# Hard equality constrained integer knapsacks

Karen Aardal[*] Arjen K. Lenstra[†]

September 12, 2002

**Abstract**

We consider the following integer feasibility problem: "Given positive integer numbers $a_0, a_1, \ldots, a_n$, with $\gcd(a_1, \ldots, a_n) = 1$ and $\boldsymbol{a} = (a_1, \ldots, a_n)$, does there exist a vector $\boldsymbol{x} \in \mathbb{Z}^n_{\geq \mathbf{0}}$ satisfying $\boldsymbol{ax} = a_0$?" Some instances of this type have been found to be extremely hard to solve by standard methods such as branch-and-bound, even if the number of variables is as small as ten. We observe that not only the sizes of the numbers $a_0$, $a_1, \ldots, a_n$, but also their structure, have a large impact on the difficulty of the instances. This particular structure enables us to derive a strong lower bound on the Frobenius number for these instances. Moreover, we demonstrate that the same structural characteristics that make the instances so difficult to solve by branch-and-bound make the solution of a certain reformulation of the problem almost trivial. We accompany our results by a small computational study.

*AMS 2000 Subject classification:* Primary: 90C10. Secondary: 45A05, 11Y50.
*OR/MS subject classification:* Programming, Integer, Theory.
*Key words:* Lattice basis reduction, Branching on hyperplanes, Frobenius number.

[*] Mathematisch Instituut, Universiteit Utrecht, Budapestlaan 6, 3584 CD Utrecht, The Netherlands. e-mail: `aardal@math.uu.nl`.

[†] Corporate Information Security Office, Citibank N.A., 1 North Gate Road, Mendham, NJ 07945-3104, USA, and Faculteit Wiskunde en Informatica, Technische Universiteit Eindhoven, Postbus 513, 5600 MB Eindhoven, The Netherlands. e-mail: `arjen.lenstra@citigroup.com`.

# 1 Introduction

## 1.1 Problem Statement and Summary of Results

In the past decade there has been substantial progress in computational integer programming. Many large and complex instances can now be solved. There are, however, still many small instances that seem extremely hard to tackle by standard methods such as branch-and-bound or branch-and-cut, and it is still quite unclear what makes these instances so hard. Examples are the so-called market share problems, Cornuéjols and Dawande (1999), and Aardal et al. (2000a), some feasibility problems reported on by Aardal et al. (2000b), and certain portfolio planning problems, Louveaux and Wolsey (2000). All of these are related to the following simple problem. Let $a_0, a_1, \ldots, a_n$ be positive integer numbers with $\boldsymbol{a} = (a_1, \ldots, a_n)$ and $\gcd(a_1, \ldots, a_n) = 1$ and $a_i \leq a_0$, $1 \leq i \leq n$, and let

$$P = \{\boldsymbol{x} \in \mathbb{R}^n : \ \boldsymbol{a}\boldsymbol{x} = a_0, \boldsymbol{x} \geq \boldsymbol{0}\}. \tag{1}$$

The problem is:

$$\text{Does } P \text{ contain an integer vector?} \tag{2}$$

If the components of $\boldsymbol{x}$ may take any integer value, then the problem is easy. There exists a vector $\boldsymbol{x} \in \mathbb{Z}^n$ satisfying $\boldsymbol{a}\boldsymbol{x} = a_0$ if and only if $a_0$ is an integer multiple of $\gcd(a_1, \ldots, a_n)$. The non-negativity requirement on $\boldsymbol{x}$ makes the problem NP-complete. In this study we focus on infeasible instances to rule out that a search algorithm terminates fast because it found a feasible solution by luck. The largest value of $a_0$ such that the instance of (2) given by the input $a_1, \ldots, a_n$ is infeasible is called the *Frobenius number* of $a_1, \ldots, a_n$, denoted by $F(a_1, \ldots, a_n)$.

Infeasible instances with large ratios $a_0/a_i$, $1 \leq i \leq n$, are particularly hard for branch-and-bound. In this context we address two topics. The first one is to provide a sufficient explanation why certain coefficients $a_1, \ldots, a_n$ will yield large Frobenius numbers than others of comparable sizes. In Theorem 1 we demonstrate that the Frobenius number is relatively large if it is possible to decompose the $\boldsymbol{a}$-coefficients as $a_i = p_i M + r_i$ with $p_i, M \in \mathbb{Z}_{>0}$, $r_i \in \mathbb{Z}$, and with $M$ large compared to $p_i$ and $|r_i|$.

This leads to the second topic: we give a sufficient condition under which the lattice reformulation using the projection suggested by Aardal, Hurkens, and Lenstra (2000b) will work significantly better than branch-and-bound on instances of type (2). We show that with $a_0, a_1, \ldots, a_n$ as above, the reformulation by Aardal, Hurkens, and Lenstra is computationally very easy to solve in a way similar to Lenstra's algorithm (H.W. Lenstra, Jr. (1983)), since the projected polytope will be thin in the direction of the last coordinate. This is demonstrated in Section 3.2. So, these instances that are very difficult for branch-and-bound are easy if first reformulated according to Aardal, Hurkens and Lenstra. Their reformulation, based on lattice basis reduction, is briefly described in Section 2. We also see that instances with $\boldsymbol{a}$-coefficients that decompose in the more general way: $a_i = p_i M + r_i N$ with $p_i, M, N \in \mathbb{Z}_{>0}$, $r_i \in \mathbb{Z}$, and with $M$ and $N$ large compared to $p_i$ and $|r_i|$, will be easy to solve after applying the reformulation. For this decomposition, however, we have not yet been able to prove that it leads to large Frobenius numbers.

To illustrate our observations we report on a modest computational study on infeasible instances from the literature, and infeasible instances that we generated ourselves. About half of the instances have $\boldsymbol{a}$-coefficients that decompose as discussed above and in Section 3, and the others have random coefficients of comparable sizes. All instances have $5 \leq n \leq 10$. The computational results, presented in Section 4, clearly confirm our theoretical observations.

Before presenting our results we will, in the following subsection, give a short description of some known results on integer programming.

### 1.2 Integer Programming and Branching on Hyperplanes

The polytope $P \subseteq \mathbb{R}^n$ as defined by (1) has dimension $n-1$, i.e., it is not full-dimensional. In the full-dimensional case the following is known. Let $S$ be a full-dimensional polytope in $\mathbb{R}^n$ given by integer input. The *width* of $S$ along the nonzero vector $\boldsymbol{d}$ is defined as $W(S, \boldsymbol{d}) = \max\{\boldsymbol{d}^T\boldsymbol{x} : \boldsymbol{x} \in S\} - \min\{\boldsymbol{d}^T\boldsymbol{x} : \boldsymbol{x} \in S\}$. Notice that this is different from the definition of the geometric width of a polytope. Consider the problem: "Does the polytope $S$ contain a vector $\boldsymbol{x}$ in the integer lattice $\mathbb{Z}^n$?" Khinchine (1948) proved that if $S$ does not contain a lattice point, then there exists a nonzero integer vector $\boldsymbol{d}$ such that $W(S, \boldsymbol{d})$ is bounded from above by a constant depending only on the dimension. H. W. Lenstra, Jr., (1983) developed an algorithm, exploiting this fact, for determining whether a given polytope $S$ contains an integer vector or not. The algorithm either finds an integer vector in $S$, or a lattice hyperplane $H$ such that at most $c(n)$ lattice hyperplanes parallel to $H$ intersect $S$, where $c(n)$ is a constant depending only on the dimension $n$. The intersection of each lattice hyperplane with $S$ gives rise to a problem of dimension at most $n-1$, and each of these lower-dimensional problems is solved recursively to determine whether or not $S$ contains an integer vector. One can illustrate the algorithm by a search tree having at most $n$ levels. The number of nodes created at each level is bounded from above by a constant depending only on the dimension at that level. Hence, the algorithm is polynomial for fixed dimension. A search node is pruned if, in the given direction, no lattice hyperplane is intersecting the polytope defined by the search node.

We are not aware of any implementation of Lenstra's algorithm. Cook et al. (1993) implemented the integer programming algorithm by Lovász and Scarf (1992), which is similar in structure to Lenstra's algorithm, and they observed that, for their instances, the number of search nodes created by the Lovász-Scarf algorithm was much less than the number of nodes of a branch-and-bound tree. To compute a good search direction in each node was, however, more time consuming than computing an LP-relaxation. This raises the question of understanding if there are situations in which good search directions can be determined fast. This is related to one of the results presented in this paper, as we demonstrate that for a class of very difficult infeasible instances, i.e., the instances that have decomposable $\boldsymbol{a}$-coefficients as outlined above, the projection proposed by Aardal, Hurkens, and Lenstra by itself yields an integer direction in which the projected polytope is provably thin. In our case this direction is the last coordinate direction. So, if we apply a tree search algorithm, such as Lenstra's, to the projected polytope, but branch only in coordinate directions in the order of decreasing variable indices, then the instances become very easy.

### 1.3 Notation

We conclude this section by introducing some definitions and notation. The Euclidean length of a vector $\boldsymbol{x} \in \mathbb{R}^n$ is denoted by $|\boldsymbol{x}|$, the $n \times n$ *identity matrix* by $\boldsymbol{I}^{(n)}$, the zero $p \times q$ matrix by $\mathbf{0}^{(p \times q)}$, where the dimensions are omitted if they are clear from the context.

The $m \times n$ matrix $\boldsymbol{A}$ is said to be in *Hermite normal form* if it has the form $(\boldsymbol{C}^{m\times m}, \mathbf{0}^{m\times(n-m)})$, where $\boldsymbol{C}$ is a lower triangular, non-negative matrix in which each diagonal element is the unique maximum row entry. The Hermite normal form of the matrix $\boldsymbol{A}$ is denoted by HNF $(\boldsymbol{A})$, and it is unique if $\boldsymbol{A}$ has full row rank.

A set of the form $L = L(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_l) = \{\sum_{i=1}^{l} \lambda_i \boldsymbol{b}_i,\ \lambda_i \in \mathbb{Z}, 1 \leq i \leq l\}$, where $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_l$ are linear independent vectors in $\mathbb{R}^n$, $l \leq n$, is called a *lattice.* The set of vectors $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_l\}$ is called a *lattice basis.* A lattice has two different bases if $l = 1$, and infinitely many if $l > 1$.

The *determinant* $d(L)$ of the lattice $L$ is defined as $d(L) = \sqrt{\det(\boldsymbol{B}^T\boldsymbol{B})}$, where $\boldsymbol{B}$ is a basis for $L$, and where $\boldsymbol{B}^T$ denotes the transpose of matrix $\boldsymbol{B}$. If the lattice $L$ is full-dimensional we have $d(L) = |\det \boldsymbol{B}|$. The *rank* $\operatorname{rk} L$ of the lattice $L$ is the dimension of the Euclidean vector space spanned by $L$. If $\operatorname{rk} L = 0$, then $d(L)$ is defined to be equal to one.

The *integer width* of a polytope $S \subset \mathbb{R}^n$ in the non-zero integer direction $\boldsymbol{d} \in \mathbb{Z}^n$ is defined as:

$$W_I(S, \boldsymbol{d}) = \lfloor \max\{\boldsymbol{d}^T\boldsymbol{x} : \ \boldsymbol{x} \in S\} \rfloor - \lceil \min\{\boldsymbol{d}^T\boldsymbol{x} : \ \boldsymbol{x} \in S\} \rceil .$$

The number of lattice hyperplanes in the direction $\boldsymbol{d}$ that intersect $S$ is equal to $W_I(S, \boldsymbol{d}) + 1$, so if $W_I(S, \boldsymbol{d}) = -1$, then $S$ does not contain an integer vector.

## 2 The Reformulation and the Search Algorithm

The starting point of the reformulation of (2) suggested by Aardal, Hurkens, and Lenstra (2000b) is the sign relaxation $X_I = \{\boldsymbol{x} \in \mathbb{Z}^n : \ \boldsymbol{a}\boldsymbol{x} = a_0\}$ of $X = \{\boldsymbol{x} \in \mathbb{Z}^n_{\geq 0} : \ \boldsymbol{a}\boldsymbol{x} = a_0\}$. The relaxation $X_I$ can be rewritten as $X_I = \{\boldsymbol{x} \in \mathbb{Z}^n : \ \boldsymbol{x} = \boldsymbol{x}_f + \boldsymbol{B}_0\boldsymbol{y}, \ \boldsymbol{y} \in \mathbb{Z}^{n-1}\}$, where $\boldsymbol{x}_f$ is an integer vector satisfying $\boldsymbol{a}\boldsymbol{x}_f = a_0$, and where $\boldsymbol{B}_0$ is a basis for the lattice $L_0 = \{\boldsymbol{x} \in \mathbb{Z}^n : \ \boldsymbol{a}\boldsymbol{x} = 0\}$. That is, there is an integer vector $\boldsymbol{x}_f$ such that any vector $\boldsymbol{x} \in X_I$ can be written as the sum of $\boldsymbol{x}_f$ and a vector $\boldsymbol{x}_0 \in L_0$. Since $\gcd(a_1, \ldots, a_n) = 1$ and $a_0$ is integer, we know that a vector $\boldsymbol{x}_f$ exists. In the paper by Aardal et al. it is shown that $\boldsymbol{x}_f$ and $\boldsymbol{B}_0$ can conveniently be determined in polynomial time using lattice basis reduction.

Let

$$Q = \{\boldsymbol{y} \in \mathbb{R}^{n-1} : \ \boldsymbol{B}_0\boldsymbol{y} \geq -\boldsymbol{x}_f\} . \tag{3}$$

Problem (2) can now be restated as:

Does $Q$ contain an integer vector?

The polytope $Q$ is a full-dimensional formulation, i.e., the dimension of $Q$ is $n-1$, and as mentioned in the previous section we can apply Lenstra's (Lenstra (1983)) algorithm, or any other integer programming algorithm, to $Q$. Here we will consider a tree search algorithm inspired by Lenstra's algorithm, but using only unit directions in the search.

Let $\boldsymbol{e}_i$, $0 \leq i \leq n-1$, be the $i$th unit vector, let $J = \{1, 2, \ldots, n-1\}$, (assume $n > 1$) and recursively define a feasibility search process $\texttt{Search}(S)$ on a set $S \subseteq J$ as follows:

```
Search($S$) :
  if $S$ is empty, output the point $\{k_j\}_{j\in J}$, print 'feasible' and quit
  otherwise:
    pick an $i \in S$
    compute $l_i = \lceil \min\{\boldsymbol{e}_i^T\boldsymbol{y} :\ \boldsymbol{y} \in Q,\ \text{and } y_j = k_j \text{ for all } j \in J \setminus S\}\rceil$
    compute $u_i = \lfloor \max\{\boldsymbol{e}_i^T\boldsymbol{y} :\ \boldsymbol{y} \in Q,\ \text{and } y_j = k_j \text{ for all } j \in J \setminus S\}\rfloor$
    for all integers $k_i$ in the interval $[l_i, u_i]$ do Search($S \setminus \{i\}$)
  print 'infeasible' and quit
```

The feasibility search is then defined as `Search`($J$). For an example of a search tree, see Figure 1. Notice that the search tree created in this way is similar to the search tree of Lenstra's algorithm in that the number of levels of the tree is no more than the number of variables in the problem instance, and that the number of nodes created at a certain level corresponds to the integer width of the polytope in the chosen search direction.

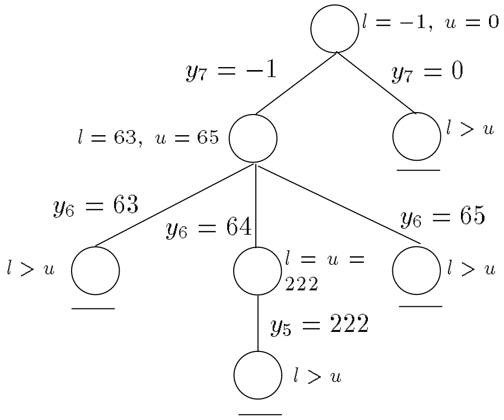


Figure 1: The search tree for instance prob2 (cf. Section 4)

Here we will investigate a class of instances that are exceptionally hard to solve by branch-and-bound when using the original formulation in $\boldsymbol{x}$-variables, but that become easy to solve when applying the branching scheme described above to the reformulated problem in $\boldsymbol{y}$-variables (3). In our implementation of the algorithm `Search`($S$), we always choose the index $i$ as the highest index in the set $S$ when we are at the step "pick an index $i \in S$", i.e., we branch in the order $n-1, \ldots, 1$. This is done because the width in the unit direction $\boldsymbol{e}_{n-1}$ is small for our class of instances as will be demonstrated in following section. Below we give an example of such an instance.

**Example 1** Let

$$P = \{\boldsymbol{x} \in \mathbb{R}^3 : \; 12223x_1 + 12224x_2 + 36672x_3 = 149389505, \boldsymbol{x} \geq \boldsymbol{0}\} \; .$$

A vector $\boldsymbol{x}_f$ and a basis $\boldsymbol{B}_0$ for this instance is:

$$\boldsymbol{x}_f = \begin{pmatrix} -1 \\ 1221 \\ 3667 \end{pmatrix} \qquad \boldsymbol{B}_0 = \begin{pmatrix} 0 & 12224 \\ -3 & -1222 \\ 1 & -3667 \end{pmatrix} \; .$$

The polytope $Q$ is:

$$Q = \{\boldsymbol{y} \in \mathbb{R}^2 : \; 12224y_2 \geq 1, \; -3y_1 - 1222y_2 \geq -1221, \; y_1 - 3667y_2 \geq -3667\} \; .$$

Moreover, we have $W_I(Q, \boldsymbol{e}_1) = 4072$ and $W_I(Q, \boldsymbol{e}_2) = -1$, so if we consider the search direction $\boldsymbol{e}_2$ first, we can immediately conclude that $Q \cap \mathbb{Z}^2 = \emptyset$.

If we solve the formulation in $\boldsymbol{x}$-variables by branch-and-bound with objective function 0 using the default settings of CPLEX 6.5, it takes 1,262,532 search nodes to verify infeasibility. □

An instance such as the one given in Example 1 may seem quite artificial. However, some of the instances reported on by Cornuéjols and Dawande (1999), Aardal et al. (2000a,b), and by Louveaux and Wolsey (2000) stem from applications and show a similar behavior. From a practical point of view it is therefore relevant to try to explain this behavior.

# 3 The Class of Instances

## 3.1 The Coefficient $a_0$

The polytope $P$ as given in (1) is an $n$-simplex. An instance of problem (2) is particularly hard to solve by branch-and-bound if it is infeasible and if the intersection points of the $n$-simplex with the coordinate axes have large values. Branch-and-bound will then be forced to enumerate many of the possible combinations of $x_1, \ldots, x_n$ with $0 \leq x_i \leq a_0/a_i$. Since the instance is infeasible we cannot “get lucky” in our search, which may happen if the instance is feasible, and if we by chance have chosen an objective function that takes us to a feasible solution quickly. Example 1 of the previous section illustrates such a hard infeasible instance. Similar, but larger, instances are virtually impossible to solve using a state-of-the-art branch-and-bound algorithm such as implemented in CPLEX.

To create infeasible instances with maximum values of $a_0/a_i$ we choose $a_0$ as the Frobenius number $F(a_1, \ldots, a_n)$. Computing the Frobenius number for given natural numbers $a_1, \ldots, a_n$ with $\gcd(a_1, \ldots, a_n) = 1$ is NP-hard. In Appendix 1 we discuss the algorithm that we used in our computational study. For $n = 2$ it is known that $F(a_1, a_2) = a_1a_2 - a_1 - a_2$. (In "Mathematics from the Educational Times, with Additional Papers and Solutions", Sylvester published the problem of proving that if $a_1$ and $a_2$ are relatively prime integers, then there are exactly $1/2(a_1 - 1)(a_2 - 1)$ non-negative integers $\alpha$ less than $a_1a_2 - a_1 - a_2$ for which $a_1x_1 + a_2x_2 = \alpha$ does not have a non-negative integer solution. The solution to this problem was provided by Curran Sharp in volume 41 (1884) of the journal. The precise reference is Sylvester (1884). See also Schrijver (1986) p. 376.) For $n = 3$ the Frobenius number can be computed in polynomial time, see Selmer and Beyer (1978), Rödseth (1978), and Greenberg (1988). Kannan (1992) developed a polynomial time algorithm for computing the Frobenius number for every *fixed n*. His algorithm is based on the relation between the Frobenius number and the covering radius of a certain polytope. Some upper bounds on the Frobenius number are also known. If $a_1 < a_2 < \cdots < a_n$, Brauer (1942) showed that $F(a_1, \ldots, a_n) \leq a_1a_n - a_1 - a_n$. Other upper bounds were provided by Erdős and Graham (1972) and Selmer (1977).

Below we determine a lower bound on $F(a_1, \ldots, a_n)$. Suppose we write $a_i$ as $a_i = p_iM + r_i$ for $1 \leq i \leq n$. We express the lower bound as a function of $\boldsymbol{p}$, $\boldsymbol{r}$ and $M$. The highest order term in $M$ is quadratic in $M$, so for large values of $M$, and relatively small values of $p_i$ and $|r_i|$, this term will be dominant.

**Theorem 1** *Let* $a_i = p_iM+r_i$, *for* $1 \leq i \leq n$, *let* $(r_j/p_j) = \max_{i=1,\ldots,n}\{r_i/p_i\}$, *and let* $(r_k/p_k) = \min_{i=1,\ldots,n}\{r_i/p_i\}$. *Assume that:*

1. $a_1 < a_2 < \cdots < a_n$,
2. $p_i, M \in \mathbb{Z}_{>0}$, $r_i \in \mathbb{Z}$ *for* $1 \leq i \leq n$,
3. $\sum_{i=1}^{n} |r_i| < 2M$,
4. $M > 2 - (r_j/p_j)$,
5. $M > (r_j/p_j) - 2(r_k/p_k)$ .

*Then, we obtain* $f(\boldsymbol{p}, \boldsymbol{r}, M) \leq F(a_1, \ldots, a_n) \leq g(\boldsymbol{p}, \boldsymbol{r}, M)$, *where*

$$f(\boldsymbol{p}, \boldsymbol{r}, M) = \frac{(M^2 p_j p_k + M(p_j r_k + p_k r_j) + r_j r_k)(1 - \frac{2}{M+(r_j/p_j)})}{p_k r_j - p_j r_k} - 1 ,$$

*and*

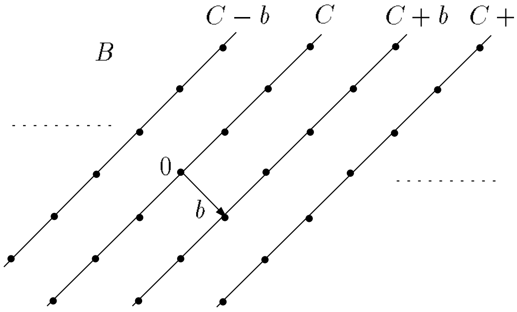
$$g(\boldsymbol{p}, \boldsymbol{r}, M) = M^2 p_1 p_n + M(p_1 r_n + p_n r_1 - p_1 - p_n) + r_1 r_n - r_1 - r_n .$$

**Proof:** The upper bound $g(\boldsymbol{p}, \boldsymbol{r}, M)$ is derived from the result by Brauer (1942) that $F(a_1, \ldots, a_n) \leq a_1 a_n - a_1 - a_n$.

In our proof of the lower bound we introduce the following notation:

$$\begin{aligned} B &= \{\boldsymbol{x} \in \mathbb{R}^n : \ \boldsymbol{ax} = 0\} \\ \Delta_t &= \{\boldsymbol{x} \in \mathbb{R}^n : \ \boldsymbol{ax} = t, \ \boldsymbol{x} \geq \boldsymbol{0}\} \\ C &= \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{px} = 0, \ \boldsymbol{rx} = 0\} . \end{aligned}$$

The lattice $L_0$ is defined as $B \cap \mathbb{Z}^n$ as in Section 2.



The lattice $L_0$ is contained in parallel hyperplanes generated by $C$

Figure 2: $B$, $C$, and $L_0$

The idea behind the proof is as follows. We define a homomorphism from $\Delta_t$ to $\mathbb{R}/\mathbb{Z}$ such that $\boldsymbol{x} \in \mathbb{Z}^n \cap \Delta_t$ maps to 0, if such a vector $\boldsymbol{x}$ exists. An integer number $t$ for which 0 is not contained in the image of $\Delta_t$ under this map then provides a lower bound on the Frobenius number. We define such a homomorphism by first defining a projection $\pi_{\boldsymbol{z}}$, along the vector $\boldsymbol{z}$, of $\Delta_1$ onto $B$, where $\boldsymbol{z}$ is in the same plane as $\Delta_1$. Then we consider a homomorphism $f : \ B \to \mathbb{R}/\mathbb{Z}$. We show that the kernel of $f$ is $L_0 + C$. Due to the First Isomorphism Theorem (see e.g. Hungerford (1996), p. 44) we know that $B$ divided out by (ker $f$), i.e., $B/(L_0+C)$, is isomorphic to $\mathbb{R}/\mathbb{Z}$. The image of $\pi_{\boldsymbol{z}}(\Delta_1)$ under the isomorphism $B/(L_0+C) \to \mathbb{R}/\mathbb{Z}$ turns out to be an interval $[l, u]$ in $\mathbb{R}/\mathbb{Z}$. Finally we determine an integer number $t$ such that $[tl, tu]$ does not contain an integer point. The integer $t$ then yields a lower bound on the Frobenius number under the conditions given in the theorem.

We first define a linear mapping $\pi : \mathbb{R}^n \to B$ given by $\pi_{\boldsymbol{z}}(\boldsymbol{x}) = \boldsymbol{x} - (\boldsymbol{a}\boldsymbol{x})\boldsymbol{z}$, where $\boldsymbol{z}$ satisfies $\boldsymbol{p}\boldsymbol{z} = 0$ and $\boldsymbol{r}\boldsymbol{z} = 1$, and hence $\boldsymbol{a}\boldsymbol{z} = 1$. When $M \to \infty$, then $\pi_{\boldsymbol{z}}(\Delta_1) \to -\boldsymbol{z}$. Notice that $-\boldsymbol{z} \notin C$ as $\boldsymbol{r}\boldsymbol{z} = 1$.

Next we define the homomorphism $f : B \to \mathbb{R}/\mathbb{Z}$ given by $\boldsymbol{x} \mapsto (\boldsymbol{p}\boldsymbol{x} \mod 1)$.

*Claim:* The kernel of $f$ is $L_0 + C$.

> First we show that $(L_0+C) \subseteq (\text{ker } f)$. If $\boldsymbol{x} \in L_0$ then $\boldsymbol{x} \in \mathbb{Z}$, which implies $\boldsymbol{p}\boldsymbol{x} \in \mathbb{Z}$, and hence $(\boldsymbol{p}\boldsymbol{x} \mod 1) = 0$. If $\boldsymbol{x} \in C$, then $\boldsymbol{p}\boldsymbol{x} = 0$.
>
> Next, show that $(\text{ker } f) \subseteq (L_0+C)$. Notice that each element in $B$ can be written as $\boldsymbol{c} + \boldsymbol{y}_1 + \boldsymbol{y}_2$ with $\boldsymbol{c} \in C$, $\boldsymbol{y}_1 \in L_0$, and $\boldsymbol{y}_2 \notin C$ such that the absolute value of each element of $\boldsymbol{y}_2$ is in the interval $(-1/2, 1/2)$. Since $f(\boldsymbol{c}+\boldsymbol{y}_1+\boldsymbol{y}_2) = f(\boldsymbol{c})+f(\boldsymbol{y}_1)+f(\boldsymbol{y}_2) = 0$ and since $f(\boldsymbol{c}) = f(\boldsymbol{y}_1) = 0$, as $\boldsymbol{c} \in C$ and $\boldsymbol{y}_1 \in L_0$, we obtain $f(\boldsymbol{y}_2) = 0$, and hence $(\boldsymbol{p}\boldsymbol{y}_2 \mod 1) = 0$. If $\boldsymbol{p}\boldsymbol{y}_2 = 0$, then, since $\boldsymbol{y}_2 \in B$, we have $\boldsymbol{r}\boldsymbol{y}_2 = 0$, but this contradicts $\boldsymbol{y}_2 \notin C$. So, $0 = \boldsymbol{a}\boldsymbol{y}_2 = M\boldsymbol{p}\boldsymbol{y}_2 + \boldsymbol{r}\boldsymbol{y}_2$, and since $\boldsymbol{p}\boldsymbol{y}_2$ is integral we have that $\boldsymbol{r}\boldsymbol{y}_2$ is an integer multiple of $M$. Now, observe that since the absolute value of each element of $\boldsymbol{y}_2$ is less than $1/2$, then, due to Assumption 3 of the theorem, $\boldsymbol{r}\boldsymbol{y}_2 < 1/2\sum_{i=1}^{n}|r_i| < M$, and therefore $\boldsymbol{y}_2 = 0$ is the only possible solution. This concludes the proof of our claim.

Due to the First Isomorphism Theorem, the homomorphism $f$ induces an isomorphism $f' : B/(L_0+C) \to \mathbb{R}/\mathbb{Z}$. Below we determine the image of $\Delta_1$ under the composition of the mappings $\pi : \mathbb{R}^n \to B$ and $B \to B/(L_0+C) \to \mathbb{R}/\mathbb{Z}$. This composition of mappings is a homomorphism.

We use $\boldsymbol{v}_i$ to denote vertex $i$ of $\Delta_1$. Vertex $\boldsymbol{v}_i$, is the vector $(0, \ldots, 0, 1/a_i, 0, \ldots, 0)^T$, where $1/a_i = 1/(p_iM + r_i)$ is the $i$th component of $\boldsymbol{v}_i$. Applying the linear mapping $\pi$ to $\boldsymbol{v}_i$ yields $\pi_{\boldsymbol{z}}(\boldsymbol{v}_i) = \boldsymbol{v}_i - \boldsymbol{z}$. Next, by the isomorphism $\boldsymbol{x} \mapsto (\boldsymbol{p}\boldsymbol{x} \mod 1)$, $\pi_{\boldsymbol{z}}(\boldsymbol{v}_i)$ becomes

$$\frac{p_i}{Mp_i + r_i} = \frac{1}{M + r_i/p_i}.$$

Let $d_i$ denote $1/(M + r_i/p_i)$, and recall that $(r_j/p_j) = \max_{i=1,\ldots,n}\{r_i/p_i\}$, and $(r_k/p_k) = \min_{i=1,\ldots,n}\{r_i/p_i\}$. Then, since $\Delta_1$ is the convex hull of the vertices $\boldsymbol{v}_i$, $1 \leq i \leq n$, and since $\pi$ is a homomorphism, the image of $\pi_{\boldsymbol{z}}(\Delta_1)$ is an interval $[d_j, d_k]$ of length

$$L = \frac{p_k r_j - p_j r_k}{M^2 p_j p_k + M(p_j r_k + p_k r_j) + r_j r_k} \ .$$

Now we will demonstrate that there exists an integer $t \geq \lfloor \frac{1-2d_j}{L} \rfloor$ such that the interval $[td_j, td_k]$ does not contain an integer point. This implies that $\lfloor \frac{1-2d_j}{L} \rfloor$ is a lower bound on the Frobenius number. Notice that $1 - 2d_j > 0$ due to Assumption 4.

Let $k = \lfloor \frac{1-2d_j}{L} \rfloor$. The interval $[I_1, \ I_2] = [kd_j, \ kd_k]$ has length less than or equal to $1 - 2d_j$. Let $\ell = \lfloor kd_j \rfloor$. Notice that $\ell \leq I_1$. Now define $k' = \ell/d_j$. The number $k' \leq k$ yields an interval $[I'_1, \ I'_2] = [k'd_j, \ k'd_k]$, such that $I'_1$ is integral. The interval $[I'_2, \ I'_2 + 2d_j)$ does not contain an integer since the length of $[I'_1, \ I'_2]$ is less than or equal to the length of the interval $[I_1, \ I_2]$, and since $I'_1$ is integral. Now define $k^* = \lfloor k' \rfloor + 1$. We claim that the interval $[I^*_1, \ I^*_2] = [k^*d_j, \ k^*d_k]$ does not contain an integer point. To prove the claim, first assume that $k'$ is integer. Then $k^*d_j = I'_1 + d_j$ and $k^*d_k < I'_1 + 1$ due to Assumption 5 that implies that $d_k < 2d_j$. Next, assume that $k'$ is fractional. In this case we obtain $I'_1 < k^*d_j < I'_1 + d_j$, and, due to the same reasoning as for $k'$ integer, we obtain $k^*d_k < I'_1 + 1$. This finishes the proof of our claim. We finally notice that $k^* \geq \lfloor \frac{1-2d_j}{L} \rfloor$, so we can conclude that $\lfloor \frac{1-2d_j}{L} \rfloor$ yields a lower bound on the Frobenius number. We obtain

$$\lfloor \frac{1-2d_j}{L} \rfloor \geq \frac{1-2d_j}{L} - 1 = \frac{(M^2 p_j p_k + M(p_j r_k + p_k r_j) + r_j r_k)(1 - \frac{2}{M + (r_j/p_j)})}{p_k r_j - p_j r_k} - 1 \ .$$

□

**Example 2** The $\boldsymbol{a}$-coefficients in Example 1 decompose as follows. Let $M = 12223$.

$$\begin{aligned} a_1 &= M + 0 \\ a_2 &= M + 1 \\ a_3 &= 3M + 3. \end{aligned}$$

Theorem 1 yields a lower bound on the Frobenius number equal to $149,381,362$ and an upper bound equal to 448,192,961. The Frobenius number for this instance is $149,389,505$, which is very close to the lower bound. □

For all our instances that decompose with vectors $\boldsymbol{p}$ and $\boldsymbol{r}$ that are short compared to $M$, the Frobenius number is large, see the computational study in Section 4. We have computed the lower bound on the Frobenius number for these instances and in all cases it was close to the actual value. It would be interesting to investigate whether it is possible to use similar techniques to tighten the upper bound on the Frobenius number for instances of this sort.

In the following subsection we demonstrate that instances with $\boldsymbol{a}$-coefficients that decompose with large $M$ and relatively short $\boldsymbol{p}$ and $\boldsymbol{r}$ are trivial to solve using the reformulation outlined in Section 2. These are the instances that are extremely hard to solve by branch-and-bound due to the large Frobenius numbers.

### 3.2 The Coefficients $a_1, \ldots, a_n$

For the further analysis of our class of instances we wish to express the determinant of the lattices $L_0$ and a sublattice of $L_0$ in terms of the input. Before presenting our results, we introduce more notation, two definitions, and present some known results. For more details, see for instance Cassels (1997).

**Definition 1** *Let $L$ be a lattice in a Euclidean vector space $E$, and let $K$ be subgroup of $L$. If there exists a subspace $D$ of $E$ such that $K = L \cap D$, then $K$ is called a* pure *sublattice.*

**Definition 2** *Let $L$ be a lattice in $\mathbb{R}^n$. Its* dual lattice *$L^\dagger$ is defined as follows:*

$$L^\dagger = \{\boldsymbol{x} \in \mathbb{R}^n : \ \boldsymbol{x}^T\boldsymbol{y} \in \mathbb{Z} \textit{ for all } \boldsymbol{y} \in L\} .$$

Suppose that $K$ is a pure sublattice of the lattice $L$. Then the following holds:

$$d(L) = d(K) \cdot d(L/K) . \tag{4}$$

For a lattice $L$ and its dual $L^\dagger$ we have

$$d(L) = \frac{1}{d(L^\dagger)} . \tag{5}$$

Let $L$ be a lattice with dual $L^\dagger$, and let $K$ be a pure sublattice of $L$. Then $K^\perp = \{x \in L^\dagger : \ \boldsymbol{x}^T\boldsymbol{y} = 0 \text{ for all } \boldsymbol{y} \in K\}$, and we can write

$$K^\perp = (L/K)^\dagger . \tag{6}$$

**Theorem 2** $d(L_0) = d(L(\boldsymbol{a}^T)) = |\boldsymbol{a}^T|$ .

**Proof:** Take $L$ to be the lattice $\mathbb{Z}^n$, and $K$ to be the lattice $L_0$. By equation (4) we have $1 = d(\mathbb{Z}^n) = d(L_0) \cdot d(\mathbb{Z}^n/L_0)$, or equivalently, by equation (5):

$$d(L_0) = \frac{1}{d(\mathbb{Z}^n/L_0)} = d((\mathbb{Z}^n/L_0)^\dagger) .$$

From (6) we obtain $(\mathbb{Z}^n/L_0)^\dagger = L_0^\perp$, and since the dual lattice of $\mathbb{Z}^n$ is again $\mathbb{Z}^n$ we have $L_0^\perp = \{\boldsymbol{x} \in \mathbb{Z}^n : \boldsymbol{x}^T\boldsymbol{y} = 0 \text{ for all } \boldsymbol{y} \in L_0\}$. Since $\gcd(a_1, \ldots, a_n) = 1$ this is exactly the lattice $L(\boldsymbol{a}^T)$ with basis $\boldsymbol{a}^T$. So, $L_0^\perp = d(L(\boldsymbol{a}^T)) = \sqrt{\boldsymbol{a}\boldsymbol{a}^T} = |\boldsymbol{a}^T|$. We have obtained

$$d(L_0) = d(L(\boldsymbol{a}^T)) .$$

□

This result is also mentioned in Section 3.2 of the survey by Nguyen and Stern (2000).

**Remark 1** *Notice that $d(L_0)$ can also be computed as $d(L_0) = \sqrt{det(\boldsymbol{B}_0^T\boldsymbol{B}_0)}$, where $\boldsymbol{B}_0$ is a basis for $L_0$.*

Write $a_i = p_iM + r_i$ with $p_i, M \in \mathbb{Z}_{>0}$ and $r_i \in \mathbb{Z}$. Let $C$ denote the orthogonal complement of the hyperplane spanned by $\boldsymbol{p}$ and $\boldsymbol{r}$, as in the proof of Theorem 1. We denote the lattice $C \cap \mathbb{Z}^n$ by $L_C$.

**Proposition 3** *The lattice $L_0$ contains the lattice $L_C$. The rank of the lattice $L_C$ is equal to $n - 2$.*

**Proof:** Assume that $\boldsymbol{x} \in \mathbb{Z}^n$ satisfies $\boldsymbol{p}\boldsymbol{x} = 0$ and $\boldsymbol{r}\boldsymbol{x} = 0$. Then, $\boldsymbol{a}\boldsymbol{x} = (M\boldsymbol{p} + \boldsymbol{r})\boldsymbol{x} = 0$ which shows that $L_0 \supseteq L_C$.

To prove that $\text{rk } L_C = n - 2$, assume, to create a contradiction, that $\boldsymbol{p}$ and $\boldsymbol{r}$ are linearly dependent, that is, we can write $\boldsymbol{r} = \lambda\boldsymbol{p}$ for some $\lambda \in \mathbb{Q}$.

Write $\lambda$ as $\lambda = \lambda_F + \lambda_I$, where $\lambda_I \in \mathbb{Z}$ and $0 \leq \lambda_F < 1$. Write $\lambda_F$ as $\lambda_F = f/g$ with $f = 0$ if $\lambda$ is integer. Since $r_i = \lambda p_i$ is integer, $p_i$ has to be an integer multiple of $g$, i.e., $p_i = k_ig$ with $k_i \in \mathbb{Z}$. We can now express $a_i$ as $a_i = k_igM + (\lambda_I + \lambda_F)k_ig = k_i(gM + g(\lambda_I + \lambda_F))$ with $gM + g(\lambda_I + \lambda_F) = gM + g\lambda_I + f$ being integer. This contradicts that $\gcd(a_1, \ldots, a_n) = 1$. Hence $\boldsymbol{p}$ and $\boldsymbol{r}$ are linearly independent, which implies that the rank of $L_C$ is equal to $n - 2$. □

Let

$$\boldsymbol{P} = \begin{pmatrix} p_1 & p_2 & \cdots & p_n \\ r_1 & r_2 & \cdots & r_n \end{pmatrix}$$

**Theorem 4** $d(L_C) \leq d(L(\boldsymbol{P}^T)) = \sqrt{det(\boldsymbol{P}\boldsymbol{P}^T)} = \sqrt{|\boldsymbol{p}|^2 \cdot |\boldsymbol{r}|^2 - (\boldsymbol{p}\boldsymbol{r}^T)^2}$.

**Proof:** This proof follows the same lines as the proof of Theorem 2. Here we choose the lattice $L$ from Definitions 1 and 2 to be the lattice $\mathbb{Z}^n$, and the sublattice $K$ to be the lattice $L_C$. We have

$$d(L_C) = d((\mathbb{Z}^n/L_C)^\dagger) = d(L_C^\perp)\,,$$

and since $(\mathbb{Z}^n)^\dagger = \mathbb{Z}^n$, we obtain $L_C^\perp = \{\boldsymbol{x} \in \mathbb{Z}^n : \ \boldsymbol{x}^T\boldsymbol{y} = 0 \text{ for all } \boldsymbol{y} \in L_C\}$. Let $\boldsymbol{B}_C$ be a basis for $L_C^\perp$. The vectors $\boldsymbol{p}$ and $\boldsymbol{r}$ are linearly independent (see proof of Proposition 3) and belong to the lattice $L_C^\perp$ generated by $\boldsymbol{B}_C$. This implies that $d(L(\boldsymbol{P}^T)) = k \cdot d(L_C^\perp)$ with $k \geq 1$.

To summarize, we have

$$d(L_C) = d((\mathbb{Z}^n/L_C)^\dagger) = d(L_C^\perp) = \frac{d(L(\boldsymbol{P}^T))}{k} \leq d(L(\boldsymbol{P}^T))\,.$$

The determinant of the lattice $L(\boldsymbol{P}^T)$ is equal to $\sqrt{\det(\boldsymbol{P}\boldsymbol{P}^T)} = \sqrt{|\boldsymbol{p}|^2 \cdot |\boldsymbol{r}|^2 - (\boldsymbol{p}\boldsymbol{r}^T)^2}$. □

Let $\boldsymbol{b}_0^1, \boldsymbol{b}_0^2, \ldots, \boldsymbol{b}_0^{n-1}$ be a basis for $L_0$, and assume without loss of generality that these basis vectors are ordered such that $\boldsymbol{b}_0^1, \boldsymbol{b}_0^2, \ldots, \boldsymbol{b}_0^{n-2}$ form a basis for the lattice $L_C$. Hence, $\boldsymbol{b}_0^{n-1}$ does not belong to $L_C$. Let $H = \sum_{i=1}^{n-2} \mathbb{R}\boldsymbol{b}_0^i$ and let $h$ be the distance of $\boldsymbol{b}_0^{n-1}$ to $H$. Notice that $h \leq |\boldsymbol{b}_0^{n-1}|$.

**Corollary 5**

$$|\boldsymbol{b}_0^{n-1}| \geq \frac{|\boldsymbol{a}^T|}{\sqrt{|\boldsymbol{p}|^2 \cdot |\boldsymbol{r}|^2 - (\boldsymbol{p}\boldsymbol{r}^T)^2}}\,.$$

**Proof:** The following holds:

$$d(L_0) = d(L_C) \cdot h \leq d(L_C) \cdot |\boldsymbol{b}_0^{n-1}|\,.$$

So,

$$|\boldsymbol{b}_0^{n-1}| \geq \frac{d(L_0)}{d(L_C)} \geq \frac{d(L_0)}{d(L(\boldsymbol{P}^T))} = \frac{|\boldsymbol{a}^T|}{\sqrt{|\boldsymbol{p}|^2 \cdot |\boldsymbol{r}|^2 - (\boldsymbol{p}\boldsymbol{r}^T)^2}}\,.$$

□

Suppose $\boldsymbol{p}$ and $\boldsymbol{r}$ are short relative to $M$. Then, Lovász' basis reduction algorithm (Lenstra et al. (1982)) yields a basis in which the basis vectors are ordered according to increasing length, up to a certain factor. In a basis $\boldsymbol{B}_0$ for $L_0$, such as we generate it, the first $n-2$ vectors form a basis for the lattice $L_C$. These vectors are short, since the basis is reduced and since the determinant of the lattice $L_C$ is bounded from above by $\sqrt{|\boldsymbol{p}|^2 \cdot |\boldsymbol{r}|^2 - (\boldsymbol{p}\boldsymbol{r}^T)^2}$. The length of the last vector of $\boldsymbol{B}_0$ will be bounded from below according to Corollary 5.

**Example 3** Recall the decomposition of the $\boldsymbol{a}$-coefficients from Examples 1 and 2. Let $M = 12223$.

$$\begin{aligned} a_1 &= M + 0 \\ a_2 &= M + 1 \\ a_3 &= 3M + 3, \end{aligned}$$

so $\boldsymbol{p} = (1,1,3)^T$ and $\boldsymbol{r} = (0,1,3)^T$. The first column of $\boldsymbol{B}_0$, $(0,-3,1)^T$, is short. This vector is orthogonal to $\boldsymbol{a}$, $\boldsymbol{p}$, and $\boldsymbol{r}$. The second, and last, column of $\boldsymbol{B}_0$, $(12224, -1222, -3667)^T$, is long. □

To summarize, if the determinant of the lattice $L_0$ is large due to a large value of $M$, then this large value basically has to be contributed by the last vector $\boldsymbol{b}_0^{n-1}$ of $\boldsymbol{B}_0$. The long vector $\boldsymbol{b}_0^{n-1}$ implies a small value of the integral width of $Q$ in the unit direction $\boldsymbol{e}_{n-1}$, so only a few, in fact often zero or one, lattice hyperplanes intersect $Q$ in this direction for the instances we consider. In Example 1 we observed that $W_I(Q, \boldsymbol{e}_2) = -1$, which immediately gave us a certificate for infeasibility.

The argument regarding the length of the columns of $\boldsymbol{B}_0$ presented above also holds in the more general case that the $\boldsymbol{a}$-coefficients decompose as follows:

$$a_i = p_i M + r_i N, \qquad \text{for } i = 1, \ldots, n,$$

where $p_i, M, N \in \mathbb{Z}_{\geq 0}$, $r_i \in \mathbb{Z}$, and where $M$ and $N$ are assumed to be large compared to $p_i$ and $|r_i|$.

## 4 Computational Results

To illustrate our results we have solved various instances of type (2). The instances are given in Table 1. In the first column the instance name is given.

Table 1: The instances

| Instance | $\boldsymbol{a}$ | | | | | | | | | | Frobenius number |
|---|---|---|---|---|---|---|---|---|---|---|---|
| cuww1 | 12223 | 12224 | 36674 | 61119 | 85569 | | | | | | 89643481 |
| cuww2 | 12228 | 36679 | 36682 | 48908 | 61139 | 73365 | | | | | 89716838 |
| cuww3 | 12137 | 24269 | 36405 | 36407 | 48545 | 60683 | | | | | 58925134 |
| cuww4 | 13211 | 13212 | 39638 | 52844 | 66060 | 79268 | 92482 | | | | 104723595 |
| cuww5 | 13429 | 26850 | 26855 | 40280 | 40281 | 53711 | 53714 | 67141 | | | 45094583 |
| prob1 | 25067 | 49300 | 49717 | 62124 | 87608 | 88025 | 113673 | 119169 | | | 33367335 |
| prob2 | 11948 | 23330 | 30635 | 44197 | 92754 | 123389 | 136951 | 140745 | | | 14215206 |
| prob3 | 39559 | 61679 | 79625 | 99658 | 133404 | 137071 | 159757 | 173977 | | | 58424799 |
| prob4 | 48709 | 55893 | 62177 | 65919 | 86271 | 87692 | 102881 | 109765 | | | 60575665 |
| prob5 | 28637 | 48198 | 80330 | 91980 | 102221 | 135518 | 165564 | 176049 | | | 62442884 |
| prob6 | 20601 | 40429 | 40429 | 45415 | 53725 | 61919 | 64470 | 69340 | 78539 | 95043 | 22382774 |
| prob7 | 18902 | 26720 | 34538 | 34868 | 49201 | 49531 | 65167 | 66800 | 84069 | 137179 | 27267751 |
| prob8 | 17035 | 45529 | 48317 | 48506 | 86120 | 100178 | 112464 | 115819 | 125128 | 129688 | 21733990 |
| prob9 | 3719 | 20289 | 29067 | 60517 | 64354 | 65633 | 76969 | 102024 | 106036 | 119930 | 13385099 |
| prob10 | 45276 | 70778 | 86911 | 92634 | 97839 | 125941 | 134269 | 141033 | 147279 | 153525 | 106925261 |
| prob11 | 11615 | 27638 | 32124 | 48384 | 53542 | 56230 | 73104 | 73884 | 112951 | 130204 | 577134 |
| prob12 | 14770 | 32480 | 75923 | 86053 | 85747 | 91772 | 101240 | 115403 | 137390 | 147371 | 944183 |
| prob13 | 15167 | 28569 | 36170 | 55419 | 70945 | 74926 | 95821 | 109046 | 121581 | 137695 | 765260 |
| prob14 | 11828 | 14253 | 46209 | 52042 | 55987 | 72649 | 119704 | 129334 | 135589 | 138360 | 680230 |
| prob15 | 13128 | 37469 | 39391 | 41928 | 53433 | 59283 | 81669 | 95339 | 110593 | 131989 | 663281 |
| prob16 | 35113 | 36869 | 46647 | 53560 | 81518 | 85287 | 102780 | 115459 | 146791 | 147097 | 1109710 |
| prob17 | 14054 | 22184 | 29952 | 64696 | 92752 | 97364 | 118723 | 119355 | 122370 | 140050 | 752109 |
| prob18 | 20303 | 26239 | 33733 | 47223 | 55486 | 93776 | 119372 | 136158 | 136989 | 148851 | 783879 |
| prob19 | 20212 | 30662 | 31420 | 49259 | 49701 | 62688 | 74254 | 77244 | 139477 | 142101 | 677347 |
| prob20 | 32663 | 41286 | 44549 | 45674 | 95772 | 111887 | 117611 | 117763 | 141840 | 149740 | 1037608 |

Table 2: A value of $M$ for instances cuww1–5 yielding short $\boldsymbol{p}$ and $\boldsymbol{r}$

| | cuww1 | cuww2 | cuww3 | cuww4 | cuww5 |
|---|---|---|---|---|---|
| $M$ | 12223 | 12228 | 12137 | 13211 | 13429 |

Next, in column "$\boldsymbol{a}$", the $a_i$-coefficients are given, and in the last column the Frobenius number can be found. For all the instances we computed the Frobenius number using the algorithm described in Appendix 1. The instances can be divided into two groups. The first group contains instances `cuww1-cuww5` and `prob1-prob10`, and the second group consists of instances `prob11-prob20`. Instances `cuww1-cuww5` were generated by Cornuéjols, Urbaniak, Weismantel, and Wolsey (1997), and the remaining instances were generated for this study. For each of the instances `cuww1-cuww5` there is a decomposition $a_i = p_iM + r_i$ with short vectors $\boldsymbol{p}$ and $\boldsymbol{r}$. In Table 2 we give values of $M$ that yield short vectors $\boldsymbol{p}$ and $\boldsymbol{r}$ for these instances. Instances `prob1-prob10` were generated such that the $\boldsymbol{a}$-coefficients have a decomposition $a_i = p_iM + r_iN$ with short $\boldsymbol{p}$ and $\boldsymbol{r}$. We randomly generate $M$ from the uniform distribution $U[10000,\ 20000]$, $N$ from $U[1000,\ 2000]$, $p_i$ from $U[1,\ 10]$, and $r_i$ from $U[-10,\ 10]$.

In contrast, the second group of instances `prob11-prob20` were randomly generated such that the $\boldsymbol{a}$-coefficients are of the same size as in `prob1-prob10`, but they do not necessarily decompose with short vectors $\boldsymbol{p}$ and $\boldsymbol{r}$. We chose the same size of the $\boldsymbol{a}$-coefficients since this yields values of $d(L_0)$ of approximately the same size as for the instances `prob1-prob10`. For instances `prob11-prob20` coefficient $a_i$ is randomly generated from $U[10000, 150000]$.

The computational results of verifying infeasibility for the instances is reported on in Table 3. For each instance $\boldsymbol{a}$ we used the Frobenius number $F(a_1, \ldots, a_n)$ as the right-hand side coefficient $a_0$. For each of the instances we computed $d(L_0)$, the length of each of the basis vectors of the basis $\boldsymbol{B}_0$, and the number of lattice hyperplanes intersecting $Q$ in the coordinate directions $\boldsymbol{e}_1$ and $\boldsymbol{e}_{n-1}$. We then applied the integer branching algorithm described in Section 2 to $Q$. The number of nodes that were generated, and the computing time in seconds are given in the columns "# Search tree nodes" and "Time". Finally, we attempted to solve the instances, using the original formulation $P$, by standard linear programming based branch-and-bound using CPLEX version 6.5.3 . The number of nodes needed by branch-and-bound, and the computing time in seconds are reported on in the columns "# B&B nodes" and "B&B time". For the branch-and-bound algorithm we set the node limit to 50 million nodes. If an instance was not solved within this node limit, this is indicated by "$> 50 \times 10^6$" in the column "# B&B nodes". The time $t$ needed to evaluate the 50 million nodes is then indicated as "$> t$" in the column "B&B time". All the computations were carried out on a Sun Ultra 60 Model 2360 workstation with two UltraSPARC-II 359 MHz processors (our implementation is sequential) and 512 MB of memory.

We make the following observations. First, the Frobenius number of the instances `cuww1-cuww5` and `prob1-prob10` is about two orders of magnitude larger than the Frobenius number of instances `prob11-prob20` (see Table 1). Infeasible instances are typically harder to solve than feasible ones, and the larger the intersection points $a_0/a_i$ between the $n$-simplex $P$ and the coordinate axes, the harder the instance becomes for branch-and-bound. So, as a class, the first group of instances is harder for branch-and-bound. In Table 3 we can see that instances `cuww1-cuww5` and `prob1-prob10` are considerably harder to solve by branch-and-bound than instances `prob11-prob20`. The presolver of CPLEX was able to verify infeasibility for instances `cuww2` and `prob10`, but none of the other instances in the first group was solved within the node limit of 50 million nodes. All of the instances `prob11-prob20` were solved by branch-and-bound within half a million search nodes and one minute of computing time.

We also observe that the shape of the polytope $Q$ is very much influenced by the decomposition of the $\boldsymbol{a}$-coefficients. If the coefficients decompose with short vectors $\boldsymbol{p}$ and $\boldsymbol{r}$ relative to $M$, then the width of the corresponding polytope $Q$ in the unit direction $\boldsymbol{e}_{n-1}$ is very small. This made the instances trivial for our tree search algorithm applied to $Q$. All instances were solved using less than twenty search nodes and a fraction of a second computing time. For instances `prob11-prob20` where the $\boldsymbol{a}$-coefficients are generated randomly from a certain interval we observe that the width of $Q$ is of the same size in all unit directions, and in general greater than two. Our tree search algorithm applied to $Q$ therefore needed more nodes and longer computing times than for the first group of instances. Still, none of the instances `prob11-prob20` needed more than 126 nodes and about a tenth of a second computing time.

Table 3: Verification of infeasibility

| Instance | $d(L_0)$ | $\|b_i\|$ | | | | | | | | | $W_I(Q, e_1) + 1$ | $W_I(Q, e_{n-1}) + 1$ | # Search tree nodes | Time | # B&B nodes | B&B time |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| cuww1 | 112700.5 | | | | | | 2.0 | 3.5 | 3.5 | 4823.1 | 1862 | 0 | 1 | .001 | $> 50 \times 10^6$ | > 8139.3 |
| cuww2 | 119803.3 | | | | | 2.0 | 2.2 | 2.6 | 3.9 | 2922.9 | 1291 | 1 | 3 | .001 | 0* | 0.0 |
| cuww3 | 97088.2 | | | | | 2.0 | 2.4 | 2.8 | 4.0 | 2218.0 | 1155 | 2 | 3 | .001 | $> 50 \times 10^6$ | > 8079.9 |
| cuww4 | 154638.3 | | | | 1.7 | 2.4 | 2.4 | 4.0 | 3.0 | 2726.8 | 2429 | 1 | 2 | .001 | $> 50 \times 10^6$ | > 7797.5 |
| cuww5 | 123066.9 | | | 2.0 | 2.2 | 2.0 | 2.2 | 2.6 | 2.8 | 1711.4 | 1279 | 1 | 3 | .001 | $> 50 \times 10^6$ | > 6080.6 |
| prob1 | 227895.5 | | | 2.0 | 2.2 | 2.6 | 2.6 | 2.8 | 4.7 | 678.4 | 347 | 2 | 7 | .001 | $> 50 \times 10^6$ | > 7912.6 |
| prob2 | 256849.8 | | | 1.7 | 1.7 | 2.6 | 3.0 | 3.2 | 4.4 | 1016.0 | 274 | 2 | 7 | .001 | $> 50 \times 10^6$ | > 6529.2 |
| prob3 | 337663.2 | | | 2.2 | 2.4 | 3.0 | 3.0 | 3.3 | 3.6 | 988.4 | 466 | 2 | 11 | .002 | $> 50 \times 10^6$ | > 6872.1 |
| prob4 | 226877.3 | | | 2.6 | 2.4 | 2.6 | 2.4 | 3.6 | 3.5 | 1058.4 | 468 | 2 | 8 | .001 | $> 50 \times 10^6$ | > 8432.2 |
| prob5 | 324461.5 | | | 2.0 | 2.4 | 2.8 | 3.2 | 3.0 | 3.7 | 937.6 | 964 | 2 | 10 | .002 | $> 50 \times 10^6$ | > 8368.4 |
| prob6 | 191805.0 | 2.0 | 2.0 | 2.2 | 2.2 | 2.4 | 2.2 | 2.8 | 2.6 | 646.6 | 502 | 2 | 8 | .001 | $> 50 \times 10^6$ | > 5550.1 |
| prob7 | 207240.4 | 1.7 | 1.7 | 1.7 | 2.2 | 2.4 | 2.4 | 2.4 | 2.8 | 888.6 | 588 | 2 | 9 | .002 | $> 50 \times 10^6$ | > 5411.5 |
| prob8 | 288168.2 | 2.2 | 2.2 | 2.2 | 2.6 | 2.6 | 2.2 | 2.4 | 2.4 | 773.4 | 455 | 2 | 7 | .001 | $> 50 \times 10^6$ | > 5565.4 |
| prob9 | 235618.6 | 1.7 | 2.8 | 2.8 | 2.6 | 2.4 | 2.4 | 2.4 | 2.8 | 788.6 | 430 | 2 | 18 | .003 | $> 50 \times 10^6$ | > 6944.7 |
| prob10 | 363052.5 | 2.0 | 2.2 | 2.2 | 2.4 | 2.2 | 2.6 | 2.4 | 2.4 | 1165.2 | 880 | 2 | 10 | .002 | 0* | 0.0 |
| prob11 | 225420.4 | 3.6 | 4.0 | 4.5 | 4.4 | 4.6 | 4.2 | 4.7 | 5.2 | 6.1 | 4 | 5 | 37 | .005 | 88858 | 9.3 |
| prob12 | 307211.3 | 4.4 | 4.5 | 4.6 | 4.4 | 4.4 | 4.5 | 4.4 | 6.0 | 5.4 | 2 | 4 | 86 | .012 | 445282 | 51.0 |
| prob13 | 266246.9 | 4.6 | 4.2 | 4.6 | 4.0 | 4.8 | 4.6 | 5.3 | 5.1 | 5.8 | 6 | 6 | 41 | .006 | 580565 | 62.6 |
| prob14 | 286676.3 | 4.4 | 4.1 | 4.0 | 4.4 | 4.7 | 4.8 | 5.1 | 5.1 | 5.6 | 9 | 7 | 112 | .012 | 371424 | 43.4 |
| prob15 | 238047.7 | 3.6 | 4.5 | 4.1 | 3.9 | 3.9 | 5.1 | 4.8 | 5.5 | 6.0 | 3 | 3 | 66 | .080 | 426692 | 49.4 |
| prob16 | 297717.2 | 4.0 | 3.7 | 3.7 | 4.2 | 4.2 | 4.2 | 4.6 | 4.7 | 9.7 | 3 | 2 | 67 | .080 | 549483 | 61.4 |
| prob17 | 294591.6 | 4.6 | 4.4 | 4.2 | 4.6 | 4.6 | 5.1 | 4.0 | 4.2 | 5.7 | 2 | 4 | 126 | .150 | 218374 | 24.1 |
| prob18 | 300087.6 | 3.5 | 4.6 | 4.6 | 4.5 | 5.1 | 5.2 | 5.5 | 5.0 | 5.8 | 4 | 5 | 90 | .120 | 425727 | 46.9 |
| prob19 | 249577.9 | 3.9 | 3.7 | 4.1 | 5.1 | 5.2 | 5.6 | 4.8 | 5.5 | 4.6 | 11 | 6 | 78 | .100 | 255112 | 27.7 |
| prob20 | 314283.7 | 3.7 | 4.7 | 4.5 | 3.9 | 4.6 | 4.7 | 5.1 | 5.5 | 6.2 | 5 | 3 | 39 | .005 | 423608 | 46.1 |

*) CPLEX Presolve determines problem is infeasible or unbounded.

## Acknowledgments

We want to thank Hendrik Lenstra for his valuable suggestions and in particular for the outline of the proof of Theorem 1. We also wish to thank Bram Verweij for providing a framework code, based on his general enumeration library, for our integer branching algorithm.

The research of the first author was partially financed by the project TMR-DONET nr. ERB FMRX-CT98-0202 of the European Community.

## References

Aardal K., R. E. Bixby, C. A. J. Hurkens, A. K. Lenstra, J. W. Smeltink. 2000a. Market split and basis reduction: Towards a solution of the Cornuéjols-Dawande instances. *INFORMS Journal on Computing* **12** 192–202.

Aardal K., C. A. J. Hurkens, A. K. Lenstra. 2000b. Solving a system of diophantine equations with lower and upper bounds on the variables. *Mathematics of Operations Research* **25** 427–442.

Brauer A. 1942. On a problem of partitions. *American Journal of Mathematics* **64** 299–312.

Brauer A., J. E. Shockley 1962. On a problem of Frobenius. *Journal für Reine und Angewandte Mathematik* **211** 399–408.

Cassels, J. W. S. 1997. *An Introduction to the Geometry of Numbers.* Second Printing, Corrected. Reprint of the 1971 ed. Springer-Verlag, Berlin, Heidelberg.

Cook, W., T. Rutherford, H. E. Scarf, D. Shallcross. 1993. An implementation of the generalized basis reduction algorithm for integer programming. *ORSA Journal on Computing* **5** 206–212.

Cornuéjols G., M. Dawande. 1999. A class of hard small 0-1 programs. *INFORMS Journal on Computing* **11** 205–210.

Cornuéjols, G., R. Urbaniak, R. Weismantel, L. A. Wolsey. 1997. Decomposition of integer programs and of generating sets. R. E. Burkard, G. J. Woeginger, eds., *Algorithms – ESA '97.* Lecture Notes in Computer Science **1284**, Springer-Verlag, Berlin, Heidelberg, 92–103.

Erdős P., R. L. Graham. 1972. On a linear diophantine problem of Frobenius. *Acta Arithmetica* **21** 399–408.

Greenberg H. 1988. Solution to a linear Diophantine equation for nonnegative integers. *Journal of Algorithms* **9** 343–353.

Hungerford T. W. 1996. *Algebra*; corrected eighth printing. Springer-Verlag, New York, USA.

Kannan R. 1991. Lattice translates of a polytope and the Frobenius Problem. *Combinatorica* **12** 161–177.

Khinchine A. 1948. A quantitative formulation of Kronecker's theory of approximation (In Russian). *Izvestiya Akademii Nauk SSR Seriya Matematika* **12** 113–122.

Lenstra, A. K., H. W. Lenstra, Jr., L. Lovász. 1982. Factoring polynomials with rational coefficients. *Mathematische Annalen* **261** 515–534.

Lenstra, H. W., Jr. 1983. Integer programming with a fixed number of variables. *Mathematics of Operations Research* **8** 538–548.

Lovász, L., H. E. Scarf. 1992. The generalized basis reduction algorithm. *Mathematics of Operations Research* **17** 751–764.

Louveaux Q., L. A. Wolsey. 2000. Combining problem structure with basis reduction to solve a class of hard integer programs. CORE Discussion Paper 2000/51, CORE, Université Catholique de Louvain, Louvain-la-Neuve, Belgium (to appear in *Mathematics of Operations Research*).

Nguyen, P. Q., J. Stern. 2000. Lattice reduction in cryptology. W. Bosma, ed., *Algorithmic Number Theory: 4th International Symposium, ANTS-IV Leiden, The Netherlands, July 2-7, 2000, Proceedings.* Lecture Notes in Computer Science **1838**, Springer-Verlag, Berlin, Heidelberg, 85–112. An updated version can be found at URL:
`http://www.di.ens.fr/~pnguyen/pub.html`

Rödseth Ö. J. 1978. On a linear diophantine problem of Frobenius. *Journal für Reine und Angewandte Mathematik* **301** 171–178.

Schrijver A. 1986. *Theory of Linear and Integer Programming*, Wiley, Chichester, UK.

Selmer E. S. 1977. On the linear diophantine problem of Frobenius. *Journal* *für Reine und Angewandte Mathematik* **293/294** 1–17.

Selmer E. S., Ö. Beyer. 1978. On the linear diophantine problem of Frobenius in three variables. *Journal für Reine und Angewandte Mathematik* **301** 161–170.

Sylvester J. J., W.J. Curran Sharp. 1884. [Problem] 7382. *Mathematics from the Educational Times, with Additional Papers and Solutions* **41** 21.

## Appendix 1: Computing the Frobenius Number

Since the main aim of this paper is not to compute the Frobenius number — we use the Frobenius number to create infeasible instances — our approach is quite simple and based on a theorem by Brauer and Shockley (1962). Assume that $a_i$ is integer for $1 \leq i \leq n$, that $0 < a_1 \leq a_2 \leq \cdots \leq a_n$, and that $\gcd(a_1, \ldots, a_n) = 1$. Let $r_l$ be the smallest positive integer congruent to $(l \mod a_1)$ that can be expressed as a non-negative integer combination of $a_2, \ldots, a_n$. Each residue class modulo $a_1$ does contain numbers representable as $a_2x_2 + \cdots + a_nx_n$ with $x_i \in \mathbb{Z}_{\geq 0}$ for $1 \leq i \leq n$. Let $r = \max_{l \in \{1,2,\ldots,a_1-1\}} r_l$.

**Theorem 6** (Brauer and Shockley (1962).)

$$F(a_1, \ldots, a_n) = r - a_1 \ .$$

**Proof:** Suppose we can express $r - a_1$ as

$$r - a_1 = a_1x_1 + a_2x_2 + \cdots + a_nx_n \text{ with } x_i \in \mathbb{Z}_{\geq 0},\ 1 \leq i \leq n \, .$$

Then,

$$r - a_1(1 + x_1) = a_2x_2 + \cdots + a_nx_n \text{ with } x_i \in \mathbb{Z}_{\geq 0},\ 1 \leq i \leq n \, ,$$

which contradicts that $r$ is the smallest number in its residue class.

Next, take any integer number $N > r - a_1$ and assume that $N$ is not an integer multiple of $a_1$, in which case we are done. Assume that $N = (\ell \mod a_1)$ with $\ell \in \{1, \ldots, a_1 - 1\}$, so we can write $N = pa_1 + \ell$ for some $p \in \mathbb{Z}_{\geq 0}$. We know that $N$ is greater than or equal to the smallest number in its residue class that can be represented as $a_2x_2 + \cdots + a_nx_n$ with $x_i \in \mathbb{Z}_{\geq 0}$ for $1 \leq i \leq n$, i.e., $N \geq r_\ell = qa_1 + \ell$ for some $q \in \mathbb{Z}_{\geq 0}$. The following holds: $N - r_\ell = pa_1 + \ell - qa_1 - \ell = a_1(p - q)$, and since $N - r_\ell \geq 0$ we have $(p - q) \geq 0$. So, $N$ can be written as

$$N = a_1(p - q) + r_\ell = a_1(p - q) + a_2x_2 + \cdots + a_nx_n \, .$$

with $(p-q) \geq 0$ and $x_i \in \mathbb{Z}_{\geq 0}$ for $2 \leq i \leq n$. □

For each $l = 1, \ldots, a_1 - 1$ we compute the value of $r_l$ as:

$$r_l = \min\{\sum_{i=2}^{n} a_i x_i : \sum_{i=2}^{n} a_i x_i = l + a_1 x_1, \; \boldsymbol{x} \in \mathbb{Z}_{\geq 0}^{n}\} \; . \tag{7}$$

Since the instances of type (7) that we tackled are hard to solve by branch-and-bound we again applied the reformulation described in Section 2 to each subproblem and solved the reformulated subproblems by branch-and-bound. Notice that the reformulation only has to be determined for $l = 1$. The basis for $L = \{x \in \mathbb{Z}^n : -a_1 x_1 + \sum_{i=2}^{n} a_i x_i = 0\}$ is independent of $l$, and if we have computed $\boldsymbol{x}_f$ for $l = 1$, then $l\boldsymbol{x}_f$ can be used in the subsequent computations of subproblems $l = 2, \ldots, a_1 - 1$. Cornuéjols et al. (1997) used a formulation similar to (7) for computing the Frobenius number, but instead of using the reformulation described in Section 2 combined with branch-and-bound, they used test sets after having decomposed the $\boldsymbol{a}$-coefficients.

In Table 4 we give the computational results for the Frobenius number computations. In the two first columns the instance name and number of variables are given. Then, the computing time and the total number of branch-and-bound nodes needed for all $a_1 - 1$ subproblems are given. Since $a_1$ can vary quite a lot, we report on the average number of branch-and-bound nodes per subproblem in the last column.

Table 4: Results for the Frobenius number computations

| Instance | # Vars | Time | Total # B&B nodes | Ave. # nodes per subprob. |
|---|---:|---:|---:|---:|
| cuww1 | 5 | 50.0 | 11652 | 1.0 |
| cuww2 | 6 | 62.3 | 25739 | 2.1 |
| cuww3 | 6 | 64.6 | 39208 | 3.2 |
| cuww4 | 7 | 76.3 | 28980 | 2.2 |
| cuww5 | 8 | 130.2 | 210987 | 15.7 |
| prob1 | 8 | 891.3 | 3782264 | 150.9 |
| prob2 | 8 | 90.2 | 53910 | 4.5 |
| prob3 | 8 | 396.2 | 571199 | 14.4 |
| prob4 | 8 | 371.1 | 204191 | 4.2 |
| prob5 | 8 | 257.6 | 349320 | 12.2 |
| prob6 | 10 | 9057.3 | 39164012 | 1901.1 |
| prob7 | 10 | 200.7 | 93987 | 5.0 |
| prob8 | 10 | 304.8 | 577948 | 33.9 |
| prob9 | 10 | 162.6 | 91223 | 24.5 |
| prob10 | 10 | 586.8 | 445777 | 9.8 |
| prob11 | 10 | 241.3 | 577134 | 49.7 |
| prob12 | 10 | 515.8 | 1518531 | 102.8 |
| prob13 | 10 | 391.8 | 998415 | 65.8 |
| prob14 | 10 | 476.7 | 1551241 | 848.6 |
| prob15 | 10 | 418.0 | 1178543 | 89.8 |
| prob16 | 10 | 821.7 | 2063690 | 58.8 |
| prob17 | 10 | 385.4 | 1027115 | 73.1 |
| prob18 | 10 | 567.3 | 1494456 | 73.6 |
| prob19 | 10 | 499.0 | 1289971 | 63.8 |
| prob20 | 10 | 799.2 | 2070667 | 63.4 |