On the computation of Graver test sets

Raymond Hemmecke*
Department of Mathematics, Gerhard-Mercator University Duisburg
Lotharstr. 65, D-47048 Duisburg, Germany

January 11, 1999

Abstract

We present a unified approach to Graver test sets in LP and IP that is inspired by the critical pair/completion procedure arising in computational algebra. Our algorithms for computing test sets entirely work in the kernel of the problem matrix.

AMS classification: 90 C 10

Keywords: Integer programming, Graver test sets

1 Introduction

Consider the problem $\min\{c^\intercal z : Az = b, z \in \mathbb{X}_{\geq 0}\}$, where $\mathbb{X} = \mathbb{Z}^{n_z} \times \mathbb{R}^{n_q}$ and $\mathbb{X}_{\geq 0}$ denotes the corresponding non-negative orthant. Moreover, $A \in \mathbb{Q}^{d\times n}$, $c \in \mathbb{R}^n$ and $b \in \mathbb{Q}^d$ with $n := n_z + n_q$ are given. Here $\mathbb{Z}, \mathbb{Q}, \mathbb{R}$ denote the integers, rationals, and reals, respectively.

In the present paper we address solving the above problem by test set methods.

Definition 1 *(Test set)*

A finite set $T \subseteq \mathbb{Q}^n$ is called a **particular test set** *for* $\min\{c^\intercal z : Az = b, z \in \mathbb{X}_{\geq 0}\}$ *if*

1. *$c^\intercal t > 0$ for all $t \in T$, and*
2. *for every non-optimal feasible solution $z_0 \in \mathbb{X}_{\geq 0}$ to $Az = b$, there exists a vector $t \in T$ and $\alpha > 0$ such that $z_0 - \alpha t$ is feasible.*

A finite set is called a **test set** *for* $\min\{c^\intercal z : Az = b, z \in \mathbb{X}_{\geq 0}\}$ *if it contains a particular test set for every $b \in \mathbb{Q}^d$.*

A finite set is called a **universal test set** *for* $\min\{c^\intercal z : Az = b, z \in \mathbb{X}_{\geq 0}\}$ *if it contains a particular test set for every $c \in \mathbb{R}^n$ and every $b \in \mathbb{Q}^d$.*

Note that a test set has to give improving directions for any right-hand-side b and hence cannot depend on the specific choice b. The only and usually already hard step specific to b is to find some feasible solution of the given problem.

With a test set T for $\min\{c^\intercal z : Az = b, z \in \mathbb{X}_{\geq 0}\}$ and a feasible solution z_0 for specific b it seems straight-forward to solve the problem by improving z_0 via elements of T as long as possible. Indeed, given solvability of the problem, this process always terminates with an optimal solution in the IP-case. In the LP-case, however, we will see that some caution is appropriate to avoid zig-zagging (even to non-optimal points).

*This work was supported by the Schwerpunktprogramm "Echtzeit-Optimierung großer Systeme" of the Deutsche Forschungsgemeinschaft.

In 1975 Graver [7] introduced universal test sets in LP and IP. However, he provided no algorithm to compute them.

We pick up this notion and present a property that is inherent to both LP- and IP-Graver test sets and implies that they are universal. Moreover, the Graver test set computations in LP and IP follow the same algorithmic scheme. In contrast to the algorithms presented in Cornuejols et al. [5] and Urbaniak [15] for the IP-case, which nicely fit into the presented algorithmic scheme, our algorithms work entirely in the kernel of the given matrix.

We conclude the paper with a section on optimizing with the help of Graver test sets. Here some care has to be taken in the LP-case, in particular.

Before starting our analysis let us add a few bibliographical remarks.

Only in 1991, Conti and Traverso [4] succeeded in actually computing a test set in IP for fixed c by using Buchberger's Gröbner basis algorithm [3]. The algorithm of Conti and Traverso, however, had a large computational overhead for this application in integer programming and lots of work has been done to improve its performance. By fundamental algebraic relations the problem was reduced to saturating a certain ideal with respect to all occurring variables. Later it was found that already saturation with respect to a subset of variables is sufficient. For more details see for example Sturmfels [12].

Buchberger's algorithm being a field of active research in computer algebra, test set computation has also benefited from efficient implementations and improvements achieved in that area (see Bigatti et al. [1] or Hosten and Sturmfels [8]).

A geometrical interpretation of the Conti-Traverso algorithm was given by Thomas [14]. This has initiated further work on geometric algorithms for the computation of test sets, see Cornuejols et al. [5] and Urbaniak et al. [16].

Structural similarities of Graver test sets in LP and IP were elaborated in Sturmfels and Thomas [13].

The only test sets for which no Buchberger-like algorithm is known so far are Scarf's neighboring systems [9]. There is a homotopy algorithm by Scarf himself computing neighboring systems, which, however, reaches its computational limits already for small examples [10].

In [17], Weismantel gives a nice survey on all currently known test sets in IP together with their computation.

2 Graver test sets

2.1 Main idea

Definition 2 *We define the relation $\sqsubseteq$ on $\mathbb{R}^n$ by $u \sqsubseteq v$ if $u^{(j)}v^{(j)} \geq 0$ and $|u^{(j)}| \leq |v^{(j)}|$ for all components, that is, u lies in the same orthant as v and its components are not greater in absolute value than the components of v.*

Definition 3 *(Positive sum property)*

A finite set $G \subseteq \mathbb{R}^n$ has the **positive sum property** *with respect to $S \subseteq \mathbb{X} \subseteq \mathbb{R}^n$ if any non-zero $v \in S$ can be written as a positive linear combination*

$$v = \sum \alpha_i g_i$$

with $g_i \in G$, $\alpha_i > 0$, $\alpha_i g_i \in \mathbb{X}$, and $\alpha_i g_i \sqsubseteq v$ for all i.

Definition 4 *(Kernel of a matrix)*

Given the matrix $A \in \mathbb{Q}^{d \times n}$ and $\mathbb{X} \subseteq \mathbb{R}^n$. Then

$$\ker_{\mathbb{X}}(A) := \{v \in \mathbb{X} : Av = 0\}$$

denotes the **kernel** *of A with respect to $\mathbb{X}$.*

If $\mathbb{X}$ is clear from the context we write $\ker(A)$ instead of $\ker_{\mathbb{X}}(A)$.

Lemma 1 *(Positive sum property implies universal test set property)*

If $G \subseteq \ker_{\mathbb{X}}(A)$ has the positive sum property with respect to $\ker_{\mathbb{X}}(A)$ then G is a universal test set for the problem $\min\{c^\intercal z : Az = b, z \in \mathbb{X}_{\geq 0}\}$.

Proof. Given some arbitrary but fixed values for b and c and some feasible solution z_0 to the problem $\min\{c^\intercal z : Az = b, z \in \mathbb{X}_{\geq 0}\}$.

We have to prove that if z_0 is not optimal then there exist $v \in G$ and $\alpha > 0$ such that $z_0 - \alpha v$ is a better feasible solution than z_0.

Suppose that z_0 is not optimal. By z_1 denote another feasible solution with smaller cost function value. Then by our assumption we can write $z_0 - z_1$ as a positive linear combination

$$z_0 - z_1 = \sum \alpha_i g_i,$$

where $g_i \in G$, $\alpha_i > 0$, $\alpha_i g_i \in \mathbb{X}$, and $\alpha_i g_i \sqsubseteq z_0 - z_1$ for all i.

Since $c^\intercal(z_0 - z_1) > 0$ we have $c^\intercal(\alpha_j g_j) > 0$ for at least one g_j. We now claim that $z_0 - \alpha_j g_j$ is feasible and has a better cost function value.

Clearly, $A(z_0 - \alpha_j g_j) = Az_0 - \alpha_j A(g_j) = b - 0 = b$ and $c^\intercal z_0 > c^\intercal(z_0 - \alpha_j g_j)$, since $g_j \in \ker(A)$ and $c^\intercal(\alpha_j g_j) > 0$. So it remains to prove $z_0 - \alpha_j g_j \geq 0$.

We prove this for each component separately. Since $\alpha_j g_j \sqsubseteq z_0 - z_1$ we have

$$\left|(z_0 - z_1)^{(k)}\right| \geq \left|\alpha_j g_j^{(k)}\right|$$

and

$$(z_0 - z_1)^{(k)} g_j^{(k)} \geq 0.$$

Now we distinguish three cases.

1. If $(z_0 - z_1)^{(k)} = 0$ then $(\alpha_j g_j)^{(k)} = 0$ and
$$(z_0 - \alpha_j g_j)^{(k)} = z_0^{(k)} \geq 0.$$
2. If $(z_0 - z_1)^{(k)} < 0$ then $(\alpha_j g_j)^{(k)} \leq 0$. Hence
$$(z_0 - \alpha_j g_j)^{(k)} = z_0^{(k)} - (\alpha_j g_j)^{(k)} \geq z_0^{(k)} \geq 0.$$
3. If $(z_0 - z_1)^{(k)} > 0$ then $(\alpha_j g_j)^{(k)} \geq 0$ and $(z_0 - z_1)^{(k)} \geq (\alpha_j g_j)^{(k)}$. Hence
$$z_0^{(k)} - (\alpha_j g_j)^{(k)} \geq z_1^{(k)} \geq 0.$$

So in any case $(z_0 - \alpha_j g_j)^{(k)} \geq 0$. □

In the sequel we introduce Graver test sets in LP and IP, and we show that they have the positive sum property with respect to $\ker(A)$. Moreover, we give geometric algorithms to compute them which work entirely in $\ker(A)$.

2.2 IP-Graver test sets

The notion of a Graver test set in IP is strongly related to the notion of a Hilbert basis, which we will recall now. For further details we refer to Schrijver [11] and Weismantel [17].

Definition 5 *(Hilbert basis)*
Let C be a rational cone. A finite set $H = \{h_1, \ldots, h_t\} \subseteq C \cap \mathbb{Z}^n$ is a **Hilbert basis** *of C if every $z \in C \cap \mathbb{Z}^n$ has a representation of the form*

$$z = \sum_{i=1}^{t} \lambda_i h_i,$$

with non-negative integral multipliers $\lambda_1, \ldots, \lambda_t$.

Theorem 1 *(Existence and uniqueness of Hilbert bases)*
Every pointed rational cone possesses a unique Hilbert basis that is minimal with respect to inclusion.

Definition 6 *Let $\mathbb{O}_j$ be the j^{th} orthant of $\mathbb{Z}^n$ and $H_j(A)$ the unique minimal Hilbert basis of $\ker(A) \cap \mathbb{O}_j$. Then we define $G_{IP}(A) := \bigcup H_j(A)$ to be the* **IP-Graver test set** *(or Graver basis) of A.*

Lemma 2 *$G_{IP}(A)$ has finite cardinality.*

By construction the elements of $G_{IP}(A)$ are minimal with respect to the partial ordering $\sqsubseteq$ on $\mathbb{Z}^n$, where $u \sqsubseteq v$ if $u^+ \leq v^+$ and $u^- \leq v^-$. Moreover, if $v \in G_{IP}(A)$ then also $-v \in G_{IP}(A)$.

Lemma 3 *$G_{IP}(A)$ has the positive sum property with respect to $\ker(A)$.*

Proof. Take any non-zero element $z \in \ker(A)$. Then z belongs to some orthant $\mathbb{O}_j$ and hence can be written as a positive linear combination of elements of the Hilbert basis $H_j \subseteq G_{IP}(A)$ of $\ker(A) \cap \mathbb{O}_j$. $\square$

Corollary 1 *$G_{IP}(A)$ is a universal test set for $\min\{c^\intercal z : Az = b, z \in \mathbb{Z}^n_{\geq 0}\}$.*

2.3 LP-Graver test sets

Definition 7 *For $v \in \mathbb{R}^n$ the set $\mathrm{supp}(v) := \{j : v^{(j)} \neq 0\}$ is called the* **support** *of v.*

Definition 8 *(Circuits of a matrix)*
A **circuit** *of a matrix A is a vector $q \in \ker(A) \cap \mathbb{Z}^n$ of minimal support in $\ker(A)$ (with respect to inclusion) whose components have greatest common divisor 1. Denote the set of all circuits of A by $G_{LP}(A)$.*

Indeed, if A is rational then for every minimal support in $\ker(A)$ there is always a rational (hence also integer) vector.

Lemma 4 *If v is a circuit of A and $w \in \ker(A)$ satisfies $\mathrm{supp}(w) \subseteq \mathrm{supp}(v)$ then $w = \alpha v$ for some $\alpha \in \mathbb{R}$. In particular, we have $w = \pm v$ if v and w are circuits with the same support.*

Proof. This proof can be found in Graver [7]. $\square$

Corollary 2 *$G_{LP}(A)$ has finite cardinality.*

Lemma 5 *$G_{LP}(A) \subseteq G_{IP}(A)$.*

Proof. Take any $v \in G_{LP}(A)$ then by construction $v \in \ker_{\mathbb{Z}}(A)$. Suppose v is not minimal in $\ker_{\mathbb{Z}}(A)$ with respect to $\sqsubseteq$. Then there is some non-zero $w \in \ker_{\mathbb{Z}}(A)$ with $w \sqsubseteq v$ which in particular requires $\mathrm{supp}(w) \subseteq \mathrm{supp}(v)$. By Lemma 4 we get $w = \alpha v$ for some $\alpha \in \mathbb{R}$. Since $w \in \mathbb{Z}^n$ and $w \sqsubseteq v$ this implies $w = v$. Hence v is minimal with respect to $\sqsubseteq$ and so $v \in G_{IP}(A)$. $\square$

Lemma 6 *$G_{LP}(A)$ has the positive sum property with respect to $\ker(A)$.*

Proof. This proof can be found in Graver [7]. $\square$

Corollary 3 *$G_{LP}(A)$ is a universal test set for the problem $\min\{c^\intercal z : Az = b, z \in \mathbb{R}^n_{\geq 0}\}$.*

3 Computation of Graver test sets

3.1 Algorithmic pattern

The algorithms given below differ only in the definitions of the function normalForm(s, G) and the set S-vectors(f, g) which have to be specified according to the test set one wants to compute.

Normal form algorithm:

Input: vector s, set G of vectors

Output: normal form of s with respect to G

while there is some $g \in G$ such that reducible?(s, g) = TRUE do
    $s :=$ reduce(s, g)
return s

The reduction involved in the above algorithm has to be defined according to the context (LP or IP). The set S-vectors(f, g) that will arise below then corresponds to "critical situations" of this reduction.

Our algorithms require a generating set of $\ker(A)$ over $\mathbb{R}$ respectively over $\mathbb{Z}$ as input. The first is easily obtained by Gaussian elimination resulting in a row-echolon form of A. The second can be obtained from the Hermitian normal form of A. This was pointed out in Sturmfels [12] and is given in more detail in Schrijver [11].

3.2 IP-case

The algorithm below is the same as Algorithm 2.1.7 in Urbaniak [15], however, instead of $\{\pm e_1, \ldots, \pm e_n\}$ we use an integer generating set of $\ker(A)$ as input and so our algorithm computes $G_{IP}(A)$ entirely in $\ker(A) \subseteq \mathbb{Z}^n$ avoiding the massive computational overhead.

As we will see, Algorithm 2.1.7 in Urbaniak [15] essentially computes the IP-Graver test set for $(A|-E)$ by plugging $\{\pm(e_1, Ae_1), \ldots, \pm(e_n, Ae_n)\}$ into the algorithm below. Thereafter all minimal vectors (v, Av) satisfying $Av = 0$ are collected. However, $G_{IP}(A)$ is usually much smaller than $G_{IP}(A| - E)$.

To reduce this computational overhead, Cornuejols et al. [5] presented a row decomposition algorithm which successively computes $G_{IP}(A_k)$, where A_k consists of the first k rows of A and finally arrives at $G_{IP}(A)$. Still a computational overhead occurs in each step. Numerical experiments, however, show that these row-wise overheads are in sum much smaller than the overhead for Algorithm 2.1.7 in Urbaniak [15], where the whole matrix A was used from the beginning.

Reduction and S-vectors

reducible?(s,g):

Input: vectors $s, g \in \mathbb{Z}^n$

Output: TRUE, if s is reducible by g, otherwise FALSE

if $g \sqsubseteq s$ then return TRUE else return FALSE

reduce(s,g):

Input: vectors $s, g \in \mathbb{Z}^n$, such that reducible?(s, g) = TRUE

Output: s reduced by g

return $s - g$

If reducible(s, g) = TRUE then s, g, and reduce$(s, g) = s - g$ all are in the same orthant of $\mathbb{Z}^n$. Since $||s - g||_1 < ||s||_1$ we conclude that normalForm(s, G) always terminates.

Finally, we define S-vectors$(v, w) := \{v + w\}$.

Algorithm

IP-Graver test set algorithm:

Input: $F = \bigcup_{f \in F(A)} \{f, -f\}$, where $F(A)$ is a generating set over $\mathbb{Z}$ of $\ker(A)$

Output: set G containing the IP-Graver test set $G_{IP}(A)$ associated with A

$G := F$
$C := \bigcup_{f,g \in G}$ S-vectors(f,g)
while $C \neq \emptyset$ do
  $s :=$ an element in C
  $C := C - \{s\}$
  $f :=$ normalForm(s, G)
  if $f \neq 0$ then
    $C := C \cup \bigcup_{g \in G}$ S-vectors(f,g)
    $G := G \cup \{f\}$
return G.

The set $G_{IP}(A)$ is the set of all elements z in G which are irreducible with respect to $G \setminus \{z\}$. For the termination proof we need the (Gordan-)Dickson Lemma (see for example section 4.2 of Cox et al. [6]). The following is a geometric formulation which is equivalent to the Lemma stated in Cox et al. [6].

Lemma 7 *(Dickson Lemma)*

Let $\{p_1, p_2, \ldots\}$ be a sequence of points in $\mathbb{N}^n$ such that $p_i \not\leq p_j$ whenever $i < j$. Then this sequence is finite.

Proof of termination. Termination of the above algorithm follows immediately by applying Dickson's Lemma to the set $\{(v^+, v^-) : v \in G \setminus F\}$, where for $v \in \mathbb{Z}^n$ we component-wise define $v^+ := \max(0, v)$ and $v^- := (-v)^+ = \max(0, -v)$. $\square$

Proof of correctness. Let G denote the set which is returned by the algorithm. We show that any non-zero $z \in \ker(A)$ can be written as a positive linear combination of elements G, where each vector in this combination lies in the same orthant as z. For each element $z \in G_{IP}(A)$ this representation is trivial, that is $z = z$, and we conclude $G_{IP}(A) \subseteq G$.

Since G contains the input set F, the vector z can be written as a positive linear integer combination $z = \sum \alpha_i v_i$ for some positive $\alpha_i \in \mathbb{Z}_{\geq 0}$ and $v_i \in G$.

From the set of all such positive linear integer combinations choose one minimal with respect to $\sum \alpha_i \|v_i\|_1$. Note that $\sum \alpha_i \|v_i\|_1 \geq \|z\|_1$ with equality if and only if all v_i lie in the same orthant as z.

Now assume for our minimally chosen linear combination that $\sum \alpha_i \|v_i\|_1 > \|z\|_1$, otherwise we are done. Take v_{i_1}, v_{i_2} from this sum which have non-zero first components of different signs. Then the vector $v_{i_1} + v_{i_2} \in$ S-vectors(v_{i_1}, v_{i_2}) was reduced with respect to G giving a representation $v_{i_1} + v_{i_2} = \sum \beta_j v'_j$ for some positive $\beta_j \in \mathbb{Z}_{\geq 0}$ and $v'_j \in G$. Moreover $\beta_j v'_j \sqsubseteq v_{i_1} + v_{i_2}$ for all j, giving for each component

$$\begin{aligned} \sum \beta_j \left| {v'_j}^{(k)} \right| &= \left| \sum \beta_j {v'_j}^{(k)} \right| \\ &= \left| (v_{i_1} + v_{i_2})^{(k)} \right| \\ &\leq \left| v_{i_1}^{(k)} \right| + \left| v_{i_2}^{(k)} \right|, \end{aligned}$$

where the last inequality is strict for $k = 1$ by construction. Hence

$$\sum \beta_j \|v'_j\|_1 = \|v_{i_1} + v_{i_2}\|_1 < \|v_{i_1}\|_1 + \|v_{i_2}\|_1.$$

But then

$$z = \sum \beta_i v'_i + (\alpha_1 - 1) v_{i_1} + (\alpha_2 - 1) v_{i_2} + \sum_{i \neq i_1, i_2} \alpha_i v_i$$

contradicts the minimality property of $v_1, \ldots, v_n$. So $\sum \alpha_i \|v_i\|_1 = \|z\|_1$ and our claim follows. $\square$

The above proof is related to ideas in the proof of Theorem 2.1.8 in Urbaniak [15], see also Cornuejols et al. [5] and Weismantel [17]. However, our scope differs from that in [5], [15], [17] and we have included the proof to indicate a close relation to the LP-case where we will identify a counterpart to the above algorithm which computes LP-Graver test sets.

LP-case

Graver [7] already proved that circuits provide a universal test set in LP. To the best of our knowledge, the only known algorithm to compute them seems to be that given for example in chapter 4 of Sturmfels [12]. In the following we will give an algorithm which uses the same algorithmic pattern as in the integer case.

Reduction and S-vectors

reducible?(s,g):

Input: vectors $s, g \in \mathbb{R}^n$

Output: TRUE, if s is reducible by g, otherwise FALSE

if $\operatorname{supp}(g) \subseteq \operatorname{supp}(s)$ then return TRUE else return FALSE

reduce(s,g):

Input: vectors $s, g \in \mathbb{R}^n$, such that reducible?(s, g) = TRUE

Output: s reduced by g

Choose $\alpha \in \mathbb{R}$ such that $\operatorname{supp}(s - \alpha g) \subsetneq \operatorname{supp}(s)$

return $s - \alpha g$

If reducible(s, g) = TRUE then reduce$(s, g) = s - \alpha g$ contains at least one zero entry more than s implying that normalForm(s, G) always terminates. Now for each pair $v, w \in G$ define S-vectors(v, w) to be the set of all vectors $v - \alpha w$ which contain a zero entry at some component at which neither v nor w had a zero entry. That is, $(v - \alpha w)^{(i)} = 0$ but $v^{(i)} w^{(i)} \neq 0$.

Algorithm

LP-Graver test set algorithm:

Input: generating set F over $\mathbb{Q}$ of $\ker(A)$

Output: a set G containing some non-zero scalar multiple of any circuit of A

```
G := F
C := ⋃_{f,g∈G} S-vectors(f, g)
while C ≠ ∅ do
        s := an element in C
        C := C − {s}
        f := normalForm(s, G)
        if f ≠ 0 then
                C := C ∪ ⋃_{g∈G} S-vectors(f, g)
                G := G ∪ {f}
return G.
```

The set $G_{LP}(A)$ is the set of all elements z in G which are irreducible with respect to $G \setminus \{z\}$ normalized to integer vectors with greatest common divisor 1 together with their negatives.

Proof of termination. Each vector in $G \setminus F$ must have a different support. Hence the algorithm terminates. $\square$

Proof of correctness. Let G denote the set which is returned by the algorithm. We prove that every circuit of A is reducible by an element of G, implying that G contains some scalar multiple of this circuit and the claim follows.

Fix any circuit q of A. Without loss of generality suppose that the zero components of q are the first m components. Since the vectors in G generate $\ker(A)$ over $\mathbb{R}$ we have

$$q = \sum \alpha_i g_i$$

for some $\alpha_i \in \mathbb{R}$ and $g_i \in G$.

We will now rewrite this linear combination such that all vectors from G appearing in the rewritten linear combination have a zero first component. Repeating this process with the second, third, $\ldots, m^{th}$ component, we arrive at a linear combination of elements of G which have zeros in their first m components. Since at least one $g_j \in G$ occurs in this linear combination, this g_j satisfies $\operatorname{supp}(g_j) \subseteq \operatorname{supp}(q)$. Since q is a circuit, we conclude by Lemma 4 that $g_j \in G$ is a scalar multiple of q and we are done.

Suppose all g_i in

$$q = \sum \alpha_i g_i$$

have already a zero first component. Then we are done. If not, there are at least two vectors g_j and g_k with non-zero first component, since

$$0 = q^{(1)} = \sum \alpha_i g_i^{(1)}.$$

Take g_j and g_k and rewrite $\alpha_j g_j + \alpha_k g_k$ as follows.

$$\begin{aligned} \alpha_j g_j + \alpha_k g_k &= \alpha_j g_j + \alpha_k \left(g_k - \frac{g_k^{(1)}}{g_j^{(1)}} g_j \right) + \alpha_k \frac{g_k^{(1)}}{g_j^{(1)}} g_j \\ &= \left(\alpha_j + \alpha_k \frac{g_k^{(1)}}{g_j^{(1)}} \right) g_j + \alpha_k \left(g_k - \frac{g_k^{(1)}}{g_j^{(1)}} g_j \right). \end{aligned}$$

Since the first component of

$$\left(g_k - \frac{g_k^{(1)}}{g_j^{(1)}} g_j \right)$$

vanishes, it belongs to S-vectors(g_j, g_k) and hence was reduced to zero with respect to G. From such a reduction process we get a representation of

$$\left(g_k - \frac{g_k^{(1)}}{g_j^{(1)}} g_j \right)$$

as a linear combination of elements in G with a zero first component. Substituting this into above linear combination for q we arrive at a new representation

$$q = \sum \alpha_i' g_i,$$

where the number of vectors in this linear combination with non-zero first component is at least one less than the corresponding number for

$$q = \sum \alpha_i g_i.$$

Hence this process terminates with a linear combination using only elements of G which have a zero first component. This proves our claim. $\square$

4 Optimizing using test sets

4.1 IP-case

In the integer case $G_{IP}(A)$ is a universal test set for $\min\{c^\intercal z : Az = b, z \in \mathbb{Z}^n_{\geq 0}\}$. However, termination of the optimization process is only ensured if $\min\{c^\intercal z : Az = b, z \in \mathbb{Z}^n_{\geq 0}\}$ is bounded with respect to c. Under the assumption that A, b have rational entries, the following are equivalent (see Schrijver [11]).

- $\min\{c^\intercal z : Az = b, z \in \mathbb{Z}^n_{\geq 0}\}$ is bounded with respect to c.
- $\min\{c^\intercal z : Az = b, z \in \mathbb{R}^n_{\geq 0}\}$ is bounded with respect to c.

The second statement holds if and only if $\ker(A) \cap \mathbb{R}^n_{\geq 0} = \{0\}$. The latter is true if and only if there is no circuit q of A with $q \geq 0$. Since the circuits $G_{LP}(A)$ are contained in $G_{IP}(A)$ by Lemma 5, we can check the second statement either by standard techniques from LP or by looking for some vector q in $G_{IP}(A)$ with $q \geq 0$.

Now assume that $\min\{c^\intercal z : Az = b, z \in \mathbb{Z}^n_{\geq 0}\}$ is bounded with respect to c. Let $M := \min\{|c^\intercal g| : g \in G_{IP}(A), c^\intercal g \neq 0\}$. Then each time we improve a given feasible solution z_0 by an element $g \in G_{IP}(A)$ to $z_0 - \alpha g$ the cost function value drops by at least $M > 0$ since α is a positive integer. Since $\min\{c^\intercal z : Az = b, z \in \mathbb{Z}^n_{\geq 0}\}$ is bounded with respect to c this can happen only finitely often.

So our optimization process always terminates if $\min\{c^\intercal z : Az = b, z \in \mathbb{Z}^n_{\geq 0}\}$ is bounded with respect to c.

4.2 LP-case

Again assume that $\min\{c^\intercal z : Az = b, z \in \mathbb{R}^n_{\geq 0}\}$ is bounded with respect to c and let z_0 be any feasible solution of the problem. If z_0 cannot be improved by any circuit then it is optimal.

Circuits provide an improving direction to any non-optimal feasible solution of the given LP-problem. However, as the following example shows, a zig-zagging effect, even to a non-optimal point, is possible and we have to take some care on how to choose the next circuit.

Zig-zagging example to a non-optimal point

Consider the problem $\min\{z_1 + z_2 - z_3 : 2z_1 + z_2 \leq 2, z_1 + 2z_2 \leq 2, z_3 \leq 1, (z_1, z_2, z_3) \in \mathbb{R}^3_{\geq 0}\}$ with optimal solution $(0, 0, 1)$. Introducing slack variables z_4, z_5, z_6 we obtain the problem

$$\min\{z_1 + z_2 - z_3 : Az = (2, 2, 1)^\intercal, z \in \mathbb{R}^6_{\geq 0}\}$$

with

$$A = \begin{pmatrix} 2 & 1 & 0 & 1 & 0 & 0 \\ 1 & 2 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

$\ker(A)$ is generated by $(1, 0, 0, -2, -1, 0)$, $(0, 1, 0, -1, -2, 0)$ and $(0, 0, 1, 0, 0, -1)$ and we obtain

$$(1, 0, 0, -2, -1, 0), (0, 1, 0, -1, -2, 0), (1, -2, 0, 0, 3, 0), (2, -1, 0, -3, 0, 0), (0, 0, 1, 0, 0, -1)\}$$

together with their negatives as the circuits of A. Our improving directions are given by all those circuits v for which $c^\intercal v > 0$. These are

$$(1, 0, 0, -2, -1, 0), (0, 1, 0, -1, -2, 0), (-1, 2, 0, 0, -3, 0), (2, -1, 0, -3, 0, 0), (0, 0, -1, 0, 0, 1).$$

Now start with the feasible solution $z_0 = (0, 1, 0, 1, 0, 1)$.

Going in the directions $(0, 1, 0, -1, -2, 0)$ and $(0, 0, -1, 0, 0, 1)$ as far as possible we immediately arrive at $(0, 0, 1, 2, 2, 0)$ which corresponds to the desired optimal solution $(0, 0, 1)$ of our original problem.

However, alternatively choosing only $(-1,2,0,0,-3,0)$ and $(2,-1,0,-3,0,0)$ as directions we keep improving z_0 infinitely often as follows:

$$\begin{aligned}(0,1,0,1,0,1) &\rightarrow \left(\frac{1}{2},0,0,1,\frac{3}{2},1\right)\\ &\rightarrow \left(0,\frac{1}{4},0,\frac{7}{4},\frac{6}{4},1\right)\\ &\rightarrow \left(\frac{1}{8},0,0,\frac{14}{8},\frac{15}{8},1\right)\\ &\rightarrow \left(0,\frac{1}{16},0,\frac{31}{16},\frac{30}{16},1\right)\\ &\quad\vdots\end{aligned}$$

In our original space $\mathbb{R}^3$ this reads as

$$(0,1,0) \rightarrow \left(\frac{1}{2},0,0\right) \rightarrow \left(0,\frac{1}{4},0\right) \rightarrow \left(\frac{1}{8},0,0\right) \rightarrow \left(0,\frac{1}{16},0\right) \rightarrow \ldots$$

clearly showing the zig-zagging effect to the non-optimal point $(0,0,0)$. So we have to impose certain constraints on the circuit which is chosen next.

Procedure to reach optimum in finitely many steps

If z_0 is not optimal then, starting with z_0, we can minimize $c^\intercal z$ by using the directions given by the circuits. To ensure termination we split the optimization process into two phases.

1. Only those circuits are allowed for improvement which lead to a better feasible solution with an additional zero entry. Repeat this step until there is no circuit with this property and go to phase 2.
2. Choose any appropriate circuit for improvement. Stop, if there is no such circuit, otherwise improve and go back to phase 1.

Proposition 1 *Under the assumption* $\ker(A) \cap \mathbb{R}^n_{\geq 0} = \{0\}$ *this process terminates with an optimal solution.*

Proof. The feasible solution z' at the end of phase 1 has to be optimal with respect to all feasible solutions whose support is contained in $\text{supp}(z')$

To see this, assume on the contrary that there exists a feasible z'' with $\text{supp}(z'') \subseteq \text{supp}(z')$ and $c^\intercal z'' < c^\intercal z'$. Since $z'' - z' \in \ker(A)$ there is some circuit q with $\text{supp}(q) \subseteq \text{supp}(z'' - z') \subseteq \text{supp}(z')$ and $c^\intercal q > 0$. This circuit improves z'. Since at least one entry of q is positive (otherwise $-q \neq 0$ and $-q \in \ker(A) \cap \mathbb{R}^n_{\geq 0}$) there is some finite positive α such that $z' - \alpha q$ is a better feasible solution with an additional zero entry, contradicting the fact that the first phase finished with z'.

If z' is optimal the algorithm terminates. Otherwise, the zero pattern of z' will never reappear after z' was improved in phase 2. Hence no zero pattern reappears at the end of the first phase and our optimization process has to terminate.

The feasible solution upon termination has to be optimal since in the second phase no augmenting circuit was found. The claim follows by the universal test set property of circuits. $\square$

5 Conclusions

We have presented a common pattern for computing test sets in LP and IP. According to this pattern test set computation essentially relies on specifying ingredients that are well known in the context of computer algebra. These are a normal form algorithm involving a proper notion of reduction and an S-pair inspection that handles situations which are "critical" with respect to the reduction.

Our pattern coincides with the critical pair/completion procedure that arises in automated theorem proving, polynomial ideal theory and in the solution of word problems in universal algebras, see Buchberger [2] for a survey.

When optimizing with the help of test sets, attention has to be paid in the LP situation. An example shows that simple augmentation may result in zig-zagging towards a non-optimal point. We have presented a strategy to prevent this.

References

[1] A.M. Bigatti, R. LaScala, and L. Robbiano. Computing toric ideals. Preprint, Università di Genova, Italy, 1998.

[2] B. Buchberger. History and Basic Features of the Critical-Pair/Completion Procedure. *Journal of Symbolic Computation*, 2:3–38, 1987.

[3] B. Buchberger. Gröbner bases: an algorithmic method in polynomial ideal theory. In N. K. Bose, ed., *Recent Trends in Multidimensional Systems Theory*, D. Reidel Publishing Company, Dordrecht, 1985, pp. 184–232.

[4] P. Conti and C. Traverso. Buchberger Algorithm and Integer Programming. In *Proceedings AAECC-9, (New Orleans)*, LNCS 539, Springer-Verlag, 1991, pp. 130–139.

[5] G. Cornuejols, R. Urbaniak, R. Weismantel, and L. Wolsey. Decomposition of Integer Programs and of Generating Sets. LNCS 1284, Springer-Verlag, 1997, pp. 92–103.

[6] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, Algorithms.* Springer-Verlag, 1992.

[7] J. E. Graver. On the foundation of linear and integer programming I. *Mathematical Programming*, 9:207–226, 1975.

[8] S. Hosten and B. Sturmfels. GRIN: 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.

[9] H. E. Scarf. Test sets for integer programming. *Mathematical Programming*, 99:355–368, 1997.

[10] H. E. Scarf. Private communication, 1998.

[11] A. Schrijver. *Theory of linear and integer programming.* Wiley, Chichester, 1986.

[12] B. Sturmfels. *Gröbner Bases and Convex Polytopes.* American Mathematics Society, Providence, Rhode Island, 1995.

[13] B. Sturmfels and R. R. Thomas. Variation of cost functions in integer programming. *Mathematical Programming*, 77:357–387, 1997.

[14] R. R. Thomas. A geometric Buchberger algorithm for integer programming. *Mathematics of Operations Research*, 20:864–884, 1995.

[15] R. Urbaniak. Decomposition of Generating sets of Integer Programs. Dissertation, Technische Universität Berlin, Germany, January 1998.

[16] R. Urbaniak, R. Weismantel, and G. M. Ziegler. A variant of the Buchberger algorithm for integer programming. *SIAM J. Discrete Mathematics*, 10:96–108, 1997.

[17] R. Weismantel. Test Sets of Integer Programming. *Mathematical Methods of Operations Research*, 47:1–37, 1998.