$\mathbb{N}$ -solutions to linear systems over $\mathbb{Z}$ *

Pilar PISON-CASARES †
and Alberto VIGNERON-TENORIO ‡

May 17, 2000

Abstract

We show how Dickson's lemma yields an algorithm computing the general $\mathbb{N}$-solution to a linear system over $\mathbb{Z}$. The method is based in determining several particular solutions. If, to find these particular solutions, one uses techniques based on Gröbner Bases, our algorithm improves the traditional Integer Programming methods.

Key Words: Diophantine equations, Integer Programming, Gröbner Bases.

Introduction

We denote $\mathbb{N}$ the set of nonnegative integers, and $\mathbb{Z}$ the set of integers.

It is a basic and elementary result in Mathematics that the general solution to a linear system over a field k is obtained as the sum of a particular solution and the general solution to the associated homogeneous system. Moreover, the general solution in the homogeneous case, is a k-vector space. Rouche-Frobenius' theorem is a criterium determining whether or not a given linear system has a solution. This criterium is constructive because the rank of a matrix can be calculated in an effective way. Algorithms computing these solutions are well known.

In the case of integer solutions to the linear system over $\mathbb{Z}$ the situation is analogous. The general solution is also the sum of a particular solution and the general solution to the associated homogeneous system, which is a finitely generated group. Constructive criteria which determine, by means of the greatest common divisor, whether or not the system has a solution, as well as algorithms computing that solution, are also well known.

A recent and computational treatment of the above methods can be found in [5], [11], or [13].

However, if one is interested in the $\mathbb{N}$-solutions to the linear system over $\mathbb{Z}$, the situation is different. The general solution to the associated homogeneous system is a finitely generated semigroup (see for example [20]). But it is not true that the general solution to the non homogeneous system is obtained by adding this semigroup to a particular solution.

The general $\mathbb{N}$-solution to a linear system over $\mathbb{Z}$ can be written as a finite union of subsets. Any subset is the sum of a particular solution and a finitely generated semigroup S. These particular solutions, which we call *vertices*, are the minimal ones for the natural partial order. The semigroup S is the general solution to the associated homogeneous system. S is finitely generated by its vertices.

On the other hand, it is easy to see that the general $\mathbb{N}$-solution to a non homogeneous system can be deduced from the general $\mathbb{N}$-solution to a homogeneous system with a new

*1991 *Mathematics Subject Classification* [13P1O]

†Supported by Plan Propio de la Universidad de Sevilla

‡Partially supported by Plan Propio de la Universidad de Sevilla and Plan Propio de la Universidad de Cádiz

variable. Then, the problem is reduced to compute vertices of homogeneous diophantine linear systems. By historical reasons, the set of these vertices is known as the *Hilbert Basis*.

Explicit bounds are known for particular $\mathbb{N}$-solutions of solvable systems (see [1], [2], [15], [21], [16]). These bounds yield algorithms determining whether or not the system has an $\mathbb{N}$-solution and, in the solvable case, they provide a particular solution. In [19] a method computing the general $\mathbb{N}$-solution to a homogeneous linear system is given. Again, a bound for the coordinates of the solution is used. Now, the bound yields a generating set of the general solution, and it is obtained by means of a polynomial ideal associated to the semigroup of the $\mathbb{N}$-solutions. Since the degree of a special type of polynomial in the ideal is bounded by an integer [22], the authors deduce the bound for the generators in the semigroup. But these methods are not very efficient because they need an exhaustive search in a large region.

An efficient algorithm in the case of one equation was given in [4]. The generalization of its techniques to the general case appeared in [7]. An alternative method was given in [10]. All these works as well as other computational researches can be found in [23].

In [24] algorithms computing particular solutions are given. They are based on the following idea: The author associates a semigroup to the system in such way that, $\mathbb{N}$-solvability for the system is equivalent to the existence of a special binomial in the semigroup ideal. To know if such a binomial lies in the ideal, a generating set for the ideal is calculated using a generalization of the methods in [9] and [12], and a Gröbner basis is computed with respect to a suitable monomial order. The application of the Gröbner bases to integer programming problems comes from [6] and [16]. However, the practical application of the Gröbner bases methods to solving large scale integer problems is hampered by the computation of Gröbner bases, which is quite time consuming in general (see [14]). For example, in the methods proposed in [6], [16] and [17], the main drawback is that it is necessary computing a Gröbner basis over a polynomial ring with a lot of variables because Elimination Theory is used. New methods eluding this problem are considered in [9] and [12].

The main idea in this paper can be summarized saying that the general $\mathbb{N}$-solution to a linear system over $\mathbb{Z}$ can be computed by using **any** method computing a particular $\mathbb{N}$-solution. This conclusion comes from that we reduce the computation of the minimal $\mathbb{N}$-solutions or vertices (which are unknown a priori), 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 in a recursive process that we explain in the section 1.

On the other hand, we give the practical performance comparison of our method (Algorithm 4.3) using Gröbner Bases (Algorithm 2.2) and using traditional linear programming methods (Algorithm 3.9). The comparison of running times between both methods is collected in Table in section 4. We conclude than Gröbner Bases provide an algorithm considerably faster than the traditional methods.

The description of the algorithm based on Gröbner bases, Algorithm 2.2, is in the section 2. The particular solutions are determined by using Algorithm 2.1, which appears in [24].

The algorithm based on traditional linear programming methods, 3.9, is described in section 3. There is a classical result in Linear Programming (Proposition 3.1) determining whether a given homogeneous system has non trivial $\mathbb{N}$-solution, and in this case, finding a particular solution. The idea appeared in [18]. 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 3.1 in the non homogeneous case, and for that we have used the orthogonal projection vector and a technical result (Proposition 3.5). Then, we get a new method to obtain particular solutions (Algorithm 3.9).

Finally, in the section 4 we describe our main algorithm (Algorithm 4.3), we see some examples and give the table with the computation time to help compare the two proposed methods, Table in section 4.

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 [23] is not explored.

1 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 1.1. : $\mathbf{s} \in S - \{\mathbf{0}\}$ *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

$$VS := \text{ set of vertices of } S.$$

Remark 1.2. : 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 $\mathbb{N}^q$:

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

Notation 1.3. : *If* $H \subset \mathbb{N}^q$, $1 \leq i \leq q$, $\alpha \in \mathbb{N}$, *we denote:*

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

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

Lemma 1.4. : *(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

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

Let $\boldsymbol{\delta}$ be an element of H. If $\mathbf{s} \leq \boldsymbol{\delta}$ there is nothing to prove. Otherwise, there exists an i, $1 \leq i \leq q$, 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 $\boldsymbol{\gamma} \leq \boldsymbol{\delta}$. □

We can identify $H(i,\alpha)$ with a subset of $\mathbb{N}^{q-1}$ and apply again 1.4 to find $V(H(i,\alpha))$. Then, by recurrence, to obtain the set VH it is enough to solve the following problems:

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

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

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

The following result is clear now.

Corollary 1.5. : *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 1.4, we obtain that the set VH is always finite. □

We apply the above argument to the case $H = S$. From 1.2 and 1.5 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 1.6. : *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 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 obvious. 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 any method solving this problem (see the introduction) and obtain the following result.

Proposition 1.7. : *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.

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, Algorithm 2.2 and Algorithm 3.9. Algorithm 2.2 uses the methods in [24] based in Semigroup Ideals and Gröbner Bases. Algorithm 3.9 uses Classical Linear Programming.

2 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.

Let $\varphi : A \to B$ 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 1.7, we shall need to determine a finite generating set of the ideal I_Γ, where Γ is obtained from the given system. 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 by the binomial set

$$\mathcal{B} = \{\mathbf{X}^\alpha - \mathbf{X}^\beta \mid \sum_{i=1}^{r} \alpha_i n_i = \sum_{i=1}^{r} \beta_i n_i \ \text{ with } \ \alpha_i \beta_i = 0 \ \ \forall i\}$$

(see [22]).

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 $\mathcal{B}$, if the set $\{\mathbf{n}_1, \ldots, \mathbf{n}_r\}$ is given. However, new techniques, [9] and [12], improve this algorithm in our particular case. Both are based in 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

$$\exists \mathbf{u} \in \mathbb{N}^q, \ \mathbf{u} \neq \mathbf{0}, \text{ such that } \ M\mathbf{u} = \mathbf{0},$$

if and only if

$$\text{the binomial } \quad 1 - \mathbf{X}^u \ \text{ is in } \quad I_\Gamma.$$

Moreover,

$$M\mathbf{x} = \mathbf{0} \ \text{ with } \ \mathbf{u} \in \mathbb{N}^q \ \text{ implies that } \ \mathbf{u} = \mathbf{0},$$

it is equivalent to 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 [3]), 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 $\mathcal{C}$ is a generating set of I_Γ contained in $\mathcal{B}$, 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 $\mathcal{C}$.

On the other hand, consider a system $M\mathbf{x} = \mathbf{c}$, with $\mathbf{c} \in \mathbb{Z}^p$. Set now Γ 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 $I_\Gamma \subset k[X_1, \ldots, X_{q+1}]$.

Notice that

$$\exists \mathbf{u} \in \mathbb{N}^q, \ \text{ such that } \ M\mathbf{u} = \mathbf{c},$$

if and only if

$$\exists \mathbf{u}' = (\mathbf{u}, 0) \in \mathbb{N}^{q+1}, \ \text{ such that } \quad (M|\mathbf{c})\mathbf{u}' = (M|\mathbf{c})\mathbf{e}_{q+1},$$

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, where $\mathbf{X}$ does not contain the variable X_{q+1}. (It is enough to take $\boldsymbol{\alpha} = \mathbf{u}$)

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

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

But, it is a contradiction because Γ is Nakayama.

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

In the case Γ non Nakayama, to find a similar condition we will need to consider a Gröbner basis of I_Γ 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 is contained in $\mathcal{B}$ (see [22]). Let $\mathcal{G}$ be this Gröbner basis. It is clear that there exists a binomial $X_{q+1} - \mathbf{X}^{\beta}$ in I_Γ if and only if there is a binomial $\pm(X_{q+1} - \mathbf{X}^{\beta'})$ in $\mathcal{G}$, where $\mathbf{X}$ does not contain the variable X_{q+1}.

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

Algorithm 2.1. : *Particular $\mathbb{N}$-solution 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: 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 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, $\mathcal{C}$.*
- *If there is a binomial $\pm(1 - \mathbf{X}^{\alpha}) \in \mathcal{C}$, output $\mathbf{u} = \boldsymbol{\alpha}$ and STOP.*
- *Otherwise, output $\mathbf{u} = \mathbf{0}$ and STOP.*

2. If $\mathbf{c} \neq \mathbf{0}$

- *Take Γ 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, $\mathcal{C}$.*
- *If there is a binomial $\pm(X_{q+1} - \mathbf{X}^{\beta}) \in \mathcal{C}$, 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 $\pm(1 - \mathbf{X}^{\alpha}) \in \mathcal{C}$, output $\emptyset$ and STOP. Otherwise, fix a monomial order giving priority to the last variable, and take a Gröbner basis for I_Γ, 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 1.7.

Algorithm 2.2. : *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 1.6 and STOP.
2. If $q \geq 2$, determine whether or not $R = \emptyset$ or $\{\mathbf{0}\}$ using Algorithm 2.1.
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 2.2.
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 2.3. : Consider the following system

$$\left\{\begin{array}{rcl} x_1 - 2x_2 + x_3 + 2x_4 & = & 0 \\ -2x_1 - x_2 - x_3 + 2x_4 & = & 0 \end{array}\right.$$

Let M be the matrix

$$M := \begin{pmatrix} 1 & -2 & 1 & 2 \\ -2 & -1 & -1 & 2 \end{pmatrix}$$

We are going to compute the set VR, where

$$R = \{\mathbf{s} \in \mathbb{N}^q \mid M\mathbf{s} = \mathbf{0}\},$$

following the steps in Algorithm 2.2. Since this system is homogeneous, we have that $R = S$ with our usual notation. Therefore, VR is a generating set of the semigroup $R = S$, and we are going to compute the *Hilbert basis* of this system. We must determine whether or not $R = \emptyset$ or $\{\mathbf{0}\}$ by using Algorithm 2.1.

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

$$I_\Gamma =< x_2x_3^5 - x_1^3, x_2^3x_3^3x_4^2 - x_1, x_2^2x_3^4x_4 - x_1^2, x_1^4x_4 - x_3^6, x_2^4x_4^3x_3^2 - 1, x_1x_2x_4 - x_3 > .$$

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 2.2, we must determine the sets $V(R(i,\alpha))$ for $i = 2,3,4$ (notice that $s_1 = 0$), and $\alpha = 0, \ldots, s_i - 1$.

The set $R(2,\alpha)$ is the general $\mathbb{N}$-solution to the system

$$\begin{pmatrix} 1 & 1 & 2 \\ -2 & -1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = -\alpha \begin{pmatrix} -2 \\ -1 \end{pmatrix}.$$

Then, in order to compute $V(R(2,0))$, we consider the semigroup

$$\Gamma_{20} :=< (1,-2), (1,-1), (2,2) >,$$

and determine its ideal

$$I_{\Gamma_{20}} =< x_1^4x_3 - x_2^6 > .$$

Since there is no binomial $\pm(1 - \mathbf{X}^\beta)$ in its generating set, we conclude $R(2,0) = \{\mathbf{0}\} = V(R(2,0))$.

In the case of $R(2,1)$ we must consider the semigroup

$$\Gamma_{21} :=< (1,-2), (1,-1), (2,2), (2,1) > .$$

We determine its ideal

$$I_{\Gamma_{21}} =< x_4^4 - x_2^2x_3^3, x_4^3x_1 - x_2^3x_3^2, x_4^2x_1^2 - x_3x_2^4, x_4x_2 - x_1x_3, x_2^6 - x_1^4x_3, x_4x_1^3 - x_2^5 > .$$

Γ_{21} is Nakayama because there is no binomial $\pm(1 - \mathbf{X})^\beta$ in its generating set. Then, since there exists no binomial $\pm(x_4 - \mathbf{X}^\beta)$ in the generating set of Γ_{21}, we conclude $R(2,1) = \emptyset = V(R(2,1))$.

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

In order to determine the vertices of $R(3,0)$ and $R(3,1)$, we consider the systems

$$\begin{pmatrix} 1 & -2 & 2 \\ -2 & -1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = -\alpha \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \tag{1}$$

with $\alpha = 0, 1$.

For $\alpha = 0$, we obtain in the same way that in cases above, a particular $\mathbb{N}$-solution $\mathbf{s}' = (2, 6, 5)$. This means $(2, 6, 0, 5) \in R(3, 0)$. Then, if we identify $R(3, 0)$ with a subset of $\mathbb{N}^3$, we must determine the finite set

$$F' = \{\mathbf{s}'\} \cup \bigcup_{i=1}^{3} \bigcup_{\beta=0}^{s'_i - 1} V(R(3,0)(i,\beta)).$$

The set $R(3,0)(1,0)$ consists of the $\mathbb{N}$-solutions to the homogeneous system with matrix

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

Using Algorithm 2.1, we obtain $R(3,0)(1,0) = \{\mathbf{0}\} = V(R(3,0)(1,0))$.

By similar arguments, we obtain

$$R(3,0)(1,1) = \{\mathbf{0}\} = V(R(3,0)(1,1)), \text{and}$$

$$R(3,0)(2,\beta) = \emptyset = V(R(3,0)(2,\beta)), \text{for } \beta = 0, 1, 2, 3, 4, 5, \text{and}$$

$$R(3,0)(3,\beta) = \emptyset = V(R(3,0)(3,\beta)) \text{for } \beta = 0, 1, 2, 3, 4.$$

Then, $F' = \{\mathbf{s}'\}$. Therefore, $V(R(3,0)) = \{(2, 6, 0, 5)\}$.

If we set $\alpha = 1$ in (1), then $R(3, 1)$ corresponds with its general $\mathbb{N}$-solution. We consider the semigroup

$$\Gamma_{31} := < (1, -2), (-2, -1), (2, 2), (-1, 1) >,$$

and determine its ideal

$$I_{\Gamma_{31}} = < x_4 - x_1 x_2^5 x_3^4, x_1^2 x_2^6 x_3^5 - 1 > .$$

The binomial $x_4 - x_1 x_2^5 x_3^4 \in I_{\Gamma_{31}}$ indicates that $\mathbf{s}'' = (1, 5, 4)$ is a particular $\mathbb{N}$-solution.

As before, we compute the following sets

$$R(3,1)(1,0) = \emptyset = V(R(3,1)(1,0)), \text{and}$$

$$R(3,1)(2,\beta) = \emptyset = V(R(3,1)(2,\beta)), \text{for } \beta = 0, 1, 2, 3, 4, \text{and}$$

$$R(3,1)(3,\beta) = \emptyset = V(R(3,1)(3,\beta)) \text{for } \beta = 0, 1, 2, 3.$$

Then, we conclude $V(R(3,1)) = \{(1, 5, 1, 4)\}$.

Finally, we consider the systems

$$\begin{pmatrix} 1 & -2 & 1 \\ -2 & -1 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = -\alpha \begin{pmatrix} 2 \\ 2 \end{pmatrix},$$

where $\alpha = 0, 1, 2$.

Using similar arguments that in cases above, we obtain

$$R(4,0) = \{\mathbf{0}\} = V(R(4,0)), R(4,1) = \emptyset = V(R(4,1)), \text{and}, R(4,2) = \emptyset = V(R(4,2)).$$

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

3 Classical Linear Programming Methods

In this section we describe an alternative algorithm to 2.2. It is also based in 1.4, but it computes the particular solutions by means of linear programming methods.

First, consider the homogeneous case using the idea given in [18]. 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 $u_i \geq 0$ for all $i = 1, \ldots, q$, 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 $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{Q}^q$ is a basis of L. Let B the $n \times q$ matrix with row vectors $\mathbf{b}_i$. Denote $\mathbf{a}_1, \ldots, \mathbf{a}_q \in \mathbb{Q}^n$ the column vectors of B.

Notice that

$$\exists \mathbf{u} \in L - \{\mathbf{0}\} \text{ with } u_i \geq 0 \ \ \forall i = 1, \ldots, q$$

if and only if

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

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

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

Then, it is enough to apply the following result which is a constructive version for Farkas' lemma.

Proposition 3.1. : *(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 $\mathbf{v} \cdot \mathbf{a}_i \geq 0 \ \ \forall i = 2, \ldots, q$. 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 $\frac{a_{1i}}{|a_{1i}|}$ and 0 otherwise.

Assume that $q \geq 2$.

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

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

If there exists $\mathbf{w}' \in \mathbb{Q}^n$ such that $\mathbf{w}' \cdot \mathbf{a}'_1 > 0$ and $\mathbf{w}' \cdot \mathbf{a}'_i \geq 0$, $\forall i = 2, \ldots, q-1$, 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 will 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 $(*)$), $1 \leq r \leq q-1$, 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) $\mathbf{w}_j \cdot \mathbf{a}_1^{(j)} > 0$, $\mathbf{w}_j \cdot \mathbf{a}_i^{(j)} \geq 0$, $\mathbf{w}_j \cdot \mathbf{a}_{l_j}^{(j)} < 0$, $\forall i = 2, \ldots, l_j - 1$.

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

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

We will prove that

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

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

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),

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

We can use the induction hypothesis to write

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

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

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

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

The following algorithm satisfies Proposition 3.1.

Algorithm 3.2. : *Farkas*
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* $\mathbf{v} \cdot \mathbf{a}_i \geq 0$ *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* $\frac{a_{1i}}{|a_{1i}|}$ *and* 0 *otherwise and STOP.*

2. If $q \geq 2$*, determine if there exists* $\mathbf{w} \in \mathbb{Q}^n$ *such that* $\mathbf{w} \cdot \mathbf{a}_1 > 0$ *and* $\mathbf{w} \cdot \mathbf{a}_i \geq 0$ *for any* $i = 2, \ldots, q-1$*, by recursively using Algorithm 3.2.*
3. If there is no $\mathbf{w}$*, then output* $\emptyset$ *and STOP.*
4. Otherwise:

- *If* $\mathbf{w} \cdot \mathbf{a}_q \geq 0$*, output* $\mathbf{v} = \mathbf{w}$ *and STOP.*

- *Otherwise, continue.*

5. Let

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

Determine if there exists $\mathbf{w}' \in \mathbb{Q}^n$ *such that* $\mathbf{w}' \cdot \mathbf{a}'_1 > 0$ *and* $\mathbf{w}' \cdot \mathbf{a}'_i \geq 0$*,* $\forall i = 2, \ldots, q-1$*, by Algorithm 3.2.*
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 $\emptyset$*.*

Remarks 3.3. :

1. *The algorithm above allows us to determine if there exists* $\mu_i \leq 0$ *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 3.4 below.*

3. *If* $S \subset \mathbb{N}^2$*, since* $S(i, \alpha) \subset \mathbb{N}$*, Remark 1.6 and the remark above allow us to compute* VS *using Lemma 1.4.*

Algorithm 3.4. : *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 1.6.*
2. If $q \geq 2$*, 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* $\mathbf{v} \cdot \mathbf{a}_j \geq 0$*, for any* $j \neq i$ *and* $1 \leq j \leq q$*, by Algorithm 3.2.*

- *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 1.4 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' 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 of matrix $(-\mathbf{c}|M')$, 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 the $n \times q$ matrix with row vectors $\mathbf{b}_i$. Denote $\mathbf{a}_1, \ldots, \mathbf{a}_q \in \mathbb{Q}^n$ the column vectors of B.

Notice that

$$\exists (1, \mathbf{u}) \in L \ \text{ with } \ \mathbf{u} \in \mathbb{N}^{q-1}$$

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 $\mathbf{u}$ and $\mathbf{v}$ 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 W the $\mathbb{Q}$-vector space generated by $\mathbf{a}_2, \ldots, \mathbf{a}_q$. If $\mathbf{a}_1 \notin W$, take $\hat{\mathbf{a}}_1$ 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 $\mu_i \leq 0$, then there is no $\mathbf{v}$ (Remark 3.3.1.).
- Otherwise, take a linear combination of type $\mathbf{a}_1 = \sum_{i=2}^{q} \mu_i \mathbf{a}_i$. Let A 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 $D = (\mathbf{s}_1 | \ldots | \mathbf{s}_h)$ and

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

Then, we get the following result.

Proposition 3.5. : *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* $m_1 w_1 + \cdots + m_h w_h = 1$.

In that case, it is enough to take $\mathbf{v}$ *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,

$$m_1 w_1 + \cdots + m_h w_h = 1.$$

$\boxed{2 \Rightarrow 1}$ Let $\mathbf{z} = D\mathbf{w}$. Since $\mathbf{z} \in S_1 \subset L_1$, we deduce that the ranks of A and $(A|\mathbf{z})$ 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

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

□

Remarks 3.6. :

1. *Notice that the proof does not use the hypothesis $\mu_i \geq 0$ for at least one i, although this case is solved by Farkas's lemma.*

2. *From 3.3.3 we can determine the condition 1 in Proposition 3.5 for $q = 3$. Now, applying 1.4 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 [4].*

Problem is solved by the following algorithm.

Algorithm 3.7. :
Input: Vectors $\mathbf{a}_1, \ldots, \mathbf{a}_q \in \mathbb{Q}^n$, $q \geq 2$.
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 3.2:

- *If $\mathbf{a}_1 = \sum_{i=2}^{r} \mu_i \mathbf{a}_i$ with $\mu_i \leq 0$ (it is equivalent to output $\emptyset$), 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 3.9.
6. Let $(m_1, \ldots, m_h) = (\mu_2, \ldots, \mu_q)D$, where $D = (\mathbf{s}_1 | \ldots | \mathbf{s}_h)$.

- *If there exists $\mathbf{w} \in \mathbb{N}^h$ such that $m_1\mathbf{w}_1 + \cdots + m_h\mathbf{w}_h = 1$, output $\mathbf{v}$ a particular solution of $A\mathbf{x} = \mathbf{z}$ with $\mathbf{z} = D\mathbf{w}$ and STOP. (See Remark 3.6.2)*
- *Otherwise, output $\emptyset$.*

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

Algorithm 3.8. : *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 3.4.
2. Otherwise, continue.
3. If $q = 2$, use remark 1.6.
4. If $q \geq 3$, 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 3.7

- *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*

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

and STOP.

- *Otherwise, output* $\emptyset$.

We can now describe a second algorithm satisfying Proposition 1.7.

Algorithm 3.9. : *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 1.6 and STOP.*
2. If $q \geq 2$*, determine whether or not* $R = \emptyset$ *or* $\{\mathbf{0}\}$ *using Algorithm 3.8.*
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 3.9.*
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 3.10. : *Notice that there is not circularity between algorithms above because 3.9 computes* $S \subset \mathbb{N}^2$ *by only Farkas' lemma (see Remark 3.3.2).*

Example 3.11. : We consider the same system that in 2.3.

$$\begin{cases} x_1 - 2x_2 + x_3 + 2x_4 &= 0 \\ -2x_1 - x_2 - x_3 + 2x_4 &= 0 \end{cases}$$

and M the matrix

$$M := \begin{pmatrix} 1 & -2 & 1 & 2 \\ -2 & -1 & -1 & 2 \end{pmatrix}$$

We are going to compute VR using Algorithm 3.9. We need to determine whether or not $R = \emptyset$ or $\{\mathbf{0}\}$ using Algorithm 3.8. Since the system is homogeneous we will use Algorithm 3.4.

A basis of the $\mathbb{Q}$-vector space given by $M\mathbf{x} = \mathbf{0}$ is

$$\mathbf{b}_1 = (4, 0, -6, 1), \text{and } \mathbf{b}_2 = (-3, 1, 5, 0).$$

Let $\mathbf{a}_1 = (4, -3)$, $\mathbf{a}_2 = (0, 1)$, $\mathbf{a}_3 = (-6, 5)$, and $\mathbf{a}_4 = (1, 0)$.

Using Algorithm 3.4, we find $\mathbf{v} = (5/18, 1/3)$ such that $\mathbf{v} \cdot \mathbf{a}_1 > 0$, and $\mathbf{v} \cdot \mathbf{a}_i \geq 0$, for $i = 2, 3, 4$.

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

We need to determine the finite set

$$F = \{\mathbf{s}\} \cup \bigcup_{i=1}^{4} \bigcup_{\alpha=0}^{s_i-1} V(R(i, \alpha)).$$

Notice that $R(1, \alpha)$ corresponds to the general $\mathbb{N}$-solution of the system

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

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

We identify $R(1,0)$ with a set in $\mathbb{N}^3$ and compute $V(R(1,0))$ using the finite set

$$F' = \{\mathbf{s}'\} \cup \bigcup_{i=1}^{3} \bigcup_{\beta=0}^{s'_i-1} V(R(1,0)(i,\beta)).$$

In order to determine whether or not $R(1,0)(1,\beta) = \emptyset$ or $\{\mathbf{0}\}$, we consider the new systems

$$\begin{pmatrix} 1 & 2 \\ -1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = -\beta \begin{pmatrix} -2 \\ -1 \end{pmatrix}. \tag{3}$$

As before we obtain

$$R(1,0)(1,0) = \{\mathbf{0}\}.$$

Set $\beta = 1$ in (3). Using Algorithm 3.8, 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 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 3.7 to show that there is not circularity between the algorithms used in this method (Remark 3.10).

With the notation of Algorithm 3.7, we can consider $\mathbf{a}'_1 = \mu_2\mathbf{a}'_2 + \mu_3\mathbf{a}'_3$, with $\mu_2 = 2$ and $\mu_3 = 0$. Let

$$A = \begin{pmatrix} 1 \\ 3/2 \end{pmatrix}.$$

We consider the implicit equations, $C\mathbf{x} = \mathbf{0}$, of the $\mathbb{Q}$-vector space generated by the column vector of A. Set $C = (-3 \;\; 2)$.

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 3.9, which calls to Algorithm 3.8, and this, to Algorithm 3.4. As before, we compute an element $\mathbf{s}'' = (2, 3) \in S_1$. Then, $VS_1 = VF''$ where

$$F'' = \{\mathbf{s}''\} \cup \bigcup_{i=1}^{2} \bigcup_{\gamma=0}^{s''_i-1} V(S_1(i,\gamma)).$$

To determine the vertices of $S_1(1,\gamma)$, we consider the systems

$$2x_1 = -\gamma(-3).$$

If $\gamma = 0$, then $x_1 = 0$ is the unique $\mathbb{N}$-solution. Therefore,

$$S_1(1,0) = \{(0,0)\} = V(S_1(1,0)).$$

If $\gamma = 1$, it is clear that there is no $\mathbb{N}$-solution. Therefore,

$$S_1(1,1) = \emptyset = V(S_1(1,1)).$$

The sets $S_1(2,\gamma)$ are obtained from the systems

$$-3x_1 = -\gamma 2.$$

If $\gamma = 0$, we obtain

$$S_1(2,0) = \{(0,0)\} = V(S_1(2,0)).$$

If $\gamma = 1$, there is no $\mathbb{N}$-solution. Thus,

$$S_1(2,1) = \emptyset = V(S_1(2,1)).$$

For $\gamma = 2$ the same situation is obtained. Therefore,

$$S_1(2,2) = \emptyset = V(S_1(2,2)).$$

Then $F'' = \{(2,3),(0,0)\}$, and $VS_1 = VF'' = \{(2,3)\}$.

With the notation in Algorithm 3.7, we have $\mathbf{s}_1 = (2,3)$, $D = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$, and $m = (2\ \ 0)D = 4$. It is clear that there exists no $w \in \mathbb{N}$ such that $wm = 1$. Then (Proposition 3.5),

$$R(1,0)(1,1) = \emptyset = V(R(1,0)(1,1)).$$

By similar arguments that above, it is obtained

$$R(1,0)(1,\beta) = \emptyset, \text{for } \ \beta = 2,3.$$

$$R(1,0)(2,0) = \{\mathbf{0}\}.$$
$$R(1,0)(2,1) = \emptyset.$$
$$R(1,0)(3,0) = \{\mathbf{0}\}.$$
$$R(1,0)(3,\beta) = \emptyset, \text{ for } \ \beta = 1,2.$$

Then, we obtain $V(R(1,0)) = VF'$ with

$$F' = \{(0,0,0,0),(0,4,2,3)\}.$$

We conclude

$$V(R(1,0)) = \{(0,4,2,3)\}.$$

Now, we set $\alpha = 1$ in (2). We must determine whether or not $R(1,1) = \emptyset$ using Algorithm 3.8. 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 3.7, 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

$$C' = (-5\ \ 1\ \ 6).$$

Let

$$S_1' = \{\mathbf{s} \in \mathbb{N}^3 \mid C'\mathbf{s} = \mathbf{0}\}.$$

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 $\mathbf{m}'\mathbf{w}' = 1$. We take for example $\mathbf{w}' = (0,0,1,0,0,0)$. Therefore, there exists $\mathbf{v}''$. (Proposition 3.5.)

We take $\mathbf{v}'' = (5,4)$ a particular solution of $A'\mathbf{x} = D'\mathbf{w}'$. From

$$(\mathbf{v}' \cdot \mathbf{a}''_2, \mathbf{v}' \cdot \mathbf{a}''_3, \mathbf{v}' \cdot \mathbf{a}''_4) = (5,1,4),$$

we obtain

$$(1,5,1,4) \in R(1,1).$$

As in previous cases, we compute the sets

$$R(1,1)(1,\delta) = \emptyset, \delta = 0,\ldots,4,$$

$$R(1,1)(2,0) = \emptyset,$$

$$R(1,1)(3,\delta) = \emptyset, \delta = 0,\ldots,3.$$

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

$$\begin{array}{l} V(R(2,5)) = \{(1,5,1,4)\} \\ V(R(4,0)) = \{\mathbf{0}\} \\ V(R(4,1)) = V(R(4,2)) = \emptyset \\ V(R(4,3)) = \{(0,4,2,3)\} \\ V(R(4,4)) = \{(1,5,1,4)\}. \end{array}$$

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.

4 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

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

and let

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

Remarks 4.1. :

1. *If* $\boldsymbol{\gamma} \in R$, *then*

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

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

Theorem 4.2. : *With assumptions and notations as above, if*

$$VR = \{\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_r\},$$

then

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

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

Proof. The formula $(*)$ is clear by 4.1. Now, since it is possible to compute VR and a generating set of S (Proposition 1.7), we get an algorithm computing all the elements in R. □

Let $M' = (-\mathbf{c}|M)$, and

$$S' := \{\mathbf{s}' \in \mathbb{N}^{q+1} \mid M'\mathbf{s}' = \mathbf{0}\}.$$

Denote

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

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 4.2.

Algorithm 4.3. : *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* $M' = (-\mathbf{c}|M)$, *and* $S' := \{\mathbf{s}' \in \mathbb{N}^{q+1} \mid M'\mathbf{s}' = \mathbf{0}\}$.
2. *Compute* VS' *using Algorithm 2.2 or Algorithm 3.9.*
3. *Output* $VS = (VS')_0$ *and* $VR = (VS')_1$ *and STOP.*

Remark 4.4. : *Solving general systems of linear equations in nonnegative 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.*

We will now give some examples.

Example 4.5. : Consider the following diophantine equation

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

We are going to compute the set

$$R = \{\mathbf{s} \in \mathbb{N}^4 \mid M\mathbf{s} = 12\},$$

where

$$M = (1 \quad -3 \quad 2 \quad -5).$$

Following Algorithm 4.3, we construct the matrix

$$M' = (-12 \quad 1 \quad -3 \quad 2 \quad -5).$$

We compute VS' where

$$S' = \{\mathbf{s}' \in \mathbb{N}^5 \mid M'\mathbf{s}' = \mathbf{0}\}.$$

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

Then, the semigroup S of the $\mathbb{N}$-solutions to the homogeneous equation

$$x_1 - 3x_2 + 2x_3 - 5x_4 = 0,$$

is generated by

$$\begin{aligned} VS = (VS')_0 = \quad &\{(1,1,1,0),(0,1,4,1),(0,0,5,2),(0,2,3,0),\\ &(1,0,2,1),(5,0,0,1),(3,1,0,0),(3,0,1,1)\}, \end{aligned}$$

and

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

Therefore,

$$\begin{aligned} R = \quad &[(0,0,6,0)+S]\bigcup[(12,0,0,0)+S]\bigcup[(10,0,1,0)+S]\bigcup\\ &[(8,0,2,0)+S]\bigcup[(6,0,3,0)+S]\bigcup[(4,0,4,0)+S]\bigcup[(2,0,5,0)+S]. \end{aligned}$$

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

$$\begin{aligned} &[[0,0,6,0],[12,0,0,0],[10,0,1,0],[8,0,2,0],[6,0,3,0],[4,0,4,0],\\ &[2,0,5,0]],[[3,1,0,0],[0,2,3,0],[0,0,5,2],[0,1,4,1],\\ &[1,1,1,0],[1,0,2,1],[3,0,1,1],[5,0,0,1]] \end{aligned}$$

Example 4.6. : Consider the system

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

Now,

$$VS' = \{(1,1,1,0,0),(0,7,0,1,2),(0,5,5,0,3),(10,21,0,3,0),(5,14,0,2,1)\}.$$

Then, the semigroup S of $\mathbb{N}-$solutions of the associate homogeneous diophantine equation system is:

$$VS = (VS')_0 = \{(7,0,1,2),(5,5,0,3)\}.$$

And the $\mathbb{N}-$solutions of the non homogeneous equation system are:

$$VR = (VS')_1 = \{(1,1,0,0)\}$$

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

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

Then, the semigroup S of $\mathbb{N}-$solutions of the associated homogeneous system is:

$$S =< (7,0,1,2),(5,5,0,3) > .$$

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

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

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

Therefore, Algorithm 4.3 has a unquestionably better computational behaviour if one uses Semigroup Ideals and Gröbner Bases (Algorithm 2.2), than if one instead uses Classical Linear Programming (Algorithm 3.9).

We can conclude that the *Hilbert basis* of a linear diophantine system can be computed by an algorithm based on Gröbner Bases, which is considerably faster than the traditional Integer Programming methods.

References

[1] BOROSH, I. (1976). A sharp bound for positive solutions of homogeneous linear diophantine equations, *Proc. Amer. Math. Soc.* **60**, 19-21.

[2] BOROSH, I.; TREYBIG, L.B. (1976). Bounds on positive integer solutions of linear Diophantine equations, *Proc. Amer. Math. Soc.* **55**, 299-304.

[3] BRIALES, E.; CAMPILLO, A.; MARIJUAN, C.; PISON, P. (1998). Minimal systems of generators for ideals of semigroups. *Journal of Pure and Applied Algebra*, **124**, 7-30.

[4] CLAUSEN, M.; FORTENBACHER, A. (1989). Efficient solution of linear Diophantine equations. *Journal of Symbolic Computation*, **8**, 201-216.

[5] COHEN H. (1993). *A Course in Computational Algebraic Number Theory* GTM **138**, Springer-Verlag.

[6] CONTI, P.; TRAVERSO, C. (1991). Buchberger algorithm and integer programming, *Proceedings AAECC-9* (New Orleans), Springer LNCS 539, 130-139.

[7] COTEJEAN, E.; DEVIE, H. (1994). An efficient Algorithm for solving systems of diophantine equations. *Information and Computation*, 113(1), 143-172.

[8] COX, D.; LITTLE, J.; O'SHEA, D. (1992). *Ideals, Varietes, and Algorithms*, Undergraduate Texts in Mathematics, Springer-Verlag.

[9] DI BIASE, F.; URBANKE, R. (1995). An Algorithm to Calculate the Kernel of Certain Polynomial Ring Homomorphisms, *Experimental Mathematics*, Vol. 4,3, 227-234.

[10] DOMENJOUD, E. (1991). *Outils pour la Déduction Automatique dans les Théories Associatives-Commutatives*. Thése de doctorat d'Université, Université de Nancy I.

[11] GEDDES, K.O.; CZAPOR, S.R.; LABAHN, G. (1992). *Algorithms for Computer Algebra*. Kluwer Academic Publishers.

[12] HOSTEN, S.; STURMFELS, B. (1995). GRIN: An implementation of Gröbner Bases for integer programming. In E. Balas and J. Clausen, editors, *Integer Programming and Combinatorial Optimization*, LNCS **920**, Springer-Verlag, 267-276.

[13] KNUTH, D. (1981). *The art of computer programming*, Vol 2. Addison-Wesley Publishing Company.

[14] MOULINET-OSSOLA, C. (1995). *Algorithmique des Réseaux et des Systèmes Diophantiens Linéaires.* Thése de doctorat. Université de Nice Sophia-Antipolis.

[15] PAPADIMITRIOU, C.H. (1981). On the complexity of integer programming, *J. Assoc. Comput. Mach.* **28**, 765-768.

[16] POTTIER, L. (1991). 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, 162-173.

[17] POTTIER, L. (1991b). Sub-groups of $\mathbb{Z}^n$, Standard Basis, and Linear Diophantine Systems, *INRIA Research Report 1510.*

[18] ROSALES, J.C. (1995) On finitely generated submonoids of $\mathbb{N}^k$, *Semigroup Forum*, Vol.**50**, 251-262.

[19] ROSALES, J.C.; GARCIA-SANCHEZ, P.A. (1998) Non negative elements of subgroups of $\mathbb{Z}^n$, *Linear Algebra and its Applications*, **270**, 351-357.

[20] STANLEY, R. (1996). *Combinatorics and commutative algebra*, 2nd ed. Progress in Mathematics Vol.**41**, Boston Basel Berlin, Birkhäuser.

[21] SHRIJVER, A. (1986) *Theory of Linear and Integer Programming* Wiley-Interscience series, John Wiley - Sons.

[22] STURMFELS, B. (1995) *Gröbner Bases and Convex Polytopes*, AMS University Lectures Series, Vol. 8.

[23] TOMAS, A.P. (1997). *On Solving Linear Diophantine Constraints*, Tese de Doutoramento, Faculdade de Ciencias da Universidada do Porto.

[24] VIGNERON, A. (1999). Semigroup Ideals and Linear Diophantine Equations, Linear Algebra and its Applications **295**, 133-144.

Pilar Pisón Casares,
Dpto de Álgebra
Universidad de Sevilla.
Facultad de Matemáticas.
Apartado 1160, 41080 Sevilla (Spain)
e-mail: pilar@algebra.us.es

Alberto Vigneron Tenorio,
Dpto. de Matemáticas.
Universidad de Cádiz.
E.U.E. Empresariales.
C/ Por-vera 54
11403 Jerez de la Frontera (Spain)
e-mail: vigneron@cica.es