Intermediate integer programming representations using value disjunctions*

Matthias Köppe Quentin Louveaux Robert Weismantel

Date: 2006/09/25 15:21:56 – Revision: 3.146

Abstract

We introduce a general technique to create an extended formulation of a mixed-integer program. We classify the integer variables into blocks, each of which generates a finite set of vector values. The extended formulation is constructed by creating a new binary variable for each generated value. Initial experiments show that the extended formulation can have a more compact complete description than the original formulation.

We prove that, using this reformulation technique, the facet description decomposes into one "linking polyhedron" per block and the "aggregated polyhedron". Each of these polyhedra can be analyzed separately. For the case of identical coefficients in a block, we provide a complete description of the linking polyhedron and a polynomial-time separation algorithm. Applied to the knapsack with a fixed number of distinct coefficients, this theorem provides a complete description in an extended space with a polynomial number of variables.

Based on this theory, we propose a new branching scheme that analyzes the problem structure. It is designed to be applied in those subproblems of hard integer programs where LP-based techniques do not provide good branching decisions. Preliminary computational experiments show that it is successful for some benchmark problems of multi-knapsack type.

1 Introduction

Extreme representations of the feasible points of a mixed-integer linear optimization problem are either given by means of the facet defining inequalities in the original space or by a set of feasible mixed-integer points whose convex hull contains the feasible region. It is well known that in principle one such extreme representation can be transformed into the other extreme representation. However from an algorithmic point of view both extreme representations are very hard to achieve.

This suggests to search for other, "intermediate" representations that are algorithmically more tractable, in the sense that they—

*Address: Otto-von-Guericke-Universität Magdeburg, Department of Mathematics/IMO, Universitätsplatz 2, 39106 Magdeburg, Germany.
E-mail addresses: {mkoeppe, louveaux, weismant}@imo.math.uni-magdeburg.de
The authors gratefully acknowledge support from the European TMR network ADONET 504438.

- require less variables than the extreme representation by the vertices,
- require less constraints compared to the total number of facets of the convex hull,
- have a simpler combinatorial constraint structure than the facets of the convex hull in the original space and hence, the separation problem in the extended space is easier to solve.

Intermediate representations of the feasible region are complete descriptions of an extended formulation of the original problem. To make this notion precise, we define:

Definition 1 (Representation by projection)**.** Let $K \subseteq \mathbf{R}^n$, $\bar{K} \subseteq \mathbf{R}^d$ be two rational polyhedra and $B \in \mathbf{Q}^{n\times d}$ a rational matrix. We call $\bar{\mathcal{F}} = \bar{K} \cap \mathbf{Z}^d$ a *representation* of $\mathcal{F} = K \cap \mathbf{Z}^n$ if the following two properties hold:

(a) $K \cap \mathbf{Z}^n = \{\, \mathbf{x} \in \mathbf{Z}^n : \mathbf{x} = B\mathbf{y},\ \mathbf{y} \in \bar{K} \cap \mathbf{Z}^d \,\}$.

(b) $\operatorname{conv}(K \cap \mathbf{Z}^n) = \{\, \mathbf{x} \in \mathbf{R}^n : \mathbf{x} = B\mathbf{y},\ \mathbf{y} \in \bar{K} \,\}$.

Such a representation is called *extreme* if either $d = n$ and $B = I$ or if $\bar{K} = \{\, \mathbf{y} \in \mathbf{R}^d_+ : \sum_{i=1}^d y_i = 1 \,\}$; otherwise, it is called *intermediate*.

We remark that R. K. Martin [17] calls the sets $\bar{\mathcal{F}}$ and $\mathcal{F}$ "strongly equivalent" in this situation.

In the literature, there are several interesting examples of this type. Chopra and Rao [7, 8] introduced a directed formulation for the Steiner tree problem and showed that exponentially many inequalities in the undirected formulation are projections of a small number of directed inequalities. R. K. Martin [18] reports on the minimum spanning tree problem, which has as an inequality formulation of size $\mathrm{O}(2^n)$. It can, however, alternatively be described as the projection of an extended formulation which requires $\mathrm{O}(n^3)$ variables and $\mathrm{O}(n^2)$ constraints. Moreover, there are many further compact extended formulations for specific combinatorial optimization problems, in particular for lot-sizing and fixed-charge network problems; see, for instance, [13, 19, 17].

Next we illustrate on an example that also quite general problems such as knapsack problems can sometimes be described in an extended space such that the higher dimensional polyhedron is much more appealing than the original facet description.

Example 2. Consider the set of $\mathbf{x} \in \{0, 1\}^8$ such that

$$8x_0 - x_1 - 2x_2 - 3x_3 - 4x_4 - 5x_5 - 6x_6 - 7x_7 \leq 0. \tag{1}$$

The convex hull of solutions to this knapsack problems is given by the following system of thirteen inequalities:

$$
\begin{array}{llllllll}
x_0 & & - x_3 & & - x_5 & - x_6 & - x_7 & \leq 0 \\
x_0 & & & - x_4 & - x_5 & - x_6 & - x_7 & \leq 0 \\
x_0 - x_1 - x_2 & & & & - x_5 & - x_6 & - x_7 & \leq 0 \\
x_0 - x_1 & & - x_3 & - x_4 & & - x_6 & - x_7 & \leq 0 \\
x_0 & - x_2 & - x_3 & - x_4 & - x_5 & & - x_7 & \leq 0 \\
x_0 & - x_2 & - x_3 & - x_4 & & - x_6 & - x_7 & \leq 0 \\
x_0 - x_1 - x_2 & & - x_3 & - x_4 & - x_5 & - x_6 & & \leq 0 \\
\mathbf{2}x_0 - x_1 - x_2 & & - x_3 & - x_4 & - x_5 & - x_6 & - x_7 & \leq 0 \\
\mathbf{2}x_0 & - x_2 & - x_3 & - x_4 & - x_5 & - x_6 & - \mathbf{2}x_7 & \leq 0 \\
\mathbf{2}x_0 - x_1 & & - x_3 & - x_4 & - x_5 & - \mathbf{2}x_6 & - \mathbf{2}x_7 & \leq 0 \\
\mathbf{3}x_0 - x_1 - x_2 & & - x_3 & - x_4 & - \mathbf{2}x_5 & - \mathbf{2}x_6 & - \mathbf{2}x_7 & \leq 0 \\
\mathbf{3}x_0 - x_1 - x_2 & & - \mathbf{2}x_3 & - \mathbf{2}x_4 & - x_5 & - \mathbf{2}x_6 & - \mathbf{2}x_7 & \leq 0 \\
\mathbf{5}x_0 - x_1 - x_2 & & - \mathbf{2}x_3 & - \mathbf{2}x_4 & - \mathbf{3}x_5 & - \mathbf{4}x_6 & - \mathbf{4}x_7 & \leq 0
\end{array}
$$

One way to obtain an extended formulation for (1) is to replace variables x_1, x_2, x_3, and x_4 by enumerating all the possible subsets of $\{1,2\}$ and $\{3,4\}$, respectively, and introducing variables for all these subsets. This yields the following reformulation:

$$
\begin{array}{l}
8x_0 - x_{\{1\}} - 2x_{\{2\}} - 3x_{\{3\}} - 4x_{\{4\}} - 5x_5 - 6x_6 - 7x_7 - 3x_{\{1,2\}} - 7x_{\{3,4\}} \leq 0 \\
x_{\{1\}} + x_{\{2\}} + x_{\{1,2\}} \leq 1 \\
x_{\{3\}} + x_{\{4\}} + x_{\{3,4\}} \leq 1
\end{array}
$$

(Here the empty set as a subset of $\{1,2\}$ and $\{3,4\}$ appears as the slack variable of the two packing constraints. The original variables relate to the new variables as follows: $x_1 = x_{\{1\}} + x_{\{1,2\}}$, etc.)

The convex hull of all feasible binary solutions to this system is given by the following list of nine inequalities:

$$
\begin{array}{l}
x_0 - x_5 - x_6 - x_7 - x_{\{3,4\}} \leq 0 \\
x_0 - x_{\{1\}} - x_{\{2\}} - x_5 - x_6 - x_7 - x_{\{1,2\}} \leq 0 \\
x_0 - x_{\{3\}} - x_{\{4\}} - x_6 - x_7 - x_{\{1,2\}} - x_{\{3,4\}} \leq 0 \\
x_0 - x_{\{2\}} - x_{\{3\}} - x_{\{4\}} - x_5 - x_7 - x_{\{1,2\}} - x_{\{3,4\}} \leq 0 \\
x_0 - x_{\{1\}} - x_{\{2\}} - x_{\{3\}} - x_{\{4\}} - x_5 - x_6 - x_{\{1,2\}} - x_{\{3,4\}} \leq 0 \\
\mathbf{2}x_0 - x_{\{1\}} - x_{\{2\}} - x_{\{3\}} - x_{\{4\}} - x_5 - x_6 - x_7 - x_{\{1,2\}} - x_{\{3,4\}} \leq 0 \\
\mathbf{2}x_0 - x_{\{2\}} - x_{\{3\}} - x_{\{4\}} - x_5 - x_6 - \mathbf{2}x_7 - x_{\{1,2\}} - \mathbf{2}x_{\{3,4\}} \leq 0 \\
+ x_{\{3\}} + x_{\{4\}} + x_{\{3,4\}} \leq 1 \\
x_{\{1\}} + x_{\{2\}} + x_{\{1,2\}} \leq 1
\end{array}
$$

Note that not only the number of inequalities for the extended formulation is smaller than in the original space. More importantly, the structure of the inequalities in the extended space is significantly nicer when compared to the structure of the inequalities in the original space. For instance, the maximum coefficient occuring in the inequalities in the higher dimensional space is 2, whereas the highest coefficient in the inequalities in the original space is already 5.

In the example, the extended formulation was constructed by introducing a new variable for each of the possible subsets of a set of original variables. An alternative interpretation of the above construction is the following. The new variable $x_{\{1,2\}}$ is the *product* of the original variables x_1 and x_2; likewise, the new variable $x_{\{1\}}$ is the product of the original variable x_1 and the complementary variable $\bar{x}_2 = 1 - x_2$. Non-linear constructions of this type are a feature of the so-called Lift-and-Project approach. This approach has its roots in the work of Egon Balas on disjunctive optimization [3, 4]. It was further refined by several authors in [20, 16, 5, 6]; see also [14]. The various Lift-and-Project approaches usually define hierarchies $K \supseteq K^1 \supseteq K^2 \supseteq \cdots \supseteq P$ of continuous relaxations, starting at a linear relaxation K of $\mathcal{F}$, that reach the convex hull $P = \operatorname{conv} \mathcal{F}$ in a finite number of steps. The hierarchies of relaxations have strong properties and a beautiful mathematical theory behind them. A common feature of the Lift-and-Project approaches is that they consider an extended formulation (by "lifting" the problem description into a higher-dimensional space using a non-linear operator), which is then projected down into the original variable space. In each of the approaches, the individual extended formulations constructed are of polynomial size. For instance, the Lovasz–Schrijver method [16] uses a sequence of lifting-and-project steps into dimension $\mathrm{O}(n^2)$. The Sherali–Adams method [20], on the other hand, does only one very strong lift-and-project step, using the so-called "level-t operator" which embeds the original variable space into a space of dimension $\mathrm{O}(n^{t+1})$; when the level t is considered as a fixed number, the formulation is again of polynomial size.

Even though both the extended formulations constructed for the individual problems cited above and the extended formulations arising in the various lift-and-project approaches are of polynomial size, they are usually *too large* for writing them down explicitly. In the case of structured problems with too large an extended formulation, one usually applies column and row generation techniques (branch-cut-and-price methods) to solve the problems. In the case of the lift-and-project approaches, the extended formulation is not written down explicitly and only used as a device for constructing stronger bounds or for constructing strong valid inequalities for the original formulation.

The method of this paper. The tool that we propose in this paper to generate an extended formulation is the *value-disjunction procedure*. Again the idea is to introduce new variables corresponding to certain subsets of original variables. Like the Sherali–Adams method, it also applies to subsets of original variables of arbitrary cardinality. It offers a lot of freedom in generating the extended formulation; however, since it cannot provide the strong linking constraints of the Sherali–Adams method, it is also much weaker. It is a general way to produce intermediate representations for mixed-integer optimization problems.

The goal of our method is to compute an intermediate formulation that is *practically small enough* to be written down explicitly. We do not prove theorems on the polynomiality of our reformulations. Indeed, in the general case, the extended formulations produced by the value-disjunction procedure have exponentially many variables. However, we propose to use *heuristics* for constructing *simplifying relaxations* of the problem at hand: If the relaxation is chosen simple enough, we can always introduce an extended formulation that keeps the number of new variables linear in the size of the subset of the original variables.

Outline. We introduce the value-disjunction procedure in Section 2. We then describe the convex hull of the given mixed-integer set as the intersection of several simpler polyhedra using the variables of the extended space. This is the *structure theorem* for the value-disjunction procedure. In Section 3 we introduce the family of linking polyhedra. In the special but important case that such a linking polyhedron comes from the unweighted sum of a set of variables, we completely describe the polyhedron by means of linear inequalities and equations. As an application of the structure theorem in Section 2 together with the polyhedral characterizations of Section 3, we are able to determine an explicit description of the convex hull of all solutions to a 0/1 knapsack problem with only a fixed number of different weights. This is the topic of Section 4.

Finally, in Section 5, we investigate one way of making computational use of value disjunctions: By branching also on the new binary variables of the extended formulation instead of only on the original variables, it is possible to take more flexible branching decisions. In fact, we propose such a branching scheme for situations where none of the usual LP-based variable selection criteria provides a solid basis for taking a branching decision. Such situations frequently occur in very hard integer programs like the market-split instances [10]. We investigate the effect of branching simplifying the facet description: A branching decision is considered good if the facet descriptions of the generated subproblems are significantly simpler than the original facet description. Using experiments with randomly generated problem instances, we show that it is possible to make a branching decision based on the structure of the problem which is better than branching on the original variables. Finally we report on simple computational experiments with a few hard integer programs, where we branch explicitly on the new binary variables and then solve the subproblems with the branch-and-cut system CPLEX. We obtain a significant reduction in both the number of nodes and the computation time.

2 Value disjunctions

In this section, we present a structural result about an extended formulation of a given mixed-integer programming model. To this end, consider a bounded mixed-integer set of the form

$$\mathcal{F} = \left\{ (\mathbf{x}, \mathbf{w}) \in \mathbf{Z}_+^n \times \mathbf{R}_+^d : \sum_{j=1}^{n} A_j x_j + \sum_{j=1}^{d} G_j w_j \leq \mathbf{b},\ \mathbf{x} \leq \mathbf{u} \right\},$$

where A_j, $G_j \in \mathbf{R}^m$ for all j, $\mathbf{b} \in \mathbf{R}^m$, and $\mathbf{u} \in \mathbf{Z}_+^n$. We set $P = \operatorname{conv} \mathcal{F}$.

Let us partition the set $N = \{1, \dots, n\}$ into subsets $N_1, \dots, N_K$. For each of the sets N_i, we determine all the possible vectors ("values") generated by the columns A_j belonging to the variables indexed by N_i:

$$\mathcal{A}_i = \left\{ \sum_{j \in N_i} A_j x_j : x_j \in \{0, \dots, u_j\} \text{ for } j \in N_i \right\}.$$

Since the integer variables are assumed to be bounded, the set $\mathcal{A}_i$ is finite; its cardinality $n_i = |\mathcal{A}_i|$ is at most $\prod_{j \in N_i}(1 + u_j)$. Let the elements of $\mathcal{A}_i$ be numbered, $\mathcal{A}_i = \{\mathbf{f}_1^{N_i}, \dots, \mathbf{f}_{n_i}^{N_i}\}$. We shall associate with $\mathbf{f}_k^{N_i}$ a new binary variable $y_k^{N_i}$. In order to simplify the subsequent expositions, we shall also use the abbreviating notations $A(\mathbf{x}^{N_i}) = \sum_{j \in N_i} A_j$, and moreover $A(\mathbf{y}^{N_i}) = \sum_{k=1}^{n_i} y_k^{N_i} \mathbf{f}_k^{N_i}$ and $A(\mathbf{y}) = \sum_{i=1}^{K} A(\mathbf{y}^{N_i})$.

We come to two major definitions that we make use of in this paper.

Definition 3. For a given subset N_i, we define the *linking polyhedron* as

$$V_i = \operatorname{conv}\left\{ (\mathbf{x}^{N_i}, \mathbf{y}^{N_i}) \in \mathbf{Z}_+^{|N_i|} \times \{0,1\}^{n_i} : \begin{array}{l} \sum_{j \in N_i} A_j x_j = \sum_{k=1}^{n_i} \mathbf{f}_k^{N_i} y_k^{N_i} \\ \sum_{k=1}^{n_i} y_k^{N_i} = 1 \\ 0 \le x_j \le u_j,\ j = 1, \dots, n \end{array} \right\}. \quad (2)$$

Furthermore we define the *aggregated polyhedron* as

$$Q = \operatorname{conv}\left\{ (\mathbf{y}, \mathbf{w}) \in \{0,1\}^{n_1 + \cdots + n_K} \times \mathbf{R}_+^d : \begin{array}{l} \sum_{i=1}^{K} \sum_{k=1}^{n_i} \mathbf{f}_k^{N_i} y_k^{N_i} + \sum_{j=1}^{d} G_j w_j \le \mathbf{b} \\ \sum_{k=1}^{n_i} y_k^{N_i} = 1 \text{ for all } i = 1, \dots, K \end{array} \right\}. \quad (3)$$

Thus, for every value $\mathbf{f}_k^{N_i}$ in a set $\mathcal{A}_i$ we are introducing a new binary variable $y_k^{N_i}$. With this family of new variables, we can obtain a new, extended formulation of $\mathcal{F}$ by linking the original variables x_j with the new "value variables" $y_k^{N_i}$ as follows.

Definition 4. We define the *value-disjunction reformulation* $\mathcal{F}$ as the extended formulation

$$\bar{\mathcal{F}} = \left\{ (\mathbf{x}, \mathbf{w}, \mathbf{y}) \in \mathbf{Z}_+^n \times \mathbf{R}_+^d \times \{0,1\}^{n_1 + \cdots + n_K} : \begin{array}{l} \sum_{j=1}^{n} A_j x_j + \sum_{j=1}^{d} G_j w_j \le \mathbf{b}, \quad \mathbf{x} \le \mathbf{u} \\ \sum_{j \in N_i} A_j x_j = \sum_{k=1}^{n_i} \mathbf{f}_k^{N_i} y_k^{N_i} \quad \text{for all } i = 1, \dots, K \\ \sum_{k=1}^{n_i} y_k^{N_i} = 1 \quad \text{for all } i = 1, \dots, K \end{array} \right\},$$

and $\bar{P} = \operatorname{conv} \bar{\mathcal{F}}$.

Remark 5. We remark that, for the important case of identical columns A_j (see section 3), the above value-disjunction reformulation was proposed in the work of Sherali and Smith [21, § 4.1]. It was used to improve the formulation of an integer programming model of a specific problem that contained many symmetric solutions.

The precise link between the extended formulation and the original formulation is given in the following theorem. Before stating the theorem we illustrate our constructions on an example.

Table 1: Sizes of facet descriptions of two reformulations of Example 6

Formulation	Equations	# Facets
original		328
binary-digit expansion	$x_1 + x_2 + x_3 + x_4 = 2^0 z_0 + 2^1 z_1 + 2^2 z_2$	217
value disjunction	$x_1 + x_2 + x_3 + x_4 = 0y_0 + 1y_1 + 2y_2 + 3y_3 + 4y_4$ $y_0 + y_1 + y_2 + y_3 + y_4 = 1$	77

Example 6. Consider the convex hull P of all binary solutions to the inequality

$$3x_1 + 3x_2 + 3x_3 + 3x_4 + 4x_5 + 7x_6 + 8x_7 + 9x_8 + 13x_9 + 15x_{10} \leq 45.$$

We then introduce the subsets

$$N_1 = \{1, 2, 3, 4\},\ N_2 = \{5\},\ \ldots,\ N_7 = \{10\}.$$

We define

$$V_1 = \text{conv}\Big\{ (\mathbf{x}, \mathbf{y}^{N_1}) \in \mathbf{Z}^4_+ \times \{0,1\}^5 : 3x_1 + 3x_2 + 3x_3 + 3x_4 = \\ 0y_0^{N_1} + 3y_1^{N_1} + 6y_2^{N_1} + 9y_3^{N_1} + 12y_4^{N_1} \\ y_0^{N_1} + y_1^{N_1} + y_2^{N_1} + y_3^{N_1} + y_4^{N_1} = 1 \\ 0 \leq x_i \leq 1,\ i = 1, \ldots, 4 \Big\}.$$

Since $V_2, \ldots, V_7$ consist of single points each, these polyhedra are trivial. No additional y-variables are needed. Then, Q becomes

$$Q = \text{conv}\Big\{ (\mathbf{y}^{N_1}, x_5, \ldots, x_{10}) \in \{0,1\}^{11} : 0y^{N_1} + 3y_1^{N_1} + 6y_2^{N_1} + 9y_3^{N_1} + 12y_4^{N_1} + 4x_5 \\ + 7x_6 + 8x_7 + 9x_8 + 13x_9 + 15x_{10} \leq 45 \\ y_0^{N_1} + y_1^{N_1} + y_2^{N_1} + y_3^{N_1} + y_4^{N_1} = 1, \\ x_i \in \{0,1\} \text{ for } i = 5, \ldots, 10 \Big\}.$$

The extended formulation $\bar{\mathcal{F}}$ is then given by:

$$3x_1 + 3x_2 + 3x_3 + 3x_4 + 4x_5 + 7x_6 + 8x_7 + 9x_8 + 13x_9 + 15x_{10} \leq 45 \\ 3x_1 + 3x_2 + 3x_3 + 3x_4 = 0y_0^{N_1} + 3y_1^{N_1} + 6y_2^{N_1} + 9y_3^{N_1} + 12y_4^{N_1} \\ y_0^{N_1} + y_1^{N_1} + y_2^{N_1} + y_3^{N_1} + y_4^{N_1} = 1 \\ \mathbf{x} \in \{0,1\}^{10}, \quad \mathbf{y}^{N_1} \in \{0,1\}^5$$

We used PORTA [9], version 1.3, to compute the facet description of the original formulation and the extended formulation $\bar{\mathcal{F}}$ resulting from the above construction; see also Table 1. In the original formulation there are 328 facets needed to describe the polyhedron. The extended formulation in the 15-dimensional space requires only 77 facets for a complete description. Moreover, as we shall see in the following theorem, the complete description has an interesting block structure.

Remark 7. We remark that there is an obvious alternative way to define an extended formulation based on introducing new variables for the values that the expression $x_1+x_2+x_3+x_4$ can attain. One could introduce an expansion for the values of $x_1+x_2+x_3+x_4$ into binary digits, i.e., one introduces binary variables z_0, z_1, z_2 and requires that $x_1+x_2+x_3+x_4 = 2^0 z_0 + 2^1 z_1 + 2^2 z_2$. In general, this type of reformulation is much more compact than the proposed reformulation. Indeed, one only needs a number of new variables that is *logarithmic* (rather than *linear*) in the number of possible values. Hence, this reformulation seems to have advantages over the proposed reformulation.

However, we observed in many experiments that the facet structure of this binary-digit reformulation is very complicated. For the problem of Example 6, one obtains a polyhedron in a 13-dimensional space that requires 217 facets for a complete description; see Table 1. The number of facets in the complete description is larger than in the proposed value-disjunction reformulation. Moreover, the structure of the individual facets is very complicated, and it seems very unlikely that one could come up with a structural theorem in the spirit of Theorem 8.

Theorem 8 (Structure Theorem for Value Disjunction)**.**

$$\bar{P} = \big\{ (\mathbf{x}, \mathbf{w}, \mathbf{y}) \in \mathbf{R}^n \times \mathbf{R}^d \times [0,1]^{n_1+\cdots+n_K} : (\mathbf{y}, \mathbf{w}) \in Q \textit{ and } (\mathbf{x}^{N_i}, \mathbf{y}^{N_i}) \in V_i \textit{ for all } i \big\}. \tag{4}$$

Proof. The inclusion $\subseteq$ is trivial. We shall prove the inclusion $\supseteq$. Let us consider $(\mathbf{x}, \mathbf{w}, \mathbf{y})$ from the right-hand-side of (4). We try to prove that $(\mathbf{x}, \mathbf{w}, \mathbf{y}) \in P$. For such an $(\mathbf{x}, \mathbf{w}, \mathbf{y})$, we know that $(x^{N_i}, y^{N_i}) \in V_i$. Therefore there exist convex multipliers $\lambda^{N_i,l} \geq 0$ with $\sum_{l=1}^{L_i} \lambda^{N_i,l} = 1$ such that

$$(\mathbf{x}^{N_i}, \mathbf{y}^{N_i}) = \sum_{l=1}^{L_i} \lambda^{N_i,l} (\bar{\mathbf{x}}^{N_i,l}, \bar{\mathbf{y}}^{N_i,l}), \tag{5}$$

where $(\bar{\mathbf{x}}^{N_i,l}, \bar{\mathbf{y}}^{N_i,l})$ is an integral element of V_i and $A(\bar{\mathbf{y}}^{N_i,l}) = A(\bar{\mathbf{x}}^{N_i,l})$. In particular the y-part is made of exactly one 1-entry. Therefore

$$y_t^{N_i} = \sum_{l \in T(N_i,t)} \lambda^{N_i,l} \tag{6}$$

with the sets $T(N_i, t)$, $t = 1, \ldots, n_i$, being a packing of $\{1, \ldots, L_i\}$, namely for all i we have

$$\{1, \ldots, L_i\} = T(N_i, 1) \mathbin{\dot\cup} \ldots \mathbin{\dot\cup} T(N_i, n_i), \tag{7}$$

where $C = A \mathbin{\dot\cup} B$ means $C = A \cup B$ and $A \cap B = \emptyset$. The insight of (6) is shown in Figure 1.

Up to now we have used the fact that $(\mathbf{x}^{N_i}, \mathbf{y}^{N_i}) \in V_i$. We also have a second condition stating that $(\mathbf{y}, \mathbf{w}) \in Q$. Therefore there exist convex multipliers $\sigma_r \geq 0$ with $\sum_{r=1}^{R} \sigma_r = 1$ such that

$$\mathbf{y} = \sum_{r=1}^{R} \sigma_r \hat{\mathbf{y}}^r \quad \text{and} \quad \mathbf{w} = \sum_{r=1}^{R} \sigma_r \hat{\mathbf{w}}^r, \tag{8}$$

where

$$\hat{\mathbf{y}}^r = \big(\hat{\mathbf{y}}^{N_1,r}, \ldots, \hat{\mathbf{y}}^{N_K,r}\big),$$

$$
\begin{pmatrix}
y_1^{N_1} \\
\vdots \\
y_{n_1}^{N_1} \\
\hline
\vdots \\
\hline
y_1^{N_K} \\
\vdots \\
y_{n_K}^{N_K}
\end{pmatrix}
=
\begin{pmatrix}
\lambda^{N_1,\cdot} + \cdots + \lambda^{N_1,\cdot} \\
\vdots \\
\lambda^{N_1,\cdot} + \cdots \\
\hline
\vdots \\
\hline
\lambda^{N_K,\cdot} + \cdots \\
\vdots \\
\lambda^{N_K,\cdot} + \cdots
\end{pmatrix}
$$

Figure 1: Each y is equal to the sum of zero, one or more λ from the convex combination.

and where $\hat{\mathbf{y}}^{N_i,r}$ is a unit vector. Furthermore

$$\sum_{i=1}^{K} A(\hat{\mathbf{y}}^{N_i,r}) + \sum_{j=1}^{d} G_j \hat{\mathbf{w}}_j^r \leq \mathbf{b}.$$

We are now able to express $(\mathbf{x}, \mathbf{w})$ as a convex combination of feasible solutions of $A\mathbf{x} + G\mathbf{w} \leq \mathbf{b}$, using the convex combinations (8) and (5). To do this, we first remark that, similarly to (6), we can express $\mathbf{y}$ in terms of σ_r only, namely

$$y_t^{N_i} = \sum_{s \in S(N_i,t)} \sigma_s, \tag{9}$$

with the sets $S(N_i, t)$, $t = 1, \ldots, n_i$ being a packing of $\{1, \ldots, R\}$, namely

$$\{1, \ldots, R\} = S(N_i, 1) \uplus \ldots \uplus S(N_i, n_i), \tag{10}$$

for all i. By using (6), we therefore conclude that

$$\sum_{s \in S(N_i,t)} \sigma_s = \sum_{l \in T(N_i,t)} \lambda^{N_i,l} \tag{11}$$

By using the similarity of decompositions (9) and (6), we can construct the desired convex combination as follows.

Let us fix r, i.e., we consider each pair $(\sigma_r, \hat{\mathbf{y}}^r)$ separately. We know that $\hat{\mathbf{y}}^r$ is divided into K blocks with a unit vector in each block. In the block N_i, we refer to the index of the non-zero component of $\hat{\mathbf{y}}^r$ as $c(\hat{\mathbf{y}}^{N_i,r})$. Using (6), we can associate to $c(\hat{\mathbf{y}}^{N_i,r})$ a set $T(N_i, c(\hat{\mathbf{y}}^{N_i,r}))$ of indices l, which correspond to multipliers $\lambda^{N_i,l}$ and vectors $\bar{\mathbf{x}}^{N_i,l}$ of the convex combination (5). For every possible choice of indices

$$l_1^r \in T(N_1, c(\hat{\mathbf{y}}^{N_1,r})), \quad \ldots, \quad l_K^r \in T(N_K, c(\hat{\mathbf{y}}^{N_K,r})),$$

we consider the point

$$\mathbf{x}(l_1^r, \ldots, l_K^r) = \left(\bar{\mathbf{x}}^{N_1,l_1^r}, \cdots, \bar{\mathbf{x}}^{N_K,l_K^r}\right)$$

together with

$$\mathbf{w}(l_1^r, \ldots, l_K^r) = \hat{\mathbf{w}}^r$$
$$\mathbf{y}(l_1^r, \ldots, l_K^r) = \hat{\mathbf{y}}^r,$$

with a corresponding coefficient

$$\nu(l_1^r,\ldots,l_K^r) = \sigma_r \frac{\lambda^{N_1,l_1^r}}{\displaystyle\sum_{l\in T(N_1,c(\hat{\mathbf{y}}^{N_1,r}))} \lambda^{N^1,l}} \cdots \frac{\lambda^{N_K,l_K^r}}{\displaystyle\sum_{l\in T(N_K,c(\hat{\mathbf{y}}^{N_K,r}))} \lambda^{N^K,l}}. \tag{12}$$

First we can see that for all $l_1^r,\ldots,l_K^r$, the vector $(\mathbf{x}(l_1^r,\ldots,l_K^r),\hat{\mathbf{w}}^r)$ satisfies $A\,\mathbf{x}(l_1^r,\ldots,l_K^r)+G\hat{\mathbf{w}}^r \le \mathbf{b}$. Indeed,

$$\begin{aligned} A\mathbf{x}(l_1^r,\ldots,l_K^r)+G\hat{\mathbf{w}}^r &= A(\bar{\mathbf{x}}^{N_1,l_1^r})+\cdots+A(\bar{\mathbf{x}}^{N_K,l_K^r})+G\hat{\mathbf{w}}^r \\ &= A(\hat{\mathbf{y}}^{N_1,r})+\cdots+A(\hat{\mathbf{y}}^{N_K,r})+G\hat{\mathbf{w}}^r \\ &= A(\hat{\mathbf{y}}^r)+G\hat{\mathbf{w}}^r \\ &\le \mathbf{b}, \end{aligned}$$

since $(\hat{\mathbf{y}}^r,\hat{\mathbf{w}}^r)$ is a mixed-0/1 solution of Q. It now suffices to prove that $(\mathbf{x},\mathbf{w},\mathbf{y})$ is the convex combination of all the $(\mathbf{x},\mathbf{w},\mathbf{y})(l_1^r,\ldots,l_K^r)$ using the corresponding coefficients $\nu(l_1^r,\ldots,l_K^r)$. First we clearly have, if we fix N_i and an index $j\in N_i$,

$$\begin{aligned} y_j^{N_i} &= \sum_{r=1}^R \sum_{l_1^r\in T(N_1,c(\hat{\mathbf{y}}^{N_1,r}))} \cdots \sum_{l_K^r\in T(N_K,c(\hat{\mathbf{y}}^{N_K,r}))} \nu(l_1^r,\ldots,l_K^r)\hat{y}_j^{N_i,r} \\ &= \sum_{r=1}^R \sigma_r \hat{y}_j^{N_i,r} \end{aligned}$$

which is clear using (8). Similarly $w_j=\sum_{r=1}^R \sigma_r\hat{w}_j^r$. If we fix again N_i and an index $j\in N_i$, we have

$$\begin{aligned} x_j^{N_i} &= \sum_{r=1}^R \sum_{l_1^r\in T(N_1,c(\hat{\mathbf{y}}^{N_1,r}))} \cdots \sum_{l_K^r\in T(N_K,c(\hat{\mathbf{y}}^{N_K,r}))} \nu(l_1^r,\ldots,l_K^r) x_j^{N_i}(l_1^r,\ldots,l_K^r) \\ &= \sum_{r=1}^R \sum_{l_1^r\in T(N_1,c(\hat{\mathbf{y}}^{N_1,r}))} \cdots \sum_{l_K^r\in T(N_K,c(\hat{\mathbf{y}}^{N_K,r}))} \nu(l_1^r,\ldots,l_K^r)\bar{x}_j^{N_i,l_i^r} \\ &= \sum_{r=1}^R \sum_{l_i^r\in T(N_i,c(\hat{\mathbf{y}}^{N_i,r}))} \sigma_r \frac{\lambda^{N_i,l_i^r}}{\displaystyle\sum_{l\in T(N_i,c(\hat{\mathbf{y}}^{N_i,r}))} \lambda^{N_i,l}} \bar{x}_j^{N_i,l_i^r}, \end{aligned} \tag{13}$$

the last identity being obtained using (12). For a fixed i, we have, using (10),

$$\{1,\ldots,R\} = S(N_i,1)\cup\ldots\cup S(N_i,n_i).$$

Therefore we can rewrite (13) using indices running over the different $S(N_i,k)$. Remark also that when we fix $r\in S(N_i,k)$, we have $c(\mathbf{y}^{\hat{N}_i,r})=k$. We hence have

$$
\begin{aligned}
x_j^{N_i} &= \sum_{k=1}^{n_i} \sum_{p \in S(N_i,k)} \sum_{l \in T(N_i,k)} \sigma_p \frac{\lambda^{N_i,l}}{\sum\limits_{q \in T(N_i,k)} \lambda^{N_i,q}} \bar{x}_j^{N_i,l} \\
&= \sum_{k=1}^{n_i} \sum_{l \in T(N_i,k)} \frac{\sum\limits_{p \in S(N_i,k)} \sigma_p}{\sum\limits_{q \in T(N_i,k)} \lambda^{N_i,q}} \lambda^{N_i,l} \bar{x}_j^{N_i,l} \\
&= \sum_{k=1}^{n_i} \sum_{l \in T(N_i,k)} \lambda^{N_i,l} \bar{x}_j^{N_i,l}, \qquad (14)
\end{aligned}
$$

where (14) is obtained using (11). We can use (7) namely

$$T(N_i, 1) \cup \ldots \cup T(N_i, n_i) = \{1, \ldots, L_i\}.$$

In particular it allows us to sum over $\{1, \ldots, L_i\}$ in (14) instead of the summation over k and l. We therefore finally have

$$x_j^{N_i} = \sum_{l=1}^{L_i} \lambda^{N_i,l} \bar{x}_j^{N_i,l},$$

which is the desired result using (5). Finally, the sum of the ν coefficients is equal to 1 due to their construction and the fact that $\sum_{r=1}^{R} \sigma_r = 1$. □

Example 9. Consider the set

$$\mathcal{F} = \{x \in \{0, 1, 2\}^4 : x_1 + x_2 + 2x_3 + 3x_4 \leq 7\}.$$

The complete facet description of conv $\mathcal{F}$ is given by the 14 inequalities $\mathbf{c}^\top \mathbf{x} \leq \gamma$ shown in Table 2.

Table 2: The complete description of Example 9 in the original space

c_1	c_2	c_3	c_4	γ	c_1	c_2	c_3	c_4	γ
-1	0	0	0	≤ 0	1	0	0	1	≤ 3
0	-1	0	0	≤ 0	0	1	0	1	≤ 3
0	0	-1	0	≤ 0	0	0	1	2	≤ 4
0	0	0	-1	≤ 0	1	1	1	1	≤ 5
1	0	0	0	≤ 2	0	1	2	2	≤ 6
0	1	0	0	≤ 2	1	0	2	2	≤ 6
0	0	1	0	≤ 2	1	1	2	3	≤ 7

We now construct a value disjunction of the set $\mathcal{F}$. To do this, we consider three blocks $N_1 = \{1, 2\}$, $N_2 = \{3\}$, $N_4 = \{4\}$. In block N_1 we consider the linear form $x_1 + x_2$, which can take the values $0, 1, \ldots, 4$ because x_1 and x_2 have an upper bound of 2. We introduce thus five variables y_0, y_1, y_2, y_3, y_4 corresponding to the possible values. The blocks N_2 and N_3 are trivial, so we do not need to introduce new variables in those cases. A valid extended formulation $\bar{\mathcal{F}}$ for $\mathcal{F}$ is thus

$$\begin{aligned}\bar{\mathcal{F}} = \{\, (\mathbf{x}, \mathbf{y}) \in \{0,1,2\}^4 \times \{0,1\}^5 : \; & 0y_0 + 1y_1 + 2y_2 + 3y_3 + 4y_4 + 2x_3 + 3x_4 \leq 7 \\ & x_1 + x_2 = 0y_0 + 1y_1 + 2y_2 + 3y_3 + 4y_4 \\ & y_0 + y_1 + y_2 + y_3 + y_4 = 1 \,\}.\end{aligned}$$

Theorem 8 now asserts that we obtain the complete description of the extended formulation $\bar{\mathcal{F}}$ of $\mathcal{F}$ by combining the complete descriptions of the polyhedra

$$\begin{aligned}V_1 = \operatorname{conv}\{\, (x_1, x_2, \mathbf{y}) \in \{0,1,2\}^2 \times \{0,1\}^5 : \; & x_1 + x_2 = 0y_0 + 1y_1 + 2y_2 + 3y_3 + 4y_4 \\ & y_0 + y_1 + y_2 + y_3 + y_4 = 1 \quad \},\end{aligned}$$

and

$$\begin{aligned}Q = \operatorname{conv}\{\, (x_3, x_4, \mathbf{y}) \in \{0,1,2\}^2 \times \{0,1\}^5 : \; & 2x_3 + 3x_4 + 0y_0 + 1y_1 + 2y_2 + 3y_3 + 4y_4 \leq 7 \\ & y_0 + y_1 + y_2 + y_3 + y_4 = 1 \quad \}.\end{aligned}$$

We obtain the facet description given by the inequalities $\mathbf{c}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y} \leq \gamma$ shown in Table 3. For each non-trivial inequality, we also mention whether it comes from V_1 or from Q.

Table 3: The complete description of Example 9 in the extended space

c_1	c_2	c_3	c_4	d_0	d_1	d_2	d_3	d_4	γ	Origin
		-1							≤ 0	
			-1						≤ 0	
				-1					≤ 0	
					-1				≤ 0	
						-1			≤ 0	
							-1		≤ 0	
								-1	≤ 0	
	-1						1	2	≤ 0	V_1
-1							1	2	≤ 0	V_1
				1	1	1	1	1	$= 1$	Q, V_1
		1						1	≤ 2	Q
			1			1	1	1	≤ 2	Q
		1	1		1	1	1	2	≤ 3	Q
		1	2			1	2	2	≤ 4	Q
1	1				-1	-2	-3	-4	$= 0$	V_1

In the example it turns out that the number of inequalities describing $\operatorname{conv} \mathcal{F}$ and $\operatorname{conv} \bar{\mathcal{F}}$ is the same in the two representations. This, however, is not always true. Moreover, an inherent advantage of the second formulation over the first formulation is that its structure is better known. In particular, it may occur that the same polyhedron V_i appears in several different problems. In this case, the knowledge about the description of the polyhedron V_i can be used over and over again.

The next section presents the case of a polyhedron that appears often in our experiments, namely the V_i polyhedron where all the coefficients of the variables x are the same. We show that we can compute a full description for this object.

3 A special family of linking polyhedra

In this section we study the linking polyhedra V_i for the case where the columns A_j for $j \in N_i$ are identical and the variables x_j are binary. Hence the set of possible values is $\mathcal{A}_i = \{\, kA_j : k = 0, \dots, |N_i| \,\}$, so $n_i = |\mathcal{A}_i| = 1 + |N_i|$. To simplify the notation, we shall index the variables $y_k^{N_i}$ by $k = 0, \dots, |N_i|$. In other words, we study the polytope

$$V_i = \operatorname{conv}\{(\mathbf{x}^{N_i}, \mathbf{y}^{N_i}) \in \{0,1\}^{|N_i|} \times \{0,1\}^{n_i} : \sum_{j \in N_i} x_j = \sum_{k=0}^{|N_i|} k y_k^{N_i} \\ \sum_{k=0}^{|N_i|} y_k^{N_i} = 1 \quad \}.$$

We are able to give the complete, exponential-size description of this polytope V_i and a polynomial-time separation algorithm.

Theorem 10. *V_i is a polytope whose affine hull is given by the equations:*

$$\sum_{j \in N_i} x_j = \sum_{k=0}^{|N_i|} k y_k^{N_i} \tag{15a}$$

$$\sum_{k=0}^{|N_i|} y_k^{N_i} = 1 \tag{15b}$$

The facets of V_i are given by:

$$\sum_{j \in T} x_j - \sum_{k=1}^{|T|} k y_k - \sum_{k=|T|+1}^{|N_i|} |T| y_k \le 0 \qquad \text{for } \emptyset \ne T \subset N_i \tag{15c}$$

$$y_k^{N_i} \ge 0 \qquad \text{for } k = 0, \dots, |N_i|. \tag{15d}$$

Proof. We first show that the inequalities (15) are valid for V_i. To this end, let $(\mathbf{x}, \mathbf{y}) \in \{0,1\}^{|N_i|} \times \{0,1\}^{n_i}$ be a vertex of V_i. If $\mathbf{y} = \mathbf{e}^0$, then we have $\mathbf{x} = \mathbf{0}$, and inequality (15c) is trivially satisfied. Otherwise, $\mathbf{y} = \mathbf{e}^k$ with $k = \sum_{j \in N_i} x_j = |\operatorname{supp} \mathbf{x}^{N_i}| \in \{1, \dots, |N_i|\}$. Let $\emptyset \ne T \subset N_i$ be arbitrary. If $k \le |T|$, we have

$$\sum_{j \in T} x_j - \sum_{k=1}^{|T|} k y_k - \sum_{k=|T|+1}^{|N_i|} |T| y_k = \sum_{j \in T} x_j - k \le 0.$$

On the other hand, if $k > |T|$, we have

$$\sum_{j \in T} x_j - \sum_{k=1}^{|T|} k y_k - \sum_{k=|T|+1}^{|N_i|} |T| y_k = \sum_{j \in T} x_j - |T| \le 0.$$

Hence, (15c) is satisfied. The remaining inequalities are trivially valid for V_i.

For the ease of notation we let $N = N_i$, $n = |N|$ and substitute the variables $y_k^{N_i}$ by simply y_k. Let $\mathbf{c}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y} \le \gamma$ be a facet-defining inequality of V_i and set

$$F = \{\, (\mathbf{x}, \mathbf{y}) \in V_i : \mathbf{c}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y} = \gamma \,\}.$$

We will show that $\mathbf{c}^\top\mathbf{x}+\mathbf{d}^\top\mathbf{y} \le \gamma$ corresponds to one of the inequalities in (15) up to multiplication by a scalar. We assume that the variables in N are reordered such that $c_1 \ge c_2 \ge \ldots \ge c_n$. Since V_i is not full dimensional, we first transform $\mathbf{c}^\top\mathbf{x}+\mathbf{d}^\top\mathbf{y} \le \gamma$ into a standard form. This can be achieved by adding multiples of the equations (15a) and (15b) to $\mathbf{c}^\top\mathbf{x}+\mathbf{d}^\top\mathbf{y} \le \gamma$. More precisely, we first proceed with the following three steps.

(1) By adding a multiple of equation (15b), ensure that $d_0 = 0$.

(2) While there exists an index $i \in N$ such that $c_i < 0$, add $-c_i$ times Equation (15a) to the inequality $\mathbf{c}^\top\mathbf{x}+\mathbf{d}^\top\mathbf{y} \le \gamma$. Let us again denote by $\mathbf{c}^\top\mathbf{x}+\mathbf{d}^\top\mathbf{y} \le \gamma$ the resulting inequality. Notice that after terminating with Step 1, we have that $c_i \ge 0$ for all $i \in N$ and $c_n = 0$.

(3) If $c_i > 0$ for all $i \in N$ and there exist $i, j \in N$ such that $c_i \ne c_j$, then $c_1 > c_n > 0$ due to our reordering. In this case we subtract c_n times Equation (15a) from the inequality $\mathbf{c}^\top\mathbf{x}+\mathbf{d}^\top\mathbf{y} \le \gamma$. Notice that also after Step (2) has been performed we have that $c_n = 0$ and $c_i \ge 0$ for all $i \in N$.

The preprocessing steps (1) and (2) guarantee that $c_i \ge 0$ for all $i \in N$. Now let $s \in \{0, \ldots, n\}$ be an index such that

$$c_1 \ge c_2 \ge \ldots \ge c_s > 0 = c_{s+1} = \ldots = c_n.$$

We define $T = \{\, i \in N : c_i > 0 \,\} = \{1, \ldots, s\}$. We consider the following cases.

Case 1. If $T = \emptyset$, i.e., $c_1 = \cdots = c_n = 0$, it follows that $\mathbf{c}^\top\mathbf{x}+\mathbf{d}^\top\mathbf{y} \le \gamma$ is a multiple of the inequality $\sum_{k=1}^n y_k \le 1$ or of the non-negativity constraints $y_k \ge 0$.

Indeed, because $(\mathbf{0}, \mathbf{0})$ is feasible, we have $\gamma \ge 0$. Since F is a facet, there must be $2n-1$ affinely independent feasible points on it. If $\gamma = 0$, we have $(\mathbf{0}, \mathbf{0}) \in F$; therefore, for all but one $k = 1, \ldots, n$, a point $(\mathbf{x}, \mathbf{e}^k)$ must be contained in F. This means that $d_k = \gamma = 0$ for all but one $k = 1, \ldots, n$. For the remaining one $\tilde{k} \in \{1, \ldots, n\}$ we have $d_{\tilde{k}} \le \gamma = 0$, so $\mathbf{c}^\top\mathbf{x}+\mathbf{d}^\top\mathbf{y} \le \gamma$ is a scalar multiple of the non-negativity constraint $y_{\tilde{k}} \ge 0$.

On the other hand, if $\gamma > 0$, then $(\mathbf{0}, \mathbf{0}) \notin F$, so we have $F \subseteq \{\, (\mathbf{x}, \mathbf{y}) \in V_i : \sum_{k=1}^n y_k = 1 \,\}$, since $(\mathbf{0}, \mathbf{0})$ is the only feasible integer point with $\mathbf{y} = \mathbf{0}$. Because F is a facet, we have $F = \{\, (\mathbf{x}, \mathbf{y}) \in V_i : \sum_{k=1}^n y_k = 1 \,\}$, which corresponds to (15b).

Case 2. If $T = N$, we conclude from our previous analysis that $c_i = c_j \ne 0$ for all $i, j \in N$. It follows that $\mathbf{c}^\top\mathbf{x}+\mathbf{d}^\top\mathbf{y} \le \gamma$ is implied by Equation (15a), a contradiction that F defines a facet of V_i.

Case 3. Therefore, we may assume that $\emptyset \ne T \subset N$, $T \ne N$. Again, since $(\mathbf{0}, \mathbf{0})$ is feasible, we have that $\gamma \ge 0$. If $\gamma > 0$, then $F \subseteq \{\, (\mathbf{x}, \mathbf{y}) \in V_i : \sum_{k=1}^n y_k = 1\}$. Hence, we can assume that $\gamma = 0$.

We next define indices $1 \le i_1 < i_2 < \ldots < i_r \le s$ as follows:

$$c_1 = \ldots = c_{i_1} > c_{i_1+1} = \ldots = c_{i_2} > \ldots > c_{i_r+1} = \ldots = c_s.$$

By testing the inequality $\mathbf{c}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y} \leq 0$ with the feasible points $(\mathbf{e}^1, \mathbf{e}^1)$, $(\mathbf{e}^1 + \mathbf{e}^2, \mathbf{e}^2)$, $(\mathbf{e}^1 + \mathbf{e}^2 + \mathbf{e}^3, \mathbf{e}^3)$, $\ldots$, we conclude that

$$
\begin{aligned}
-d_1 &\geq c_1 \\
-d_2 &\geq c_1 + c_2 \\
&\vdots \\
-d_{i_1} &\geq c_1 + c_2 + \ldots + c_{i_1} \\
-d_{i_1+1} &\geq \textstyle\sum_{j=1}^{i_1} c_j + c_{i_1+1} \\
-d_{i_1+2} &\geq \textstyle\sum_{j=1}^{i_1} c_j + c_{i_1+1} + c_{i_1+2} \\
&\vdots \\
-d_{i_2} &\geq \textstyle\sum_{j=1}^{i_1} c_j + c_{i_1+1} + c_{i_1+2} + \ldots + c_{i_2} \\
&\vdots \\
-d_{i_r+1} &\geq \textstyle\sum_{j=1}^{i_r} c_j + c_{i_r+1} \\
-d_{i_r+2} &\geq \textstyle\sum_{j=1}^{i_r} c_j + c_{i_r+1} + c_{i_r+2} \\
&\vdots \\
-d_s &\geq \textstyle\sum_{j=1}^{i_r} c_j + c_{i_r+1} + c_{i_r+2} + \ldots + c_s
\end{aligned}
$$

Therefore, the inequality $\mathbf{c}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y} \leq \gamma = 0$ is dominated by the following conic combination of the inequalities (15c):

$$
\begin{aligned}
c_{i_r} &\times \left(\textstyle\sum_{i=1}^{s} x_i - \sum_{k=1}^{s} k y_k - \sum_{k=s+1}^{n} s y_k \leq 0 \right) \\
+ (c_{i_r} - c_{i_r-1}) &\times \left(\textstyle\sum_{i=1}^{i_r} x_i - \sum_{k=1}^{i_r} k y_k - \sum_{k=i_r+1}^{n} i_r y_k \leq 0 \right) \\
&\vdots \\
+ (c_{i_1} - c_{i_2}) &\times \left(\textstyle\sum_{i=1}^{i_1} x_i - \sum_{k=1}^{i_1} k y_k - \sum_{k=i_1+1}^{n} i_1 y_k \leq 0 \right).
\end{aligned}
$$

This completes the proof. □

Theorem 11. *The separation problem over the linking polyhedron V_i in the case of identical coefficients can be solved in polynomial time.*

Proof. Let $(\mathbf{x}^*, \mathbf{y}^*)$ be a point satisfying the polynomially many constraints (15a, 15b, 15d). We show that, in polynomial time, we can decide whether $(\mathbf{x}^*, \mathbf{y}^*)$ satisfies the exponentially many inequalities (15c); if it does not, we can construct a maximally violated inequality.

It is clear that among the inequalities (15c) with equal cardinality $|T| = s$, a most violated inequality is the one where T is the index set of the s largest components x^*_j. Therefore it suffices to sort the variables $x^*_1, \ldots, x^*_{|N_i|}$ such that

$$x^*_1 \geq x^*_2 \geq \cdots \geq x^*_s > 0 = x_{s+1} = \cdots = x^*_{|N_i|}.$$

Then we can simply evaluate the violation of inequality (15c) for the sets $\{1\}$, $\{1, 2\}$, $\{1, 2, 3\}$, $\ldots$, $\{1, \ldots, s\}$ and pick the set which yields the maximal violation. □

4 An application: The knapsack with three distinct coefficients

In this section, we show that the value-disjunction procedure is a tool to compute complete descriptions in an extended space. As an example we consider the 0/1 knapsack problem with three distinct coefficients:

$$\sum_{j\in N_1} \mu x_j + \sum_{j\in N_2} \lambda x_j + \sum_{j\in N_3} \sigma x_j \leq \beta, \tag{16}$$

where N_1, N_2, N_3 are pairwise disjoint index sets. The convex hull of the feasible solutions can have exponentially many vertices and facets. Moreover, the complete facet description for (16) is not known in general. In [22], the case of the knapsack with two different coefficients was solved. By applying the structure theorem for value disjunctions (Theorem 8), we are able to give a complete description for an extended formulation of (16) using only polynomially many variables.

We consider the extended formulation of (16),

$$\begin{aligned}
&\sum_{j\in N_1} \mu x_j + \sum_{j\in N_2} \lambda x_j + \sum_{j\in N_3} \sigma x_j \leq \beta \\
&\qquad \sum_{j\in N_i} x_j = \sum_{k=0}^{|N_i|} k y_k^i && \text{for } i = 1,2,3 \\
&\qquad \sum_{k=0}^{|N_i|} y_k^i = 1 && \text{for } i = 1,2,3 \\
&\qquad \mathbf{x} \in \{0,1\}^{|N_1|+|N_2|+|N_3|} \\
&\qquad \mathbf{y}^i \in \{0,1\}^{|N_i|+1} && \text{for } i = 1,2,3.
\end{aligned}$$

Theorem 8 provides us the framework to describe the convex hull of such an extended formulation. It is given by the intersection of the linking polyhedron and the aggregated polyhedron. The linking polyhedron was studied in the last section. Theorem 10 gives a complete facet description of it. Concerning the aggregated polyhedron, we will make use of a vertex description. It is the convex hull of the set described by

$$\begin{aligned}
&\mu \sum_{k=1}^{|N_1|} k y^{N_1,k} + \lambda \sum_{k=1}^{|N_2|} k y^{N_2,k} + \sigma \sum_{k=1}^{|N_3|} k y^{N_3,k} \leq \beta \\
&\qquad \sum_{k=0}^{|N_i|} y^{N_i,k} = 1 && \text{for } i = 1,2,3 \\
&\qquad \mathbf{y}^{N_i} \in \{0,1\}^{|N_i|+1} && \text{for } i = 1,2,3.
\end{aligned}$$

Clearly there are at most $(1 + |N_1|) \cdot (1 + |N_2|) \cdot (1 + |N_3|)$ vertices. We denote them by $\mathbf{v}^1, \ldots, \mathbf{v}^p \in \{0,1\}^{|N_1|+|N_2|+|N_3|+3}$.

Theorem 12. *The complete facet description of* (16) *in an extended space is* *given by:*

$$\begin{aligned}
&\mathbf{y} = \sum_{j=1}^{p} \mathbf{v}^j z_j \\
&\sum_{j=1}^{p} z_j = 1 \\
&z_j \geq 0 && \textit{for } j = 1, \dots, p \\
&\sum_{j \in N_i} x_j^{N_i} = \sum_{k=0}^{|N_i|} k y^{N_i,k} && \textit{for } i = 1, 2, 3 \\
&\sum_{j \in T} x_j^{N_i} \geq \sum_{\substack{k \in \{1,\dots,|N_i|\}: \\ |T|+k>|N_i|}} (|T| + k - |N_i|) y^{N_i,k} && \textit{for } i = 1, 2, 3 \textit{ and } \emptyset \neq T \subset N_i \\
&\mathbf{x} \in \mathbf{R}^{|N_1|+|N_2|+|N_3|} \\
&\mathbf{y} \in \mathbf{R}^{|N_1|+|N_2|+|N_3|+3} \\
&\mathbf{z} \in \mathbf{R}^{p}.
\end{aligned}$$

Proof. This follows from Theorem 8. □

It is straightforward to extend our construction to binary integer programs with a fixed number of different columns.

5 Branching on value disjunctions

So far we have presented the value-disjunction technique as a theoretical tool to define extended formulations which may yield more tractable polyhedral descriptions. Clearly it would be too much to expect general results on the existence or constructability of an intermediate representation for an arbitrary integer program that is better than the original formulation. The more modest goal of this section is to provide evidence for the practical usefulness of the value-disjunction technique, using a limited set of computational experiments.

We shall restrict ourselves to experiments where we perform branching on the new binary variables of the extended formulation. We first need to discuss the situations for which we propose to make use of our new technique, so as to complement the existing branch-and-cut techniques.

On the simplification effect of branching. Today mixed-integer linear programs are solved using branch-and-cut algorithms, i.e., such an algorithm consists of the combination of two techniques, the cutting technique (with the objective to tighten a current formulation) and the branching technique (with the objective to split a problem into a disjunction of subproblems with fewer variables). However as of today there are essentially no mathematical arguments available that help to decide when it is more efficient to branch or to cut. This question is fundamental since computational experiments clearly reveal that neither a pure branch-and-bound algorithm nor a pure cutting plane algorithm can solve the instances that the combination of the two can manage to solve. One partial answer to this question is given by the fact that branching does not only generate subproblems with fewer variables, but, more importantly, the polyhedral description of each of the two subproblems is significantly easier than the original facet description. We illustrate this point through an example.

Example 13. We consider the feasible region

$$\begin{aligned} 7x_1 + 5x_2 - x_3 - x_4 - 2x_5 - 3x_6 - 4x_7 - 6x_8 \leq 1 \\ x_i \in \{0, 1\}. \end{aligned}$$

The non-trivial facets of the convex hull are shown in Table 4. If we consider the four subproblems where the variables x_7 and x_8 are fixed to the possible values, we obtain much simpler facet descriptions; see Table 5.

This example illustrates why branching is such an important tool in solving mixed-integer programs. The question emerges how to obtain branching decisions such that the polyhedral description for each of the branches becomes as easy as possible. Thus, when we compare branching decisions in our experiments, we shall use the following definition.

Definition 14. The *complete description size* of an n-way branching decision is defined as the sum of the numbers of facets in the complete descriptions of the n subproblems.

Clearly this definition should only be used for comparing branching decisions with an equal number of subproblems. For our experiments, we used PORTA [9], version 1.3, to enumerate the feasible solutions and to compute the facet description of their convex hull. As the computation times for problems of higher dimension would be prohibitive, we had to restrict ourselves to experiments with very small integer programs. Specifically, we generated dense 0/1 problems with twelve binary variables and two rows. The four test instances are shown in Table 6.

On the limitations of current LP-based branching schemes. A single-variable branching scheme, which is used in today's branch-and-cut systems, is usually driven by information obtained from the current LP relaxation ("most infeasible variable selection"), by look-ahead-based techniques ("strong branching"), and history-based prediction ("pseudo-cost branching"). There is a large class of problems that are extremely difficult to solve for current branch-and-cut systems because none of the above criteria provides a meaningful basis for a branching decision. An extreme example for this are the market split instances by Cornuéjols and Dawande [10]: Here the LP relaxations of all subproblems have the value 0, until most of the variables have already been fixed. However, it was shown that branch-and-bound is indeed the right tool for solving the market split instances: While LP-based single-variable branching fails, it is very successful to branch on certain general disjunctions that can be derived from the problem structure via lattice basis reduction [2]. Though this technique has proved very successful for solving market split problems [1] and also for the so-called banker's problem [15], it has not become a general tool for branch-and-cut systems.

We also refer to the recent work [12] where a branching method along general disjunctions is proposed. Here the quality of a disjunction (branching direction) is measured by the depth of the intersection cut corresponding to the disjunction; among the best disjunctions, strong branching is used to select one. The

Table 4: Full description of Example 13

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	γ	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	γ
				−1		−1	−1	≤ 0	3	2	−1	−1	−1	−1	−1	−2	≤ 1
					−1	−1	−1	≤ 0	2	2	−1	−1	−2		−2	−1	≤ 1
		−1	−1			−1	−1	≤ 0	3	2	−1	−1	−2		−2	−2	≤ 1
		−1		−1	−1		−1	≤ 0	3	1	−1	−1	−2	−2		−2	≤ 1
			−1	−1	−1		−1	≤ 0	3	3	−1	−1	−2	−1	−2	−2	≤ 1
	1	−1			−1	−1	−1	≤ 0	3	2	−1		−1	−1	−1	−3	≤ 1
	1		−1		−1	−1	−1	≤ 0	3	2		−1	−1	−1	−1	−3	≤ 1
	1			−1	−1	−1	−1	≤ 0	3	2			−1	−1	−2	−3	≤ 1
	1	−1	−1	−1		−1	−1	≤ 0	3	3	−1		−1	−2	−3	−2	≤ 1
2	1	−1		−1	−1	−1	−2	≤ 0	3	3		−1	−1	−2	−3	−2	≤ 1
2	1		−1	−1	−1	−1	−2	≤ 0	4	2	−1		−2	−1	−2	−3	≤ 1
2	1			−1	−1	−2	−2	≤ 0	4	2		−1	−2	−1	−2	−3	≤ 1
2	1	−1	−1	−1		−2	−2	≤ 0	4	3	−1	−1	−2	−1	−2	−3	≤ 1
3	2	−1	−1	−1	−1	−2	−3	≤ 0	4	3	−1		−1	−2	−3	−3	≤ 1
3	2	−1		−1	−2	−2	−3	≤ 0	4	3		−1	−1	−2	−3	−3	≤ 1
3	2		−1	−1	−2	−2	−3	≤ 0	4	2	−1	−1	−2		−3	−3	≤ 1
3	2	−1	−1		−2	−3	−3	≤ 0	4	4	−1	−1	−1	−3	−3	−3	≤ 1
3	3	−1	−1	−1	−2	−3	−3	≤ 0	4	3	−1	−1	−1	−1	−2	−4	≤ 1
6	4	−1	−1	−2	−3	−4	−6	≤ 0	4	3	−1		−1	−2	−2	−4	≤ 1
4	3	−1	−1	−1	−2	−3	−4	≤ 0	4	3		−1	−1	−2	−2	−4	≤ 1
5	3	−1	−1	−2	−2	−3	−5	≤ 0	7	5	−1	−1	−2	−3	−4	−6	≤ 1
	1	−1					−1	≤ 1	5	4	−1	−1	−1	−3	−3	−4	≤ 1
	1		−1				−1	≤ 1	5	5	−1	−1	−2	−3	−4	−4	≤ 1
	1			−1			−1	≤ 1	5	4	−1	−1	−1	−2	−3	−5	≤ 1
	1				−1	−1		≤ 1	6	4	−1	−1	−2	−2	−3	−5	≤ 1
	1				−1		−1	≤ 1	6	4	−1		−2	−3	−4	−5	≤ 1
	1					−1	−1	≤ 1	6	4		−1	−2	−3	−4	−5	≤ 1
2	1				−1	−1	−1	≤ 1	3	2	−1		−1		−1	−2	≤ 2
	1	−1		−1		−1		≤ 1	3	2		−1	−1		−1	−2	≤ 2
	1		−1	−1		−1		≤ 1	3	2	−1	−1			−1	−3	≤ 2
2	1	−1		−1		−1	−1	≤ 1	3	3	−1	−1	−1		−1	−3	≤ 2
2	1		−1	−1		−1	−1	≤ 1	4	3	−1	−1	−1	−1	−1	−3	≤ 2
	1	−1	−1	−1	−1			≤ 1	5	3	−1		−2	−1	−2	−4	≤ 2
2	1	−1	−1	−1	−1		−1	≤ 1	5	3		−1	−2	−1	−2	−4	≤ 2
2	2	−1	−1	−1	−1	−1	−1	≤ 1	5	4	−1	−1	−2	−1	−2	−4	≤ 2
2	1			−1		−1	−2	≤ 1	5	4	−1		−1	−2	−3	−4	≤ 2
2	2	−1			−1	−1	−2	≤ 1	5	4		−1	−1	−2	−3	−4	≤ 2
2	2		−1		−1	−1	−2	≤ 1	6	5	−1	−1	−1	−3	−3	−5	≤ 2
2	2			−1	−1	−1	−2	≤ 1									

Table 5: Full description of the subproblems of Example 13

Branch $x_7 = 0, x_8 = 0$

c_1	c_2	c_3	c_4	c_5	c_6	γ
1				−1		≤ 0
1					−1	≤ 0
1		−1	−1			≤ 0
2	1			−1	−1	≤ 0
1	1	−1			−1	≤ 0
1	1		−1		−1	≤ 0
2	1	−1	−1	−1		≤ 0
3	2	−1	−1		−2	≤ 0
4	3	−1	−1	−1	−2	≤ 0

Branch $x_7 = 1, x_8 = 0$

c_1	c_2	c_3	c_4	c_5	c_6	γ
1		−1		−1	−1	≤ 0
1			−1	−1	−1	≤ 0
1						≤ 1
1	1	−1				≤ 1
1	1		−1			≤ 1
1	1			−1		≤ 1
1	1				−1	≤ 1
3	2	−1		−1	−1	≤ 2
3	2		−1	−1	−1	≤ 2

Branch $x_7 = 0, x_8 = 1$

c_1	c_2	c_3	c_4	c_5	c_6	γ
1						≤ 1
1	1				−1	≤ 1
1	1	−1		−1		≤ 1
1	1		−1	−1		≤ 1

Branch $x_7 = 1, x_8 = 1$

c_1	c_2	c_3	c_4	c_5	c_6	γ
1	1	−1	−1	−1	−1	≤ 1

Table 6: Randomly generated problem instances. Instances 1 and 2 have been generated randomly by drawing the coefficients independently and uniformly from the set $\{-20, \ldots, +20\}$. The right-hand side is always 0. Instance 3 has been modified manually, so that the first three variables have identical coefficients. Finally, instance 4 is a variation of instance 3 where the coefficients of the first three variables are very close to each other.

Matrix data												
A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	**b**
Instance 1												
11	−7	9	10	−2	7	14	−15	4	−5	−2	−19	≤ 0
6	18	−4	−9	17	−11	5	−12	5	3	−18	7	≤ 0
Instance 2												
3	−7	0	8	12	−1	7	−14	13	20	−18	2	≤ 0
9	11	−13	19	8	−15	−5	3	7	18	−6	−10	≤ 0
Instance 3												
7	7	7	15	−21	−15	−23	−12	12	−6	11	10	≤ 0
10	10	10	−21	4	−3	4	13	−1	−14	2	−6	≤ 0
Instance 4												
7	6	7	15	−21	−15	−23	−12	12	−6	11	10	≤ 0
10	10	9	−21	4	−3	4	13	−1	−14	2	−6	≤ 0

computational results for many benchmark problems from MIPLIB are very promising. However, for a few instances the proposed branching scheme fails to close any gap. This includes the market split instances `markshare1` and `markshare2`.

A branching scheme based on value disjunctions. We propose a new branching scheme based on value disjunctions, which we hope is general enough to be useful as a branching scheme for general integer programs. It is purely based on the analysis of the structure of the integer program, and is designed to complement the above mentioned LP-based prediction methods.

The basic idea of the new branching scheme is to partition the set N of problem variables into blocks N_i and to move over to the extended formulation given by the value disjunction. In addition to the original variables, we can then branch on the newly introduced binary variables. In fact, because exactly one binary variable of each block can be set to 1, we can perform SOS branching on these variables. The question, of course, is how to construct a suitable partition of N.

Claim 1. *One should choose a set of variables whose columns are structurally similar and perform a value disjunction according to a relaxation where we replace the original coefficients by simpler ones.*

For our experiment, we decided to pick three of the twelve binary variables, x_i, x_j, x_k, say. We then add the (redundant) constraint $x_i + x_j + x_k \leq 3$. When we construct a value disjunction with respect to this constraint, we need to introduce four variables y_0, y_1, y_2, y_3, corresponding to the possible values of the form $x_i + x_j + x_k$. Performing SOS branching on $y_0 + y_1 + y_2 + y_3 = 1$ yields four subproblems. To compute the complete description size of the value disjunction branching on x_i, x_j, x_k, we sum up the numbers of facets in each of these four subproblems. To make a comparison with traditional single-variable branching, we need to consider a branching strategy that yields the same number of subproblems. To this end, we pick two original variables, x_p, x_q say, and consider the subproblems where we fix these variables to the possible values.

We next defined a "ranking formula" for the selection of the three variables x_i, x_j, x_k that give rise to the value disjunction. Let A_i, A_j, A_k denote the columns of these variables. Then let

$$R(\{i,j,k\}) = \min_{r=1}^{2} \frac{(\max\{A_{r,i}, A_{r,j}, A_{r,k}\} - \min\{A_{r,i}, A_{r,j}, A_{r,k}\})^2}{2 + |\mathrm{med}\{A_{r,i}, A_{r,j}, A_{r,k}\}|}$$

where $\mathrm{med}\{A_{r,i}, A_{r,j}, A_{r,k}\}$ denotes the median of the three values. The formula was designed so that (i) columns that have "similar" coefficients in at least one of the rows yield a low (good) result; (ii) columns with large coefficients yield a low result. The rationale of this ranking is that, intuitively, the value disjunction for a selection of similar columns should lead to simpler subproblems; also columns with large coefficients should have a larger impact on the rest of the problem than columns with small coefficients.

Example 15. For test instance 4, selecting the variables x_1, x_2, x_3 has the rank $R(\{1,2,3\}) = 0.083$; selecting the variables x_7, x_9, x_{10} has the rank $R(\{7,9,10\}) = 108$.

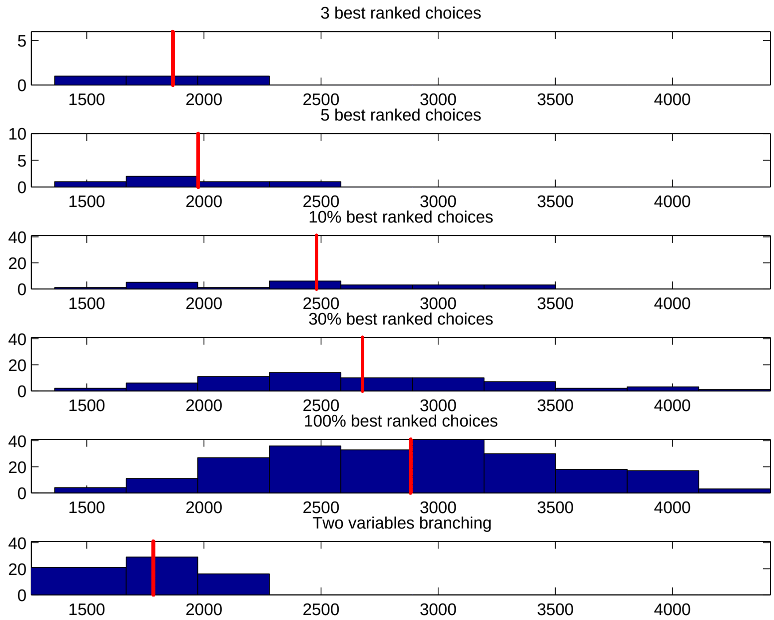

Figure 2: Branching on value disjunctions vs. 2-variable branching (instance 1). The figure shows histograms of the total number of facets in the subproblems; the vertical line is the average.

For all possible branching decisions (i.e., the $\binom{12}{3}$ choices of three variables), we now computed the rank and the complete description size. We grouped the branching decisions according to their rank into sets of the 5 best ranked, 10 % best ranked, 30 % best ranked, etc. choices. For each of the test instances, we show histograms of the complete description sizes corresponding to branching decisions within these rankings in Figures 2–5. As a comparison, the bottom part in each figure shows a histogram of the complete description sizes obtained by the $\binom{12}{2}$ possible choices for two-variable branching. In each histogram the vertical line shows the average (arithmetic mean) of the complete description sizes.

From the computational results, we can draw the following conclusions:

1. It is possible to use the rank formula to predict which branching decisions will lead to low complete description sizes.

2. For instances 1 and 2 that do not contain selections of very low rank, two-variable branching performs better than branching on value disjunctions. However, instances 3 and 4 that contain selections of very low rank, it is possible to take branching decisions that are better than two-variable branching decisions by making use of the rank formula.

We have to remark that there is room for improvement of the proposed ranking formula. Clearly it needs to be generalized for blocks of different cardinalities. It would also need adjustment for unequally scaled rows.

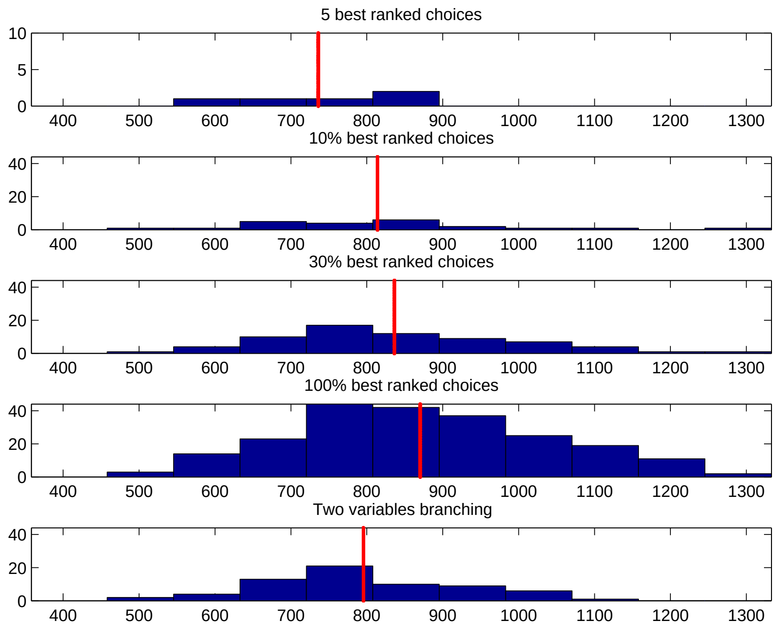

Figure 3: Branching on value disjunctions vs. 2-variable branching (instance 2)

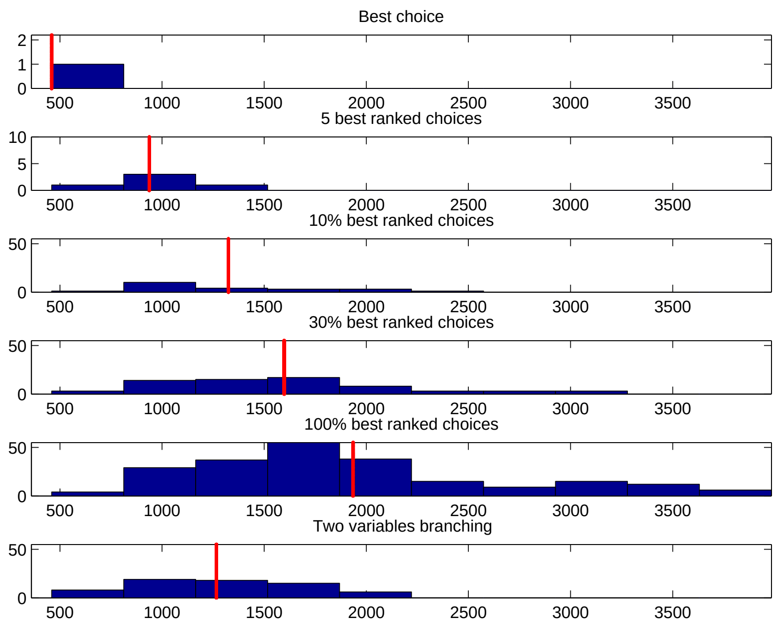

Figure 4: Branching on value disjunctions vs. 2-variable branching (instance 3)

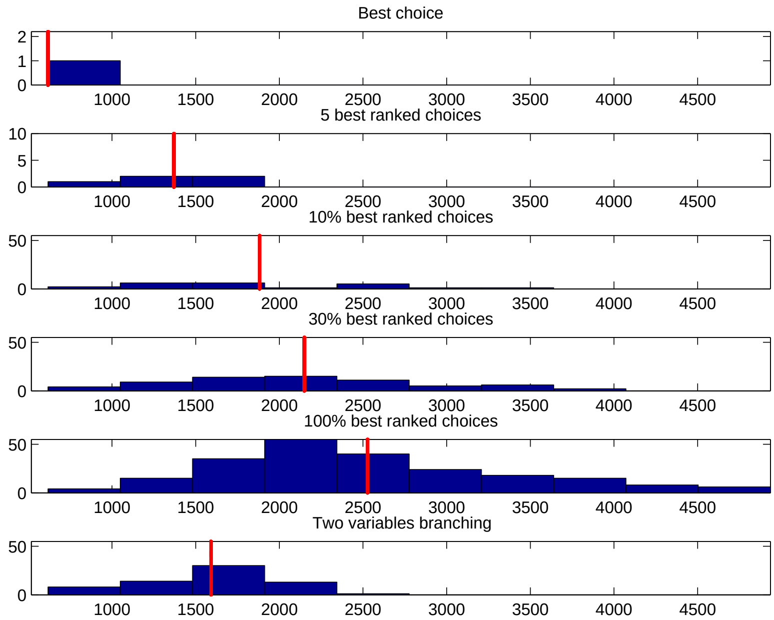

Figure 5: Branching on value disjunctions vs. 2-variable branching (instance 4)

Value-disjunction branching on larger problems. Based on the evidence obtained with the above experiments, we tried to use the new branching scheme to solve larger test problems. Our set of test instances consists of instances with several dense rows (multi-knapsack problems). We focused on problems where the solutions to LP relaxations of subproblem only give little information for taking branching decisions. The test instances are:

- Six randomly generated market split instances with 35 and 40 variables.
- The models `mas74` and `mas76` from the MIPLIB.

It seems difficult to apply Theorem 8 directly to these problems. The reason is that typically many constraints in a model are present. In this case the probability that we can come up with a block decomposition such that some values repeat, is quite low. Hence, one may expect that in such cases the value-reformulation requires to introduce as many variables as we have subsets in each of the elements of the partition $N_1, \ldots, N_K$. Therefore, we decided to perform the following steps:

1. We consider one of the dense rows at a time. We add a relaxation of this row that we obtain by replacing the coefficients by simpler ones. From the row
$$\sum_{i=1}^{n} a_i x_i + \sum_{j=1}^{d} g_j w_j \le b,$$
we generate the relaxation
$$\sum_{i=1}^{n} f(a_i) x_i \le M,$$

Table 7: Branching on value disjunctions for the market split and mas instances. Computation times are given in CPU seconds on a Sun Fire V890 with 1200 MHz UltraSPARC-IV processors

			CPLEX 9.1		Value Disjunctions	
Name	Rows	Cols	Nodes (10^6)	Time (s)	Nodes (10^6)	Time (s)
corn535-1	5	40	13.8	2 431	3.8	809
corn535-2	5	40	11.9	2 084	4.2	865
corn535-3	5	40	17	2 946	9.8	1 970
corn540-4	5	45	321	55 918	105	20 873
corn540-5	5	45	231	39 787	87	17 267
corn540-6	5	45	188	30 532	97	19 162
mas74	13	151	4.4	2 463	1.2	1 194
mas76	12	151	0.667	289	0.063	35

where $f(x)$ is a non-linear function of the type

$$f(x) = \begin{cases} 1 & \text{if } x \geq U \\ 0 & \text{if } L < x < U \\ -1 & \text{if } x \leq L. \end{cases}$$

2. We reformulate the problem using a value disjunction for each of the new rows separately. Because of the simple structure of the coefficients of the new rows, only a linear number of variables is added.

3. Finally, we manually perform SOS branching on the new variables. Then we solve each of the subproblems with the standard branch-and-cut system CPLEX 9.1 [11] using the default settings of the Callable Library. We use the optimal solution value from a subproblem as a primal bound for the remaining subproblems.

The results of this approach on the set of test instances are shown in Table 7. It can be seen that the approach provides a clear gain on all these instances. Both the number of nodes and the computation times are reduced in comparison to the performance of CPLEX 9.1 (with the default settings of the Callable Library) on the original problem.

Acknowledgment. The authors wish to thank the anonymous referees for their helpful comments.

References

[1] Karen Aardal, Robert E. Bixby, Cor A. J. Hurkens, Arjen K. Lenstra, and Job W. Smeltink, *Market split and basis reduction: towards a solution of the Cornuéjols-Dawande instances*, INFORMS J. Comput. **12** (2000), no. 3, 192–202.

[2] Karen Aardal, Cor A. J. Hurkens, and Arjen K. Lenstra, *Solving a system of diophantine equations with lower and upper bounds on the variables*, Mathematics of Operations Research **25** (2000), 427–442.

[3] Egon Balas, *Intersection cuts – a new type of cutting planes for integer programming*, Operations Research **19** (1971), 19–39.

[4] ______, *Disjunctive programming: Cutting planes from logical conditions*, Nonlinear Programming 2 (O. L. Mangasarian et al., ed.), Academic Press, London, 1975, pp. 279–312.

[5] Egon Balas, Sebastián Ceria, and Gérard Cornuéjols, *A lift-and-project cutting plane algorithm for mixed* $0-1$ *programs*, Mathematical Programming **58** (1993), 295–324.

[6] Daniel Bienstock and Mark Zuckerberg, *Subset algebra lift operators for 0-1 programming*, Tech. Report 1, CORC, 2002.

[7] S. Chopra and M. R. Rao, *The Steiner tree problem I: Formulations, compositions and extension of facets*, Mathematical Programming **64** (1994), no. 2, 209–229.

[8] ______, *The Steiner tree problem II: Properties and classes of facets*, Mathematical Programming **64** (1994), no. 2, 231–246.

[9] Thomas Christof and Andreas Löbel, *PORTA – polyhedron representation transformation algorithm*, available electronically from URL http://www.zib.de/Optimization/Software/Porta/.

[10] Gérard Cornuéjols and Milind Dawande, *A class of hard small* 0-1 *programs*, Integer programming and combinatorial optimization (Houston, TX, 1998), Lecture Notes in Computer Science, vol. 1412, Springer, Berlin, 1998, pp. 284–293. MR 2000h:90042

[11] ILOG, *CPLEX*, 1997–2004, http://www.ilog.com/products/cplex/.

[12] Miroslav Karamanov and Gérard Cornuéjols, *Branching on general disjunctions*, Manuscript, 2005.

[13] J. Krarup and O. Bilde, *Plant location, set covering and economic lot sizes: an* $O(mn)$ *algorithm for structured problems*, Optimierung bei graphentheoretischen und ganzzahligen Problemen (L. Collatz et al., ed.), Birkhäuser-Verlag, Basel, 1977, pp. 155–180.

[14] Monique Laurent, *A comparison of the Sherali–Adams, Lovász–Schrijver and Lasserre relaxations for 0-1 programming*, Mathematics of Operations Research **28** (2003), no. 3, 470–496.

[15] Quentin Louveaux and Laurence A. Wolsey, *Combining problem structure with basis reduction to solve a class of hard integer programs*, Mathematics of Operations Research **27** (2002), no. 3, 470–484.

[16] László Lovász and Alexander Schrijver, *Cones of matrices and set-functions and* 0–1 *optimization*, SIAM Journal on Optimization **1** (1991), 166–190.

[17] R. Kipp Martin, *Generating alternative mixed-integer programming models using variable redefinition*, Operations Research **35** (1987), 331–359.

[18] ______, *Using separation algorithms to generate mixed integer model reformulations*, Operations Research Letters **10** (1991), 119–128.

[19] R. L. Rardin and U. Choe, *Tighter relaxations of fixed charge network flow problems*, report J-79-18, Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia, 1979.

[20] Hanif D. Sherali and Warren P. Adams, *A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems*, SIAM Journal of Discrete Mathematics **3** (1990), 411–430.

[21] Hanif D. Sherali and J. Cole Smith, *Improving discrete model representations via symmetry considerations*, Management Science **47** (2001), no. 10, 1396–1407.

[22] Robert Weismantel, *Hilbert bases and the facets of special knapsack polytopes*, Mathematics of Operations Research **21** (1996), 886–904.