Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Linear Algebra and its Applications 384 (2004) 135–154
LINEAR ALGEBRA AND ITS APPLICATIONS
www.elsevier.com/locate/laa

$\mathbb{N}$-solutions to linear systems over $\mathbb{Z}$ [☆]

Pilar Pisón-Casares [a,*], Alberto Vigneron-Tenorio [b]

[a] Dpto de Álgebra, University of Sevilla, Apartado 1160, 41080 Sevilla, Spain
[b] Dpto de Matemáticas, University of Cádiz, C/ Por-vera, 54, 11403 Jerez de la Frontera, Cádiz, Spain

Received 11 February 2003; accepted 15 January 2004

Submitted by J. Dias da Silva

Abstract

We show how Dickson's lemma yields an algorithm for computing the general $\mathbb{N}$-solution to a linear system over $\mathbb{Z}$. The method is based in determining several particular solutions. We propose and compare two methods computing these particular solutions. The first one uses techniques based on Gröbner Bases and the second one uses other traditional linear programming methods.

AMS classification: 13P1O

Keywords: Diophantine equations; Integer programming; Gröbner bases

1. Introduction

We denote by $\mathbb{N}$ the set of non-negative integers, and by $\mathbb{Z}$ the set of integers.

We are interested in methods for computing $\mathbb{N}$-solutions to linear system over $\mathbb{Z}$. Some geometrical and homological approaches are considered in [18]. We remind that solving these systems in non-negative integer variables in known to be a NP-complete problem.

There are some methods [2,3,13,14,17,19] which need an exhaustive search in a large region. Therefore, they are not practical.

☆ Both authors supported by MCyT Spain, BFM2003-00933 and Junta de Andalucía FQM304.
* Corresponding author.
E-mail addresses: ppison@us.es (P. Pisón-Casares), alberto.vigneron@uca.es (A. Vigneron-Tenorio).

0024-3795/$ - see front matter
doi:10.1016/j.laa.2004.01.003

A geometric algorithm in the case of one equation was given in [5]. The generalization of its techniques to the general case appeared in [7]. An alternative algebraic method was given in [10]. All these works as well as other computational researches can be found in [21].

The application of the Gröbner bases to integer programming problems comes from [6,14]. Nevertheless, the practical behavior of Gröbner bases methods proposed in [6,14,15] to solving large scale integer problems is hampered by the computation of Gröbner bases, which is quite time consuming in general (see [12]). The main drawback is that it is necessary to compute a Gröbner basis over a polynomial ring with a lot of variables because Elimination Theory is used. New methods avoiding this problem are considered in [9,11], and applied in [22] for computing particular solutions.

The main result in this paper (Algorithm 5.3) is based on the following fact: The general $\mathbb{N}$-solution to a linear system over $\mathbb{Z}$ can be computed by using any method for computing a particular $\mathbb{N}$-solution. This conclusion comes from the fact that we reduce the computation of the minimal $\mathbb{N}$-solutions or *vertices*, to determining a particular solution of the given system, and particular solutions of a finite number of new systems where some variables have been fixed. The reduction consists of a recursive process that we explain in Section 2.

On the other hand, we give a comparison of the practical performance of our method (Algorithm 5.3) using Gröbner Bases (Algorithm 3.2) and using traditional linear programming methods (Algorithm 4.9), concretely Farkas' lemma. The comparison of running times between both methods is collected in Table in Section 5. We conclude that Gröbner Bases provide an algorithm considerably faster that the traditional methods.

The description of the algorithm based on Gröbner bases, Algorithm 3.2, is in Section 3. The particular solutions are determined by using Algorithm 3.1, which appears in [22].

The algorithm based on traditional linear programming methods, 4.9, is described in Section 4. There is a classical result in Linear Programming (Proposition 4.1) determining whether a given homogeneous system has a non-trivial $\mathbb{N}$-solution, and in this case, finding a particular solution. The idea appeared in [16]. The result is a constructive version of Farkas' lemma, because it is an effective method to determine whether or not a vector is in the cone generated by a finite number of vectors. The computational behaviour of this method is very good, but in the recursive scheme that we propose, the non-homogeneous systems appear even if one starts with a homogeneous one. For this reason, we have looked for a generalization of Proposition 4.1 in the non-homogeneous case, and for that we have used the orthogonal projection vector and a technical result (Proposition 4.5). Then, we get a new method to obtain particular solutions (Algorithm 4.9).

Finally, in the Section 5 we describe our main algorithm (Algorithm 5.3), we see some examples and give a table with the computation time to compare the two proposed methods, Table in Section 5.

We have implemented our algorithm in MapleV and it is available by ftp at anonymous ftp.uca.es/pub/matematicas/nsol.zip

The comparison with the computational techniques summarized in [21] is not explored.

2. The general solution to a homogeneous system

Let M be a $p \times q$ $\mathbb{Z}$-matrix. Let $S := \{\mathbf{s} \in \mathbb{N}^q \mid M\mathbf{s} = \mathbf{0}\}$. S is clearly a semigroup of $\mathbb{N}^q$ with zero element. We will see that it is finitely generated.

Definition 2.1. $\mathbf{s} \in S - \{\mathbf{0}\}$ is a *vertex* if $\mathbf{s} = \boldsymbol{\gamma} + \boldsymbol{\delta}$, $\boldsymbol{\gamma}, \boldsymbol{\delta} \in S$, implies $\mathbf{s} = \boldsymbol{\gamma}$ or $\mathbf{s} = \boldsymbol{\delta}$.

We denote $VS :=$ set of vertices of S.

Remark 2.2. VS generates S.

Notice that VS is the set of the non-null elements in S which are minimal for the natural partial order in $\mathbb{N}^q$:

$$\boldsymbol{\gamma} \leqslant \boldsymbol{\delta} \iff \boldsymbol{\delta} - \boldsymbol{\gamma} \in \mathbb{N}^q.$$

This means that the set VS is the *Hilbert basis* of the system.

Notation 2.3. If $H \subset \mathbb{N}^q$, $1 \leqslant i \leqslant q$, $\alpha \in \mathbb{N}$, we denote:

- $H(i,\alpha) := \{\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_q) \in H \mid \gamma_i = \alpha\}$.
- If $H \neq \{\mathbf{0}\}$, $VH := \{\boldsymbol{\gamma} \in H - \{\mathbf{0}\} \mid \boldsymbol{\gamma}$ is minimal for $<\}$.
- If $H = \{\mathbf{0}\}$, $VH := \{\mathbf{0}\}$.

We call *vertices* of H the elements in VH.

Dickson's lemma (see for example [1]) states that VH is a finite set. This result is a corollary of our following lemma.

Lemma 2.4 (Dickson's lemma). *Let* $H \subset \mathbb{N}^q$, $\mathbf{s} = (s_1, \ldots, s_q) \in H$, $\mathbf{s} \neq \mathbf{0}$, *and let*

$$F = \{\mathbf{s}\} \cup \bigcup_{i=1}^{q} \bigcup_{\alpha=0}^{s_i-1} V(H(i,\alpha)).$$

Then, $VH = VF$.

Proof. It is enough to prove that

$$\forall \boldsymbol{\delta} \in H\ \exists \boldsymbol{\gamma} \in F \quad \text{with } \boldsymbol{\gamma} \leqslant \boldsymbol{\delta}.$$

Let $\boldsymbol{\delta}$ be an element of H. If $\mathbf{s} \leqslant \boldsymbol{\delta}$ there is nothing to prove. Otherwise, there exists an i, $1 \leqslant i \leqslant q$, such that $\delta_i < s_i$. Then, for $\alpha = \delta_i$, $\boldsymbol{\delta} \in H(i, \alpha)$ and there exists $\boldsymbol{\gamma} \in V(H(i, \alpha))$ with $\boldsymbol{\gamma} \leqslant \boldsymbol{\delta}$. $\square$

We can identify $H(i, \alpha)$ with a subset of $\mathbb{N}^{q-1}$ and apply again 2.4 to find $V(H(i, \alpha))$. Then, by recurrence, to obtain the set VH it is enough to solve the following problems:

If $H' = H(i_1, \alpha_1)(i_2, \alpha_2) \cdots (i_r, \alpha_r)$, $1 \leqslant i_j \leqslant q, \alpha_j \in \mathbb{N}, 1 \leqslant j \leqslant r$:

Problem 1. Determine if $H' = \emptyset$ or not. In the second case, get $\mathbf{s} \in H'$.

Problem 2. Obtain VH' for $H' \subset \mathbb{N}$.

The usual version of Dickson's lemma is clear now.

Corollary 2.5. *VH is finite.*

Proof. If $H' \subset \mathbb{N}$, since $\mathbb{N}$ is a well ordered set, VH' is empty or has only a unique element. Thus, by recursively applying 2.4, we obtain that the set VH is always finite. $\square$

We apply the above argument to the case $H = S$. From 2.2 and 2.5 it follows that S is a finitely generated semigroup in $\mathbb{N}^q$. We are interested in computing the generating set VS. Notice that, with the above notation, $S(i, \alpha)$ is the set of the $\mathbb{N}$-solutions to the linear system $M\mathbf{x} = 0$, where $x_i = \alpha$. This system can be non-homogeneous.

Remark 2.6. Consider the system $Mx = \mathbf{c}$ with M a $p \times 1$ $\mathbb{Z}$-matrix and $\mathbf{c} \in \mathbb{Z}^p$. Notice that it is obvious to determine whether or not there exists $s \in \mathbb{N}$ such that $Ms = \mathbf{c}$. Moreover, if there exists such an s, it is unique.

Then, in the case $H = S$, or more generally, in the case $H = R$ where

$$R := \{\mathbf{s} \in \mathbb{N}^q \mid M\mathbf{s} = \mathbf{c}\},$$

with $\mathbf{c} \in \mathbb{Z}^p$, Problem 2 is easy because H' is unitary or empty (2.6). On the other hand, Problem 1 is equivalent to determining whether or not there exists an $\mathbb{N}$-solution to a linear system over $\mathbb{Z}$ and, in the case that there exists an $\mathbb{N}$-solution, finding a particular one. We can use every method solving this problem (see the introduction) and obtain the following result.

Proposition 2.7. *Let M be a $p \times q$ $\mathbb{Z}$-matrix, and $\mathbf{c} \in \mathbb{Z}^p$. Let*

$$R := \{\mathbf{s} \in \mathbb{N}^q \mid M\mathbf{s} = \mathbf{c}\}.$$

There exists an algorithm computing VR, by computing particular solutions of integer systems.

In particular, the algorithm computes a generating set of the semigroup

$$S := \{\mathbf{s} \in \mathbb{N}^q \mid M\mathbf{s} = \mathbf{0}\}.$$

In the next sections, we explain two algorithms computing the vertices of S and R, Algorithms 3.2 and 4.9. Algorithm 3.2 uses the methods in [22] based on Semigroup Ideals and Gröbner Bases. Algorithm 4.9 uses Classical Linear Programming.

3. Semigroup ideals methods

Let $\Gamma \subset \mathbb{Z}^p$ be a finitely generated subsemigroup with zero element. Let $\{\mathbf{n}_1, \ldots, \mathbf{n}_r\} \subset \Gamma$ be a set of generators for Γ.

Let k be a field. We consider $A = k[X_1, \ldots, X_r]$ the polynomial ring in r indeterminates, and $B = k[t_1^{\pm}, \ldots, t_p^{\pm}] = k[\mathbf{t}^{\pm}]$ the Laurent ring in p indeterminates. We denote by $t^n = t_1^{n_1} \cdots t_p^{n_p}$, where $\mathbf{n} = (n_1, \ldots, n_p) \in \mathbb{Z}^p$.

Let $\varphi : A \to B$ be the k-algebra homomorphism, defined by $\varphi(X_i) = \mathbf{t}^{\mathbf{n}_i}$. We denote $I_\Gamma := ker(\varphi)$.

In order to provide our first algorithm satisfying 2.7, we associate with the system a semigroup Γ and determine a finite generating set of the ideal I_Γ. In fact, in the recursive process we will need to consider several semigroups like Γ, but only a finite number of them.

The ideal I_Γ is generated (see [20]) by the binomial set

$$\mathscr{B} = \left\{ \mathbf{X}^\alpha - \mathbf{X}^\beta \,\middle|\, \sum_{i=1}^{r} \alpha_i n_i = \sum_{i=1}^{r} \beta_i n_i \quad \text{with } \alpha_i \beta_i = 0 \ \forall i \right\}.$$

It is well known that by using the Implicitization Algorithm for rational parametrizations (see for example [8]) one can obtain a finite generating set of I_Γ contained in $\mathscr{B}$, if the set $\{\mathbf{n}_1, \ldots, \mathbf{n}_r\}$ is given. However, new techniques, [9,11], improve this algorithm in our particular case. Both are based on Gröbner Bases.

Let $M\mathbf{x} = \mathbf{0}$ be a system, where M is a $p \times q$ $\mathbb{Z}$-matrix. Let Γ be the subsemigroup of $\mathbb{Z}^p$ generated by the column vectors of M, $\{\mathbf{n}_1, \ldots, \mathbf{n}_q\}$. We have that $I_\Gamma \subset k[X_1, \ldots, X_q]$.

Notice that $\exists \mathbf{u} \in \mathbb{N}^q$, $\mathbf{u} \neq \mathbf{0}$, such that $M\mathbf{u} = \mathbf{0}$, if and only if the binomial $1 - \mathbf{X}^u$ is in I_Γ.

Moreover, $M\mathbf{u} = \mathbf{0}$ with $\mathbf{u} \in \mathbb{N}^q$ implies that $\mathbf{u} = \mathbf{0}$, it is equivalent to the fact that the semigroup Γ satisfies $\Gamma \cap (-\Gamma) = \{\mathbf{0}\}$, because the unique way to write $\mathbf{0}$ as a linear combination of the generators of Γ is the trivial one.

The condition $\Gamma \cap (-\Gamma) = \{\mathbf{0}\}$ guarantees Nakayama lemma for Γ-graded modules (see [4]), whence it is called *Nakayama condition*.

Then, Γ is Nakayama if and only if there exists no binomial $1 - \mathbf{X}^\alpha$ in I_Γ.

Moreover, if $\mathscr{C}$ is a generating set of I_Γ contained in $\mathscr{B}$, there exists a binomial $1 - \mathbf{X}^\alpha$ in I_Γ if and only if there exists a binomial $\pm(1 - \mathbf{X}^\beta)$ in $\mathscr{C}$.

On the other hand, consider a system $M\mathbf{x} = \mathbf{c}$, with $\mathbf{c} \in \mathbb{Z}^p$, $\mathbf{c} \neq \mathbf{0}$. Set now Γ_c as the subsemigroup of $\mathbb{Z}^p$ generated by the column vectors of M and $\mathbf{c}$, $\{\mathbf{n}_1, \ldots, \mathbf{n}_q, \mathbf{c}\}$. With this notation, we have now that $I_{\Gamma_c} \subset k[X_1, \ldots, X_{q+1}]$.

Notice that $\exists \mathbf{u} \in \mathbb{N}^q$, such that $M\mathbf{u} = \mathbf{c}$, if and only if $\exists \mathbf{u}' = (\mathbf{u}, 0) \in \mathbb{N}^{q+1}$, such that $(M \mid \mathbf{c})\mathbf{u}' = (M \mid \mathbf{c})\mathbf{e}_{q+1}$, where $\mathbf{e}_{q+1} = (0, \ldots, 0, 1) \in \mathbb{N}^{q+1}$.

Therefore, the $\mathbb{N}$-solvability for the system is equivalent to the existence of a binomial $X_{q+1} - \mathbf{X}^{\alpha}$ in I_{Γ_c}, where $\mathbf{X}$ does not contain the variable X_{q+1}. (It is enough to take $\boldsymbol{\alpha} = \mathbf{u}$.)

Suppose that Γ_c is Nakayama, and let $\mathscr{C}$ be a generating set of I_{Γ_c} contained in $\mathscr{B}$. In particular, there is no binomial $\pm(1 - \mathbf{X}^{\alpha})$ in $\mathscr{C}$. Then, if there exists a binomial $X_{q+1} - \mathbf{X}^{\beta}$ in I_{Γ_c}, where $\mathbf{X}$ does not contain the variable X_{q+1}, there exists a binomial $\pm(X_{q+1} - \mathbf{X}^{\beta'})$ in $\mathscr{C}$. Moreover, if $\mathbf{X}^{\beta'}$ contains the variable X_{q+1}, since $\Gamma_c \subset \mathbb{Z}^p$ is cancellative, we have that the binomial

$$\pm\left(1 - \frac{\mathbf{X}^{\beta'}}{X_{q+1}}\right) \in I_{\Gamma_c}.$$

But, it is a contradiction because Γ_c is Nakayama.

Therefore, if Γ_c is Nakayama, the system is $\mathbb{N}$-solvable if and only if there exists a binomial $\pm(X_{q+1} - \mathbf{X}^{\beta}) \in \mathscr{C}$, where $\mathbf{X}$ does not contain the variable X_{q+1} and $\mathscr{C}$ is an arbitrary generating set of I_{Γ_c} contained in $\mathscr{B}$.

In the case Γ_c non-Nakayama, to find a similar condition we will need to consider a Gröbner basis of I_{Γ_c} with respect to a suitable monomial order. Fix a monomial order giving priority to the last variable. This means that $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{N}^{q+1}$ with $\alpha_{q+1} < \beta_{q+1}$ implies $\boldsymbol{\alpha} < \boldsymbol{\beta}$. It is well known that the reduced Gröbner basis of I_{Γ_c} is contained in $\mathscr{B}$ (see [20]). Let $\mathscr{G}$ be this Gröbner basis. It is clear that there exists a binomial $X_{q+1} - \mathbf{X}^{\beta}$ in I_{Γ_c} if and only if there is a binomial $\pm(X_{q+1} - \mathbf{X}^{\beta'})$ in $\mathscr{G}$, where $\mathbf{X}$ does not contain the variable X_{q+1}. Therefore, this condition is equivalent to the $\mathbb{N}$-solvability for the system.

Particular $\mathbb{N}$-solutions to a linear diophantine system can be computed by means of Semigroup Ideals as follows.

Algorithm 3.1. Particular $\mathbb{N}$-solution by means of Semigroup Ideals ([22])
Input: A system $M\mathbf{x} = \mathbf{c}$, where M is a $p \times q\,\mathbb{Z}$-matrix and $\mathbf{c} \in \mathbb{Z}^p$.
Output: A vector $\mathbf{u} \in \mathbb{N}^q$ such that $M\mathbf{u} = \mathbf{c}$, $\mathbf{u} \neq \mathbf{0}$ if it exists, or $\emptyset$ in the case there is no $\mathbf{u} \in \mathbb{N}^q$ such that $M\mathbf{u} = \mathbf{c}$.

1. If $\mathbf{c} = \mathbf{0}$
   - Take Γ to be the subsemigroup of $\mathbb{Z}^p$ generated by the column vectors of M, $\{\mathbf{n}_1, \ldots, \mathbf{n}_q\}$.
   - Compute a generating set of I_Γ, $\mathscr{C}$.
   - If there is a binomial $\pm(1 - \mathbf{X}^{\alpha}) \in \mathscr{C}$, output $\mathbf{u} = \boldsymbol{\alpha}$ and STOP.
   - Otherwise, output $\mathbf{u} = \mathbf{0}$ and STOP.

2. If $\mathbf{c} \neq \mathbf{0}$
   - Take Γ_c to be the subsemigroup of $\mathbb{Z}^p$ generated by the column vectors of M and $\mathbf{c}$, $\{\mathbf{n}_1, \ldots, \mathbf{n}_q, \mathbf{c}\}$.
   - Compute a generating set of I_{Γ_c}, $\mathscr{C}$.
   - If there is a binomial $\pm(X_{q+1} - \mathbf{X}^{\beta}) \in \mathscr{C}$, where $\mathbf{X}$ does not contain the variable X_{q+1}, output $\mathbf{u} = \boldsymbol{\beta}$ and STOP. Otherwise, continue.
   - If there is no binomial $\pm(1 - \mathbf{X}^{\alpha}) \in \mathscr{C}$, output $\emptyset$ and STOP. Otherwise, fix a monomial order giving priority to the last variable, and take a Gröbner basis for I_{Γ_c}, G.
   - If there is a binomial $\pm(X_{q+1} - \mathbf{X}^{\beta}) \in G$, where $\mathbf{X}$ does not contain the variable X_{q+1}, output $\mathbf{u} = \boldsymbol{\beta}$ and STOP.
   - Otherwise, output $\emptyset$ and STOP.

We can now describe a first algorithm satisfying Proposition 2.7.

Algorithm 3.2. Vertices by means of Semigroup Ideals
Input: A system $M\mathbf{x} = \mathbf{c}$, where M is a $p \times q$ $\mathbb{Z}$-matrix and $\mathbf{c} \in \mathbb{Z}^p$.
Output: VR for $R = \{\mathbf{s} \in \mathbb{N}^q \mid M\mathbf{s} = \mathbf{c}\}$.

1. If $q = 1$ use Remark 2.6 and STOP.
2. If $q \geqslant 2$, determine whether or not $R = \emptyset$ or $\{\mathbf{0}\}$ using Algorithm 3.1.
3. If $R = \emptyset$ or $\{\mathbf{0}\}$, output $VR = R$ and STOP.
4. Otherwise, take $\mathbf{s} = (s_1, \ldots, s_q) \in R - \{\mathbf{0}\}$.
5. For $i = 1, \ldots, q$, and $\alpha = 0, \ldots, s_i - 1$, compute $V(R(i, \alpha))$ by recursively calling Algorithm 3.2.
6. Compute VF for
$$F = \{\mathbf{s}\} \cup \bigcup_{i=1}^{q} \bigcup_{\alpha=0}^{s_i-1} V(R(i, \alpha)).$$
7. Output $VR = VF$.

Example 3.3. Consider the following system
$$\begin{cases} x_1 - 2x_2 + x_3 + 2x_4 = 0, \\ -2x_1 - x_2 - x_3 + 2x_4 = 0, \end{cases} \quad M := \begin{pmatrix} 1 & -2 & 1 & 2 \\ -2 & -1 & -1 & 2 \end{pmatrix}.$$
Let Γ be the subsemigroup of $\mathbb{Z}^2$ generated by the column vectors of M,
$$\Gamma := \langle (1, -2), (-2, -1), (1, -1), (2, 2) \rangle.$$
The associated ideal of Γ is
$$I_\Gamma = \left\langle x_2x_3^5 - x_1^3, x_2^3x_3^3x_4^2 - x_1, x_2^2x_3^4x_4 - x_1^2, x_1^4x_4 - x_3^6, x_2^4x_4^3x_3^2 - 1, x_1x_2x_4 - x_3 \right\rangle.$$
Since the binomial $x_2^4x_3^2x_4^3 - 1 \in I_\Gamma$, we obtain the particular $\mathbb{N}$-solution $\mathbf{s} := (0, 4, 2, 3)$.

Now, in order to construct the set F in step 6 of Algorithm 3.2, we must determine the sets $V(R(i,\alpha))$ for $i = 2, 3, 4$ (notice that $s_1 = 0$), and $\alpha = 0, \dots, s_i - 1$.

We obtain $R(2,0) = \{\mathbf{0}\} = V(R(2,0))$ using the fact that the ideal of the semigroup

$$\Gamma_{20} := \langle (1,-2), (1,-1), (2,2)\rangle \text{ is } I_{\Gamma_{20}} = \left\langle x_1^4 x_3 - x_2^6 \right\rangle.$$

By similar arguments we obtain that $R(2,1) = \emptyset = V(R(2,1))$ and $V(R(2,2)) = V(R(2,3)) = \emptyset$.

In order to determine the vertices of $R(3,0)$ we obtain a particular $\mathbb{N}$-solution $(2,6,0,5) \in R(3,0)$, $R(3,0)(1,0) = \{\mathbf{0}\} = V(R(3,0)(1,0))$, $R(3,0)(1,1) = \{\mathbf{0}\} = V(R(3,0)(1,1))$, $R(3,0)(2,\beta) = \emptyset = V(R(3,0)(2,\beta))$, for $\beta = 0,1,2,3,4,5$, and $R(3,0)(3,\beta) = \emptyset = V(R(3,0)(3,\beta))$ for $\beta = 0,1,2,3,4$. Therefore, $V(R(3,0)) = \{(2,6,0,5)\}$.

Using similar arguments as in cases above, we obtain $V(R(3,1)) = \{(1,5,1,4)\}$, $R(4,0) = \{\mathbf{0}\} = V(R(4,0))$, $R(4,1) = \emptyset = V(R(4,1))$, and $R(4,2) = \emptyset = V(R(4,2))$.

Then, $F = \{\mathbf{0}, (0,4,2,3), (2,6,0,5), (1,5,1,4)\}$, and we conclude that $VR = VF = F - \{\mathbf{0}\}$ is the Hilbert basis of the given system.

4. Classical linear programming methods

In this section we describe an alternative algorithm to 3.2. It is also based on 2.4, but it computes the particular solutions by means of linear programming methods.

First, consider the homogeneous case. Let M be a $p \times q$ $\mathbb{Z}$-matrix. Let

$$S := \{\mathbf{s} \in \mathbb{N}^q \mid M\mathbf{s} = \mathbf{0}\}.$$

Notice that if one is interested in the existence of a non-trivial $\mathbb{N}$-solution to $M\mathbf{x} = \mathbf{0}$, it is enough to study if there exists $\mathbf{u} \in \mathbb{Q}^q - \{\mathbf{0}\}$ with $u_i \geqslant 0$ for all $i = 1, \dots, q$, such that $M\mathbf{u} = \mathbf{0}$.

Suppose that L is the $\mathbb{Q}$-vector space of the solutions in $\mathbb{Q}^q$ to the linear system $M\mathbf{x} = \mathbf{0}$. Assume that $\mathbf{b}_1, \dots, \mathbf{b}_n \in \mathbb{Q}^q$ is a basis of L. Let B be the $n \times q$ matrix with row vectors $\mathbf{b}_i$. Denote $\mathbf{a}_1, \dots, \mathbf{a}_q \in \mathbb{Q}^n$ the column vectors of B.

Notice that $\exists \mathbf{u} \in L - \{\mathbf{0}\}$ with $u_i \geqslant 0 \forall i = 1, \dots, q$ if and only if

$$\exists \mathbf{v} \in \mathbb{Q}^n \text{ with } \mathbf{v}\cdot\mathbf{a}_i \geqslant 0\, \forall i = 1, \dots, q \text{ and } \mathbf{v}\cdot\mathbf{a}_i > 0 \text{ for at least one i.}$$

The relation between the vectors $\mathbf{u}$ and $\mathbf{v}$ is given by

$$\mathbf{u} = v_1\mathbf{b}_1 + \cdots + v_n\mathbf{b}_n = (\mathbf{v}\cdot\mathbf{a}_1, \dots, \mathbf{v}\cdot\mathbf{a}_q).$$

Then, it is enough to apply the following result which is a constructive version of Farkas' lemma. Its proof provides the correctness of Algorithm FP in [16] (Algorithm 4.2). Geometrically, one looks for a hyperplane which leaves all the vectors $\mathbf{a}_i$ in the same closed semispace, and at least an $\mathbf{a}_i$ is in the opened semispace.

Proposition 4.1 (Effective Farkas' lemma). *Let* $\mathbf{a}_1, \ldots, \mathbf{a}_q \in \mathbb{Q}^n$. *There exists an algorithm to determine whether or not there exists a vector* $\mathbf{v} \in \mathbb{Q}^n$ *such that* $\mathbf{v} \cdot \mathbf{a}_1 > 0$ *and* $\mathbf{v} \cdot \mathbf{a}_i \geqslant 0\ \forall i = 2, \ldots, q$. *In the case that it exists, the algorithm gives such a vector* $\mathbf{v}$.

Proof. We proceed by recurrence on q.

Suppose that $q = 1$. If $\mathbf{a}_1 = \mathbf{0}$, then there is no solution. Otherwise, if i is such that $a_{1i} \neq 0$, take $\mathbf{v}$ having i-coordinate equal to a_{1i} and 0 otherwise.

Assume that $q \geqslant 2$.

If there is no $\mathbf{w} \in \mathbb{Q}^n$ such that $\mathbf{w} \cdot \mathbf{a}_1 > 0$ and $\mathbf{w} \cdot \mathbf{a}_i \geqslant 0$, $\forall i = 2, \ldots, q-1$, there is no $\mathbf{v}$. If there exists $\mathbf{w}$ and $\mathbf{w} \cdot \mathbf{a}_q \geqslant 0$, take $\mathbf{v} = \mathbf{w}$. But if $\mathbf{w} \cdot \mathbf{a}_q < 0$, then let

$$(*) \quad \mathbf{a}'_i = \mathbf{a}_i - \frac{\mathbf{w} \cdot \mathbf{a}_i}{\mathbf{w} \cdot \mathbf{a}_q}\mathbf{a}_q \ \forall i = 1, \ldots, q-1.$$

If there exists $\mathbf{w}' \in \mathbb{Q}^n$ such that $\mathbf{w}' \cdot \mathbf{a}'_1 > 0$ and $\mathbf{w}' \cdot \mathbf{a}'_i \geqslant 0$, $\forall i = 2, \ldots, q-1$, it is enough to take

$$\mathbf{v} = \mathbf{w}' - \frac{\mathbf{w}' \cdot \mathbf{a}_q}{\mathbf{w} \cdot \mathbf{a}_q}\mathbf{w},$$

because $\mathbf{v} \cdot \mathbf{a}_i = \mathbf{w}' \cdot \mathbf{a}'_i$, $i = 1, \ldots, q-1$, and $\mathbf{v} \cdot \mathbf{a}_q = \mathbf{0}$. Otherwise, we now prove that there is no solution $\mathbf{v}$. We proceed by induction.

If $q = 2$, since there is no $\mathbf{w}'$, we have that $\mathbf{a}'_1 = 0$. Then, $\mathbf{a}_1 = \lambda \mathbf{a}_2$, with $\lambda = \frac{\mathbf{w} \cdot \mathbf{a}_1}{\mathbf{w} \cdot \mathbf{a}_2} < 0$. It is clear that there is no $\mathbf{v}$.

Suppose the result true for any integer less than q. Then, since there is no $\mathbf{w}'$, there exists r (the number of times that one has used $(*)$), $1 \leqslant r \leqslant q-1$, and for any $j = 1, \ldots, r-1$, there exist l_j with $l_j > l_{j-1}$, and $\mathbf{w}_j, \mathbf{a}_1^{(j)}, \ldots, \mathbf{a}_{l_j}^{(j)} \in \mathbb{Q}^n$ such that:

1) $\mathbf{a}_i^{(1)} = \mathbf{a}'_i$, $\forall i = 1, \ldots, q-1$.
2) $\mathbf{w}_j \cdot \mathbf{a}_1^{(j)} > 0$, $\mathbf{w}_j \cdot \mathbf{a}_i^{(j)} \geqslant 0$, $\mathbf{w}_j \cdot \mathbf{a}_{l_j}^{(j)} < 0$, $\forall i = 2, \ldots, l_j - 1$.
3) $\mathbf{a}_i^{(j+1)} = \mathbf{a}_i^{(j)} - \frac{\mathbf{w}_j \cdot \mathbf{a}_i^{(j)}}{\mathbf{w}_j \cdot \mathbf{a}_{l_j}^{(j)}}\mathbf{a}_{l_j}^{(j)}$, $1 \leqslant i \leqslant l_j - 1$.
4) $\mathbf{a}_1^{(r)} = \mathbf{0}$.

Denote by

$$\lambda_i^{(1)} = -\frac{\mathbf{w} \cdot \mathbf{a}_i}{\mathbf{w} \cdot \mathbf{a}_q}, \ i = 1, \ldots, q-1,$$

and

$$\lambda_i^{(j+1)} = -\frac{\mathbf{w}_j \cdot \mathbf{a}_i^{(j)}}{\mathbf{w}_j \cdot \mathbf{a}_{l_j}^{(j)}}, \ j = 1, \ldots, r-1, \ i = 1, \ldots, l_j - 1.$$

Notice that $\lambda_i^{(j)} \geqslant 0$, $\lambda_1^{(j)} > 0$, $\forall j, \forall i$.

We will prove that

$$\mathbf{a}_i^{(j)} = \mathbf{a}_i + \sum_{l=i+1}^{q} \mu_{il}^{(j)} \mathbf{a}_l, \text{ with } \mu_{il}^{(j)} \geqslant 0,$$

$\forall j = 1, \ldots, r, \forall i = 1, \ldots, l_j, \forall l = i+1, \ldots, q$.

We proceed by induction on j. For $j = 1$, it is enough to notice that from $(*)$

$$\mathbf{a}_i^{(1)} = \mathbf{a}_i' = \mathbf{a}_i + \lambda_i^{(1)} \mathbf{a}_q.$$

Assume that it is true for j. We will prove it for $j+1$. From 3),

$$\mathbf{a}_i^{(j+1)} = \mathbf{a}_i^{(j)} + \lambda_i^{(j+1)} \mathbf{a}_{l_j}^{(j)}, \; 1 \leqslant i \leqslant l_j - 1.$$

We can use the induction hypothesis to write

$$\mathbf{a}_i^{(j)} = \mathbf{a}_i + \sum_{l=i+1}^{q} \mu_{il}^{(j)} \mathbf{a}_l, \text{ and } \mathbf{a}_{l_j}^{(j)} = \mathbf{a}_{l_j} + \sum_{l=l_j+1}^{q} \mu_{l_j l}^{(j)} \mathbf{a}_l,$$

and obtain the result.

Now, since $\mathbf{a}_1^{(r)} = \mathbf{0}$, we have that

$$\mathbf{a}_1 = -\sum_{l=2}^{q} \mu_{1l}^{(r)} \mathbf{a}_i, \text{ with } \mu_{1l}^{(r)} \geqslant 0.$$

It is clear that there is no $\mathbf{v}$. □

The following algorithm satisfies Proposition 4.1.

Algorithm 4.2. Farkas ([16])
Input: Vectors $\mathbf{a}_1, \ldots, \mathbf{a}_q \in \mathbb{Q}^n$.
Output: A vector $\mathbf{v} \in \mathbb{Q}^n$ such that $\mathbf{v} \cdot \mathbf{a}_1 > 0$ and $\mathbf{v} \cdot \mathbf{a}_i \geqslant 0$ for any $i = 2, \ldots, q$, or $\emptyset$ in the case that there is no such $\mathbf{v}$.

1. If $q = 1$:
   - If $\mathbf{a}_1 = 0$, output $\emptyset$ and STOP.
   - Otherwise, determine i with $a_{1i} \neq 0$ and output $\mathbf{v}$ having i-coordinate equal to a_{1i} and 0 otherwise and STOP.
2. If $q \geqslant 2$, determine if there exists $\mathbf{w} \in \mathbb{Q}^n$ such that $\mathbf{w} \cdot \mathbf{a}_1 > 0$ and $\mathbf{w} \cdot \mathbf{a}_i \geqslant 0$ for every $i = 2, \ldots, q-1$, by recursively using Algorithm 4.2.
3. If there is no $\mathbf{w}$, then output $\emptyset$ and STOP.
4. Otherwise:
   - If $\mathbf{w} \cdot \mathbf{a}_q \geqslant 0$, output $\mathbf{v} = \mathbf{w}$ and STOP.
   - Otherwise, continue.
5. Let
$$\mathbf{a}_i' = \mathbf{a}_i - \frac{\mathbf{w} \cdot \mathbf{a}_i}{\mathbf{w} \cdot \mathbf{a}_q} \mathbf{a}_q \; \forall i = 1, \ldots, q-1.$$

Determine if there exists $\mathbf{w}' \in \mathbb{Q}^n$ such that $\mathbf{w}' \cdot \mathbf{a}'_1 > 0$ and $\mathbf{w}' \cdot \mathbf{a}'_i \geqslant 0$, $\forall i = 2, \ldots, q-1$, by Algorithm 4.2.

6. If there exists $\mathbf{w}'$, output

$$\mathbf{v} = \mathbf{w}' - \frac{\mathbf{w}' \cdot \mathbf{a}_q}{\mathbf{w} \cdot \mathbf{a}_q}\mathbf{w}$$

and STOP.

7. Otherwise, output $\emptyset$.

Remark 4.3

1. The algorithm above allows us to determine if there exists $\mu_i \leqslant 0$ such that $\mathbf{a}_1 = \sum_{i=2}^{q} \mu_i \mathbf{a}_i$, or equivalently, if $-\mathbf{a}_1$ is in the cone of $\mathbf{a}_2, \ldots, \mathbf{a}_q$, whence the name Farkas' lemma.
2. The above algorithm solves Problem 1 in the case $H' = S$. We describe this solution in Algorithm 4.4 below.
3. If $S \subset \mathbb{N}^2$, then since $S(i, \alpha) \subset \mathbb{N}$, Remark 2.6 and the remark above allow us to compute VS using Lemma 2.4.

Algorithm 4.4. Particular $\mathbb{N}$-solution to a homogeneous system
Input: A system $M\mathbf{x} = \mathbf{0}$, where M is a $p \times q\mathbb{Z}$-matrix.
Output: A vector $\mathbf{u} \in \mathbb{N}^q$, such that $M\mathbf{u} = \mathbf{0}$, $\mathbf{u} \neq \mathbf{0}$ if it exists.

1. If $q = 1$ use Remark 2.6.
2. If $q \geqslant 2$, let $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{Q}^q$ a basis of the $\mathbb{Q}$-vector space given by $M\mathbf{x} = \mathbf{0}$. Let B be the matrix with row vectors $\mathbf{b}_i$. Denote by $\mathbf{a}_1, \ldots, \mathbf{a}_q$ the columns of B.
3. While $i = 1, \ldots, q$
   – Determine if there exists a vector $\mathbf{v}$ such that $\mathbf{v} \cdot \mathbf{a}_i > 0$ and $\mathbf{v} \cdot \mathbf{a}_j \geqslant 0$, for any $j \neq i$ and $1 \leqslant j \leqslant q$, by Algorithm 4.2.
   – If there exists $\mathbf{v}$, let $\mathbf{u}' = (\mathbf{v} \cdot \mathbf{a}_1, \ldots, \mathbf{v} \cdot \mathbf{a}_q) \in \mathbb{Q}^q$. Output $\mathbf{u} = m\mathbf{u}'$, where m is the least common multiple of the denominators of $\mathbf{v}_i \cdot \mathbf{a}_q$, and STOP.
4. Output $\mathbf{u} = \mathbf{0}$.

Recall that to carry out the recursive technique in Lemma 2.4 we need to find a particular solution to a non-homogeneous system obtained by fixing a variable in $M\mathbf{x} = \mathbf{0}$.

Let M' be a $p \times (q-1)$ matrix over $\mathbb{Z}$ and $\mathbf{c} \in \mathbb{Z}^p$. To determine if there exists $\mathbf{u} \in \mathbb{N}^{q-1}$ such that $M'\mathbf{u} = \mathbf{c}$, we consider the homogeneous linear system with matrix $(-\mathbf{c} \mid M')$, and let $L \subset \mathbb{Q}^q$ be the $\mathbb{Q}$-vector space of its solutions. Then, there exists $\mathbf{u} \in \mathbb{N}^{q-1}$ if and only if there exists $(1, \mathbf{u}) \in L$ with $\mathbf{u} \in \mathbb{N}^{q-1}$.

Assume that $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{Q}^q$ is a basis of L. Let B be the $n \times q$ matrix with row vectors $\mathbf{b}_i$. Denote by $\mathbf{a}_1, \ldots, \mathbf{a}_q \in \mathbb{Q}^n$ the column vectors of B.

Notice that $\exists(1,\mathbf{u}) \in L$ with $\mathbf{u} \in \mathbb{N}^{q-1}$ if and only if

$$\exists \mathbf{v} \in \mathbb{Q}^n \text{ with } \mathbf{v}\cdot\mathbf{a}_1 = 1 \text{ and } \mathbf{v}\cdot\mathbf{a}_i \in \mathbb{N}, \forall i = 2,\ldots,q.$$

The relation between the vectors **u** and **v** is given by

$$(1,\mathbf{u}) = v_1\mathbf{b}_1 + \cdots + v_n\mathbf{b}_n = (\mathbf{v}\cdot\mathbf{a}_1,\ldots,\mathbf{v}\cdot\mathbf{a}_q).$$

Then, we consider the following problem:

Problem. Given vectors $\mathbf{a}_1,\ldots,\mathbf{a}_q \in \mathbb{Q}^n$, determine whether or not there exists $\mathbf{v} \in \mathbb{Q}^n$ such that $\mathbf{v}\cdot\mathbf{a}_1 = 1$ and $\mathbf{v}\cdot\mathbf{a}_i \in \mathbb{N}$, $\forall i = 2,\ldots,q$.

We denote by W the $\mathbb{Q}$-vector space generated by $\mathbf{a}_2,\ldots,\mathbf{a}_q$. If $\mathbf{a}_1 \notin W$, take $\hat{\mathbf{a}}_1$ to be the orthogonal projection of $\mathbf{a}_1$ onto $W^\perp$. It is clear that $\hat{\mathbf{a}}_1 \neq \mathbf{0}$ and $\mathbf{a}_1\cdot\hat{\mathbf{a}}_1 > 0$. Then, it is enough to take

$$\mathbf{v} = \frac{\hat{\mathbf{a}}_1}{\mathbf{a}_1\cdot\hat{\mathbf{a}}_1},$$

because $\mathbf{v}\cdot\mathbf{a}_1 = 1$ and $\mathbf{v}\cdot\mathbf{a}_i = 0$ for any $i = 2,\ldots,q$.

If $\mathbf{a}_1 \in W$, we distinguish two cases:

– If $\mathbf{a}_1 = \sum_{i=2}^q \mu_i\mathbf{a}_i$ with $\mu_i \leqslant 0$, then there is no **v** (Remark 4.3.1).
– Otherwise, take a linear combination of type $\mathbf{a}_1 = \sum_{i=2}^q \mu_i\mathbf{a}_i$. Let A be the $(q-1)\times n$ matrix with row vectors $\mathbf{a}_i$, $i = 2,\ldots,q$. Denote by $L_1 \subset \mathbb{Q}^{q-1}$ the $\mathbb{Q}$-vector space generated by the column vectors of A. Suppose that $C\mathbf{x} = \mathbf{0}$ are implicit equations of L_1, and let

$$S_1 = \{\mathbf{s} \in \mathbb{N}^{q-1} \mid C\mathbf{s} = \mathbf{0}\}.$$

Consider $\{\mathbf{s}_1,\ldots,\mathbf{s}_h\}$ a generating set of the semigroup S_1. Denote $D = (\mathbf{s}_1 \mid \ldots \mid \mathbf{s}_h)$ and

$$(m_1,\ldots,m_h) := (\mu_2,\ldots,\mu_q)D.$$

Then, we get the following result.

Proposition 4.5. *With assumptions and notations as above, the following conditions are equivalent:*

1. $\exists \mathbf{v} \in \mathbb{Q}^n$ *with* $\mathbf{v}\cdot\mathbf{a}_1 = 1$ *and* $\mathbf{v}\cdot\mathbf{a}_i \in \mathbb{N}$, $\forall i = 2,\ldots,q$.
2. $\exists \mathbf{w} \in \mathbb{N}^h$ *such that* $m_1w_1 + \cdots + m_hw_h = 1$.

In that case, it is enough to take **v** *as a particular solution of* $A\mathbf{x} = \mathbf{z}$, *with* $\mathbf{z} = D\mathbf{w}$.

Proof. $\boxed{1 \Rightarrow 2}$ Let $\mathbf{z} = A\mathbf{v}$. By 1, it is clear that $\mathbf{z} \in S_1$. Then, there exists $\mathbf{w} \in \mathbb{N}^h$ such that $\mathbf{z} = D\mathbf{w}$. The linear combination $\mathbf{a}_1 = \sum_{i=2}^q \mu_i\mathbf{a}_i$ and the equality $\mathbf{v}\cdot\mathbf{a}_1 = 1$, implies that $(\mu_2,\ldots,\mu_q)\cdot\mathbf{z} = 1$. Then,

$$m_1w_1 + \cdots + m_hw_h = 1.$$

$\boxed{2 \Rightarrow 1}$ Let $\mathbf{z} = D\mathbf{w}$. Since $\mathbf{z} \in S_1 \subset L_1$, we deduce that the ranks of A and $(A \mid \mathbf{z})$ are equal. Take $\mathbf{v}$ as a particular solution of $A\mathbf{x} = \mathbf{z}$. Now, it is enough to notice that the linear combination $\mathbf{a}_1 = \sum_{i=2}^{q} \mu_i \mathbf{a}_i$ implies that

$$\mathbf{v} \cdot \mathbf{a}_1 = (\mu_2, \ldots, \mu_q)A\mathbf{v} = (\mu_2, \ldots, \mu_q)D\mathbf{w} = 1. \qquad \square$$

Remark 4.6
1. Notice that the proof does not use the hypothesis $\mu_i \geqslant 0$ for at least one i, although this case is solved by Farkas's lemma.
2. From 4.3.3 we can determine the condition 1 in Proposition 4.5 for $q = 3$. Now, applying 2.4 we can calculate VS for $q = 3$. Then, by recurrence, we obtain a new method for computing vertices. In the last step, we need to find a particular $\mathbb{N}$-solution to a unique equation. For this we can use the method in [5].

   Problem is solved by the following algorithm.

Algorithm 4.7
Input: Vectors $\mathbf{a}_1, \ldots, \mathbf{a}_q \in \mathbb{Q}^n$, $q \geqslant 2$.
Output: A vector $\mathbf{v} \in \mathbb{Q}^n$ such that $\mathbf{v} \cdot \mathbf{a}_1 = 1$ and $\mathbf{v} \cdot \mathbf{a}_i \in \mathbb{N}$, or $\emptyset$ in the case there is no such $\mathbf{v}$.

1. Consider W the $\mathbb{Q}$-vector space generated by $\mathbf{a}_2, \ldots, \mathbf{a}_q$.
2. If $\mathbf{a}_1 \notin W$, take $\hat{\mathbf{a}}_1$ the orthogonal projection of $\mathbf{a}_1$ onto $W^\perp$. Output $\mathbf{v} = \frac{\hat{\mathbf{a}}_1}{\mathbf{a}_1 \cdot \hat{\mathbf{a}}_1}$ and STOP.
3. Otherwise, apply Algorithm 4.2:
   – If $\mathbf{a}_1 = \sum_{i=2}^{q} \mu_i \mathbf{a}_i$ with $\mu_i \leqslant 0$ (Remark 3.3.1), then output $\emptyset$ and STOP.
   – Otherwise, continue.
4. Take a linear combination $\mathbf{a}_1 = \sum_{i=2}^{q} \mu_i \mathbf{a}_i$.
5. Let A be the matrix with row vectors $\mathbf{a}_i$, $i = 2, \ldots, q$. Consider $C\mathbf{x} = \mathbf{0}$ implicit equations of $L_1 \subset \mathbb{Q}^{q-1}$ the $\mathbb{Q}$-vector space generated by the column vectors of A. Compute $\{\mathbf{s}_1, \ldots, \mathbf{s}_h\}$ a generating set of

$$S_1 = \{\mathbf{s} \in \mathbb{N}^{q-1} \mid C\mathbf{s} = \mathbf{0}\},$$

   using Algorithm 4.9.
6. Let $(m_1, \ldots, m_h) = (\mu_2, \ldots, \mu_q)D$, where $D = (\mathbf{s}_1 \mid \ldots \mid \mathbf{s}_h)$.
   – If there exists $\mathbf{w} \in \mathbb{N}^h$ such that $m_1\mathbf{w}_1 + \ldots + m_h\mathbf{w}_h = 1$, output $\mathbf{v}$ a particular solution of $A\mathbf{x} = \mathbf{z}$ with $\mathbf{z} = D\mathbf{w}$ and STOP (see Remark 4.6.2).
   – Otherwise, output $\emptyset$.

Particular $\mathbb{N}$-solutions can be computed by means of Classical Linear Programming as follows.

Algorithm 4.8. Particular $\mathbb{N}$-solution by means of Classical Linear Programming
Input: A system $M'\mathbf{x} = \mathbf{c}$, where M' is a $p \times (q-1)\mathbb{Z}$- matrix and $\mathbf{c} \in \mathbb{Z}^p$.
Output: A vector $\mathbf{u} \in \mathbb{N}^{q-1}$ such that $M'\mathbf{u} = \mathbf{c}$, or $\emptyset$ in the case there is no such $\mathbf{u}$.

1. If $\mathbf{c} = \mathbf{0}$, use Algorithm 4.4.
2. Otherwise, continue.
3. If $q = 2$, use Remark 2.6.
4. If $q \geqslant 3$, let $M = (-\mathbf{c} \,|\, M')$. Consider $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{Q}^q$ a basis of the $\mathbb{Q}$-vector space
$$L = \{\mathbf{x} \in \mathbb{Q}^q \mid M\mathbf{x} = \mathbf{0}\}.$$
5. Let B be the matrix with row vectors $\mathbf{b}_i$, and let $\mathbf{a}_1, \ldots, \mathbf{a}_q \in \mathbb{Q}^n$ be the column vectors of B. Apply Algorithm 4.7
   – If there exists $\mathbf{v} \in \mathbb{Q}^n$ such that $\mathbf{v} \cdot \mathbf{a}_1 = 1$ and $\mathbf{v} \cdot \mathbf{a}_i \in \mathbb{N}$, then output $\mathbf{u}$ where
   $$(1, \mathbf{u}) = v_1\mathbf{b}_1 + \ldots + v_n\mathbf{b}_n = (\mathbf{v} \cdot \mathbf{a}_1, \ldots, \mathbf{v} \cdot \mathbf{a}_q),$$
   and STOP.
   – Otherwise, output $\emptyset$.

We can now describe a second algorithm satisfying Proposition 2.7.

Algorithm 4.9. Vertices by means of Classical Linear Programming
Input: A system $M\mathbf{x} = \mathbf{c}$, where M is a $p \times q\mathbb{Z}$-matrix and $\mathbf{c} \in \mathbb{Z}^p$.
Output: VR for $R = \{\mathbf{s} \in \mathbb{N}^q \mid M\mathbf{s} = \mathbf{c}\}$.

1. If $q = 1$ use Remark 2.6 and STOP.
2. If $q \geqslant 2$, determine whether or not $R = \emptyset$ or $\{\mathbf{0}\}$ using Algorithm 4.8.
3. If $R = \emptyset$ or $\{\mathbf{0}\}$, output $VR = R$ and STOP.
4. Otherwise, take $\mathbf{s} = (s_1, \ldots, s_q) \in R - \{\mathbf{0}\}$.
5. For $i = 1, \ldots, q$, and $\alpha = 0, \ldots, s_i - 1$, compute $V(R(i, \alpha))$ by recursively calling Algorithm 4.9.
6. Compute VF for
$$F = \{\mathbf{s}\} \cup \bigcup_{i=1}^{q} \bigcup_{\alpha=0}^{s_i-1} V(R(i, \alpha)).$$
7. Output $VR = VF$.

Remark 4.10. Notice that there is not circularity between algorithms above because 4.9 computes $S \subset \mathbb{N}^2$ by only Farkas' lemma (see Remark 4.3.3).

Example 4.11. We consider the same system that in 3.3.

A basis of the $\mathbb{Q}$-vector space given by $M\mathbf{x} = \mathbf{0}$ is $\mathbf{b}_1 = (4, 0, -6, 1)$, and $\mathbf{b}_2 = (-3, 1, 5, 0)$. Let $\mathbf{a}_1 = (4, -3)$, $\mathbf{a}_2 = (0, 1)$, $\mathbf{a}_3 = (-6, 5)$, and $\mathbf{a}_4 = (1, 0)$.

Using Algorithm 4.4, we find $\mathbf{v} = (5/18, 1/3)$. Since $(\mathbf{v} \cdot \mathbf{a}_1, \dots, \mathbf{v} \cdot \mathbf{a}_4) = (1/9, 1/3, 0, 5/18)$, we obtain $\mathbf{s} = (2, 6, 0, 5) \in R$.

As before, we compute a particular $\mathbb{N}$-solution $(0, 4, 2, 3) \in R(1, 0)$, and $R(1, 0)(1, 0) = \{\mathbf{0}\}$.

In order to compute $V(R(1, 0)(1, 1))$, using Algorithm 4.8, we need to consider the homogeneous system with matrix

$$\begin{pmatrix} -2 & 1 & 2 \\ -1 & -1 & 2 \end{pmatrix}.$$

A basis of the $\mathbb{Q}$-vector space of its solutions is given by $\mathbf{b}'_1 = (2, 1, 3/2)$. Let $\mathbf{a}'_1 = 2$, $\mathbf{a}'_2 = 1$, and $\mathbf{a}'_3 = 3/2$. We must determine whether or not there exists $\mathbf{v}' \in \mathbb{Q}$ such that $\mathbf{v}' \cdot \mathbf{a}'_1 = 1$, and $\mathbf{v}' \cdot \mathbf{a}'_i \in \mathbb{N}$, for $i = 2, 3$. In this case it is clear that there is no such $\mathbf{v}'$. Anyway we will use Algorithm 4.7 to show that there is not circularity between the algorithms used in this method (Remark 4.10).

With the notation of Algorithm 4.7, we can consider $\mathbf{a}'_1 = \mu_2 \mathbf{a}'_2 + \mu_3 \mathbf{a}'_3$, with $\mu_2 = 2$ and $\mu_3 = 0$. Therefore,

$$A = \begin{pmatrix} 1 \\ 3/2 \end{pmatrix} \text{ and } C\mathbf{x} = \mathbf{0}.$$

We need to compute a generating set for the semigroup $S_1 = \{\mathbf{s} \in \mathbb{N}^2 \mid C\mathbf{s} = \mathbf{0}\}$ using Algorithm 4.9, which calls to Algorithm 4.8, and this, to Algorithm 4.4. As before, we compute an element $\mathbf{s}'' = (2, 3) \in S_1$. It is easy to see that $S_1(1, 0) = \{(0, 0)\} = V(S_1(1, 0))$, $S_1(1, 1) = \emptyset = V(S_1(1, 1))$, $S_1(2, 0) = \{(0, 0)\} = V(S_1(2, 0))$, $S_1(2, 1) = \emptyset = V(S_1(2, 1))$, $S_1(2, 2) = \emptyset = V(S_1(2, 2))$. Therefore, $VS_1 = \{(2, 3)\}$.

With the notation in Algorithm 4.7, we have $D = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$, and $m = (2 \;\; 0)D = 4$. It is clear that there exists no $w \in \mathbb{N}$ such that $wm = 1$. Therefore (Proposition 4.5), $R(1, 0)(1, 1) = \emptyset = V(R(1, 0)(1, 1))$.

By similar arguments to those above, we obtain $R(1, 0)(1, \beta) = \emptyset$, for $\beta = 2, 3$, $R(1, 0)(2, 0) = \{\mathbf{0}\}$, $R(1, 0)(2, 1) = \emptyset$, $R(1, 0)(3, 0) = \{\mathbf{0}\}$, $R(1, 0)(3, \beta) = \emptyset$, for $\beta = 1, 2$. Thus, $V(R(1, 0)) = \{(0, 4, 2, 3)\}$.

Now, we determine whether or not $R(1, 1) = \emptyset$ using Algorithm 4.8. We consider the homogeneous system with matrix

$$\begin{pmatrix} 1 & -2 & 1 & 2 \\ -2 & -1 & -1 & 2 \end{pmatrix}.$$

A basis of the $\mathbb{Q}$-vector space is $\mathbf{b}''_1 = (-3, 1, 5, 0)$ and $\mathbf{b}''_2 = (4, 0, -6, 1)$. Let $\mathbf{a}''_1 = (-3, 4)$, $\mathbf{a}''_2 = (1, 0)$, $\mathbf{a}''_3 = (5, -6)$, and $\mathbf{a}''_4 = (0, 1)$.

We must see whether or not there exists $\mathbf{v}'' \in \mathbf{Q}^2$ such that $\mathbf{v}'' \cdot \mathbf{a}''_1 = 1$, and $\mathbf{v}'' \cdot \mathbf{a}''_i \in \mathbb{N}$, $i = 2, 3, 4$. With the notation in Algorithm 4.7, we can consider $\mathbf{a}''_1 = \mu'_2 \mathbf{a}''_2 + \mu'_3 \mathbf{a}''_3 + \mu'_4 \mathbf{a}''_4$, with $\mu'_2 = -3$, $\mu'_3 = 0$, and $\mu'_4 = 4$. (Notice that it is not possible that $\mu_i < 0$ for every i.)

Let A' be the matrix

$$A' := \begin{pmatrix} 1 & 0 \\ 5 & -6 \\ 0 & 1 \end{pmatrix}.$$

We need to compute the implicit equations, $C'\mathbf{x} = \mathbf{0}$, of the $\mathbb{Q}$-vector space generated by the column vectors of A'. We consider $C' = (-5\ 1\ 6)$, and $S'_1 = \{\mathbf{s} \in \mathbb{N}^3 \mid C'\mathbf{s} = \mathbf{0}\}$. As before we obtain

$$VS'_1 = \{(1, 5, 0), (6, 0, 5), (5, 1, 4), (4, 2, 3), (3, 3, 2), (2, 4, 1)\}.$$

Then

$$D' := \begin{pmatrix} 1 & 6 & 5 & 4 & 3 & 2 \\ 5 & 0 & 1 & 2 & 3 & 4 \\ 0 & 5 & 4 & 3 & 2 & 1 \end{pmatrix},$$

and $\mathbf{m}' := (-3\ 0\ 4)D' = (-3\ 2\ 1\ 0\ -1\ -2)$. It is clear that there exists $\mathbf{w}' \in \mathbb{N}^6$ such that $\mathbf{m}'\mathbf{w}' = 1$, for example $\mathbf{w}' = (0, 0, 1, 0, 0, 0)$. We take $\mathbf{v}'' = (5, 4)$ a particular solution of $A'\mathbf{x} = D'\mathbf{w}'$. Since $(\mathbf{v}' \cdot \mathbf{a}''_2, \mathbf{v}' \cdot \mathbf{a}''_3, \mathbf{v}' \cdot \mathbf{a}''_4) = (5, 1, 4)$, we obtain $(1, 5, 1, 4) \in R(1, 1)$.

As in previous cases, we compute the sets $R(1, 1)(1, \delta) = \emptyset$, $\delta = 0, \ldots, 4$, $R(1, 1)(2, 0) = \emptyset$, $R(1, 1)(3, \delta) = \emptyset$, $\delta = 0, \ldots, 3$, and we conclude that $V(R(1, 1)) = \{(1, 5, 1, 4)\}$.

Similarly we obtain $R(2, 0) = \{\mathbf{0}\}$, $R(2, \alpha) = \emptyset$, $\alpha = 1, \ldots, 3$, but $R(2, 4) \neq \emptyset$, $(0, 2, 3) \in R(2, 4)$ (consider $R(2, 4)$ as a subset of $\mathbb{N}^3$). Moreover, $R(2, 4)(2, 0) = R(2, 4)(2, 1) = R(2, 4)(3, 0) = R(2, 4)(3, 1) = R(2, 4)(3, 2) = \emptyset$. Then, $V(R(2, 4)) = \{(0, 4, 2, 3)\}$.

Similarly we obtain that $V(R(2, 5)) = \{(1, 5, 1, 4)\}$, $V(R(4, 0)) = \{\mathbf{0}\}$, $V(R(4, 1)) = V(R(4, 2)) = \emptyset$, $V(R(4, 3)) = \{(0, 4, 2, 3)\}$, and $V(R(4, 4)) = \{(1, 5, 1, 4)\}$.

Therefore $F = \{\mathbf{0}, (2, 6, 0, 5), (0, 4, 2, 3), (1, 5, 1, 4)\}$ and $VR = VF = F - \{\mathbf{0}\}$ is the Hilbert basis of the given system.

5. The general solution to a non-homogeneous system

Let M be a $p \times q$ $\mathbb{Z}$-matrix, and $\mathbf{c} \in \mathbb{Z}^p$. Let

$$S := \{\mathbf{s} \in \mathbb{N}^q \mid M\mathbf{s} = \mathbf{0}\}, \text{ and } R := \{\mathbf{s} \in \mathbb{N}^q \mid M\mathbf{s} = \mathbf{c}\}.$$

Remark 5.1

1. If $\gamma \in R$, then

$$\gamma + S := \{\gamma + \mathbf{s} \mid \mathbf{s} \in S\} \subset R.$$

2. If $\gamma, \boldsymbol{\beta} \in R$ and $\gamma \leqslant \boldsymbol{\beta}$, then $\boldsymbol{\beta} \in \gamma + S$.

Theorem 5.2. *With assumptions and notations as above, if* $VR = \{\gamma_1, \ldots, \gamma_r\}$, *then*

$$(*) \quad R = \bigcup_{i=1}^{r} (\gamma_i + S).$$

Therefore, there exists an algorithm computing all the elements in R.

Proof. The formula $(*)$ is clear by 5.1. Now, since it is possible to compute VR and a generating set of S (Proposition 2.7), we get an algorithm computing all the elements in R. □

Let $M' = (-\mathbf{c} \mid M)$, and $S' := \{\mathbf{s}' \in \mathbb{N}^{q+1} \mid M'\mathbf{s}' = \mathbf{0}\}$. Denote by

$$(VS')_0 := \{\mathbf{s} \in \mathbb{N}^q \mid (0, \mathbf{s}) \in VS'\}, \text{ and } (VS')_1 := \{\mathbf{s} \in \mathbb{N}^q \mid (1, \mathbf{s}) \in VS'\}.$$

It is easy to see that $VS = (VS')_0$ and $VR = (VS')_1$. Then, we obtain the following algorithm which satisfies Theorem 5.2.

Algorithm 5.3. General $\mathbb{N}$-solution to a linear system
Input: A system $M\mathbf{x} = \mathbf{c}$, where M is a $p \times q\mathbb{Z}$-matrix and $\mathbf{c} \in \mathbb{Z}^p$.
Output: VS and VR.

1. Take $M' = (-\mathbf{c} \mid M)$, and $S' := \{\mathbf{s}' \in \mathbb{N}^{q+1} \mid M'\mathbf{s}' = \mathbf{0}\}$.
2. Compute VS' using Algorithm 3.2 or Algorithm 4.9.
3. Output $VS = (VS')_0$ and $VR = (VS')_1$ and STOP.

Remark 5.4. Solving general systems of linear equations in non-negative integer variables is known to be a NP-complete problem. Then, in some situations to introduce an extra variable may drastically increase the complexity of solving the problem. In these cases, to compute directly VR and VS may be faster.

Example 5.5. Consider the following diophantine equation

$$x_1 - 3x_2 + 2x_3 - 5x_4 = 12.$$

We compute VS' where

$$S' = \{\mathbf{s}' \in \mathbb{N}^5 \mid M'\mathbf{s}' = \mathbf{0}\}, \quad M' = (-12\ 1\ \ -3\ 2\ \ -5).$$

$$\begin{aligned} VS' = \{ & (0, 1, 1, 1, 0), (0, 0, 2, 3, 0), (0, 0, 0, 5, 2), (1, 0, 0, 6, 0), (0, 0, 1, 4, 1), \\ & (0, 1, 0, 2, 1), (0, 5, 0, 0, 1), (1, 12, 0, 0, 0), (0, 3, 0, 1, 1), \\ & (1, 10, 0, 1, 0), (1, 8, 0, 2, 0), (1, 6, 0, 3, 0), (1, 4, 0, 4, 0), \\ & (1, 2, 0, 5, 0), (0, 3, 1, 0, 0)\} \end{aligned}$$

Thus, if S is the semigroup of the $\mathbb{N}$-solutions to $x_1 - 3x_2 + 2x_3 - 5x_4 = 0$, we have that

$$VS = (VS')_0 = \{(1, 1, 1, 0), (0, 1, 4, 1), (0, 0, 5, 2), (0, 2, 3, 0), (1, 0, 2, 1), (5, 0, 0, 1), (3, 1, 0, 0), (3, 0, 1, 1)\},$$

and

$$VR = (VS')_1 = \{(0, 0, 6, 0), (12, 0, 0, 0), (10, 0, 1, 0), (8, 0, 2, 0), (6, 0, 3, 0), (4, 0, 4, 0), (2, 0, 5, 0)\}.$$

Therefore,

$$R = [(0, 0, 6, 0) + S] \cup [(12, 0, 0, 0) + S] \cup [(10, 0, 1, 0) + S] \cup [(8, 0, 2, 0) + S] \cup [(6, 0, 3, 0) + S] \cup [(4, 0, 4, 0) + S] \cup [(2, 0, 5, 0) + S].$$

If one wants to use our implementation (see Introduction): Do the following:

```
> sol_general_nohomo([[1,-3,2,-5]],[12]);
```

It will be obtained as output

[[0, 0, 6, 0], [12, 0, 0, 0], [10, 0, 1, 0], [8, 0, 2, 0], [6, 0, 3, 0], [4, 0, 4, 0], [2, 0, 5, 0]], [[3, 1, 0, 0], [0, 2, 3, 0], [0, 0, 5, 2], [0, 1, 4, 1], [1, 1, 1, 0], [1, 0, 2, 1], [3, 0, 1, 1], [5, 0, 0, 1]]

Example 5.6. Consider the system

$$\begin{cases} x_1 + 2x_2 + 3x_3 - 5x_4 = 3 \\ -2x_1 - x_2 + 4x_3 + 5x_4 = -3 \end{cases}$$

$VS' = \{ (1, 1, 1, 0, 0), (0, 7, 0, 1, 2), (0, 5, 5, 0, 3), (10, 21, 0, 3, 0), (5, 14, 0, 2, 1) \}$. Thus, $VS = (VS')_0 = \{(7, 0, 1, 2), (5, 5, 0, 3)\}$, and $VR = (VS')_1 = \{(1, 1, 0, 0)\}$. Therefore

$$R = (1, 1, 0, 0) + S.$$

Using our implementation,

```
> sol_general_nohomo([[1,2,3,-5],[-2,-1,4,5]],[3,-3]);
```

we obtain

[[1, 1, 0, 0]], [[7, 0, 1, 2], [5, 5, 0, 3]]

Our practical performance is collected in the following table.[1] We give the comparison of running times between the two proposed methods, as well the used particular solution of the considered system used. (Notice that it bounds the searching space of the particular solutions of the new systems where some variables are fixed).

We conclude Algorithm 5.3 has a better computational behaviour if one uses Semigroup Ideals and Gröbner Bases (Algorithm 3.2), than if one instead uses Classical Linear Programming (Algorithm 4.9).

[1] All the computations have been done using MapleV R3, AMD-K6II-350, 64Mb RAM.

Homogeneous systems	Gröbner Bases	Classical integer programming
$\begin{pmatrix} 3 & -10 & 4 \end{pmatrix}$	1 s, $s = [2, 1, 1]$	3 s, $s = [10, 3, 0]$
$\begin{pmatrix} 1 & -3 & 2 & -5 \end{pmatrix}$	2 s, $s = [1, 1, 1, 0]$	5 s, $s = [5, 0, 0, 1]$
$\begin{pmatrix} 1 & -2 & 1 & 2 \\ -2 & -1 & -1 & 2 \end{pmatrix}$	4 s, $s = [0, 4, 2, 3]$	16 s, $s = [2, 6, 0, 5]$
$\begin{pmatrix} 1 & 2 & 3 & -5 \\ -2 & -1 & 4 & 5 \end{pmatrix}$	5 s, $s = [7, 0, 1, 2]$	7 s, $s = [5, 5, 0, 3]$
$\begin{pmatrix} 3 & -1 & -2 & -3 \\ 3 & -7 & 2 & -1 \end{pmatrix}$	7 s, $s = [2, 1, 1, 1]$	111 s, $s = [8, 6, 9, 0]$
$\begin{pmatrix} -4 & 1 & 0 & -1 & 0 & -2 \\ 0 & -1 & 0 & 2 & -3 & 1 \end{pmatrix}$	5 s, $s = [0, 0, 1, 0, 0, 0]$	1961 s, $s = [1, 8, 0, 4, 0, 0]$
$\begin{pmatrix} -1 & 2 & -3 & 0 & -1 \\ 0 & 1 & 0 & -3 & 0 \\ -1 & -2 & 0 & 0 & 1 \end{pmatrix}$	4 s, $s = [0, 3, 0, 1, 6]$	9 s, $s = [0, 3, 0, 1, 6]$
$\begin{pmatrix} -2 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 2 & 0 & -3 & 1 \\ 1 & -3 & 0 & 1 & -1 & 0 \\ 2 & 0 & 0 & -2 & 1 & 0 \end{pmatrix}$	17 s, $s = [1, 0, 2, 3, 4, 8]$	500 s, $s = [1, 0, 2, 3, 4, 8]$
$\begin{pmatrix} 0 & -1 & 2 & -3 & 0 & 0 \\ 1 & 0 & 1 & 0 & -3 & 0 \\ -1 & 4 & -2 & 0 & 0 & -1 \end{pmatrix}$	102 s, $s = [2, 2, 1, 0, 1, 4]$	Stop to 40000 s, $s = [18, 6, 3, 0, 7, 0]$
$\begin{pmatrix} 1 & 2 & -3 & -2 & -4 \\ 2 & -1 & -3 & 2 & 5 \end{pmatrix}$	49 s, $s = [1, 3, 1, 2, 0]$	Stop to 40000 s, $s = [9, 3, 5, 0, 0]$

References

[1] T. Becker, V. Weispfenning, H. Kredel, Gröbner Bases. A Computational Approach to Commutative Algebra, GTM 141, Springer, 1993.
[2] I. Borosh, A sharp bound for positive solutions of homogeneous linear diophantine equations, Proc. Amer. Math. Soc. 60 (1976) 19–21.
[3] I. Borosh, L.B. Treybig, Bounds on positive integer solutions of linear Diophantine equations, Proc. Amer. Math. Soc. 55 (1976) 299–304.
[4] E. Briales, A. Campillo, C. Marijuan, P. Pison, Minimal systems of generators for ideals of semigroups, J. Pure Appl. Algebra 124 (1998) 7–30.
[5] M. Clausen, A. Fortenbacher, Efficient solution of linear Diophantine equations, J. Symbolic Comput. 8 (1989) 201–216.
[6] P. Conti, C. Traverso, Buchberger algorithm and integer programming, in: Proceedings AAECC-9 (New Orleans), Springer LNCS 539, 1991, pp. 130–139.
[7] E. Cotejean, H. Devie (1994). An efficient Algorithm for solving systems of diophantine equations. Information and Computation, 113 (1) (1991), 143–172.
[8] D. Cox, J. Little, D. O'Shea, Ideals, Varietes, and Algorithms, Undergraduate Texts in Mathematics, Springer-Verlag, 1992.
[9] F. Di Biase, R. Urbanke, An algorithm to calculate the kernel of certain polynomial ring homomorphisms, Experimental Mathematics, 4, 3 (1995) 227–234.
[10] E. Domenjoud, Outils pour la Déduction Automatique dans les Théories Associatives-Commutatives. Thèse de doctorat d'Université, Université de Nancy I, 1991.
[11] S. Hosten, B.G. Sturmfels, An implementation of Gröbner Bases for integer programming, in: E. Balas, J. Clausen (Eds.), Integer Programming and Combinatorial Optimization, LNCS 920, Springer-Verlag, 1995, pp. 267–276.
[12] C. Moulinet-Ossola, Algorithmique des Réseaux et des Systèmes Diophantiens Linéaires, Thèse de doctorat, Université de Nice Sophia-Antipolis, 1995.
[13] C.H. Papadimitriou, On the complexity of integer programming, J. Assoc. Comput. Mach. 28 (1981) 765–768.
[14] L. Pottier, Minimal solutions of linear diophantine systems: bounds and algorithms, in: R.V. Book (Ed.), Proceedings of the 4th International Conference on Rewriting Techniques and Applications, Lecture Notes in Computer Science 488, Springer-Verlag, 1991, pp. 162–173.
[15] L. Pottier, Sub-groups of $\mathbb{Z}^n$, Standard Basis, and Linear Diophantine Systems, INRIA Research Report 1510, 1991.
[16] J.C. Rosales, On finitely generated submonoids of $\mathbb{N}^k$, Semigroup Forum 50 (1995) 251–262.
[17] J.C. Rosales, P.A. Garcia-sanchez, Non negative elements of subgroups of $\mathbb{Z}^n$, Linear Algebra Appl. 270 (1998) 351–357.
[18] R. Stanley, Combinatorics and commutative algebra, 2nd ed., Progress in Mathematics Vol.41, Boston Basel Berlin, Birkhäuser, 1996.
[19] A. Schrijver, Theory of Linear and Integer Programming, Wiley-Interscience series, John Wiley-Sons, 1986.
[20] B. Sturmfels, Gröbner Bases and Convex Polytopes, vol. 8, AMS University Lectures Series, 1995.
[21] A.P. Tomas, On Solving Linear Diophantine Constraints, Tese de Doutoramento, Faculdade de Ciencias da Universidada do Porto, 1997.
[22] A. Vigneron, Semigroup ideals and linear diophantine equations, Linear Algebra Appl. 295 (1999) 133–144.