Discrete Optimization 6 (2009) 242–270

Contents lists available at ScienceDirect

Discrete Optimization

journal homepage: www.elsevier.com/locate/disopt

Column basis reduction and decomposable knapsack problems

Bala Krishnamoorthy [a], Gábor Pataki [b,*]

[a] Department of Mathematics, Washington State University, United States

[b] Department of Statistics and Operations Research, UNC Chapel Hill, United States

ARTICLE INFO

Article history:
Received 13 November 2006
Received in revised form 17 December 2008
Accepted 22 January 2009
Available online 9 March 2009

Keywords:
Integer programming
Split disjunctions
Basis reduction

ABSTRACT

We propose a very simple preconditioning method for integer programming feasibility problems: replacing the problem

$$\begin{aligned} b' &\le Ax \le b \\ x &\in \mathbb{Z}^n \end{aligned}$$

with

$$\begin{aligned} b' &\le (AU)y \le b \\ y &\in \mathbb{Z}^n, \end{aligned}$$

where U is a unimodular matrix computed via *basis reduction*, to make the columns of AU short (i.e. have small Euclidean norm), and nearly orthogonal (see e.g. [Arjen K. Lenstra, Hendrik W. Lenstra, Jr., László Lovász, Factoring polynomials with rational coefficients, Mathematische Annalen 261 (1982) 515–534; Ravi Kannan, Minkowski's convex body theorem and integer programming, Mathematics of Operations Research 12 (3) (1987) 415–440]). Our approach is termed column basis reduction, and the reformulation is called rangespace reformulation. It is motivated by the technique proposed for equality constrained IPs by Aardal, Hurkens and Lenstra. We also propose a simplified method to compute their reformulation.

We also study a family of IP instances, called *decomposable knapsack problems (DKPs)*. DKPs generalize the instances proposed by Jeroslow, Chvátal and Todd, Avis, Aardal and Lenstra, and Cornuéjols et al. They are knapsack problems with a constraint vector of the form $pM + r$, with $p > 0$ and r integral vectors, and M a large integer. If the parameters are suitably chosen in DKPs, we prove

- hardness results, when branch-and-bound branching on individual variables is applied;
- that they are easy, if one branches on the constraint px instead; and
- that branching on the last few variables in either the rangespace or the AHL reformulations is equivalent to branching on px in the original problem.

We also provide recipes to generate such instances.

Our computational study confirms that the behavior of the studied instances in practice is as predicted by the theory.

* Corresponding author.
E-mail addresses: bkrishna@math.wsu.edu (B. Krishnamoorthy), gabor@unc.edu (G. Pataki).

1572-5286/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.disopt.2009.01.003

1. Introduction and overview of the main results

Basis reduction. Basis reduction (BR for short) is a fundamental technique in computational number theory, cryptography, and integer programming. If A is a real matrix with m rows, and n independent columns, the *lattice* generated by the columns of A is

$$\mathbb{L}(A) = \{Ax \mid x \in \mathbb{Z}^n\}. \tag{1.1}$$

The columns of A are called a *basis* of $\mathbb{L}(A)$. A square, integral matrix U is *unimodular* if $\det U = \pm 1$. Given A as above, BR computes a unimodular U such that the columns of AU are "short" and "nearly" orthogonal. The following example illustrates the action of BR:

$$A = \begin{pmatrix} 289 & 18 \\ 466 & 29 \\ 273 & 17 \end{pmatrix}, \qquad U = \begin{pmatrix} 1 & -15 \\ -16 & 241 \end{pmatrix}, \qquad AU = \begin{pmatrix} 1 & 3 \\ 2 & -1 \\ 1 & 2 \end{pmatrix}.$$

We have $\mathbb{L}(A) = \mathbb{L}(AU)$. In fact for two matrices A and B, $\mathbb{L}(A) = \mathbb{L}(B)$ holds, if and only if $B = AU$ for some U unimodular matrix (see e.g. Corollary 4.3a, [1]).

In this work we use two BR methods. The first is the Lenstra, Lenstra, and Lovász (LLL for short) reduction algorithm [2] which runs in polynomial time for rational lattices. The second is Korkhine–Zolotarev (KZ for short) reduction – see [3,4] – which runs in polynomial time for rational lattices only when the number of columns of A is fixed.

Basis reduction in Integer Programming. The first application of BR for integer programming is in Lenstra's IP algorithm that runs in polynomial time in fixed dimension, see [5]. Later IP algorithms which share polynomiality for a fixed number of variables also relied on BR: see, for instance Kannan's algorithm [6]; Barvinok's algorithm to count the number of lattice points in fixed dimension [7,8], and its variant proposed by de Loera et al. in [9]. A related method in integer programming is *generalized basis reduction* due to Lovász and Scarf [10]. For its implementation see Cook et al. in [11]. Mehrotra and Li in [12] proposed a modification and implementation of Lenstra's method, and of generalized basis reduction. For surveys, we refer to [1,13].

A computationally powerful reformulation technique based on BR was proposed by Aardal, Hurkens, and Lenstra in [14]. They reformulate an equality constrained IP feasibility problem

$$\begin{aligned} &Ax = b \\ &\ell \le x \le u \\ &x \in \mathbb{Z}^n \end{aligned} \tag{1.2}$$

with integral data, and A having m independent rows, as follows: they find a matrix B, and a vector x_b with $[B, x_b]$ having short, and nearly orthogonal columns, x_b satisfying $Ax_b = b$, and the property

$$\{x \in \mathbb{Z}^n \mid Ax = 0\} = \{B\lambda \mid \lambda \in \mathbb{Z}^{n-m}\}. \tag{1.3}$$

The reformulated instance is

$$\begin{aligned} &\ell - x_b \le B\lambda \le u - x_b \\ &\lambda \in \mathbb{Z}^{n-m}. \end{aligned} \tag{1.4}$$

For several families of hard IPs, the reformulation (1.4) turned out to be much easier to solve for commercial MIP solvers than the original one; a notable family was the *marketshare* problems of Cornuéjols and Dawande [15]. The solution of these instances using the above reformulation technique is described by Aardal, Bixby, Hurkens, Lenstra and Smeltink in [16].

The matrix B and the vector x_b are found as follows. Assume that A has m independent rows. They embed A and b in a matrix, say D, with $n + m + 1$ rows, and $n + 1$ columns, with some entries depending on two large constants N_1 and N_2:

$$D = \begin{pmatrix} I_n & 0_{n\times 1} \\ 0_{1\times n} & N_1 \\ N_2 A & -N_2 b \end{pmatrix}. \tag{1.5}$$

The lattice generated by D looks like

$$\mathbb{L}(D) = \left\{ \begin{pmatrix} x \\ N_1 x_0 \\ N_2(Ax - bx_0) \end{pmatrix} \,\middle|\, \begin{pmatrix} x \\ x_0 \end{pmatrix} \in \mathbb{Z}^{n+1} \right\}, \tag{1.6}$$

in particular, all vectors in a reduced basis of $\mathbb{L}(D)$ have this form.

For instance, if $A = [2, 2, 2]$, and $b = 3$, (this corresponds to the infeasible IP $2x_1 + 2x_2 + 2x_3 = 3$, $x_i \in \{0, 1\}$, when the bounds on x are 0 and e), then

$$\mathbb{L}(D) = \left\{ \begin{pmatrix} x \\ N_1 x_0 \\ N_2(2x_1 + 2x_2 + 2x_3 - 3x_0) \end{pmatrix} \,\middle|\, \begin{pmatrix} x \\ x_0 \end{pmatrix} \in \mathbb{Z}^3 \times \mathbb{Z}^1 \right\}. \tag{1.7}$$

It is shown in [14], that if $N_2 \gg N_1 \gg 1$ are suitably chosen, then in a reduced basis of $\mathbb{L}(D)$

- $n - m$ vectors arise from some $\binom{x}{x_0}$ with $Ax = bx_0,\ x_0 = 0$, and
- 1 vector will arise from an $\binom{x}{x_0}$ with $Ax = bx_0,\ x_0 = 1$.

So the x vectors from the first group can form the columns of B, and the x from the last can serve as x_b. If LLL- or KZ-reduction (the precise definition is given later) is used to compute the reduced basis of $\mathbb{L}(D)$, then B is a basis reduced in the same sense.

Followup papers on this reformulation technique were written by Louveaux and Wolsey [17], and Aardal and Lenstra [18, 19].

Questions to address. The speedups obtained by the Aardal–Hurkens–Lenstra (AHL) reformulation lead to the following questions:

(Q1) Is there a similarly effective reformulation technique for *general* (not equality constrained) IPs?

(Q2) Why does the reformulation work? Can we analyse its action on a reasonably wide class of difficult IPs?

More generally, one can ask:

(Q3) What kind of integer programs are hard for a certain standard approach, such as branch-and-bound branching on individual variables, and easily solvable by a different approach?

As to (Q1), one could simply add slacks to turn inequalities into equalities, and then apply the AHL reformulation. This option, however, has not been studied. The mentioned papers emphasize the importance of reducing the dimension of the space, and of the full-dimensionality of the reformulation. Moreover, reformulating an IP with n variables, m dense constraints, and some bounds in this way leads to a D matrix (see (1.5)) with $n + 2m + 1$ rows and $n + m + 1$ columns.

A recent paper of Aardal and Lenstra [18,19] addressed the second question. They considered an equality constrained knapsack problem with unbounded variables

$$\begin{aligned} &ax = \beta \\ &x \geq 0 \\ &x \in \mathbb{Z}^n, \end{aligned} \tag{KP-EQ}$$

with the constraint vector a decomposing as $a = pM + r$, with $p, r \in \mathbb{Z}^n,\ p > 0,\ M$ a positive integer, under the following assumption:

Assumption 1. (1) $r_j/p_j = \max_{i=1,\dots,n} \{r_i/p_i\},\ r_k/p_k = \min_{i=1,\dots,n} \{r_i/p_i\}$.
(2) $a_1 < a_2 < \cdots < a_n$;
(3) $\sum_{i=1}^n |r_i| < 2M$;
(4) $M > 2 - r_j/p_j$;
(5) $M > r_j/p_j - 2r_k/p_k$.

They proved the following:

(1) Let Frob(a) denote the Frobenius number of $a_1, \dots, a_n$, i.e., the largest β integer for which (KP-EQ) is infeasible. Then

$$\mathrm{Frob}(a) \geq \frac{(M^2 p_j p_k + M(p_j r_k + p_k r_j) + r_j r_k)(1 - \frac{2}{M + r_j/p_j})}{p_k r_j - p_j r_k} - (M + r_j/p_j). \tag{1.8}$$

(2) In the reformulation (1.4), if we denote the last column of B by b_{n-1}, then

$$\|b_{n-1}\| \geq \frac{\|a\|}{\sqrt{\|p\|^2 \|r\|^2 - (pr^{\mathrm{T}})^2}}. \tag{1.9}$$

It is argued in [18] that the large right-hand side explains the hardness of the corresponding instance, and that the large norm of b_{n-1} explains why the reformulation is easy: if we branch on b_{n-1} in the reformulation, only a small number of nodes are created in the branch-and-bound tree. In Section 8 we show that there is a gap in the proof of (1.9). Here we also show an instance of a bounded polyhedron where the columns of the constraint matrix are LLL-reduced, but branching on a variable corresponding to the longest column produces exponentially many nodes.

Among the other papers that motivated this research, two are "classical": [20], and parts of [21]. They all address the hardness question in (Q3), and the easiness is straightforward to show.

Jeroslow's knapsack instance in [20] is

$$\begin{aligned} \min \quad & x_{n+1} \\ \text{st.} \quad & 2\sum_{i=1}^n x_i + x_{n+1} = n \\ & x_i \in \{0, 1\}\ (i = 1, \dots, n+1), \end{aligned} \tag{1.10}$$

where n is an odd integer. The optimal solution of (1.10) is trivially 1, but branch-and-bound requires an exponential number of nodes to prove this, if we branch on individual variables.

In [21] Todd and Avis constructed knapsack problems of the form

$$\begin{array}{ll} \max & ax \\ \text{st.} & ax \le \beta \\ & x \in \{0,1\}^n, \end{array} \tag{1.11}$$

with a decomposing as $a = eM + r$ (M and r are chosen differently in the Todd- and in the Avis-problems). They showed that these instances exhibit a similar behavior.

Though this is not mentioned in [20], or [21], it is straightforward to see that the Jeroslow-, Todd-, and Avis-problems can be solved at the rootnode, if one branches on the constraint $\sum_{i=1}^n x_i$ instead of branching on the x_i.

A more recent work that motivated us is [22]. Here a family of instances of the form

$$\begin{array}{ll} \max & ax \\ s.t. & ax \le \beta \\ & x \in \mathbb{Z}_+^n, \end{array} \tag{1.12}$$

with a decomposing as $a = pM + r$, where p and r are integral vectors, and M is a positive integer, was proposed. The authors used $\text{Frob}(a)$ as β in (1.12). These problems turned out to be hard for commercial MIP solvers, but easy if one uses a test-set approach. One can also verify computationally that if one branches on the constraint px in these instances, then feeds the resulting subproblems to a commercial solver, they are solved quite quickly.

Contributions, and organization of the paper. We first fix basic terminology. When branch-and-bound (B&B for short) branches on individual variables, we call the resulting algorithm *ordinary branch-and-bound*.

Definition 1. If p is an integral vector, and k an integer, then the logical expression $px \le k \,\vee\, px \ge k+1$ is called a *split disjunction*. We say that the infeasibility of an integer programming problem is proven by $px \le k \,\vee\, px \ge k+1$, if both polyhedra $\{x \mid px \le k\}$ and $\{x \mid px \ge k+1\}$ have empty intersection with the feasible set of its LP relaxation.

We say that the infeasibility of an integer programming problem is proven by branching on px, if px is nonintegral for all x in its LP relaxation.

We call a knapsack problem with weight vector a a *decomposable knapsack problem* (DKP for short), if $a = pM + r$, where p and r are integral vectors, $p > 0$, and M is a large integer. We could not find a good definition of DKPs which would not be either too restrictive, or too permissive, as far as how large M should be. However, we will show how to find M and the bounds for given p and r so the resulting DKP has interesting properties.

The paper focuses on the interplay of these concepts, and their connection to IP reformulation techniques.

(1) In the rest of this section we describe a simple reformulation technique, called the *rangespace reformulation* for arbitrary integer programs. The dimension of the reformulated instance is the same as of the original. We also show a simplified method to compute the AHL reformulation, and illustrate how the reformulations work on some simple instances.

For a convenient overview of the paper we state Theorems 1 and 2 as a sample of the main results.

(2) In Section 2 we consider knapsack feasibility problems with a positive weight vector. We show a somewhat surprising result: if the infeasibility of such a problem is proven by $px \le k \vee px \ge k+1$, with p positive, then a lower bound follows on the number of nodes that must be enumerated by ordinary B&B to prove infeasibility. So, easiness for constraint branching implies hardness for ordinary B&B.

(3) In Section 3 we give two recipes to find DKPs, whose infeasibility is proven by the split disjunction $px \le k \vee px \ge k+1$. Split disjunctions for deriving cutting planes have been studied e.g. in [23–26]. This paper seems to be the first systematic study of knapsack problems with their infeasibility having such a short certificate.

Thus (depending on the parameters), their hardness for ordinary B&B follows using the results of Section 2. We show that several well-known hard integer programs from the literature, such as Jeroslow's problem [20], and the Todd- and Avis-problems from [21] can be found using Recipe 1. Recipe 2 generates instances of type (KP-EQ), with a short proof (a split disjunction) of their infeasibility.

So this section provides a unifying framework to show the hardness of instances (for ordinary B&B) which are easy for constraint branching. These results add to the understanding of hard knapsacks described in [20,21,18], as follows. We deal with arbitrary knapsacks, both with bounded, and unbounded variables; we give explicit lower bounds on the number of nodes that ordinary B&B must enumerate, which is done in [21] for the Todd- and Avis instances; and our instances have a short, split disjunction certificate.

Using the recipes we generate some new, interesting examples. For example, Example 8 is a knapsack problem whose infeasibility is proven by a single split disjunction, but ordinary B&B needing a *superexponential* number of nodes to prove the same. Example 5 reverses the role of the two vectors in the Avis-problem, and gives an instance which is computationally more difficult than the original.

(4) In Section 4 we extend the lower bound (1.8) in two directions. We first show that for given p and r integral vectors, and sufficiently large M, there is a range of β integers for which the infeasibility of (KP-EQ) with $a = pM + r$ is proven by branching on px. The smallest such integer is essentially the same as the lower bound in (1.8).

Any such β right-hand side is a lower bound on $\text{Frob}(a)$, with a short certificate of being a lower bound, i.e. a split disjunction certificate of the infeasibility of (KP-EQ).

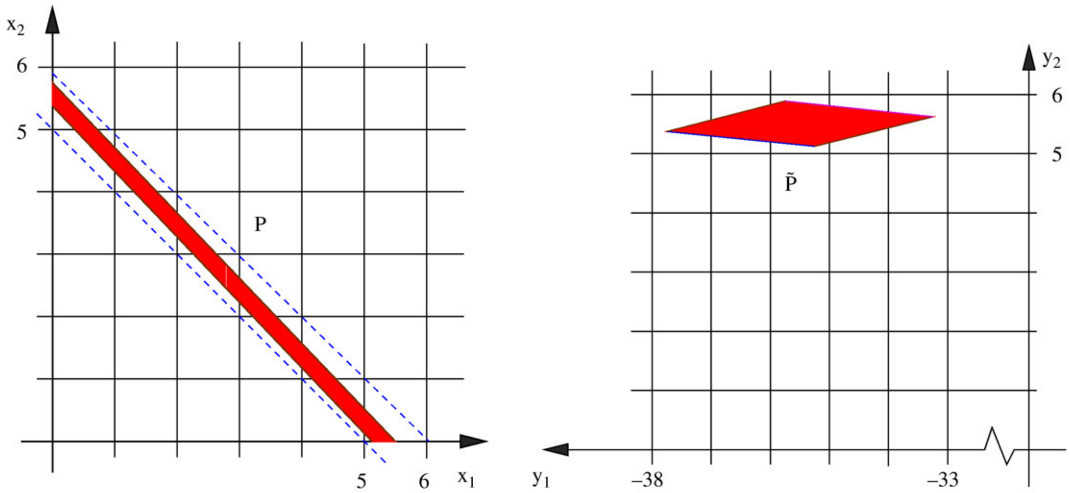

Fig. 1. The polyhedron in Example 1 before and after reformulation.

We then study the *largest* integer for which the infeasibility of (KP-EQ) with $a = pM + r$, and M sufficiently large, is proven by branching on px. We call this number the p-branching Frobenius number, and give a lower and an upper bound on it.

(5) In Section 5 we show some basic results on the geometry of the reformulations. Namely, given a vector say c, we find a vector which achieves the same width in the reformulation, as c does in the original problem.

(6) Section 6.1 shows why DKPs become easy after the rangespace reformulation is applied. In Theorem 10 we prove that if M is sufficiently large, and the infeasibility of a DKP is proven by branching on px, then the infeasibility of the reformulated problem is proven by branching on the last few variables in the reformulation. How many "few" is will depend on the magnitude of M. We give a similar analysis for the AHL reformulation in Section 6.2.

Here we remark that a method which *explicitly* extracts "dominant" directions in an integer program was proposed by Cornuéjols et al. in [22].

(7) In Section 7 we present a computational study that compares the performance of an MIP solver before and after the application of the reformulations on certain DKP classes.

(8) In Section 8 we point out a gap in the proof of (1.9), and show a correction. We also describe a bounded polyhedron with the columns of the constraint matrix forming an LLL-reduced basis, where branching on a variable corresponding to the longest column creates exponentially many subproblems.

The rangespace reformulation. Given

$$\begin{aligned} b' &\le Ax \le b \\ x &\in \mathbb{Z}^n, \end{aligned} \tag{IP}$$

we compute a unimodular (i.e. integral, with ± 1 determinant) matrix U that makes the columns of AU short, and nearly orthogonal; U is computed using basis reduction, either the LLL- or the KZ-variant (our analysis will be unified). We then recast (IP) as

$$\begin{aligned} b' &\le (AU)y \le b \\ y &\in \mathbb{Z}^n. \end{aligned} \tag{$\widetilde{\text{IP}}$}$$

The dimension of the problem is unchanged; we will call this technique *rangespace reformulation*.

Example 1. Consider the infeasible problem

$$\begin{aligned} 106 &\le 21x_1 + 19x_2 \le 113 \\ 0 &\le x_1, x_2 \le 6 \\ x_1&, x_2 \in \mathbb{Z}, \end{aligned} \tag{1.13}$$

with the feasible set of the LP-relaxation depicted on the first picture in Fig. 1. In a sense it is both hard, and easy. On the one hand, branching on either variable will produce at least 5 feasible nodes. On the other hand, the maximum and the minimum of $x_1 + x_2$ over the LP relaxation of (1.13) are 5.94, and 5.04, respectively, thus "branching" on this constraint proves infeasibility at the root node.

When the rangespace reformulation with LLL-reduction is applied, we have

$$A = \begin{pmatrix} 21 & 19 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad U = \begin{pmatrix} -1 & -6 \\ 1 & 7 \end{pmatrix}, \qquad AU = \begin{pmatrix} -2 & 7 \\ -1 & -6 \\ 1 & 7 \end{pmatrix},$$

so the reformulation is

$$\begin{aligned} 106 \ &\leq -2y_1 + 7y_2 \leq \ 113 \\ 0 \ &\leq -y_1 - 6y_2 \leq \ 6 \\ 0 \ &\leq y_1 + 7y_2 \leq \ 6 \\ y_1, y_2 &\in \mathbb{Z}. \end{aligned} \tag{1.14}$$

Branching on y_2 immediately proves infeasibility, as the second picture in Fig. 1 shows. The linear constraints of (1.14) imply

$$5.04 \leq y_2 \leq 5.94. \tag{1.15}$$

These bounds are the same as the bounds on $x_1 + x_2$. This fact will follow from Theorem 7, a general result about how the widths are related along certain directions in the original and the reformulated problems.

Example 2. This example is a simplification of Jeroslow's problem (1.10) from [20]. Let n be a positive odd integer. The problem

$$\begin{aligned} &2\sum_{i=1}^{n} x_i = n \\ &0 \leq x \leq e \\ &x \in \mathbb{Z}^n \end{aligned} \tag{1.16}$$

is integer infeasible.

Ordinary B&B (i.e. B&B branching on the x_i variables) must enumerate at least $2^{(n-1)/2}$ nodes to prove infeasibility. To see this, suppose that at most $(n-1)/2$ variables are fixed to either 0 or 1. The sum of the coefficients of these variables is at most $n-1$, while the sum of the coefficients of the free variables is at least $n+1$. Thus, we can set some free variable(s) to a possibly fractional value to get an LP-feasible solution.

On the other hand, denoting by e the vector of all ones, the maximum and minimum of ex over the LP relaxation of (1.16) is $n/2$, thus branching on ex proves infeasibility at the root node.

For the rangespace reformulation using LLL-reduction, we have

$$A = \begin{pmatrix} 2e_{1\times n} \\ I_n \end{pmatrix}, \qquad U = \begin{pmatrix} I_{n-1} & 0_{(n-1)\times 1} \\ -e_{1\times(n-1)} & 1 \end{pmatrix}, \qquad AU = \begin{pmatrix} 0_{1\times(n-1)} & 2 \\ I_{n-1} & 0_{(n-1)\times 1} \\ -e_{1\times(n-1)} & 1 \end{pmatrix},$$

thus the reformulation is

$$\begin{aligned} &2y_n = n \\ &0 \leq y_1, \ldots, y_{n-1} \leq 1 \\ &0 \leq -\sum_{i=1}^{n-1} y_i + y_n \leq 1 \\ &y \in \mathbb{Z}^n. \end{aligned} \tag{1.17}$$

So branching on y_n immediately implies the infeasibility of (1.17), and thus of (1.16).

A simplified method to compute the AHL reformulation. Rangespace reformulation only affects the constraint matrix, so it can be applied unchanged, if some of the two-sided inequalities in (IP) are actually equalities, as in Example 2. We can still choose a different way of reformulating the problem. Suppose that

$$A_1 x = b_1 \tag{1.18}$$

is a system of equalities contained in the constraints of (IP), and assume that A_1 has m_1 rows. First compute an integral matrix B_1,

$$\{x \in \mathbb{Z}^n \mid A_1 x = 0\} = \{B_1 \lambda \mid \lambda \in \mathbb{Z}^{n-m_1}\},$$

and an integral vector x_1 with $Ax_1 = b_1$. B_1 and x_1 can be found by a Hermite Normal Form computation—see e.g. [1], page 48.

In general, the columns of $[B_1, x_1]$ will *not* be reduced. So, we substitute $B_1\lambda + x_1$ into the part of (IP) excluding (1.18), and apply the rangespace reformulation to the resulting system.

If the system (1.18) contains all the constraints of the integer program other than the bounds, then this way we get the AHL reformulation.

Example 3 (Example 2 *Continued*). In this example (1.16) has no solution over the integers, irrespective of the bounds.
However, we can rewrite it as

$$\begin{aligned} &2\sum_{i=1}^{n} x_i + x_{n+1} = n \\ &0 \le x_{1:n} \le e \\ &-1/2 \le x_{n+1} \le 1/2 \\ &x \in \mathbb{Z}^n. \end{aligned} \tag{1.19}$$

The x integer vectors that satisfy the first equation in (1.19) can be parametrized with $\lambda \in \mathbb{Z}^n$ as

$$\begin{aligned} x_1 &= \lambda_1 + \cdots + \lambda_n \\ x_2 &= -\lambda_1 \\ &\vdots \\ x_n &= -\lambda_{n-1} \\ x_{n+1} &= -2\lambda_n + n. \end{aligned} \tag{1.20}$$

Substituting (1.20) into the bounds of (1.19) we obtain the reformulation

$$\begin{aligned} &0 \le \sum_{j=1}^{n-1} \lambda_j + \lambda_n \le 1 \\ &0 \le -\lambda_j \le 1 \ (j = 1, \ldots, n-1) \\ &-1/2 \le -2\lambda_n + n \le 1/2 \\ &\lambda \in \mathbb{Z}^n. \end{aligned} \tag{1.21}$$

The columns of the constraint matrix of (1.21) are already reduced in the LLL-sense. The last constraint is equivalent to

$$(n+1)/2 - 3/4 \le \lambda_n \le (n+1)/2 - 1/4, \tag{1.22}$$

so the infeasibility of (1.21) and thus of (1.19) is proven by branching on λ_n.

Right-hand side reduction. On several instances we found that reducing the right-hand side in (IP) yields an even better reformulation. To do this, we rewrite (IP) as

$$\begin{aligned} Fx &\le f \\ x &\in \mathbb{Z}^n, \end{aligned} \tag{IP2}$$

then reformulate the latter as

$$\begin{aligned} (FU)y &\le f - (FU)x_r \\ y &\in \mathbb{Z}^n, \end{aligned} \tag{$\tilde{\text{IP2}}$}$$

where the unimodular U is again computed by basis reduction, and $x_r \in \mathbb{Z}^n$ to make $f - (FU)x_r$ short, and near orthogonal to the columns of FU. For the latter task, we may use – for instance – Babai's algorithm [27] to find x_r, so that $(FU)x_r$ is a nearly closest vector to f in the lattice generated by the columns of F.

It is worth to do this, if the original constraint matrix, and right-hand side (rhs) *both* have large numbers. Since the rangespace reformulation reduces the matrix coefficients, leaving large numbers in the rhs may lead to numerical instability. Our analysis, however, will rely only on the reduction of the constraint matrix.

Rangespace, and AHL reformulation. To discuss the connection of these techniques, we assume for simplicity that right-hand-side reduction is not applied.

Suppose that A is an integral matrix with m independent rows, and b is an integral column vector with m components. Then the equality constrained IP

$$\begin{aligned} &Ax = b \\ &\ell \le x \le u \\ &x \in \mathbb{Z}^n \end{aligned} \tag{1.23}$$

has another, natural formulation:

$$\begin{aligned} &\ell \leq B\lambda + x_b \leq u \\ &\lambda \in \mathbb{Z}^{n-m}, \end{aligned} \tag{1.24}$$

where

$$\{x \in \mathbb{Z}^n \mid Ax = 0\} = \{B\lambda \mid \lambda \in \mathbb{Z}^{n-m}\}, \tag{1.25}$$

and x_b satisfies $Ax_b = b$. The matrix B can be constructed from A using an HNF computation.

Clearly, to (1.23) we can apply

- the rangespace reformulation (whether the constraints are inequalities, or equalities), or
- the AHL method, which is equivalent to applying the rangespace reformulation to (1.24).

So, on (1.23) the rangespace reformulation method can be viewed as a "primal" and the AHL reformulation as a "dual" method. The somewhat surprising fact is, that for a fairly large class of problems *both* work, both theoretically, and computationally. When both methods are applicable, we did not find a significant difference in their performance on the tested problem instances.

An advantage of the rangespace reformulation is its simplicity. For instance, there is a one-to-one correspondence between "thin" branching directions in the original, and the reformulated problems, so in this sense the geometry of the feasible set is preserved. The correspondence is described in Theorem 7 in Section 5. The situation is more complicated for the AHL method, and correspondence results are described in Theorems 8 and 9. These results use ideas from, and generalize Theorem 4.1 in [12].

In a sense the AHL method can be used to simulate the rangespace method on an *inequality constrained* problem: we can simply add slacks beforehand. However:

- the rangespace reformulation can be applied to an equality constrained problem as well, where there are no slacks;
- the main point of our paper is not simply presenting a reformulation technique, but *analysing* it. The analysis must be carried out separately for the rangespace and AHL reformulations. In particular, the bounds on M that ensure that branching on the "backbone" constraint px in (KP) will be mimicked by branching on a small number of individual variables in the reformulation will be smaller in the case of rangespace reformulation.

Using the rangespace reformulation is also natural when dealing with an optimization problem of the form

$$\begin{aligned} &\max cx \\ &\text{s.t.} \quad b' \leq Ax \leq b \\ &x \in \mathbb{Z}^n. \end{aligned} \tag{IP-OPT}$$

Of course, we can reduce solving (IP-OPT) to a sequence of feasibility problems.

A simpler method is solving (IP-OPT) by *direct reformulation*, i.e. by solving

$$\begin{aligned} &\max \tilde{c}y \\ &\text{st.} \quad b' \leq \tilde{A}y \leq b \\ &y \in \mathbb{Z}^n, \end{aligned} \tag{$\widetilde{\text{IP-OPT}}$}$$

where

$$\tilde{c} = cU, \quad \tilde{A} = AU,$$

with U having been computed to make the columns of

$$\begin{pmatrix} c \\ A \end{pmatrix} U$$

reduced.

Some other reformulation methods. Among early references, the all-integral simplex algorithm of Gomory [28] can be viewed as a reformulation method. Bradley in [29] studied integer programs connected via unimodular transformations, akin to how the rangespace reformulation works. However, the transformations in [29] do not arise from basis reduction.

The Integral Basis Method [30] has two reformulation steps: in the first an integral basis of an IP from a nonintegral basis of the LP relaxation is found. In the second, an augmentation vector leading to a better integral solution is found, or shown not to exist. Haus in his dissertation [31] studied the question of how to derive such augmentation vectors for general IPs.

Notation. Vectors are denoted by lower case letters. In notation we do not distinguish between row and column vectors; the distinction will be clear from the context. Occasionally, we write $\langle x, y\rangle$ for the inner product of vectors x and y.

We denote the sets of nonnegative, and positive integers by $\mathbb{Z}_+$, and $\mathbb{Z}_{++}$, respectively. The sets of nonnegative, and positive integral n-vectors are denoted by $\mathbb{Z}^n_+$, and $\mathbb{Z}^n_{++}$, respectively. If n a positive integer, then N is the set $\{1, \ldots, n\}$. If S is a subset of N, and v an n-vector, then $v(S)$ is defined as $\sum_{i \in S} v_i$.

For a matrix A we use a Matlab-like notation, and denote its jth row, and column by $A_{j,:}$ and $A_{:,j}$, respectively. Also, we denote the subvector $(a_k, \dots, a_\ell)$ of a vector a by $a_{k:\ell}$.

For $p \in \mathbb{Z}^n_{++}$, and an integer k we write

$$\ell(p,k) = \max\{\ell \mid p(F) \le k, \text{ and } p(N \setminus F) \ge k+1\,\forall F \subseteq N, \ |F| = \ell\}. \tag{1.26}$$

The definition implies that $\ell(p,k) = 0$ if $k \le 0$, or $k \ge \sum_i p_i$, and $\ell(p,k)$ is large if the components of p are small relative to k, and not too different from each other. For example, if $p = e, \ k < n/2$, then $\ell(p,k) = k$.

Sometimes $\ell(p,k)$ is not easy to compute exactly, but we can use a good lower bound, which is usually easy to find. For instance, let n be an integer divisible by 4, $p = (1, 2, \dots, n)$. The first $3n/4$ components of p sum to strictly more than $(\sum_{i=1}^n p_i)/2$, and the last $n/4$ sum to strictly less than this. Since the components of p are ordered increasingly, it follows that

$$\ell(p, n(n+1)/4) \ge n/4.$$

On the other hand, $\ell(p,k)$ can be zero, even if k is positive. For example, if p is *superincreasing*, i.e. $p_i > p_1 + \cdots + p_{i-1}$ for $i = 2, \dots, n$, then it is easy to see that $\ell(p,k) = 0$ for any positive integer k.

Knapsack problems. We will study knapsack feasibility problems

$$\begin{aligned} \beta_1 \le\ & ax \le \beta_2 \\ 0 \le\ & x \le u \\ & x \in \mathbb{Z}^n. \end{aligned} \tag{KP}$$

In the rest of the paper for the data of (KP) we will use the following assumptions, that we collect here for convenience:

Assumption 2. The row vectors a, u are in $\mathbb{Z}^n_{++}$. We allow some or all components of u to be $+\infty$. If $u_i = +\infty$, and $\alpha > 0$, then we define $\alpha u_i = +\infty$, and if $b \in \mathbb{Z}^n_{++}$ is a row vector, then we define $bu = +\infty$. We will assume $0 < \beta_1 \le \beta_2 < au$.

Recall the definition of a decomposable knapsack problem from Definition 1. For the data vectors p and r from which we construct a we will occasionally (but not always) assume

Assumption 3. $p \in \mathbb{Z}^n_{++}, \ r \in \mathbb{Z}^n$, p is not a multiple of r, and

$$r_1/p_1 \le \cdots \le r_n/p_n. \tag{1.27}$$

Examples 1 and 2 continued. The problems (1.13) and (1.16) are DKPs with

$$\begin{aligned} p &= (1,1), \\ r &= (1,-1), \\ u &= (6,6), \\ M &= 20, \\ a &= pM + r = (21, 19), \end{aligned}$$

and

$$\begin{aligned} p &= e, \\ r &= 0, \\ u &= e, \\ M &= 2, \\ a &= pM + r = 2e, \end{aligned}$$

respectively.

Width and integer width

Definition 2. Given a polyhedron Q, and an integral vector c, the width and the integer width of Q in the direction of c are

$$\begin{aligned} \text{width}(c, Q) &= \max\{cx \mid x \in Q\} - \min\{cx \mid x \in Q\}, \\ \text{iwidth}(c, Q) &= \lfloor \max\{cx \mid x \in Q\} \rfloor - \lceil \min\{cx \mid x \in Q\} \rceil + 1. \end{aligned}$$

If an integer programming problem is labeled by (P), and c is an integral vector, then with some abuse of notation we denote by width$(c, \text{(P)})$ the width of the LP-relaxation of (P) in the direction c, and the meaning of iwidth$(c, \text{(P)})$ is similar.

The quantity iwidth(c, Q) is the number of nodes generated by B&B when branching on the constraint cx.

Basis reduction. Recall the definition of a lattice generated by the columns of a rational matrix A from (1.1). Suppose

$$B = [b_1, \ldots, b_n], \tag{1.28}$$

with $b_i \in \mathbb{Z}^m$. Due to the nature of our application, we will generally have $n \leq m$. While most results in the literature are stated for full-dimensional lattices, it is easy to see that they actually apply to the general case. Let $b_1^*, \ldots, b_n^*$ be the Gram–Schmidt orthogonalization of $b_1, \ldots, b_n$, that is

$$b_i = \sum_{j=1}^{i} \mu_{ij} b_j^*, \tag{1.29}$$

with

$$\begin{aligned} &\mu_{ii} = 1 \quad (i = 1, \ldots, n), \\ &\mu_{ij} = b_i^{\mathrm{T}} b_j^* / \|b_j^*\|^2 \quad (i = 1, \ldots, n;\ j = 1, \ldots, i-1). \end{aligned} \tag{1.30}$$

We call $b_1, \ldots, b_n$ *LLL-reduced* if

$$|\mu_{ij}| \leq \frac{1}{2} \quad (1 \leq j < i \leq n), \tag{1.31}$$

$$\|\mu_{i,i-1} b_{i-1}^* + b_i^*\|^2 \geq \frac{3}{4} \|b_{i-1}^*\|^2. \tag{1.32}$$

An LLL-reduced basis can be computed in polynomial time for varying n.

Define the truncated sums

$$b_i(k) = \sum_{j=k}^{i} \mu_{ij} b_j^* \quad (1 \leq k \leq i \leq n), \tag{1.33}$$

and for $i = 1, \ldots, n$ let L_i be the lattice generated by

$$b_i(i), b_{i+1}(i), \ldots, b_n(i).$$

We call $b_1, \ldots, b_n$ *Korkhine–Zolotarev reduced* (KZ-reduced for short) if $b_i(i)$ is the shortest lattice vector in L_i for all i, and (1.31) holds. Since $L_1 = L$ and $b_1(1) = b_1$, in a KZ-reduced basis the first vector is the shortest vector of L. Computing the shortest vector in a lattice is expected to be hard, though it is not known to be NP-hard. It can be done in polynomial time when the dimension is fixed, and so can be computing a KZ-reduced basis.

Definition 3. Given a BR method (for instance LLL, or KZ), suppose there is a constant c_n dependent only on n with the following property: for all full-dimensional lattices $\mathbb{L}(A)$ in $\mathbb{Z}^n$, and for all reduced bases $\{b_1, \ldots, b_n\}$ of $\mathbb{L}(A)$,

$$\max\{\|b_1\|, \ldots, \|b_i\|\} \leq c_n \max\{\|d_1\|, \ldots, \|d_i\|\} \tag{1.34}$$

for all $i \leq n$, and any choice of linearly independent $d_1, \ldots, d_i \in \mathbb{L}(A)$. We will then call c_n the *reduction factor* of the BR method.

The reduction factors of LLL- and KZ-reduction are $2^{(n-1)/2}$ (see [2]) and $\sqrt{n}$ (see [4]), respectively. For KZ-reduced bases, [4] gives a better bound, which depends on i, but for simplicity, we use $\sqrt{n}$.

The kth successive minimum of the lattice $\mathbb{L}(A)$ is

$$\Lambda_k(\mathbb{L}(B)) = \min\{t \mid \exists\, k \text{ linearly independent vectors in } \mathbb{L}(A) \text{ with norm at most } t\}.$$

So (1.34) can be rephrased as

$$\max\{\|b_1\|, \ldots, \|b_i\|\} \leq c_n \Lambda_i(\mathbb{L}(A)) \quad \text{for all } i \leq n. \tag{1.35}$$

Other notation. Given an integral matrix C with independent rows, the *null lattice*, or *kernel lattice* of C is

$$\mathbb{N}(C) = \{v \in \mathbb{Z}^n \mid Cv = 0\}. \tag{1.36}$$

For vectors $f,\ p,$ and u we write

$$\begin{aligned} \max(f, p, \ell, u) &= \max\{fx \mid px \leq \ell,\ 0 \leq x \leq u\}, \\ \min(f, p, \ell, u) &= \min\{fx \mid px \geq \ell,\ 0 \leq x \leq u\}. \end{aligned} \tag{1.37}$$

Theorems 1 and 2 below give a sample of our results from the following sections. The overall results of the paper are more detailed, but Theorems 1 and 2 are a convenient sample to first look at.

Theorem 1. *Let $p \in \mathbb{Z}^n_{++}$, $r \in \mathbb{Z}^n$, and k and M integers with*

$$\begin{aligned} 0 &\le k < \sum_{i=1}^n p_i, \\ M &> 2\sqrt{n}(\|r\|+1)^2\|p\|+1. \end{aligned} \tag{1.38}$$

Then there are β_1, and β_2 integers that satisfy

$$kM + \sqrt{n}\|r\| < \beta_1 \le \beta_2 < -\sqrt{n}\|r\| + (k+1)M, \tag{1.39}$$

and for all such (β_1, β_2) the problem (KP) with $a = pM + r$ and $u = e$ has the following properties:

(1) *Its infeasibility is proven by $px \le k \vee px \ge k+1$.*
(2) *Ordinary B&B needs at least $2^{\ell(p,k)}$ nodes to prove its infeasibility regardless of the order in which the branching variables are chosen. (Recall the definition of $\ell(p,k)$ from (1.26)).*
(3) *The infeasibility of its rangespace reformulation computed with KZ-reduction is proven at the rootnode by branching on the last variable.* □

Theorem 2. *Let p, and r be integral vectors satisfying Assumption 3, k and M integers with*

$$\begin{aligned} &k \ge 0, \\ &M > \max\{kr_n/p_n - kr_1/p_1 - r_1/p_1 + 1,\ 2\sqrt{n}\|r\|^2\|p\|^2\}. \end{aligned} \tag{1.40}$$

Then there exists a β integer such that

$$k(M + r_n/p_n) < \beta < (k+1)(M + r_1/p_1), \tag{1.41}$$

and for all such β the problem (KP-EQ) with $a = pM + r$ has the following properties:

(1) *Its infeasibility is proven by $px \le k \vee px \ge k+1$.*
(2) *Ordinary B&B needs at least*

$$\binom{\lfloor k/\|p\|_\infty \rfloor + n - 1}{n-1}$$

nodes to prove its infeasibility, independently of the sequence in which the branching variables are chosen.
(3) *The infeasibility of the AHL reformulation computed with KZ-reduction is proven at the rootnode by branching on the last variable.* □

2. Why easiness for constraint branching implies hardness for ordinary branch-and-bound

In this section we prove a somewhat surprising result on instances of (KP). If the infeasibility is proven by branching on px, where p is a positive integral vector, then this implies a lower bound on the number of nodes that ordinary B&B must take to prove infeasibility. So in a sense easiness implies hardness!

A node of the branch-and-bound tree is identified by the subset of the variables that are fixed there, and by the values that they are fixed to. We call $(\bar{x}, F)$ a *node-fixing*, if $F \subseteq N$, and $\bar{x} \in \mathbb{Z}^F$ with $0 \le \bar{x}_i \le u_i\ \forall i \in F$, i.e. $\bar{x}$ is a collection of integers corresponding to the components of F.

Theorem 3. *Let $p \in \mathbb{Z}^n_{++}$, and k an integer such that the infeasibility of (KP) is proven by $px \le k \vee px \ge k+1$. Recall the notation of $\ell(p,k)$ from (1.26).*

(1) *If $u = e$, then ordinary B&B needs at least $2^{\ell(p,k)}$ nodes to prove the infeasibility of (KP), independently of the sequence in which the branching variables are chosen.*
(2) *If $u_i = +\infty\ \forall i$, then ordinary B&B needs at least*

$$\binom{\lfloor k/\|p\|_\infty \rfloor + n - 1}{n-1}$$

nodes to prove the infeasibility of (KP), independently of the sequence in which the branching variables are chosen. □

To have a large lower bound on the number of B&B nodes that are necessary to prove infeasibility, it is sufficient for $\ell(p,k)$ to be large, which is true, if the components of p are relatively small compared to k, and are not too different. That is, we do not need the components of the *constraint vector* a to be small, and not too different, as in Jeroslow's problem.

First we need a lemma, for which one needs to recall the definition (1.37).

Lemma 1. *Let k be an integer with $0 \le k < pu$. Then (1) and (2) below are equivalent:*

(1) *The infeasibility of* (KP) *is proven by* $px \le k \vee px \ge k+1$.
(2)

$$\max(a, p, k, u) < \beta_1 \le \beta_2 < \min(a, p, k+1, u). \tag{2.1}$$

Furthermore, if (1) *holds, then ordinary B&B cannot prune any node with node-fixing* $(\bar{x}, F)$ *that satisfies*

$$\sum_{i \in F} p_i \bar{x}_i \le k, \quad \text{and} \quad \sum_{i \notin F} p_i u_i \ge k+1. \tag{2.2}$$

Proof. Recall that we assume $0 < \beta_1 \le \beta_2 < au$. For brevity we will denote the box with upper bound u by

$$B_u = \{x \mid 0 \le x \le u\}. \tag{2.3}$$

The implication (2) $\Rightarrow$ (1) is trivial. To see (1) $\Rightarrow$ (2) first assume to the contrary that the lower inequality in (2.1) is violated, i.e. there is y_1 with

$$y_1 \in B_u, \quad py_1 \le k, \qquad ay_1 \ge \beta_1. \tag{2.4}$$

Let $x_1 = 0$. Then clearly

$$x_1 \in B_u, \quad px_1 \le k, \qquad ax_1 < \beta_1. \tag{2.5}$$

So a convex combination of x_1 and y_1, say z satisfies

$$z \in B_u, \quad pz \le k, \qquad az = \beta_1, \tag{2.6}$$

a contradiction. Next, assume to the contrary that the upper inequality in (2.1) is violated, i.e. there is y_2 with

$$y_2 \in B_u, \quad py_2 \ge k+1, \qquad ay_2 \le \beta_2. \tag{2.7}$$

Define x_2 by setting its ith component to u_i, if $u_i < +\infty$, and to some large number α to be specified later, if $u_i = +\infty$. If α is large enough, then

$$x_2 \in B_u, \quad px_2 \ge k+1, \qquad ax_2 > \beta_2. \tag{2.8}$$

Then a convex combination of x_2 and y_2, say w satisfies

$$w \in B_u, \quad pw \ge k+1, \qquad aw = \beta_2, \tag{2.9}$$

a contradiction. So (1) $\Rightarrow$ (2) is proven.

Let $(\bar{x}, F)$ be a node-fixing that satisfies (2.2). Define x' and x'' as

$$x'_i = \begin{cases} \bar{x}_i & \text{if } i \in F \\ 0 & \text{if } i \notin F \end{cases}, \qquad x''_i = \begin{cases} \bar{x}_i & \text{if } i \in F \\ u_i & \text{if } i \notin F. \end{cases} \tag{2.10}$$

If $u_i = +\infty$, then $x''_i = u_i$ means "set x''_i to an α sufficiently large number". We have $px' \le k$, so $ax' < \beta_1$; also, $px'' \ge k+1$, so $ax'' > \beta_2$ holds as well. Hence a convex combination of x' and x'', say z is LP-feasible for (KP). Also, $z_i = \bar{x}_i$ $(i \in F)$ must hold, so the node with node-fixing $(\bar{x}, F)$ is LP-feasible. $\square$

Proof of Theorem 3. Again, we use the notation B_u as in (2.3).

First we show that $0 \le k < pu$ must hold. (The upper bound of course holds trivially, if any u_i is $+\infty$.) If $k < 0$, then $px \ge k+1$ is true for all $x \in B_u$, so the infeasibility of (KP) could not be proven by $px \le k \vee px \ge k+1$. Similarly, if $k \ge pu$, then $px \le k$ is true for all $x \in B_u$, so the infeasibility of (KP) could not be proven by $px \le k \vee px \ge k+1$.

For both parts, assume w.l.o.g. that we branch on variables $x_1, x_2, \ldots$ in this sequence. For part (1), let $F = \{1, 2, \ldots, \ell(p,k)\}$. From the definition of $\ell(p,k)$ it follows that any fixing of the variables in F will satisfy (2.2), so the corresponding node will be LP-feasible. Since there are $2^{\ell(p,k)}$ such nodes, the claim follows.

For part (2), let $F = \{1, \ldots, n-1\}$, and assume that x_i is fixed to $\bar{x}_i$ for all $i \in F$. Since all u_is are $+\infty$, this node-fixing will satisfy (2.2) if

$$\bar{x}_i \ge 0 \quad \forall i \in F, \quad \sum_{i \in F} p_i \bar{x}_i \le k. \tag{2.11}$$

We will now give a lower bound on the number of $\bar{x} \in \mathbb{Z}^F$ that satisfy (2.11). Clearly, (2.11) holds, if

$$\sum_{i \in F} \bar{x}_i \le \lfloor k / \|p\|_\infty \rfloor \tag{2.12}$$

RECIPE 1

`Input:` Vectors $p, u \in \mathbb{Z}^n_{++}$, $r \in \mathbb{Z}^n$, k integer with $0 \le k < pu$.
`Output:` $M \in \mathbb{Z}_{++}, a \in \mathbb{Z}^n_{++}$, β_1, β_2 s.t. $a = pM + r$,
and the infeasibility of (KP) is proven by $px \le k \vee px \ge k+1$.

Choose M, β_1, β_2 s.t. $pM + r > 0$, and

$$\max(r, p, k, u) + kM < \beta_1 \le \beta_2 < \min(r, p, k+1, u) + (k+1)M. \tag{3.1}$$

Set $a = pM + r$.

Fig. 2. Recipe 1 to generate DKPs.

RECIPE 2

`Input:` Vectors $p, r \in \mathbb{Z}^n$ satisfying Assumption 3, k nonnegative integer.
`Output:` $M, \beta \in \mathbb{Z}_{++}, a \in \mathbb{Z}^n_{++}$ s.t. $a = pM + r$, and
the infeasibility of (KP-EQ) is proven by $px \le k \vee px \ge k+1$.

Choose $M, \beta \in \mathbb{Z}_{++}$ s.t. $pM + r > 0$, and

$$0 \le k(M + r_n/p_n) < \beta < (k+1)(M + r_1/p_1). \tag{3.2}$$

Set $a = pM + r$.

Fig. 3. Recipe 2 to generate instances of (KP-EQ).

does. It is known (see e.g. [32], page 30) that the number of nonnegative integral $(m_1, \dots, m_d)$ with $m_1 + \cdots + m_d \le t$ is

$$\binom{t+d}{d}. \tag{2.13}$$

Using this with $t = \lfloor k/\|p\|_\infty \rfloor$, $d = n-1$, the number of $\bar{x} \in \mathbb{Z}^F$ that satisfy (2.11) is at least

$$\binom{\lfloor k/\|p\|_\infty \rfloor + n - 1}{n-1},$$

and so the number of LP feasible nodes is lower bounded by the same quantity. $\square$

3. Recipes for decomposable knapsacks

In this section we give simple recipes to find instances of (KP) and (KP-EQ) with a decomposable structure. The input of the recipes is the p and r vectors, an integer k, and the output is an integer M, a vector a with $a = pM + r$, and the bounds β_1 and β_2, or β. The found instances will have their infeasibility proven by $px \le k \vee px \ge k+1$, and if k is suitably chosen, be difficult for ordinary B&B by Theorem 3. We will show that several well-known hard integer programming instances are found by our recipes.

The recipes are given in Fig. 2 and in Fig. 3, respectively.

Theorem 4. *Recipes 1 and 2 are correct.*

Proof. Since $a = pM + r$,

$$\max(a, p, k, u) \le \max(r, p, k, u) + kM, \tag{3.3}$$

and

$$\min(a, p, k+1, u) \ge \min(r, p, k+1, u) + kM. \tag{3.4}$$

So the output of Recipe 1 satisfies

$$\max(a, p, k, u) < \beta_1 \le \beta_2 < \min(a, p, k+1, u), \tag{3.5}$$

and so the infeasibility of the resulting DKP is proven by $px \le k \vee px \ge k+1$.

For Recipe 2, note that with the components of u all equal to $+\infty$, we have

$$\max(r, p, k, u) = kr_n/p_n, \quad \text{and} \tag{3.6}$$

$$\min(r, p, k+1, u) = (k+1)r_1/p_1, \tag{3.7}$$

so Recipe 2 is just a special case of Recipe 1. $\square$

Example 1 continued. We created Example 1 using Recipe 1: here $pu = 12$, so $k = 5$ has $0 \le k < pu$, and

$$\max(r, p, k, u) = \max\{x_1 - x_2 \mid 0 \le x_1, x_2 \le 6,\ x_1 + x_2 \le 5\} = 5,$$
$$\min(r, p, k+1, u) = \min\{x_1 - x_2 \mid 0 \le x_1, x_2 \le 6,\ x_1 + x_2 \ge 6\} = -6.$$

So (3.1) becomes

$$5 + 5M < \beta_1 \le \beta_2 < -6 + 6M,$$

hence $M = 20,\ \beta_1 = 106,\ \beta_2 = 113$ is a possible output of Recipe 1.

Example 2 continued. Example 2 can also be constructed via Recipe 1: now $pu = n$, so $k = (n-1)/2$ satisfies $0 \le k < pu$. Then $r = 0$ implies

$$\max(r, p, k, u) = \min(r, p, k+1, u) = 0,$$

so (3.1) becomes

$$\frac{n-1}{2}M < \beta_1 \le \beta_2 < \frac{n+1}{2}M,$$

and $M = 2,\ \beta_1 = \beta_2 = n$ is a possible output of Recipe 1.

Example 4. Let n be an odd integer, $k = \lfloor n/2 \rfloor,\ p = u = e$, and r an integral vector with

$$r_1 \le r_2 \le \cdots \le r_n. \tag{3.8}$$

Then we claim that any M and $\beta = \beta_1 = \beta_2$ is a possible output of Recipe 1, if

$$\begin{aligned} &\beta = \left\lfloor (1/2)\sum_{i=1}^{n}(M + r_i) \right\rfloor, \\ &M + r_1 \ge 0, \\ &M + r_{k+1} > (r_{k+2} + \cdots + r_n) - (r_1 + \cdots + r_k). \end{aligned} \tag{3.9}$$

Indeed, this easily follows from

$$\max(r, p, k, u) = r_{k+2} + \cdots + r_n,$$
$$\min(r, p, k+1, u) = r_1 + \cdots + r_k + r_{k+1}.$$

Two interesting, previously proposed hard knapsack instances can be obtained by picking $r,\ M,$ and β that satisfy (3.9). When

$$r = (2^{\ell+1} + 1, \ldots, 2^{\ell+n} + 1), \qquad M = 2^{n+\ell+1}, \tag{3.10}$$

with $\ell = \lfloor \log 2n \rfloor$, we obtain a feasibility version of a hard knapsack instance proposed by Todd in [21]. When

$$r = (1, \ldots, n), \qquad M = n(n+1), \tag{3.11}$$

we obtain a feasibility version of a hard knapsack instance proposed by Avis in [21].

So the instances are

$$ax = \left\lfloor \frac{1}{2}\sum_{i=1}^{n} a_i \right\rfloor, \quad x \in \{0,1\}^n,$$

with

$$a = (2^{n+\ell+1} + 2^{\ell+1} + 1, \ldots, 2^{n+\ell+1} + 2^{\ell+n} + 1), \tag{3.12}$$

for the Todd-problem, and

$$a = (n(n+1) + 1, \ldots, n(n+1) + n) \tag{3.13}$$

for the Avis-problem.

Example 5. In this example we reverse the role of p and r from Example 4, and will call the resulting DKP instance a *reverse-Avis instance*. This example illustrates how we can generate provably infeasible and provably hard instances from any p and r; also, the reverse-Avis instance will be harder from a practical viewpoint, as explained in Remark 5 below.

Let n be a positive integer divisible by 4,

$$\begin{aligned} p &= (1, \ldots, n),\\ r &= e,\\ k &= n(n+1)/4. \end{aligned} \tag{3.14}$$

Since $k = (\sum_{i=1}^{n} p_i)/2$, the first $3n/4$ components of p sum to strictly more than k, and the last $n/4$ sum to strictly less than k, so

$$\begin{aligned} \max(r, p, k, u) &< 3n/4,\\ \min(r, p, k+1, u) &> n/4. \end{aligned} \tag{3.15}$$

Hence a straightforward computation shows that

$$\begin{aligned} M &= n/2 + 2,\\ \beta &= \beta_1 = \beta_2 = 3n/4 + k(n/2+2) + 1 \end{aligned} \tag{3.16}$$

are a possible output of Recipe 1.

Corollary 4. *Ordinary B&B needs at least $2^{(n-1)/2}$ nodes to prove the infeasibility of Jeroslow's problem in Example 2, of the instances in Example 4 including the Avis- and Todd-problems, and at least $2^{n/4}$ nodes to prove the infeasibility of the reverse-Avis instance.*

Proof. We use Part (1) of Theorem 3. In the first three instances n is odd, $p = u = e,\ k = (n-1)/2,$ so $\ell(p, k) = k$. In the reverse-Avis instance we have $\ell(p, k) \geq n/4$ as explained after the definition (1.26). □

Remark 5. While we can prove a $2^{(n-1)/2}$ lower bound for the Avis- and Todd instances, they are easy from a practical viewpoint: it is straightforward to see that a single knapsack cover cut proves their infeasibility.

For the reverse-Avis problem we can prove only a $2^{n/4}$ lower bound, but this problem is hard even from a practical viewpoint. We chose $n = 60$, and ran the resulting instance using the CPLEX 11 MIP solver. After enumerating 10 million nodes the solver could not verify the infeasibility.

Next we give examples on the use of Recipe 2.

Example 6. Let $n = 2$,

$$\begin{aligned} k &= 1,\\ p &= (1, 1),\\ r &= (-11, 5). \end{aligned}$$

Then (3.2) in Recipe 2 becomes

$$0 \leq M + 5 < \beta < 2(M - 11), \tag{3.17}$$

hence $M = 29$, $\beta = 35$ is a possible output of Recipe 2. So the infeasibility of

$$\begin{aligned} &18x_1 + 34x_2 = 35\\ &x_1, x_2 \in \mathbb{Z}_+ \end{aligned} \tag{3.18}$$

is proven by $x_1 + x_2 \leq 1 \vee x_1 + x_2 \geq 2$, a fact that is easy to check directly. □

Example 7. In Recipe 2 M and β are constrained only by r_1/p_1 and r_n/p_n. So, if $n = 17,\ k = 1$, and

$$\begin{aligned} p &= (1, 1, \ldots, 1),\\ r &= (-11, -10, \ldots, 0, 1, \ldots 5), \end{aligned}$$

then $M = 29$, $\beta = 35$ is still a possible output of Recipe 2. So the infeasibility of

$$\begin{aligned} &18x_1 + 19x_2 + \cdots + 34x_2 = 35\\ &x_1, x_2, \ldots, x_{17} \geq 0\\ &x_1, x_2, \ldots, x_{17} \in \mathbb{Z}_+ \end{aligned} \tag{3.19}$$

is proven by $\sum_{i=1}^{17} x_i \leq 1 \vee \sum_{i=1}^{17} x_i \geq 2$. □

We finally give an example, in which the problem data has polynomial size in n, the infeasibility is proven by a split disjunction, but the number of nodes that ordinary B&B must enumerate to do the same is a *superexponential* function of n.

Example 8. Let n and t be integers, $n, t \geq 2$. We claim that the infeasibility of

$$\begin{aligned} & (n^{t+1}+1)x_1 + \cdots + (n^{t+1}+n)x_n = n^{2t+1} + n^{t+1} + 1 \\ & x_i \in \mathbb{Z}_+ \quad (i = 1, \ldots, n) \end{aligned} \tag{3.20}$$

is proven by

$$\sum_{i=1}^{n} x_i \leq n^t \vee \sum_{i=1}^{n} x_i \geq n^t + 1, \tag{3.21}$$

but ordinary B&B needs at least

$$n^{(n-1)(t-1)}$$

nodes to prove the same. Indeed,

$$\begin{aligned} p &= e, \\ r &= (1, 2, \ldots, n), \\ k &= n^t, \\ M &= n^{t+1}, \\ \beta &= n^{2t+1} + n^{t+1} + 1 \end{aligned} \tag{3.22}$$

satisfy (3.2). So the fact that the infeasibility is proven by (3.21) follows from the correctness of Recipe 2. By Part (2) of Theorem 3 ordinary B&B needs to enumerate at least

$$\binom{n^t + n - 1}{n - 1}$$

nodes to prove the infeasibility of (3.20). But

$$\binom{n^t + n - 1}{n - 1} \geq \binom{n^t}{n - 1} \geq \left(\frac{n^t}{n - 1}\right)^{n-1} \geq n^{(n-1)(t-1)}.$$

4. Large right-hand sides in (KP-EQ). The p-branching Frobenius number

In this section we assume that p and r integral vectors which satisfy Assumption 3 are given, and let

$$q = (r_1/p_1, \ldots, r_n/p_n). \tag{4.1}$$

Recipe 2 returns a vector $a = pM + r$, and an integral β, such that the infeasibility of (KP-EQ) with this β is proven by branching on px.

The Frobenius number of a is defined as the largest integer β for which (KP-EQ) is infeasible, and it is denoted by $\mathrm{Frob}(a)$. This section extends the lower bound result (1.8) of Aardal and Lenstra in [18,19] in two directions. First, using Recipe 2, we show that for sufficiently large M there is a range of β integers for which the infeasibility of (KP-EQ) with $a = pM + r$ is proven by branching on px. The smallest such integer is essentially the same as the lower bound in (1.8).

We will denote

$$f(M, \delta) = \left\lceil \frac{M + q_1 - \delta}{q_n - q_1} \right\rceil - 1 \tag{4.2}$$

(for simplicity, the dependence on p, and r is not shown in this definition).

Theorem 5. *Suppose that $f(M, 1) \geq 0$, (i.e. $M \geq q_n - 2q_1 + 1$), $a \in \mathbb{Z}^n_{++}$, with $a = pM + r$. Then there is an integer β with*

$$f(M, 1)(M + q_n) < \beta < (f(M, 1) + 1)(M + q_1), \tag{4.3}$$

and for all such β integers the infeasibility of (KP-EQ) is proven by $px \leq f(M, 1) \vee px \geq f(M, 1) + 1$.

Proof. There is an integer β satisfying (3.2) in Recipe 2, if

$$k(M + q_n) + 1 < (k + 1)(M + q_1). \tag{4.4}$$

But it is straightforward to see that (4.4) is equivalent to $k \leq f(M, 1)$. Choosing $k = f(M, 1)$ turns (3.2) into (4.3). □

Clearly, for all β right-hand sides found by Recipe 2

$$\beta \le \mathrm{Frob}(a). \tag{4.5}$$

Since for the β rhs values found by Recipe 2, the infeasibility of (KP-EQ) has a short, split disjunction certificate, and there is no known "easy" method to prove the infeasibility of (KP-EQ) with β equal to $\mathrm{Frob}(a)$, such β right-hand sides are interesting to study.

Definition 6. Assume that $f(M, 1) \ge 0$, and a is a positive integral vector of the form $a = pM + r$. The *p-branching Frobenius number* of a is the largest right-hand side for which the infeasibility of (KP-EQ) is proven by branching on px. It is denoted by

$$\mathrm{Frob}_p(a).$$

Theorem 6. *Assume that $f(M, 1) \ge 0$, and a is a positive integral vector of the form $a = pM + r$. Then*

$$f(M, 1)(M + q_n) < \mathrm{Frob}_p(a) < (f(M, 0) + 1)\,(M + q_1)\,. \tag{4.6}$$

Proof. The lower bound comes from Theorem 5. Recall the notation (1.37). If all components of u are $+\infty$, then

$$\max(a, p, k, u) = ka_n/p_n = k(M + q_n), \tag{4.7}$$
$$\min(a, p, k + 1, u) = (k + 1)a_1/p_1 = (k + 1)(M + q_1). \tag{4.8}$$

So Lemma 1 implies that if the infeasibility of (KP-EQ) is proven by $px \le k \vee px \ge k + 1$, then

$$k(M + q_n) < \beta < (k + 1)\,(M + q_1)\,, \tag{4.9}$$

hence

$$k(M + q_n) < (k + 1)\,(M + q_1)\,, \tag{4.10}$$

which is equivalent to

$$k < \frac{M + q_1}{q_n - q_1} \Leftrightarrow k < \left\lceil \frac{M + q_1}{q_n - q_1} \right\rceil \Leftrightarrow k \le f(M, 0). \tag{4.11}$$

The infeasibility of (KP-EQ) is proven by branching on px iff it is proven by $px \le k \vee px \ge k + 1$ for some nonnegative integer k. So, the largest such β is strictly less than

$$(k + 1)\,(M + q_1)\,, \tag{4.12}$$

with $k \le f(M, 0)$, so it is strictly less than $(f(M, 0) + 1)\,(M + q_1)$, as required. □

Example 6 continued. Recall that in this example

$$p = (1, 1),$$
$$r = (-11, 5),$$

so we have $q_1 = -11,\ q_2 = 5$. So if $M = 29$, then $f(M, 0) = f(M, 1) = 1$, and the bounds in Theorem 5 become

$$34 < \beta < 36.$$

Hence Theorem 5 finds only $\beta = 35$, as the only integer for which the infeasibility of (3.18) is proven by branching on $x_1 + x_2 \le 1 \vee x_1 + x_2 \ge 2$.

Letting $a = pM + r = (18, 34)$, Theorem 6 shows

$$34 < \mathrm{Frob}_p(a) < 36$$

so $\mathrm{Frob}_p(a) = 35$.

5. The geometry of the original set, and the reformulation

This section proves some basic results on the geometry of the reformulations using ideas from the recent article of Mehrotra and Li [12]. Our goal is to relate the width of a polyhedron to the width of its reformulation in a given direction.

Theorem 7. *Let*

$$Q = \{x \in \mathbb{R}^n \mid Ax \le b\},$$
$$\tilde{Q} = \{y \in \mathbb{R}^n \mid AUy \le b\},$$

where U is a unimodular matrix, and $c \in \mathbb{Z}^n$.

Then

(1)

$$\max\{cx \mid x \in Q\} = \max\{cUy \mid y \in \tilde{Q}\},$$

with x^* attaining the maximum in Q if and only if $U^{-1}x^*$ attains it in $\tilde{Q}$.

(2)

$$\mathrm{width}(c, Q) = \mathrm{width}(cU, \tilde{Q}).$$

(3)

$$\mathrm{iwidth}(c, Q) = \mathrm{iwidth}(cU, \tilde{Q}).$$

Proof. Statement (1) follows from

$$Q = \{Uy \mid y \in \tilde{Q}\}, \tag{5.1}$$

and an analogous result holds for "min". Statements (2) and (3) are easy consequences. □

Theorem 7 immediately implies

Corollary 7.

$$\min_{c \in \mathbb{Z}^n \setminus \{0\}} \mathrm{width}(c, Q) = \min_{d \in \mathbb{Z}^n \setminus \{0\}} \mathrm{width}(d, \tilde{Q}). \quad \square$$

Theorem 8. *Suppose that the integral matrix A has n columns, and m linearly independent rows, let S be a polyhedron, and*

$$Q = \{x \in \mathbb{R}^n \mid x \in S,\ Ax = b\},$$
$$\hat{Q} = \{\lambda \mid V\lambda + x_b \in S,\ \lambda \in \mathbb{R}^{n-m}\},$$

where V is a basis matrix for $\mathbb{N}(A)$, and x_b satisfies $Ax_b = b$. If $c \in \mathbb{Z}^n$ is a row vector, then

(1)

$$\max\{cx \mid x \in Q\} = cx_b + \max\{cV\lambda \mid \lambda \in \hat{Q}\},$$

with x^* attaining the maximum in Q if and only if λ^* attains it in $\hat{Q}$, where $x^* = V\lambda^* + x_b$.

(2)

$$\mathrm{width}(c, Q) = \mathrm{width}(cV, \hat{Q}).$$

(3)

$$\mathrm{iwidth}(c, Q) = \mathrm{iwidth}(cV, \hat{Q}).$$

Proof. Statement (1) follows from

$$Q = \{V\lambda + x_b \mid \lambda \in \hat{Q}\}.$$

An analogous result holds for "min", and statements (2) and (3) are then straightforward consequences. □

Theorem 8 can be "reversed". That is, given a row vector $d \in \mathbb{Z}^{n-m}$, we can find a row vector $c \in \mathbb{Z}^n$, such that

$$\max\{cx \mid x \in Q\} = \max\{d\lambda \mid \lambda \in \hat{Q}\} + \text{const}.$$

Looking at (1) in Theorem 8, for the given d it suffices to solve

$$cV = d, \quad c \in \mathbb{Z}^n. \tag{5.2}$$

The latter task is trivial, if we have a V^* integral matrix such that

$$V^*V = I_{n-m}; \tag{5.3}$$

then $c = dV^*$ will solve (5.2). To find V^*, let W be an integral matrix such that $U = [W, V]$ is unimodular; for instance W will do, if

$$A[W, V] = [H, 0],$$

where H is the Hermite Normal Form of A. Then we can choose V^* as the submatrix of U^{-1} consisting of the last $n - m$ rows.

In this way we have proved Theorem 9 and Corollary 8, which are essentially the same as Theorem 4.1, and Corollary 4.1 proven by Mehrotra and Li in [12]:

Theorem 9 (*Mehrotra and Li*). *Let Q, $\hat{Q}$, V be as in Theorem 8, and V^* a matrix satisfying (5.3). Then*

(1)

$$\max\{d\lambda \mid \lambda \in \hat{Q}\} = \max\{dV^*x \mid x \in Q\} - dV^*x_b,$$

with x^* attaining the maximum in Q if and only if λ^* attains it in $\hat{Q}$, where $x^* = V\lambda^* + x_b$.

(2)

$$\text{width}(dV^*, Q) = \text{width}(d, \hat{Q}).$$

(3)

$$\text{iwidth}(dV^*, Q) = \text{iwidth}(d, \hat{Q}). \quad \square$$

Corollary 8 (*Mehrotra and Li*). *Let Q, $\hat{Q}$, V, and V^* be as before. Then*

$$\min_{d \in \mathbb{Z}^{n-m} \setminus \{0\}} \text{width}(d, \hat{Q}) = \min_{c \in \mathbb{L}(V^{*T}) \setminus \{0\}} \text{width}(c, Q). \quad \square$$

6. Why the reformulations make DKPs easy

This section will assume a decomposable structure on (KP) and (KP-EQ), that is

$$a = pM + r, \tag{6.1}$$

with $p \in \mathbb{Z}^n_{++}$, $r \in \mathbb{Z}^n$, and M an integer. We show that for large enough M the phenomenon of Examples 1 and 2 *must* happen, i.e. the originally difficult DKPs will turn into easy ones.

We recall that for a given matrix A, we use a Matlab-like notation, and denote its jth row, and column by $A_{j,:}$ and $A_{:,j}$, respectively.

An outline of the results is:

(1) If M is large enough, and U is the transformation matrix of the rangespace reformulation, then pU will have a “small” number of nonzeros. Considering the equivalence between the old and new variables $Uy = x$, this means that branching on just a few variables in the reformulation will “simulate” branching on the backbone constraint px in the original problem. An analogous result will hold for the AHL reformulation.

(2) It is interesting to look at what happens, when branching on px does not prove infeasibility in the original problem, but the width in the direction of p is relatively small—this is the case in (KP-EQ) as we prove in Lemma 2.
Invoking the results in Section 5 will prove that when M is sufficiently large, the same, or smaller width is achieved along a unit direction in either one of the reformulations.

Lemma 2. *Suppose that Assumption 3 holds. Then*

$$\text{width}(p, \text{(KP-EQ)}) = \Theta(\beta/M^2), \tag{6.2}$$

$$\text{width}(e_i, \text{(KP-EQ)}) = \Theta(\beta/M) \quad \forall i \in \{1, \dots, n\}. \tag{6.3}$$

In both equations the constant depends on p and r.

Proof. Since $a_i = p_iM + r_i$,

$$r_1/p_1 \le \cdots \le r_n/p_n \tag{6.4}$$

implies

$$p_1/a_1 \ge \cdots \ge p_n/a_n. \tag{6.5}$$

So

$$\max\{px \mid ax = \beta,\ x \ge 0\} = \beta p_1/a_1,$$
$$\min\{px \mid ax = \beta,\ x \ge 0\} = \beta p_n/a_n,$$

and therefore

$$\begin{aligned}\text{width}(p, \text{(KP-EQ)}) &= \beta(p_1/a_1 - p_n/a_n)\\ &= \beta(p_1 a_n - p_n a_1)/(a_1 a_n)\\ &= \beta(p_1 r_n - p_n r_1)/(a_1 a_n).\end{aligned}$$

Also,

$$\begin{aligned}&\max\{x_i \mid ax = \beta,\ x \geq 0\} = \beta/a_i,\\ &\min\{x_i \mid ax = \beta,\ x \geq 0\} = 0,\end{aligned}$$

hence

$$\text{width}(e_i, \text{(KP-EQ)}) = \beta/a_i.$$

Since

$$a_i = \Theta(M) \quad \forall i \in \{1, \ldots, n\}$$

both (6.2) and (6.3) follow. □

6.1. Analysis of the rangespace reformulation

After the rangespace reformulation is applied, the problem (KP) becomes

$$\begin{aligned}&\beta_1 \leq (aU)y \leq \beta_2\\ &0 \leq Uy \leq u\\ &y \in \mathbb{Z}^n,\end{aligned} \tag{KP-R}$$

where the matrix U was computed by a BR algorithm with input

$$A = \begin{pmatrix} a \\ I \end{pmatrix} = \begin{pmatrix} pM + r \\ I \end{pmatrix}. \tag{6.6}$$

Let us write

$$\tilde{A} = AU, \qquad \tilde{a} = aU, \qquad \tilde{p} = pU, \qquad \tilde{r} = rU,$$

and fix c_n, the reduction factor of the used BR algorithm.

Recall that for a lattice L, $\Lambda_k(L)$ is the smallest real number t for which there are k linearly independent vectors in L with norm at most t.

For brevity, we will denote

$$\alpha_k = \Lambda_k(\mathbb{N}(p)) \quad (k = 1, \ldots, n-1). \tag{6.7}$$

First we need a technical lemma:

Lemma 3. *Let A be as in (6.6). Then*

$$\Lambda_k(\mathbb{L}(A)) \leq (\|r\| + 1)\alpha_k \quad \textit{for } k \in \{1, \ldots, n-1\}. \tag{6.8}$$

Proof. We need to show that there are k linearly independent vectors in $\mathbb{L}(A)$ with norm bounded by $(\|r\| + 1)\alpha_k$.

Suppose that $w_1, \ldots, w_k$ are linearly independent vectors in $\mathbb{N}(p)$ with norm bounded by α_k. Then $Aw_1, \ldots, Aw_k$ are linearly independent in $\mathbb{L}(A)$, and

$$Aw_i = \begin{pmatrix} a \\ I \end{pmatrix} w_i = \begin{pmatrix} pM + r \\ I \end{pmatrix} w_i = \begin{pmatrix} rw_i \\ w_i \end{pmatrix} \quad \forall i,$$

hence

$$\|Aw_i\| \leq (\|r\| + 1)\|w_i\| \leq (\|r\| + 1)\alpha_k \quad (i = 1, \ldots, k)$$

follows, which proves (6.8). □

Theorem 10. *The following hold:*

(1) *Let* $k \le n-1$, *and suppose*

$$M > c_n(\|r\|+1)^2\alpha_k. \tag{6.9}$$

Then

$$\tilde{p}_{1:k} = 0. \tag{6.10}$$

Also, if the infeasibility of (KP) *is proven by branching on* px, *then the infeasibility of* (KP-R) *is proven by branching on* $y_{k+1}, \ldots, y_n$.

(2) *Suppose*

$$M > c_n(\|r\|+1)^2\|p\|. \tag{6.11}$$

Then

$$\tilde{p}_{1:n-1} = 0, \tag{6.12}$$

and

$$\begin{aligned} &\mathrm{width}(e_n, \text{(KP-R)}) \le \mathrm{width}(p, \text{(KP)}) \\ &\mathrm{iwidth}(e_n, \text{(KP-R)}) \le \mathrm{iwidth}(p, \text{(KP)}) \end{aligned} \tag{6.13}$$

In particular, in the rangespace reformulation of (KP-EQ) *the width, and the integer width in the direction of* e_n *are*

$$\Theta(\beta/M^2). \quad \square$$

Before proving Theorem 10, we give some intuition to the validity of (6.10) and (6.12). Suppose M is "large", compared to $\|p\|$, and $\|r\|$. In view of how the matrix A looks in (6.6), it is clear that its columns are *not* short, and near orthogonal, due to the presence of the nonzero p_i components. Thus to make its columns short and nearly orthogonal, the best thing to do is to apply a unimodular transformation that eliminates "many" nonzero p_is.

Proof. For brevity, denote by Q and $\tilde{Q}$ the feasible set of the LP-relaxation of (KP) and (KP-R), respectively.

Proof of (1). To show (6.10), fix $j \le k$; we will prove $\tilde{p}_j = 0$.

Since $\tilde{A}$ was computed by a BR algorithm with reduction factor c_n, Lemma 3 implies

$$\|\tilde{A}_{:,j}\| \le c_n(\|r\|+1)\alpha_k. \tag{6.14}$$

To get a contradiction, suppose $\tilde{p}_j \neq 0$. Then, since $\tilde{p}_j$ is integral,

$$\begin{aligned} \|\tilde{A}_{:,j}\| &\ge |\tilde{a}_j| \\ &= |\tilde{p}_j M + \tilde{r}_j| \\ &\ge M - |\tilde{r}_j|. \end{aligned} \tag{6.15}$$

Hence

$$\begin{aligned} M &\le \|\tilde{A}_{:,j}\| + |\tilde{r}_j| \\ &\le \|\tilde{A}_{:,j}\| + \|r\|\|U_{:,j}\|, \\ &\le \|\tilde{A}_{:,j}\| + \|r\|\|\tilde{A}_{:,j}\|, \\ &= (\|r\|+1)\|\tilde{A}_{:,j}\| \\ &\le c_n(\|r\|+1)^2\alpha_k, \end{aligned} \tag{6.16}$$

with the second inequality coming from Cauchy–Schwarz, the third from $U_{:,j}$ being a subvector of $\tilde{A}_{:,j}$, and the fourth from (6.14). Thus, we obtained a contradiction to the choice of M, which proves $\tilde{p}_j = 0$.

Suppose now that the infeasibility of (KP) is proven by branching on px. We need to show:

$$y_i \in \mathbb{Z} \quad \forall\, i \in \{k+1, \ldots, n\} \;\Rightarrow\; y \notin \tilde{Q}. \tag{6.17}$$

Let $y \in \tilde{Q}$. Then

$$Uy \in Q \;\Rightarrow\; pUy \notin \mathbb{Z} \;\Rightarrow\; \tilde{p}_{k+1}y_{k+1} + \cdots + \tilde{p}_n y_n \notin \mathbb{Z} \Rightarrow y_i \notin \mathbb{Z} \quad \text{for some } i \in \{k+1, \ldots, n\},$$

as required.

Proof of (2). The statement (6.12) follows from (6.10), and the obvious fact, that $\alpha_{n-1} \leq \|p\|$, since there are $n-1$ linearly independent vectors in $\mathbb{N}(p)$ with norm bounded by $\|p\|$.

To see (6.13), we claim

$$\begin{aligned} \operatorname{width}(e_n, \tilde{Q}) &\leq \operatorname{width}(\tilde{p}_n e_n, \tilde{Q}) \\ &= \operatorname{width}(pU, \tilde{Q}) \\ &= \operatorname{width}(p, Q). \end{aligned}$$

Indeed, the inequality follows from $\tilde{p}_n$ being a nonzero integer. The first equality comes from (6.12), and the second from (1) in Theorem 7. The inequalities hold, even if we replace "width" by "iwidth", so this proves the second inequality in (6.13).

The claim about the width in the direction of e_n follows from (6.13), and Lemma 2. □

6.2. Analysis of the AHL reformulation

The technique we use to analyse the AHL reformulation is similar, but the bound on M, which is necessary for the dominant p direction to turn into a unit direction is different. If $\beta_1 = \beta_2 = \beta$, then the AHL reformulation of (KP) is

$$\begin{aligned} 0 &\leq V\lambda + x_\beta \leq u \\ \lambda &\in \mathbb{Z}^{n-1}, \end{aligned} \tag{KP-N}$$

where the matrix V is a basis of $\mathbb{N}(a)$ computed by a BR algorithm, and $ax_\beta = \beta$.

Let us write $\hat{p} = pV$, $\hat{r} = rV$ and recall the notation for α_k from (6.7). Again we need a lemma.

Lemma 4. *Let $k \in \{1, \ldots, n-2\}$. Then*

$$\Lambda_k(\mathbb{N}(p) \cap \mathbb{N}(r)) \leq 2\|r\|\alpha_{k+1}^2. \tag{6.18}$$

Proof. We need to show that there are k linearly independent vectors in $\mathbb{N}(p) \cap \mathbb{N}(r)$ with norm bounded by $2\|r\|\alpha_{k+1}^2$.

Suppose that $w_1, \ldots, w_{k+1}$ are linearly independent vectors in $\mathbb{N}(p)$ with norm bounded by α_{k+1}. Let $W = [\, w_1, \ldots, w_{k+1}\,]$, and

$$d = rW \in \mathbb{Z}^{k+1}.$$

Suppose w.l.o.g. that for some $t \in \{1, \ldots, k+1\}$

$$d_1 \neq 0, \ldots, d_t \neq 0, \qquad d_{t+1} = \cdots = d_{k+1} = 0.$$

Then

$$d_2 w_1 - d_1 w_2,\ d_3 w_1 - d_1 w_3, \ldots,\ d_t w_1 - d_1 w_t$$

are $t-1$ linearly independent vectors in $\mathbb{N}(p) \cap \mathbb{N}(r)$ with norm bounded by

$$\begin{aligned} 2\|d\|_\infty \alpha_{k+1} &= 2\left(\max_{i=1,\ldots,t} |rw_i|\right)\alpha_{k+1} \\ &\leq 2\|r\|\left(\max_{i=1,\ldots,t} \|w_i\|\right)\alpha_{k+1} \\ &\leq 2\|r\|\alpha_{k+1}^2. \end{aligned}$$

The $k+1-t$ vectors

$$w_{t+1}, \ldots, w_{k+1}$$

are obviously in $\mathbb{N}(p) \cap \mathbb{N}(r)$, with their norm obeying the same bound, and the two groups together are linearly independent. □

Theorem 11. *Suppose that p and r are not parallel. Then the following hold:*

(1) *Let $k \leq n-2$, and suppose*

$$M > 2c_n\|r\|^2\alpha_{k+1}^2. \tag{6.19}$$

Then

$$\hat{p}_{1:k} = 0. \tag{6.20}$$

Also, if the infeasibility of (KP) is proven by branching on px, then the infeasibility of (KP-N) is proven by branching on $\lambda_{k+1}, \ldots, \lambda_{n-1}$.

(2) *Suppose*

$$M > 2c_n\|r\|^2\|p\|^2. \tag{6.21}$$

Then

$$\hat{p}_{1:n-2} = 0, \tag{6.22}$$

and

$$\begin{aligned} &\text{width}(e_{n-1}, \text{(KP-N)}) \leq \text{width}(p, \text{(KP)}) \\ &\text{iwidth}(e_{n-1}, \text{(KP-N)}) \leq \text{iwidth}(p, \text{(KP)}). \end{aligned} \tag{6.23}$$

In particular, in the AHL reformulation of (KP-EQ) *the width, and the integer width in the direction of* e_{n-1} *are*

$$\Theta(\beta/M^2).$$

Proof. First note that $pV \neq 0$, since $aV = 0$, $pV = 0$ implies $rV = 0$, hence p and r would be parallel. Also, for brevity, denote by Q and $\hat{Q}$ the feasible set of the LP-relaxation of (KP) and (KP-N), respectively.

Proof of (1). To show (6.20), fix $j \leq k$; we will prove $\hat{p}_j = 0$. Suppose to the contrary that $\hat{p}_j \neq 0$, then its absolute value is at least 1. Hence

$$\begin{aligned} 0 = |aV_{:,j}| &= |\hat{p}_j M + \hat{r}_j| \\ &\geq M - |\hat{r}_j|. \end{aligned} \tag{6.24}$$

Therefore

$$\begin{aligned} M &\leq |\hat{r}_j| \\ &= |rV_{:,j}| \\ &\leq \|r\|\|V_{:,j}\| \\ &\leq 2c_n\|r\|^2\alpha_{k+1}^2. \end{aligned}$$

Here the second inequality comes from Cauchy–Schwarz. The third is true, since the columns of V are a reduced basis of $\mathbb{N}(a) \subseteq \mathbb{N}(p) \cap \mathbb{N}(r)$, and by using Lemma 4.

Suppose now, that the infeasibility of (KP) is proven by branching on px. We need to show:

$$\lambda_i \in \mathbb{Z} \quad \forall i \in \{k+1, \ldots, n-1\} \Rightarrow \lambda \notin \hat{Q}. \tag{6.25}$$

Let $\lambda \in \hat{Q}$. Then

$$\begin{aligned} &V\lambda + x_\beta \in Q \Rightarrow p(V\lambda + x_\beta) \notin \mathbb{Z} \Rightarrow \hat{p}_{k+1}\lambda_{k+1} + \cdots + \hat{p}_{n-1}\lambda_{n-1} + px_\beta \notin \mathbb{Z} \Rightarrow \\ &\lambda_i \notin \mathbb{Z} \quad \text{for some } i \in \{k+1, \ldots, n-1\}, \end{aligned}$$

as required.

Proof of (2). The statement (6.22) again follows from the fact, that there are $n-1$ linearly independent vectors in $\mathbb{N}(p)$ with norm bounded by $\|p\|$.

We will now prove (6.23). Since $\hat{p}_{n-1}$ is an integer, its absolute value is at least 1. Hence

$$\begin{aligned} \text{width}(e_{n-1}, \hat{Q}) &\leq \text{width}(\hat{p}_{n-1}e_{n-1}, \hat{Q}) \\ &= \text{width}(pV, \hat{Q}) \\ &= \text{width}(p, Q), \end{aligned}$$

with first equality true because of (6.22), and the second one due to (2) in Theorem 8. The proof of the integer width follows analogously.

The claim about the width in the direction of e_{n-1} follows from (6.23), and Lemma 2. □

6.3. *Proof of* Theorems 1 *and* 2

Proof of Theorem 1. Recipe 1 requires

$$\max(r, p, k, u) + kM < \beta_1 \leq \beta_2 < \min(r, p, k+1, u) + (k+1)M. \tag{6.26}$$

Since now $u = e$, both $\max(r, p, k, u)$ and $\min(r, p, k + 1, u)$ are bounded by $\|r\|_1 \le \sqrt{n}\|r\|$ in absolute value. So if β_1 and β_2 satisfy (1.39), then they are a possible output of Recipe 1, so the infeasibility of the resulting DKP is proven by $px \le k \vee px \ge k + 1$. If

$$M > 2\sqrt{n}\|r\| + 1, \tag{6.27}$$

then there is room in (1.39) for β_1 and β_2 to be integers. Theorem 3 implies the lower bound on the number of nodes that ordinary B&B must enumerate to prove infeasibility.

On the other hand, (2) in Theorem 10 with $c_n = \sqrt{n}$ implies that if

$$M > \sqrt{n}(\|r\| + 1)^2\|p\|, \tag{6.28}$$

then the infeasibility of the rangespace reformulation is proven by branching on the last variable.

Finally, the bound on M in (1.38) implies both (6.27) and (6.28). □

Proof of Theorem 2. From the lower bound on M, there is a β integer that satisfies (1.41), and the fact that the resulting instance's infeasibility is proven by $px \le k \vee px \ge k + 1$ follows from the correctness of Recipe 2. The lower bound on the number of nodes that ordinary B&B must enumerate to prove infeasibility follows from Theorem 3.

The fact that the infeasibility of the AHL reformulation is proven by branching on the last variable follows from (2) in Theorem 11 with $c_n = \sqrt{n}$. □

7. A computational study

The theoretical part of the paper shows that

- DKPs with suitably chosen parameters are hard for ordinary B&B, and easy for branching on px, just like Examples 1 and 2, and
- both the rangespace, and AHL reformulations make them easy. The key point is that branching on the last few variables in the reformulation simulates the effect of branching on px in the original problem.

We now look at the question whether these results translate into practice. The papers [16,14,17,18] tested the AHL reformulation on the following instances:

- In [14], equality constrained knapsacks arising from practical applications.
- In [16], the marketshare problems [15].
- In [17], an extension of the marketshare problems.
- In [18] the instances of (KP-EQ), with the rhs equal to $\mathrm{Frob}(a)$.

Our tested instances are bounded DKPs both with equality and inequality constraints, and instances of (KP-EQ).

In summary, we found that

(1) On *infeasible* problems, both reformulations are effective in reducing the solution time of proving infeasibility.

(2) They are also effective on *feasible* problems.

In feasible problems a solution may be found by accident, so it is not clear how to theoretically quantify the effect of various branching strategies, or the reformulations on such instances.

(3) They are also effective on optimization versions of DKPs.

(4) When $\beta_1 = \beta_2$, i.e. both reformulations are applicable, there is no significant difference in their performance.

The calculations are done on a Linux PC with a 3.2 GHz CPU. The MIP solver was CPLEX 9.0. For feasibility versions of integer programs, we used the sum of the variables as a dummy objective function. The basis reduction computations called the Korkhine–Zolotarev (KZ) subroutines from the Number Theory Library (NTL) version 5.4 (see [33]).

We let $n = 50$, and first generate 10 vectors $p, r \in \mathbb{Z}^n$ with the components of p uniformly distributed in $[1, 10]$ and the components of r uniformly distributed in $[-10, 10]$. We use these ten p, r pairs for all families of our instances.

Recall the notation that for $k \in \mathbb{Z},\ u \in \mathbb{Z}^n_{++}$

$$\max(r, p, k, u) = \max\{rx \mid px \le k,\ 0 \le x \le u\},$$
$$\min(r, p, k + 1, u) = \min\{rx \mid px \ge k + 1,\ 0 \le x \le u\}.$$

7.1. Bounded knapsack problems with $u = e$

We used Recipe 1 to generate 10 difficult DKPs, with bounds on the variables, as follows:
For each p, r we let

$$u = e, \qquad M = 10\,000, \qquad k = n/2 = 25, \qquad a = pM + r,$$

and set

$$\begin{aligned} \beta_1 &= \lceil \max(r, p, k, u) + kM \rceil \\ \beta_2 &= \lfloor \min(r, p, k+1, u) + (k+1)M \rfloor, \end{aligned}$$

By the choice of the data $\beta_1 \le \beta_2$ holds in all cases. We considered the following problems using these $a,\ u,\ \beta_1,\ \beta_2$:

- The basic infeasible knapsack problem:

$$\begin{aligned} \beta_1 &\le ax \le \beta_2 \\ 0 &\le x \le u \\ x &\in \mathbb{Z}^n. \end{aligned} \tag{DKP-INFEAS}$$

- The optimization version:

$$\begin{aligned} &\max ax \\ &s.t.\, ax \le \beta_2 \\ &0 \le x \le u \\ &x \in \mathbb{Z}^n. \end{aligned} \tag{DKP-OPT}$$

  We denote by β_a the optimal value, and will use β_a for creating further instances.
- The feasibility problem, with the rhs equal to β_a:

$$\begin{aligned} &ax = \beta_a \\ &0 \le x \le u \\ &x \in \mathbb{Z}^n, \end{aligned} \tag{DKP-FEAS-MAX}$$

- The feasibility problem, with the rhs set to make it infeasible:

$$\begin{aligned} ax &= \beta_a + 1 \\ 0 &\le x \le u \\ x &\in \mathbb{Z}^n. \end{aligned} \tag{DKP-INFEAS-MIN}$$

On the last two families both reformulations are applicable.

The results are in Table 1. In the columns marked ‘R’, and ‘N’ we display the number of B&B nodes taken by CPLEX after rangespace and AHL reformulation was applied, respectively. In the columns marked ‘ORIG’ we show the number of B&B nodes taken by CPLEX on the original formulation.

Since the LP subproblems of these instances are easy to solve, we feel that the number of B&B nodes is a better way of comparing the performance of the MIP solver with and without the reformulation.

We also verified that providing px as a branching direction in the original formulation makes these problems easy. We ran CPLEX on the original instances, after adding a new variable z, and the equation $z = px$ to the formulation. The results with this option are essentially the same, as the results in the ‘R’ and ‘N’ columns.

7.2. Bounded knapsack problems with $u = 10e$

We repeated the above experiment with $u = 10e$, but all other settings the same. That is, using the same ten p, r pairs, we let

$$u = 10e, \qquad M = 10\,000, \qquad k = n/2 = 25, \qquad a = pM + r,$$

and set

$$\begin{aligned} \beta_1 &= \lceil \max(r, p, k, u) + kM \rceil \\ \beta_2 &= \lfloor \min(r, p, k+1, u) + (k+1)M \rfloor \end{aligned}$$

then solved the instances (DKP-INFEAS) and (DKP-OPT), (DKP-FEAS-MAX) and (DKP-INFEAS-MIN) as before. The results are in Table 2. The original formulations turned out to be more difficult now, whereas the reformulated problems were just as easy as in the $u = e$ case.

Table 1
DKPs with $n = 50$, $k = 25$, $u = e$, $M = 10\,000$.

Ins	RHS values			(DKP-INFEAS)		(DKP-OPT)		(DKP-FEAS-MAX)			(DKP-INFEAS-MIN)		
	β_1	β_2	β_a	R	ORIG	R	ORIG	R	N	ORIG	R	N	ORIG
1	250 040	259 972	250 039	1	4 330 785	10	7 360 728	10	1	1 219 304	1	1	3 181 671
2	250 044	259 979	250 043	1	2 138 598	10	2 329 217	1	1	24 130	1	1	1 880 980
3	250 069	259 973	250 068	1	12 480 272	20	14 006 843	10	3	13 800	1	1	11 993 912[a]
4	250 034	259 961	250 033	1	1 454 260	10	2 800 898	1	5	555 144	1	1	2 531 222
5	250 037	259 975	250 036	1	4 811 440	10	6 715 586	1	10	155 670	1	1	4 131 652
6	250 038	259 981	250 037	1	3 239 982	10	2 659 752	10	10	283 776	1	1	3 155 522
7	250 085	259 948	250 084	1	11 579 118	10	14 598 901	10	1	107 170	1	1	10 871 441[a]
8	250 052	259 961	250 051	1	8 659 516	10	15 440 957	10	1	486 255	1	1	8 097 370
9	250 045	259 984	250 044	1	6 393 700	20	12 520 666	1	10	82 455	1	1	6 346 153
10	250 061	259 968	250 060	1	12 244 168	10	14 848 327	10	1	3 600	1	1	11 929 161[a]

[a] 1 h time limit exceeded.

Table 2
DKPs with $n = 50$, $k = 25$, $u = 10$, $M = 10\,000$.

Ins	RHS values			(DKP-INFEAS)		(DKP-OPT)		(DKP-FEAS-MAX)			(DKP-INFEAS-MIN)		
	β_1	β_2	β_a	R	ORIG	R	ORIG	R	N	ORIG	R	N	ORIG
1	250 083	259 719	250 082	1	13 204 411	1	12 927 001	1	1	2 571 521	1	1	11 968 829[a]
2	250 111	259 779	250 110	1	13 674 751	1	13 369 911	1	1	12 441 612[a]	1	1	11 968 829[a]
3	250 156	259 729	250 155	1	10 939 735	1	13 737 652	1	1	1 702 224	1	1	10 342 918[a]
4	250 098	259 619	250 097	1	14 678 404	1	12 762 803	1	1	25 917	1	1	13 480 436[a]
5	250 059	259 759	250 058	1	14 128 736	1	13 464 255	1	1	5 829 029	1	1	13 070 602[a]
6	250 051	259 799	250 050	1	13 979 145	10	12 310 057	1	1	597 113	1	1	13 211 779[a]
7	250 206	259 489	250 205	1	8 895 772	10	8 725 886	1	10	10 046 297[a]	1	1	13 211 779[a]
8	250 111	259 619	250 110	1	13 198 252	1	13 799 370	1	1	12 235 292[a]	1	1	13 211 779[a]
9	250 081	259 849	250 080	1	13 136 603	10	13 082 057	1	1	18 687	1	1	12 448 850[a]
10	250 206	259 689	250 205	1	9 251 523	10	12 947 576	1	1	9 692 170[a]	1	1	12 448 850[a]

[a] 1 h time limit exceeded.

7.3. Equality constrained, unbounded knapsack problems

In this section we consider instances of the type

$$\begin{aligned} ax &= \beta \\ x &\geq 0 \\ x &\in \mathbb{Z}^n. \end{aligned} \tag{KP-EQ}$$

We recall the following facts:

- If we choose M sufficiently large, and a β integer satisfying
$$0 \leq \left(\left\lceil \frac{M + q_1 - 1}{q_n - q_1} \right\rceil - 1 \right) (M + q_n) < \beta < \left\lceil \frac{M + q_1 - 1}{q_n - q_1} \right\rceil (M + q_1), \tag{7.1}$$
then the infeasibility of (KP-EQ) is proven by branching on px.
- If β^* is the largest integer satisfying (7.1), and by $\mathrm{Frob}(a)$ the Frobenius number of a, then clearly $\beta^* \leq \mathrm{Frob}(a)$.
- Finding β^* is trivial, while computing $\mathrm{Frob}(a)$ requires solving a sequence of integer programs.

We generated 20 instances as follows: using the same p, r pairs as in the previous experiments, we let

$$M = 10\,000.$$

Then the first instance with a fixed p, r pair arises by letting the rhs in (KP-EQ) to be β^*, and the second by letting it to be equal to $\mathrm{Frob}(a)$.

The (KP-EQ) instances with $\beta = \mathrm{Frob}(a)$ were already considered in [18].

Our computational results are in Table 3, where we go further than [18] is by showing that

- β^* and $\mathrm{Frob}(a)$ are not too different, and neither is the difficulty of (KP-EQ) with these two different rhs values.
- Now both reformulations can be applied, and their performance is similar.
- According to Lemma 2, $\mathrm{width}(p, \text{(KP-EQ)})$ and $\mathrm{iwidth}(p, \text{(KP-EQ)})$ both should be small compared to the width in unit directions, even when the infeasibility of (KP-EQ) is not proven by branching on px. This is indeed the case when $\beta = \mathrm{Frob}(a)$, and we list $\mathrm{iwidth}(p, \text{(KP-EQ)})$ in Table 3 as well.
- In the column "px" we list the number of B&B nodes necessary to solve the problems, when the variable z, and the equation $z = px$ is added to the original problems. The results are similar to the ones obtained with the reformulations.

Table 3
$n = 50, M = 10\,000$.

	RHS, width			$ax = \beta^*$				$ax = \text{Frob}(a)$			
#	β^*	Frob(a)	iwidth(p)	R	N	px	ORIG	R	N	px	ORIG
1	7 683 078	7 703 088	1	1	1	1	7 110 020*	5	1	13	7 320 060*
2	8 683 916	8 703 917	1	1	1	1	6 997 704*	8	3	15	7 123 300*
3	8 325 834	8 345 840	1	1	1	1	15 383 299*	8	1	19	15 313 074*
4	10 347 239	10 367 238	2	1	1	1	10 053 497*	14	24	18	9 928 134*
5	16 655 001	16 665 004	1	1	1	1	9 254 836*	33	19	57	7 519 023*
6	9 081 818	9 121 828	1	1	1	1	6 802 797*	21	1	45	7 011 946*
7	6245 624	6 245 632	1	1	1	1	7 180 978*	1	1	67	7 151 382*
8	10 514 739	10 534 740	1	1	1	1	7 164 967*	1	1	20	7 178 052*
9	14 275 715	14 285 716	1	1	1	1	7 319 379*	1	1	11	7 368 436*
10	9 838 851	9 838 851	0	1	1	1	7 520 143*	1	1	1	7 230 420*

* 1 h time limit exceeded.

7.4. Reformulated problems with basic MIP settings

To confirm the easiness of the reformulated instances we reran all of them with the most basic CPLEX settings: no cuts, no aggregator, no presolve, and node selection set to depth first search. All instances finished within a hundred nodes.

The instances and parameter files are publicly available from [34].

8. Comments on the analysis in [18]

In [18] Aardal and Lenstra studied the instances (KP-EQ) with the constraint vector a decomposing as

$$a = pM + r, \tag{8.1}$$

with $p \in \mathbb{Z}^n_{++}$, $r \in \mathbb{Z}^n$, M a positive integer, under Assumption 1. Recall that the reformulation (1.4) is constructed so that the columns of B form an LLL-reduced basis of $\mathbb{N}(a)$.

Denoting the last column of B by b_{n-1}, Theorem 4 in [18] proves (1.9), which we recall here:

$$\|b_{n-1}\| \geq \frac{\|a\|}{\sqrt{\|p\|^2\|r\|^2 - (pr^{\mathrm{T}})^2}}, \tag{8.2}$$

and the following claims are made:

- It can be assumed without loss of generality, that the columns of B are ordered in a way that the first $n-2$ form a basis for $\mathbb{N}(p) \cap \mathbb{N}(r)$. This claim is used in the proof of Theorem 4.
- Denoting by Q the feasible set of the LP-relaxation of (1.4), b_{n-1} being long implies that $\text{iwidth}(e_{n-1}, Q)$ is small.

To reconcile the notation with that of [18], we remark that in the latter L_0, and L_C is used, where

$$L_0 = \mathbb{N}(a),$$
$$L_C = \mathbb{N}(p) \cap \mathbb{N}(r).$$

Here we provide Example 9, in which p, r, M satisfy Assumption 1, B is an LLL-reduced basis of $\mathbb{N}(a)$, but pB has 2 nonzero components, so the first claim does not hold. In Example 10, using a modification of a construction of Kannan in [13] we show a bounded polyhedron where the columns of the constraint matrix are LLL-reduced, but branching on a variable corresponding to the longest column produces exponentially many nodes. (Note that the polyhedron in [18] is unbounded.) Finally, in Remark 9 we clarify the connection with our results.

Example 9. Let $n = 6$,

$$p = (1,\ 1,\ 3,\ 3,\ 3,\ 3),$$
$$r = (-7, -4, -11, -6, -5, -1),$$
$$M = 24,$$
$$a = (17,\ 20,\ 61, 66, 67, 71),$$
$$B = \begin{bmatrix} 1 & 0 & -3 & 1 & 0 \\ 2 & -1 & -1 & -1 & 0 \\ -1 & -2 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 2 \\ -1 & 0 & 0 & -2 & 0 \\ 1 & 2 & 1 & 1 & -1 \end{bmatrix}.$$

The columns of B form an LLL-reduced basis of $\mathbb{N}(a)$. They form a basis, since with

$$v = (0, -3, 1, 0, 0, 0)^{\mathrm{T}},$$

the matrix $[B, v]$ is unimodular, and

$$a[B, v] = [0_{1\times(n-1)},\ \gcd(a)].$$

LLL-reducedness is straightforward to check using the definition. But

$$pB = (0, -1, -1,\ 0,\ 0),$$

so we cannot choose $n - 2 = 4$ columns of B which would form a basis of $\mathbb{N}(p) \cap \mathbb{N}(r)$.

Example 10. Let ρ be a real number in $(\sqrt{3}/2, 1)$, and define the columns of the matrix $B \in \mathbb{R}^{n\times n}$ as

$$\begin{aligned}
b_1 &= (\rho^0, 0, \ldots, 0)^{\mathrm{T}}\\
b_2 &= (\rho^0/2, \rho^1, \ldots, 0, 0)^{\mathrm{T}}\\
b_3 &= (\rho^0/2, \rho^1/2, \rho^2, \ldots, 0, 0)^{\mathrm{T}}\\
&\ \cdots\\
b_n &= (\rho^0/2, \rho^1/2, \rho^2/2, \ldots, \rho^{n-2}/2, \rho^{n-1})^{\mathrm{T}}.
\end{aligned} \tag{8.3}$$

Consider the polyhedron

$$Q = \{\lambda \mid 0 \leq B\lambda \leq e_n\}.$$

Proposition 1. *The following hold:*

(1) *The columns of B are an LLL-reduced basis of the lattice that they generate.*
(2) *b_n is the longest among the b_i.*
(3) $\mathrm{width}(e_n, Q) > c^n\|b_n\|$ *for some* $c > 1$.

Proof. We have

$$b_i^* = \rho^{i-1}e_i \quad (i = 1, \ldots, n),$$

and when writing $b_i = \sum_{j=1}^{i} \mu_{ij} b_j^*$,

$$\mu_{i,i-1} = 1/2. \tag{8.4}$$

Thus (1.32) in the definition of LLL-reducedness becomes

$$\|b_i^*\| \geq \frac{1}{\sqrt{2}}\|b_{i-1}^*\|, \tag{8.5}$$

which follows from $\rho \geq 1/\sqrt{2}$. Since

$$\|b_n\|^2 = \|b_{n-1}\|^2 + \rho^{2(n-2)}\left(\rho^2 - \frac{3}{4}\right), \tag{8.6}$$

this implies (2). By the definition of Gram–Schmidt orthogonalization, for any λ_n we can always set the other λ_i so

$$B\lambda = \lambda_n b_n^* = \lambda_n \rho^{n-1} e_n,$$

so

$$\mathrm{width}(e_n, Q) \geq \frac{2}{\rho^{n-1}}.$$

and (3) follows, since $\rho < 1$, and $\|b_n\| \leq n + 1$.

We can make B integral and still have (1) through (3) hold, by scaling, and rounding it. □

Remark 9. Our Theorem 11 proves that if $M > 2^{(n+1)/2}\|r\|^2\|p\|^2$, and the reformulation is computed using LLL-reduction, then $(pB)_{1:(n-2)} = 0$, and from this it does follow that the first $n - 2$ columns of B form a basis of $\mathbb{N}(p) \cap \mathbb{N}(r)$.

Theorem 11 then finishes the analysis in a different way, by directly proving small width, namely showing

$$\begin{aligned}
\mathrm{width}(e_{n-1}, \text{(KP-N)}) &\leq \mathrm{width}(p, \text{(KP)})\\
\mathrm{iwidth}(e_{n-1}, \text{(KP-N)}) &\leq \mathrm{iwidth}(p, \text{(KP)}).
\end{aligned} \tag{8.7}$$

Acknowledgements

We thank the two referees, and the Associate Editor for their suggestions, which greatly improved the paper. Thanks are due to Mustafa Tural for his careful reading, and suggestions, and finding reference [29]. The second author was supported by NSF award 0200308.

References

[1] Alexander Schrijver, Theory of Linear and Integer Programming, Wiley, Chichester, United Kingdom, 1986.
[2] Arjen K. Lenstra, Hendrik W. Lenstra Jr., László Lovász, Factoring polynomials with rational coefficients, Mathematische Annalen 261 (1982) 515–534.
[3] Ravi Kannan, Improved algorithms for integer programming and related lattice problems, in: Proceedings of the 15th Annual ACM Symposium on Theory of Computing, The Association for Computing Machinery, New York, 1983, pp. 193–206.
[4] Claus P. Schnorr, A hierarchy of polynomial time lattice basis reduction algorithms, Theoretical Computer Science 53 (1987) 201–225.
[5] Hendrik W. Lenstra Jr., Integer programming with a fixed number of variables, Mathematics of Operations Research 8 (1983) 538–548.
[6] Ravi Kannan, Minkowski's convex body theorem and integer programming, Mathematics of Operations Research 12 (3) (1987) 415–440.
[7] Alexander I. Barvinok, A polynomial time algorithm for counting integral points in polyhedra when the dimension is fixed, Mathematics of Operations Research 19 (4) (1994) 769–779.
[8] Martin Dyer, Ravi Kannan, On Barvinok's algorithm for counting lattice points in fixed dimension, Mathematics of Operations Research 22 (3) (1997) 545–549.
[9] Jesus A. De Loera, Raymond Hemmecke, Jeremy Tauzer, Ruriko Yoshida, Effective lattice point counting in rational convex polytopes, Journal of Symbolic Computation 38 (4) (2004) 1273–1302.
[10] László Lovász, Herbert E. Scarf, The generalized basis reduction algorithm, Mathematics of Operations Research 17 (1992) 751–764.
[11] William Cook, Thomas Rutherford, Herbert E. Scarf, David F. Shallcross, An implementation of the generalized basis reduction algorithm for integer programming, ORSA Journal on Computing 5 (2) (1993) 206–212.
[12] Sanjay Mehrotra, Zhifeng Li, On generalized branching methods for mixed integer programming, Research Report, Department of Industrial Engineering, Northwestern University, 2004.
[13] Ravi Kannan, Algorithmic geometry of numbers, Annual Review of Computer Science 2 (1987) 231–267.
[14] Karen Aardal, Cor A.J. Hurkens, Arjen K. Lenstra, Solving a system of linear Diophantine equations with lower and upper bounds on the variables, Mathematics of Operations Research 25 (3) (2000) 427–442.
[15] Gérard Cornuéjols, Milind Dawande, A class of hard small 0–1 programs, INFORMS Journal on Computing 11 (2) (1999) 205–210.
[16] Karen Aardal, Robert E. Bixby, Cor A.J. Hurkens, Arjen K. Lenstra, Job W. Smeltink, Market split and basis reduction: Towards a solution of the Cornuéjols-Dawande instances, INFORMS Journal on Computing 12 (3) (2000) 192–202.
[17] Quentin Louveaux, Laurence A. Wolsey, Combining problem structure with basis reduction to solve a class of hard integer programs, Mathematics of Operations Research 27 (3) (2002) 470–484.
[18] Karen Aardal, Arjen K. Lenstra, Hard equality constrained integer knapsacks, Mathematics of Operations Research 29 (3) (2004) 724–738.
[19] Karen Aardal, Arjen K. Lenstra, Erratum to: Hard equality constrained integer knapsacks, Mathematics of Operations Research 31 (4) (2006) 846.
[20] Robert G. Jeroslow, Trivial integer programs unsolvable by branch-and-bound, Mathematical Programming 6 (1974) 105–109.
[21] Vašek Chvátal, Hard knapsack problems, Operations Research 28 (6) (1980) 1402–1411.
[22] Gérard Cornuéjols, Regina Urbaniak, Robert Weismantel, Laurence A. Wolsey, Decomposition of integer programs and of generating sets, in: Algorithms—ESA '97, in: Lecture notes in Computer Science, 1284, Springer-Verlag, 1997, pp. 92–103.
[23] William J. Cook, Ravi Kannan, Alexander Schrijver, Chvátal closures for mixed integer programming problems, Mathematical Programming 47 (1990) 155–174.
[24] Alberto Caprara, Adam N. Letchford, On the separation of split cuts and related inequalities, Mathematical Programming 94 (2003) 279–294.
[25] Egon Balas, Disjunctive programming, Annals of Discrete Mathematics 5 (1979) 1–22.
[26] Egon Balas, Sebastian Ceria, Gérard Cornuéjols, A lift-and-project cutting plane algorithm for 0–1 programs, Mathematical Programming 58 (1993) 295–324.
[27] László Babai, On Lovász lattice reduction, and the nearest lattice point problem, Combinatorica 6 (1986) 1–13.
[28] Ralph E. Gomory, Solving linear programming problems in integers, in: Combinatorial Analysis, Proceedings of Symposia in Applied Mathematics X, American Mathematical Society, Providence, R. I., 1960, pp. 211–215.
[29] Gordon Bradley, Equivalent integer programs and canonical problems, Management Science 17 (1971) 354–366.
[30] Utz-Uwe Haus, Matthias Köppe, Robert Weismantel, A primal all-integer algorithm based on irreducible solutions, Mathematical Programming B 96 (2003) 205–246.
[31] Utz-Uwe Haus, An Augmentation Framework for Integer Programming, Dissertation, Otto-von-Guericke Universität Magdeburg, Der Andere Verlag, Toenning, ISBN: 3-89959-269-7, 2004.
[32] Matthias Beck, Sinai Robins, Computing the Continuous Discretely: Integer-point Enumeration in Polyhedra, Springer, 2006.
[33] Victor Shoup, NTL: A Number Theory Library, 1990, http://www.shoup.net.
[34] DKP instances, http://www.math.wsu.edu/faculty/bkrishna/DKP/.