Solving a system of diophantine equations with lower and upper bounds on the variables

Karen Aardal* Cor Hurkens† Arjen K. Lenstra‡

Abstract

We develop an algorithm for solving a system of diophantine equations with lower and upper bounds on the variables. The algorithm is based on lattice basis reduction. It first finds a short vector satisfying the system of diophantine equations, and a set of vectors belonging to the null-space of the constraint matrix. Due to basis reduction, all these vectors are relatively short. The next step is to branch on linear combinations of the null-space vectors, which either yields a vector that satisfies the bound constraints or provides a proof that no such vector exists. The research was motivated by the need for solving constrained diophantine equations as subproblems when designing integrated circuits for video signal processing. Our algorithm is tested with good results on real-life data, and on instances from the literature.

Subject classification: Primary: 90C10. Secondary: 45F05, 11Y50.

1 Introduction and problem description

We develop an algorithm for solving the following integer feasibility problem:

$$\text{does there exist a vector } \mathbf{x} \in \mathbb{Z}^n \text{ such that } \mathbf{A}\mathbf{x} = \mathbf{d}, \quad \mathbf{0} \leq \mathbf{x} \leq \mathbf{u}? \tag{1}$$

We assume that $\mathbf{A}$ is an integral $m \times n$ matrix, where $m \leq n$, $\mathbf{d}$ an integral m-vector, and $\mathbf{u}$ an integral n-vector. This is an NP-complete problem; in the absence of bound constraints, it can be solved in polynomial time.

The research was motivated by a need for solving such problems when designing integrated circuits (ICs) for video signal processing, but the problem is

*`aardal@cs.uu.nl`. Department of Computer Science, Utrecht University. Research partially supported by the ESPRIT Long Term Research Project nr. 20244 (Project ALCOM-IT: *Algorithms and Complexity in Information Technology*), by the project TMR-DONET nr. ERB FMRX-CT98-0202, both of the European Community, by NSF grant CCR-9307391 through David B. Shmoys, Cornell University, and by NSF through the Center for Research on Parallel Computation, Rice University, under Cooperative Agreement No. CCR-9120008.

†`wscor@win.tue.nl`. Department of Mathematics and Computing Science, Eindhoven University of Technology. Research partially supported by the project TMR-DONET nr. ERB FMRX-CT98-0202 of the European Community.

‡`arjen.lenstra@citicorp.com`. Emerging Technology, Citibank N.A. Research partially supported by ESPRIT Long Term Research Project No. 20244 (Project ALCOM-IT: *Algorithms and Complexity in Information Technology*) of the European Community.

central in discrete optimization and linear algebra, and has several interpretations and applications. Examples are the *Frobenius* problem that was recently considered by Cornuéjols, Urbaniak, Weismantel and Wolsey [5], and the "*market split problem*" considered by Cornuéjols and Dawande [4].

The main ingredients of our algorithm are as follows. First we choose a lattice that seems particularly useful for our problem. We write down an initial basis that spans the given lattice and apply Lovász' basis reduction algorithm, as described in the paper by Lenstra, Lenstra and Lovász [11], to the initial basis. We will refer to this algorithm as "the basis reduction algorithm". The parameters of the initial basis are chosen such that the reduced basis contains one vector $\mathbf{x}_d$ satisfying $\mathbf{A}\mathbf{x}_d = \mathbf{d}$, and $n - m$ linearly independent vectors satisfying $\mathbf{A}\mathbf{x} = \mathbf{0}$. If the vector $\mathbf{x}_d$ satisfies the bounds, then we are done, and if not, we observe that $\mathbf{A}(\mathbf{x}_d + \lambda\mathbf{x}) = \mathbf{d}$ for any integer multiplier λ and any vector $\mathbf{x}$ such that $\mathbf{A}\mathbf{x} = \mathbf{0}$. Hence, we can branch on integer linear combinations of vectors satisfying $\mathbf{A}\mathbf{x} = \mathbf{0}$ in order to obtain a vector satisfying the diophantine equations as well as the lower and upper bounds, or a proof that no such vector exists.

The test instances from the IC-design were difficult to tackle by linear programming (LP) based branch-and-bound due to the characteristics of the input. In order to explain the structure of these instances we briefly explain the origin of the problem in Section 2. There, we also describe the Frobenius problem and the market split problem. In Section 3 we give a short description of the basis reduction algorithm and a brief review of the use of basis reduction in integer programming. In Section 4 we introduce a suitable lattice for our problem (1), and provide an initial basis spanning that lattice. We also derive structural properties of the reduced basis. Our algorithm is outlined in Section 5. We tested the algorithm on the IC design problem, the Frobenius instances of Cornuéjols et al. [5], and on some market split instances as described by Cornuéjols and Dawande [4]. Our computational experience is presented in Section 6.

2 Background and Applications

In one of the steps of the design of ICs for video signal processing one needs to assign so-called *data streams* to processors. A data stream is a repetitive set of arithmetic operations. The attributes of a data stream are the starting time of the first execution, the number of repetitions expressed as a so-called iterator vector, and the period of repetition. One can view a data stream as a set of nested loops. The outer loop has an iterator $i_0 : 0 \leq i_0 \leq I_0$. The following loop has iterator $i_1 : 0 \leq i_1 \leq I_1$, and so forth. The periodicity corresponding to a loop is the time interval between the start of two consecutive iterations of that loop.

Example 1 Consider the following data stream in "loop form".

for $i_0 = 0$ **to** 1 $\{10\}$
  for $i_1 = 0$ **to** 2 $\{2\}$
    do <expression>;

The numbers within the braces indicate the period of the loops. We assume that the execution of the expression takes at most one unit of time. The corresponding data stream has a period vector $\mathbf{p} = (p_0, p_1)^T = (10, 2)^T$, and an iterator vector $\mathbf{i} = (i_0, i_1)^T$. The bounds on the iterator vector are:

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \leq \begin{pmatrix} i_0 \\ i_1 \end{pmatrix} \leq \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

The data stream on a time scale is illustrated in Figure 1. Here, we assume that the starting time $s = 5$.

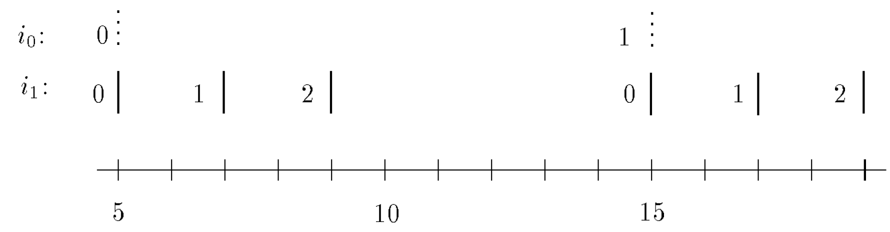

Figure 1: A DATA STREAM.

The dotted lines indicate the time at which the outer loop iterates, and the solid lines when the inner loop iterates, and hence the time at which the execution of the expression begins. ■

An important difference between a loop in a computer program and a data stream is that, in the latter, the outer loop does not necessarily start a new iteration immediately after the inner loop has finished. The advantage with a stream is that one can compute the precise time at which a certain iteration begins. When constructing an assignment of multiple streams to one or more processors the following *conflict detection problem* occurs: check whether there is any point in time at which operations of two different streams are carried out simultaneously. If such a point in time exists, then the streams should not be assigned to the same processor. Consider an arbitrary data stream f. Let $\mathbf{i}_f = (i_{f0}, i_{f1}, \ldots, i_{fk})^T$ be the *iterator vector* of the stream. The iterator vector lies between upper and lower bounds, $\mathbf{0} \leq \mathbf{i}_f \leq \mathbf{I}_f$. Let $\mathbf{p}_f$ denote the *period vector* and s_f the *starting time* of the stream. The point in time at which execution $\mathbf{i}_f$ of data stream f begins is expressed as $t(\mathbf{i}_f) = s_f + \mathbf{p}_f^T \mathbf{i}_f$. The conflict detection problem can be formulated mathematically as the following integer feasibility problem: Given data streams f and g, do there exist iterator vectors $\mathbf{i}_f$ and $\mathbf{i}_g$ such that

$$s_f + \mathbf{p}_f^T \mathbf{i}_f = s_g + \mathbf{p}_g^T \mathbf{i}_g, \text{ and such that } \mathbf{0} \leq \mathbf{i}_f \leq \mathbf{I}_f, \ \ \mathbf{0} \leq \mathbf{i}_g \leq \mathbf{I}_g?$$

This problem is a special case of problem (1) with $m = 1$.

The Frobenius problem is defined as follows: given nonnegative integers $(a_1, \ldots, a_n)$ with $\gcd(a_1, \ldots, a_n) = 1$, find the largest integer d that cannot be expressed as a nonnegative integer combination of $a_1, \ldots, a_n$. The number d is called the Frobenius number. Here, the values of the variables x_j indicate the nonnegative integer combination. In this case the upper bounds on the variables are equal to infinity. The instances considered by Cornuéjols et al. [5] were also hard to solve using LP-based branch-and-bound. They developed a test set approach that was successful on their instances.

In the market split problem studied by Cornuéjols and Dawande [4], a company with two divisions supplies retailers with several products. The goal is to allocate each retailer to one of the divisions such that division one controls $100c_i\%$, $0 \le c_i \le 1$, of the market for product i, and division two controls $(100-100c_i)\%$. Here, n is the number of retailers and m the number of products. The coefficient a_{ij} is the demand of retailer j for product i, and the right-hand side coefficient d_i is determined as $\lfloor c_i d'_i \rfloor$, where d'_i is the total amount of product i that is supplied to the retailers. The decision variable x_j takes value 1 if retailer j is allocated to division one and 0 otherwise. The question is: "does there exist an allocation of the retailers to the divisions such that the desired market split is obtained?" In this version of problem (1) the upper bounds on the variables are all equal to 1. Cornuéjols and Dawande generated instances of the market split problem that are provably hard for LP-based branch-and-bound. It is worth noting that there is an optimization version of this problem as well. If a market split such as described above is not possible, then one would like to find an assignment of the retailers to the divisions such that the deviation from the desired split is minimized. Here, we only deal with the feasibility version.

When solving a feasibility problem such as (1) by LP-based branch-and-bound, in particular two difficulties often seem to arise. First, the search tree may become large. The larger the upper bounds on the variables are, the worse the growth in the tree size typically becomes. Second, round-off errors may occur if the difference in the sizes of the input numbers is large. The size of the branch-and-bound tree may also be sensitive to the objective function that is used. For our problem (1) an objective function does not have any meaning since it is a feasibility problem; as long as we either find a feasible vector, or are able to verify that no feasible vector exists, the objective function as such does not matter. The problem is that one objective function may give an answer faster than another, but which one is best is hard to predict. An objective function also introduces an aspect to the problem that is not natural. Round-off errors occur quite frequently for the instances related to the conflict detection problem, since the coefficients of some of the variables are very large ($\approx 10^7$). The special characteristics of these instances – some very large and some relatively small coefficients and a very large right-hand side coefficient d – are due to the difference in periodicity of the nested loops. This difference is explained by the composition of a television screen image. The time between two pictures is 40 ms, and the time between two lines and between two pixels are 64 μs and 74 ns respectively. Since the output rate of the signals has to be equal to the input rate, we get large differences in periodicity when the data stream corresponds to operations that have to be repeated for all screens, lines and pixels. Due to the large difference in the magnitude of the coefficients we often observe that the LP-based branch-and-bound algorithm terminates with a solution in which for instance variable x_j takes value 4.999999, simply because the standard implementations do not allow for greater precision. If one would round x_j to $x_j = 5.0$, then one would obtain a vector $\mathbf{x}$ such that $\mathbf{ax} \neq d$. It is obviously a serious drawback that the algorithm terminates with an unsubstantiated claim that a solution exists.

To overcome the mentioned deficiencies we have developed an algorithm based on the basis reduction algorithm. The motivation behind choosing basis reduction as a core of our algorithm is twofold. First, we work directly with arbitrarily sized integers, which avoids the round-off problems. Second, basis reduction finds short, nearly orthogonal vectors belonging to the lattice described by the given basis. In our case we obtain a short vector $\mathbf{x}_d$ satisfying $\mathbf{A}\mathbf{x}_d = \mathbf{d}$ and $m - n$ vectors satisfying $\mathbf{Ax} = \mathbf{0}$. Given the lower and upper bounds on the variables, we can interpret problem (1) as checking whether there exists a *short vector* satisfying a given set of diophantine equations. Furthermore, our algorithm is designed for feasibility problems. A useful by-product of our algorithm is that it yields an alternative formulation of problem (1) in terms of a full-dimensional polytope. Such a description could also be obtained by for instance deriving the Hermite normal form of the matrix $\mathbf{A}$, but that typically creates large (but polynomially bounded) numbers. A full-dimensional polytope is useful if one wants to apply the algorithm of Lenstra [12], or of Lovász and Scarf [14].

3 Basis reduction and its use in integer programming

We begin by giving the definition of a lattice and a reduced basis.

Definition 1 *A subset $L \subset \mathbb{R}^n$ is called a* lattice *if there exists a basis $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_l$ of $\mathbb{R}^n$ such that*

$$L = \{\sum_{j=1}^{l} \alpha_j \mathbf{b}_j : \ \alpha_j \in \mathbb{Z}, \ 1 \leq j \leq l\}. \tag{2}$$

Gram-Schmidt orthogonalization is a transformation that derives orthogonal vectors $\mathbf{b}_j^*$, $1 \leq j \leq l$ from independent vectors $\mathbf{b}_j$, $1 \leq j \leq l$. The vectors $\mathbf{b}_j^*$, $1 \leq j \leq l$ and the real numbers μ_{jk}, $1 \leq k < j \leq l$ are determined from $\mathbf{b}_j$, $1 \leq j \leq l$ by the recurrence

$$\mathbf{b}_j^* = \mathbf{b}_j - \sum_{k=1}^{j-1} \mu_{jk} \mathbf{b}_k^* \tag{3}$$

$$\mu_{jk} = (\mathbf{b}_j)^T \mathbf{b}_k^* / (\mathbf{b}_k^*)^T \mathbf{b}_k^*. \tag{4}$$

Let $|| \ ||$ denote the Euclidean length in $\mathbb{R}^n$. Lenstra, Lenstra and Lovász [11] used the following definition of a reduced basis:

Definition 2 *A basis* $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_l$ *is called* reduced *if*

$$|\mu_{jk}| \leq \frac{1}{2} \text{ for } 1 \leq k < j \leq l \tag{5}$$

and

$$||\mathbf{b}_j^* + \mu_{j,j-1}\mathbf{b}_{j-1}^*||^2 \geq \frac{3}{4}||\mathbf{b}_{j-1}^*||^2 \text{ for } 1 < j \leq l. \tag{6}$$

The vector $\mathbf{b}_j^*$ is the projection of $\mathbf{b}_j$ on the orthogonal complement of $\sum_{k=1}^{j-1} \mathbb{R}\mathbf{b}_k$, and the vectors $\mathbf{b}_j^* + \mu_{j,j-1}\mathbf{b}_{j-1}^*$ and $\mathbf{b}_{j-1}^*$ are the projections of $\mathbf{b}_j$ and $\mathbf{b}_{j-1}$ on the orthogonal complement of $\sum_{k=1}^{j-2} \mathbb{R}\mathbf{b}_k$. The purpose with the reduced basis is to have short basis vectors that are nearly orthogonal. Moreover, the first basis vector should be an approximation of the shortest vector in the lattice. To ascertain whether these goals have been reached, the Gram-Schmidt vectors are used as a reference. Note that the Gram-Schmidt vectors do not necessarily belong to the lattice. In this light we shall interpret conditions (5) and (6). Inequality (5) states that the vectors $\mathbf{b}_j$, $1 \leq j \leq l$ are nearly orthogonal in the following sense. If μ_{jk} is small, then either the vectors $\mathbf{b}_j$ and $\mathbf{b}_k^*$ are almost orthogonal, or $\mathbf{b}_j$ is relatively short compared to $\mathbf{b}_k^*$. Consider the case where $k = j-1$, and suppose that $\mathbf{b}_j$ is short compared to $\mathbf{b}_{j-1}^*$, which implies that $\mathbf{b}_j^*$ is short compared to $\mathbf{b}_{j-1}^*$ as $||\mathbf{b}_j^*|| \leq ||\mathbf{b}_j||$. Assume we interchange $\mathbf{b}_j$ and $\mathbf{b}_{j-1}$. Then the new $\mathbf{b}_{j-1}^*$ will be the vector $\mathbf{b}_j^* + \mu_{j,j-1}\mathbf{b}_{j-1}^*$, which will be short compared to the old $\mathbf{b}_{j-1}^*$, i.e., condition (6) will be violated. This implies that the order of the basis vectors $\mathbf{b}_1, \ldots, \mathbf{b}_j$ is wrong with respect to the criterion that we want the first basis vector of the reduced basis to be an approximation of the shortest vector in the lattice. The constant $\frac{3}{4}$ in inequality (6) is arbitrarily chosen and can be replaced by any fixed real number $\frac{1}{4} < y < 1$. Lovász' basis reduction algorithm [11] is a polynomial time algorithm that finds a reduced basis for a lattice given an initial basis. The algorithm consists of a sequence of size reductions and interchanges as described below. For the precise algorithm we refer to [11].

Size reduction: If for any pair of indices $j, k:\ 1 \leq k < j \leq l$ condition (5) is violated, then replace $\mathbf{b}_j$ by $\mathbf{b}_j - \lceil \mu_{jk} \rfloor \mathbf{b}_k$, where $\lceil \mu_{jk} \rfloor = \lceil \mu_{jk} - \frac{1}{2} \rceil$.

Interchange: If condition (6) is violated for an index j, $1 < j \leq l$, then interchange vectors $\mathbf{b}_{j-1}$ and $\mathbf{b}_j$.

Basis reduction was introduced in integer programming by H.W. Lenstra, Jr. [12], who showed that the problem of determining if there exists a vector $\mathbf{x} \in \mathbb{Z}^n$ such that $\mathbf{A}\mathbf{x} \leq \mathbf{d}$ can be solved in polynomial time when n is fixed. Before this result was published, only the cases $n = 1, 2$ were known to be polynomially solvable. The idea behind Lenstra's algorithm can be explained considering a two-dimensional polytope. Suppose that this polytope is "thin" as illustrated in Figure 2. If it extends arbitrarily far in both directions, as indicated in the figure, then an LP-based branch-and-bound tree will become arbitrarily deep before concluding that no feasible solution exists. It is easy to construct a similar example in which a feasible vector does exist. So, even if $n = 2$, an LP-based branch-and-bound algorithm may require arbitrarily many iterations. What Lenstra observed was the following. Assume that we start

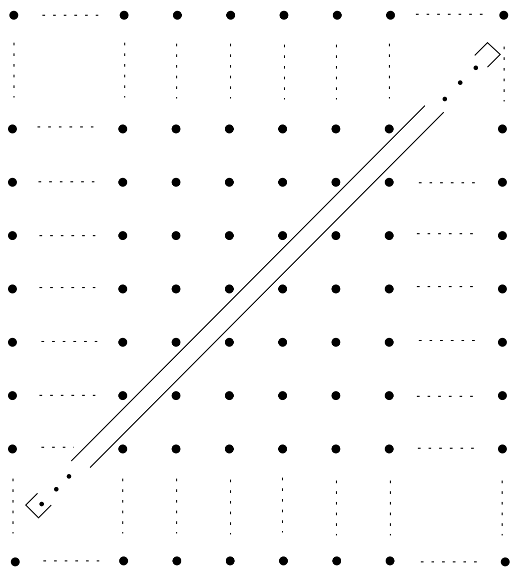

Figure 2: A THIN POLYTOPE IN $\mathbb{Z}^2$.

with the full-dimensional polytope $X \in \mathbb{R}^n$ and that we consider the lattice $\mathbb{Z}^n$. The problem is to determine whether there exists a vector $\mathbf{x} \in (X \cap \mathbb{Z}^n)$. We refer to this problem as problem P. We use $\mathbf{b}_j = \mathbf{e}_j$, $1 \le j \le n$ as a basis for the lattice $\mathbb{Z}^n$, where $\mathbf{e}_j$ is the jth column of the $n \times n$ identity matrix. To avoid working with a thin polytope we apply a linear transformation τ to X to make it appear "regular". Problem P is equivalent to the problem of determining whether there exists a vector $\mathbf{x} \in (\tau X \cap \tau\mathbb{Z}^n)$. The new polytope τX has a regular shape but the basis vectors $\tau\mathbf{e}_j$ are not necessarily orthogonal any longer, so from the point of view of branching the difficulty is still present. We can view this as having shifted the problem we had from the polytope to the lattice. This is where basis reduction proves useful. By applying the basis reduction algorithm to the basis vectors $\tau\mathbf{e}_j$, we obtain a new basis $\hat{\mathbf{b}}_1, \ldots, \hat{\mathbf{b}}_n$ spanning the same lattice, $\tau\mathbb{Z}^n$, but having short, nearly-orthogonal vectors. In particular it is possible to show that the distance between any two consecutive hyperplanes $H + k\hat{\mathbf{b}}_n$, $H + (k+1)\hat{\mathbf{b}}_n$, where $H = \sum_{j=1}^{n-1} \mathbb{R}\mathbf{b}_j$ and $k \in \mathbb{Z}$, is not too short, which means that if we branch on these hyperplanes, then there cannot be too many of them. Each branch at a certain level of the search tree corresponds to a subproblem with dimension one less than the dimension of its predecessor. In Figure 3 we show how the distance between hyperplanes $H + k\hat{\mathbf{b}}_n$ increases if we use a basis with nearly orthogonal vectors instead of a basis with non-orthogonal ones.

For the integer programming problem P, Lovász and Scarf [14] developed an algorithm that, as Lenstra's algorithm, uses branching on hyperplanes. Instead of using a transformation τ to transform the polytope and the initial basis vectors, and then applying a basis reduction algorithm, their algorithm produces a "Lovász-Scarf-reduced" basis by measuring the width of the considered

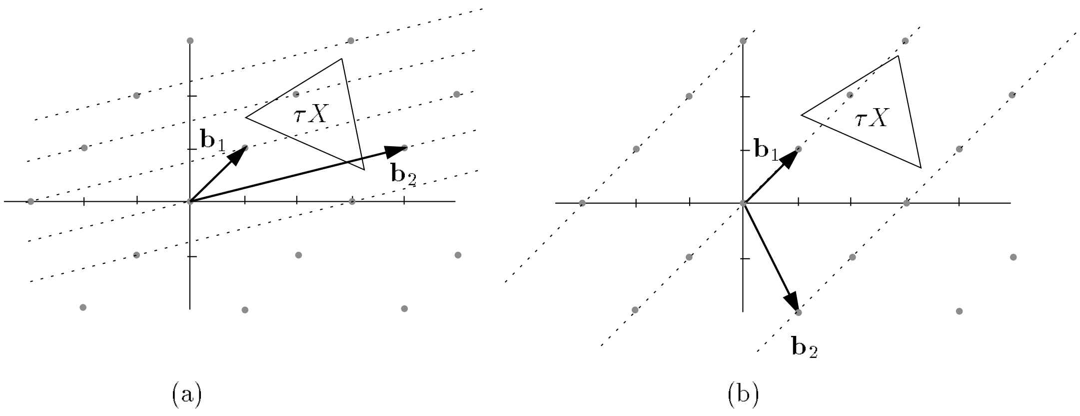

Figure 3: (a) Non-orthogonal basis. (b) Nearly orthogonal basis.

polytope in different independent directions. Lovász and Scarf's definition of a reduced basis is a generalization of the definition given by Lenstra et al. [11]. Cook, Rutherford, Scarf and Shallcross [2] report on a successful implementation of the Lovász-Scarf algorithm. Cook et al. were able to solve some integer programming problems arising in network design that could not be solved by traditional LP-based branch-and-bound.

Basis reduction has also been used successfully when trying to find solutions to subset sum problems that arise in cryptography. This is another application of problem (1) in which the matrix $\mathbf{A}$ consists of one row only. Here, the message that the "sender" wants to transmit to the "receiver" is represented by a sequence $\mathbf{x} \in \{0,1\}^n$ of "bits". The receiver knows a sequence of numbers $a_1, \ldots, a_n$, and instead of sending the actual message $\mathbf{x}$, the sender sends a number $d = \mathbf{ax}$. Once the receiver knows d he can recover the message $\mathbf{x}$ by solving the subset sum problem $\mathbf{ax} = d$. Here, the equation $\mathbf{ax} = d$ is known to have a solution, and the receiver knows certain secret information, called a *trapdoor*, about the a_j-coefficients. Lagarias and Odlyzko [10] considered the lattice in $\mathbb{R}^{n+1}$ spanned by the following basis:

$$\mathbf{B} = \begin{pmatrix} \mathbf{I}^{(n)} & \mathbf{0}^{(n\times 1)} \\ -\mathbf{a} & d \end{pmatrix}$$

Here, $\mathbf{I}^{(n)}$ denotes the n-dimensional identity matrix, and $\mathbf{0}^{(p\times q)}$ denotes the $p \times q$-matrix consisting of zeros only. They observed that the vector $\mathbf{x}$ satisfies $\mathbf{ax} = d$, if and only if the vector $(\mathbf{x}^T, 0)^T = \mathbf{B}(\mathbf{x}^T, 1)^T$ belongs to the lattice.

Lagarias and Odlyzko applied the basis reduction algorithm to the basis $\mathbf{B}$ and checked if any of the reduced basis vectors were of the form $(\mathbf{x}^T, 0)^T$, with $\mathbf{x} \in \{0,1\}^n$. If such a vector is found then $\mathbf{x}$ solves $\mathbf{ax} = d$. There is no guarantee that the algorithm finds a feasible vector $\mathbf{x}$, but the authors show that for "almost all" instances that satisfy $n/\log_2(\max_j\ a_j) < 0.645$, a feasible vector is found. Several similar bases have been considered later as input to algorithms for trying to find solutions to subset sum problems. Schnorr and

Euchner [15], for instance, used the basis

$$\mathbf{B} = \begin{pmatrix} \mathrm{diag}(2)^{(n\times n)} & \mathbf{1}^{(n\times 1)} \\ n\mathbf{a} & nd \\ \mathbf{0}^{(1\times n)} & 1 \end{pmatrix}$$

where $\mathrm{diag}(2)^{(n\times n)}$ is the $n \times n$-matrix with twos along the main diagonal and zeros otherwise. Here, a lattice vector $\mathbf{v} \in \mathbb{Z}^{n+2}$ that satisfies $|v_{n+2}| = 1$, $v_{n+1} = 0$ and $v_j \in \{\pm 1\}$ for $0 \le j \le n$, corresponds to a feasible vector $x_j = \frac{1}{2}|v_j - v_{n+2}|$, $0 \le j \le n$. Schnorr and Euchner [15] proposed an algorithm based on basis reduction that for "almost all" subset sum problem instances with $n/\log_2(\max_j\ a_j) < 0.9408$ finds a feasible vector. The algorithm uses the above basis as input. For further details on finding feasible solutions to subset sum problems arising in cryptography we refer to the above references and to the papers [6], [16], and [9]. A recent application in cryptography is due to Coppersmith [3] who uses basis reduction to find small integer solutions to a polynomial in a single variable modulo N, and to a polynomial in two variables over the integers. This has applications to some RSA-based cryptographic schemes.

Integer programming and cryptography are not the only application of lattice basis reduction. A prominent application is factoring polynomials with rational coefficients. Lenstra et al. [11] developed a polynomial-time algorithm based on basis reduction for finding a decomposition into irreducible factors of a non-zero polynomial in one variable with rational coefficients. In extended g.c.d.-computations, basis reduction is used by for instance Havas, Majewski and Matthews [8]. Here, the aim is to find a short multiplier vector $\mathbf{x}$ such that $\mathbf{ax} = d$, where $d = \gcd(a_1, a_2, \ldots, a_n)$.

In our algorithm we use a basis that is similar to the bases used by Lagarias and Odlyzko [10], and by Schnorr and Euchner [15]. The important differences between the approach described above and our approach are the following. First the question that is posed is different. We do not know a priori whether our instances have a feasible solution or not, and we want to *solve* the feasibility problem, i.e., if a feasible solution exists we want to find one, and if no feasible solution exists this should be verified. Hence, we propose a branching algorithm as described in Section 5. We also use two large constants N_1 and N_2 to "force" the basis reduction algorithm to find interesting vectors as described in section 4. Moreover, our algorithm can handle systems of linear diophantine equations.

4 Structure of initial and reduced basis

Here we consider a lattice that is suitable for our problem (1):

$$\text{does there exist a vector } \mathbf{x} \in \mathbb{Z}^n \text{ such that } \mathbf{Ax} = \mathbf{d}, \quad \mathbf{0} \le \mathbf{x} \le \mathbf{u}?$$

The ith row of the matrix $\mathbf{A}$ is denoted by $\mathbf{a}_i$. Without loss of generality we assume that $\gcd(a_{i1}, a_{i2}, ..., a_{in}) = 1$ for $1 \le i \le m$, and that $\mathbf{A}$ has full row rank. We formulate an initial basis $\mathbf{B}$ that generates this lattice and derive structural properties of the reduced basis $\hat{\mathbf{B}}$ obtained after applying the basis reduction algorithm to $\mathbf{B}$.

The lattice $L \subset \mathbb{R}^{n+m+1}$ we consider contains vectors

$$(x_1, \ldots, x_n,\ N_1 y, N_2(\mathbf{a}_1\mathbf{x} - d_1 y), \ldots, N_2(\mathbf{a}_m\mathbf{x} - d_m y))^T, \tag{7}$$

where y is a variable associated with the right-hand-side vector $\mathbf{d}$, and N_1 and N_2 are integer numbers. The basis $\mathbf{B}$ given below spans the lattice L.

$$\mathbf{B} = \begin{pmatrix} \mathbf{I}^{(n)} & \mathbf{0}^{(n\times 1)} \\ \mathbf{0}^{(1\times n)} & N_1 \\ N_2\mathbf{A} & -N_2\mathbf{d} \end{pmatrix} \tag{8}$$

The basis vectors are given columnwise, and the basis consists of $n+1$ vectors $\mathbf{b}_j = (b_{1j}, \ldots, b_{n+m+1,j})^T$, $1 \leq j \leq n+1$. In the first draft of our paper [1] we used one variable y_i for each right-hand side coefficient d_i. Laurence Wolsey [18] observed that it is sufficient to add one variable y for the entire vector $\mathbf{d}$. We are indebted to him for allowing us to include this improvement here.

Proposition 1 *The integer vector $\mathbf{x}_d$ satisfies $\mathbf{A}\mathbf{x}_d = \mathbf{d}$ if and only if the vector*

$$\begin{pmatrix} \mathbf{x}_d \\ N_1 \\ \mathbf{0}^{(m\times 1)} \end{pmatrix} = \mathbf{B}\begin{pmatrix} \mathbf{x}_d \\ 1 \end{pmatrix} \tag{9}$$

belongs to the lattice L, and the integer vector $\mathbf{x}_0$ satisfies $\mathbf{A}\mathbf{x}_0 = \mathbf{0}$ if and only if the vector

$$\begin{pmatrix} \mathbf{x}_0 \\ 0 \\ \mathbf{0}^{(m\times 1)} \end{pmatrix} = \mathbf{B}\begin{pmatrix} \mathbf{x}_0 \\ 0 \end{pmatrix} \tag{10}$$

belongs to the lattice L.

As we will show in Theorem 4, if there exists an integer vector that is a solution to the system $\mathbf{A}\mathbf{x} = \mathbf{d}$, and if the numbers N_1 and N_2 are chosen large enough, then the reduced basis will contain one vector of the form $(\mathbf{x}_d^T, N_1, \mathbf{0}^{(1\times m)})^T$, and $n-m$ vectors of the form $(\mathbf{x}_0^T, 0, \mathbf{0}^{(1\times m)})^T$. Due to Proposition 1, we can conclude that the vector $\mathbf{x}_d$ satisfies the system $\mathbf{A}\mathbf{x}_d = \mathbf{d}$. Since basis reduction finds short vectors belonging to a given lattice, we may hope that the vector $\mathbf{x}_d$ is short in the sense that it satisfies the bounds. If not, we will show in Section 5 how we can add integer linear combinations of the vectors satisfying $\mathbf{A}\mathbf{x} = \mathbf{0}$ to $\mathbf{x}$ in order to obtain a vector that satisfies the bounds as well if such a vector exists.

Lemma 2 (Lenstra, Lenstra, Lovász [11]) *Let $\Lambda \subset R^n$ be a lattice with reduced basis $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n \in \mathbb{R}^n$. Let $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_t \in \Lambda$ be linearly independent. Then we have*

$$||\mathbf{b}_j||^2 \leq 2^{n-1}\max\{||\mathbf{y}_1||^2, ||\mathbf{y}_2||^2, ..., ||\mathbf{y}_t||^2\} \textit{ for } 1 \leq j \leq t. \tag{11}$$

Definition 3 *A matrix of full row rank is said to be in* Hermite normal form *if it has the form* $(\mathbf{D},\ \mathbf{0})$, *where* $\mathbf{D}$ *is a nonsingular, lower triangular, nonnegative matrix, in which each row has a unique maximum entry, which is located on the main diagonal of* $\mathbf{D}$.

Lemma 3 *Let* $\mathbf{U}$ *be the* $n \times n$ *unimodular matrix such that* $\mathbf{AU} = (\mathbf{D}^{(m\times m)},\ \mathbf{0}^{(m\times(n-m))})$ *is the Hermite normal form of* $\mathbf{A}$. *Consider the following* $n-m+1$ *vectors:*

$$(\mathbf{x}_d, \mathbf{x}_0^1, \ldots, \mathbf{x}_0^{n-m}) = \mathbf{U}\begin{pmatrix} (\mathbf{D}^{-1}\mathbf{d})^{(m\times 1)} & \mathbf{0}^{(m\times(n-m))} \\ \mathbf{0}^{((n-m)\times 1)} & \mathbf{I}^{(n-m)} \end{pmatrix}. \tag{12}$$

The vector $\mathbf{x}_d$ *satisfies* $\mathbf{Ax}_d = \mathbf{d}$ *and the vectors* $\mathbf{x}_0^j,\ 1 \le j \le n-m$ *are linearly independent and satisfy* $\mathbf{Ax}_0^j = \mathbf{0}$. *The sizes of the vectors* $\mathbf{x}_d, \mathbf{x}_0^1, \ldots, \mathbf{x}_0^j$ *are polynomially bounded in the sizes of* $\mathbf{A}$ *and* $\mathbf{d}$.

Proof: Evaluating $\mathbf{Ax}_d$ yields

$$\mathbf{Ax}_d = \mathbf{AU}\begin{pmatrix} (\mathbf{D}^{-1}\mathbf{d})^{(m\times 1)} \\ \mathbf{0}^{((n-m)\times 1)} \end{pmatrix} = (\mathbf{D}^{(m\times m)},\ \mathbf{0}^{(m\times(n-m))})\begin{pmatrix} (\mathbf{D}^{-1}\mathbf{d})^{(m\times 1)} \\ \mathbf{0}^{((n-m)\times 1)} \end{pmatrix} = \mathbf{d}.$$

Let $\mathbf{e}_j^{(n-m)}$ be the jth column of the identity matrix $\mathbf{I}^{(n-m)}$. For each vector $\mathbf{x}_0^j$ we have:

$$\mathbf{Ax}_0^j = \mathbf{AU}\begin{pmatrix} \mathbf{0}^{(m\times 1)} \\ \mathbf{e}_j^{(n-m)} \end{pmatrix} = (\mathbf{D}^{(m\times m)},\ \mathbf{0}^{(m\times(n-m))})\begin{pmatrix} \mathbf{0}^{(m\times 1)} \\ \mathbf{e}_j^{(n-m)} \end{pmatrix} = \mathbf{0}.$$

The linear independence of vectors $\mathbf{x}_0^j,\ 1 \le j \le n-m$ follows from the identity matrix.

It is known that the matrix $\mathbf{U}$ is unique if $\mathbf{A}$ is a rational matrix of full row rank, and that the sizes of $\mathbf{U}$ and of the Hermite normal form $(\mathbf{D},\ \mathbf{0})$ are polynomially bounded by the size of $\mathbf{A}$, see Chapter 5 of Schrijver [17]. Hence, the vectors (12) are of sizes polynomially bounded by $\mathbf{A}$ and $\mathbf{d}$. ■

Let $\hat{\mathbf{b}}_j = (\hat{b}_{1j}, \ldots, \hat{b}_{n+m+1,j})^T,\ 1 \le j \le n+1$, denote the jth vector of the reduced basis $\hat{\mathbf{B}}$, obtained by applying the basis reduction algorithm to $\mathbf{B}$.

Theorem 4 *Assume that there exists an integral vector* $\mathbf{x}$ *satisfying the system* $\mathbf{Ax} = \mathbf{d}$. *There exist numbers* N_{01} *and* N_{02} *such that if* $N_1 > N_{01}$, *and if* $N_2 > N_1 N_{02}$, *then the vectors* $\hat{\mathbf{b}}_j \in \mathbb{Z}^{n+m+1}$ *of the reduced basis* $\hat{\mathbf{B}}$ *have the following properties:*

1. $\hat{b}_{n+1,j} = 0$ *for* $1 \le j \le n-m$,
2. $\hat{b}_{ij} = 0$ *for* $n+2 \le i \le n+m+1$ *and* $1 \le j \le n-m+1$,
3. $\hat{b}_{n+1,n-m+1} = N_1$.

Moreover, the sizes of N_{01} *and* N_{02} *are polynomially bounded by the sizes of* $\mathbf{A}$ *and* $\mathbf{d}$.

Proof: We first determine N_{01}. Let $\mathbf{x}_d$ and $\mathbf{x}_0^j$, $1 \leq j \leq n-m$, be determined as in equation (12). Let

$$\mathbf{v}_j = \begin{pmatrix} \mathbf{x}_0^j \\ 0 \end{pmatrix} \text{ for } 1 \leq j \leq n-m,$$

and let

$$\mathbf{v}_{n-m+1} = \begin{pmatrix} \mathbf{x}_d \\ 1 \end{pmatrix}.$$

Next, define vectors $\mathbf{z}_1, \ldots, \mathbf{z}_{n-m+1}$ as follows:

$$\begin{aligned} \mathbf{z}_j &= \mathbf{B}\mathbf{v}_j &= \begin{pmatrix} \mathbf{v}_j \\ \mathbf{0}^{(m\times 1)} \end{pmatrix} \quad \text{for } 1 \leq j \leq n-m, \\ \mathbf{z}_{n-m+1} &= \mathbf{B}\mathbf{v}_{n-m+1} &= \begin{pmatrix} \mathbf{x}_d \\ N_1 \\ \mathbf{0}^{(m\times 1)} \end{pmatrix}. \end{aligned}$$

The vectors $\mathbf{z}_1, \ldots, \mathbf{z}_{n-m+1}$ are linearly independent and belong to the lattice L (cf. vectors (10) and (9)).

Select N_{01} such that

$$N_{01}^2 > 2^{n+m} \max\{||\mathbf{z}_1||^2, \ldots, ||\mathbf{z}_{n-m}||^2\} = 2^{n+m} \max\{||\mathbf{v}_1||^2, \ldots, ||\mathbf{v}_{n-m}||^2\}.$$

From Lemma 2 we have that

$$||\hat{\mathbf{b}}_j||^2 \leq 2^{n+m} \max\{||\mathbf{z}_1||^2, \ldots, ||\mathbf{z}_{n-m}||^2\} < N_{01}^2 \quad \text{for } 1 \leq j \leq n-m.$$

Suppose that $\hat{b}_{n+1,j} \neq 0$ for some $j : 1 \leq j \leq n-m$. Then $||\hat{\mathbf{b}}_j||^2 \geq \hat{b}_{n+1,j}^2 \geq N_1^2 > N_{01}^2$ as N_1 divides $\hat{\mathbf{b}}_{n+1,j}$, which contradicts the outcome of Lemma 2. We therefore have that $\hat{b}_{n+1,j} = 0$ for $1 \leq j \leq n-m$.

Next, select N_{02} such that

$$N_{02}^2 > 2^{n+m} \max\{||\mathbf{v}_1||^2, \ldots, ||\mathbf{v}_{n-m+1}||^2\}.$$

Due to Lemma 2, the following holds for the reduced basis vectors $\hat{\mathbf{b}}_j, 1 \leq j \leq n-m+1$:

$$\begin{aligned} ||\hat{\mathbf{b}}_j||^2 &\leq 2^{n+m} \max\{||\mathbf{z}_1||^2, \ldots, ||\mathbf{z}_{n-m+1}||^2\} \leq \\ &2^{n+m} \max\{||\mathbf{z}_1||^2, \ldots, ||\mathbf{z}_{n-m}||^2, N_1^2||\mathbf{v}_{n-m+1}||^2\} = \\ &2^{n+m} \max\{||\mathbf{v}_1||^2, \ldots, ||\mathbf{v}_{n-m}||^2, N_1^2||\mathbf{v}_{n-m+1}||^2\} \leq \\ &N_1^2\, 2^{n+m} \max\{||\mathbf{v}_1||^2, \ldots, ||\mathbf{v}_{n-m}||^2, ||\mathbf{v}_{n-m+1}||^2\} < N_1^2 N_{02}^2. \end{aligned}$$

Suppose that $\hat{b}_{ij} \neq 0$ for some index pair $i,j:\ n+2 \leq i \leq n+m+1,\ 1 \leq j \leq n-m+1$. Then $||\hat{\mathbf{b}}_j||^2 \geq \hat{b}_{ij}^2 \geq N_2^2$ as N_2 divides $\hat{b}_{ij}$. As a consequence, $||\hat{\mathbf{b}}_j||^2 \geq N_2^2 > N_1^2 N_{02}^2$, which contradicts the outcome of Lemma 2. We therefore have that $\hat{b}_{ij} = 0$ for all $i,j:\ n+2 \leq i \leq n+m+1,\ 1 \leq j \leq n-m+1$.

Next, we prove Property 3. The vector $\mathbf{z}_{n-m+1} = (\mathbf{x}_d^T, N_1, \mathbf{0}^{(1\times m)})^T = \mathbf{B}(\mathbf{x}_d^T, 1)^T$ belongs to L spanned by $\mathbf{B}$. The lattice L is also spanned by the reduced basis $\hat{\mathbf{B}}$, and hence the vector $(\mathbf{x}_d^T, N_1, \mathbf{0}^{(1\times m)})^T$ can be obtained as $(\mathbf{x}_d^T, N_1, \mathbf{0}^{(1\times m)})^T = \hat{\mathbf{B}}(\lambda_1, \ldots, \lambda_{n+1})^T$. Properties 1 and 2 imply the following:

$$0\lambda_1 + \cdots + 0\lambda_{n-m} + \hat{b}_{n+1,n-m+1}\lambda_{n-m+1} \tag{13}$$

$$+\hat{b}_{n+1,n-m+2}\lambda_{n-m+2} + \cdots + \hat{b}_{n+1,n+1}\lambda_{n+1} = N_1$$

and

$$0\lambda_1 + \cdots + 0\lambda_{n-m} + 0\lambda_{n-m+1} \tag{14}$$

$$+\hat{b}_{i,n-m+2}\lambda_{n-m+2} + \cdots + \hat{b}_{i,n+1}\lambda_{n+1} = 0 \quad \text{for all } n+2 \le i \le n+m+1.$$

The last m rows of the basis $\mathbf{B}$ have rank m, since we assume that the matrix $\mathbf{A}$ has full row rank. The reduced basis $\hat{\mathbf{B}}$ is obtained by applying a series of elementary column operations to $\mathbf{B}$. Such operations preserve rank, and hence, the rank of the last m rows of $\hat{\mathbf{B}}$ is m as well. Consider the system of m equations (14). The $\hat{b}_{ij}$-elements in these equations correspond to the last m rows of $\hat{\mathbf{B}}$. For the sake of clearness we rewrite these equations below without all elements $\hat{b}_{ij}$ that are equal to zero.

$$\hat{b}_{i,n-m+2}\lambda_{n-m+2} + \cdots + \hat{b}_{i,n+1}\lambda_{n+1} = 0 \quad \text{for all } n+2 \le i \le n+m+1.$$

This is an $m \times m$ system that according to the previous statement has rank m. The only solution to this system is therefore $\lambda_{n-m+2} = \cdots = \lambda_{n+1} = 0$. If we use this solution in equation (13) we obtain $\hat{b}_{n+1,n-m+1}\lambda_{n-m+1} = N_1$, which implies that $\hat{b}_{n+1,n-m+1}$ divides N_1. Moreover, N_1 divides $\hat{b}_{n+1,n-m+1}$ (cf. the proof of Property 1). We can therefore conclude that $\hat{b}_{n+1,n-m+1} = N_1$.

To prove that the sizes of N_{01} and N_{02} are polynomially bounded by the sizes of $\mathbf{A}$ and $\mathbf{d}$ we observe that $||\mathbf{z}_j||^2 = ||\mathbf{v}_j||^2 = ||\mathbf{x}_0^j||^2$ for $1 \le j \le n-m$, and that $||\mathbf{v}_{n-m+1}||^2 = ||\mathbf{x}_d||^2 + 1$. We then use Lemma 3 to obtain the result. ■

Notice that Theorem 4 says that the first $n-m$ columns of the reduced basis $\hat{\mathbf{B}}$ are of the form

$$\begin{pmatrix} \mathbf{x}_0 \\ 0 \\ \mathbf{0}^{(m\times 1)} \end{pmatrix},$$

and that the $(n-m+1)$st column of $\hat{\mathbf{B}}$ is of the form

$$\begin{pmatrix} \mathbf{x} \\ N_1 \\ \mathbf{0}^{(m\times 1)} \end{pmatrix},$$

cf. Proposition 1. The integers N_1 and N_2 can in practice be chosen considerably smaller than the formulae in the proof of Theorem 4 indicate, since the upper bound (11) on the length of the reduced basis vectors in most cases contains a considerable slack. In our computational study we typically used the values

$N_1 = 10^3$ and $N_2 = 10^4$. Also, note that the definitions of N_{01} and N_{02} are independent of whether or not there exists a solution to $\mathbf{Ax} = \mathbf{d}$ that satisfies the bounds as well.

Example 2 Consider the following instance of problem (1). Here $m = 3$, $n = 6$. Determine whether there exists a vector $\mathbf{x} \in \mathbb{Z}^6$ such that

$$\begin{array}{ccccccccccccc}
6x_1 & + & x_2 & + & 3x_3 & + & 3x_4 & & & & & = & 17 \\
 & & & & & & & & 2x_5 & + & x_6 & = & 11 \\
 & & & & 4x_3 & + & 1x_4 & & & + & 2x_6 & = & 27
\end{array}$$

$$0 \le x_1 \le 2;\ 0 \le x_2 \le 3;\ 0 \le x_3 \le 5;\ 0 \le x_4 \le 2;\ 0 \le x_5 \le 5;\ 0 \le x_6 \le 14$$

The initial basis $\mathbf{B}$ for the lattice L for this instance is:

$$\mathbf{B} = \left(\begin{array}{cccccc|c}
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & N_1 \\
\hline
6N_2 & N_2 & 3N_2 & 3N_2 & 0 & 0 & -17N_2 \\
0 & 0 & 0 & 0 & 2N_2 & N_2 & -11N_2 \\
0 & 0 & 4N_2 & N_2 & 0 & 2N_2 & -27N_2
\end{array} \right)$$

After applying the basis reduction algorithm to this basis we obtain:

$$\hat{\mathbf{B}} = \left(\begin{array}{ccc|c|ccc}
0 & 1 & 1 & 0 & 0 & 0 & 1 \\
-3 & -3 & 0 & -1 & 1 & 0 & 0 \\
1 & -1 & 2 & 5 & 0 & -1 & 0 \\
0 & 0 & -4 & 1 & 0 & 1 & -2 \\
1 & -1 & 1 & 4 & 0 & -1 & 0 \\
-2 & 2 & -2 & 3 & 0 & 2 & 1 \\
\hline
0 & 0 & 0 & N_1 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & N_2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & N_2 \\
0 & 0 & 0 & 0 & 0 & N_2 & 0
\end{array} \right)$$

The vector $\mathbf{x}_d = (0, -1, 5, 1, 4, 3)^T$ (the first $n = 6$ elements of column $n - m + 1 = 4$) is a solution to the system of equations, whereas each of the vectors that are formed by taking the first n elements of the first $n - m$ vectors constitutes a solution to $\mathbf{Ax} = \mathbf{0}$. Note that the second element of the vector $\mathbf{x}_d$ violates its lower bound. ■

5 The algorithm

Here we discuss how we can use the properties stated in Theorem 4 to design an algorithm for solving the feasibility problem (1). From Proposition 1 and

Theorem 4 we can conclude that the first $n-m+1$ columns of $\hat{\mathbf{B}}$ are interesting for our problem as they are of the form given in Proposition 1. From these columns we in particular want to use the first $n+1$ elements. Hence, let $\hat{\mathbf{B}}'$ be the matrix consisting of the first $n+1$ elements of the first $n-m+1$ columns of $\hat{\mathbf{B}}$. The columns of $\hat{\mathbf{B}}'$ form a basis of the null-space $\mathbb{Z}^{n+1} \cap \mathrm{N}(\mathbf{A}, -\frac{1}{N_1}\mathbf{d}) = \{(\mathbf{x}, y) \in \mathbb{Z}^{n+1} : \mathbf{A}\mathbf{x} - \frac{1}{N_1}\mathbf{d}y = \mathbf{0}\}$.

$$\hat{\mathbf{B}}' = \begin{pmatrix} \mathbf{X}_0^{(n\times(n-m))} & \mathbf{x}_d \\ \mathbf{0}^{(1\times(n-m))} & N_1 \end{pmatrix} \tag{15}$$

Notice that each column $\mathbf{x}_0^j$ of the submatrix $\mathbf{X}_0$ satisfies $\mathbf{A}\mathbf{x}_0^j = \mathbf{0}$. Hence, $\mathbf{x}_0^j \in \mathbb{Z}^n \cap \mathrm{N}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^n : \mathbf{A}\mathbf{x} = \mathbf{0}\}$. In our algorithm we first check whether the vector $\mathbf{x}_d$ satisfies the lower and upper bounds. If yes, we are done. If any of the bounds is violated we search for a vector that is feasible, or for a proof that no feasible vector exists, by branching on linear integer multiples of the columns of $\mathbf{X}_0$, i.e., vectors in $\mathrm{N}(\mathbf{A})$. Note that by adding any linear integer combination of vectors in $\mathrm{N}(\mathbf{A})$, $\mathbf{x}_\lambda$, to $\mathbf{x}_d$ we obtain a vector $\mathbf{x}_d + \mathbf{x}_\lambda$ that satisfies $\mathbf{A}(\mathbf{x}_d + \mathbf{x}_\lambda) = \mathbf{d}$. For the feasible instances with $m = 1$ the search for a feasible vector turned out to be particularly easy. To speed up the algorithm for most of these instances we developed a heuristic as follows. Suppose that we are at iteration t of the heuristic and that an integer linear combination of $t_0 < t$ vectors of $\mathbf{X}_0$ has been added to vector $\mathbf{x}_d$. The vector obtained in this way is called the "current vector". For simplicity we assume that only variable x_k of the current vector violates one of its bound constraints. At iteration t we add or subtract an integer multiple λ_t of the column vector $\mathbf{x}_0^t$ if the violation of variable x_k's bound constraint is reduced and if no other bound constraints becomes violated. As soon as the value of x_k satisfies its bounds, we do not consider any larger values of λ_t. If the heuristic does not find any feasible solution, we call an exact branching algorithm that branches on linear combinations of vectors of $\mathbf{X}_0$. A summary of the complete algorithm is given in Figure 4.

Example 2 *(cont.)* In our example we obtain

$$\hat{\mathbf{B}}' = \left(\begin{array}{rrr|r} 0 & 1 & 1 & 0 \\ -3 & -3 & 0 & -1 \\ 1 & -1 & 2 & 5 \\ 0 & 0 & -4 & 1 \\ 1 & -1 & 1 & 4 \\ -2 & 2 & -2 & 3 \\ \hline 0 & 0 & 0 & N_1 \end{array}\right)$$

The second element of the vector $\mathbf{x}_d = (0, -1, 5, 1, 4, 3)^T$ violates its lower bound. By subtracting the vector $\mathbf{x}_0^1 = (0, -3, 1, 0, 1, -2)^T$ from to $\mathbf{x}_d$, we obtain the vector $\mathbf{x} = (0, 2, 4, 1, 3, 5)^T$ that satisfies the equations as well as all the bounds. Another possibility is for instance to add the vector $-2\mathbf{x}_0^1 + \mathbf{x}_0^2$ to $\mathbf{x}_d$, where $\mathbf{x}_0^2 = (1, -3, -1, 0, -1, 2)^T$. This results in the longer vector $(1, 2, 2, 1, 1, 9)$. ■

```
procedure main(A, d, u)
begin
    store initial basis B;
    compute B̂;
    extract B̂′ from B̂;
    if 0 ≤ x_d ≤ u then return x_d;
    heuristic(B̂′);
    if heuristic fails then
        branch on linear combinations of columns j = 1, ..., n − m of the
        submatrix X_0;
    return feasible vector x, or a proof that no such vector exists;
end
```

Figure 4: ALGORITHM 1.

6 Computational experience

We solved 23 instances of problem (1). Twelve of the instances were feasible and eleven infeasible. The instances with names starting with "P" in Table 1 were obtained from Philips Research Labs, and the instances with names starting with "F" are the Frobenius instances of Cornuéjols et al. [5]. Here we used the Frobenius number as right-hand side d. The two instances, with names starting with "E", were derived from F3 and F4, and the "M"-instances are market split instances that were generated according to Cornuéjols and Dawande [4]. They generated the a_{ij}-coefficients uniformly and independently in the interval $[0, 99]$. The d_i-coefficients are computed as $d_i = \lfloor \frac{1}{2} \sum_{j=1}^{n} a_{ij} \rfloor$. It should be noted that this formulae yield instances that are provably difficult for LP-based branch-and-bound, see [4]. An important purpose of these instances was to stimulate the development of algorithms that use concepts different from traditional branch-and-bound.

The information in Table 1 is interpreted as follows. In the first three columns, "Instance", "m", and "n", the instance names and the dimension of the instances are given. A "Y" in column "Type" means that the instance is feasible, and an "N" that it is not feasible. In the two columns of LP-based branch-and-bound, "LP B&B", the number of nodes and the computing time are given. In the "# Nodes" column, $500,000^{*}$ means that we terminated the search after 500,000 nodes without reaching a result. Two asterisks after the number of nodes indicate that a rounding error occurred, i.e., that the rounded solution given by the algorithm did not satisfy the diophantine equation. In both cases we do not report on the computing times since no result was obtained. For the market split problem we terminated the branch-and-bound algorithm after 54,000 seconds without any result. Hence, the $54,000^{+}$ in the "Time" column of LP B&B. In the three columns corresponding to our algorithm, "Algorithm", the column "Heur." gives the number of vectors of $\mathbf{X}_0$ that was used in the integer linear combination of vectors added to the vector $\mathbf{x}_d$ by

Instance	m	n	Type	LP B&B # Nodes	LP B&B Time (s)	Algorithm Heur.	Algorithm # Nodes	Algorithm Time (s)
P1	1	5	Y	420^{**}	–	1		$< 10^{-5}$
P2	1	5	Y	327	0.09	1		$< 10^{-5}$
P3	1	4	Y	75	0.05	0		$< 10^{-5}$
P4	1	5	Y	313^{**}	–	1		$< 10^{-5}$
P5	1	5	Y	231	0.11	2		$< 10^{-5}$
P6	1	5	Y	313^{**}	–	1		$< 10^{-5}$
E1	1	6	Y	3,271	0.97	0		$< 10^{-5}$
E2	1	7	Y	$500,000^{*}$	–	0		$< 10^{-5}$
F1	1	5	N	$500,000^{*}$	–	–	1	$< 10^{-3}$
F2	1	6	N	$500,000^{*}$	–	–	5	0.01
F3	1	6	N	$500,000^{*}$	–	–	1	$< 10^{-3}$
F4	1	7	N	$500,000^{*}$	–	–	1	0.01
F5	1	8	N	$500,000^{*}$	–	–	5	0.01
M1	5	40	N	–	$54,000^{+}$	–	29,420	211.5
M2	5	40	Y	–	$54,000^{+}$	–	21,890	131.8
M3	5	40	Y	–	$54,000^{+}$	–	14,998	97.3
M4	5	40	N	–	$54,000^{+}$	–	24,168	148,1
M5	5	40	N	–	$54,000^{+}$	–	23,682	144.4
M6	6	50	N	–	$54,000^{+}$	–	1,908,353	17,385.2
M7	6	50	Y	–	$54,000^{+}$	–	535,079	4,051.6
M8	6	50	N	–	$54,000^{+}$	–	2,032,090	16,037.6
M9	6	50	N	–	$54,000^{+}$	–	1,759,106	13,175.0
M10	6	50	Y	–	$54,000^{+}$	–	1,034,202	7,710.8

Table 1: Results of the computational experiments.

the heuristic in order to obtain a feasible solution. A zero in column "Heur." therefore means that the vector $\mathbf{x}_d$ was feasible. Notice that for every feasible instance with $m = 1$, the heuristic found a feasible solution. For the feasible market split instances the heuristic was not successful and hence we always used exact branching. For the infeasible instances the heuristic obviously failed, and therefore the sign "–" is given in the "Heur."-column. A one in the column "# Nodes" means that we solved the problem in the root node by using logical implications. The computing times are given in seconds on a 144MHz Sun Ultra-1. For the LP-based branch-and-bound we used CPLEX version 4.0.9 [7], and in our algorithm we used LiDIA, a library for computational number theory [13], for computing the reduced basis.

7 Discussion

Our results indicate that the instances are relatively easy once they are represented in a suitable way. Using the basis $\mathbf{e}_j$ and branching on variables as in LP-based branch-and-bound is clearly not a good approach here, but it is the standard way of tackling integer programs. Using basis reduction seems to give a more natural representation of the problem. Instead of representing the problem in terms of multiples of 1, which is the case when we use unit vectors as a basis, we obtain a scaling of the problem. For our instances the computing times were short, and, contrary to standard LP-based branch-and-bound, we avoid round-off errors. It is also worth noticing that the infeasibility of instances F1–F5, M1, M4 and M5 was quickly verified using our algorithm. The instances M6, M8 and M9 took longer to compute, but the time needed is still short relative to other methods that have been used. It should be mentioned here that the market split instances up to size 5×40 can also be solved quickly by a sorting algorithm, see Cornuéjols and Dawande [4]. This approach is, however, rather memory consuming and therefore does not seem to be practical for larger instances. Another algorithm that was used by Cornuéjols and Dawande was based on group theory. This algorithm seems quite robust, but took an order of magnitude longer time than our algorithm for the 5×40 instances. For the 6×50 instances the computing times were comparable.

Unlike the algorithms of H.W. Lenstra, Jr. [12], and of Lovász and Scarf [14] our algorithm does not run in polynomial time for fixed n. After applying basis reduction we obtain and $\mathbf{x}$-vector that satisfies $\mathbf{Ax} = \mathbf{d}$,

$$\mathbf{x} = \mathbf{x}_d + \mathbf{X}_0\lambda, \tag{16}$$

where $\lambda \in \mathbb{Z}^{n-m}$. By using this reformulation we can formulate our problem (1) in the following equivalent way:

does there exist a vector $\lambda \in \mathbb{Z}^{n-m}$ such that $-\mathbf{x}_d \leq \mathbf{X}_0\lambda \leq \mathbf{u} - \mathbf{x}_d$? (16)

The polytope P defined by the linear inequalities of (16) is full-dimensional and provides a "scaled" formulation of the original problem. There is no guarantee, however, that the polytope P is not thin in certain directions, cf. Section 3. We are currently investigating the effect of applying the generalized basis reduction algorithm of Lovász and Scarf [14] to the polytope P. We are also studying how to implement the algorithm by Lenstra [12]. For the test instances we have considered here such an algorithm was not necessary, but in order to solve larger instances of the market split problem we need a more refined branching algorithm.

References

[1] K. Aardal, A.K. Lenstra, C. Hurkens (1997). An algorithm for solving a diophantine equation with lower and upper bounds on the variables. Research report UU-CS-1997-40, Department of Computer Science, Utrecht University.

[2] W. Cook, 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.

[3] D. Coppersmith (1997). Small solutions to polynomial equations, and low exponent RSA vulnerability. *Journal of Cryptology* 10, 233–260.

[4] G. Cornuéjols, M. Dawande (1998). A class of hard small 0-1 programs. In: R.E. Bixby, E.A. Boyd, R Z. Ríos-Mercado (eds.) *Integer Programming and Combinatorial Optimization, 6th International IPCO Conference.* Lecture Notes in Computer Science 1412, pp 284–293, Springer-Verlag, Berlin Heidelberg.

[5] G. Cornuéjols, R. Urbaniak, R. Weismantel, L. Wolsey (1997). Decomposition of integer programs and of generating sets. In: R. Burkard, G. Woeginger (eds.), *Algorithms – ESA '97.* Lecture Notes in Computer Science 1284, pp 92–103, Springer-Verlag, Berlin Heidelberg.

[6] M.J. Coster, A. Joux, B.A. LaMacchia, A.M. Odlyzko, C.P. Schnorr (1992). Improved low-density subset sum algorithms. *Computational Complexity* 2, 111–128.

[7] CPLEX Optimization Inc. (1989). Using the CPLEX Callable Library.

[8] G. Havas, B.S. Majewski, K.R. Matthews (1996). Extended gcd and Hermite normal form algorithms via lattice basis reduction. Working paper, Department of Mathematics, The University of Queensland, Australia.

[9] A. Joux, J. Stern (1998). Lattice reduction: A toolbox for the cryptanalyst. *Journal of Cryptology* 11, 161–185.

[10] J.C. Lagarias, A.M. Odlyzko (1985). Solving low-density subset sum problems. *Journal of the Association for Computing Machinery* 32, 229–246.

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

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

[13] LiDIA – A library for computational number theory. TH Darmstadt / Universität des Saarlandes, Fachbereich Informatik, Institut für Theoretische Informatik.
`http://www.informatik.th-darmstadt.de/pub/TI/LiDIA`

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

[15] C.P. Schnorr, M. Euchner (1994). Lattice basis reduction: improved practical algorithms and solving subset sum problems. *Mathematical Programming* 66, 181–199.

[16] C.P. Schnorr, H.H. Hörner (1995). Attacking the Chor-Rivest Cryptosystem by improved lattice reduction. In: L.C. Guillou, J.-J Quisquater (eds.), *Advances in Cryptology – EUROCRYPT '95.* Lecture Notes in Computer Science 921, pp 1–12, Springer-Verlag, Berlin Heidelberg.

[17] A. Schrijver (1986). *Theory of Integer and Linear Programming.* Wiley, Chichester.

[18] L.A. Wolsey (1998). Private Communication.