On the Positive Sum Property of Graver Test Sets

Raymond Hemmecke*
Department of Mathematics, Gerhard-Mercator University Duisburg
Lotharstr. 65, D-47048 Duisburg, Germany
hemmecke@math.uni-duisburg.de

April 4, 2000

Abstract

We discuss the positive sum property, a property inherent to Graver test sets and show how this property leads to a simple algorithm to compute LP and IP Graver test sets. MIP test sets, however, need not be finite in general. We present a finite set of vectors from which an improving vector can be reconstructed. Moreover, we show that universal test sets can help to find an initial feasible solution and that fixed upper bounds on the variables allow a truncation of the test set and an acceleration of the algorithm for its computation.

AMS classification: 90 C 10, 90 C 11

Keywords: Mixed-integer programming, Graver test sets

1 Introduction

Let $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$ denote the integers, rationals, and reals, respectively, let $\mathbb{X} = \mathbb{Z}^{n_z} \times \mathbb{R}^{n_q}$, and denote by $\mathbb{X}_{\geq 0}$ the corresponding non-negative orthant. For given $n = n_z + n_q$, $A \in \mathbb{Q}^{d\times n}$, $c \in \mathbb{R}^n$, and $b \in \mathbb{R}^d$, consider the family of optimization problems

$$(P)_{c,b}: \qquad \min\{c^\intercal z : Az = b,\, z \in \mathbb{X}_{\geq 0}\}$$

as $c \in \mathbb{R}^n$ and $b \in \mathbb{R}^d$ vary. Instead of $(P)_{c,b}$ we write $(LP)_{c,b}$ if $\mathbb{X} = \mathbb{R}^n$ and $(IP)_{c,b}$ if $\mathbb{X} = \mathbb{Z}^n$. In the present paper we address solving $(P)_{c,b}$ by test set methods.

Definition 1.1 *(Test Set)*
A set $\mathcal{T} \subseteq \mathbb{R}^n$ is called a test set for $(P)_{c,b}$ if

1. *$c^\intercal t > 0$ for all $t \in \mathcal{T}$, and*
2. *for every $b \in \mathbb{R}^d$ and for every non-optimal feasible point z_0 of $(P)_{c,b}$ there exist a vector $t \in \mathcal{T}$ and a scalar $\alpha > 0$ such that $z_0 - \alpha t$ is feasible.*

A vector $t \in \mathcal{T}$ satisfying these two conditions is called an improving vector or an improving direction.

A set is called a universal test set for $(P)_{c,b}$ if it contains a test set for every cost vector $c \in \mathbb{R}^n$.

In contrast to the common definition of test sets we do not impose finiteness on $\mathcal{T}$. This allows a treatment of test sets for mixed-integer programs which need not be finite in general (see for example Cook et. al [5]).

Once a finite (universal) test set $\mathcal{T}$ for $(P)_{c,b}$ is available the following augmentation algorithm can be applied in order to solve the above optimization problem.

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

Algorithm 1.2 *(Augmentation Algorithm)*

Input: $\mathbb{X}$, *a matrix* A, *a finite test set* $\mathcal{T}$ *for* $(P)_{c,b}$, *a cost vector* c, *a feasible solution* z_0 *for given* b

Output: an optimum $z_{\min}$ *of* $(P)_{c,b}$

while there is $t \in \mathcal{T}$ *with* $c^\intercal t > 0$ *and a scalar* $\alpha > 0$ *such that* $z_0 - \alpha t$ *is feasible do*

$z_0 := z_0 - \alpha t$, *where* α *is chosen maximal such that* $z_0 - \alpha t$ *is still feasible*

return z_0

In 1975 Graver [12] introduced finite universal test sets for linear programs (LP) and for linear integer programs (IP). However, he provided no algorithm to compute them. He also introduced a similar test set for linear mixed-integer programs (MIP) and already noted that they need not be finite in general [8].

A property that is inherent to all these test sets is the following.

Definition 1.3 *(Positive Sum Property)*

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

- $g_i \in G$, $\alpha_i > 0$, $\alpha_i g_i \in S$, *and*
- *for all* i, g_i *and* v *belong to the same orthant, that is,* $g_i^{(k)} v^{(k)} \geq 0$ *for every component* $k = 1, \ldots, n$.

Graver test sets are inclusion minimal sets having the positive sum property with respect to the set $\ker_{\mathbb{X}}(A) := \{v \in \mathbb{X} : Av = 0\}$, the kernel of A. Already Graver [12] showed that the positive sum property implies the universal test set property. We include a proof for this important fact in section 2.

In addition to this we present criteria (Lemmas 2.3, 2.9) characterizing those sets which have the positive sum property with respect to $\ker_{\mathbb{R}^n}$ or with respect to $\ker_{\mathbb{Z}^n}$. These criteria immediately lead us to algorithms for the computation of LP and IP Graver test sets using the algorithmic pattern of a completion procedure ([3]): step by step an initial set of vectors is completed with respect to the positive sum property by adding new vectors to the set as long as necessary. The algorithm for the IP-case can already be found in [17] in the disguised form of an n-dimensional Euclidean algorithm.

In the second part of the paper, section 3, we separately treat Graver test sets for LP, IP, and MIP in more detail. In the LP- and in the IP-cases we relate the Graver test set to the circuits of the matrix A and to Hilbert bases of certain pointed rational cones, respectively. Finally, we complete each subsection with a discussion on termination of the above augmentation procedure. We demonstrate that particularly in the LP-case some caution is appropriate to avoid zig-zagging (even to non-optimal solutions) and we present a strategy which ensures termination.

For the MIP-case we present an example similar to the one by Cook et. al [5] which implies that MIP-test sets need not be finite in general. In [5] the existence of finite MIP-test sets is shown if considerations are restricted to a smaller family of optimization problems. In contrast to this approach we introduce a finite set of integer vectors which is no test set by the above definition. However, we can nonetheless reconstruct from it an improving vector to any non-optimal feasible solution of a given problem. This reconstruction property enables us to use the above Augmentation Algorithm 1.2 as well. We conclude this subsection with a discussion on termination of the augmentation procedure in MIP.

A test set has to give improving directions for any right-hand-side b and hence cannot depend on the specific choice of b. The only and usually already hard step specific to b is to find some feasible solution of the given problem. Although there are algorithmic alternatives, we show in section 4 that universal test sets can be used to solve this feasibility problem as well. Again, some care has to be taken in the LP-case to ensure termination of this procedure.

Finally, we conclude the paper with a section on truncated Graver test sets for the family of problems $\min\{c^\intercal z : Az = b,\, 0 \leq z \leq u,\, z \in \mathbb{Z}^n\}$ as only $c \in \mathbb{R}^n$ and $b \in \mathbb{R}^d$ vary. Here, fixed finite upper bounds on z are used to truncate the Graver test set for $\min\{c^\intercal z : Az = b,\, z \in \mathbb{Z}^n_{\geq 0}\}$ to a sufficient subset and to speed up the direct computation of this truncated set. The presented algorithm, however, is more general and computes all non-zero minimal integer vectors in the set $\{z : Az = 0, l \leq z \leq u, z \in \mathbb{Z}^n\}$, $l \leq 0 \leq u$, where a vector v is called minimal if there is no w in this set such that v and w belong to the same orthant and such that the components of w are not greater in absolute value than the corresponding components of v. Again, the same algorithmic pattern as for Graver test set computations is used. For special choices of l and u this includes an algorithmic solution to the following problems:

- For $l = 0$ and $u = 1$ (component-wise) the algorithm computes test sets for combinatorial optimization problems.
- For $l = -u$ the algorithm computes the desired truncated Graver test set.
- For $l = -\infty$ and $u = +\infty$ the algorithm is similar to Algorithm 2.1.7 in Urbaniak [23] and computes the IP Graver test set corresponding to A. However, it is much slower than Algorithm 2.10 (section 2), since the computation is done in a higher dimensional space.
- For $l = 0$ and $u = +\infty$ the algorithm computes the unique minimal Hilbert basis for the pointed rational cone $\{z : Az = 0, z \in \mathbb{Z}^n_{\geq 0}\}$.

The notion of a truncated Graver test set already appeared in a paper by Thomas and Weismantel [22]. They define a partial order on the right-hand-sides (b, u) in order to truncate test sets for problems of type $\min\{c^\intercal z : Az = b,\ 0 \leq z \leq u,\ z \in \mathbb{Z}^n\}$. Our algorithm to compute a truncated Graver test set could be deduced from their approach by defining their partial order not on all components of the right-hand-side (b, u) but only on the components of u, the upper bounds on z. Our approach uses a simpler algebraic machinery, whereas the one by Thomas and Weismantel is more general.

Before starting our analysis let us add some bibliographical remarks.

Only in 1991, Conti and Traverso [4] succeeded in computing a test set for $(IP)_{c,b}$ for fixed c by using Buchberger's Gröbner basis algorithm [2]. The algorithm by Conti and Traverso, however, had a large computational overhead and since then a lot of work has been done to improve its performance. By fundamental algebraic relations the problem was reduced to saturating a certain polynomial 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 [19].

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 [13]).

A geometrical interpretation of the Conti-Traverso algorithm was given by Thomas [21]. This has initiated further work on geometric algorithms for the computation of test sets, see Cornuejols et al. [6] and Urbaniak et al. [24].

Structural similarities of Graver test sets in LP and IP were elaborated in Sturmfels and Thomas [20]. In [25], Weismantel gives a survey on all currently known test sets in IP together with their computation.

Some recent work has been done on MIP test sets. Following a geometric approach, Köppe [14, 15] obtained a finiteness result similar to the one we obtained by an algebraic argument in section 3.

2 Positive Sum Property

In this section we show that the positive sum property implies the universal test set property. Afterwards we present two criteria which allow us to decide whether a given symmetric set G has the positive sum property with respect to $\ker_{\mathbb{Z}^n}(A)$ or $\ker_{\mathbb{R}^n}(A)$. Herein, a set G is called symmetric if $v \in G$ implies $-v \in G$.

The two criteria lead us to a completion procedure: a given (symmetric) generating set G of $\ker_{\mathbb{Z}^n}(A)$ over $\mathbb{Z}$ or of $\ker_{\mathbb{R}^n}(A)$ over $\mathbb{R}$ is completed with respect to the positive sum property by adding new elements to G as long as necessary. Once the completion procedure terminates, the two criteria below already imply its correctness, that is, the set of vectors returned has indeed the positive sum property with respect to $\ker_{\mathbb{Z}^n}(A)$ or $\ker_{\mathbb{R}^n}(A)$.

2.1 Positive Sum Property implies Universal Test Set Property

To simplify the subsequent proofs it is convenient to introduce the following relation.

Definition 2.1 *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* $j = 1, \ldots, n$*, that is,* u *belongs to the same orthant as* v *and its components are not greater in absolute value than the corresponding components of* v*.*

Clearly, if $u \sqsubseteq v$ then also $v - u \sqsubseteq v$.

The following lemma was already used by Graver [12]. We include a proof because of its importance.

Lemma 2.2 *(Positive Sum Property implies Universal Test Set Property)*

If G *has the positive sum property with respect to* $\ker_{\mathbb{X}}(A)$ *then* G *is a universal test set for* $(P)_{c,b}$*.*

Proof. Given $b \in \mathbb{R}^d$, $c \in \mathbb{R}^n$, and a non-optimal feasible point z_0 of $(P)_{c,b}$. We have to prove that there exist a vector $v \in G$ and a scalar $\alpha > 0$ such that $z_0 - \alpha v$ is a better feasible solution than z_0.

By z_1 denote another feasible solution with smaller cost function value than z_0. Then by the assumptions on G we can write $z_0 - z_1$ as a 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 claim that $z_0 - \alpha_j g_j$ is feasible and has a better cost function value than z_0.

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 $g_j \in \ker_{\mathbb{X}}(A)$ and $c^\intercal(\alpha_j g_j) > 0$. So it remains to prove that $z_0 - \alpha_j g_j \geq 0$. We prove this for each component k separately. By the definition of the relation $\sqsubseteq$ we conclude from $\alpha_j g_j \sqsubseteq z_0 - z_1$ that

$$\alpha_j g_j^{(k)} (z_0 - z_1)^{(k)} \geq 0 \tag{1}$$

and

$$\left|\alpha_j g_j^{(k)}\right| \leq \left|(z_0 - z_1)^{(k)}\right| \tag{2}$$

If $(\alpha_j g_j)^{(k)} \leq 0$ then $(z_0 - \alpha_j g_j)^{(k)} = z_0^{(k)} - (\alpha_j g_j)^{(k)} \geq z_0^{(k)} \geq 0$. If, on the contrary, $(\alpha_j g_j)^{(k)} > 0$ then by (1) and (2) also $(z_0 - z_1)^{(k)} \geq (\alpha_j g_j)^{(k)} > 0$ which implies $z_0^{(k)} - (\alpha_j g_j)^{(k)} \geq z_1^{(k)} \geq 0$ as desired. $\square$

2.2 Criterion to check Positive Sum Property with respect to $\ker_{\mathbb{Z}^n}(A)$

Lemma 2.3 *(Criterion for Positive Sum Property with respect to* $\ker_{\mathbb{Z}^n}(A)$*)*

A symmetric set $G \subseteq \ker_{\mathbb{Z}^n}(A)$ *has the positive sum property with respect to* $\ker_{\mathbb{Z}^n}(A)$ *if and only if the following two conditions hold:*

- G *finitely generates* $\ker_{\mathbb{Z}^n}(A)$ *over* $\mathbb{Z}$*, and*
- *for every pair* $v, w \in G$*, the vector* $v+w$ *can be written as a finite linear combination* $v+w = \sum \alpha_i g_i$ *where for all* i *we have* $g_i \in G$*,* $\alpha_i \in \mathbb{Z}_{>0}$*, and* $g_i \sqsubseteq v + w$*.*

Proof. We have to show that any non-zero $z \in \ker_{\mathbb{Z}^n}(A)$ can be written as a finite positive linear integer combination of elements from G where each vector in this combination belongs to the same orthant as z. Since G generates $\ker_{\mathbb{Z}^n}(A)$ over $\mathbb{Z}$ and since G is symmetric, the vector z can be written as a linear combination $z = \sum \alpha_i v_i$ for finitely many $\alpha_i \in \mathbb{Z}_{>0}$ and $v_i \in G$. Note that $\sum \alpha_i \|v_i\|_1 \geq \|z\|_1$ with equality if and only if $v_i \sqsubseteq z$ for all i.

From the set of all such linear integer combinations $\sum \alpha_i v_i$ choose one such that $\sum \alpha_i \|v_i\|_1$ is minimal and assume that $\sum \alpha_i \|v_i\|_1 > \|z\|_1$, since otherwise we are done. Select vectors v_{i_1}, v_{i_2} from this representation which have some component $k = k_0$ of different signs.

By the assumptions on G, the vector $v_{i_1} + v_{i_2}$ can be written as $v_{i_1} + v_{i_2} = \sum \beta_j v'_j$ for finitely many $\beta_j \in \mathbb{Z}_{>0}$ and $v'_j \in G$. Moreover, from $\beta_j v'_j \sqsubseteq v_{i_1} + v_{i_2}$ for all j we conclude for each component $k = 1, \ldots, n$, that

$$\sum_j \beta_j |{v'_j}^{(k)}| = |\sum_j \beta_j {v'_j}^{(k)}| = |(v_{i_1} + v_{i_2})^{(k)}| \leq |v_{i_1}^{(k)}| + |v_{i_2}^{(k)}|,$$

where the last inequality is strict for $k = k_0$ by construction. Summing up over $k = 1, \ldots, n$, yields $\sum \beta_j \|v'_j\|_1 = \|v_{i_1} + v_{i_2}\|_1 < \|v_{i_1}\|_1 + \|v_{i_2}\|_1$. But now z can be represented as

$$\begin{aligned} z &= \alpha_{i_1} v_{i_1} + \alpha_{i_2} v_{i_2} + \sum_{i \neq i_1, i_2} \alpha_i v_i \\ &= \sum \beta_j v'_j + (\alpha_{i_1} - 1) v_{i_1} + (\alpha_{i_2} - 1) v_{i_2} + \sum_{i \neq i_1, i_2} \alpha_i v_i \end{aligned}$$

and it holds

$$\sum \beta_j \|v'_j\|_1 + (\alpha_{i_1} - 1)\|v_{i_1}\|_1 + (\alpha_{i_2} - 1)\|v_{i_2}\|_1 + \sum_{i \neq i_1, i_2} \alpha_i \|v_i\|_1 < \sum \alpha_i \|v_i\|_1$$

in contradiction to the minimality required on $\sum \alpha_i \|v_i\|_1$. Altogether we obtain that $\|z\|_1 = \sum \alpha_i \|v_i\|_1$, i. e., G has the positive sum property with respect to $\ker_{\mathbb{Z}^n}(A)$. $\square$

2.3 Criterion to check Positive Sum Property with respect to $\ker_{\mathbb{R}^n}(A)$

In order to prove the criterion of Lemma 2.9 below we need to introduce the set of circuits associated with a matrix A. This set will turn out to be a minimal set having the positive sum property with respect to $\ker_{\mathbb{R}^n}(A)$.

Definition 2.4 *For $v \in \mathbb{R}^n$ the set* $\mathrm{supp}(v) := \{j : v^{(j)} \neq 0\}$ *is called the support of v.*

Definition 2.5 *(Circuits of a Matrix)*
A circuit of a matrix $A \in \mathbb{Q}^{d \times n}$ is a vector $q \in \ker_{\mathbb{R}^n}(A) \cap \mathbb{Z}^n$ of inclusion minimal support in $\ker_{\mathbb{R}^n}(A)$ whose components have greatest common divisor 1*.*

Note that if A has only rational entries then there is always a rational (and thus also an integer) vector for every minimal support in $\ker_{\mathbb{R}^n}(A)$. Proofs for the following three statements can be found in [12].

Lemma 2.6 *If v is a circuit of A and $w \in \ker_{\mathbb{R}^n}(A)$ 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.*

Corollary 2.7 *Every matrix $A \in \mathbb{Q}^{d \times n}$ has only finitely many circuits.*

Lemma 2.8 *The set of all circuits of A has the positive sum property with respect to $\ker_{\mathbb{R}^n}(A)$.*

Now we are in the position to state and to prove the main result of this subsection.

Lemma 2.9 *(Criterion for Positive Sum Property with respect to $\ker_{\mathbb{R}^n}(A)$)*
A symmetric set $G \subseteq \ker_{\mathbb{R}^n}(A)$ has the positive sum property with respect to $\ker_{\mathbb{R}^n}(A)$ if and only if the following two conditions hold:

- *G finitely generates $\ker_{\mathbb{R}^n}(A)$ over $\mathbb{R}$, and*
- *for every pair $v, w \in G$, and all $\alpha \in \mathbb{R}$, the vector $v + \alpha w$ can be written as a finite linear combination $v + \alpha w = \sum \alpha_i g_i$ where for all $i = 1, \ldots, n$, we have $g_i \in G$, $\alpha_i > 0$, and $\alpha_i g_i \sqsubseteq v + \alpha w$. Herein only those values for $\alpha \in \mathbb{R}$ need to be considered, for which $v + \alpha w$ contains a zero entry at some component k at which neither v nor w have a zero entry, that is, $(v + \alpha w)^{(k)} = 0$ but $v^{(k)} w^{(k)} \neq 0$.*

Proof. It suffices to prove that every circuit q of A is reducible by an element of G, since this implies that G contains some non-zero scalar multiple of q and Lemma 2.9 follows by application of Lemma 2.8.

Let q be an arbitrary circuit of A. Without loss of generality we may assume that the zero components of q are the first m components. Otherwise we may rearrange the variables to ensure this property. Since the vectors in G generate $\ker_{\mathbb{R}^n}(A)$ over $\mathbb{R}$ we have $q = \sum \alpha_i g_i$ for finitely many non-zero $\alpha_i \in \mathbb{R}$ and $g_i \in G$.

We will now rewrite this linear combination into a linear combination where all appearing vectors from G have a zero first component. Repeating this process with the second, third, $\ldots, m^{th}$ component, we arrive at a linear representation of q by elements from G which have zeros in their first m components. Choose any $g_j \in G$ that occurs in this last linear combination. Since $\operatorname{supp}(g_j) \subseteq \operatorname{supp}(q)$ and since q is a circuit, we conclude by Lemma 2.6 that $g_j \in G$ is a non-zero scalar multiple of q and our claim follows. So it remains to show that this rewriting step is always possible.

If all g_i in $q = \sum \alpha_i g_i$ have already a zero first component we can go on with the second, third, $\ldots, m^{th}$ component. If not, $0 = q^{(1)} = \sum \alpha_i g_i^{(1)}$ implies that there exist at least two vectors g_j and g_k with non-zero first components. 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, the assumptions on G yield 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 where each element in this sum belongs to the same orthant as $\left(g_k - \frac{g_k^{(1)}}{g_j^{(1)}} g_j \right)$. Substituting this representation back into the 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 finite linear representation of q using only elements from G with a zero first component. This finally concludes the proof. $\square$

2.4 Completion Algorithm

Lemmas 2.3 and 2.9 reduce the decision on whether a finite symmetric set G has the positive sum property with respect to $\ker_{\mathbb{Z}^n}$ or $\ker_{\mathbb{R}^n}$ to a finite number of representability tests which may be treated algorithmically. In order to complete a set G with respect to the positive sum property, these lemmas also suggest the following procedure which adds further elements to G as long as G does not have the desired positive sum property. This procedure is known as a completion procedure ([3]). In our case, the set G is completed with respect to the positive sum property.

Algorithm 2.10 *(Completion Procedure)*

Input: a finite symmetric set $F \subseteq \ker_{\mathbb{R}^n}(A)$ *generating* $\ker_{\mathbb{Z}^n}(A)$ *over* $\mathbb{Z}$ *or* $\ker_{\mathbb{R}^n}(A)$ *over* $\mathbb{R}$

Output: a set G *which contains* F *and has the positive sum property with respect to* $\ker_{\mathbb{Z}^n}(A)$ *or* $\ker_{\mathbb{R}^n}(A)$

$G := F$

$C := \bigcup_{f,g \in G}$ *S-vectors*(f,g)

<u>*while*</u> $C \neq \emptyset$ <u>*do*</u>

  $s :=$ *an element in* C

  $C := C - \{s\}$

  $f :=$ *normalForm*(s,G)

  <u>*if*</u> $f \neq 0$ <u>*then*</u>

    $C := C \cup \bigcup_{g \in G}$ *S-vectors*(f,g)

    $G := G \cup \{f\}$

<u>*return*</u> G.

In this algorithm the set S-vectors(f,g) corresponds to the "critical" vectors described in Lemmas 2.3 and 2.9, respectively. That is S-vectors$(f,g) = \{f+g\}$ for the integer case and S-vectors$(f,g) = \bigcup_\alpha \{f+\alpha g\}$ for the continuous case. As specified in Lemma 2.9, only those values for $\alpha \in \mathbb{R}$ need to be considered, for which $f+\alpha g$ contains a zero entry at some component k at which neither f nor g has a zero entry, that is, $(f+\alpha g)^{(k)} = 0$ but $f^{(k)} g^{(k)} \neq 0$. Note that if $s \in$ S-vectors(v,w) then $\mathrm{supp}(s) \subseteq (\mathrm{supp}(v) \cup \mathrm{supp}(w)) \setminus \{e\}$ for some $e \in \mathrm{supp}(v) \cap \mathrm{supp}(w)$. This construction is also used in matroid theory to characterize circuits (see for example Oxley [16]).

Behind the function normalForm(s,G) there is the following algorithm which returns 0 if a representation $s = \sum \alpha_i g_i$ with finitely many $\alpha_i \in \mathbb{Z}_{>0}$ (or $\alpha_i \in \mathbb{R}_{>0}$), $g_i \in G$, and $\alpha_i g_i \sqsubseteq s$ is found, or it returns a vector t such that a representation of this kind is possible for $G \cup \{t\}$.

The normalForm algorithm aims at finding a representation $s = \sum \alpha_i g_i$ with finitely many $\alpha_i \in \mathbb{Z}_{>0}$ (or $\alpha_i \in \mathbb{R}_{>0}$), $g_i \in G$, and $\alpha_i g_i \sqsubseteq s$ by reducing s by elements of G in such a way that, if at some point of this reduction the zero vector is reached, a desired representation $s = \sum \alpha_i g_i$ has been found. If the reduction process terminates with a vector $t \neq 0$ then a desired representation $s = \sum \alpha_i g_i$ with vectors from $G \cup \{t\}$ has been constructed. The vector t is called a normal form of s with respect to the set G.

Algorithm 2.11 *(normalForm Algorithm)*

<u>*Input:*</u> *a vector* s*, a set* G *of vectors*

<u>*Output:*</u> *a normal form of* s *with respect to* G

<u>*while*</u> *there is some* $g \in G$ *such that reducible?*$(s,g) =$ *TRUE* <u>*do*</u>

  $s :=$ *reduce*(s,g)

<u>*return*</u> s

The reduction involved in the normalForm algorithm has to be defined separately for the integer and the continuous cases. Our definition for the integer case leads to an algorithm already presented by Pottier [17] in the disguised form of an n-dimensional Euclidean algorithm.

Integer Case

Here we say that $s \in \mathbb{Z}^n$ can be reduced by $g \in \mathbb{Z}^n$ to $s-g$ if $g \sqsubseteq s$. If $g \sqsubseteq s$ then $s = g + (s-g)$ where $g \sqsubseteq s$ and $s - g \sqsubseteq s$. Since $\|s-g\|_1 < \|s\|_1$ we conclude that normalForm(s,G) always terminates.

To prove termination of the completion algorithm in the integer case we need the (Gordan-)Dickson Lemma (see for example section 4.2 of Cox et al. [7]). The following is an equivalent geometric formulation.

Lemma 2.12 *(Dickson Lemma)*

Let $\{p_1, p_2, \ldots\}$ *be a sequence of points in* $\mathbb{Z}^n_{\geq 0}$ *such that* $p_i \not\leq p_j$ *whenever* $i < j$*. Then this sequence is finite.*

Lemma 2.13 *With the above definitions of S-vectors and of normalForm for the integer case the Completion Procedure 2.10 terminates and satisfies its specifications.*

Proof. Termination of the above algorithm follows immediately by applying Dickson's Lemma to the sequence $\{(v^+, v^-) : v \in G \setminus F\}$, where for $v \in \mathbb{Z}^n$ we component-wise define $v^+ := \max(0, v)$ and $v^- := (-v)^+ = \max(0, -v)$. To see this, note that $f = \text{normalForm}(s, G)$ implies that there is no $g \in G$ with $g \sqsubseteq f$, or in other words, there is no $g \in G$ with $(g^+, g^-) \leq (f^+, f^-)$. So the algorithm produces a sequence of vectors $\{(v^+, v^-) : v \in G \setminus F\} = \{f_1, f_2, \dots\}$, where $(f_i^+, f_i^-) \not\leq (f_j^+, f_j^-)$ for $i < j$. Such a sequence is always finite by the Dickson Lemma. Correctness of the algorithm follows immediately from Lemma 2.3, since upon termination $\text{normalForm}(v + w, G) = 0$ for all $v, w \in G$ giving a representation $v + w = \sum \alpha_i g_i$ with $\alpha_i \in \mathbb{Z}_{>0}$, $g_i \in G$, and $g_i \sqsubseteq v + w$. $\square$

Continuous Case

Here we say that $s \in \mathbb{R}^n$ can be reduced by $g \in \mathbb{R}^n$ if g and s belong to the same orthant of $\mathbb{R}^n$ and $\text{supp}(g) \subseteq \text{supp}(s)$. In case of reducibility s is reduced to $s - \alpha g$ where $\alpha \in \mathbb{R}$ is chosen in such a way that $s - \alpha g$ and s belong to the same orthant of $\mathbb{R}^n$ and $\text{supp}(s - \alpha g) \subsetneq \text{supp}(s)$, which implies that $\text{normalForm}(s, G)$ always terminates. Moreover, $s = \alpha g + (s - \alpha g)$ where $\alpha g \sqsubseteq s$ and $s - \alpha g \sqsubseteq s$.

Remark. The proof of Lemma 2.9 did not take advantage of the fact that g, $s - \alpha g$, and s belong to the same orthant of $\mathbb{R}^n$. Thus, the condition that g and $s - \alpha g$ have to belong to the same orthant as s can be dropped in the definition of the above reduction. $\square$

Lemma 2.14 *With the above definitions of S-vectors and of normalForm for the continuous case the completion procedure 2.10 terminates and satisfies its specifications.*

Proof. Each vector in $G \setminus F$ must have a different support. Hence the algorithm terminates. Correctness of the above algorithm follws from Lemma 2.9 since upon termination $\text{normalForm}(v + \alpha w, G) = 0$ for all $v, w \in G$, and for all α such that $v + \alpha w$ contains a zero entry at some component at which both v and w are non-zero. This normalForm reduction to zero provides a representation $v + w = \sum \alpha_i g_i$ with $\alpha_i \in \mathbb{R}_{>0}$, $g_i \in G$, and $\alpha_i g_i \sqsubseteq v + \alpha w$ as required. $\square$

3 Graver Test Sets

In the sequel we introduce, separately for each situation, Graver test sets in LP, IP, and MIP as minimal sets of vectors having the positive sum property with respect to $\ker_{\mathbb{X}}(A)$.

LP and IP Graver test sets are always finite as was already shown by Graver in 1975 [12]. We show how the two completion procedures presented in section 2 for the integer and the continuous cases can be used to compute both sets. We concentrate on termination of the augmentation algorithm as presented in section 1. It turns out that some caution is appropriate in the LP-case to avoid zig-zagging.

MIP Graver test sets need not be finite in general. To demonstrate this we present a small example similar to the one by Cook et. al [5]. Thereafter we give a solution to this finiteness problem by presenting a finite set of integer vectors which is inherent to the MIP Graver test set and from which an improving vector to a non-optimal solution can be reconstructed by solving a finite number of pure LP's. We conclude this subsection by showing termination of the MIP augmentation algorithm if the optimization problem is solvable.

3.1 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 [18] and Weismantel [25].

Definition 3.1 *(Hilbert Basis)*
Let C be a rational cone. A finite set $H = \{h_1, \ldots, h_t\} \subseteq C \cap \mathbb{Z}^n$ is called 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 3.2 *(Existence and Uniqueness of Hilbert Bases)*
Every pointed rational cone possesses a unique Hilbert basis that is minimal with respect to inclusion.

Definition 3.3 *Let $\mathbb{O}_j$ be the j^{th} orthant of $\mathbb{R}^n$ and H_j the unique minimal Hilbert basis of $\ker_{\mathbb{Z}^n}(A) \cap \mathbb{O}_j$. Then we define $G_{IP}(A) := \bigcup H_j$ to be the IP Graver test set (or IP Graver basis) of A.*

As $G_{IP}(A)$ is a finite union of finite sets, $G_{IP}(A)$ has finite cardinality. By construction, the elements of $G_{IP}(A)$ are minimal with respect to the partial ordering $\sqsubseteq$ on $\mathbb{Z}^n$. Moreover, if $v \in G_{IP}(A)$ then also $-v \in G_{IP}(A)$, that is, G is symmetric.

Lemma 3.4 *$G_{IP}(A)$ has the positive sum property with respect to $\ker_{\mathbb{Z}^n}(A)$.*

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

Corollary 3.5 *$G_{IP}(A)$ is a universal test set for $(IP)_{c,b}$.*

In section 2 we have already seen how a generating set of $\ker_{Z^n}(A)$ over $\mathbb{Z}$ can be completed with respect to the positive sum property. The resulting set has to contain the IP Graver test set. In that way we may compute the IP Graver test set.

Termination of Augmentation Algorithm

In the integer case, termination of the Augmentation Algorithm 1.2 is ensured if and only if $(IP)_{c,b}$ is bounded with respect to c. Under the assumption that A, b have rational entries, the following are equivalent (see Schrijver [18]):

- $(IP)_{c,b}$ is bounded with respect to c.
- $(LP)_{c,b}$ is bounded with respect to c.

The second statement can be checked by standard techniques from LP.

Proposition 3.6 *If the relaxed problem $(LP)_{c,b}$ is bounded with respect to c then the IP augmentation algorithm terminates.*

Proof. Let $\epsilon := \min\{|c^\intercal g| : g \in G_{IP}(A), c^\intercal g \neq 0\}$. 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 $\alpha(c^\intercal g) \geq \epsilon > 0$ since α is a positive integer. Since $(LP)_{c,b}$ is bounded with respect to c this can happen only finitely often and the IP augmentation algorithm has to terminate. □

3.2 LP Graver Test Sets

For every orthant $\mathbb{O}_j$ of $\mathbb{R}^n$ consider the pointed rational cone $C_j := ker_{\mathbb{R}^n}(A) \cap \mathbb{O}_j$. Up to scalar factors this cone has a unique inclusion minimal generating set. Since C_j is a pointed rational cone we may scale each generator to have only integer components with greatest common divisor 1. Denote by H_j the (uniquely determined) generating set of C_j consisting of these integer vectors.

Definition 3.7 *For every orthant $\mathbb{O}_j$ of $\mathbb{R}^n$ let H_j be the unique minimal generating set of $\ker_{\mathbb{R}^n}(A) \cap \mathbb{O}_j$ which contains only integer vectors whose components have a greatest common divisor 1. Then we define $G_{LP}(A) := \bigcup H_j$ to be the LP Graver test set (or LP Graver basis) of A.*

As $G_{LP}(A)$ is a finite union of finite sets, $G_{LP}(A)$ has finite cardinality.

Lemma 3.8 *$G_{LP}(A)$ has the positive sum property with respect to $\ker_{\mathbb{R}^n}(A)$. Thus $G_{LP}(A)$ is a universal test set for $(LP)_{c,b}$.*

Proof. Take any non-zero element $z \in \ker_{\mathbb{R}^n}(A)$. Thus z belongs to some orthant $\mathbb{O}_j$ and can be written as a positive linear combination of elements of the generating set $H_j \subseteq G_{LP}(A)$ of $\ker_{\mathbb{R}^n}(A) \cap \mathbb{O}_j$. $\square$

Lemma 3.9 *$G_{LP}(A)$ coincides with the set of all circuits of A.*

Proof. We show that $G_{LP}(A)$ contains every circuit. The claim then follows by the positive sum property of the set of circuits.

Let q be a circuit of A belonging to the orthant $\mathbb{O}_j$ of $\mathbb{R}^n$. Then q can be written as $q = \sum \alpha_i g_i$ where $\alpha_i > 0$ and $g_i \in H_j$. This implies $\alpha_i g_i \sqsubseteq q$ for all i. We conclude that $\text{supp}(g_i) \subseteq \text{supp}(q)$ and thus $q = g_i$ by Lemma 2.6. $\square$

In section 2 we have already seen how a generating set of $\ker_{R^n}(A)$ over $\mathbb{R}$ can be completed with respect to the positive sum property. The resulting set has to contain a scalar multiple of each circuit. In that way we may compute the LP Graver test set.

Termination of Augmentation Algorithm

Again assume that $(LP)_{c,b}$ is bounded with respect to c and let z_0 be some 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. Thus we can apply the Augmentation Algorithm 1.2 to solve $(LP)_{c,b}$. 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

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

$\ker_{\mathbb{R}^6}(A)$ 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. The 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 along 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 in the augmentation algorithm. □

To ensure termination of Algorithm 1.2 we split the augmentation process into two phases.

Algorithm 3.10 *(Augmentation Strategy to reach Optimum in finitely many Steps)*

1. *In phase* 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. *In phase* 2 *any appropriate circuit is chosen for improvement. Stop, if there is no such circuit, otherwise improve and go back to phase* 1.

Proposition 3.11 *Under the assumption that the $(LP)_{c,b}$ is solvable, the LP Augmentation Algorithm 1.2 which uses the Augmentation Strategy 3.10 terminates with an optimal solution.*

Proof. At the end of phase 1 the current feasible solution z' has to be optimal with respect to all feasible solutions whose support is contained in $\operatorname{supp}(z')$, that is, there is no feasible solution z'' with $\operatorname{supp}(z'') \subseteq \operatorname{supp}(z')$ and $c^\intercal z'' < c^\intercal z'$. To see this, assume to the contrary that there is such a feasible solution z''. Since $z' - z'' \in \ker_{\mathbb{R}^n}(A)$ there exists some circuit q with $\operatorname{supp}(q) \subseteq \operatorname{supp}(z' - z'') \subseteq \operatorname{supp}(z')$ and $c^\intercal q > 0$. This circuit, however, improves z': Choose the largest positive scalar α such that $z' - \alpha q$ is still feasible. This scalar is finite since $LP_{c,b}$ is bounded with respect to the given cost function. We conclude that $z' - \alpha q$ contains an additional zero entry and $c^\intercal(z' - \alpha q) < c^\intercal z'$, contradicting the fact that the first phase finished with z'.

Now, if z' is optimal then the algorithm terminates in phase 2. Otherwise, the zero pattern of z' will never reappear after z' was improved in phase 2. Thus no zero pattern reappears at the end of the first phase and the augmentation 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 now follows by the universal test set property of circuits. □

3.3 MIP Graver Test Sets

Let $A = (A_1|A_2)$, where the columns of A_1 and A_2 correspond to the integer and continuous variables, respectively. Throughout this section A_1 and A_2 are assumed to be integer matrices of sizes $d \times n_1$ and $d \times n_2$. Analogously, we subdivide $c = (c_1, c_2)$. Let $z \in \mathbb{Z}^{n_1}$ and $q \in \mathbb{R}^{n_2}$ denote the integer and continuous variables, respectively. Our aim is to construct a universal test set for the family of optimization problems $(P)_{c,b}$ as $c \in \mathbb{R}^n$ and $b \in \mathbb{R}^d$ vary.

Again, a Graver test set in the mixed-integer situation is defined to be an inclusion minimal subset of $\ker_{\mathbb{X}}(A)$ which has the positive sum property with respect to $\ker_{\mathbb{X}}(A)$. This leads us to the following set of vectors.

Definition 3.12 *The MIP Graver test set $G_{MIP}(A)$ contains all vectors $(z, q) \in \ker_{\mathbb{X}}(A)$ such that $z = 0$ and $q \in G_{LP}(A_2)$ together with all vectors $(z, q) \in \ker_{\mathbb{X}}(A)$ such that $z \neq 0$ and there is no $(z', q') \in \ker_{\mathbb{X}}(A)$ satisfying $(z', q') \sqsubseteq (z, q)$.*

The proof of the following lemma can be found in Foroudi and Graver [8]. (To be self-contained we reproduce its proof in the appendix.)

Lemma 3.13 *$G_{MIP}(A)$ is an inclusion minimal set which has the positive sum property with respect to $\ker_{\mathbb{X}}(A)$. Consequently, $G_{MIP}(A)$ is a universal test set for $(P)_{c,b}$.*

The set $G_{MIP}(A)$, however, need not be finite in general. To demonstrate this we present an example similar to the one by Cook et. al [5].

Example. Consider the family of optimization problems

$$\min\{c_0 z + c_1 q_1 + c_2 q_2 : z + q_1 + q_2 = b, z \in \mathbb{Z}_{\geq 0}, q_1, q_2 \in \mathbb{R}_{\geq 0}\}$$

as $c = (c_0, c_1, c_2) \in \mathbb{R}^3$ and $b \in \mathbb{R}$ vary. Thus we have $A = (A_1|A_2)$ where $A_1 = (1)$ and $A_2 = (1\ 1)$. Every element in $\ker_{\mathbb{Z}\times\mathbb{R}^2}(A)$ can be written as $\alpha_1(1, -1, 0) + \alpha_2(0, 1, -1)$ with $\alpha_1 \in \mathbb{Z}$ and $\alpha_2 \in \mathbb{R}$. We will prove that for every $0 < \beta < 1$ the vector $u := (1, -1, 0) + \beta(0, 1, -1) = (1, -1 + \beta, -\beta) \in \ker_{\mathbb{Z}\times\mathbb{R}^2}(A)$ cannot be written as a sum $v + w$ with $v, w \in \ker_{\mathbb{Z}\times\mathbb{R}^2}(A)$ and $v, w \sqsubseteq u$.

Suppose that such vectors v, w exist. Then one of the two vectors, say v, has zero integer component. Hence $v = \gamma(0, 1, -1)$ for some $\gamma \in \mathbb{R}$. But, clearly, for no value of β with $0 < \beta < 1$ the vectors $(1, -1 + \beta, -\beta)$ and $\gamma(0, 1, -1)$ belong to the same orthant of $\mathbb{R}^3$, which is a contradiction to $v \sqsubseteq u$. We conclude that for all $0 < \beta < 1$ the vector $(1, -1 + \beta, -\beta)$ belongs to $G_{MIP}(A)$. □

Having only an infinite test set available we cannot yet use the above augmentation algorithm to improve a feasible initial solution. However, we will construct a finite set $G_{\mathbb{Z}}(A) \subseteq \mathbb{Z}^{n_1}$ from which an improving vector to a non-optimal solution of the given problem can be reconstructed.

For this consider the projection $\phi : \mathbb{Z}^{n_1} \times \mathbb{R}^{n_2} \to \mathbb{Z}^{n_1}$ which maps each mixed-integer vector onto its n_1 integer components. Define $G_{\mathbb{Z}}(A) = G_{\mathbb{Z}}(A_1|A_2) := \phi(G_{MIP}(A))$ to be the set of images of elements in $G_{MIP}(A)$.

Lemma 3.14 *$G_{\mathbb{Z}}(A_1|A_2)$ is finite for every matrix $A \in \mathbb{Z}^{d\times n}$ and for any subdivision $A = (A_1|A_2)$.*

Proof. Each element $(z, q) \in G_{MIP}(A)$ satisfies $\|z\|_1 \leq \Delta(A_1, A_2)$ for some scalar $\Delta(A_1, A_2)$ depending only on A_1 and A_2 ([8], again for self-containedness this is reproduced in the appendix), which can be true only for finitely many $z \in \mathbb{Z}^{n_1}$. □

In [14], Köppe obtained the above finiteness result by a geometric description of the rational parts of mixed-integer test set elements when the integer parts are fixed. The analysis in [14] then also led to the conclusions we obtain in the subsequent lemmas.

Lemma 3.15 *Given b, c and a non-optimal feasible solution (z_0, q_0) to $A_1 z + A_2 q = b$, there exists $z_1 \in G_{\mathbb{Z}}(A)$ from which an improving vector (z_1, q_1) to (z_0, q_0) can be reconstructed.*

Proof. Since (z_0, q_0) is not optimal there is an improving vector $(z', q') \in G_{MIP}(A)$. Thus, an improving vector can be reconstructed from $z_1 := z' \in G_{\mathbb{Z}}(A)$. It remains to show that we can actually find z_1 and a suitable q_1 algorithmically.

For every $z_1 \in G_{\mathbb{Z}}(A)$ with $z_0 - z_1 \geq 0$ we try to find q_1 such that $(z_0 - z_1, q_0 - q_1)$ is feasible and $c^\intercal(z_0 - z_1, q_0 - q_1)$ is as small as possible. If $c^\intercal(z_0 - z_1, q_0 - q_1) < c^\intercal(z_0, q_0)$ then (z_1, q_1) is an improving vector.

This problem is equivalent to the optimization problem

$$q_1 := \arg\max_q \{c_z^\intercal z_1 + c_q^\intercal q : A_1(z_0 - z_1) + A_2(q_0 - q) = b, q_0 - q \geq 0, q \in \mathbb{R}^{n_2}\}. \tag{3}$$

Call the optimal solution q_1. Clearly, there is an improving vector for (z_0, q_0) in $G_{MIP}(A)$ starting with z_1 if and only if (z_1, q_1) is an improving vector, that is, if the above maximization problem has a feasible solution with strictly positive cost function value. $\square$

Corollary 3.16 *If for all $z_1 \in G_{\mathbb{Z}}(A)$ the maximal value of* (3) *is non-positive then (z_0, q_0) is optimal.*

Proof. If (z_0, q_0) is not optimal then there exists an improving vector $(z_1, q_1) \in G_{MIP}(A_1|A_2)$. But for this z_1 the maximal value of (3) is at least $c_z^\intercal z_1 + c_q^\intercal q > 0$ contradicting the assumption that no such z_1 exists. $\square$

We need to solve many LP-subproblems in order to reconstruct an improving vector. Compared to the original mixed-integer problem these LP-computations are cheap.

The advantages of our approach are clear. $G_{\mathbb{Z}}(A)$ is always a finite set of vectors from which for every given $b \in \mathbb{R}^d$ improving vectors to non-optimal solutions can be reconstructed. Moreover, this approach allows us to use powerful LP-solvers to take care of the continuous part of our mixed-integer problem.

Remark. This project-and-lift approach is not restricted to MIP-Graver test sets. It can be used for any mixed-integer test set, as well. $\square$

Termination of Augmentation Algorithm

Proposition 3.17 *If the relaxed problem $(LP)_{c,b}$ is bounded with respect to c then the MIP augmentation algorithm terminates.*

Proof. First transform the problem $(P)_{c,b}$ by Gaussian elimination into an equivalent optimization problem $(P)'_{c,b'}$ with problem matrix

$$A' := \begin{pmatrix} A_1' & A_2' \\ A_1'' & 0 \end{pmatrix},$$

wherein A_2' has full rank. Note that the right-hand-side b will change as well to b' but the cost function c remains fixed. Moreover, since the kernel of the problem matrix does not change, the finite set $G_{\mathbb{Z}}(A_1|A_2)$ remains unchanged after this transformation.

Now suppose on the contrary that the augmentation process does not terminate, i.e., it generates an infinite sequence of feasible solutions to $(P)'_{c,b'}$ with strictly decreasing objective value. Divide this sequence into consecutive subsequences of length $K + 1$ where K denotes the number of invertible submatrices B of A_2'. As we will see, within each of these subsequences the cost function value drops by at least some constant $\epsilon > 0$ depending only A_1', A_1'', A_2', and c. Since $(LP)_{c,b'}$ is bounded with respect to c this can happen only finitely often, and the MIP augmentation algorithm has to terminate after finitely many steps.

Since A' is a rational matrix the set $\text{conv}\{(z, q) \in \mathbb{Z}^{n_1} \times \mathbb{R}_+^{n_2} : A'(z, q) = b'\}$ is a polyhedron. After each augmentation step the continuous part q is a vertex of $\{q \in \mathbb{R}_+^{n_2} : A(z, q) = b'\}$ (or a point with the same cost function value). Thus, after possible rearrangement of continuous variables, we can write A_2' as $(B|\bar{A}_2)$ where B is an invertible $d \times d$-matrix and the optimal point is given by $(z, B^{-1}(b' - A_1'z), 0)$.

Let $c = (c_1, c_2, c_3)$ be divided analogously. Then the cost function value of $(z, B^{-1}(b' - A'_1 z), 0)$ is given by

$$c_1^\intercal z + c_2^\intercal B^{-1}(b' - A'_1 z) = (c_1^\intercal - c_2^\intercal B^{-1} A'_1) z + c_2^\intercal B^{-1} b' =: \tilde{c}^\intercal z + \tilde{c}_0$$

where $\tilde{c}$ and $\tilde{c}_0$ depend only on the given problem data and on the specific choice of B.

Consider a consecutive sequence of $K+1$ feasible solutions as constructed above. By the pigeon-hole principle there are two solutions $(z_1, B^{-1}(b' - A'_1 z_1), 0)$ and $(z_2, B^{-1}(b' - A'_1 z_2), 0)$ whose continuous parts are determined by the same submatrix B as already indicated by the notation. Moreover, let the second solution have a better cost function value than the first one.

Thus $0 < \tilde{c}^\intercal z_1 + \tilde{c}_0 - (\tilde{c}^\intercal z_2 + \tilde{c}_0) = \tilde{c}^\intercal (z_1 - z_2)$. In the MIP augmentation algorithm the vector $z_1 - z_2$ was represented as as a sum $\sum_{i=1}^{\leq K} g_i$ of at most K elements from $G_{\mathbb{Z}}(A') = G_{\mathbb{Z}}(A)$ since z_2 was obtained from z_1 by at most K augmentation steps. Since $G_{\mathbb{Z}}(A')$ is finite, there are only finitely many possible positive values of $\tilde{c}^\intercal(\sum_{i=1}^{\leq K} g_i)$. Thus, there exists a constant $\epsilon_B > 0$ depending only on the choice of B and the given data A'_1, A''_1, A'_2, and c, such that

$$\tilde{c}^\intercal (z_1 - z_2) = \tilde{c}^\intercal \left(\sum_{i=1}^{\leq K} g_i \right) \geq \epsilon_B > 0.$$

Therefore, the cost function value drops by at least $\epsilon_B > 0$ between the two mixed-integer solutions $(z_1, B^{-1}(b' - A'_1 z_1), 0)$ and $(z_2, B^{-1}(b' - A'_1 z_2), 0)$ in the augmentation algorithm.

Since there are only finitely many invertible submatrices of A'_2, $\epsilon := \min_B \{\epsilon_B\}$ is well defined and depends only on A'_1, A''_1, A'_2, and c, the given problem data. Moreover, we proved that within each subsequence of length $K+1$ the cost function value drops by at least $\epsilon_B \geq \epsilon > 0$, which implies that the augmentation algorithm has to terminate if $(LP)_{c,b'}$ is bounded. $\square$

4 Feasible Initial Solutions

So far we have addressed the problem of improving a given feasible solution for $(P)_{c,b}$ using directions given by test set elements. In doing so, the optimization problem can be solved in finitely many augmentation steps. But, particularly in the IP- and in the MIP-case, it is often already a very hard problem to find an initial feasible solution at all.

However, one can at least compute a (mixed-) integer solution to $Az = b$ in polynomial time, ignoring the lower bounds 0 on the variables ([9, 10, 11, 18]). Computing a solution in the LP-case can be done for example by Gaussian elimination or even faster methods.

In the following we will demonstrate how universal test sets can be used to transform any given solution to $Az = b, z \in \mathbb{X}$ into a feasible solution of $(P)_{c,b}$, that is a solution satisfying also the lower bounds $z \geq 0$. The proposed algorithm always terminates and returns either a feasible solution or the answer that no such solution exists.

Algorithm 4.1 *(Algorithm to find a Feasible Solution)*

Input: a solution $z_1 \in \mathbb{X}$ to $Az = b$, a universal test set $\mathcal{T}$ for $(P)_{c,b}$

Output: a feasible solution or "FAIL" if no such solution exists

while there exist $t \in \mathcal{T}$ and $\alpha \in \mathbb{R}_{>0}$ such that $z_1 - \alpha t \in \mathbb{X}$, $\|(z_1 - \alpha t)^-\|_1 < \|z_1^-\|_1$, and $(z_1 - \alpha t)^{(k)} \geq 0$ whenever $z_1^{(k)} \geq 0$ do

$z_1 := z_1 - \alpha t$, where α is chosen maximal such that all three conditions are still satisfied

if $\|z_1^-\|_1 > 0$ then return "FAIL" else return z_1

In the MIP-case we have to decide for every $g_z \in G_{\mathbb{Z}}(A)$ whether we can find g_q and a scalar $\alpha > 0$ such that $t := (g_z, g_q)$ has the desired properties. For this we distinguish two cases. Either we have $g_z = 0$ or $g_z \neq 0$. In both cases we can fix $\alpha = 1$ and are left with a pure LP-problem: Find $(g_z, g_q) \in \ker_{\mathbb{X}}(A)$ minimizing $\|(z_1 - \alpha g)^-\|_1 = \|z_1 - (g_z, g_q)^-\|_1$, where z_1 and g_z are given.

Lemma 4.2 *The Algorithm 4.1 satisfies its specifications.*

Proof. Suppose that the algorithm terminates and that at the end z_1 is still infeasible although there is some feasible solution z_0 to our problem. Consider the optimization problem

$$\min\{\sum_{i:z_1^{(i)}<0} (-z)^{(i)} : Az = b + Az_1^-, z \in \mathbb{X}_{\geq 0}\}. \tag{4}$$

Then z_1^+ is feasible for problem (4) and it admits an objective function value of 0. Moreover, $z_0 + z_1^-$ is feasible, too, with strictly negative objective function value, since by construction $z_1^{(k)} < 0$ implies $(z_0 + z_1^-)^{(k)} > 0$. By the universal test set property of $\mathcal{T}$, there have to exist a vector $t \in \mathcal{T}$ and a scalar $\alpha > 0$ such that $z_1^+ - \alpha t$ is feasible for problem (4) with strictly negative objective function value. This implies that $t^{(k)} \leq 0$ whenever $z_1^{(k)} \leq 0$, and that $\sum_{k:z_1^{(k)}<0} (-t)^{(k)} > 0$. We conclude that at least one member of that sum is strictly positive, which yields $\|(z_1 - \alpha t)^-\|_1 < \|z_1^-\|_1$. Moreover, if $z_1^{(k)} \geq 0$ then $(z_1^+ - \alpha t)^{(k)} \geq 0$ implies that also $(z_1 - \alpha t)^{(k)} = (z_1^+ - \alpha t)^{(k)} \geq 0$. Thus, the vector t and the scalar α satisfy the conditions of the while-loop which is a contradiction to z_1 being the output of the above algorithm. □

Lemma 4.3 *Algorithm 4.1 terminates in the IP- and MIP-case. The algorithm terminates in the LP-case if the following strategy similar to the Augmentation Strategy 3.10 is followed.*

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

Proof. The algorithm terminates in the IP-case since in each step $\|z_1^-\|_1$ drops by at least 1.

The algorithm terminates in the LP-case since, analogously to the termination proof of the LP Augmentation Strategy 3.10, every support of z_1 appears at most once at the end of phase 1.

Suppose that the algorithm does not terminate in the MIP-case and that it generates an infinite sequence $z_1, z_2, \ldots$ of points in $\mathbb{X}$. Note that all non-negative components of a point z_j remain non-negative for all subsequent points during the run of the above algorithm. Thus, since the algorithm does not terminate, there is a number N such that $\emptyset \neq \operatorname{supp}(z_N^-) = \operatorname{supp}(z_j^-)$ for all $j \geq N$. Now consider problem (4) where the cost function, however, is defined on the support of z_N^- instead of z_1^-. By construction, the cost function value is strictly positive for all z_j with $j \geq N$. However, since the algorithm does not terminate, we can use a similar argument as in the termination proof of the MIP augmentation algorithm to show that the cost function value drops below any given value within a finite number of steps. Thus, in particular, the cost function value becomes eventually negative for some z_j with $j \geq N$. From this contradiction we can conclude that the above algorithm terminates in the MIP-case, too. □

5 Truncated Graver Test Sets for $(IP)_{c,b}$

The notion of a truncated Graver test set already appeared in a paper by Thomas and Weismantel [22]. Our algorithm to compute a truncated Graver test set could be deduced from their approach as well by defining their partial order not on all components of the right-hand-side (b, u) but only on the components of u, the upper bounds.

Consider the family of optimization problems $\min\{c^\intercal z : Az = b, z \leq u, z \in \mathbb{Z}^n_{\geq 0}\}$ where only $b \in \mathbb{R}^d$ and $c \in \mathbb{R}^n$ are allowed to vary. The upper bounds $u \in (\mathbb{Z} \cup \infty)^n$ are assumed to remain fixed. Let $u^{(i)} = \infty$ if $z^{(i)}$ is not bounded. Prominent examples arise in combinatorics as 0-1-problems.

Example. Before we introduce truncated Graver test sets, let us demonstrate how the algorithm presented in section 3 deals with the upper bounds when it computes a Graver test set for this problem.

Introduction of slack variables leads to the computation of $G_{IP}(B)$, where

$$B = \begin{pmatrix} A & 0 \\ E & E \end{pmatrix}.$$

Each element of $\ker_{\mathbb{Z}^{n+d}}(B)$ has the form $(v, -v)$. Hence $(w, -w) \sqsubseteq (v, -v)$ if and only if $w \sqsubseteq v$. This shows that the computations of $G_{IP}(A)$ and $G_{IP}(B)$ are isomorphic using the correspondence $v \leftrightarrow (v, -v)$. That is, to speed up calculation of $G(B)$ we may compute $G(A)$ and finally map each element $v \in G_{IP}(A)$ to $(v, -v)$.

So, essentially, the bounds $z \leq u$ are ignored during the computation, which is due to the fact that u is assumed to vary as well. Thus, computations are in $\mathbb{Z}^{n+d}$ and do not exploit the given fixed upper bounds. □

However, the fixed bounds allow a truncation of the Graver test set $G_{IP}(A)$ for $\min\{c^\intercal z : Az = b, z \in \mathbb{Z}^n_{\geq 0}\}$ since only those vectors v with $|v^{(i)}| \leq u^{(i)}$ for all $i = 1, \ldots, n$, may transform a feasible solution $0 \leq z_1 \leq u$ into a feasible solution $z_2 = z_1 - v$ satisfying $0 \leq z_2 \leq u$ since $v = z_1 - z_2$.

The following lemma is an immediate consequence from the positive sum property of $G(A)$ with respect to $\ker_{\mathbb{Z}^n}(A)$.

Lemma 5.1 *The truncated Graver test set* $G_u(A) := G_{IP}(A) \cap \{z \in \mathbb{Z}^n : -u \leq z \leq u\}$ *is a universal test set for* $\min\{c^\intercal z : Az = b, z \leq u, z \in \mathbb{Z}^n_{\geq 0}\}$ *where only* $b \in \mathbb{R}^d$ *and* $c \in \mathbb{R}^n$ *may vary.*

Example. To see that the truncated Graver test set $G_u(A)$ may be much smaller than $G_{IP}(A)$ consider for given integer $k > 2$ the problem $\min\{c^\intercal z : kz_1 + z_2 + z_3 = b, z \leq 1, z \in \mathbb{Z}^3_{\geq 0}\}$. Then we have $G_{IP}(A) = \{\pm(0, 1, -1), \pm(1, 0, -k), \pm(1, -1, -(k-1)), \ldots, \pm(1, -k, 0)\}$ but $G_{(1,1,1)}(A) = \{\pm(0, 1, -1)\}$. Note that this truncated test set does not change if we do not impose a finite bound on z_1. □

Thus, truncation may drastically reduce the number of interesting vectors in a Graver test set. Of major importance, however, is the fact that $G_u(A)$ can be computed directly without computing the much bigger set $G_{IP}(A)$ first. In the following, we will present a more general algorithm which computes all $\sqsubseteq$-minimal solution in the set $\{z \in \mathbb{Z}^n : Az = 0, l \leq z \leq u\}$, where $l \leq 0 \leq u$. Thus, for special choices of l and u this includes algorithmic solutions to the following problems:

- For $l = 0$ and $u = 1$ (component-wise) the algorithm computes test sets for combinatorial optimization problems.
- For $l = -u$ the algorithm computes the desired truncated Graver test set $G_u(A)$.
- For $l = -\infty$ and $u = +\infty$ the algorithm is similar to Algorithm 2.1.7 in Urbaniak [23] and computes the IP Graver test set corresponding to A. However, it is much slower than Algorithm 2.10 (section 2), since the computation is done in a higher dimensional space.
- For $b = 0, l = 0$, and $u = +\infty$ the algorithm computes the unique minimal Hilbert basis for the pointed rational cone $\{z : Az = 0, z \in \mathbb{Z}^n_{\geq 0}\}$.

Analogously to Graver test sets one can prove that the set of all $\sqsubseteq$-minimal solutions in the set $\{z \in \mathbb{Z}^n : Az = 0, l \leq z \leq u\}$ has the positive sum property with respect to $\{z \in \mathbb{Z}^n : Az = 0, l \leq z \leq u\}$. Thus, it is not surprising that its computation follows the pattern of a completion procedure as given in Algorithm 2.10. Now we have to specify the input set, the reduction needed in the normalForm algorithm and the set of S-vectors.

As input set we choose $F = \bigcup_{i:l^{(i)}<0} \{(-e_i, -Ae_i)\} \cup \bigcup_{i:u^{(i)}>0} \{(e_i, Ae_i)\}$, where e_i denotes the i^{th} unit vector in $\mathbb{Z}^n$. Moreover, $s \in \mathbb{Z}^{n+d}$ reduces by $g \in \mathbb{Z}^{n+d}$ to $s - g$ if $g \sqsubseteq s$. Finally, we define

$$\text{S-vectors}((v, Av), (w, Aw)) := \begin{cases} \{(v+w, A(v+w))\} & \text{if } l \leq v+w \leq u \text{ and both, } v \text{ and } w, \\ & \text{lie in the same orthant of } \mathbb{Z}^n, \\ \emptyset & \text{otherwise.} \end{cases}$$

Lemma 5.2 *With the above definitions of input set, normalForm, and S-vectors, the Completion Algorithm 2.10 terminates and returns a set G which contains $(z_0, 0)$ for all $\sqsubseteq$-minimal integer solution z_0 of $Az = 0, l \le z \le u$.*

The set of all minimal integer solutions to $Az = 0, l \le z \le u$ is the set of all those vectors z such that $(z, 0) \in G$ and $(z, 0)$ is irreducible with respect to $G \setminus \{(z, 0)\}$.

The following proof is similar to that of Lemma 2.3. However, we use the special input set F to show that much fewer S-vectors need to be considered here.

Proof. Again, termination of the above algorithm follows immediately by applying the Gordan-Dickson Lemma to the sequence $\{(v^+, v^-) : v \in G \setminus F\}$.

For the correctness proof let G denote the set G at the end of the algorithm. We show that any non-zero $(z, 0)$ with $z \in \ker_{\mathbb{Z}^n}(A)$ and $l \le z \le u$ can be written as a positive linear integer combination of elements G, where each vector in this combination lies in the same orthant as $(z, 0)$. From this the claim immediately follows since there is exactly one such representation $(z, 0) = 1 \cdot (z, 0)$ if $z \in \ker_{\mathbb{Z}^n}(A)$ is minimal with respect to $\sqsubseteq$.

Since G contains the input set F, the vector $(z, 0)$ can be written as a positive linear integer combination $(z, 0) = \sum \alpha_i (v_i, Av_i)$ for $\alpha_i \in \mathbb{Z}_{>0}$ and vectors $(v_i, Av_i) \in G$ with $v_i \sqsubseteq z$ for all i. From the set of all such positive linear integer combinations $\sum \alpha_i (v_i, Av_i)$ choose one such that $\sum \alpha_i \|Av_i\|_1$ is minimal. Note that $\sum \alpha_i \|Av_i\|_1 \ge 0$ with equality if and only if $\|Av_i\|_1 = 0$ for all i.

If $\sum \alpha_i \|Av_i\|_1 = 0$ were true we could conclude $Av_i = 0$ for all i and we had found a desired linear representation of z. So assume to the contrary that $\sum \alpha_i \|Av_i\|_1 > 0$ holds.

Take $(v_{i_1}, Av_{i_1}), (v_{i_2}, Av_{i_2})$ such that $(Av_{i_1})^{(k_0)} (Av_{i_2})^{(k_0)} < 0$ for some component k_0. By construction, v_{i_1} and v_{i_2} lie in the same orthant of $\mathbb{Z}^n$ and from $v_{i_1} + v_{i_2} \sqsubseteq z$ we get $l \le -z^- \le v_{i_1} + v_{i_2} \le z^+ \le u$ and so the vector $(v_{i_1}, Av_{i_1}) + (v_{i_2}, Av_{i_2}) \in$ S-vectors $((v_{i_1}, Av_{i_1}), (v_{i_2}, Av_{i_2}))$ was reduced to 0 by G in the course of the algorithm which provides us with a representation $(v_{i_1}, Av_{i_1}) + (v_{i_2}, Av_{i_2}) = \sum \beta_j (v'_j, Av'_j)$ for some positive integers β_j and some $(v'_j, Av'_j) \in G$. Moreover, $\beta_j (v'_j, Av'_j) \sqsubseteq (v_{i_1}, Av_{i_1}) + (v_{i_2}, Av_{i_2})$ for all j, implying that $v'_j \sqsubseteq z$ and that for each component k

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

holds, where the last inequality is strict for $k = k_0$ by construction. Summation over k now gives

$$\sum \beta_j \|Av'_j\|_1 = \|A(v_{i_1} + v_{i_2})\|_1 < \|Av_{i_1}\|_1 + \|Av_{i_2}\|_1.$$

But then $(z, 0) = \sum \beta_i (v'_i, Av'_i) + (\alpha_1 - 1)(v_{i_1}, Av_{i_1}) + (\alpha_2 - 1)(v_{i_2}, Av_{i_2}) + \sum_{i \ne i_1, i_2} \alpha_i (v_i, Av_i)$ contradicts the minimality of $\sum \alpha_i \|Av_i\|_1$. We conclude that already $\sum \alpha_i \|Av_i\|_1 = 0$ and the claim follows. $\square$

Lemma 5.3 *If $A = (A'|E)$ then no additional variables are needed.*

Proof. The input vectors of the above algorithm become $(e_i, A'e_i, e_i)$. Thus, each vector appearing in the algorithm will have the form (v, Av, v) and $(w, Aw, w) \sqsubseteq (v, Av, v)$ if and only if $(w, Aw) \sqsubseteq (v, Av)$. So we can ignore the additional variables. $\square$

Remark. A similar proof works for $A = (A'| - E)$ or for the case, where only some equations arise from inequalities with the help of slack-variables. Thus we can avoid all or at least some of the additional variables in the algorithm. $\square$

6 Appendix

Lemma 6.1 *(Foroudi and Graver [8], General Decomposition Theorem)*
$G_{MIP}(A)$ has the positive sum property with respect to $\ker_{\mathbb{X}}(A)$.

Proof. Suppose that $(z, q) \in \ker_{\mathbb{X}}(A)$ cannot be written as a positive linear combination of elements in $G_{MIP}(A)$. We know that $\|z\|_1 > 0$ since $G_{LP}(A_2)$ has the positive sum property with respect to $\ker_{\mathbb{R}^{n_2}}(A_2)$. From all such vectors $(z, q) \in \ker_{\mathbb{X}}(A)$ choose one such that $\|z\|_1 + |\operatorname{supp}(q)|$ is minimal.

Now suppose that there is some circuit $q_1 \in G_{LP}(A_2)$ such that $\operatorname{supp}(q_1) \subseteq \operatorname{supp}(q)$ and such that $(0, q_1)$ lies in the same orthant as (z, q). Choose the largest $\alpha_1 \in \mathbb{R}_{>0}$ such that $(z, q - \alpha_1 q_1)$ still belongs to the same orthant as (z, q). Thus, $\operatorname{supp}(q - \alpha_1 q_1) \subsetneq \operatorname{supp}(q)$ and consequently

$$\|z\|_1 + |\operatorname{supp}(q - \alpha_1 q_1)| < \|z\|_1 + |\operatorname{supp}(q)|.$$

By the minimality required on $\|z\|_1 + |\operatorname{supp}(q)|$ we can conclude that there is a linear representation $(z, q - \alpha_1 q_1) = \sum \beta_j (z_j, q_j)$ where for all j we have $\beta_j (z_j, q_j) \in \mathbb{X}$, $\beta_j (z_j, q_j) \sqsubseteq (z, q - \alpha_1 q_1)$ and $\beta_j \in \mathbb{R}_{>0}$. Hence $(z, q) = \alpha_1 q_1 + \sum \beta_j (z_j, q_j)$ is a valid representation of (z, q) in contrast to our initial assumption.

Thus, we may assume that there is no vector $(0, q_1) \in \ker_{\mathbb{X}}(A)$ with $(0, q_1) \sqsubseteq (z, q)$.

Clearly, $(z, q) \notin G_{MIP}(A)$, and we conclude that there is some $(z', q') \in \ker_{\mathbb{X}}(A)$ with $(z', q') \sqsubseteq (z, q)$. Hence $(z, q) = (z', q') + (z - z', q - q')$, where also $(z - z', q - q') \sqsubseteq (z, q)$. This implies in particular, that $\operatorname{supp}(q') \subseteq \operatorname{supp}(q)$ and $\operatorname{supp}(q - q') \subseteq \operatorname{supp}(q)$.

But neither $z' = 0$ nor $z - z' = 0$, by construction. Hence $\|z'\|_1 < \|z\|_1$ and $\|z - z'\|_1 < \|z\|_1$. So, by the minimality assumption on $\|z\|_1 + |\operatorname{supp}(q)|$, both (z', q') and $(z - z', q - q')$ can be written as a positive linear combination of elements from $G_{MIP}(A)$, all of which belong to the same orthant as (z', q') and $(z - z', q - q')$, respectively.

Substituting these representations into $(z, q) = (z', q') + (z - z', q - q')$ gives a valid positive linear combination representing (z, q), contradicting our initial assumption that no such representation exists. $\square$

Lemma 6.2 *(Foroudi and Graver [8], Lemma 13)*
Given $A = (A_1|A_2)$ the following inequality holds for any $(z_0, q_0) \in G_{MIP}(A_1|A_2)$:

$$\|z_0\|_1 \leq \sum_{(z,q) \in G_{LP}(A)} \|z\|_1.$$

Proof. Let $(z_0, q_0) \in G_{MIP}(A_1|A_2) \subseteq \mathbb{R}^n$ be given. The positive sum property of $G_{LP}(A)$ with respect to $\ker_{\mathbb{R}^n}(A)$ yields a finite representation $(z_0, q_0) = \sum \alpha_i (z_i, q_i)$ where $(z_i, q_i) \in G_{LP}(A)$ and $\alpha_i \in \mathbb{R}_{>0}$.

Suppose that there exists a summand $\alpha_j (z_j, q_j)$ in this representation with $z_j \neq 0$ and $\alpha_j > 1$. From $(z_j, q_j) \in G_{LP}(A) \subseteq \mathbb{Z}^n$ we conclude that

$$(z_j, q_j) \in \mathbb{X} \qquad \text{and} \qquad (\alpha_j - 1)(z_j, q_j) + \sum_{i \neq j} \alpha_i (z_i, q_i) = (z_0, q_0) - (z_j, q_j) \in \mathbb{X}.$$

By construction, both vectors have non-zero integer parts, $(z_j, q_j) \sqsubseteq (z_0, q_0)$, $(z_0 - z_j, q_0 - q_j) \sqsubseteq (z_0, q_0)$, and $(z_0, q_0) = (z_j, q_j) + (z_0 - z_j, q_0 - q_j)$. Hence $(z_0, q_0) \notin G_{MIP}(A_1|A_2)$.

Thus, $(z_0, q_0) \in G_{MIP}(A_1|A_2)$ implies that $(z_0, q_0) = \sum \alpha_i (z_i, q_i)$ with $\alpha_j \leq 1$ whenever $z_j \neq 0$. Consequently,

$$\|z_0\|_1 \leq \sum_i \|\alpha_i z_i\|_1 \leq \sum_{i: z_i \neq 0} \|z_i\|_1 + \sum_{i: z_i = 0} \|\alpha_i z_i\|_1 = \sum_{i: z_i \neq 0} \|z_i\|_1 \leq \sum_{(z,q) \in G_{LP}(A)} \|z\|_1$$

and the claim is proved. $\square$

Remark. With the additional information that z_0 belongs to some given orthant $\mathbb{O}_j$ the above estimate can be strengthened to

$$\|z_0\|_1 \leq \sum_{(z,q) \in G_{LP}(A), z \in \mathbb{O}_j} \|z\|_1.$$

$\square$

7 Conclusions

We have discussed the positive sum property, a property inherent to Graver test sets in LP, IP, and MIP. Already Graver [12] showed that this property implies the universal test set property. We added to this analysis by presenting two criteria which allow an *algorithmic* test of the positive sum property with respect to $\ker_{\mathbb{R}^n}$ or $\ker_{\mathbb{Z}^n}$. These criteria led us immediately to a completion algorithm: an initial set of vectors is completed with respect to the positive sum property by adding new vectors to the set as long as necessary. Critical pair/completion procedures arise in automated theorem proving, polynomial ideal theory and in the solution of word problems in universal algebras, see Buchberger [3] for a survey.

Via the positive sum property the notion of a Graver test sets can be extended to the mixed-integer situation. Although these test sets need not be finite in general we have identified a finite set of integer vectors from which an improving vector can be recomputed. In order to do so one has to solve a finite number of LP's.

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.

It is usually already a hard problem to find an initial feasible point of $(P)_{c,b}$ when some or all variables are required to be integer. Although there are algorithmic alternatives, we have shown that universal test sets contain enough information to solve this problem, too. Thus, once a universal test set is available, the problem $(P)_{c,b}$ can be solved by test set methods only.

Finally we have demonstrated how fixed (finite) upper bounds on variables can be used to truncate the Graver test set of $(IP)_{c,b}$ and to speed up its computation. The presented algorithm can also be used to compute the unique minimal Hilbert basis of $\ker_{Z^n}(A) \cap \mathbb{Z}^n_{\geq 0}$ if the input set is chosen accordingly.

Acknowledgement. The author is grateful to Jack E. Graver, Syracuse University, for giving access to the unpublished manuscript [8], and to Rüdiger Schultz, University of Duisburg, for stimulating discussions.

References

[1] A.M. Bigatti, R. LaScala, and L. Robbiano. Computing toric ideals. Journal of Symbolic Computation, 27:351–365, 1999.

[2] 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.

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

[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] W. Cook, A. M. H. Gerards, A. Schrijver, and É. Tardos. Sensitivity theorems in integer linear programming. *Mathematical Programming*, 34:251–264, 1986.

[6] 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.

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

[8] A. D. Foroudi and J. E. Graver. On the foundations of mixed-integer programming. *unpublished manuscript*, Department of Mathematics, Syracuse University, 1989.

[9] M. A. Frumkin. An application of modular arithmetic to the construction of algorithms for solving systems of linear equations. (in Russian) Doklady Akademii Nauk SSSR 222:1067–1070, 1976. (English translation: Soviet Mathamatics Doklady 17:1165-1168, 1976.)

[10] M. A. Frumkin. Algorithms for the solution in integers of systems of linear equations. (in Russian) in: Issledovaniya po diskretnoi optimizatsii, (A. A. Fridman, ed.), Izdat. "Nauka", Moscow, 97–127, 1976.

[11] J. von zur Gathen and M. Sieveking. Weitere zum Erfüllungsproblem polynomial äquivalente kombinatorische Aufgaben. in: *Komplexität von Entscheidungsproblemen*, E. Speckers and V. Strassen, eds., LNCS 43:49–71, Springer, Heidelberg, 1976.

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

[13] 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.

[14] M. Köppe. Erzeugende Mengen für gemischt-ganzzahlige Programme. Diploma thesis, Otto-von-Guericke-Universität Magdeburg, August 1999. Available from http://www.math.uni-magdeburg.de/~mkoeppe/art/mkoeppe-diplom.ps.

[15] M. Köppe. Integral Decomposition of Polyhedra and Test Sets for Mixed-Integer Programming. talk given at the Workshop on Combinatorial Optimization, Aussois, March 2000.

[16] J. G. Oxley. *Matroid Theory.* Oxford University Press, 1992.

[17] L. Pottier. Euclide's algorithm in dimension n. Research report, ISSAC 96, ACM Press, July 1996.

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

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

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

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

[22] R. R. Thomas and R. Weismantel. Truncated Gröbner Bases for Integer Programming. *Applicable Algebra in Engineering, Communication and Computing*, 8:241–257, 1997.

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

[24] 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.

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