Extended Formulations for Gomory Corner Polyhedra

Matthias Köppe [a], Quentin Louveaux [b], Robert Weismantel [a], Laurence A. Wolsey [b]

[a] *Otto-von-Guericke-Universität Magdeburg, Department of Mathematics/IMO, Universitätsplatz 2, 39106 Magdeburg, Germany*

[b] *CORE, Université catholique de Louvain, Voie du Roman Pays 34, 1348 Louvain-la-Neuve, Belgium*

Abstract

We present several types of extended formulations for integer programs, based on irreducible integer solutions to Gomory's group relaxations. We present algorithmic schemes based on an iterative reformulation technique using these extended formulations. We give computational results for benchmark problems, which illustrate the primal and dual effect of the reformulation.

Key words: Integer programming, Gomory group relaxation, extended formulations, Integral Basis Method
1991 MSC: 90C10

1 Introduction

More than 30 years ago, Ralph Gomory [3] introduced an approach to solve integer linear programs based on group relaxations. This approach was later refined by Gomory and Johnson [4,5] leading to a theory of valid inequalities and duality based on superadditivity [6].

Though theoretically very fruitful, little computational work was reported using this approach. The principal study was that of Gorry, Northup, and

Email addresses: `mkoeppe@imo.math.uni-magdeburg.de` (Matthias Köppe), `louveaux@core.ucl.ac.be` (Quentin Louveaux), `weismant@imo.math.uni-magdeburg.de` (Robert Weismantel), `wolsey@core.ucl.ac.be` (Laurence A. Wolsey).

Shapiro [9]. They developed a branch-and-bound code in which the bounds were obtained by solving relaxed group problems by dynamic programming. The major obstacle encountered was the huge size of the groups arising in practice (basis determinants of 10^8 or 10^{10}) while the group problems that could be solved by dynamic programming were of the order of 10^4.

The central starting point of this line of research is the Gomory corner polyhedron that may be derived from an integer programming problem as follows. Starting with a linear integer program

$$\begin{aligned} &\max\ c^T x \\ &\text{s.t.}\ \ Ax = b \\ &\qquad\ \ x \in \mathbb{Z}^{m+n}_+, \end{aligned} \tag{1}$$

one may select a subset of linearly independent columns A_B and relax the integer program by

$$S = \Big\{\, x \in \mathbb{Z}^{m+n} : A_B x_B + A_{\bar B} x_{\bar B} = b,\ x_{\bar B} \geq 0 \,\Big\}, \tag{2}$$

that is, one ignores the nonnegativity requirement for the variables in the index set B but maintains integrality of all the variables and the nonnegativity conditions for the variables in $\bar B$, the complementary set of B. The optimization problem over the set S is called a *group relaxation* of the integer program (1).

By the linear independence of A_B, the set S is in bijection with its projection to the space of complementary variables

$$S' = \Big\{\, x_{\bar B} \in \mathbb{Z}^{|\bar B|} : A_{\bar B} x_{\bar B} \equiv b \pmod{L(A_B)},\ x_{\bar B} \geq 0 \,\Big\},$$

where $L(A_B) = \{\, A_B z :\ z \in \mathbb{Z}^{|B|} \,\}$ denotes the lattice generated by the column vectors of A_B.

The convex hull of all points in S' is called the *Gomory corner polyhedron*. Gomory and coauthors [7,8,2] claim that although the corner polyhedron seems to be a quite general mathematical object, its geometry might be substantially easier to describe than that of the original integer programming polyhedron. The key is to consider the lattice $L(A_B)$ as a subgroup of $\mathbb{Z}^{|B|}$ and to study the factor group $\mathbb{Z}^{|B|}/L(A_B)$. The elements of this factor group are the cosets $x_B + L(A_B)$ of the group $L(A_B)$, i.e., the equivalence classes of vectors of $\mathbb{Z}^{|B|}$ whose differences are vectors of the lattice $L(A_B)$. In particular, group automorphisms of the abelian group $\mathbb{Z}^{|B|}/L(A_B)$ or its cyclic subgroups can be used to make connections between different facets of the corner polyhedron. In a similar vein, composition rules can be used to "lift" facets of corner polyhedra associated with small groups to corner polyhedra associated with bigger groups. All these results certainly provide theoretical evidence that Gomory corner polyhedra ought to be investigated in more depth. Recently Gomory, Johnson, and Evans [8, page 338, section 3] wrote:

> “If we were able to come close to solving the Corner Polyhedron – say by having an adequate supply of cutting planes or perhaps, in other ways, such as finding solutions to the group problems, we could come close to a different kind of algorithm – one based on solving a sequence of Corner Polyhedron problems. . . ”

A very different line of research has been taken recently by Haus, Köppe, and Weismantel [10]. Their Integral Basis Method is a primal simplex-type algorithm based on a sequence of feasible all-integer tableaux of the following form:

$$X = \left\{ x_{\bar B} \in \mathbb{Z}^{|\bar B|} : \bar A_{\bar B} x_{\bar B} \le \bar b,\ x_{\bar B} \ge 0 \right\}$$

Rather than add cutting planes, they use an *extended formulation* of the problem involving additional variables. The extended formulation is typically obtained by computing the irreducible (minimal) solutions to a *knapsack relaxation* of the integer program. The knapsack relaxation is obtained by taking a single row of an all-integral tableau corresponding to the current feasible solution,

$$\sum_{j \in \bar B} \bar A_{ij} x_j \le \bar b_i$$

and relaxing it to

$$\bar A_{i1} x_1 + \sum_{j \in N_-} \bar A_{ij} x_j \le \bar b_i \tag{3}$$

where $\bar A_{i1} > \bar b_i \ge 0$ and $\bar A_{ij} < 0$ for $j \in N_- \subset \bar B$. The irreducible integer solutions to this inequality are used to replace the integer program by an extended formulation where the distinguished variable x_1 is replaced by a set of variables, each corresponding to an irreducible solution of the knapsack relaxation. Let T be a matrix whose columns are the irreducible solutions to equation (3). Then the extended formulation is given by:

$$X = \{\, x_{\bar B} \in \mathbb{Z}_+^{|\bar B|} : x_{\bar B} = Ty,\ \bar A T y \le \bar b,\ y \in \mathbb{Z}_+^s \,\}.$$

This reformulation has two possible advantages. The main objective is to find an augmenting vector allowing one to move to an improved primal feasible solution, and the columns of T or simple integer combinations of these columns should provide good candidates for augmentation. The second advantage is that the LP relaxation of the extended formulation can be stronger than that of the original formulation.

The objective of this paper is to study the possible combination of these two approaches. Specifically,

(i) it appears natural to consider extended formulations based on a group relaxation as an alternative to the extended formulations based on knapsack reformulations that are used in the Integral Basis Method;

(ii) extended formulations for the group problem provide an alternative to the use of cutting planes if one wishes to test their effectiveness in tightening dual LP bounds.

The algorithmic scheme that we propose works in an iterative fashion. Each iteration requires the analysis of a group relaxation of the original integer program. The key step is to define an extended formulation for such a group relaxation whose variables are in one-to-one correspondence with the irreducible solutions of the group relaxation.

More precisely, the general model that we study in this paper is

$$Y(f) = \Big\{ x \in \mathbb{Z}^n_+ : Fx \equiv f \pmod{L(\Delta)} \Big\},$$

where Δ is a regular diagonal positive integer matrix, and $F \in \mathbb{Z}^{r \times n}$. We will show in section 2, how this model arises from the group relaxation (2). As before, the convex hull of the points in $Y(f)$ defines a Gomory corner polyhedron. An *extended formulation* of $Y(f)$ is a representation of the form

$$Y(f) = \Big\{ x \in \mathbb{Z}^n_+ : x = Ty,\ Uy = d,\ y \in \mathbb{Z}^p_+ \Big\}.$$

Of particular relevance are reformulations whose linear relaxations produce the convex hull, i.e.,

$$\mathrm{conv}(Y(f)) = \Big\{ x \in \mathbb{R}^n_+ : x = Ty,\ Uy = d,\ y \in \mathbb{R}^p_+ \Big\}.$$

In this case, we say that the formulation has the *convex hull property.*

In the following sections, we examine such sets. In Section 2 we present four straightforward (but probably not well-known) extended formulations for the group relaxation. The first one uses all the irreducible solutions of the group problem. We call it the *disaggregated formulation.* The second one is a first generalization where we reduce the number of new variables by aggregating variables with identical residue class. We call it the *aggregated formulation.* The third reformulation is based on an *advanced aggregation* technique that reduces even further the number of new variables. Finally we present a reformulation, related to [13], based on the representation of groups by paths in a digraph.

Section 3 analyzes the different formulations. We show that the disaggregated, the aggregated and the path reformulation satisfy the convex hull property. We also show how to generate valid inequalities that tighten the advanced aggregation formulation. In Section 4 we address the question of how to generate the complete sets of irreducible solutions used in the extended formulations proposed in Section 2. In practice we can compute these sets for groups up to order 30. We also show how for composite groups $G = G_1 \times G_2$, it is possible to generate the irreducibles for G from those of G_1 and G_2. In Section 5 we outline possible algorithmic schemes based on these extended formulations. In Section 6 we report on our computational experience. Essentially though our algorithmic schemes are very different, the conclusions are similar to those in earlier studies,

(1) working with small subgroups is insufficient to take one close to an optimal solution whether in primal or dual mode;
(2) unbounded group problems do not capture the 0-1 nature of many practical IPs.

2 Extended Formulations for Group Relaxations

We consider an integer program

$$\begin{aligned} &\max\ c^T x \\ &\text{s.t.}\ \ Ax = b \\ &\qquad\ x \in \mathbb{Z}_+^{m+n}. \end{aligned} \tag{4}$$

with $A \in \mathbb{Z}^{m\times(m+n)}$, $c \in \mathbb{Z}^{m+n}$, and $b \in \mathbb{Z}^m$. Let B be the index set of a simplex basis for the LP relaxation of (4) and let $\bar{B} = \{1, \dots, m+n\} \setminus B$ denote the complementary index set. Then the feasible region can be written in the form

$$\{x \in \mathbb{Z}_+^{m+n} : x_B + A_B^{-1} A_{\bar{B}} x_{\bar{B}} = A_B^{-1} b\}.$$

Let $\Delta \in \mathbb{Z}^{m\times m}$ be a diagonal matrix such that $\Delta A_B^{-1} A_{\bar{B}}$ and $\Delta A_B^{-1} b$ are integral. Then we obtain the equivalent formulation

$$\{x \in \mathbb{Z}_+^{m+n} : \Delta x_B + \Delta A_B^{-1} A_{\bar{B}} x_{\bar{B}} = \Delta A_B^{-1} b\}.$$

We investigate the group relaxation

$$Y^G = \{x \in \mathbb{Z}_+^n : \Delta A_B^{-1} A_{\bar{B}} x_{\bar{B}} \equiv \Delta A_B^{-1} b \ (\mathrm{mod}\ L(\Delta))\}. \tag{5}$$

Setting $F = \Delta A_B^{-1} A_{\bar{B}}$ and $f = \Delta A_B^{-1} b$, this is of the general form

$$Y(f) = \{x \in \mathbb{Z}_+^n : Fx \equiv f \ (\mathrm{mod}\ L(\Delta))\}, \tag{6}$$

where $\Delta \in \mathbb{Z}^{r\times r}$ is a regular diagonal positive integer matrix, $F \in \mathbb{Z}^{r\times n}$, and $f \in \mathbb{Z}_+^r$. Associated with $Y(f)$ is the abelian group

$$\begin{aligned} G &= \{y \in \mathbb{Z}_+^r : y_i \in \{0, \dots, \Delta_i - 1\}, i = 1, \dots, r\} \\ &= (\mathbb{Z}/\Delta_1\mathbb{Z}) \times \cdots \times (\mathbb{Z}/\Delta_r\mathbb{Z}), \end{aligned}$$

consisting of all residue classes of $\mathbb{Z}^r$ modulo the vector diag(Δ). For a set $Y(f)$, we now examine several possible extended formulations. To do this we need to introduce the notion of *irreducible* solutions.

Definition 1. **(a)** A vector $x \in \mathbb{Z}_+^n$ is an *irreducible solution* of $Y(f)$ if $x \in Y(f)$, and there is no other distinct non-zero $\tilde{x} \in Y(f)$ with $\tilde{x} \leq x$.
(b) A vector x in $Y(0)$ is called a *homogeneous* solution. A vector x in $Y(f)$ is called an *inhomogeneous* solution whenever $f \neq 0$.

An important property is that every integer point in $Y(f)$ can be represented as the sum of exactly one inhomogeneous irreducible solution of $Y(f)$ and a nonnegative integer combination of the homogeneous irreducible solutions of $Y(0)$.

2.1 Disaggregated formulation

In order to come up with a first reformulation, we determine a matrix $C \in \mathbb{Z}^{n\times s}$ whose column vectors correspond to all inhomogeneous irreducible solutions of $Y(f)$. In the same way we introduce a matrix $D \in \mathbb{Z}^{n\times t}$ whose column vectors are all the homogeneous irreducible solutions of $Y(0)$. We associate integer λ and μ variables with the columns of C and D, respectively, to define the first formulation.

Proposition 2.

$$Y(f) = \{x \in \mathbb{R}_+^n : x = C\lambda + D\mu,\ 1^T\lambda = 1,\ \lambda \in \mathbb{Z}_+^s, \mu \in \mathbb{Z}_+^t\}.$$

Therefore the right-hand side of the equation is a valid extended formulation for $Y(f)$ referred to as the disaggregated formulation*.*

Example 3. Consider the set $X = \{x_1, x_2, x_3 \in \mathbb{Z}_+ : 3x_1 + 7x_2 + 9x_3 = 22\}$. By taking the equation (mod 4), we obtain the valid group relaxation

$$Y(2) = \{x_1, x_2, x_3 \in \mathbb{Z}_+ : 3x_1 + 3x_2 + x_3 \equiv 2 \pmod 4\}. \tag{7}$$

The inhomogeneous irreducible solutions of $Y(2)$ are represented in the matrix

$$C = \begin{pmatrix} 2 & 1 & 0 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}.$$

The homogeneous irreducible solutions of $Y(0)$ are represented in the matrix

$$D = \begin{pmatrix} 1 & 0 & 4 & 3 & 2 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 2 & 3 & 4 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 4 \end{pmatrix}.$$

We refer to Section 4 for more details on how to compute these matrices of irreducibles. A valid reformulation for $Y(2)$ is thus to associate a new variable to each irreducible solution and hence write

$$\begin{aligned} Y(2) = \{x \in \mathbb{Z}^3_+ : & \\ & \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 & 1 & 0 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_4 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 4 & 3 & 2 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 2 & 3 & 4 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 4 \end{pmatrix} \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_8 \end{pmatrix}, \\ & \lambda_1 + \cdots + \lambda_4 = 1 \\ & \lambda_1, \ldots, \lambda_4 \in \{0, 1\},\ \mu_1, \ldots, \mu_8 \in \mathbb{Z}_+ \qquad \}. \end{aligned}$$

2.2 Aggregated formulation

Inspecting the irreducible solutions in detail suggests that we can classify some of the irreducible solutions according to the value given by the sum of the first and second component. For instance, all the vectors

$$\begin{pmatrix} 4 & 3 & 2 & 1 & 0 \\ 0 & 1 & 2 & 3 & 4 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

have the property that the sum of the two first components is 4. This property can be highlighted for some other sets of vectors of C and D and is due to the fact that x_1 and x_2 have the same coefficient in the group equation of (7). This suggests the idea of aggregating these two variables into one variable w so as to reduce the size of the reformulation. We now let $N = \{1, \ldots, n\}$ and $N_\alpha = \{j \in N : F_{:j} \equiv \alpha \pmod{L(\Delta)}\}$, where $\alpha \in G$ and $F_{:j}$ denotes the j^{th} column of F. We aggregate the variables with the same coefficients into a new variable $w_\alpha = \sum_{j \in N(\alpha)} x_j$ and consider the set

$$W(f) = \left\{ w \in \mathbb{Z}^{|\tilde{G}|}_+ : \sum_{\alpha \in \tilde{G}} \alpha w_\alpha \equiv f \pmod{L(\Delta)} \right\},$$

where $\tilde{G} = \{\alpha \in G : \text{there exists } j \text{ with } F_{:j} \equiv \alpha \pmod{L(\Delta)}\}$. We denote by $\tilde{C} \in \mathbb{Z}^{|\tilde{G}| \times \tilde{s}}$ and $\tilde{D} \in \mathbb{Z}^{|\tilde{G}| \times \tilde{t}}$ the matrices whose columns are the inhomogeneous and homogeneous irreducibles of $W(f)$ and $W(0)$ respectively. Now we are able to write a second formulation for $Y(f)$.

Proposition 4.

$$Y(f) = \{x \in \mathbb{Z}_+^n : w = \tilde{C}\lambda + \tilde{D}\mu,\ 1^T\lambda = 1,\ \lambda \in \mathbb{Z}_+^{\tilde{s}},\ \mu \in \mathbb{Z}_+^{\tilde{t}},$$
$$w_\alpha = \textstyle\sum_{j \in N(\alpha)} x_j,\ w \in \mathbb{Z}_+^{|\tilde{G}|}\}.$$

Therefore the right-hand side of the equation is a valid extended formulation for $Y(f)$ referred to as the aggregated formulation.

Example 5. Consider again the group relaxation (7),

$$Y(2) = \{x \in \mathbb{Z}_+^3 : 3x_1 + 3x_2 + x_3 \equiv 2 \pmod 4\}.$$

We aggregate the first two variables in $w_3 = x_1 + x_2$ and substitute $w_1 = x_3$. We now consider an aggregated version of $Y(2)$,

$$W(2) = \{(w_3, w_1) \in \mathbb{Z}_+^2 : 3w_3 + w_1 \equiv 2 \pmod 4\}.$$

The correponding matrices of irreducibles of $W(2)$ and $W(0)$ are

$$\tilde{C} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \quad \text{and} \quad \tilde{D} = \begin{pmatrix} 1 & 4 & 0 \\ 1 & 0 & 4 \end{pmatrix}.$$

The aggregated reformulation of $Y(2)$ is

$$Y(2) = \{x \in \mathbb{Z}_+^3 : x_1 + x_2 = w_3$$
$$\begin{pmatrix} w_3 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} + \begin{pmatrix} 1 & 4 & 0 \\ 1 & 0 & 4 \end{pmatrix} \begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{pmatrix}$$
$$\lambda_1 + \lambda_2 = 1,\ \lambda \in \mathbb{Z}_+^2,\ \mu \in \mathbb{Z}_+^3,\ w \in \mathbb{Z}_+\}.$$

Compared to the disaggregated formulation presented earlier, the aggregated formulation is much more compact.

2.3 Advanced aggregation

One way of further reducing the size of the reformulation of $Y(f)$ consists of generalizing our aggregation technique to variables with different coefficients. We study here the case of a recursive aggregation of variables whose residue classes are integer multiples of each other. Consider any ordering of the variables. Without loss of generality, we assume that $\{1, \ldots, n\}$ reflects the ordering. We partition $L = \{1, \ldots, n\}$ iteratively using the algorithm of

Table 1
Iteratives definition of an advanced aggregation

1. **Initialize** $L := \{1, \ldots, n\},\ I := \emptyset$.
2. **While** $L \neq \emptyset$ **do**
3. **Let** i be the first list element of L.
4. **Set** $I := I \cup \{i\}$.
5. Define a number $u_i \in \mathbb{Z}_+ \setminus \{0\}$ and a subset $L'(i) \subseteq L$.
6. **Set** $N(i) := \{j \in L'(i) : \exists \lambda \in \mathbb{Z}_+, \lambda \leq u_i$ with $F_{:j} \equiv \lambda F_{:i} \pmod{L(\Delta)}\}$.
7. **Update** $L := L \setminus N(i)$.
8. **Return** I and $N(i)$ for $i \in I$.

Table 1. We remark that step 5 of the algorithm leaves the selection rules for the number u_i and the subsets $L'(i)$ unspecified; we will discuss them later.

The algorithm computes a set I and for every $i \in I$ a set $N(i)$ of variable indices. For every $i \in I$, we define a new variable z_i that collects all integer variables x_j with $j \in N(i)$. To define the coefficients of every variable in the aggregation, we denote by h_j the smallest positive integer such that $F_{:j} \equiv h_j F_{:i} \pmod{L(\Delta)}$. The "aggregated" variable is then

$$z_i = \sum_{j \in N(i)} h_j x_j.$$

Using the z variables we can define a group relaxation

$$Z(f) = \left\{ z \in \mathbb{Z}_+^{|I|} : \sum_{i \in I} F_{:i} z_i \equiv f \pmod{L(\Delta)} \right\}.$$

We denote by $\hat{C} \in \mathbb{Z}^{|I| \times \hat{s}}$ and $\hat{D} \in \mathbb{Z}^{|I| \times \hat{t}}$ the matrices whose columns are the inhomogeneous and homogeneous irreducibles of $Z(f)$ and $Z(0)$ respectively. We are now able to present a third formulation for $Y(f)$.

Proposition 6.

$$\begin{aligned} Y(f) = \{x \in \mathbb{Z}_+^n : {} & z = \hat{C}\lambda + \hat{D}\mu,\ 1^T\lambda = 1,\ \lambda \in \mathbb{Z}_+^{\hat{s}},\ \mu \in \mathbb{Z}_+^{\hat{t}}, \\ & z_i = \textstyle\sum_{j \in N(i)} h_j x_j, \quad \text{for all } i \in I \\ & z \in \mathbb{Z}_+^{|I|}\}. \end{aligned} \tag{8}$$

Therefore the right hand side of (8) *is a valid extended formulation of* $Y(f)$*. We refer to it as the* advanced aggregation reformulation*.*

The advanced aggregation formulation is quite general and includes, in particular, the special cases discussed earlier. If we define $L'(i) = \{i\}$ and $u_i = 1$ in step (5) for all $i \in \{1, \ldots, n\}$, then the advanced aggregation formulation coincides with the disaggregated formulation. Similarly setting $L'(i) = L$ and $u_i = 1$ for all $i \in I$ in step (5) yields the aggregated formulation. A natural generalization of the latter is obtained by setting $u_i = 2$ or $u_i = 3$ in step (5), respectively. We will show in Section 3 that we can analyze the strength of the corresponding extended formulation and enrich it by linear inequalities so as to satisfy the convex hull property.

Example 7. We study the set

$$X = \Big\{ x \in \mathbb{Z}_+^5 : \quad x_1 + 7x_2 + 9x_3 + 8x_4 + 6x_5 = 22 \\ 3x_1 + 2x_2 + 6x_3 + 4x_4 + 7x_5 = 13 \Big\},$$

and a corresponding group relaxation

$$Y\begin{pmatrix}2\\1\end{pmatrix} = \Bigg\{ x \in \mathbb{Z}_+^5 : \\ \begin{pmatrix}1\\3\end{pmatrix} x_1 + \begin{pmatrix}2\\2\end{pmatrix} x_2 + \begin{pmatrix}4\\2\end{pmatrix} x_3 + \begin{pmatrix}3\\0\end{pmatrix} x_4 + \begin{pmatrix}1\\3\end{pmatrix} x_5 \equiv \begin{pmatrix}2\\1\end{pmatrix} \mod \begin{pmatrix}5\\4\end{pmatrix} \Bigg\}.$$

The aggregation is obtained as follows. We first set $L = \{1, \ldots, 5\}$ and $I = \emptyset$. For the first iteration, we choose $L'(1) = \{2, \ldots, 5\}$ and $u_1 = 3$. We note that the residue class of x_2 is twice the residue class of x_1. We put them together in the same aggregation. Therefore, $N(1) = \{1, 2\}$. Now $i = 3$ and with the parameters $L'(3) = \{4, 5\}$ and $u_3 = 3$, we note that the residue class of x_4 is twice that of x_3. We also observe that x_2 could be included in the same aggregation but as it is already in $N(1)$, we do not consider it here. Hence $N(3) = \{3, 4\}$. Finally, the last step provides $N(5) = \{5\}$ and $I = \{1, 3, 5\}$. The advanced aggregation provides the three variables $z_1 = x_1 + 2x_2$, $z_3 = x_3 + 2x_4$ and $z_5 = x_5$. Now to write the advanced aggregation formulation, we need to compute the irreducible solutions of

$$Z\begin{pmatrix}2\\1\end{pmatrix} = \Bigg\{ z \in \mathbb{Z}_+^3 : \begin{pmatrix}1\\3\end{pmatrix} z_1 + \begin{pmatrix}4\\2\end{pmatrix} z_3 + \begin{pmatrix}1\\3\end{pmatrix} z_5 \equiv \begin{pmatrix}2\\1\end{pmatrix} \mod \begin{pmatrix}5\\4\end{pmatrix} \Bigg\},$$

which we do not do in this small example since it would involve too many vectors.

2.4 Path reformulation

Finally we present a fourth reformulation that is based on the "path" structure of the group equation [13]. Let (V, A) be a digraph with $|G|$ nodes corresponding to each element of the group G, and arcs $(\alpha, \alpha + F_{:j} \pmod{L(\Delta)})$

for all $\alpha \in G$ and $j \in N$. Figure 1 shows such a graph related to the $x_1 + 2x_2 \equiv 2 \pmod 4$ group problem. The solid arcs (above the nodes) correspond to x_1 while the dotted arcs (below the nodes) correspond to $2x_2$. In the figure, any solution to the group problem corresponds to a walk from 0 to 2, or equivalently to the pushing of one unit of flow from 0 to 2.

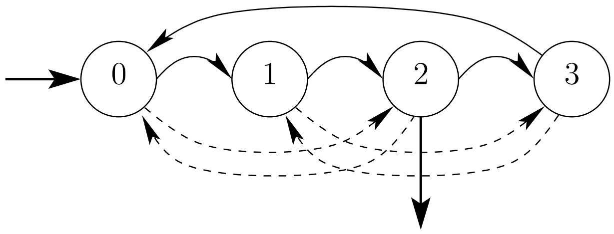

Fig. 1. The path representation of $x_1 + 2x_2 \equiv 2 \pmod 4,\ x \in \mathbb{Z}^2_+$.

Now if we go back to the general case, any walk from 0 to f corresponds to a point in $Y(f)$. Specifically let w^j_α be the number of times the arc $(\alpha, \alpha + F_{:j})$ occurs in the walk. Then

$$\sum_{\alpha \in G} \sum_{j \in N} F_{:j} w^j_\alpha \equiv f \pmod{L(\Delta)}.$$

We can now formulate the group problem by flow constraints on each node of the graph, with a flow of 1 coming into node 0 and a flow of 1 going out of f, the new variables being the flow variables w.

Proposition 8.

$$
\begin{aligned}
Y(f) = \{x \in \mathbb{Z}^n_+ : x_j = &\sum_{\alpha \in G} w^j_\alpha \\
&\sum_{j \in N} w^j_0 - \sum_{j \in N} w^j_{\Delta - F_{:j}} = 1 \\
&\sum_{j \in N} w^j_\beta - \sum_{j \in N} w^j_{\beta - F_{:j}} = 0 \qquad \textit{for } \beta \in G \setminus \{0, f\} \\
&\sum_{j \in N} w^j_f - \sum_{j \in N} w^j_{f - F_{:j}} = -1 \\
&w \in \mathbb{Z}^{n|G|}_+ \qquad\qquad \},
\end{aligned}
$$

The right hand side of the equation above is a valid extended formulation of $Y(f)$ referred to as the path reformulation *of $Y(f)$.*

Example 9. Let us consider the following knapsack

$$x_1 + 5x_2 - 2x_3 = 3$$

and a corresponding group relaxation

$$Y(3) = \{x \in \mathbb{Z}^3_+ : x_1 + x_2 + 2x_3 \equiv 3 \pmod 4\}.$$

Table 2
The sizes of the different formulations

	Irreducibles		
Formulation	Homogeneous	Inhomogeneous	Variables
Disaggregated	378	76	454
Aggregated	54	26	80
Advanced aggregated	13	8	21
Path			77

The path approach provides the following extended formulation

$$
\begin{aligned}
Y(3) = \{\, x \in \mathbb{Z}^3_+ : & x_1 = w^1_0 + w^1_1 + w^1_2 + w^1_3 \\
& x_2 = w^2_0 + w^2_1 + w^2_2 + w^2_3 \\
& x_3 = w^3_0 + w^3_1 + w^3_2 + w^3_3 \\
& w^1_0 + w^2_0 + w^3_0 - w^1_3 - w^2_3 - w^3_2 = 1 \\
& w^1_1 + w^2_1 + w^3_1 - w^1_0 - w^2_0 - w^3_3 = 0 \\
& w^1_2 + w^2_2 + w^3_2 - w^1_1 - w^2_1 - w^3_0 = 0 \\
& w^1_3 + w^2_3 + w^3_3 - w^1_2 - w^2_2 - w^3_1 = -1 \\
& w \in \mathbb{Z}^{12}_+ \qquad \}.
\end{aligned}
$$

A solution $w^2_0 = w^3_1 = 1$ corresponds to $x_2 = x_3 = 1$.

Example 10. The following example illustrates the sizes of the different formulations. We consider the single row group problem

$$3x_1 + 3x_2 + 3x_3 + 6x_4 + 5x_5 + 10x_6 + 7x_7 \equiv 1 \pmod{11}.$$

Table 2 shows the number of new variables for each of the different formulations. We remark that the corner polyhedron corresponding to this group problem has 18 nontrivial facets.

3 Analysis of the reformulations

In this section we analyze the four formulations presented in the previous section. We focus on showing under which conditions the formulations have the convex hull property, i.e., under which conditions the linear relaxation of each formulation produces the convex hull of feasible integer points. We denote by P^1_Y the corresponding polyhedron when the integrality requirement of λ and μ is dropped in the disaggregated reformulation. Similarly we denote by P^2_Y, P^3_Y, P^4_Y the polyhedra corresponding to the aggregated, the advanced aggregation and the path reformulation respectively, when all the integrality constraints are dropped.

An important result is that the convex hull property presented in Section 1 holds for three among the four reformulations, namely for the disaggregated, the aggregated and the path reformulation. To prove this result, two intermediate propositions are needed.

Proposition 11. *For $f \neq 0$, every extreme point of* $\mathrm{conv}(Y(f))$ *is an irreducible inhomogeneous solution.*

Proof: Let y be a vertex (extreme point) of $\mathrm{conv}(Y(f))$. It is obvious that y is an inhomogeneous solution of (6). Let us suppose now that y is reducible. Therefore there exists $y_1 \leq y, y_1 \neq y$, inhomogeneous solution of (6). We also have that $z = y - y_1$ is nonnegative and is an homogeneous solution of the group problem. Hence $y_2 = y + z$ is also a solution of (6) different from y. Furthermore

$$y = \frac{1}{2}y_1 + \frac{1}{2}y_2,$$

contradicting the fact that y is a vertex. □

It can also be noticed that every unit vector e_α, multiplied by the product of all Δ_i, for $i = 1, \ldots, r$ is definitely an homogeneous solution of the problem. Hence there always exists a number $o \leq \prod_{i=1}^{r} \Delta_i$ such that the vector oe_α is an irreducible homogeneous solution. (The number o is the order $o(\alpha)$ of the group element α in the group G.) Therefore, we have:

Proposition 12. *For all $\alpha \in \tilde{G}$, $\prod_{i=1}^{r} \Delta_i e_\alpha \in W(0)$. Therefore* $\mathrm{conv}(W(0)) = \mathbb{R}_+^{|\tilde{G}|}$. *Similarly,* $\mathrm{conv}(Y(0)) = \mathbb{R}_+^{|N|}$.

Based on Propositions 11 and 12, it can be shown that the disaggregated, the aggregated and the path formulation not only model the group problem, but define "ideal formulations" in the sense that they produce the Gomory corner polyhedron, i.e., the convex hull of all the integer points of a group problem.

Theorem 13. $P_Y^1 = P_Y^2 = P_Y^4 = \mathrm{conv}(Y(f))$.

Proof: For P_Y^1, the proof follows immediately from Proposition 11 and 12 as the extreme points are columns of C, and the extreme rays are columns of D.

For P_Y^2, let us fix $x \in P_Y^2$ and show that we also have that $x \in P_Y^1$. The other inclusion is trivial. We have

$$w = \tilde{C}\lambda + \tilde{D}\lambda, \quad 1^T\lambda = 1 \quad \text{and} \quad w_\alpha = \sum_{j \in N(\alpha)} x_j, \quad x, w, \lambda, \mu \geq 0.$$

Let us fix some α such that $w_\alpha \neq 0$, we can write

$$w_\alpha = x_\alpha^1 + \cdots + x_\alpha^{|N(a)|} = \sum_{i=1}^{\tilde{s}} \tilde{C}_{\alpha i}\lambda_i + \sum_{j=1}^{\tilde{t}} \tilde{D}_{\alpha j}\mu_j,$$

where $x_\alpha^1, \ldots, x_\alpha^{|N(\alpha)|}$ represent all the variables included in the set $N(\alpha)$. We also have

$$\begin{pmatrix} x_\alpha^1 \\ \vdots \\ x_\alpha^{|N(\alpha)|} \end{pmatrix} = \sum_{i=1}^{\tilde{s}} \begin{pmatrix} \tilde{C}_{\alpha i} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \lambda_i \frac{x_\alpha^1}{w_\alpha} + \cdots + \sum_{i=1}^{\tilde{s}} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \tilde{C}_{\alpha i} \end{pmatrix} \lambda_i \frac{x_\alpha^{|N(\alpha)|}}{w_\alpha}$$
$$+ \sum_{j=1}^{\tilde{t}} \begin{pmatrix} \tilde{D}_{\alpha j} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \mu_j \frac{x_\alpha^1}{w_\alpha} + \cdots + \sum_{j=1}^{\tilde{t}} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \tilde{D}_{\alpha j} \end{pmatrix} \mu_j \frac{x_\alpha^{|N(\alpha)|}}{w_\alpha}.$$

This can be rewritten using the unit vectors e_k as

$$\begin{pmatrix} x_\alpha^1 \\ \vdots \\ x_\alpha^{|N(\alpha)|} \end{pmatrix} = \sum_{k=1}^{|N(\alpha)|} \sum_{i=1}^{\tilde{s}} \tilde{C}_{\alpha i}\lambda_i \frac{x_\alpha^k}{w_\alpha} e_k + \sum_{k=1}^{|N(\alpha)|} \sum_{j=1}^{\tilde{t}} \tilde{D}_{\alpha j}\mu_j \frac{x_\alpha^k}{w_\alpha} e_k. \tag{9}$$

The expression (9) is valid when one fixes some α. We can write such a similar expression for the complete vector x. In the summation, we need to consider all possible combinations of choosing an index $k(\alpha)$ for each $\alpha = 1, \ldots, |G|$. Remark that, to simplify the notation, we consider that $\frac{x_\alpha^k}{w_\alpha} = 0$ whenever $w_\alpha = 0$. We then have

$$\begin{aligned} x = & \sum_{i=1}^{\tilde{s}} \sum_{k_1=1}^{|N(1)|} \cdots \sum_{k_{|G|}=1}^{|N(|G|)|} \frac{x_1^{k_1} \cdots x_{|G|}^{k_{|G|}}}{w_1 \cdots w_{|G|}} \lambda_i \tilde{C}_{\alpha i} \begin{pmatrix} e_{k_1} \\ \vdots \\ e_{k_{|G|}} \end{pmatrix} \\ & + \sum_{j=1}^{\tilde{t}} \sum_{k_1=1}^{|N(1)|} \cdots \sum_{k_{|G|}=1}^{|N(|G|)|} \frac{x_1^{k_1} \cdots x_{|G|}^{k_{|G|}}}{w_1 \cdots w_{|G|}} \mu_j \tilde{D}_{\alpha j} \begin{pmatrix} e_{k_1} \\ \vdots \\ e_{k_{|G|}} \end{pmatrix} \end{aligned} \tag{10}$$

The result follows from (10) since all the terms

$$\tilde{C}_{\alpha i} \begin{pmatrix} e_{k_1} \\ \vdots \\ e_{k_{|G|}} \end{pmatrix}$$

lie in C and similarly for D. Furthermore we can check that the sum of the coefficients of those terms is 1, fulfilling the constraint $1^T\lambda = 1$.

For P_Y^4, the result follows from the fact that the matrix of flow constraints on the arcs w is totally unimodular. Hence every extreme point has integer values for w. Therefore x is integer as well. $\square$

3.1 A bounded version of the path reformulation

The structure of the path reformulation is interesting because it leads to a totally unimodular matrix. This fact can be further used. We can also produce the convex hull for a bounded corner polyhedron as we outline below. Consider a bounded group relaxation

$$Y_u(f) = \{x \in \mathbb{Z}_+^n : Fx \equiv f \ (\mathrm{mod}\ L(\Delta)),\ x \leq u\}, \tag{11}$$

where the vector $u \in \mathbb{Z}_+^n$ contains the bounds on the variables. We construct a digraph (V, A) with $n+1$ levels of nodes, one level per variable and a source level. Specifically the digraph has $(n+1)|G|$ nodes denoted by $V_{0\alpha}$ for the source level and by $V_{i\alpha}$ for $i = 1, \ldots, n$, $\alpha \in G$, corresponding to a group element at the i^{th} level. For each variable i, and each group element α, the arcs

$$(V_{(i-1),\alpha}, V_{i,\alpha}), (V_{(i-1),\alpha}, V_{i,((\alpha+F_{:i}) \bmod L(\Delta))}), \cdots, (V_{(i-1),\alpha}, V_{i,((\alpha+u_iF_{:i}) \bmod L(\Delta))})$$

belong to the graph. A solution to (11) is now any walk from the source node $V_{0,0}$ to the "target node" $V_{n,f}$. If we denote by $w(V_{(i-1)\alpha}, V_{i\beta})$ the flow going through the arc $(V_{(i-1),\alpha}, V_{i,\beta})$, for any $i \in N, \alpha, \beta \in G$, we have

$$\sum_{\alpha\in G}\sum_{i=1}^{n}\sum_{k=0}^{u_i} kF_{:i}w\left(V_{(i-1),\alpha}, V_{i,((\alpha+kF_{:i}) \bmod L(\Delta))}\right) \equiv f \quad (\mathrm{mod}\ L(\Delta)).$$

Hence in any solution, the value of the variables is given by

$$x_i = \sum_{k=0}^{u_i}\sum_{\alpha\in G} kw\left(V_{(i-1),\alpha}, V_{i,((\alpha+kF_{:i}) \bmod L(\Delta))}\right),$$

for all $i = 1, \ldots, n$. Figure 2 shows such a graph for the $x_1 + 2x_2 \equiv 2 \ (\mathrm{mod}\ 4)$ group problem with $x_1 \in \{0, 1, 2\}$ and $x_2 \in \{0, 1\}$. A walk from $V_{0,0}$ to $V_{2,2}$ represents a solution. We note that several nodes and their incident arcs (dashed in the figure) can be removed from the formulation: Nodes $V_{2,0}$, $V_{2,1}$, $V_{2,3}$ are fixed to zero in any solution because they have no leaving arc; node $V_{1,3}$ is fixed to zero because it has no entering arc. After removing these nodes, node $V_{1,1}$ can be removed by the same argument.

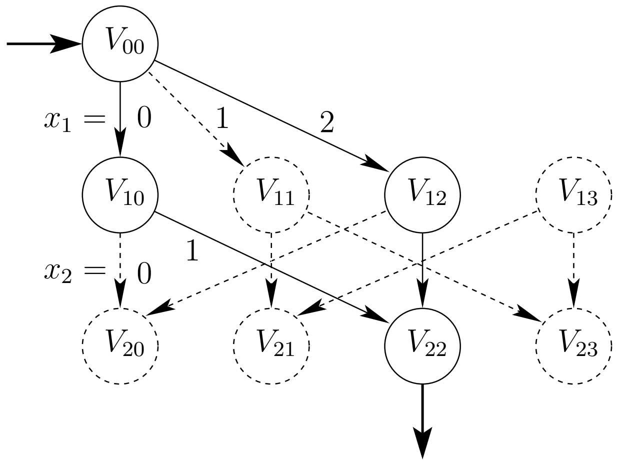

Fig. 2. The graph related to the group problem $x_1 + 2x_2 \equiv 2 \pmod 4$, $x_1 \in \{0, 1, 2\}$, $x_2 \in \{0, 1\}$. Every arcs represents the decision to set one of the variables to a certain value; the label on the arc indicates the value.

Proposition 14.

$$
\begin{aligned}
Y_u(f) = \{&x \in \mathbb{Z}^n_+ : \\
&x_i = \sum_{k=0}^{u_i} \sum_{\alpha \in G} kw \left(V_{(i-1),\alpha}, V_{i,((\alpha + kF_{:i}) \bmod L(\Delta))} \right), \quad \textit{for } i = 1, \dots, n \\
&\sum_{k=0}^{u_1} w(V_{0,0}, V_{1,(kF_{:1})}) = 1 \\
&\sum_{k=0}^{u_i} w(V_{(i-1),(\alpha - kF_{:(i-1)})}, V_{i,\alpha}) - \sum_{k=0}^{u_{i+1}} w(V_{i,\alpha}, V_{(i+1),(\alpha + kF_{:i})}) = 0 \\
&\qquad\qquad\qquad\qquad\qquad \textit{for all } i = 1, \dots, n, \ \textit{and } \alpha \in G, \ (i, \alpha) \neq (n, f) \\
&\sum_{k=0}^{u_n} w(V_{(n-1),(f - kF_{:i})}, V_{n,f}) = 1 \\
&w \in \mathbb{Z}^M_+ \qquad\qquad \},
\end{aligned}
$$

where $M = |G| \sum_{i=1}^n (u_i + 1)$. *Therefore, the expression in brackets is a valid extended formulation for* $Y_u(f)$*, referred to as the* bounded path reformulation.

The key argument to prove that the bounded path reformulation provides the convex hull of $Y_u(f)$ is again that the flow conservation constraints form a totally unimodular matrix. If we denote by P_Y^5 the polyhedron obtained from the bounded path reformulation by relaxing the integrality constraints on the variables, we have the following result.

Proposition 15.

$$P_Y^5 = \operatorname{conv}(Y_u(f)).$$

Here we suppose that we have used the advanced aggregation procedure outlined in Table 1. In general, the convex hull property does not hold anymore for such a reformulation. However, we show in this section that we can obtain a stronger formulation by adding inequalities and in one special case we can recover the convex hull property.

We consider the case in which each set $N(k)$ contains at most two elements. Thus we handle substitutions of the form

$$z_k = x_k + h_k x_{k'},$$

for $k \in I$, and $N(k) = \{k, k'\}$. Now the reformulation of Proposition 6 can be written row-wise as

$$\tilde{P} = \Bigg\{ (x, \lambda, \mu) : x_k + h_k x_{k'} = \sum_{i=1}^{s} \tilde{C}_{ki}\lambda_i + \sum_{j=1}^{t} \tilde{D}_{kj}\mu_j \text{ for } k \in I, \tag{12a}$$

$$x \in \mathbb{Z}^n_+, \quad \lambda \in \mathbb{Z}^s_+, \quad \mu \in \mathbb{Z}^t_+, \quad \sum_{i=1}^{s} \lambda_i = 1 \quad \Bigg\}. \tag{12b}$$

This can be viewed as

$$\tilde{P} = \bigcap_{k \in I} \Big\{ (x, \lambda, \mu) : x \in \mathbb{Z}^n_+, (x_k, x_{k'}, \lambda, \mu) \in \tilde{P}_k \Big\}$$

where

$$\tilde{P}_k = \Bigg\{ (x_k, x_{k'}, \lambda, \mu) : x_k + h_k x_{k'} = \sum_{i=1}^{s} \tilde{C}_{ki}\lambda_i + \sum_{j=1}^{t} \tilde{D}_{kj}\mu_j,$$
$$(x_k, x_{k'}) \in \mathbb{Z}^2_+, \quad \lambda \in \mathbb{Z}^s_+, \quad \mu \in \mathbb{Z}^t_+, \quad \sum_{i=1}^{s} \lambda_i = 1 \quad \Bigg\}.$$

Let P_k be the corresponding polyhedron obtained by dropping the integrality requirements on the variables.

Observation 16. *For fixed k,*

$$P_k \neq \operatorname{conv}(\tilde{P}_k)$$

unless h_k divides $\tilde{C}_{ki}$ for all i and h_k divides $\tilde{D}_{kj}$ for all j.

If $h_k = 1$, the advanced aggregation can be viewed as a standard aggregated formulation. In this case, we have seen that the convex hull property is satisfied. But in general, the polyhedron P_k has fractional extreme points. However the convex hull of $\tilde{P}_k$ can be related to the convex hull of a group problem modulo h_k. We present a partial result. Let

$$\tilde{P}_{kl} = \left\{ (x_k, x'_k, \mu) : (x_k, x'_k, \lambda, \mu) \in \tilde{P}_k,\ \lambda_l = 1 \right\},$$

i.e., we fix some $\lambda_l = 1$ in $\tilde{P}_k$. The next proposition states that the convex hull of $\tilde{P}_{kl}$ is exactly obtained by adding all the corner polyhedron facets of the equation (12a) defining $\tilde{P}_{kl}$ modulo h_k.

Proposition 17.

$$\begin{aligned}\mathrm{conv}(\tilde{P}_{kl}) = \mathrm{conv}\Bigg\{ &(x_k, x_{k'}, \mu) : -x_k + \sum_{j=1}^{t} \tilde{D}_{kj}\mu_j \equiv -\tilde{C}_{kl} \pmod{h_k}, \\ &x_k \in \mathbb{Z}_+, \quad x'_k \in \mathbb{R}, \quad \mu \in \mathbb{Z}^t_+ \Bigg\} \qquad (13a)\\ \cap \Bigg\{ &(x_k, x_{k'}, \mu) : x_{k'} = \frac{1}{h_k}\Bigg(\tilde{C}_{kl} + \sum_{j=1}^{t} \tilde{D}_{kj}\mu_j - x_k\Bigg) \Bigg\}. \qquad (13b)\end{aligned}$$

Proof: Let us denote

$$\begin{aligned}Q_{kl} = \Bigg\{ &(x_k, x_{k'}, \mu) : -x_k + \sum_{j=1}^{t} \tilde{D}_{kj}\mu_j \equiv -\tilde{C}_{kl} \pmod{h_k}, \\ &x_k \in \mathbb{Z}_+, \quad x'_k \in \mathbb{R}, \quad \mu \in \mathbb{Z}^t_+ \Bigg\}.\end{aligned}$$

Every integer point in $\tilde{P}_{kl}$ is also clearly in Q_{kl} since the modulo constraint (13a) is valid for the set $\tilde{P}_{kl}$. The difference is that $x_{k'}$ can be negative and not integer in Q_{kl}. We show now that every extreme point of (13a) with the definition (13b) is such that $x_{k'}$ is nonnegative and integer.

First, $x_{k'}$ is clearly integer for every extreme point by (13a). Then, we know that every extreme point of a modulo constraint of the type (13a) has the property that all variables take values less than the order of the group and thus $x_k \leq h_k - 1$. Hence

$$\tilde{C}_{kl} + \sum_{j=1}^{t} \tilde{D}_{kj}\mu_j - x_k \geq -h_k + 1,$$

as $\tilde{C}_{kl}, \tilde{D}_{kj}, \mu_j \geq 0$. But from (13a), we also know that the expression is a multiple of h_k. Therefore

$$\tilde{C}_{kl} + \sum_{j=1}^{t} \tilde{D}_{kj}\mu_j - x_k \geq 0,$$

which proves that $x_{k'} \geq 0$ for every extreme point. $\square$

This proposition shows that we can strengthen a reformulation obtained by an advanced aggregation using the facets of a corner polyhedron. The size of the group problem considered is related to the coefficient with which the aggregation is made in the case of the aggregation of two variables. It is well known that the number of facets of a group problem grows with the value of the modulus. Therefore, if the coefficient of the aggregation remains small, the number of facets to add to the reformulation stays small. The extreme case is, of course, if the coefficient $h_k = 1$, no facets have to be added.

Proposition 17 was shown for the particular case of a fixed $\lambda_l = 1$. In general the proposition allows one to add valid inequalities to any reformulation obtained by an advanced aggregation, but does not guarantee the convex hull property. When $h_k = 2$, the result of Proposition 17 can be generalized to obtain the convex hull property.

Proposition 18. *The polyhedron*

$$P_k^2 = \Big\{ (x_1, x_2, \lambda, \mu) \in \mathbb{R}^{2+s+t} : x_1 + 2x_2 - \sum_{j=1}^{t} \tilde{D}_{kj}\mu_j = \sum_{i=1}^{s} \tilde{C}_{ki}\lambda_i \tag{14a}$$

$$x_2 - \sum_{j=1}^{t} \left\lceil \frac{\tilde{D}_{kj}}{2} \right\rceil \mu_j \leq \sum_{i=1}^{s} \left\lfloor \frac{\tilde{C}_{ki}}{2} \right\rfloor \lambda_i \tag{14b}$$

$$\sum_{i=1}^{s} \lambda_i = 1 \qquad \Big\} \tag{14c}$$

is integral.

Proof: We must show that every extreme point of P_k^2 is integer. First we consider the extreme points for which exactly one λ_i is non-zero. Let us fix $1 \leq l \leq s$ such that $\lambda_l = 1$. The inequality (14b) is the only facet that needs to be added to obtain the convex hull of the group problem

$$x_1 - \sum_{j=1}^{t} \tilde{D}_{kj}\mu_j \equiv \tilde{C}_{kl} \ (\mathrm{mod}\ 2).$$

Therefore we can apply Proposition 17 and we know that every extreme point corresponding to $\lambda_l = 1$ is integer.

We show now that no extreme point x^* can be such that $\lambda^*_{i_1}, \lambda^*_{i_2} \neq 0$, for $i_1 \neq i_2$. If this were the case, then exactly one other variable would be non-zero making the three constraints tight. It clearly cannot be one of the other μ variables because it would not fulfill (14a). Suppose $x^*_1 \neq 0$. From (14a), we have $x^*_1 = \tilde{C}_{ki_1}\lambda^*_{i_1} + \tilde{C}_{ki_2}\lambda^*_{i_2}$. This is the convex combination of the two feasible points $(x_1^1 = \tilde{C}_{ki_1}, \lambda_{i_1}^1 = 1, \lambda_{i_2}^1 = 0)$ and $(x_1^2 = \tilde{C}_{ki_2}, \lambda_1^2 = 0, \lambda_2^2 = 1)$. Therefore, x^* cannot be extreme. Suppose now that $x_2^*, \lambda_{i_1}^*, \lambda_{i_2}^* \neq 0$. As (14a) and (14b) are tight, it means that $\tilde{C}_{ki_1}$ and $\tilde{C}_{ki_2}$ are even because otherwise (14a) cannot be tight. Then x^* is a convex combination of two points with $\lambda_{i_1} = 1$ and $\lambda_{i_2} = 1$ respectively like in the case where x_1 is non-zero. Therefore no point with more than one λ_i non-zero can be extreme.

Now, all the possible cases have been explored, and they all lead to either an integer extreme point or to an unbounded solution. Therefore, all the extreme points of the polyhedron are integer. □

Example 19. Consider the set

$$\begin{aligned} Q =\{ & (x_1, x_2, \lambda_1, \lambda_2, \mu_1, \mu_2, \mu_3) \in \mathbb{Z}_+^7 : \\ & x_1 + 2x_2 = \lambda_1 + 2\lambda_2 + 7\mu_1 + 2\mu_2 + 9\mu_3 \\ & \lambda_1 + \lambda_2 = 1 \qquad \} \end{aligned}$$

Its convex hull is given by Proposition 18. Therefore

$$\begin{aligned} Q^* =\{ & (x_1, x_2, \lambda_1, \lambda_2, \mu_1, \mu_2, \mu_3) \in \mathbb{R}_+^7 : \\ & x_1 + 2x_2 = \lambda_1 + 2\lambda_2 + 7\mu_1 + 2\mu_2 + 9\mu_3 \\ & \lambda_1 + \lambda_2 = 1 \\ & x_2 \leq \lambda_2 + 4\mu_1 + \mu_2 + 5\mu_3 \qquad \} \end{aligned}$$

is integral and $Q^* = \mathrm{conv}(Q)$.

It is an interesting research problem to characterize the convex hull of all integer points in a general advanced aggregation extended formulation.

4 Computation of irreducible group solutions

The irreducible solutions of a group problem can be computed by using a Buchberger-type algorithm (see [1]) or by lexicographic enumeration. Both types of methods are very slow, and are therefore not suited to be used within an iterative integer programming algorithm.

4.1 Connection to knapsack master solutions

In [10,11], it was proposed to pre-compute tables of irreducible solutions to knapsack master equations, in order to make use of them in an iterative algo-

Table 3
The cardinality of knapsack master sets and group master sets. The irreducible knapsack solutions do not include trivial solutions of the form (e^i, e^i), and the symmetric solutions (x, y) and (y, x) are only counted once. The column "Orbits" shows the number of orbits of the irreducible solutions under the action of the automorphism group of $\mathbb{Z}/D\mathbb{Z}$.

D	Knapsack	Group	Orbits	D	Knapsack	Group	Orbits
2	1	2	2	15	58 171	6 375	824
3	5	7	4	16	99 328	9 369	1 264
4	15	15	9	17	181 514	17 208	1 076
5	47	38	10	18	287 239	18 852	3 193
6	102	56	30	19	502 116	36 591	2 034
7	276	143	25	20	775 710	40 031	5 195
8	578	209	64	21	1 239 710	63 472	5 356
9	1 261	402	73	22	1 956 334	87 618	8 767
10	2 465	598	152	23	3 210 736	145 717	6 624
11	5 362	1 267	128	24	4 660 786	147 231	19 217
12	9 285	1 445	397	25	7 297 823	258 184	12 929
13	18 900	3 238	271	26	10 997 235	318 010	26 507
14	33 269	4 054	681	27	16 536 803	450 183	25 152

rithm. Tables of the irreducible solutions to a knapsack master equation

$$\begin{gathered} \sum_{i=1}^{n} i x_i - \sum_{i=1}^{n} i y_i = 0 \\ x \in \mathbb{Z}_+^n, \quad y \in \mathbb{Z}_+^n \end{gathered} \tag{15}$$

up to $n = 27$ were computed with a specialized recursive algorithm. These tables and the implementation of the algorithm are available at [12].

We can make use of these tables to read off the irreducible solutions to the single-row master group problems

$$x_1 + 2x_2 + 3x_3 + \cdots + (D-1)x_{D-1} \equiv D_0 \pmod{D} \tag{16}$$

with varying right-hand side D_0 as follows. Eq. (16) can be written in the form

$$x_1 + 2x_2 + 3x_3 + \cdots + (D-1)x_{D-1} - D_0 y - Dz = 0, \tag{17}$$

where the y variable can be left out in the homogeneous case $D_0 = 0$. The irreducible solutions to (17) are contained in the pre-computed table. The solutions with $y = 1$ correspond to the inhomogeneous solutions to (16), and the solutions with $y = 0$ correspond to the homogeneous solutions ($D_0 = 0$).

Because there are only two variables with negative coefficients in (17) and variable y is bounded above by 1, the knapsack master database stores more solutions than we need. Table 3 shows the cardinality of both the knapsack master database and the subset that contains the irreducible group solutions for all possible right-hand sides D_0.

4.2 The disaggregation procedure

Suppose, for example, that we construct a table $\mathcal{T}_D$ of the irreducibles of the master group problem

$$x_1 + 2x_2 + 3x_3 + \cdots + (D-1)x_{D-1} \equiv D_0 \pmod{D}, \tag{18}$$

for some $D \in \mathbb{Z}_+$ and all possible right-hand sides $0 \leq D_0 \leq D-1$. Once this table is precomputed, we are able to read off the irreducibles of every (mod D) group problem from the irreducibles given in the table. Suppose that we want to find the irreducibles for the group problem

$$\sum_{i=1}^{n} \alpha_i x_i \equiv D_0 \pmod{D}. \tag{19}$$

Let ρ_α denotes the number of times the coefficient α appears in (19) for $\alpha \in G$. We can then write (19) in the form

$$\sum_{\alpha \in G} \sum_{k=1}^{\rho_\alpha} \alpha x_{\alpha k} \equiv D_0 \pmod{D}. \tag{20}$$

It was shown in [10] how to read off solutions from master solution tables. We summarize the procedure given there. The irreducible solutions to (20) correspond to the solutions v to (18) where all components v_α whose coefficients α do not occur in (20) are zero. It was pointed out that the enumeration of these solutions can be carried out efficiently using data structures for "orthogonal range searching."

Now let $v \in T_D$ be such a solution where $\rho_\alpha = 0$ implies $v_\alpha = 0$. For every $\alpha \in G$ with $\rho_\alpha \neq 0$, we consider all possible "number partitions"

$$v_\alpha = \sum_{k=1}^{\rho_\alpha} w_{\alpha k}, \quad w_{\alpha k} \in \mathbb{Z}_+. \tag{21}$$

(There are $\binom{\rho_\alpha + v_\alpha - 1}{\rho_\alpha}$ number partitions.) The combinations of all possible number partitions (21) for all $\alpha \in G$ define the irreducible solutions to (20).

Table 4
Disaggregation Algorithm

1. **Set** $\mathcal{I} := \emptyset$.
2. **For each** vector $v \in \mathcal{T}_D$ do
3. **If** for all $\alpha \in G$, $\rho_\alpha = 0$ implies $v_\alpha = 0$ **then**
4. Create irreducibles w^j corresponding to the solutions of the number partition problem

   $\sum_{k=1}^{\rho_\alpha} w_{\alpha k} = v_\alpha$

   for all $\alpha \in G$ such that $\rho_\alpha > 0$.
5. **Set** $\mathcal{I} := \mathcal{I} \cup \left(\bigcup_j \{w^j\}\right)$.
6. **Return** $\mathcal{I}$.

We show the disaggregation algorithm in Table 4. In the following proposition we summarize its properties.

Proposition 20. **(i)** *The disaggregation algorithm provides the set of all irreducibles of* (20)*.*
(ii) *The number of irreducibles generated from a vector v for which $\rho_\alpha = 0$ implies $v_\alpha = 0$ for all $\alpha \in G$ is*

$$\prod_{\alpha \in G: v_\alpha > 1} \binom{\rho_\alpha + v_\alpha - 1}{\rho_\alpha}. \tag{22}$$

The number (22) of irreducible solutions of a group problem grows exponentially with the size of ρ_α. This particularly affects the disaggregated formulation. However the other formulations based on irreducibles group together the variables having the same coefficients so that ρ_α is small for all $\alpha \in G$.

4.3 Making use of group automorphisms

For the convenience of the reader, we recall the definition of group automorphisms and of actions of a group.

Definition 21. **(a)** Let G be an additively written group. A *(group) automorphism* of G is a bijective mapping $\phi\colon G \to G$ with $\phi(g+h) = \phi(g) + \phi(h)$ for all $g, h \in G$.
(b) The set of all automorphisms of G forms a (multiplicatively written) group and is denoted by $\mathrm{Aut}(G)$.

In [3], Gomory proposes to make use of group automorphisms that induce mappings between faces of the Corner Polyhedron. He observes in particular that group automorphisms induce mappings between the vertices. It is clear that this result also carries over to the (irreducible) solutions of

$$S_G(b) = \Big\{ x \in \mathbb{Z}_+^{|G|} : \sum_{\alpha \in G} \alpha x_\alpha = b \Big\},$$

which in general strictly includes the set of vertices. We denote the set of irreducible solutions of $S_G(b)$ by $S_G^{\mathrm{irr}}(b)$.

Lemma 22. *Let G be a group and $\phi \in \operatorname{Aut}(G)$ be an automorphism of G. Let $b \in G$.*

(a) *Then $(x_g)_{g\in G}$ is a solution in $S_G(b)$ if and only if*

$$\phi(x) := (x_{\phi^{-1}(g)})_{g\in G} \tag{23}$$

is a solution in $S_G(\phi(b))$.

(b) *Moreover, $(x_g)_{g\in G}$ is an irreducible solution in $S_G^{\mathrm{irr}}(b)$ if and only if $\phi(x)$ is an irreducible solution in $S_G^{\mathrm{irr}}(\phi(b))$.*

Definition 23. Let X be a set and let Φ be a (multiplicatively written) group with the unit element e. Then Φ is said to *act* on the set X if for every $\phi \in \Phi$ and every $x \in X$ there is a uniquely defined element $\phi(x) \in X$, such that the following conditions hold:

(i) $e(x) = x$ for all $x \in X$;
(ii) $\phi_1(\phi_2(x)) = (\phi_1\phi_2)(x)$ for all $x \in X$ and $\phi_1, \phi_2 \in \Phi$.

If $x \in X$, the set $\{\, \phi(x) \in X : \phi \in \Phi \,\}$ is called the *orbit* of the element x under the action of Φ.

In our application, we observe two actions of the automorphism group $\operatorname{Aut}(G)$. First, $\operatorname{Aut}(G)$ acts on the group G by the normal definition of $\phi(g)$. Second, by the defining equation (23), $\operatorname{Aut}(G)$ acts on the sets

$$\bigcup_{b\in G} S_G(b) = \mathbb{Z}_+^{|G|} \quad \text{and} \quad \bigcup_{b\in G} S_G^{\mathrm{irr}}(b) \tag{24}$$

of (irreducible) solutions for all possible right-hand sides $b \in G$. Therefore, by Lemma 22, we need not study the (irreducible) solutions for every possible right-hand side b. It suffices to consider representatives of the orbits of G under the action of $\operatorname{Aut}(G)$.

In the case of a cyclic group $G = \mathbb{Z}/D\mathbb{Z}$, corresponding to a single-row group relaxation modulo D, the automorphism group is isomorphic to the multiplicative group $(\mathbb{Z}/D\mathbb{Z})^*$ formed by the the elements α that are coprime to D, i.e., $\gcd(\alpha, D) = 1$. More precisely, every automorphism $\phi \in \operatorname{Aut}(G)$ is of the form

$$\phi(\alpha) = (k \cdot \alpha) \bmod D,$$

where k is a number coprime to D. The orbits of G under the action of $\mathrm{Aut}(G)$ consist of the elements that have the same greatest common divisor with D. Therefore, the orbits correspond to the divisors of the group order D.

In particular, for a cyclic group $G = \mathbb{Z}/p\mathbb{Z}$ of prime order p, the only divisors are 1 and p. The divisor 1 corresponds to the orbit of all non-zero group elements; we can pick $1 \in G$ as a representative. The divisor p corresponds to the single element $0 \in G$, which stays fixed under all group automorphisms.

In the same way, we do not need to store all irreducible solutions for a given right-hand side b. Let Φ_b denote the subgroup of automorphisms of G that keep b fixed (Φ_b is the *isotropy group* of b). Then Φ_b acts on the sets $S_G(b)$ and $S_G^{\mathrm{irr}}(b)$ of (irreducible) solutions for the right-hand side b. In particular, because the zero right-hand side stays fixed under all automorphisms, the full automorphism group $\Phi_0 = \mathrm{Aut}(G)$ acts on the set $S_G^{\mathrm{irr}}(0)$ of homogeneous irreducible solutions via (23). Therefore it suffices to store representatives of all orbits. For instance, we could store the representative that is lexicographically minimal.

Table 3 shows the total number of representatives of irreducible solutions to the master group problem for the cyclic groups $G = \mathbb{Z}/D\mathbb{Z}$ up to $D = 27$. The number of representatives is much smaller than the number of irreducible knapsack solutions. It is an interesting problem to devise an algorithm that computes only the representatives, avoiding all the symmetry induced by the automorphism group during the computation.

4.4 An iterative procedure using subgroups

It is computationally intractable to build tables for single-row master group problems if the modulus D is too big. However, there is an alternative for composite groups with D non prime, in the case of a single-row group problem. It is possible to build up the set of irreducibles from the irreducibles of smaller groups. Suppose specifically that $D = pq$ with $p, q > 1$ integer. Starting from

$$W(a_0) = \left\{ w \in \mathbb{Z}_+^D : \sum_{j=1}^{D-1} a_j w_j \equiv a_0 \pmod{D} \right\}, \tag{25}$$

we consider the relaxation

$$\tilde{W}(g_0) = \left\{ w \in \mathbb{Z}_+^D : \sum_{j=1}^{D-1} g_j w_j \equiv g_0 \pmod{p} \right\}$$

where $a_j = f_j p + g_j$ for all $0 \leq j \leq D-1$. Given the matrices of irreducibles $\tilde{C}$ and $\tilde{D}$ for $\tilde{W}(g_0)$, we have that

$$\tilde{W}(g_0) = \Big\{ w : w = \tilde{C}\lambda + \tilde{D}\mu,\ 1^T\lambda = 1, \lambda_i \in \mathbb{Z}_+, \mu_j \in \mathbb{Z}_+ \Big\}.$$

Substituting for w in $W(a_0)$, we obtain that

$$a^T\tilde{C}\lambda + a^T\tilde{D}\mu \equiv a_0 \ (\text{mod } D),$$

where $a^T = pf^T + g^T$. In other words

$$\begin{aligned} &pf^T\tilde{C}\lambda + g^T\tilde{C}\lambda + pf^T\tilde{D}\mu + g\tilde{D}\mu \equiv f_0 p + g_0 \ (\text{mod } pq) \\ &1^T\lambda = 1. \end{aligned}$$

Because $g^T\tilde{C} = g_0(1\cdots 1)$ and $g\tilde{D} = 0$, we obtain

$$\begin{aligned} &pf^T\tilde{C}\lambda + pf^T\tilde{D}\mu \equiv f_0 p \ (\text{mod } pq) \\ &1^T\lambda = 1. \end{aligned}$$

or

$$\begin{aligned} &f^T\tilde{C}\lambda + f^T\tilde{D}\mu \equiv f_0 \ (\text{mod } q) \\ &1^T\lambda = 1. \end{aligned}$$

We now consider the possible choices for λ separately. If λ is the s-th unit vector, $s \in \{1, \ldots, S\}$, let A^s be the matrix of inhomogeneous irreducibles for

$$\Big\{ \mu : f^T\tilde{D}\mu \equiv f_0 - f^T\tilde{C}_{:s} \ (\text{mod } q) \Big\}$$

and B the matrix of homogeneous irreducibles. We now have the representation

$$\begin{aligned} \mu &= A^s\alpha^s + B\beta \\ \lambda_s &= 1 \end{aligned}$$

Finally we obtain the representation

$$\mu = \sum_{s=1}^{S} A^s\alpha^s + B\beta \tag{26a}$$

$$\sum_{s=1}^{S} 1^T\alpha^s = 1 \tag{26b}$$

The columns of the representation (26) contain all the irreducible inhomogeneous and homogeneous solutions to the group problem (25). We remark, however, that the same solution can appear multiple times, and it is also possible that reducible solutions show up. As it is easy to remove the duplicates and the reducible solutions from (26), the construction above opens up the possibility of computing the set of irreducibles of bigger groups from the database that we have precomputed. In other words, it suffices to compute group master sets for cyclic groups of prime order.

We remark that the iterative procedure above can be applied in a more general situation. Above we applied it to compute irreducible solutions to a group problem in $\mathbb{Z}/(pq)\mathbb{Z}$ from the (homogeneous and inhomogeneous) irreducible solutions to the group problems in the subgroup $\mathbb{Z}/p\mathbb{Z}$ and in the factor group $\mathbb{Z}/q\mathbb{Z}$. The procedure can be applied for an arbitrary (abelian) group G to compute the irreducible solutions from the irreducible solutions to a given subgroup G' of G and to the corresponding factor group G/G'. We will make use of this observation when we deal with multi-row group relaxations.

Example 24. Consider the problem of finding irreducibles for

$$\begin{aligned} &w_1 + 2w_2 \equiv 3 \pmod 4 \\ &w \in \mathbb{Z}^2_+. \end{aligned}$$

By first considering the equation (mod 2), it yields

$$\begin{aligned} &w_1 + 0w_2 \equiv 1 \pmod 2 \\ &w \in \mathbb{Z}^2_+, \end{aligned}$$

which means that

$$\begin{aligned} &\begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \lambda_1 + \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \mu \\ &\lambda_1 = 1. \end{aligned} \tag{27}$$

Substituting gives

$$\begin{aligned} &\lambda_1 + 2\mu_1 + 2\mu_2 \equiv 3 \pmod 4 \\ &\lambda_1 = 1. \end{aligned}$$

By replacing λ_1 by its value and by dividing by 2, we now have

$$\mu_1 + \mu_2 \equiv 1 \pmod 2.$$

By using the representation by irreducibles, we can write

$$\begin{aligned} &\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \alpha + \begin{pmatrix} 2 & 1 & 0 \\ 0 & 1 & 2 \end{pmatrix} \beta \\ &\alpha_1 + \alpha_2 = 1. \end{aligned} \tag{28}$$

By substituting (28) in (27), we finally obtain

$$\begin{aligned}\begin{pmatrix} w_1 \\ w_2 \end{pmatrix} &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \alpha + \begin{pmatrix} 4 & 2 & 0 \\ 0 & 1 & 2 \end{pmatrix} \beta \\ &= \begin{pmatrix} 3 & 1 \\ 0 & 1 \end{pmatrix} \alpha + \begin{pmatrix} 4 & 2 & 0 \\ 0 & 1 & 2 \end{pmatrix} \beta \qquad (29)\end{aligned}$$

with $\alpha_1 + \alpha_2 = 1$ and $\alpha, \beta \geq 0$ and integer. In this example, we obtain a representation that consists of irreducible solutions only.

4.5 Multi-row group relaxations

So far we have focused on single-row group relaxations, which lead to cyclic groups $\mathbb{Z}/D\mathbb{Z}$. For multi-row group relaxations with moduli $\Delta_1, \ldots, \Delta_k$, we are investigating a group problem in the direct product

$$G = (\mathbb{Z}/\Delta_1\mathbb{Z}) \times \cdots \times (\mathbb{Z}/\Delta_k\mathbb{Z}).$$

Its structure depends on the existence of common divisors of the moduli. If $\Delta_1, \ldots, \Delta_k$ are pairwise coprime, then

$$(\mathbb{Z}/\Delta_1\mathbb{Z}) \times \cdots \times (\mathbb{Z}/\Delta_k\mathbb{Z}) \cong \mathbb{Z}/(\Delta_1 \cdots \Delta_k)\mathbb{Z}. \qquad (30)$$

Therefore, we could read off the irreducible solutions to the multi-row group relaxations from the single-row master group table of modulus $\Delta_1 \cdots \Delta_k$.

In the more general case, we can consider a chain of subgroups of G and apply the iterative construction above. For instance, we could take the chain

$$\begin{aligned}\mathbb{Z}/\Delta_1\mathbb{Z} &\leq (\mathbb{Z}/\Delta_1\mathbb{Z}) \times (\mathbb{Z}/\Delta_2\mathbb{Z}) \\ &\leq (\mathbb{Z}/\Delta_1\mathbb{Z}) \times (\mathbb{Z}/\Delta_2\mathbb{Z}) \times (\mathbb{Z}/\Delta_3\mathbb{Z}) \leq \cdots \leq G,\end{aligned}$$

where $G_1 \leq G_2$ denotes that G_1 is a subgroup of G_2.

Table 5 shows the number of irreducible solutions to two-row master group problems. One can see that in the case of coprime moduli, we obtain the same number of irreducible solutions as in the single-row master table of the product of the moduli.

Table 5
The number of irreducibles for two-row master group problems. An asterisk marks the numbers corresponding to coprime pairs of moduli.

Second modulus	First modulus 2	3	4	5
2	10			
3	56*	252		
4	132	1445*	5228	
5	598*	6375*	40031*	169892
6	1048	12954	106022	
7	4054*	63472*		
8	6324	147231*		
9	18852*	311253		

5 Algorithmic schemes based on reformulations

We now discuss two ways, one dual and one primal, to incorporate the extended reformulations presented above algorithmically. We are trying to solve an integer program

$$\begin{aligned} &\max\ c^T x \\ &\text{s.t.}\quad Ax = b \\ &\qquad\quad x \in \mathbb{Z}_+^{m+n}. \end{aligned} \tag{31}$$

For the primal algorithmic scheme we assume that a solution $x_0 \in \mathbb{Z}_+^{m+n}$ feasible for (31) is given.

In Table 6 we present the dual scheme. On the other hand the primal approach closely follows the Integral Basis Method; it is shown in Table 7.

For the computational experiments that we will present in section 6, we have used preliminary implementations of both algorithmic schemes. The implementation was carried out within the algorithmic framework of the Integral Basis Method introduced by Haus, Köppe and Weismantel [10]. This means that even for the dual algorithmic scheme, we maintain a primal all-integer tableau, which we update in our reformulations. As a result, in our implementation of the dual algorithmic scheme, we solve the linear relaxation from scratch every time. (In a real implementation, some effort would need to be spent to update the optimal solution of the relaxation using the dual simplex method after the reformulation.) Because of these shortcomings of our implementations, we also do not report the computation times of our experiments.

In particular the following four phases are the same as in the Integral Basis

Table 6
The dual algorithmic scheme

1. **Initialization**
   Take any valid formulation of the integer program.

2. **Search for Fractionality**
   Compute a fractional point x_{frac} of the polyhedron of the linear relaxation, for instance the LP optimum.
   If no such point exists, return the LP optimum as an optimal solution of (31).

3. **Group Relaxation**
   Generate a group relaxation
   $$Y^G = \{x \in \mathbb{Z}_+^n : Fx \equiv f \ (\text{mod } L(\Delta))\}$$
   from a subset of tight constraints defining x_{frac}.

4. **Reformulation**
   Compute an extended reformulation for Y^G.

5. **Make Compact**
   Use bounds on variables, as well as other problem constraints or problem knowledge to eliminate as many of the new variables as possible, adding also GUB or SOS constraints to the description of Y^G. The goal here is to prevent an excessive increase in the number of variables. If the number is still too large, the set Y^G must be relaxed further.

6. **Update Problem Formulation**

7. **Go to** Step 2.

Method and have therefore been described in [10]: Initialization, Make Compact, Update Simplex Tableau, Augmentation. In Section 4, we addressed the question of computing the reformulation. In the following, we address some questions related to the phases Search for Fractionality and Group Relaxation.

Search for fractionality and group relaxation. For the dual algorithmic scheme, we compute a fractional point x_{frac} of the polyhedron of the linear relaxation. The goal is to define a group relaxation that cuts off this fractional point. Now there are several possible choices for the fractional point.

1. One possibility is to consider the linear programming optimum of the current formulation. This is the most obvious choice for a dual algorithm. The disadvantage of this choice is that the determinant of the optimal simplex basis is typically very large, leading to a group relaxation with respect to a group of very large order. The problem is to choose an appropriate subgroup such that the fractional point is still cut off.

Table 7
The primal algorithmic scheme

1. **Initialization**
   Compute an integer tableau for which x_0 is the basic feasible solution.
2. **Group Relaxation**
   How to choose the appropriate group relaxation is less clear than in the dual algorithmic scheme, as there is no obvious point to be cut off.
3. **Reformulation**
4. **Make Compact**
5. **Update Simplex Tableau**
   Update the integer tableau introducing the new columns and rows. Select the right variables to enter the basis in order to recover a tableau.
6. **Augmentation**
   Check for augmentation, i.e., check if there exists a new column v of positive reduced cost such that $x_i + v$ is feasible. In that case, pivot in v in an integer fashion and obtain an integer tableau representing the new feasible solution $x_{i+1} = x_i + v$.
7. **Go to** Step 2.

2. Another possibility is to maintain a feasible integer point that is a vertex of the polyhedron of the linear relaxation and to consider fractional vertices that are in its graph-theoretic vicinity. In other words, we perform a few non-degenerate simplex iterations and consider the basic feasible solution that we reach. The advantage of this choice could be that after only a few simplex iterations starting from an integer solution, the determinant of the simplex basis stays fairly small. It is not clear, however, how to choose an appropriate fractional vertex.

In our experiments with the dual method, we decided to use the first possibility. We compute the linear programming optimum of our current tableau using floating point arithmetic (with CPLEX). We extract one or two fractional basic variables whose fractional part is closest to 1/2. We obtain the index sets of the optimal basis from CPLEX and reconstruct in exact rational arithmetic the optimal simplex tableau by performing Gaussian elimination. For this, we use the GNU Multiple Precision library.

It turns out that on our entire test set, the least common multiple of all the denominators occuring in a fractional row is gigantic, namely up to hundreds of digits. Clearly one cannot work with a group of this size. Our strategy is to define a tractable group problem by selecting a modulus (or two moduli) between 2 and 20 (between 2 and 8) for which the irreducible solutions have been tabulated in our database. Within this range, we choose a modulus that

(i) is not a divisor of the right hand side, in order to cut off the optimum point with our group relaxation,
(ii) is a divisor of the coefficient of the basic variable, if possible. Otherwise, there always exists some optimal solution to the group problem that has a zero reduced cost and the value of the dual bound does not change.
(iii) produces the maximum number of 0-residue coefficients in order to reduce the size of the reformulation.

Example 25. Suppose that an interesting row provided by the fractional tableau at the optimum LP point is

$$x_0 + \frac{930}{1000}x_1 + \frac{1724}{1000}x_2 - \frac{937}{1000}x_3 - \frac{620}{1000}x_4 + \frac{30}{1000}x_5 = \frac{127}{1000},$$

where x_0 is the basic variable. It can also be written in integer form

$$1000x_0 + 930x_1 + 1724x_2 - 937x_3 - 620x_4 + 30x_5 = 127.$$

As we pointed out before, it is computationally intractable to work with a (mod 1000) group. We decide therefore to choose some modulus between 2 and 20 in order to be able to use our precomputed table of irreducible solutions. We preferably choose a divisor of 1000. If we choose 5 or 10 as a modulus, four variables will disappear from the group relaxation, which can be interesting from the point of view of having a reformulation that is not too large. Furthermore, as 127 is not divisible by 5, the group relaxation will cut off the fractional point. Choosing 5 as the order of the group, we obtain

$$Y(2) = \{x_2, x_3 \in \mathbb{Z}_+ : 4x_2 + 3x_3 \equiv 2 \pmod 5\}.$$

6 Computational experiments

In our computational experiments, we have investigated only two of the four proposed reformulations, namely the disaggregated and the aggregated formulation.

6.1 Improving the dual bound

In our first set of computational experiments, we explored how the strength of the formulation is changed by the reformulation steps in the dual algorithmic scheme. For several 0-1 problems from the benchmark library MIPLIB, we set up an all-integer simplex tableau corresponding to the optimal integer solution. Starting from this formulation, we apply aggregated reformulations based on single-row group relaxations, in order to improve the linear programming/dual

Table 8
Improving the LP bound for some MIPLIB instances

`p0033`, IP value 3089			`lseu`, IP value 1120		
Iteration	Columns	LP bound	Iteration	Columns	LP bound
0	33	2520.6	0	89	837.0
10	95	2610.2	10	146	854.8
20	155	2647.4	20	199	901.6
30	195	2687.8	30	223	1001.1
40	221	2824.1	40	281	1006.8
50	290	2830.9	50	329	1008.2
70	361	2832.7	60	414	1008.9
110	626	2833.9	70	454	1009.2
120	653	2840.6	80	490	1009.3

Table 9
Improving the LP bound for MIPLIB instances after IP preprocessing

`p0548`, IP value 8691			`p2756`, IP value 3124		
Iteration	Columns	LP bound	Iteration	Columns	LP bound
0	371	7665.6	0	1557	2808.8
10	464	7681.1	10	1659	2809.5
20	613	7699.4	20	1715	2814.6
30	736	7710.1	30	1782	2816.6
40	2070	7764.2	50	2225	2817.6
50	3307	7801.6	60	2311	2818.0
60	3574	7807.7	70	2379	2818.9
70	3893	7842.2	80	2431	2824.7
80	3979	7941.4	90	2486	2827.5
90	4034	7975.1	100	2564	2836.0
100	4129	7997.3	110	2633	2853.9
110	4199	8017.2	120	2703	2861.7
120	4251	8026.7	130	2764	2862.5
130	4355	8047.7	150	2939	2863.3

bound. Table 8 shows the results on the instances `p0033` and `lseu`, for which we could obtain a significant increase of the LP bound in our experiments.

Some of the tested instances, like the Padberg–Crowder–Johnson instances `p0548` and `p2756`, involve rows with big-M coefficients. It is clear that a group reformulation with such rows is very weak or even meaningless. Indeed, we were not able to obtain any increase of the LP bound for these instances.

Table 10
Improving the LP bound for MIPLIB instances augmented with strong cuts

`p0201`, IP value 7615			`p0548`, IP value 8691		
Iteration	Columns	LP bound	Iteration	Columns	LP bound
0	159	7185.0	0	362	8135.6
2	187	7235.5	5	376	8142.7
4	201	7239.6	10	394	8152.1
6	229	7243.5	15	437	8154.5
8	263	7250.7	20	477	8156.3
10	494	7260.1	25	528	8162.3
12	504	7276.0	30	567	8166.1
14	513	7281.6	35	676	8184.5
16	525	7284.6	40	1072	8191.1
18	570	7286.8	45	1584	8204.0
20	588	7288.3	50	1929	8212.2
22	655	7290.6	53	1934	8237.0

Therefore, we ran these instances through the IP preprocessor of CPLEX 8.1 before starting our procedure. Table 9 shows the results. In other instances, such as the Padberg–Crowder–Johnson instance `p0201`, we found that it was impossible to get an improvement of the LP value with the reformulation technique unless we started from a formulation augmented with strong cutting planes. To this end, we used CPLEX 8.1 to generate cutting planes at the root node of its computation. For technical reasons, we disabled all cut classes that lead to fractional coefficients. Table 10 shows the results for the instance `p0201`, where GUB cover cuts and cover cuts were applied. It also shows the results for the instance `p0548`, where only clique cuts were applied. We remark that CPLEX 8.1 can solve the latter instance in the root node, so we had to disable most cut classes for the experiment.

In all instances that we tested, we had to observe that after a number of iterations, the LP bound will not change any more, or only improve by tiny amounts. The reason is that the determinant of the optimal simplex basis of the problem increases during the reformulation method. This implies that the effect of the group reformulations with small moduli becomes smaller.

We also tried the dual algorithmic scheme on several market-split problems and their general-integer variant, the so-called banker's problem. Moreover, we tested it on three families of general-integer knapsack problems, which we describe below in the section on the primal effect. On all of these problems, the experience with the dual method was negative, as the LP bound did not change.

Some experiments were also carried out to check that the group reformulation provides augmenting vectors in the reformulation when we follow the primal algorithmic scheme. In Step 6 (Augmentation), if one uses an aggregated or path reformulation, the augmenting vectors are still hidden in the rows modeling the aggregation or in the rows modeling the flow conservation constraints. On the other hand, a disaggregated reformulation shows explicitly all the basic solutions to the group problem and therefore all the candidates for an augmentation. That is why we chose to perform disaggregated reformulations in our tests.

We first tested the search for augmentation on three families of general-integer knapsack problems with varying right-hand sides β:

$$\begin{aligned} &\min\{\, s : 101x_1 + 117x_2 + 120x_3 + 127x_4 + 113x_5 + s = \beta, \\ &\qquad x_1, \dots, x_5 \in \mathbb{Z}_+ \,\} \end{aligned} \tag{32a}$$

$$\begin{aligned} &\min\{\, s : 1011x_1 + 1179x_2 + 220x_3 + 327x_4 + 613x_5 \\ &\qquad + 1001x_6 + 1073x_7 + s = \beta, \\ &\qquad x_1, \dots, x_7 \in \mathbb{Z}_+ \,\} \end{aligned} \tag{32b}$$

$$\begin{aligned} &\min\{\, s : 7011x_1 + 9179x_2 + 2220x_3 + 2327x_4 + 3613x_5 \\ &\qquad + 8001x_6 + 2073x_7 + 10491x_8 + 20631x_9 + s = \beta, \\ &\qquad x_1, \dots, x_9 \in \mathbb{Z}_+ \,\} \end{aligned} \tag{32c}$$

The range of right-hand side values β was chosen to include problems with zero and non-zero optimal solution value.

We tested the search for augmentation in a very simple algorithmic setting. The computational experiment is designed to show whether a single disaggregated reformulation step improves the ability of a given augmentation procedure to find augmentation vectors, and to show the influence of the choice of the modulus. The augmentation procedure that we used is a version of a primal all-integer simplex method. To this end, we set up each of the problems as an all-integer simplex tableau representing the trivial solution $x^0 = 0$, $s^0 = \beta$. The augmentation procedure now considers non-basic columns of the integer tableau and weighted sums of two columns as augmentation vectors. Whenever an improving non-basic column with simplex ratio at least 1 is found, we apply an integral pivoting step to move to an all-integer simplex tableau representing the new solution; when necessary, an integer Gomory cut is added on which we then perform an integral pivot. When a weighted sum of two non-basic columns is an augmentation vector, we add it as a new non-basic column to the tableau and perform an integral pivoting step to pivot it into the basis. Again an integer Gomory cut is added if necessary. Our augmen-

Table 11
Augmentation for knapsack instances

	Problem (32a) $\beta = 300, \ldots, 999$		Problem (32b) $\beta = 4000, \ldots, 4999$		Problem (32c) $\beta = 22000, \ldots, 22999$	
Modulus	% opt.	Avg. gap	% opt.	Avg. gap	% opt.	Avg. gap
none	71.0	0.50	7.6	9.67	17.1	23.98
2	37.6	2.36	38.9	3.68	11.5	38.81
3	42.9	3.79	2.3	19.57	18.8	19.06
4	63.1	2.16	55.6	2.08	23.5	17.46
5	67.0	1.96	58.7	1.90	20.9	22.66
6	55.3	3.61	24.1	3.71	18.0	21.26
7	76.1	1.48	41.3	3.20	42.0	6.97
8	77.4	1.29	38.1	2.48	25.1	16.36
9	73.1	1.71	58.7	1.34	12.6	30.33
10	79.3	1.07	61.8	2.01	27.8	16.95
11	90.1	0.19	77.7	1.21	34.6	11.64
12	75.0	1.65	55.0	0.97	25.8	16.76
13	78.1	1.59	43.3	1.37	20.1	35.72
14	92.9	0.15	71.3	0.91	71.6	3.73
15	82.0	1.10	44.8	2.06	30.7	14.74
16	81.0	0.96	60.7	1.00	40.5	9.81
17	83.6	1.88	71.1	0.50	51.4	8.23
18	88.1	0.28	76.5	0.42	33.7	13.15
19	95.7	0.37	75.7	0.42	47.7	10.35
20	91.6	0.12	72.7	0.93	24.7	22.73
21	94.0	0.43	88.7	0.20	54.1	6.47
22	94.9	0.23	95.7	0.10	51.7	8.10
23	95.3	0.10	79.0	0.33	76.3	2.99

tation procedure does not perform degenerate simplex steps; it simply stops when it does not find an improving vector any more.

We first tested this augmentation procedure in the original formulation. Then we tested, independently for every modulus in the range from 2 to 23, the augmentation procedure in the formulation obtained by performing a single disaggregated reformulation step. We repeated this test for all right-hand sides in the chosen range. Table 11 shows the computational results. For each of the problem families, the left column in the table shows the percentage of the problems where we reached the optimal solution. The right column shows the absolute gap to the optimal solution, averaged over all problems. The results for the original formulation are shown in the first row of the table.

Table 12
Augmentation for the MIPLIB instance `lseu`, IP value 1120

Objective	Row	Modulus	Gap closed
1660	R123	10	35 %
1472	R123	10	66 %
1303			

Table 13
Augmentation for the MIPLIB instance `p0282`, IP value 258411

Objective	Row	Modulus	Gap closed
366 777	R1026	5	34 %
329 640	R1026	5	38 %
325 655	R1056	5	41 %
322 538			

For a single problem (with fixed right-hand side β), we cannot be sure of reaching a better solution if we choose a higher modulus, even if we choose a modulus that is an integer multiple of another modulus. The reason is that the solution constructed by the augmentation procedure depends on the order in which augmentation vectors are found. However, Table 11 shows that, on the average,

(i) we can obtain better solutions in the disaggregated reformulation than in the original formulation,
(ii) reformulations with larger moduli lead to the construction of better solutions than reformulations with small moduli.

We also tested the search for augmentation on some 0/1 instances of the MIPLIB. We note that finding augmentation vectors in these problems is much more complicated than for the knapsack problems, so we did not come up with an automatic procedure that was successful in finding reformulations that produce augmentation vectors. However, for some instances we found, by manual experiments, sequences of reformulations that led to significant improvements of the current solutions. After each reformulation that leads to the construction of an augmentation vector, we recompute an all-integer tableau representing the new solution in the original variable space, in order to keep a problem size for which disaggregated formulations are computable. The construction of an all-integer tableau representing the new solution is easy because of the 0/1 setting. The computation is stopped when no augmentation vector can be found on any row of the current tableau. The results are presented in Tables 12 and 13.

Conclusions. Computational experiments in the past suggested that one cannot expect a significant effect on the value of the LP relaxation from adding the facets of corner polyhedra corresponding to groups of low order. As the computations in this paper show, the same holds true if we perform reformulations based on irreducible solutions, rather than adding facets. We believe that in order to make effective use of the group relaxation, we need to find a way to deal with much larger groups than is possible today.

Even if we manage to work with much larger groups, we still cannot expect that the technique works on its own as a pure method to solve arbitrary integer programs. For instance, as we have already pointed out, group relaxations seem to be especially unsuited to certain bad problem formulations. Thus it should be combined with IP preprocessing and strong cutting planes.

There still appear to be numerous ways to be explored to combine reformulations based on group relaxations with other techniques, in order to successfully solve integer programs. One highly interesting aspect of Gomory's group relaxation is that the irreducible solutions of the corner polyhedron encode partial solutions to the underlying integer program. We still believe that this has potential for use in an augmentation procedure.

References

[1] K. Aardal, R. Weismantel, and L. Wolsey. Non-standard approaches to integer programming. *Discrete Applied Mathematics*, 123:5–74, 2002.

[2] J. Aráoz, L. Evans, R. E. Gomory, and E. L. Johnson. An approach to integer programming. *Math. Programming Series B*, 96(2):377–408, 2003.

[3] R. E. Gomory. Some polyhedra related to combinatorial problems. *Linear Algebra and Its Applications*, 2:451–558, 1969.

[4] R. E. Gomory and E. Johnson. Some continuous functions related to corner polyhedra I. *Math. Programming*, 3:23–85, 1972.

[5] R. E. Gomory and E. Johnson. Some continuous functions related to corner polyhedra II. *Math. Programming*, 3:359–389, 1972.

[6] R. E. Gomory and E. Johnson. The group problem and subadditive functions. *Math. Programming*, pages 157–184, 1973.

[7] R. E. Gomory and E. Johnson. T-space and cutting planes. *Math. Programming Series B*, 96:341–375, 2003.

[8] R. E. Gomory, E. Johnson, and L. Evans. Corner polyhedra and their connection with cutting planes. *Math. Programming Series B*, 96:321–339, 2003.

[9] G. A. Gorry, W. D. Northup, and J. F. Shapiro. Computational experience with a group theoretic integer programming algorithm. *Mathematical Programming*, 4:171–192, 1973.

[10] U.-U. Haus, M. Köppe, and R. Weismantel. A primal all-integer algorithm based on irreducible solutions. *Math. Programming Series B*, 96:205–246, 2003.

[11] M. Köppe. *Exact Primal Algorithms for General Integer and Mixed-Integer Linear Programs*. PhD thesis, Otto-von-Guericke-Universität Magdeburg, 2002. Published by Shaker Verlag, Aachen, 2003.

[12] M. Köppe. Tables of primitive partition identities. Available from `http://www.math.uni-magdeburg.de/~mkoeppe/art/ppi/`, 2003.

[13] L. A. Wolsey. A number theoretic reformulation and decomposition method for integer programming. *Discrete Mathematics*, 7:393–403, 1974.