Some Integer Programs Arising in the Design of Main Frame Computers

C.E. Ferreira M. Grötschel S. Kiefl L. Krispenz
A. Martin R. Weismantel

Abstract

In this paper we describe and discuss a problem that arises in the (global) design of a main frame computer. The task is to assign certain functional units to a given number of so called multi chip modules or printed circuit boards taking into account many technical constraints and minimizing a complex objective function. We describe the real world problem. A thorough mathematical modelling of all aspects of this problem results in a rather complicated integer program that seems to be hopelessly difficult – at least for the present state of integer programming technology. We introduce several relaxations of the general model, which are also $\mathcal{NP}$-hard, but seem to be more easily accessible. The mathematical relations between the relaxations and the exact formulation of the problem are discussed as well.

1 Introduction

This paper is the first in a series that grew out of a cooperation of Siemens Nixdorf with a research group at the Konrad-Zuse-Zentrum Berlin. The paper addresses certain problems that arise in the design of main frame computers. The topic considered here comes up in the phase of the global design where certain functional units (components) have been defined, where the networks connecting the components have been determined and where decisions are made such as how to group the components and how to integrate them on a given number of multi chip modules and/or printed circuit boards. Decisions of this type are initially rather tentative and are iteratively (and frequently) reconsidered after making design changes or further progress in the details of the global design or of the component layout.

The mathematical problem that arises by thoroughly modelling all (or at least the most important) aspects of this question is a rather complicated integer program with almost a million variables. The aim of this paper is a description of the real world problem and our way of modelling it mathematically.

The full model appears to be hopelessly difficult – at least for the present state of integer programming technology. In a series of follow up papers we will investigate a hierarchy of combinatorial relaxations of the complete model. Each of these relaxations is $\mathcal{NP}$-hard itself, but, for the instance sizes considered here, structurally simpler and easier to solve. We will study these relaxed models theoretically using the theory of polyhedral combinatorics and develop algorithms for their solution with the aim of combining these models and algorithms to obtain an algorithmic machinery that provides provably good approximate solutions of our real problem.

The overall organization of the paper is as follows:

In section 2 we describe the practical problem in full detail and, simultaneously, our way of mathematically modelling it. The high complexity of the model as well as its large size give rise to the study of relaxations of this model by discarding some of its side constraints as well as some of its variables. Here, a first relaxation consists of the multiple knapsack problem, which can be viewed as the task of assigning a given set of items to a given set of knapsacks. The second relaxation extends the multiple knapsack problem in the sense that nets running between different modules are approximately taken into account. The last relaxation we are going to consider improves the approximation of the nets and leads to a hypergraph clustering problem. All these relaxations are explained in section 3. Section 4 discusses the mathematical relations between the relaxations and the exact formulation of the problem.

2 The General Model

In this section we will first present a complete description of the practical problem. It will turn out that some of the side constraints can be satisfied in a preprocessing phase. This will be explained in the second subsection. We will then describe our model of the problem, i.e., we will formulate the overall problem as a 0/1 integer program.

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* $f_i \in \mathbb{R}_+$.

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 etc. 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. It is customary to call a collection of pins that have to be connected 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 $T_t \in \mathcal{Z}$.

There are some nets t, where it is necessary to partition the set of items 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:*
  Some of the nets must satisfy one of the following two rules. These rules read, for some net $t \in Z$, as follows.

  (R1) All items of T_t must be assigned to the same module.

  (R2) Either all items of T_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 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; and we refrain from explaining the details here.

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). For each module k with $B(k) \cap T_t \neq \emptyset$ an additional pin at the border of the module, a so-called *external pin*, is introduced (see the black rectangles in Figure 1). A routing for net t is obtained by connecting the items of $B(k) \cap T_t$ within each module with the external pin (see dotted lines in Figure 1) and by connecting the external pins via a so-called *external wire* (see dashed lines in Figure 1).

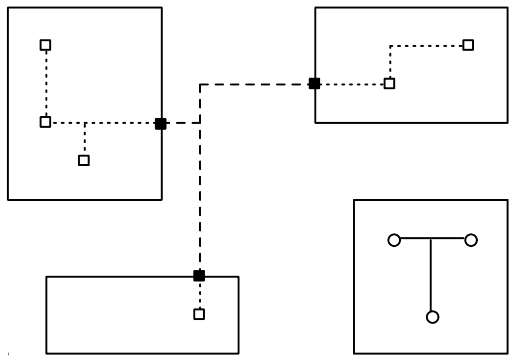

Figure 1:

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

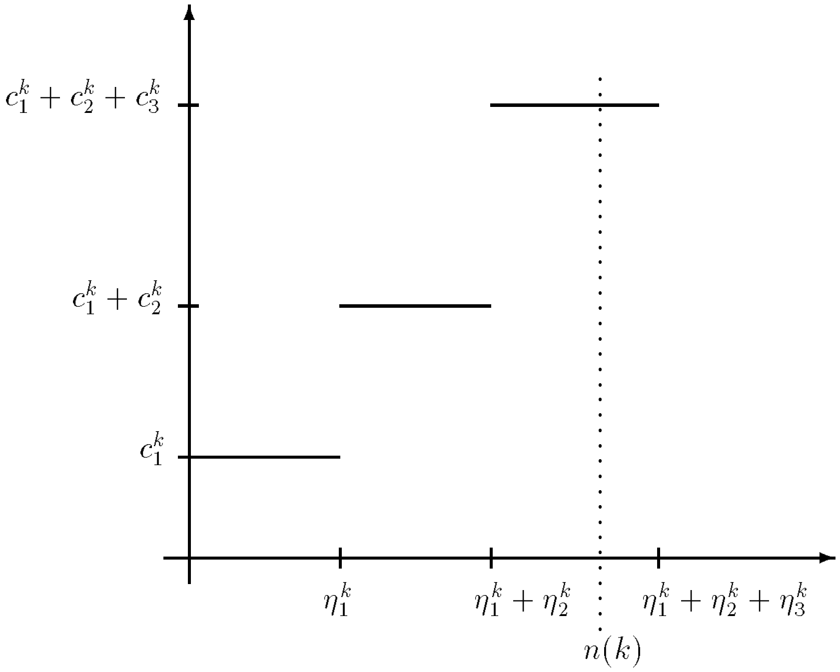

Figure 2: The internal cost function I_k of some module k

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*.

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 larger the number $n(k)$ is, the more layers are necessary. 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 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 that we treat here 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 $t \in Z = \{1, \ldots, z\}$.

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 taken care of easily. We do this in a preprocessing stage and describe here two such cases concerning the net constraints.

Preprocessing

Suppose t is a net that must satisfy net constraint (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 $N' := (N \setminus T_t) \cup \{i'\}$. 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 net that must meet (R2). 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 net t which must satisfy net constraint (R2) 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.

The 0/1 Program

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 y_{tk}-, y^1_{tk}- and y^2_{tk}-variables. 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:

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

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.

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

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

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

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

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

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

$$x_{sk} + x_{ik} \leq 2 - y_{tk}^1, \quad \begin{array}{l} \text{for all nets } t \in Z \text{ that must satisfy} \\ \text{net constraint (R2) and for all } i \in R_t \\ \text{(where } \{s\} = S_t). \end{array} \tag{2.4}$$

These inequalities are derived from the following reasoning. If $y_{tk}^1 = 0$, that is all items of net t are assigned to module k, inequality (2.4) is obviously valid. On the other hand, if $y_{tk}^1 = 1$, 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 costraints (R2) are met. 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.

$$\sum_{i \in T_t} x_{ik} + |T_t| y_{tk}^1 \geq |T_t|, \quad \text{for all } k \in M,\ t \in Z, \tag{2.5}$$

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_{tk}^1 must be one.

$$\sum_{i \in T_t} x_{ik} + y_{tk}^1 \leq |T_t|, \quad \text{for all } k \in M,\ t \in Z, \tag{2.6}$$

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.

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

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.

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

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.

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

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.

$$(2.10) \qquad \begin{aligned} &\sum_{t\in Z}\sum_{i\in T_t}(x_{ik} + y_{tk} - y^2_{tk}) - \sum_{l=1}^{l_k} \eta^k_l v^k_l \le 0, \\ &\qquad v^k_1 \ge v^k_2 \ge \ldots \ge v^k_{l_k}, \quad \text{for all } k \in M. \end{aligned}$$

It is easy to see that $w_k(t) = \sum_{i\in T_t}(x_{ik} - y_{tk} - y^2_{tk})$. This implies that $n(k) = \sum_{t\in Z}\sum_{i\in T_t}(x_{ik} - y_{tk} - y^2_{tk})$. 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.

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

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

$$\min \sum_{k\in M} K_k \sum_{l=1}^{l_k} c_l^k v_l^k \;+\; \lambda \sum_{t\in Z} \sum_{k\in M} y_{tk}. \tag{2.12}$$

This objective function corresponds to the one depicted in $(*)$. This follows from the fact that constraints (2.10) ensure that, for every $k \in M$, the expression $\sum_{l=1}^{l_k} c_l^k v_l^k$ 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 $p(t)$.

3 Relaxations for the General Model

The Multiple Knapsack Problem

In this subsection we present a first relaxation of the general model. Here, we neglect the nets completely and concentrate on the packing aspect of the problem. More precisely, let us introduce the variables $x_{ik}, i \in N, k \in M$, with the interpretation

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

Using these variables we can model the requirement that every item is assigned to some module such that the capacities of the modules are not exceeded. This can be expressed in the following inequalities:

$$\sum_{k\in M} x_{ik} = 1, \text{ for all } i \in N. \tag{3.1}$$

$$\sum_{i\in N} f_i x_{ik} \le F_k, \text{ for all } k \in M, \tag{3.2}$$

saying that the sum of weights corresponding to the items assigned to module k must not exceed the capacity of module k.

$$x_{ik} \in \{0,1\}, \text{ for all } i \in N, k \in M. \tag{3.3}$$

The objective function estimates the cost for the number of wires running within some module. More precisely, we approximate the staircase function I_k, introduced in section 2, via

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

Then, the objective function can be stated as follows:

$$\text{(3.4)} \qquad \min \quad \sum_{k \in M} D_k \sum_{t=1}^{z} (\sum_{i \in T_t} x_{ik}).$$

This is only a rough estimate, since wires running between different modules are completely neglected. Also, some net t for which all items of T_t are assigned to the same module, k say, contributes to the overall objective value the amount $D_k|T_t|$, rather than the amount $D_k(|T_t| - 1)$.

Let us now perform a transformation that shows that the model defined via (3.1), ..., (3.4) is equivalent to a well known combinatorial problem, namely the *multiple knapsack problem.* We replace conditions (3.1) by

$$\text{(3.1}'\text{)} \qquad \sum_{k \in M} x_{ik} \leq 1 \text{ for all } i \in N,$$

by slightly modifying the objective function (3.4) using a transformation that was employed in [MT91]. More precisely, with every constraint (3.1) we associate a slack variable $\gamma_i, i \in N$, and introduce a number $Q := \sum_{k \in M} D_k |N||M||Z|$. Now, consider the optimization problem

$$\begin{aligned} \min \quad & \sum_{k \in M} D_k \sum_{t \in Z} (\sum_{i \in T_t} x_{ik}) + \sum_{i \in N} Q\gamma_i \\ s.t. \quad & \sum_{k \in M} x_{ik} + \gamma_i = 1, i \in N \\ & \text{and } x \text{ satisfies (3.2), (3.3)} \\ & \gamma_i \in \{0, 1\}, i \in N. \end{aligned}$$

Clearly, in every optimal solution of this problem, all variables γ_i are equal to zero. Thus, this optimization problem is equivalent to the one defined by (3.1), (3.2), (3.3), (3.4). If we now eliminate the variables γ_i again by substituting $\gamma_i = 1 - \sum_{k \in M} x_{ik}$, we obtain, up to a constant term $|N|Q$, an equivalent formulation:

$$
\begin{aligned}
\min \quad & \sum_{k \in M} \sum_{i \in N} c_{ik} x_{ik} \\
s.t \quad & \sum_{k \in M} x_{ik} \leq 1, i \in N \\
& \text{and } x \text{ satisfies } (3.2), (3.3),
\end{aligned}
$$

where $c_{ik} := D_k |\{t \in Z \mid i \in T_t\}| - Q$.

The latter problem is the multiple knapsack problem and hence, this multiple knapsack problem is a (rather coarse) combinatorial relaxation of our module design problem.

The Graph Clustering Problem

To estimate the number of nets running between different modules we extend the previous model by introducing an additional class of Boolean variables. We define a graph $G = (V, E)$, where $V = N$ is the set of nodes (representing the items) and the edge set E is the set of pairs ij such that items i and j are simultaneously contained in at least one net T_t. For every edge $ij \in E$ we introduce a 0/1-variable η_{ij} with the interpretation:

$$
\eta_{ij} = \begin{cases} 1, & \text{if items } i \text{ and } j \text{ are assigned} \\ & \text{to different modules,} \\ 0, & \text{otherwise,} \end{cases}
$$

Our second relaxation has the following form.

The set of constraints (3.5) – (3.7) models – as before – that every item is assigned to some module and that the capacities of the modules are not exceeded.

$$
\sum_{k \in M} x_{ik} = 1, \text{ for all } i \in N, \tag{3.5}
$$

$$
\sum_{i \in N} f_i x_{ik} \leq F_k, \text{ for all } k \in M, \tag{3.6}
$$

$$(3.7) \qquad x_{ik} \in \{0,1\}, \text{ for all } i \in N,\ k \in M.$$

In addition to these constraints we must introduce side constraints that reflect the linking between the x- and η-variables. This can be done as follows:

$$(3.8a) \qquad x_{ik} + x_{jl} - \eta_{ij} \leq 1, \text{ for all } ij \in E, k, l \in M, k \neq l,$$

$$(3.8b) \qquad x_{ik} + x_{jk} + \eta_{ij} \leq 2, \text{ for all } ij \in E, k \in M.$$

Finally, we must require that the η-variables are either zero or one.

$$(3.9) \qquad \eta_{ij} \in \{0,1\}, \text{ for all } ij \in E.$$

The constraints of type (3.8a) ensure that, whenever some item i is assigned to some module k and some item j is assigned to some module l different from k, the variable η_{ij} must be set to one. Similarly, if both of the items i and j are assigned to the same module, the corresponding variable η_{ij} must be equal to zero. This is the interpretation of the constraints (3.8b).

For the exposition of the objective function, let λ denote the cost for a wire that runs between different modules (cf. section 2). Moreover, for every $ij \in E$ let us introduce some weighting factor τ_{ij}. This parameter, to be set by the designer, should reflect the number of times items i and j appear in some net and the cardinalities of the nets involved. By using this parameter τ_{ij}, we try to avoid that nets of high cardinality are completely overestimated. With the coefficients D_k as introduced in the previous model, the objective function can be formulated as follows:

$$(3.10) \qquad \min \quad \sum_{k \in M} D_k \sum_{t \in Z} \Big(\sum_{i \in T_t} x_{ik} \Big) + \lambda \sum_{ij \in E} \tau_{ij} \eta_{ij}.$$

Let us briefly discuss this model.

First of all, the cut constraints are still completely neglected. Moreover, the objective function provides only a very rough estimate of the original one, yet obviously improves the one of the previous model.

In graphtheoretic terms, every feasible solution of this model can be interpreted as follows. For every $k \in M$, we set $P_k := \{i \in V \mid x_{ik} = 1\}$. Let $M' := \{k_1, \ldots, k_{m'}\} \subseteq M$ denote the set of indices such that $P_k \neq \emptyset$, i.e., the set of modules to which at least one item is assigned. Then, $(P_{k_1}, \ldots, P_{k_{m'}})$ is a partition of the node set V into no more than m clusters such that certain knapsack constraints are satisfied. The set $E' := \{ij \in E \mid \eta_{ij} = 1\}$ corresponds to the set of edges $ij \in E$ such that items i and j are contained in different elements of the partition $(P_{k_1}, \ldots, P_{k_{m'}})$. This problem is called *graph clustering problem* and can be viewed as a generalization of the well known *multicut problem* in graphs.

Finally, we employ the transformation described in the previous subsection to this model, yielding the following optimization problem:

$$\begin{array}{ll} \min & \sum_{k \in M} \sum_{i \in N} c_{ik} x_{ik} + \lambda \sum_{ij \in E} \tau_{ij} \eta_{ij} \\ s.t. & \sum_{k \in M} x_{ik} \leq 1, \text{ for all } i \in N, \\ & \text{and } x, \eta \text{ satisfies } (3.6), (3.7), (3.8a), (3.8b), (3.9). \end{array}$$

The Hypergraph Clustering Problem

The first two relaxations neglect the cut constraints. The third relaxation, presented now, reduces this drawback, although it does not completely take all the constraints into account.

We use again the variables $x_{ik}, i \in N, k \in M$, with the interpretation outlined above. In addition, with every net $t \in Z$ and module $k \in M$ we associate a Boolean variable y_{tk} with the interpretation

$$y_{tk} = \begin{cases} 1, & \text{if some item of } T_t \text{ is assigned to module } k, \\ & \text{but not all of the items belonging to } T_t \\ & \text{are assigned to } k, \\ 0, & \text{otherwise.} \end{cases}$$

Using these two classes of variables the constraints of our third model can be stated as follows:

$$(3.11) \qquad \sum_{k \in M} x_{ik} = 1, \text{ for all } i \in N,$$

$$(3.12) \qquad \sum_{i \in N} f_i x_{ik} \leq F_k, \text{ for all } k \in M,$$

$$(3.13) \qquad x_{ik} \in \{0,1\}, \text{ for all } i \in N, \ k \in M,$$

$$(3.14a) \qquad x_{ik} + x_{jl} - y_{tk} \leq 1, \text{ for all } t \in Z, i, j \in T_t, k, l \in M, k \neq l,$$

$$(3.14b) \qquad \sum_{i \in T_t} x_{ik} + y_{tk} \leq |T_t|, \text{ for all } t \in Z, k \in M,$$

$$(3.14c) \qquad -\sum_{i \in T_t} x_{ik} + y_{tk} \leq 0, \text{ for all } t \in Z, k \in M,$$

$$(3.15) \qquad \sum_{t \in Z} y_{tk} \leq S_k, \text{ for all } k \in M,$$

$$(3.16) \qquad y_{tk} \in \{0,1\}, \text{ for all } t \in Z, k \in M.$$

Constraints (3.14a), (3.14b) and (3.14c) are the linking constraints between the x- and y-variables. In particular, the constraints (3.14b) guarantee that variable y_{tk} is equal to 0, if all the items belonging to T_t are assigned to module k. Analogously, if none of the items of T_t is assigned to module k, the corresponding y_{tk}-variable must be set to zero. This is expressed in the constraints (3.14c).

Conversely, conditions (3.14a) make sure that the value of variable y_{tk} equals one, if a proper subset of T_t is assigned to module k.

The inequalities (3.15) model the cut capacity of the modules, i.e., the requirement that the number of wires leaving module k must not exceed the value S_k.

Our objective function,

$$\min \quad \sum_{k \in M} D_k \sum_{t \in Z} (\sum_{i \in T_t} x_{ik}) + \lambda \sum_{t \in Z} \sum_{k \in M} y_{tk}, \tag{3.17}$$

reflects an estimate for the number of pins within the modules. This is expressed by the first term of the sum. For every $t \in Z$, the value $\sum_{k \in M} y_{tk}$ amounts to the number of external pins of net t. Hence, the second term of the objective function approximates the external cost (cf. the explanation of the objective function, section 2).

With this model we associate the hypergraph $H = (N, Z)$, where every item in N is represented by a node in H. Moreover, every net in Z corresponds to a hyperedge in H and vice versa. Using this notation, every feasible solution of the above model can be interpreted as follows: For every $k \in M$, we set $P_k := \{i \in V \mid x_{ik} = 1\}$ and we define $E_k := \{t \in Z \mid y_{tk} = 1\}$. Let $M' := \{k_1, \ldots, k_{m'}\} \subseteq M$ denote the set of indices such that $P_k \neq \emptyset$. Then, $(P_{k_1}, \ldots, P_{k_{m'}})$ is a partition of the node set V into no more than m clusters such that certain knapsack constraints are satisfied. The set E_k consists of all hyperedges $t \in Z$ such that a proper subset of the nodes of t belongs to P_k. This graphtheoretic formulation of our model will be called *hypergraph clustering problem*.

Finally, we apply the transformation mentioned in the previous subsections to this model. Then, the model translates into the following optimization problem:

$$\begin{aligned}
\min \quad & \sum_{k=1}^{m} \sum_{i \in N} c_{ik} x_{ik} + \lambda \sum_{k \in M} \sum_{t \in Z} y_{tk} \\
s.t. \quad & \sum_{k \in M} x_{ik} \leq 1, i \in N \\
& \text{and } x, y \text{ satisfies } (3.12), (3.13), \\
& (3.14a), (3.14b), (3.14c), (3.15), (3.16).
\end{aligned}$$

4 Mathematical Relations between the Relaxations

With each of the three models introduced in section 3 we will associate a polyhedron whose vertices are in one to one correspondence to the feasible solutions of the model. Then, solving one of the models is equivalent to optimizing a linear objective function over the corresponding polyhedron. In order to apply linear programming techniques, we need a description of this polytope by means of equations and inequalities. Thus, a first step in solving these problems via a polyhedral approach consists in a thorough investigation of the underlying polyhedra. However, our true objective is to find good solutions for the general model introduced in section 2. This model gave rise to the three relaxations presented in section 3. If we now start investigating the polyhedra associated with the three relaxations, these studies contribute to a better understanding of the general model if and only if properties of one of these polytopes help to understand the general model or another relaxation. In other words, if, for example, valid or facet defining inequalities for the polytope associated with the first relaxation are inherited by the second polytope or at least provide ideas how to find inequalities for the polytope associated with the second relaxation, then this justifies the study of the polytope associated with the first relaxation. Clearly, the same relation should hold for the polytope associated with the second and third relaxation. This issue will be addressed in the remainder of this section. Before investigating these polytopes in more detail, let us introduce some notation that will be used troughout this section.

We denote by $e_i \in \mathbb{R}^n$ the unit vector with value one in the i-th component and zero otherwise. Let be given a subset S of some vector space. The *dimension* of S is defined as the maximum number of affinely independent elements of S minus 1. A *polyhedron* is the intersection of a finite number of half spaces. An inequality $a^T x \leq \alpha$ is *valid* with respect to a polyhedron P if $P \subseteq \{x \mid a^T x \leq \alpha\}$. The set $\{x \mid a^T x = \alpha\}$ is called the *face* of P defined by $a^T x \leq \alpha$. A *vertex* is a face of dimension 0. A bounded polyhedron is called *polytope*. Note that a polytope is uniquely described as the convex hull of its vertices. Let be given a polyhedron and an inequality $a^T x \leq \alpha$ that is valid for P. If the face defined by $a^T x \leq \alpha$ is not the empty set and if the dimension of this face is one less than the dimension of P, it is called a *facet* and $a^T x \leq \alpha$ is called *facet-defining*. Let $a^T x \leq \alpha$ be a valid inequality for a polytope P, and define $EQ(P, a^T x \leq \alpha) := \{x' \in P \mid a^T x' = \alpha\}$. In order to simplify notation, we frequently abbreviate $EQ(P, a^T x \leq \alpha)$ by $EQ(a^T x \leq \alpha)$, if there is no confusion possible. For a subset $N' \subseteq \{1, \ldots, n\}$ and a vector $(f_1, \ldots, f_n)^T$, we introduce the symbol $f(N')$ to denote the value $\sum_{i \in N'} f_i$. Finally, for the exposition of the proofs that will be given in this section we need some graphtheoretic concepts. Let $G = (V, E)$ denote a graph with *node set* V and *edge set* E. A *path* in G from $u \in V$ to $v \in V$ is an edge set $\{e_1, \ldots, e_r\}$ such that $e_i = \{u_i, u_{i+1}\}$ $i = 1, \ldots r-1$, $u_1 = u$, $u_r = v$ and $u \neq u_i, v \neq u_i \neq u_j \neq u, u_j \neq v$ for all $i, j \in \{2, \ldots, r-1\}, i \neq j$. A path is called a *cycle* if $u = v$. A graph is *connected*, if for every pair of disjoint nodes u and v there exists a path from u to v in G. Let be given a graph $G = (V, E)$ and a set $W \subseteq V$. By $E(W)$ we denote the set of all edges with both endnodes in W. For $F \subseteq E$ the symbol $V(F)$ denotes the set of all nodes in V which are incident to at least one edge in F. For hypergraphs $H = (V, E)$, the terms "connected, path, hypercycle" and the symbols $V(F)$ $(F \subseteq E)$, $E(W)$ $(W \subseteq V)$ can be defined accordingly. In particular, we say that a hypergraph is connected if for every pair of nodes there exists a sequence of nodes $u_1, \ldots, u_r$ such that u_i is incident to u_{i+1} $(i = 1, \ldots, r-1)$ and $u_1 = u$ and $u_r = v$. A *hypercycle* is a set of hyperedges $\{e_1, \ldots, e_r\}$ with the following property: there exist nodes $u_1, \ldots, u_r$ such that $\{u_i, u_{i+1}\} \subseteq e_i$ $i = 1, \ldots r-1$, $u_1 = u_r$ and $u_1 \neq u_i \neq u_j$ for all $i, j \in \{2, \ldots, r-1\}, i \neq j$.

Let be given a subset N of items and a set M of modules. Let $f = (f_1, \ldots, f_n)$ denote the list of weights for the items and assume, $F = (F_1, \ldots, F_k)$ is the list of module capacities. Moreover, let us denote the index set for the nets by Z and define $S = (S_1, \ldots, S_k)$ as the cut capacities for the modules. The graph associated with the graph clustering problem will be denoted by $G = (V, E)$. We introduce the following three polytopes associated with the three relaxations discussed in section 3.

$$\begin{aligned} MK(N \times M, f, F) := \; & \text{conv}\{x \in \mathbb{R}^{|N \times M|} \text{ s.t.} \\ & \textstyle\sum_{i \in N} f_i x_{ik} \leq F_k, \text{ for all } k \in M, \\ & \textstyle\sum_{k \in M} x_{ik} \leq 1, \text{ for all } i \in N, \\ & x_{ik} \in \{0,1\}, \text{ for all } i \in N, k \in M\}. \end{aligned}$$

$$\begin{aligned} C(N \times M, f, F, E) := \; & \text{conv}\{(x^T, \eta^T)^T, x \in \mathbb{R}^{|N \times M|}, \eta \in \mathbb{R}^E \text{ s.t.} \\ & \textstyle\sum_{i \in N} f_i x_{ik} \leq F_k, \text{ for all } k \in M, \\ & \textstyle\sum_{k \in M} x_{ik} \leq 1, \text{ for all } i \in N, \\ & x_{ik} + x_{jl} - \eta_{ij} \leq 1, \text{ for all } ij \in E, k, l \in M, k \neq l, \\ & x_{ik} + x_{jk} + \eta_{ij} \leq 2, \text{ for all } ij \in E, k \in M, \\ & x_{ik} \in \{0,1\}, \text{ for all } i \in N, k \in M, \\ & \eta_{ij} \in \{0,1\}, \text{ for all } ij \in E\}. \end{aligned}$$

$$
\begin{aligned}
HC(N\times M,f,F,Z,S) := \; & \operatorname{conv}\{(x^T,y^T)^T, x\in \mathbb{R}^{|N\times M|}, y\in \mathbb{R}^{|\mathcal{Z}\times M|} \text{ s.t.} \\
& \textstyle\sum_{i\in N} f_i x_{ik} \le F_k, \text{ for all } k\in M, \\
& \textstyle\sum_{k\in M} x_{ik} \le 1, \text{ for all } i\in N, \\
& x_{ik}+x_{jl}-y_{tk} \le 1, \text{ for all } t\in Z, i,j\in T_t, i\ne j, \\
& k,l\in M, k\ne l, \\
& \textstyle\sum_{i\in T_t} x_{ik}+y_{tk} \le |T_t| \text{ for all } t\in Z, k\in M, \\
& -\textstyle\sum_{i\in T_t} x_{ik}+y_{tk} \le 0 \text{ for all } t\in Z, k\in M, \\
& \textstyle\sum_{t\in Z} y_{tk} \le S_k, \text{ for all } k\in M, \\
& x_{ik}\in\{0,1\}, \text{ for all } i\in N, k\in M, \\
& y_{tk}\in\{0,1\}, \text{ for all } t\in Z, k\in M\}.
\end{aligned}
$$

$MK(N\times M, f, F)$ is the polytope associated with the multiple knapsack problem. The polytopes $C(N\times M,f,F,E)$ and $HC(N\times M,f,F,Z,S)$ correspond to the graph - and hypergraph clustering models, respectively, that were introduced in section 3.

For our line of investigation, the polytope parameter that has to be determined first is its dimension.

Here, it is easy to see that the dimension of $MK(N\times M,f,F)$ equals $|N||M|$, if and only if $f_i\le F_k$ for all $i\in N, k\in M$, since the $|N||M|+1$ affinely independent vectors 0 and $e_{ik}, i\in N, k\in M$ are elements of the polytope if and only if $f_i\le F_k$ for all $i\in N,k\in M$. Similarly, $C(N\times M,f,F,E)$ has dimension $|N||M|+|E|$, if and only if $f_i\le F_k$ for all $i\in N,k\in M$. Finally, one can convince oneself that the dimension of the polytope $HC(N\times M,f,F,Z,S)$ equals $|N||M|+|\mathcal{Z}||M|$ if and only if $f_i\le F_k$ for all $i\in N,k\in M$.

In the following we assume w.l.o.g. that $f_i\le F_k$ holds for all $i\in N,k\in M$.

The subsequent theorem states that every valid or facet-defining inequality for the polytope $MK(N\times M,f,F)$ is valid or facet-defining for $C(N\times M,f,F,E)$.

Theorem 4.1 *Let $a^Tx\le\beta$ be a valid inequality for the multiple knapsack polytope $MK(N\times M,f,F)$ and define the vector a' via $a'_{ik}=a_{ik}$ for all $i\in N,k\in M$ and $a'_{ij}=0$ for all $ij\in E$. Then, $a'^T(x^T,\eta^T)^T\le\beta$ is valid for the polytope $C(N\times M,f,F,E)$. In case, $a^Tx\le\beta$ defines a facet of $MK(N\times M,f,F)$ and is not a multiple of the inequalities $\sum_{k\in M}x_{ik}\le 1, i\in N$ or $x_{ik}\le 1, i\in N,k\in M$, the inequality $a'^T(x^T,\eta^T)^T\le\beta$ defines a facet of $C(N\times M,f,F,E)$.*

Proof. By definition, every inequality valid for $MK(N \times M, f, F)$ is valid for $C(N \times M, f, F, E)$ as well.

Now suppose, $EQ(C(N \times M, f, F, E), a'^T\ (x^T, \eta^T)^T \leq \beta) \subseteq EQ(C(N \times M, f, F, E), \hat{a}^T x + \hat{b}^T \eta \leq \hat{\beta})$ such that $\hat{a}^T x + \hat{b}^T \eta \leq \hat{\beta}$ defines a facet of $C(N \times M, f, F, E)$.

Let be given $rs \in E$ and suppose, $x^0 \in EQ(MK(N \times M, f, F), a^T x \leq \beta)$ is a vector such that $x^0_{rk} = 0$ for all $k \in M$. The vector x^0 exists, since otherwise $EQ(MK(N \times M, f, F), a^T x \leq \beta) \subseteq EQ(MK(N \times M, f, F), \sum_{k \in M} x_{rk} \leq 1)$, which contradicts the assumption.

We define $\eta(rs)$ via :

$$\eta(rs)_{ij} = \begin{cases} 0, & \text{if } ij = rs \text{ or } ij \in E \text{ and} \\ & i \text{ and } j \text{ are assigned to the same module,} \\ 1, & \text{otherwise.} \end{cases}$$

Then, the vectors $(x^0, \eta(rs))$ and $(x^0, \eta(rs) + e_{rs})$ belong to the face $EQ(C\ (N \times M, f, F, E), \hat{a}^T x + \hat{b}^T \eta \leq \hat{\beta})$. Thus, they also satisfy the equation $\hat{a}^T x + \hat{b}^T \eta = \hat{\beta}$. This yields $\hat{b}_{rs} = 0$ for all $rs \in E$.

Finally, let $x \in \mathbb{R}^{|N \times M|}$ be an element of the face $EQ(MK(N \times M, f, F),\ a^T x \leq \beta)$. We define $\eta' \in \mathbb{R}^E$ via:

$$\eta'_{ij} = \begin{cases} 0, & \text{if } ij \in E \text{ and} \\ & i \text{ and } j \text{ are assigned to the same module,} \\ 1, & \text{otherwise.} \end{cases}$$

Then, the vector (x, η') is valid and satisfies the equations $a'^T x = a^T x = \beta$ and thus $\hat{a}^T x + \hat{b}^T \eta' = \hat{\beta}$. Hence, every vector $x \in EQ(MK(N \times M, f, F), a'^T x + b'^T \eta \leq \beta)$ yields a vector $(x, \eta') \in EQ(C(N \times M, f, F, E), \hat{a}^T x + \hat{b}^T \eta \leq \hat{\beta})$. Since $\hat{b}_{ij} = 0$ for all $if \in E$ and since $a^T x \leq \beta$ defines a facet, we conclude that $\hat{a}_{ik} = a'_{ik} = a_{ik}$ for all $i \in N,\ k \in M$. This proves the statement. ■

Example: Let $S \subseteq N$ be a subset of items such that $f(S) > F_k$ for some $k \in M$ and $f(S \setminus \{s\}) \leq F_k$ for every $s \in S$. The inequality

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

is valid for the polytope $MK(N \times M, f, F)$ and defines a facet for $MK(S \times M, f, F)$. A subset S with the above properties is called a *minimal cover* with respect to module k. The corresponding inequality is called *minimal cover inequality* and was discussed in [B75], [HJP75], [W75]. By applying the above theorem we can conclude that the minimal cover inequality defines a facet of the polytope $C(S \times M, f, F, E)$. ■

For other classes of inequalities that are valid or facet-defining for $MK(N \times M, f, F)$ we refer the reader to [P80], [GR90a], [GR90b] and [FMW92].

As mentioned before, the above theorem serves as a theoretical justification for the study of the polytope $MK(N \times M, f, F)$ before investigating the more complex polytope $C(N \times M, f, F, E)$, since inequalities for the first one are inherited by the second one.

Unfortunately, the same relation does not hold between the polytope $MK(N \times M, f, F)$ and $HC(N \times M, f, F, Z, S)$. More precisely, given an inequality $a^T x \leq \alpha$ that defines a facet for $MK(N \times M, f, F)$, then $a^T x \leq \alpha$ does not necessarily define a facet for the polytope $HC(N \times M, f, F, Z, S)$. An example of this kind is shown below.

Example: Let be given an instance of the general model with $M = \{1, 2\}$, $N = \{1, 2, 3, 4, 5, 6\}$, $\mathcal{Z} = \{T_1\}$ with $T_1 = \{1, 2, 3, 4, 5, 6\}$. We set $F_1 = F_2 = 12$ and $f_1 = 2, f_2 = 3, f_3 = 4, f_4 = 4, f_5 = 5, f_6 = 7$. Under these assumptions, the set $S = \{1, 2, 3, 4\}$ is a minimal cover. The set $T = \{5, 6\}$ satisfies $f(T) \leq F_2$ and for every $i \in S$, the set $T \cup \{i\}$ is a minimal cover. Finally, the set $T \setminus \{5\} \cup \{1, 2\}$ satisfies $f(T \setminus \{5\} \cup \{1, 2\}) = 12 \leq F_2$. The inequality

$$\sum_{i \in S} x_{i1} + \sum_{j \in S \cup T} x_{j2} \leq |S| + |T| - 1$$

is called *extendend cover inequality*. It is shown in [FMW92] that it defines a facet for the polytope $MK(S \times M \cup T \times \{2\}, f, F)$.

However, it can easily be checked that every vector $(x, y) \in HC(N \times M, f, F, Z, S)$ such that $\sum_{i \in S} x_{i1} + \sum_{j \in S \cup T} x_{j2} = |S| + |T| - 1$ holds, also satisfies $y_{1,1} = y_{1,2} = 1$. This proves that this particular extended cover inequality does not define a facet for the polytope $HC(S \times M \cup T \times \{2\}, f, F, Z, S)$. ■

This example shows that facets for the polytope $C(N \times M, f, F, E)$ are not necessarily inherited by the polytope $HC(N \times M, f, F, Z, S)$. Yet, the following example demonstrates that the "logic" that characterizes a particular class of facet-defining inequalities for $C(N \times M, f, F, E)$ can be used to obtain facets for $HC(N \times M, f, F, Z, S)$.

Theorem 4.2 *Let be given an instance of the general model, i.e., a set N of items, a set M of modules, a net list $\mathcal{Z}$, item weights f, module capacities F and* *a list of cut capacities S. With this instance we associate the graph $G = (V, E)$ as described before. Suppose $|M| \geq 2$ and $F_k = F_l$ for all $k, l \in M, k \neq l$. Let W denote a cycle in $G = (V, E)$ and assume, $V(W)$ defines a minimal cover with respect to some $k \in M$ (note that the capacities are all equal, i.e., $V(W)$ is a minimal cover with respect to every module). Then, the inequality*

$$2 \sum_{k \in M} \sum_{i \in V(W)} x_{ik} - \sum_{ij \in W} y_{ij} \leq 2(|V(W)| - 1)$$

is called cycle inequality. It defines a facet for the polytope $C(V(W) \times M, f, F, W)$.

∎

A paper containing the proof of this statement is in preparation.

Now, let us use the "logic" that characterizes the cycle inequality and create a facet-defining inequality for the appropriate hypergraph clustering polytope.

Theorem 4.3 *Again, let be given an instance of the general model, i.e., a set N of items, a set M of modules, a net list $\mathcal{Z}$, item weights f, module capacities F and a list of cut capacities S. Suppose $|M| \geq 2$ and $F_k = F_l$ for all $k, l \in M, k \neq l$. With this instance we associate the hypergraph $H = (N, Z)$, where every item in N is represented by a node in H and a net $t \in Z$ corresponds to a hyperedge in $H = (N, Z)$. Let W denote a hypercycle in $H = (N, Z)$ which satisfies the additional requirements: no hyperedge of W is a subset of another; every node of $N(W)$ is contained in exactly two hyperedges and every hyperedge has a nonempty intersection with exactly two other hyperedges. Moreover, assume that $N(W)$ defines a minimal cover with respect to every $k \in M$. Then, for every $k \in M$ the inequality*

$$2 \sum_{l \in M} \sum_{i \in N(W)} x_{il} - \sum_{l \in M \setminus \{k\}} \sum_{t \in W} y_{tl} \leq 2(|N(W)| - 1)$$

defines a facet for the polytope $HC(N(W) \times M, f, F, W, S)$.

∎

A proof of theorem 4.3 will also appear in a subsequent paper.

This example shows, indeed, that there is a strong relation between facet defining inequalties for these two polytopes.

As a final statement, let us remark that it is also true that valid inequalities for $C(N \times M, f, F, E)$ translate into valid inequalities for $HC(N \times M, f, F, Z, S)$ in the following sense.

Theorem 4.4 *Let $a^T x + b^T \eta \leq \alpha$ be a valid inequality for the polytope $C(N \times M, f, F, E)$ such that $a_{ik} \geq 0$ for all $i \in N, k \in M$ and $b_{ij} \leq 0$ for all $ij \in E$.*

We define coefficients $\hat{b}_{tk} := \frac{\sum_{i,j \in T_t, i \neq j} b_{ij}}{2}$ for $t \in Z$ and $k \in M$.

Then, the inequality $a^T x + \hat{b}^T y \leq \alpha$ is valid for the polytope $HC(N \times M, f, F, Z, S)$.

Proof.

For every $ij \in E$ we define $Z_{ij} = \{t \in Z \mid \{i,j\} \subseteq T_t\}$, i.e. Z_{ij} is the set of all nets T_t such that items i and j are simultaneously contained in T_t.

Let (x, y) be any element of $HC(N \times M, f, F, Z, S)$. With the vector $x \in \mathbb{R}^{|N||M|}$ we associate a feasible solution (x, η) of the polytope $C(N \times M, f, F, E)$ as follows:

For every $ij \in E$ we set

$$\eta_{ij} = \begin{cases} 1, & \text{if there exist } k, l \in M, k \neq l \text{ such that} \\ & x_{ik} = 1 \text{ and } x_{jl} = 1, \\ 0, & \text{otherwise.} \end{cases}$$

Clearly, $(x, \eta) \in C(N \times M, f, F, E)$, which implies that $a^T x + b^T \eta \leq \alpha$ holds.

Moreover, let be given $ij \in E$ with $\eta_{ij} = 1$. Then, we can conclude that for every $t \in Z_{ij}$ there exist $k, l \in M, k \neq l$ such that $y_{tk} = y_{tl} = 1$. Set $Y := \{t \in Z \mid \text{ there exist } k, l \in M, k \neq l \text{ and } i, j \in T_t, i \neq j \text{ such that } x_{ik} = x_{jl} = 1\}$ and $Y^2 := \{ij \in E \mid \text{ there exist } k, l \in M, k \neq l \text{ such that } x_{ik} = x_{jl} = 1\}$.

Taking all together, we obtain

$$\begin{aligned}
& a^T x + \hat{b}^T y = \\
& a^T x + \sum_{t \in Z} \sum_{k \in M} \hat{b}_{tk} y_{tk} = \\
& a^T x + \sum_{t \in Z} \sum_{k \in M} \frac{\sum_{i,j \in T_t, i \neq j} b_{ij}}{2} y_{tk} \leq \\
& a^T x + \sum_{t \in Y} \sum_{i,j \in T_t, i \neq j} b_{ij} \leq \\
& a^T x + \sum_{ij \in Y^2} b_{ij} = \\
& a^T x + \sum_{ij \in E} b_{ij} \eta_{ij} = a^T x + b^T \eta.
\end{aligned}$$

This implies that the inequality $a^T x + \hat{b}^T y \leq \alpha$ is valid for the polytope $HC(N \times M, f, F, Z, S)$, which completes the proof. ■

These theorems support our opinion that it is a reasonable idea to study the polytope $C(N \times M, f, F, E)$ before investigating $HC(N \times M, f, F, Z, S)$, since inequalities for $HC(N \times M, f, F, Z, S)$ are much more complex than the corresponding ones for $C(N \times M, f, F, E)$. Second, there is still a close relationship beween the two polytopes (see the example above), and experiences made with $C(N \times M, f, F, E)$ may be very important on the way towards a solution of the real instances associated with the polytope $HC(N \times M, f, F, Z, S)$.

Finally, we mention a few relationships between the model associated with the polytope $HC(N \times M, f, F, Z, S)$ and the general model introduced in section 2.

The hypergraph clustering model considers the exact formulation of the capacity constraints concerning the items as well as the modules. Thus, by applying the preprocessing phase of section 2 the most important side constraints of the general model are taken into account. The only difference between the two models is that the hypergraph clustering relaxation neglects the net constraints, and it uses a simplified objective function which provides just a heuristic estimate of what should really be minimized. However, if one is able to handle the polytope $HC(N \times M, f, F, Z, S)$ from a theoretical as well as from a practical point of view, one could start with some objective function and optimize over the polytope $HC(N \times M, f, F, Z, S)$. If the solution is feasible for the general model, we stop. Otherwise, we modify the estimate of the objective function in a Lagrangian fashion and repeat this process until we terminate with a globally feasible solution. Surely, the solution provided that way is not necessarily optimal for the original problem. However, the objective function is somehow related to the original one. Thus, an optimal solution to the hypergraph clustering model that is still feasible for the original one should be not too bad for practical purposes.

Our mathematical model of the module design problem for main frame computers introduced in section 2, although already a simplification of the real task, seems to be way beyond our present algorithmic and computational capabilities. The parameters of the practical instances of our application are in the following ranges:

number of items: $[250, 2000]$,
number of modules: $[4, 30]$,
number of nets: $[5000, 100000]$.

Taking largest sizes, this leads to an integer programming formulation for the

Module Design Problem involving more than 12.000.000 variables. In order to generate primal and dual information, in particular, reasonable solutions, we derived the three (somewhat simpler) combinatorial relaxations explained in Section 3 and showed that they are intimitely related to the original problem in a precise polyhedral sense. To pave the way for the solution of the real problem we are presently investigating the three relaxations from both the theoretical and algorithmic point of view. We will report about our results in a series of forthcoming papers that, hopefully, will help solve the original module design problem for main frame computers in a satisfactory way.

References

[B75] E. Balas, "Facets of the Knapsack Polytope", *Mathematical Programming* 8, 146 - 164 (1975).

[FMW93] C. E. Ferreira, A. Martin and R. Weismantel, "Facets for the Multiple Knapsack Problem", Konrad-Zuse-Zentrum für Informationstechnik Berlin, Preprint SC 93-4 (1993).

[GR90a] E. S. Gottlieb e M. R. Rao, "The Generalized Assignment Problem: Valid Inequalities and Facets", *Mathematical Programming* 46, 31 - 52 (1990).

[GR90b] E. S. Gottlieb e M. R. Rao, "(1,k)-Configuration Facets for the Generalized Assignment Problem", *Mathematical Programming* 46, 53 - 60 (1990).

[HJP75] P. L. Hammer, E. L. Johnson and U. N. Peled, "Facets of Regular 0-1 Polytopes", *Mathematical Programming* 8, 179 - 206 (1975).

[MT91] S. Martello and P. Toth, *Knapsack Problems - Algorithms and Computer Implementations*, John Wiley and Sons, New York (1991).

[P80] M. W. Padberg, "(1,k)-Configurations and Facets for Packing Problems", *Mathematical Programming* 18, 94 - 99 (1980).

[W75] L. A. Wolsey, "Faces of Linear Inequalities in 0-1 Variables", *Mathematical Programming* 8, 165 - 178 (1975).