Algebra for Machine Learning and Stochastic Programming

Yuchen Ge

August 2022

Contents

Abstract

This article determines the relationship between Gröbner Basis of Integer Programming and Gröbner Basis of Stochastic Integer Programming.

1 What is a Stochastic Integer Programming(SIP)?

Instead of SIP (A universally acknowledged definition is unavailable), we give the definition of Stochastic Mixed Integer Program.

Definition 1. A Stochastic Mixed Integer Program (SMIP) is to solve   *i.e. linear*

$$\min c^T x + \mathbb{E}[Q(x,\xi)]$$
$$\text{s.t. } Ax \geq b \tag{1}$$
$$x \in \mathbb{R}^{n_1}_+ \times \mathbb{Z}^{p_1}_+$$

where $\xi = (q, h, T, W)$ and

$$Q(x,\xi) = \min q^T y$$
$$\text{s.t. } Wy = h - Tx$$
$$y \in \mathbb{R}^{n_2}_+ \times \mathbb{Z}^{p_2}_+$$

where x denotes the first-stage decision variables and y denotes the second-stage decision variables and sometimes we assume n_1, p_1, n_2 , or p_2 are zero, which depends.

With Sample Average Approximation, we assume the random data ξ is represented by a finite set of scenarios: $(q^i, h^i, T^i, W^i), i = 1, \ldots, N$, where scenario i occurs with probability p_i.

So we have the following extensive form (deterministic equivalent) of an SMIP. *(pulling min over y to the front)*

Definition 2. Deterministic equivalent of an SMIP is

$$\min \quad c^T x + \sum_{i=1}^{N} p_i q_i^T y_i$$
$$\text{s.t.} \quad Ax \geq b$$
$$T_i x + W_i y_i = h_i \tag{2}$$
$$x \in \mathbb{R}^{n_1}_+ \times \mathbb{Z}^{p_1}_+$$
$$y_i \in \mathbb{R}^{n_2}_+ \times \mathbb{Z}^{p_2}_+, \quad i = 1, \ldots, N$$

purely combinatorial

When $n_1 = 0$ and $n_2 = 0$, we have Method 1 to deal with it. Moreover, a Stochastic Program with Complete Integer Recourse is a SMIP when $p_1 = n_2 = 0$ with method 2 to deal with.

first stage continuous, second stage purely combinatorial

2 How can GB/AG methods help solve SIP?

2.1 IP is a special case of SIP

An Integer Programming is a special case of SIP. Actually we can think of an Integer Programming problem as a SIP with only 1 scenario and without the first-stage decision. Writing as

(or c=0)

$$\min \quad p_1 q_1^T y_1$$
$$\text{s.t.} \quad W_1 y_1 = h_1 \tag{3}$$
$$y_1 \in \mathbb{Z}^{p_2}_+, \quad s = 1, \ldots, S$$

where the probability $p_1 = 1$.

Then we recall how can GB/AG methods help solve SIP.

2.2 Method 1: Application of Graver Basis[4]

2.2.1 Prerequisites on Test Sets

First before starting our analysis of stochastic programs, we collect some necessary prerequisites on test sets. For given $A \in \mathbb{Q}^{l \times d}$ (WLOG, we can transform the situation to $A \in \mathbb{Z}^{l \times d}$) consider the family of optimization problems

$$(IP)_{c,b}: \quad \min \left\{ c^T z : Az = b, z \in \mathbb{Z}_+^d \right\} = \alpha_{c,b}.$$

as $c \in \mathbb{R}^d$ and $b \in \mathbb{R}^l$ vary.

Definition 3. A set $\mathcal{T}_c \subseteq \mathbb{Z}^d$ is called a test set for the family of problems $(IP)_{c,b}$ as $b \in \mathbb{R}^l$ varies if
  1. $c^\top t > 0$ for all $t \in \mathcal{T}_c$
  2. for every $b \in \mathbb{R}^l$ and for every non-optimal feasible solution $z_0 \in \mathbb{Z}_+^d$ to $Az = b$, there exists a vector $t \in \mathcal{T}_c$ such that $z_0 - t$ is feasible. Such a vector is called an improving vector or an improving direction.

Definition 4. A set $\mathcal{T}$ is called a universal test set for the family of problems $(IP)_{c,b}$ as $b \in \mathbb{R}^l$ and as $c \in \mathbb{R}^d$ vary if it contains a test set $\mathcal{T}_c$ for every $c \in \mathbb{R}^d$.

We can relate them as: Test Set $\subset$ Universal Test Set. Then we have

Algorithm 1. Augmentation Algorithm
Input: a feasible solution z_0 to $(IP)_{c,b}$, a test set $\mathcal{T}_c$ for $(IP)_{c,b}$
Output: an optimal point $z_{\min}$ of $(IP)_{c,b}$
while there is $t \in \mathcal{T}_c$ with $c^T t > 0$ such that $z_0 - t$ is feasible do

$$z_0 := z_0 - t$$

return: z_0

Algorithm 2. Algorithm to Find a Feasible Solution
Input: a solution $z_1 \in \mathbb{Z}^d$ to $Az = b$, a universal test set $\mathcal{T}$ for $(IP)_{c,b}$
Output: a feasible solution to $(IP)_{c,b}$ or "FAIL" if no such solution exists
while there is some $g \in \mathcal{T}$ such that $g \le z_1^+$ and $\left\| (z_1 - g)^- \right\|_1 < \left\| z_1^- \right\|_1$ do

$$z_1 := z_1 - g$$

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

Finally by applying the above two algorithms we can easily find an optimum of $(IP)_{c,b}$.

Algorithm 3. Algorithm to Find an Optimum of $(IP)_{c,b}$
Input: $(IP)_{c,b}$, a finite universal test set $\mathcal{T}$ for $(IP)_{c,b}$
Output: an optimum solution to $(IP)_{c,b}$ or "FAIL" if problem is not solvable
$z_1 :=$ solution to $Az = b$, $z \in \mathbb{Z}^d$
if no such solution exists then return "FAIL"
$z_0 :=$ feasible solution $(z_1, \mathcal{T})$ (Algorithm 2)
if no such solution exists then return "FAIL"
$z_{\min} :=$ optimal solution $(z_0, c, \mathcal{T})$ (Algorithm 1)
return: $z_{\min}$

2.2.2 Construction of IP Graver Test Set

In the following we present a particular universal test set, the IP Graver test set, and show how to compute it. Let us first look at the Hilbert basis, and then use it construct the IP Graver basis.

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

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

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

[handwritten: please first come out the definition]
[handwritten: $\Rightarrow \lambda_j \in \mathbb{Z}_{>0}$?]

Then we define Graver basis, a special case of Gröbner Basis.

Definition 6. Let O_j denote the j-th orthant in $\mathbb{R}^n$ where $j \in \{+, -\}^n$. Consider the cones $C_j = \{u \in \mathbb{Z}^n : A.u = 0\} \cap O_j$ for $j \in \{+, -\}^n$. Let H_j be the unique, minimal Hilbert basis of non-zero cone C_j and then we define the Graver Basis of A to be $\mathcal{GR}_A = \bigcup H_j \backslash \{0\}$.

[handwritten: (i.e. 2^n of those)]
[handwritten: the]
[handwritten: do we know this or do we assume this?]

We shall prove the proposition below.

[handwritten: why would $0 \in H_j$? seems redundant.]

Proposition 2. The Graver basis is a universal test set.

[handwritten: for which IP?]

2.2.3 (Optional) relationship between the reduced Gröbner bases of $(IP)_{c,b}$ and the Graver Basis $\mathcal{GR}_A$

[handwritten: what does this mean? Just fix arbitrary c?]

Proposition 2. The Graver basis is a universal test set.
Proof. Let c be a generic refined cost function and let α be a non-optimal integer point in some fiber of $(IP)_{c,b}$ in which β is the unique optimum. Then $\alpha - \beta$ lies in one of the cones C_j described above. Let H_j be the Hilbert basis of this cone. Then $\alpha - \beta = \sum z_{j_i} h_{j_i}$ where z_{j_i} are non-negative integers and the elements h_{j_i} belong to $H_j \subset \mathcal{GR}_A$. Since $c \cdot \alpha > c \cdot \beta$, there exists some h_{j_i} from the sum above such that $ch_{j_i} > 0$ which allows movement from α. Therefore $\mathcal{GR}_A$ contains a test set for $(IP)_{c,b}$ with respect to every cost function c and so is a universal Gröbner basis for $(IP)_{c,b}$.

[handwritten: $A^{-1}(b)$?]

Proposition 3. Let $\mathcal{G}_{A,c}$ be the reduce Gröbner basis of $(IP)_{\{c,b\}}$. Then

$$\bigcup_{c \in \mathbb{Z}^n} \mathcal{G}_{A,c} \subseteq \mathcal{GR}_A.$$

[handwritten: why does a unique optimum exist? (also why needed for the proof?)]

[handwritten margin: I) Fix arbitrary c, want that $\mathcal{GR}_A$ contains a TS for $(IP)_{c}$ (any b). II) Need to identify T_c. Fix b. Construct the ... content ... one of them has $ch_{j_i}>0$. Then $T_c = \{h_{j_i},b\} : b \in \mathbb{R}\} \subseteq H_j \subseteq \mathcal{GR}_A$ ...]

Proof. Suppose $\alpha - \beta$ lies in the pointed cone C_j with Hilbert basis H_j. If $\alpha - \beta$ is not an element of H_j, then $\alpha - \beta = v + w$ for $v, w \in C_j \cap \mathbb{Z}^n$. At least one of $cv > 0$ or $cw > 0$ since $c(\alpha - \beta) > 0$. Therefore we can move from α to a less expensive point using one of v or w. This contradicts that α was an essential generator of the set of non-optimal points in the various fibers of $(IP)_{c,b}$. □

So intuitively, a Graver basis is a larger Universal Gröbner Basis.

We give an important lemma important in the following context, which is the following extension of Gordan-Dickson Lemma ([[9], p.44])

Lemma 1.(Gordan-Dickson Lemma, $\sqsubseteq$-version)
1. Every sequence $\{p_1, p_2, \ldots\}$ of points in $\mathbb{Z}^n$ such that $p_i \not\sqsubseteq p_j$ whenever $i < j$ is finite.
2. Every infinite set $S \subseteq \mathbb{Z}^n$ contains only finitely many $\sqsubseteq$-minimal points.

[handwritten: do you mean $\not\sqsubseteq$?]
[handwritten: I don't understand this statement. If the $p_i \in \mathbb{Z}^d$, how can $p_i \sqsubseteq p_i$ make sense?]
[handwritten: see Def. 7: that definition should be before this Lemma]

By constricting $\mathbb{Z}^n$ to $\mathbb{Z}^n_+$ we come back to the classical version of Gordan-Dickson Lemma. For completeness, we also give the monomial-version of the lemma which will be used later.

Lemma 2.(Gordan-Dickson Lemma, monomial-version)
1. Every sequence $\{I_1, I_2, \ldots\}$ of monomial ideals in a polynomial ring with $I_j \not\subset I_i$ whenever $i < j$ is finite.
2. Every infinite collection of $\mathcal{I}$ of monomial ideals in a polynomial ring contains only finitely many inclusion maximal monomial ideals.

it's better to state all definitions where they are used.

Lemma 3. $\mathcal{GR}_A$ is the inclusion minimal subset of $\ker_{\mathbb{Z}^d}(A)$ that has the positive sum property with respect to $\ker_{\mathbb{Z}^d}(A)$. (Definition of positive sum property is in the appendix.)

Proof. Take any non-zero element $z \in \ker_{\mathbb{Z}^d}(A)$. Then z belongs to some orthant $\mathbb{O}_j$ and thus can be written as a positive integer linear combination of elements of the Hilbert basis $H_j \subseteq \mathcal{GR}_A$ of $\ker_{\mathbb{Z}^d}(A) \cap \mathbb{O}_j$. As, by construction, each element $z \in \mathcal{GR}_A$ cannot be written as a non-trivial sum of two vectors from $\ker_{\mathbb{Z}^d}(A)$ that lie in the same orthant as z, z must be contained in every set that has the minimal sum property with respect to $\ker_{\mathbb{Z}^d}(A)$. This proves minimality of $\mathcal{GR}_A$. $\qquad \square$

Theorem 1. $\mathcal{GR}_A$ is finite.

Proof. From lemma 3, we give an alternative definition of the Graver basis of an integer matrix A: **the finite set of $\sqsubseteq$-minimal elements in** $\{u \in \mathbb{Z}^n : A \cdot u = 0, u \neq 0\}$. Then we apply the famous Gordan-Dickson Lemma ($\sqsubseteq$-version) above. $\qquad \square$

2.2.4 Decomposition of Graver Test Sets for Two-Stage Stochastic Integer Programs

To solve the SIP, which is equivalent to the mixed-integer linear program *Only h is random here, right?*

$$\min \left\{ c^T x + \sum_{i=1}^{S} p_i q^T y_i : Ax = b, x \in \mathbb{Z}_+^d, Tx + Wy_i = h_i, y_i \in \mathbb{Z}_+^d, i = 1, \ldots, N \right\} \qquad (4)$$

where we assume all entries in A, T , and W to be rational (integral). *Is this a necessary condition for this to work? Restricted a for...*

To study Graver test sets of (4) we consider the matrix with N denotes the number of scenarios:

$$A_N := \begin{pmatrix} A & 0 & 0 & \cdots & 0 \\ T & W & 0 & \cdots & 0 \\ T & 0 & W & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ T & 0 & 0 & \cdots & W \end{pmatrix}$$

Problem (4) may then be written as $\min \left\{ c^T z : A_N z = b, z \in \mathbb{Z}_+^d \right\}$ with $d = m + N \cdot n$. For completeness, we shall write down the Algorithm to Compute Integer Programming Graver Test Sets as below.

Definition 7. Assume $a = (a_1, a_2, ..., a_n)$ and $b = (b_1, b_2, ..., b_n)$ in $\mathbb{Z}^n$. Then $a \sqsubseteq b$ if and only if $\forall i: a_i b_i \geq 0$ and $|a_i| \leq |b_i|$.

state this before Lemma 1.

Algorithm 4. (Normal Form Algorithm) *vectors in what space? $\mathbb{Z}^d$?*
Input: a vector s , a set G of vectors
Output: a normal form of s with respect to G
while there is some $g \in G$ such that $g \sqsubseteq s$ do
$s := s - g$
return: s *please define this.*

Algorithm 5. (Algorithm to Compute IP Graver Test Sets) *where do we get this from?*
Input: $F = \bigcup_{f \in F(A)} \{f, -f\}$, where F(A) is a set of vectors generating ker(A) over $\mathbb{Z}$
Output: a set G which contains the IP Graver test set $\mathcal{G}(A)$
$G := F$
$C := \bigcup_{f,g \in G} \{f + g\}$
while $C \neq \emptyset$ do
$\quad s :=$ an element in C
$\quad C := C \setminus \{s\}$
$\quad f := \mathrm{normalForm}(s, G)$

<u>if</u> $f \neq 0$ then (forming S-vectors)

$$C := C \cup \bigcup_{g \in G} \{f + g\} \quad \text{(adding S-vectors)}$$
$$G := G \cup \{f\}$$

<u>return:</u> G

From the lemma below, we can have a glimpse of the decomposition method.

Lemma 3. $(u, v_1, \ldots, v_N) \in \ker(A_N)$ if and only if $(u, v_1), \ldots, (u, v_N) \in \ker(A_1)$.

Definition 8. Let $z = (u, v_1, \ldots, v_N) \in \ker(A_N)$ and call the vectors $u, v_1, \ldots, v_N$ the building blocks of z. Denote by $\mathcal{G}_N$ the Graver test set associated with A_N and collect into $\mathcal{H}_N$ all those vectors arising as building blocks of some $z \in \mathcal{G}_N$. By $\mathcal{H}_\infty$ denote the set $\bigcup_{N=1}^\infty \mathcal{H}_N$.

The set $\mathcal{H}_\infty$ contains both m-dimensional vectors u associated with the first-stage in (4) and n-dimensional vectors v related to the second-stage in (4). For fixed $u \in \mathcal{H}_\infty$, all those vectors $v \in \mathcal{H}_\infty$ are collected into V_u for which $(u, v) \in \ker(A_1)$. For convenience, we will arrange the vectors in $\mathcal{H}_\infty$ into pairs (u, V_u).

Theorem 2. $\mathcal{H}_\infty$ is finite.

To prove it, we first give a definition of inclusion (pairs version).

Definition 9. We define $(u', V_{u'})$ reduces (u, V_u), or $(u', V_{u'}) \sqsubseteq (u, V_u)$, if the following conditions are satisfied:
1. $u' \sqsubseteq u$,
2. for every $v \in V_u$ there exists a $v' \in V_{u'}$ with $v' \sqsubseteq v$.
3. $u' \neq 0$ or there exist vectors $v \in V_u$ and $v' \in V_{u'}$ with $0 \neq v' \sqsubseteq v$.

Definition 10. We associate with $(u, V_u), u \neq 0$, the monomial ideal $I(u, V_u) \in Q[x_1, \ldots, x_{2m+2n}]$ generated by all the monomials $x^{(u^+, u^-, v^+, v^-)}$ with $v \in V_u$, whereas we associate with $(0, V_0)$ the monomial ideal $I(0, V_0) \in Q[x_1, \ldots, x_{2n}]$ generated by all the monomials $x^{(v^+, v^-)}$ with $v \neq 0$ and $v \in V_0$.

Lemma 4. Let $(u, V_u), (u', V_{u'})$ with $u, u' \neq 0$. Then $I(u, V_u) \subseteq I(u', V'_u)$ implies $(u', V_{u'}) \sqsubseteq (u, V_u)$.
Proof. Since both $I(u, V_u)$ and $I(u', V_{u'})$ are monomial ideals, we have $I(u, V_u) \subseteq I(u', V_{u'})$ iff. every generator $x^{(u^+, u^-, v^+, v^-)}$ of $I(u, V_u)$ is divisible by some generator $x^{((u')^+, (u')^-, (v')^+, (v')^-)}$ of $I(u', V_{u'})$ (cf. [3]). The latter implies that $u' \sqsubseteq u$ and that for every $v \in V_u$ there exists $v' \in V_{u'}$ with $v' \sqsubseteq v$. In other words, we have $(u', V_{u'}) \sqsubseteq (u, V_u)$. □

In a parallel way, we can prove

lemma 5. $I(0, V_0) \subset I(0, V'_0)$ implies $(0, V'_0) \sqsubseteq (0, V_0)$.

Now we are in the position to prove the mentioned lemma on sequences of pairs.

Lemma 6. Let $\{(u_1, V_{u_1}\}, (u_2, V_{u_2}), \ldots)$ be a sequence of pairs such that $u_i \neq 0$ for all $i = 1, 2, \ldots$, and such that $(u_i, V_{u_i}) \not\sqsubseteq (u_j, V_{u_j})$ whenever i¡j. Then this sequence is finite.
Proof. Consider the sequence $\{I(u_1, V_{u_1}), I(u_2, V_{u_2}), \ldots\}$ of monomial ideals. By Lemma 3.9, it fulfils $I(u_j, V_{u_j}) \not\subseteq I(u_i, V_{u_i})$ whenever $i < j$. Thus, by Corollary 3.6, this sequence is finite implying that the sequence $((u_1, V_{u_1}), (u_2, V_{u_2}), \ldots)$ is finite, as well. □

In a parallel way, we can also prove

Lemma 7. Let $\{(0, V_1), (0, V_2), \ldots\}$ be a sequence of pairs such that such that $(0, V_i) \not\sqsubseteq (0, V_j)$ whenever $i < j$. Then this sequence is finite.

Together with above two lemmas we can prove the following lemma.

Lemma 8. Let $((u_1, V_{u_1}), (u_2, V_{u_2}), \ldots)$ be a sequence of pairs such that $(u_i, V_{u_i}) \not\sqsubseteq (u_j, V_{u_j})$ whenever $i < j$. Then this sequence is finite.
Proof. Suppose the sequence $((u_1, V_{u_1}), (u_2, V_{u_2}), \ldots)$ is not finite. Consider the subsequence where all u_i are non-zero and the subsequence where all u_i are zero. One of these subsequences is not finite and satisfies $(u_i, V_{u_i}) \not\sqsubseteq (u_j, V_{u_j})$ whenever $i < j$. But this contradicts one of the two preceding lemmas. $\square$

A result from the finite steps of algorithm 7 below.

Definition 11. For fixed $u \in \mathcal{H}_\infty$, all those vectors $v \in \mathcal{H}_\infty$ are collected into V_u for which $(u, v) \in \ker(A_1)$.

By generalizing the operators to pairs, we can compute $\mathcal{H}_\infty$ as below.

Definition 12. The plus of pairs is $(u, V_u) \oplus (u', V_{u'}) := (u + u', V_u + V_{u'})$ and correspondingly the minus of pairs to be $(u, V_u) \ominus (u', V_{u'}) := (u - u', \{v - v' : v \in V_u, v' \in V_{u'}, v' \sqsubseteq v\})$.

Algorithm 6. (Normal Form Algorithm)
Input: a vector s, a set G of pairs
Output: a normal form of s with respect to G
while there is some $g \in G$ such that $g \sqsubseteq s$ do
$s := s \ominus g$
return: s

Algorithm 7. (Algorithm to Compute $\mathcal{H}_\infty$ (building blocks of IP Graver Test Sets))
Input: a generating set F of $\ker(A_1)$ which contains a generating set for $\{(0, v) : Wv = 0\} \subset \ker(A_1)$
Output: a set G which contains $\mathcal{H}_\infty$
$G := F$
$C := \bigcup_{f,g \in G} \{f \oplus g\}$
while $C \neq \emptyset$ do
    $s :=$ an element in C
    $C := C \backslash \{s\}$
    $f :=$ normalForm(s, G)
    if $f \neq (0, \{0\})$ then (forming S-vectors)

$$C := C \cup \bigcup_{g \in G} \{f \oplus g\} \quad \text{(adding S-vectors)}$$
$$G := G \cup \{f\}$$

    return: G

Theorem 3. Algorithm 6 and 7 terminates and satisfies its specifications.
Proof. In the course of the algorithm, a sequence of pairs f in $G \backslash F$ is generated that satisfies the conditions of Lemma 8. Therefore, Algorithm 7 terminates.

To show that $\mathcal{H}_\infty \subseteq G$, we have to prove that $\mathcal{H}_N \subseteq G$ for all $N \in \mathbb{Z}_+$. Fix N and start a Graver test set computation (Algorithm 2.7) with $\bar{F} := \{(u, v_1, \ldots, v_N) : (u, V_u) \in G, v_i \in V_u\}$ as input set. $\bar{F}$ generates $\ker(A_N)$ over $\mathbb{Z}$ for all $N \in \mathbb{Z}_+$, since $F_N \subseteq \bar{F}$ by the assumption on the input set to Algorithm 3.15, cf. Lemma 3.17.

It's easy to show that all sums $z + z'$ of two elements $z, z' \in \bar{F}$ reduce to 0 with respect to $\bar{F}$. In this case, Algorithm 2.7 returns the input set $\bar{F}$ which implies $\mathcal{G}_N \subseteq \bar{F}$ ($\mathcal{G}_N$ denotes the Graver test set associated with A_N). Therefore, $\mathcal{H}_N \subseteq G$ as desired. $\square$

2.2.5 Algorithm to Two-Stage Stochastic Integer Programs with $\mathcal{H}_\infty$

Suppose we are given an integer solution $z_1 = (u, v_1, \ldots, v_N)$ to $A_N z = b$.

Lemma 9. Suppose $z_1 = (u, v_1, \ldots, v_N) \ngeq 0$ and there exists no pair $(u', V_{u'}) \in \mathcal{H}_\infty$ such that

1. $u' \leq u^+$
2. for all $i = 1, \ldots, N$ there exists $\bar{v}_i \in V_{u'} : \bar{v}_i \leq v_i^+$,
3. $(c')^T z' > 0$, where $z' = (u', v_1', \ldots, v_N')$ and $v_i' \in \mathrm{argmax} \left\{ (c_i')^T v : v \leq v_i^+, v \in V_{u'} \right\}$ for $i = 1, \ldots, N$.

Then there exists no feasible solution of $A_N z = b, z \in \mathbb{Z}_+^{m+Nn}$. If there exists such a pair $(u', V_{u'}) \in \mathcal{H}_\infty$ then $\left\| (z_1 - z')^- \right\|_1 < \left\| z_1^- \right\|_1$.

And Suppose we are given $\mathcal{H}_\infty$, a cost function c and a feasible solution $z_0 = (u, v_1, \ldots, v_N)$.

Lemma 10. Suppose there exists no pair $(u', V_{u'}) \in \mathcal{H}_\infty$ with the properties

1. $u' \leq u$,
2. for all $i = 1, \ldots, N$ there exists $\bar{v}_i \in V_{u'} : \bar{v}_i \leq v_i$,
3. $c^T z' > 0$, where $z' = (u', v_1', \ldots, v_N')$ and $v_i' \in \mathrm{argmax} \left\{ c_i^T v : v \leq v_i, v \in V_{u'} \right\}$ for $i = 1, \ldots, N$.

Then $z_0 = (u, v_1, \ldots, v_N)$ is optimal for $\min \left\{ c^T z : A_N z = b, z \in \mathbb{Z}_+^d \right\}$. If there exists such a pair $(u', V_{u'}) \in \mathcal{H}_\infty$ then $z_0 - z'$ is feasible and it holds $c^T (z_0 - z') < c^T z_0$.

Proof. Suppose that z_0 is not optimal. Then there is some vector $z'' = (u'', v_1'', \ldots, v_N'') \in \mathcal{G}_N$ such that $z_0 - z''$ is feasible and $c^T (z_0 - z'') < c^T z_0$. Feasibility of $z_0 - z''$ implies $z_0 - z'' \geq 0$, hence $z'' \leq z_0$. Therefore, $u'' \leq u$ and $v_i'' \leq v_i, i = 1, \ldots, N$, the latter implying that for any $i = 1, \ldots, N$ there exists a $\bar{v}_i \in V_{u''}$ such that $\bar{v}_i \leq v_i$. Let $z' := (u'', v_1', \ldots, v_N')$ where $v_i' \in \mathrm{argmax} \left\{ c_i^T \bar{v}_i : \bar{v}_i \leq v_i, \bar{v}_i \in V_{u''} \right\}$. Now $c^T (z_0 - z'') < c^T z_0$ implies that $c^T z'' > 0$. Moreover, $c^T z' \geq c^T z'' > 0$. In conclusion, the pair $(u'', V_{u''})$ fulfils conditions 1. to 3. , proving the first claim of the lemma.

With $z' = (u', v_1', \ldots, v_N')$ according to 3. we obtain $c^T (z_0 - z') < c^T z_0$. Moreover $v_i' \leq v_i, i = 1, \ldots, N$, and $u' \leq u$ together imply $z_0 - z' \geq 0$. Finally, $(u', v_1', \ldots, v_N') \in \ker(A_N)$, and therefore $A_N (z_0 - z') = A_N z_0 + 0 = b$ which completes the proof. $\qquad \square$

As $\mathcal{H}_\infty$ contains only finitely many pairs $(u', V_{u'})$, Lemmas 1 and 3 provide a finite algorithm to construct an improving vector or to prove that no such vector can be constructed from the pairs in $\mathcal{H}_\infty$.

Theorem 4. Under the assumption that the optimization problem $\min \left\{ c^T z : A_N z = b, z \in Z_+^{m+N \cdot n} \right\}$ is solvable, an optimal solution can be computed in finitely many steps by application of Algorithm 3 together with the improving vector construction procedures based on Lemma 9 and lemma 10.

The improving vector construction procedures based on Lemma 9 and lemma 10 yield an improving vector in linear time with respect to the number N of scenarios. In accordance with that, I observed in the preliminary test runs that the method is fairly **insensitive** with respect to the number of scenarios. Of course, this becomes effective only after $\mathcal{H}_\infty$ has been computed. *and how complex is that?*

2.3 Method 2: Application of Level Sets for Candidates[5]

2.3.1 Transformation of the Original Problem

To solve the two-stage stochastic integer programming

$$\min \left\{ c^T x + \sum_{i=1}^{S} p_i q^T y_i : Ax = b, x \in \mathbb{R}_+^n, W y_i \geq Tx - h_i, y_i \in \mathbb{Z}_+^d, i = 1, \ldots, N \right\} \tag{5}$$

where all A, T, W are integer matrices without loss of generality.

Here we change (5) to the equivalent form (6)

$$\min \left\{ c^T x + Q(x) : Ax = b, x \in \mathbb{R}_+^n \right\} \tag{6}$$

where

$$Q(x) = \min \left\{ \sum_{i=1}^{S} p_i q^T y_i : W y_i \geq Tx - h_i, y_i \in \mathbb{Z}_+^d, i = 1, \ldots, N \right\} \tag{7}$$

2.3.2 Construct of the Set of Candidates

It's known that function Q is constant on

$$C(\bar{x}) = \bigcap_{i=1}^{r} \{x : \lceil Tx - h_i \rceil = \lceil T\bar{x} - h_i \rceil\} = \bigcap_{i=1}^{r} \left\{x : \left\lceil T^j x - h_i^j \right\rceil = \left\lceil T^j \bar{x} - h_i^j \right\rceil\right\}$$
$$= \bigcap_{i=1}^{r} \bigcap_{j=1}^{p} \left\{x : \left\lceil T^j \bar{x} - h_i^j \right\rceil + h_i^j - 1 < T^j x \leqslant \left\lceil T^j \bar{x} - h_i^j \right\rceil + h_i^j \right\}.$$

for every $\bar{x}$ in the domain, where the superscripts denotes the components of vectors.

Definition 13. The countable set V , given by $V = \{x \in \mathbb{R}^n : x \text{ is a vertex of } C(x) \cap C\}$, is called the ==set of candidates==; an element of V is called a ==candidate point.==

The following theorem justifies its name.

Theorem 5. Let V , the set of candidates, be non-empty. If $\arg\min\{cx + Q(x) : x \in C\} \neq \emptyset$ then $V \bigcap \arg\min\{cx + Q(x) : x \in C\} \neq \emptyset$.

Theorem 5 guarantees us that we only need to find $argmin\{cx + Q(x)\}$ within V. But it might not be a finite set. With the help of the following lemma, we can further restrict it to a finite set.

Lemma 11. Let X be a non-empty set, and f and $\bar{f}$ real functions on X such that $\bar{f}(x) \leqslant f(x)$ for all $x \in X$. Then, for all $\bar{x} \in X$,

$$\underset{x \in X}{\arg\min} f(x) \subset \{x \in X : \bar{f}(x) \leqslant f(\bar{x})\}.$$

Moreover, the difference between these sets is smaller according as $\bar{f}$ is a better approximation of f and $f(\bar{x})$ is a better approximation of $\inf_{x \in X} f(x)$. In particular, if $\bar{f}(\bar{x}) = f(\bar{x})$ and $\bar{x} \in \arg\min_{x \in X} \bar{f}(x)$ then $\bar{x} \in \arg\min_{x \in X} f(x)$.

Proof. For any $\bar{x} \in X$, we have.

$$\underset{x \in X}{\arg\min} f(x) = \bigcap_{y \in X} \{x \in X : f(x) \leqslant f(y)\} \subset \bigcap_{y \in X} \{x \in X : \bar{f}(x) \leqslant f(y)\}$$
$$\subset \{x \in X : \bar{f}(x) \leqslant f(\bar{x})\}$$

$\square$

So we can further restrict the V to finite sets with the help of corresponding level sets $L(c\bar{x} + Q(\bar{x})) = \{x : cx + Q_R(x) \leq c\bar{x} + Q(\bar{x})\}$, where the subscript R denotes continuous relaxation of the problem $Q(x)$. It suffices that we can compute the objective value $cx_r + Q(x_r)$ and construct the partial level sets $L(cx_r + Q(x_r))$ where x_r be the optimal solution of the continuous relaxation of (6).

2.3.3 Algorithm to Two-Stage Stochastic Integer Programming with candidates

We outline the overall algorithm as follows.

1. Compute the Gröbner basis for the second stage IP problem

2. Solve the continuous relaxation of (6) and obtain a partial list of vertices in the dual feasible region. Let x_r be the optimal solution.

3. Compute the objective value $cx_r + Q(x_r)$ and construct the partial level sets $L(cx_r + Q(x_r))$

4. For every candidate point in the level set, evaluate the objective function, using the Gröbner basis to compute the expected value function Q. They are evaluated as per an enumeration scheme.

3   What's the relationship between the corresponding Gröbner Basis?

3.1   Relationship concerning Graver Basis in Method 1

We shall see the relationship between the Graver Basis (Universal Gröbner Basis) of the SIP

$$\min \left\{ c^T x + \sum_{i=1}^{N} p_i q^T y_i : Ax = b, x \in \mathbb{Z}_+^d, Tx + Wy_i = h_i, y_i \in \mathbb{Z}_+^d, i = 1, \ldots, N \right\} \tag{8}$$

and that of the IP

$$\min \left\{ c^T x + p_1 q^T y : Ax = b, x \in \mathbb{Z}_+^d, Tx + Wy = h, y \in \mathbb{Z}_+^d \right\} \tag{9}$$

Then we assert

Theorem 6. (new) Denote the Graver Basis of (8) by $\mathcal{GR}_N$, and that of (9) by $\mathcal{GR}_1$, $i = 1, 2, \ldots, N$. Then if A is of full rank (specially if A is a zero matrix), we have

$$\mathcal{GR}_N = \{(0, 0, \ldots, v_i, \ldots, 0) : (0, v_i) \in \mathcal{GR}_1, i = 1, 2, \ldots, n\}$$

Proof. First we present the corresponding coefficient matrix of (5) is of the form

$$A_N := \begin{pmatrix} A & 0 & 0 & \cdots & 0 \\ T & W & 0 & \cdots & 0 \\ T & 0 & W & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ T & 0 & 0 & \cdots & W \end{pmatrix}$$

I don't understand this proof...

and the corresponding coefficient matrix of (6) is of the form

$$A_1 := \begin{pmatrix} A & 0 \\ T & W \end{pmatrix}$$

So it's trivial that $\mathrm{Ker}_{\mathbb{Z}}(A) = (0, v_1, v_2, \ldots, v_n) : (0, v_i) \in \mathrm{Ker}_{\mathbb{Z}}(A)\}$ since A is of full rank.

Lemma 12. Graver basis of an integer matrix A can also be described by the finite set of $\sqsubseteq$-minimal elements in $\{u \in \mathbb{Z}^n : A \cdot u = 0, u \neq 0\}$.

Recall that for two vectors $a = (a_1, a_2, \ldots, a_n)$ and $b = (b_1, b_2, \ldots, b_n)$ in $\mathbb{Z}^n$, $a \sqsubseteq b$ if and only if $\forall i: a_i b_i \geq 0$ and $|a_i| \leq |b_i|$. So from $\mathrm{Ker}_{\mathbb{Z}}(A) = (0, v_1, v_2, \ldots, v_n) : (0, v_i) \in \mathrm{Ker}_{\mathbb{Z}}(A_i)\}$, we can see that the $\sqsubseteq$-minimal elements of the whole matrix A has a single component being the $\sqsubseteq$-minimal elements of the matrix A_i, and vice versa. So we immediately arrive at the conclusion. $\square$

Theorem 7. (new) Denote all building blocks of Graver Basis of (5) by $\mathcal{H}_N$. Then we have for $N \geq 2$

$$\mathcal{H}_N = \mathcal{H}_2$$

Proof. First it's clear from algorithm that every element x of $\mathcal{H}_N$ is a conformal sum of $\mathcal{H}_1$ elements

$$x = \sum g_i$$

where g_i denotes elements of $\mathcal{H}_N$ and some elements possibly may appear with repetitions. So from the definition of graver basis we can easily see that $\mathcal{H}_N = \mathcal{H}_2$. $\square$

3.2 Relationship concerning Gröbner Basis in Method 2

This is clear to see, for the SIP

$$\min\left\{ c^T x + \sum_{i=1}^{N} p_i q^T y_i : Ax = b, x \in \mathbb{R}_+^n, Wy_i \geq Tx - h_i, y_i \in \mathbb{Z}_+^d, i = 1, \ldots, N \right\} \tag{10}$$

and that of the IP

$$\min\left\{ c^T x + p_1 q^T y_1 : Ax = b, x \in \mathbb{R}_+^n, Wy_1 \geq Tx - h_1, y \in \mathbb{Z}_+^d \right\} \tag{11}$$

Then we assert that they **share the same Gröbner Basis** since every step of the algorithm only manipulates the sub-problem

$$Q(x) = \min\left\{ \sum_{i=1}^{S} p_i q^T y_i : Wy_i \geq Tx - h_i, y_i \in \mathbb{Z}_+^d, i = 1, \ldots, N \right\} \tag{12}$$

4 Computational Results

We use Maple to compute some examples by implementing the algorithm.

Implementation Result 1	
Input Matrix	Graver Basis

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 2 & 4 & 8 \\ 0 & 1 & 3 & 6 & 12 \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & 0 & 2 & -1 \\ 0 & 0 & 2 & -1 & 0 \\ 0 & 0 & 2 & 1 & -1 \\ 0 & 0 & 4 & 0 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 2 & 4 & 8 & 0 & 0 & 0 \\ 0 & 1 & 3 & 6 & 12 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 2 & 4 & 8 \\ 0 & 1 & 0 & 0 & 0 & 3 & 6 & 12 \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 2 & -1 & 0 \\ 0 & 0 & 0 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 1 & -1 \\ 0 & 0 & 2 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & -1 \\ 0 & 0 & 4 & 0 & -1 & 0 & 0 & 0 \end{bmatrix}$$

and another demonstrating example.

Implementation Result 2	
Input Matrix	Graver Basis

$$\begin{bmatrix} 4 & 5 & 0 & 0 \\ 4 & 5 & 0 & 0 \\ 5 & 7 & 3 & 9 \\ 10 & 14 & 6 & 18 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 3 & -1 \\ 5 & -4 & 1 & 0 \\ 5 & -4 & -2 & 1 \\ 10 & -8 & -1 & 1 \\ 15 & -12 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 4 & 5 & 0 & 0 & 0 & 0 \\ 4 & 5 & 0 & 0 & 0 & 0 \\ 5 & 7 & 3 & 9 & 0 & 0 \\ 10 & 14 & 6 & 18 & 0 & 0 \\ 5 & 7 & 0 & 0 & 3 & 9 \\ 10 & 14 & 0 & 0 & 6 & 18 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 3 & -1 \\ 0 & 0 & 3 & -1 & 0 & 0 \\ 5 & -4 & 1 & 0 & 1 & 0 \\ 5 & -4 & 1 & 0 & -2 & 1 \\ 5 & -4 & -2 & 1 & 1 & 0 \\ -5 & 4 & 2 & -1 & 2 & -1 \\ 10 & -8 & -1 & 1 & -1 & 1 \\ 10 & -8 & 2 & 0 & -1 & 1 \\ 10 & -8 & -1 & 1 & 2 & 0 \\ 15 & -12 & 0 & 1 & 0 & 1 \\ 15 & -12 & 3 & 0 & 0 & 1 \\ 15 & -12 & 0 & 1 & 3 & 0 \end{bmatrix}$$

The two examples above are consistent with the results of theorem 6 and theorem 7.

5   Further Study

1. We can further generalize method 1 to the situation of SMIP by the corresponding SMIP Graver test set , thanks to [1]. Some progress has been made in appendix.
2. Refinement of level sets in method 2.

6   Appendix

6.1   (Optional/new) Test Set = (Reduced) Gröbner Basis

Assume A to be an integer matrix, we further study the relationship between Test Set and (Reduced) Gröbner Basis of
$$(IP)_{c,b} : \min \{cx : Ax = b \text{ and } x \in N^n\}$$

First we need to transfer $IP_{c,b}$ to $IP_{>_c,b}$ where $>_c$ denotes the complete total order.

Def 14. $>_c$ is the complete total order satistifying:
1. first compares two points by cost.
2. break ties by the previously assigned term order (monomial order) $>$.

   Note that $>_c$ satisfies (Attention: it's not a term order but almost is!)
1. $>_c$ is a total order on N^n.
2. $>_c$ is cmpatible with sum.

[handwritten margin note: letter to write this out mathematically: e.g. $x \geq_c y$ if 1) $c \cdot x > c \cdot y$ or 2) $c \cdot x = c \cdot y$ and $x > y$ where $>$ is the monomial order.]

Therefore we may make use of the refinement so as to get to the unique optimum.

Theorem 8. $\exists$ testing set.
Proof. (Geometric) First we construct $\mathcal{G}_A = \{(\alpha(i) - \beta(i)), i = 1, 2, ..., t\}$ where $\alpha(i)$ is given below and $\beta(i)$ is the corresponding unique optimum point of $(IP)_{>_c,b}(A\alpha(i))$.

Lemma 13. $\exists \alpha_i$ s.t. {non-optimal solutions of all fibres} $= \bigcup_{i=1}^{t}(\alpha(i) + N^n)$ (immediate consequence of Gordan Dickson Lemma[3])

We can therefore draw an arrow from $\alpha(i)$ to $\beta(i)$ and translate it to all feasible points in $IP_{\{A,c\}}$ such that the translated vector is incident at the corresponding feasible points. By this construction we get to a connected digraph with the unique sink at the optimum point.

It's an easy corollary that $\mathcal{G}_A$ is a test set. $\qquad\square$

How does the name birth? We need to look at the map below and Theorem 7 explains everything. (We denote $A = [A_1, A_2, ..., A_n]$ by column blocking and y^{A_1} short for a monomial of $k[y_1, y_2, ..., y_m]$ where m is the column number of A)

$$\pi : k[x_1, x_2, ..., x_n] \rightarrow k[y^{A_1}, y^{A_2}, ..., y^{A_n}]$$
$$x_i \longmapsto y^{A_i}$$

First we define a $\mathbb{Z}$-linear mapping

$$\pi_* : \mathbb{Z}^n \rightarrow \mathbb{Z}^m$$
$$u \longmapsto Au$$

Then π and π_* are related as $\pi(x^u) = \pi(x_1^{u_1} x_2^{u_2} ... x_n^{u_n}) = y^{A_1 u_1} ... y^{A_n u_n} = y^{Au} = y^{\pi_*(u)}$. we call the kernel of π as **toric ideal** of A, denoted by I_A.

Theorem 9. (optional/new) $Ker(\pi) = < (x^{\alpha(i)} - x^{\beta(i)}), i = 1, 2, ..., s >$. And actually $\{x^{\alpha(i)} - x^{\beta(i)}, i = 1, 2, ..., s\}$ forms a reduced Grobner basis.
Proof. Define

$$\phi(u) := x^{u^+} - x^{u^-}$$

Lemma. The toric ideal I_A is spanned as k-vector space by the set of binomials

$$J = \{x^u - x^v \mid u, v \in \mathbb{N}^n \text{ with } \pi_*(u) = \pi_*(v)\}.$$

A binomial $x^u - x^v$ lies in I_A if and only if $\pi(x^u - x^v) = \pi(x^u) - \pi(x^v) = x^{\pi_*(u)} - x^{\pi_*(v)} = 0$. What remains to show is that every polynomial $f \in I_A$ is a linear combination of such binomials with coefficients in k. Fix a term order $<$ on $k[x_1, ..., x_n]$. Suppose $f \in I_A$ can not be written as such a linear combination. Choose f such that $LM_<(f) = x^u$ is smallest with respect to $<$ for all such polynomials. Since $f \in I_A$ we know

$$0 = \pi(f) = f\left(y^{A_1} \cdots y^{A_n}\right) = y^{\pi_*(u)} + \text{ other terms}.$$

In particular, we know that the term $y^{\pi_*(u)}$ must cancel. Therefore, there exists a monomial x^v in f , with $x^u > x^v$ such that $\pi(u) = \pi(v)$. We know that $f' = f - (x^u - x^v)$ cannot be written as a k-linear combination of binomials since otherwise f could. Since now $LM(f) < LM(f')$, we come to a contradiction for the minimality property of f. The lemma is then proved.

Then we carry out to prove the theorem. First, it's trivial that $x^{\alpha_i} - x^{\beta_i} \in Ker(\pi_*)$ since $A\alpha_i = A\beta_i$.

Since $\alpha_i >_c \beta_i$ for all $i = 1, ..., s$, we have that

$$LM_{>_c}(x^{\alpha_i} - x^{\beta_i}) = x^{\alpha_i} \implies \langle x^{\alpha_i} \rangle \subset LM_{>_c}(Ker(\pi)).$$

From the lemma it is enough to show that $x^\alpha \in \langle x^{\alpha(i)}, i = 1, \ldots, s \rangle$ for each binomial $x^\alpha - x^\beta \in J$. We may assume that $LM_>(x^\alpha - x^\beta) = x^\alpha$. Now $LM_>\left(x^\alpha - x^\beta\right) = x^\alpha$ implies that $\alpha >_c \beta$. Therefore α is a nonoptimal

point with respect to $>_c$ in the $A\alpha$-fiber of IP. Therefore α is in the set of all nonoptimal points from all fibers of IP, which is $\bigcup_{i=1}^{s}(\alpha(i) + \mathbb{N}^n)$. This implies that $\alpha = \alpha(i) + v$ for some $i \in \{1, \ldots, s\}$ and $v \in \mathbb{N}^n$. Therefore $x^{\alpha(i)}$ divides x^α which in turn implies that $x^\alpha \in \langle x^{\alpha(i)}, i = 1, \ldots, s\rangle$. Therefore $\{x^{\alpha(i)} - x^{\beta(i)}, i = 1, 2, ..., s\}$ is a Gröbner basis for I_A with respect to $>_c$, which is clearly reduced observing from their construction. $\qquad\square$

Theorem 10. (optional/new) By giving different term order $>$ for $>_c$, we can get to all optimum points of the problem $IP_{c,b}$ *Actually, I don't even really understand the statement...*

Proof. We first give a general method for specifying monomial orders on $k[x_1, ..., x_n]$.

I don't understand this proof...

Given any $m \times n$ real matrix M and write the rows of M as $w_1, ..., w_m$. Then we can compare monomials x^α and x^β first comparing their w_1-weights $\alpha \cdot w_1$ and $\beta \cdot w_1$. If $\alpha \cdot w_1 > \beta \cdot w_1$ or $\alpha \cdot w_1 < \beta \cdot w_1$, then we order the monomials accordingly. If $\alpha \cdot w_1 = \beta \cdot w_1$, then we continue to the later rows, breaking ties successively with the w_2-weights, the w_3-weights, and so on through the w_m-weights. This process defines an order relation $>_M$. In symbols: $x^\alpha >_M x^\beta$ if there is an $l \leq m$ such that $\alpha \cdot w_i = \beta \cdot w_i$ for i $= 1, \ldots, l-1$, but $\alpha \cdot w_l > \beta \cdot w_l$.

you mean to), right?

Lemma 14. Let M be an $m \times n$ real matrix with non-negative entries s.t. $ker(M) \cap \mathbb{Z}^n = 0$. Then $>_M$ is a monomial order on $k[x_1, ..., x_n]$. (See Exercise 8 of §2 of [2].)

definition? & is it like $\succeq$ but $Mx > My$ instead?

Suppose the optimum of the $(IP)_{c,b}$ forms a set $\{x_1, x_2, ..., x_m\}$. Then it's clear $\forall j \exists c > 0 (c \cdot x_j \neq c \cdot x_i, \forall i \neq j)$. Actually we can prove it by contradiction: since $\sum_i m(\{c : c \cdot x_j = c \cdot x_i\}) = 0$, there must exist some c such that $c \cdot x_j \neq c \cdot x_i, \forall i \neq j$.

Therefore we can find the interval $(a_j, \tilde{a}_j) \subset \mathbb{R}$ s.t.
 1. $c \cdot x_i \in (a_i, \tilde{a}_i], \forall i$.
 2. $(a_j, \tilde{a}_j] \cap (a_i, \tilde{a}_i] = \emptyset, \forall i \neq j$.

Then we define a total order on $\mathbb{R}$, denoted by $>_s$ s.t.
 1. Define $A = (\mathbb{R} - \bigcup_{i \neq j}(a_i, \tilde{a}_i])$, $B = \bigcup_{i \neq j}(a_i, \tilde{a}_i]$, and $C = (a_j, \tilde{a}_j]$.
 2. All the elements in A, B, C are compared in the usual way.
 3. If $a \in A, b \in B, c \in C$, a and b are compared in the usual way; $b < c$; $a < c$ iff. $\exists d \in B(a \leq d)$.

Therefore we get to a total order on $\mathbb{R}$ such that $b <_s c$ whenever $b \in B$ and $c \in C$. Applying the order $>_s$ to build up the monomial order on $k[x_1, ..., x_n]$ as follows. We let

$$M = \begin{pmatrix} c^T \\ \ldots \end{pmatrix}$$

where ... is filled with positive numbers such that M satisfies the conditions of lemma 3.

Finally we can form the composite order $>_c$ of cost and $>_M$, which satisfies $x_j >_c x_i, \forall j \neq i$. $\qquad\square$

6.2 Improvement in IP

Recently, various algebraic integer programming (IP) solvers have been proposed based on the theory of Gröbner bases. The main difficulty of these solvers is the size of the Gröbner bases generated. So we propose an algorithm calculating the test set of $(IP)_{c,b}$ much faster.

Recall that

$$\pi : k[x_1, x_2, ..., x_n] \to k[y^{A_1}, y^{A_2}, ..., y^{A_n}]$$
$$x_i \rightarrowtail y^{A_i}$$

and

$$\pi_* : \mathbb{Z}^n \to \mathbb{Z}^m$$
$$u \rightarrowtail Au$$

From the above section, we know that the algorithm in Gröbner bases of IP is to find a "minimal" generator of $Ker(\pi)$, i.e. $Ker(\pi_*)$.

From [6], we have an important observation.

Theorem 11. Let $K \in \mathbb{N}^{k \times n}$. Then $\phi(K) = \phi(span(K))$

(handwritten: what is ϕ? — what is the span of a matrix?)

We let K be a basis for $Ker(\pi_*)$ consisting of k elements (to simplify notation, we will use K to denote a basis for $Ker(\pi_*)$ as well as the matrix in $\mathbb{Z}^{k \times n}$ whose rows are the vectors in K).

definition 15. For $K, K' \in \mathbb{Z}^{k \times n}$, define $K' \sim K$ if $\text{span} K' = \text{span} K$, i.e. $K' = AK$ for some $A \in \mathbb{Z}^{k \times k}$ s.t. $|\det(A)| = 1$.

From definition we immediately get

(handwritten: $\mathbb{N}^{k \times u}$?)

Proposition 4. Let $K \in \mathbb{Z}^{k \times n}$. Then there exists a $\tilde{K} \sim K$ such that each column of $\tilde{K}$ is either in $\mathbb{N}^k$ or $(-\mathbb{N})^k$.

(handwritten: Proof? Also why only one or the other? If $Q \in (-\mathbb{N})^{k \times n}$ isn't $-\tilde{Q}=(?)\tilde{Q}\sim K$ and $-\tilde{Q} \in \mathbb{N}^{k \times n}$?)

From Proposition 3, we get to $\tilde{K}$. Then let $J \subset \{1, 2, ..., n\}$ be the index set of all columns with negative entries, and let K' be the matrix obtained from $\tilde{K}$ by reversing all signs in the columns indexed by J.

Here's another important observation. First we define $T_j : \mathbb{Z}^n \to \mathbb{Z}^n$ as the operator that switches the sign of the j-th component of the vectors in $\mathbb{Z}^n$. Further, if $p \in k[x_1, ..., x_n]$ has the form $p = \phi(u)$ for some $u \in \mathbb{Z}^n$, we denote $T_j(p) = \phi(T_j(u))$.

Theorem 12. Let $K \in \mathbb{Z}^{k \times n}$ and assume that there exists a finite set $U \subset \text{span} K$ such that $\langle \varphi(U) \rangle = \langle \varphi(\text{span} K) \rangle$. If G is the reduced Gröbner basis for $\langle \varphi(U) \rangle$, with respect to a term order that eliminates x_j, then $\langle T_j G \rangle = \langle \varphi(\text{span}(T_j K)) \rangle$.

We are now ready to describe our algorithm to calculate $\ker \pi$. Let K be a basis for $\ker \pi_*$. By Lemma 3.8 there exists an equivalent basis K' such that each column of K' is either in $\mathbb{N}^n$ or in $(-\mathbb{N})^n$. Let $J \subseteq \{1, 2, ..., n\}$ be the index set of all columns with negative entries, and let K'_J be the matrix obtained from K' by reversing all signs in the columns indexed by J. By Theorem 2,

$$\langle \varphi(K'_J) \rangle = \langle \varphi(\text{span} K'_J) \rangle.$$

If $J = \varnothing$ we are done. If $J \neq \varnothing$, let j be any element of J. Theorem 3 enables us to derive from $\varphi(K'_J)$ a finite set of generators for $\left\langle \varphi\left(\text{span} K'_{J \setminus \{j\}}\right) \right\rangle$. Compute the Gröbner basis for $\varphi(K'_J)$ with respect to a term order that eliminates x_j and apply the operator T_j to it. Proceeding recursively, we can calculate a finite set of generators for $\varphi(\text{span} K'_J)$, which by Theorem 2 equals $\ker \pi_*$.

Proposition 5. (new) The proposed algorithm requires the determination of at most $[\frac{1}{2}n]$ Grobner bases over $k[x_1, x_2, ..., x_n]$.
Proof. Actually the number of determination is at most the number of the elements of J.

Remark: based on the (empirical) fact that the complexity of the Buchberger algorithm is a strongly growing function of the number of variables, we conclude that it's in general more efficient to evaluate $[\frac{1}{2}n]$ Grobner bases over $k[x_1, x_2, ..., x_n]$ than one Grobner basis over $k[x_1, x_2, ..., x_n, y_1, y_2, ..., y_m]$. $\qquad \square$

The algorithm in P392 of [2] is based on the Elimination lemma applied to the $k[x_1, x_2, .., x_n, y_1, y_2, ..., y_m]$, which is **not applicable in large scale**. And so is the old fashioned Geometric Buchberger's algorithm[1]. So from literature [4], we give a new idea of algorithm below *(handwritten: harsh...)*

<div align="center">Finding Reduced Grobener Basis → Augmentation Algorithm</div>

The detailed algorithm to solve the IP is gathered as follows. The value of the algorithm is that it is more efficient to calculate a moderate number of Gröbner bases over $k[x_1, ..., x_n]$ instead of over $K[x_1, ..., x_n, y_1, ..., y_m]$. This is especially true for the memory requirements of the proposed algorithm.

Algorithm 8. (new/ computation of reduced Gröbner bases)
1. Calculate a basis K for $\ker \pi_*$.
2. Find an equivalent basis K' such that all rows of K' lie in the same orthant.
3. Let J be the index set of all columns with negative entries and let K'_J be the matrix obtained from K' by

reversing the signs of the columns indexed by J.

4. Let $G_J = \varphi(K'_J)$.

5. Until $J = \varnothing$, repeat this: Take $j \in J$ and let $G_{J\setminus\{j\}}$ be the result of T_j operating on the reduced Gröbner basis for $< G_J >$ with respect to a term order that eliminates x_j ; then let $J \leftarrow J\setminus\{j\}$.

6. Output $G_\varnothing$, a generating set for $\ker \pi$ which is finite. 7. Use $G_\varnothing$ to generate the reduced grobner Basis $\{x^{\alpha(i)} - x^{\beta(i)} : i = 1, 2..., s\}$. (e.g. apply BunchBerger Algorithm)

8. Augmentation Algorithm.

6.3 Generalization to SMIP[1]

Method 2 applying Graver Basis can be generalized to the SMIP case as we will discover below.

Let $\mathbb{Z}, \mathbb{Q}, \mathbb{R}$ denote the integers, rationals, and reals, respectively. Moreover, let $\mathbb{X} = \mathbb{Z}^{d_1} \times \mathbb{R}^{d_2}$, and denote by $\mathbb{X}_+$ the corresponding non-negative orthant. For given $d = d_1 + d_2$, $A \in \mathbb{Z}^{l \times d}$, $c \in \mathbb{R}^d$, and $b \in \mathbb{R}^l$, let

$$(P)_{c,b} : \quad \min\{c^T z : Az = b, z \in \mathbb{X}_+\}$$

be the family of mixed-integer linear optimization problems as $c \in \mathbb{R}^d$ and $b \in \mathbb{R}^l$ vary. Then for MIP, we first generalize the concept of test sets.

Definition 16. (Test Set)
A set $\mathcal{T} \subseteq \mathbb{R}^d$ is called a test set for $(P)_{c,b}$ if

1. $c^T t > 0$ for all $t \in \mathcal{T}$

2. for every $b \in \mathbb{R}^l$ 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.

In contrast to the common definition of test sets we do not impose finiteness on T. This allows a treatment of test sets for mixed-integer programs which need not be finite in general. Once a test set $\mathcal{T}$ for and a feasible solution z_0 to $(P)_{c,b}$ are available, we have a clear corresponding augmentation algorithm. We omit writing it down.

We will discuss the positive sum property, **a property inherent to LP, IP, and MIP Graver test sets**, and develop, based on this property, a common notational and algorithmic framework for Graver test sets. This simplification of notation and algorithms is an important basis for the development and presentation of fast algorithms to compute Hilbert bases and extremal rays of pointed rational cones, and of a novel decomposition approach to two- and multi-stage stochastic programs. In both applications even the LP cases turn out to be important and of interest.

Definition 17. (Positive Sum Property)
A set G has the positive sum property with respect to $S \subseteq \mathbb{R}^d$ 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

1. $g_i \in G$, $\alpha_i > 0$, $\alpha_i g_i \in S$

2. for all i, g_i and v belong to the same orthant, that is, $g_i^{(k)} v^{(k)} \geq 0$ for every component $k = 1, \dots, d$.

The following lemma was already proved by Graver [8].

Lemma 15. (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 $(P)_{c,b}$.

First, we give without the proof of following lemma.

Lemma 16. (Criterion for Positive Sum Property with respect to Integer Lattice)
Let Λ be an integer sublattice of $\mathbb{Z}^d$. A symmetric set $G \subseteq \Lambda$ has the positive sum property with respect to Λ if and only if the following two conditions hold:

1. G finitely generates Λ over $\mathbb{Z}$

2. 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 $g_i \sqsubseteq v + w$.

Herein, a set G is called symmetric if $v \in G$ implies $-v \in G$.

From the above lemma we can prove that

Lemma 17. The Completion Algorithm 5 terminates and satisfies its specifications.

Proof. Termination of the above algorithm follows immediately by application of the Gordan-Dickson Lemma to the sequence $\{(v^+, v^-) : v \in G \backslash F\} \subseteq \mathbb{Z}_+^{2d}$. To this end, note that $f = normalForm(s, G)$ implies that there is no $g \in G$ with $g \sqsubseteq f$, i.e., there is no $g \in G$ with $(g^+, g^-) \leq (f^+, f^-)$. Thus, the algorithm produces a sequence $\{(v^+, v^-) : v \in G \backslash F\} = \{f_1, f_2, \ldots\}$, where $(f_i^+, f_i^-) \not\leq (f_j^+, f_j^-)$ for $i < j$. This sequence is finite by the classical Gordan-Dickson Lemma. Correctness of the algorithm follows immediately from the above Lemma , since upon termination $normalForm(v + w, G) = 0$ 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 $g_i \sqsubseteq v + w$. $\square$

Let $A = (A_1 \mid A_2)$, 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 $l \times d_1$ and $l \times d_2$. Analogously, we subdivide $c = (c_1, c_2)$, where $c_1 \in \mathbb{R}^{d_1}$ and $c_2 \in \mathbb{R}^{d_2}$. Let $z \in \mathbb{Z}^{d_1}$ and $q \in \mathbb{R}^{d_2}$ denote the integer and continuous variables, respectively, and let $d = d_1 + d_2$. Our aim is to construct a universal test set for the family of optimization problems $(\mathrm{P})_{c,b}$ as $c \in \mathbb{R}^d$ and $b \in \mathbb{R}^l$ 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 18. The MIP Graver test set $\mathcal{G}_{\mathrm{MIP}}(A)$ contains all vectors
  1. $(0, q), q \in \mathcal{G}_{\mathrm{LP}}(A_2)$
  2. $(z, q) \in \ker_{\mathbb{X}}(A), z \neq 0$, and such that there is no $(z', q') \in \ker_{\mathbb{X}}(A)$ satisfying $(z', q') \sqsubseteq (z, q)$.

The set $\mathcal{G}_{\mathrm{MIP}}(A)$, however, is not necessarily finite.

Having only an infinite test set available, we cannot yet make algorithmic use of the Augmentation Algorithm 1 to improve a feasible initial solution to optimality. However, we will construct a finite set $\mathcal{G}_{\mathbb{Z}}(A) \subseteq \mathbb{Z}^{d_1}$ from which an improving vector to a nonoptimal solution of the given problem can be reconstructed in finitely many steps. Thus, the Augmentation Algorithm 1 can be employed again to find an optimal solution.

Consider the projection $\phi : \mathbb{Z}^{d_1} \times \mathbb{R}^{d_2} \to \mathbb{Z}^{d_1}$ which maps each mixed-integer vector onto its d_1 integer components. Define $\mathcal{G}_{\mathbb{Z}}(A) = \mathcal{G}_{\mathbb{Z}}(A_1 \mid A_2) := \phi(\mathcal{G}_{\mathrm{MIP}}(A))$ to be the set of images of the elements in $\mathcal{G}_{\mathrm{MIP}}(A)$.

Lemma 18. $\mathcal{G}_{\mathbb{Z}}(A_1 \mid A_2)$ is finite for every matrix $A \in \mathbb{Z}^{l \times d}$ and for any subdivision $A = (A_1 \mid A_2)$.
Proof. Each element $(z, q) \in \mathcal{G}_{\mathrm{MIP}}(A)$ satisfies $\|z\|_1 \leq \Delta(A_1, A_2)$ for some scalar $\Delta(A_1, A_2)$ which depends only on A_1 and A_2. This inequality can be true only for finitely many vectors $z \in \mathbb{Z}^{d_1}$. $\square$

6.4 Maple Codes

6.4.1 Graver Basis

```
GrBasis:=proc(A)
local n,r,X,Y,L,A1,Id,Alaw,Xlist,Gtor,k,H,vp,vm,K,p,i,pos,neg,v,G;
uses LinearAlgebra, Groebner;

n := ColumnDimension(A);
r := RowDimension(A);
X := Vector(n, symbol = x);
Y := Vector(n, symbol = y);
L := [];

A1 := <A | ZeroMatrix(r, n)>;
Id := <IdentityMatrix(n, n) | IdentityMatrix(n, n)>;
Alaw := <A1, Id>;
Xlist := [op(convert(Transpose(X), list)), op(convert(Transpose(Y), list))];
Gtor := ToricIdealBasis(Alaw, Xlist, tdeg(op(Xlist)));

for k to n do
H:=subs(y[k]=1,Gtor);
Gtor:=H;
end do;

vp:=Vector(n);
vm:=Vector(n);  K:=[];
for p in Gtor do
pos:=op(1,p);
    neg:=-1*op(2,p);
    for i to n do
     vp[i]:=degree(pos,x[i]);
        vm[i]:=degree(neg,x[i]);
      end do;
      v:=convert(vp-vm,list):
      K:=[op(K),v];
 end do;
 G:=convert(K,Matrix);
 print(G);
 end proc;
```

6.4.2 Matrix of SIP

```
SIP := proc(A, T, W, n)
local a, b, c, d, k, A1, A2, A3;
uses LinearAlgebra, Groebner ;
a := ColumnDimension(A);
b := RowDimension(A);
c := ColumnDimension(W);
d := RowDimension(W);
A1 := <A | ZeroMatrix(b, c*n)>;
for k to n do
A2 := <T| LinearAlgebra:-ZeroMatrix(d, c*(k - 1))|W|ZeroMatrix(d, c*(n - k))>;
    A1 := <A1, A2>;
end do;
print(A1);
end proc
```

References

[1] Hemmecke, Raymond. "On the Positive Sum Property and the Computation of Graver Test Sets." Mathematical Programming, vol. 96, no. 2, 2003, pp. 247–269., doi: https://doi.org/10.1007/s10107-003-0385-7.

[2] Cox D., et al. "Using Algebraic Geometry".

[3] Cox D., J. Little, et al. "Ideals Varieties and Algorithms."

[4] Hemmecke, R., and R. Schultz. "Decomposition of Test Sets in Stochastic Integer Programming." Mathematical Programming, vol. 94, no. 2-3, 2003, pp. 323–341., doi: https://doi.org/10.1007/s10107-002-0322-1.

[5] Schultz, Rüdiger, et al. "Solving Stochastic Programs with Integer Recourse by Enumeration: A Framework Using Gröbner Basis." Mathematical Programming, vol. 83, no. 1-3, 1998, pp. 229–252., doi: https://doi.org/10.1007/bf02680560.

[6] Biase, Fausto Di, and Rüdiger Urbanke. "An Algorithm to Calculate the Kernel of Certain Polynomial Ring Homomorphisms." Experimental Mathematics, vol. 4, no. 3, 1995, pp. 227–234., doi: https://doi.org/10.1080/10586458.1995.10504323.

[7] Herzog, Juergen. "Generators and Relations of Abelian Semigroups and Semigroup Rings." Manuscripta Mathematica, vol. 3, no. 2, 1970, pp. 175–193., doi: https://doi.org/10.1007/bf01273309.

[8] Graver, J.E., et al. "On the foundation of linear and integer programming I." Mathematical Programming 9, 1975, pp. 207–226.

[9] A., De Loera Jesús, et al. Algebraic and Geometric Ideas in the Theory of Discrete Optimization. Society for Industrial and Applied Mathematics, 2013.