1992) (in German).

[W75] L. A. Wolsey, "Faces of Linear Inequalities in 0-1 Variables", *Mathematical Programming* 8, 165 - 178 (1975).

[W90] L. A. Wolsey, "Valid Inequalities for 0-1 Knapsacks and MIPS with Generalised Upper Bound Constraints", *Discrete Applied Mathematics* 29, 251 - 261 (1990).

[Ze89] E. Zemel, "Easily Computable Facets of the Knapsack Polytope", *Mathematics of Operations Research* 14, 760 - 764 (1989).

Index

Lebenslauf

Carlos Eduardo Ferreira

21. April 1965	geboren in São Paulo (Brasilien) als Sohn von Mario Ferreira und Olivia Bernardo Ferreira.
1972 bis 1979	Besuch des "Primeiro Grau" an der Schule "Ginásio Santa Gema" in São Paulo, Brasilien.
1980 bis 1982	Besuch des "Segundo Grau" an der Schule "Colégio Jardim São Paulo" in São Paulo, Brasilien.
1983 bis 1986	Studium der Informatik am Institut für Mathematik und Statistik der Universität São Paulo, Brasilien.
1987 bis 1989	Aufbaustudium in Angewandter Mathematik am Institut für Mathematik und Statistik der Universität São Paulo. Grad: "Mestre"(äquivalent zu Diplom in Mathematik). Titel der Arbeit: *O problema de Steiner em grafos: uma abordagem poliédrica.*
Seit 1987	Tätig als wissenschaftliche Hilfskraft (1987 bis 1989) und wissenschaftlicher Mitarbeiter (seit 1989) am Institut für Mathematik und Statistik der Universität São Paulo.
Seit April 1990	in der Bundesrepublik Deutschland. Von April bis September 1990 Sprachkurs am Goethe-Institut Mannheim. Von Oktober 1990 bis September 1991 Doktorand an der Universität Augsburg. Seit Oktober 1991 Doktorand bei Prof. Dr. Martin Grötschel an der Technische Universität Berlin. Tätig am Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB).