EXPLORING INTEGER PROGRAMMING WITH GRÖBNER BASES

JULIANNE BRYCE FRANCISCO, LEONARDO MARCIAGA, JORDI MUNOZ

Abstract. Integer programming is a kind of mathematical optimization problem in which the variables are restricted to the set of integers, and both the objective function and constraints are linear. We will be looking into the uses of Gröbner bases for finding optimal solutions to standard integer programs. Concretely, we will describe the properties of the directed graph induced by a Gröbner basis and a term ordering, and its implications for optimization techniques. Finally, we will briefly introduce the connections of Gröbner bases to Graver bases, which arise in the theory of integer programming.

1. Overview of integer programming

In mathematical optimization, there are often problems in which both the objective function and the constraints are linear functions of the variables involved. These kinds of problems are called *linear programs*. If we further impose the restrictions that all possible solutions must have integer values, we get what is called an *integer program* instead. We refer to [2], [3] and [4] for the definitions in this section.

Often, additional variables (*slack variables*) in order to transform any existing inequalities into equalities. We say that such an integer program is in *standard form*. More specifically, we are addressing the following standard integer program $IP_{A,\omega}(\mathbf{b})$:

$$\text{Minimize } \omega \cdot \mathbf{x} \text{ subject to } A\mathbf{x} = \mathbf{b} \text{ and } \mathbf{x} \in \mathbf{N}^n,$$

where A is a $d \times n$ integral matrix, $\mathbf{b} \in \mathbf{Z}^d$, and $\omega \in \mathbf{N}^n$ is a *cost vector*. We also define $IP_{A,\omega}$ as the family of all integer programs obtained when fixing A and ω, and IP_A the family obtained when fixing only A. A useful technique for tackling integer programs is by studying their *linear programming relaxation*:

$$\text{Minimize } \omega \cdot \mathbf{x} \text{ subject to } A\mathbf{x} = \mathbf{b} \text{ and } \mathbf{x} \in \mathbf{R}^n_{\geq 0},$$

where $A, \mathbf{b}$ and ω are as before. We will call this linear program $LP_{A,\omega}(\mathbf{b})$.

Integer programs can also be treated geometrically. A closed half-space in the Euclidean space $\mathbf{R}^n$ is one of the two regions in which a hyperplane divides such space, including its boundary. Then we define a *polyhedron* as any intersection of closed half-spaces in $\mathbf{R}^n$. Note that a polyhedron can be unbounded; when it is bounded, we say it is a *polytope*.

Let $cone(A) = \{A\mathbf{u} : \mathbf{u} \in \mathbf{R}^n_{\geq 0}\}$ denote the set of all possible linear combinations of the columns of A with non-negative coefficients. For a given $\mathbf{b} \in cone(A)$, the set $\{\mathbf{x} \in \mathbf{R}^n_{\geq 0} : A\mathbf{x} = \mathbf{b}\}$ forms a polyhedron of feasible solutions for the linear program $LP_{A,\omega}(\mathbf{b})$, which we will call $P_{\mathbf{b}}$.

Notice that A defines a linear map $\pi_A : \mathbf{R}^n_{\geq 0} \to \mathbf{R}^d$ given by the matrix multiplication $\mathbf{x} \mapsto A\mathbf{x}$. In this sense, $P_{\mathbf{b}}$ is the set of all vectors $\mathbf{x}$ such that $\pi_A(\mathbf{x}) = A\mathbf{x} = \mathbf{b}$. Then $P_{\mathbf{b}}$ is the preimage of $\mathbf{b}$ under π_A; that is, $P_{\mathbf{b}} = \pi_A^{-1}(b)$.

Now we restrict ourselves to the integral solutions, our domain of interest for integer programming. Note that the set $\{\mathbf{x} \in \mathbf{N}^n : A\mathbf{x} = \mathbf{b}\} \subset P_{\mathbf{b}}$ forms the set of feasible solutions to the integer program $IP_{A,c}(b)$. Once again, A defines a linear map $\pi_A^I : \mathbf{N}^n \to \mathbf{Z}^d$ given by $x \mapsto Ax$.

Date: November 25, 2024.

2. Overview of toric ideals

Now we seek to deal with the integer programming problem algebraically: as each possible solution to our integer program is a vector with non-negative coordinates, it can be represented as a monomial. We will mostly refer to [1] in this section. Let $k[\mathbf{x}] = k[x_1, \dots, x_n]$ be the polynomial ring in the variables $x_1, \dots, x_n$. Any vector $\mathbf{u} = (u_1, \dots, u_n) \in \mathbf{N}^n$ will now be identified with the monomial $\mathbf{x}^{\mathbf{u}} = x_1^{u_1} \dots x_n^{u_n} \in k[\mathbf{x}]$.

In order to represent cost functions algebraically, for any cost vector ω and any term order $\prec$ we define the term order $\prec_\omega$ as follows: for $\mathbf{a}, \mathbf{b} \in \mathbf{N}^n$,

$$\mathbf{a} \prec_\omega \mathbf{b} \iff \omega \cdot \mathbf{a} < \omega \cdot \mathbf{b} \text{ or } (\omega \cdot \mathbf{a} = \omega \cdot \mathbf{b} \text{ and } \mathbf{a} \prec \mathbf{b}).$$

Something similar can be done to algebraically represent the vectors in $\mathbf{Z}^d$. For any vector $\mathbf{a} = (a_1, \dots, a_d) \in \mathbf{Z}^d$, we will identify $\mathbf{a}$ with the monomial $t_1^{a_1} \dots t_d^{a_d}$, where the variables in this monomial are allowed to have negative exponents. Such monomials live in the *Laurent polynomial ring* $k[\mathbf{t}^{\pm 1}] = k[t_1, \dots, t_d, t_1^{-1}, \dots, t_d^{-1}]$, in which negative exponents in the variables are allowed.

Given that we can relate $\mathbf{N}^n$ with monomials in $k[\mathbf{x}]$ and $\mathbf{Z}^d$ with $k[\mathbf{t}^{\pm 1}]$, the map π_A^I from before now relates to a map $\hat{\pi}_A^I$ between $k[\mathbf{x}]$ and $k[\mathbf{t}^{\pm 1}]$ given by

$$\hat{\pi}_A^I \colon k[\mathbf{x}] \to k[\mathbf{t}^{\pm 1}], \ x_j \mapsto \mathbf{t}^{\mathbf{a}_j},$$

where $\mathbf{a}_j$ is the j-th column of A. Then the kernel of $\hat{\pi}_A^I$, that is, the set of polynomials $f \in k[\mathbf{x}]$ such that $\hat{\pi}_A^I(f) = 0$, is called the *toric ideal* of A, denoted I_A. Its main defining property is that it is the *ideal generated by differences of monomials with the same image*:

Proposition 2.1. ([1]) The toric ideal I_A is generated by the set of binomials

$$\{\mathbf{x}^{\mathbf{u}} - \mathbf{x}^{\mathbf{v}} \colon \mathbf{u}, \mathbf{v} \in \mathbf{N}^n, \pi_A^I(\mathbf{u}) = \pi_A^I(\mathbf{v})\}.$$

For any $\mathbf{a} = (a_1, \dots, a_n) \in \mathbf{R}^n$, denote $\mathbf{a}^+$ the vector whose i-th is $|a_i|$ if $a_i \geq 0$, and zero otherwise. We define $\mathbf{a}^-$ analogously.

Then, note that since $\mathbf{u} \in \ker(\pi_A^I)$ (that is, $A\mathbf{u} = \mathbf{0}$), the vectors $\mathbf{u}^+$ and $\mathbf{u}^-$ satisfy $A\mathbf{u}^+ - A\mathbf{u}^- = A(\mathbf{u}^+ - \mathbf{u}^-) = A\mathbf{u} = \mathbf{0}$. Hence $A\mathbf{u}^+ = A\mathbf{u}^-$; that is, $\pi_A(\mathbf{u}^+) = \pi_A(\mathbf{u}^-)$, implying that the monomial difference $\mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-}$ belongs to I_A. This leads to the following corollary:

Corollary 2.1. ([1]) The toric ideal I_A is generated by the set of binomials

$$\{\mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-} \colon \mathbf{u} \in \ker(\pi_A^I)\}.$$

From this point on, instead of identifying a vector $\mathbf{u} \in \ker(\pi_A^I)$ with its monomial $\mathbf{x}^{\mathbf{u}}$, we will associate it with the binomial $\mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-}$.

Computing the Gröbner basis of I_A will be relevant in the next section, so now we introduce an algorithm for computing it. The fact that the variable x_j is mapped to a monomial $\mathbf{t}^{\mathbf{a}_j}$ with possibly negative exponents (i.e., a fraction) can be represented by the polynomial condition $x_j \mathbf{t}^{\mathbf{a}_j^-} - \mathbf{t}^{\mathbf{a}_j^+}$. Hence we introduce the ideal

$$J = \langle t_0 t_1 \dots t_d - 1, x_1 \cdot \mathbf{t}^{\mathbf{a_1^-}} - \mathbf{t}^{\mathbf{a_1^+}}, \dots, x_n \cdot \mathbf{t}^{\mathbf{a_n^-}} - \mathbf{t}^{\mathbf{a_n^+}} \rangle.$$

(where t_0 is an additional variable) to impose such conditions on all the x_j's; note that this represents a rational parameterization of the x_j's. Such ideal has the property that $J = I_A \cap k[\mathbf{x}]$ (see proof of Algorithm 3.5 in [3]), hence I_A is the $(d+1)$-th elimination ideal of J. We therefore have the following algorithm based on elimination theory to compute the Gröbner basis of I_A.

Algorithm 1 ([1]) Computing the Gröbner basis of the toric ideal I_A

1. Introduce $n+d+1$ indeterminates $t_0, t_1, \ldots, t_d, x_1, x_2, \ldots, x_n$. Let $\prec$ be any elimination term order with $\{t_i\} \succ \{x_j\}$.
2. Compute the reduced Gröbner basis $\mathcal{G}$ for the ideal

$$J = \langle t_0 t_1 \ldots t_d - 1, x_1 \cdot \mathbf{t}^{\mathbf{a_1^-}} - \mathbf{t}^{\mathbf{a_1^+}}, \ldots, x_n \cdot \mathbf{t}^{\mathbf{a_n^-}} - \mathbf{t}^{\mathbf{a_n^+}} \rangle.$$

3. Output: the set $\mathcal{G} \cap k[\mathbf{x}]$ is the reduced Gröbner basis for I_A with respect to $\prec$.

3. Graphs Defined by Term Orders

In this section we will describe an algorithm for solving integer programs, based on an auxiliary directed graph we will describe below. Refer to [1] for the following results.

We start by defining a graph $\pi_A^{-1}(\mathbf{b})_{\mathcal{F}}$, for any $\mathcal{F} \subset \ker(\pi_A)$, as follows. Let the nodes of this graph be the elements of the preimage $\pi_A^{-1}(\mathbf{b})$. Two nodes $\mathbf{u}$ and $\mathbf{u'}$ are connected by an edge if and only if $\mathbf{u} - \mathbf{u'}$ or $\mathbf{u'} - \mathbf{u}$ belong to $\mathcal{F}$, i.e., they are "congruent" modulo $\mathcal{F}$. The following theorem provides a necessary and sufficient condition for the connectivity of this graph:

Proposition 3.1. Let $\mathcal{F} \in \ker(\pi_A)$. The graphs $\pi_A^{-1}(\mathbf{b})_{\mathcal{F}}$ are connected for all $\mathbf{b} \in \mathit{cone}(A)$ if and only if the set $\{\mathbf{x}^{\mathbf{v}^+} - \mathbf{x}^{\mathbf{v}^-} : \mathbf{v} \in \mathcal{F}\}$ generates the toric ideal I_A.

When $\mathcal{F}$ is a Gröbner basis of I_A, the graphs $\pi_A^{-1}(\mathbf{b})_{\mathcal{F}}$ are not only connected, but a stronger property can be observed by introducing a term order $\prec$. We will now define a directed graph $\pi_A^{-1}(\mathbf{b})_{\mathcal{F},\prec}$ as follows. The underlying directed graph is $\pi_A^{-1}(\mathbf{b})_{\mathcal{F}}$, and an edge $(\mathbf{u}, \mathbf{u'})$ is directed from $\mathbf{u}$ to $\mathbf{u'}$ if $\mathbf{u'} \prec \mathbf{u}$.

Proposition 3.2. Let $\mathcal{G} \subset \ker(\pi_A)$ and $\prec$ be any term order on $\mathbf{N}^n$. The directed graph $\pi_A^{-1}(\mathbf{b})_{G,\prec}$ has a unique sink (i.e., a node with no edges leaving it) for every $\mathbf{b} \in \mathit{cone}(A)$ if and only if the set of binomials $\{x^{v^+} - x^{v^-} : v \in G\}$ is a Gröbner basis for the toric ideal I_A with respect to $\prec$.

Hence, when applying the division algorithm to a monomial $\mathbf{x}^{\mathbf{u}}$ divided by a Gröbner basis $\mathcal{G}$, the monomials arising as partial results form a directed path from $\mathbf{x}^{\mathbf{u}}$ until its normal form modulo $\mathcal{G}$ (i.e., its remainder), and that remainder will be obtained no matter the starting monomial.

This fact can be immediately applied to integer programming recalling that any cost vector $\omega \in \mathbf{R}^n$ corresponds to a term order $\prec_\omega$.

Algorithm 2 ([1]) Integer programming for a fixed matrix and cost function

Input: A $d \times n$-matrix A and a cost function $\omega \in \mathbf{R}^n$.
Output: An optimal point $\mathbf{u} \in \pi^{-1}(\mathbf{b})$ with $\mathbf{u} \cdot \omega$ minimal, for any given $\mathbf{b} \in \mathit{cone}(A)$.
1. Compute the reduced Gröbner basis $\mathcal{G}_{A,\prec_\omega}$ for I_A with respect to $\prec_\omega$
2. For any given right-hand side vector $\mathbf{b} \in \mathbb{N}A$ do
2.1 Find any feasible solution $\mathbf{v} \in \pi^{-1}(\mathbf{b})$
2.2 Compute the normal form (i.e., remainder) $\mathbf{x}^{\mathbf{u}}$ of $\mathbf{x}^{\mathbf{v}}$ with respect to $\mathcal{G}_{A,\prec_\omega}$
2.3 Output $\mathbf{u}$

Example. Consider the following setup:

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \quad \omega = [1, 2, 3], \quad b = \begin{bmatrix} 3 \\ 3 \end{bmatrix}.$$

We execute the steps of the algorithm:

(1) Compute the reduced Gröbner basis $\mathcal{G}_\omega$ for the ideal I_A:

$$I_A = \langle x_1x_3 - x_2\rangle,$$
$$\mathcal{G}_{A,\prec_\omega} = \{x_1x_3 - x_2\}.$$

Note: The Gröbner basis was computed using the Macaulay2 command `gens gb I`, and using the default term order (i.e., grevlex).

(2) For the given right-hand side vector $\mathbf{b}$:

(a) Solve $A \cdot \mathbf{v} = \mathbf{b}$ to find a feasible solution $\mathbf{v}$:

- Write the system of equations based on $A \cdot \mathbf{v} = \mathbf{b}$:

$$v_1 + v_2 = 3,$$
$$v_2 + v_3 = 3.$$

- Solve the equations step by step:
  - From the first equation: $v_1 = 3 - v_2$.
  - Substitute v_2 into the second equation: $v_2 + v_3 = 3 \implies v_3 = 3 - v_2$.
- Let $v_2 = 1$ (an arbitrary choice satisfying integer solutions), so:

$$v_1 = 3 - 1 = 2, \quad v_3 = 3 - 1 = 2.$$

- Hence, $v = [2, 1, 2]$ is a feasible solution.

(b) Compute the normal form of $\mathbf{x}^{\mathbf{v}} = x_1^2x_2x_3^2$ with respect to $\mathcal{G}_{A,\prec_\omega}$. Using the Gröbner basis relation $x_1x_3 = x_2 \implies x_1^2x_3^2 = x_2^2$. we substitute to get:

$$x_1^2x_2x_3^2 = x_2^2x_2 = x_2^3$$

This yields $\mathbf{u} = [0, 3, 0]$.

(3) Therefore, output $\mathbf{u}$.

4. Brief overview of parametric integer programming

In this section we address the problem of parametric integer programming, where we seek methods for solving integer programs having only the matrix A as input. We also study the notion of test sets, which allow us to generate simple solution algorithms to such integer programs.

Following [3], we say that a set $\mathcal{R}_\omega \subset \ker(\pi_A^I)$ is a *test set* for the family $IP_{A,\omega}$ if $\omega \cdot \mathbf{r} > 0$ for all $\mathbf{r} \in \mathcal{R}_\omega$ and, for each non-optimal solution $\mathbf{u}$ to a program $IP_{A,\omega}(\mathbf{b})$, there is $\mathbf{r} \in \mathcal{R}_\omega$ such that $\mathbf{u} - \mathbf{r}$ is a feasible solution of $IP_{A,\omega}(\mathbf{b})$.

If we have such a finite test set, we can easily solve all programs in $IP_{A,\omega}$ by starting at any solution of $IP_{A,\omega}(\mathbf{b})$ and subtracting elements from $\mathcal{R}_\omega$: we will be done when there is no such $\mathbf{r}$ that allows us to continue. The most immediate example of a test set is given by the reduced Gröbner basis from before.

Proposition 4.1. ([3]) The reduced Gröbner basis $\mathcal{G}_{\prec_\omega}$ of I_A is a uniquely defined test set for $IP_{A,\omega}$.

A universal test set for IP_A is a subset of $\ker(\pi_A^I)$ that contains a test set for $IP_{A,\omega}$, for any ω that is generic for IP_A (that is, if it has a unique optimal point). This would allow us to apply the above technique without having to recompute the test set when varying $\mathbf{b}$ and ω. To achieve that, we introduce the concept of Graver bases.

First we define the *Hilbert basis* of A, as described in [2], as

$$\mathcal{H}_A = \{\mathbf{x} \in cone(A)\backslash\{0\} : \text{no element } y \in cone(A)\backslash\{0, \mathbf{x}\} \text{ satisfies } \mathbf{y} \le \mathbf{x}\}.$$

where $\leq$ denotes element-wise comparison. Now, for any sign pattern $\sigma \in \{-1, 1\}^n$ let D_σ be the $n \times n$-diagonal matrix with i-th entry σ_i. The *Graver basis* of A, $\mathcal{GR}_A$, defined as

$$\mathcal{GR}_A := \bigcup_{\sigma \in \{-1,1\}^n} D_\sigma \cdot \mathcal{H}_{AD_\sigma}$$

is the symmetric finite set that contains a set of *primitive* binomials, meaning that binomial $\mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-} \in I_A$ (or vector $\mathbf{u}$) has no other binomial $\mathbf{x}^{\mathbf{v}^+} - \mathbf{x}^{\mathbf{v}^-} \in I_A$ (or vector $\mathbf{v}$) such that both $\mathbf{u}^+ - \mathbf{v}^+$ and $\mathbf{u}^- - \mathbf{v}^-$ are non-negative. Its main property is the following:

Proposition 4.2. ([3]) The Graver basis of A is a universal test set for IP_A.

Computing the Graver basis of a matrix A involves a process called Lawrence lifting: we define the $(d+n) \times 2n$ matrix

$$\Lambda_A = \begin{bmatrix} A & 0 \\ I & I \end{bmatrix}$$

as the *Lawrence lifting* of A, where I is the $n \times n$-identity matrix and 0 is the $d \times n$-zero matrix.

The above matrix $\Lambda(A)$ is chosen due to the following properties. First, it has an isomorphic kernel to that of A: $\ker(\Lambda(A)) = \{(\mathbf{x}, -\mathbf{x}) : \mathbf{x} \in \ker(A)\}$. To determine its toric ideal, we introduce the polynomial ring in $2n$ variables $k[\mathbf{x}, \mathbf{y}] = k[x_1, \ldots, x_n, y_1, \ldots, y_n]$ to accomodate the extra n variables. Then the toric ideal $I_{\Lambda(A)}$ will have a similar form to that of I_A:

$$I_{\Lambda(A)} = \langle \mathbf{x}^{\mathbf{u}^+}\mathbf{y}^{\mathbf{u}^-} - \mathbf{x}^{\mathbf{u}^-}\mathbf{y}^{\mathbf{u}^+} : \mathbf{u} \in \ker(\pi_A) \rangle.$$

These similarities lead to the following result.

Proposition 4.3. ([1]) For a Lawrence type matrix Λ_A, the following sets of binomials coincide:

(1) The Graver basis of Λ_A;
(2) Any reduced Gröbner basis of I_{Λ_A};
(3) Any minimal generating set of I_{Λ_A} (up to scalar multiples).

An observation from [1] is the following:

$$\mathcal{GR}_{\Lambda(A)} = \{\mathbf{x}^{\mathbf{u}^+}\mathbf{y}^{\mathbf{u}^-} - \mathbf{x}^{\mathbf{u}^-}\mathbf{y}^{\mathbf{u}^+} : \mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-} \in \mathcal{GR}_A\}$$

Notice that replacing all y_i's by 1 in the above expression leads to a subset of $k[\mathbf{x}]$. We leverage the above observation together with Proposition 4.3 to get the following algorithm.

Algorithm 3 ([1]) Computing the Graver basis of A

1. Fix any term order on the polynomial ring $k[\mathbf{x}, \mathbf{y}]$. Compute the Gröbner basis $\mathcal{G}$ of $I_{\Lambda(A)}$.
2. Substitute $y_1, \ldots, y_n \mapsto 1$ in $\mathcal{G}$.
3. The resulting subset of $k[x_1, \ldots, x_n]$ is the Graver basis of A, $\mathcal{GR}_A$.

Computing $\mathcal{GR}_A$ only requires a single reduced Gröbner basis for Lawrence lifting Λ_A. As an example, consider the 2×4 matrix A and feasible solution ω

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 5 & 10 & 25 \end{bmatrix} \text{ and } \omega = \begin{bmatrix} 0, 1, 0, 1 \end{bmatrix}.$$

We can use programs such as Macaulay2 to compute both $\mathcal{G}_A$ and $\mathcal{GR}_A$, using the functions `toricGroebner(A)` and `toricGraver(A)` respectively. This gives us the Gröbner and Graver basis

$$\mathcal{G}_{A,\prec_\omega} = \begin{bmatrix} 0 & 3 & -4 & 1 \\ -5 & 6 & 0 & 1 \\ -5 & 3 & 4 & -2 \\ 5 & 0 & -8 & 3 \end{bmatrix}, \mathcal{GR}_A = \begin{bmatrix} 0 & 3 & -4 & 1 \\ -5 & 6 & 0 & 1 \\ -5 & 3 & 4 & -2 \\ 5 & 0 & -8 & 3 \\ -5 & 9 & -4 & 0 \end{bmatrix}.$$

Notice how the Graver basis creates an additional row in comparison to the Gröbner basis, but the Graver basis can be used as test set for any cost function ω and vector $\mathbf{b}$.

References

[1] Bernd Sturmfels. *Gröbner Bases and Convex Polytopes*, volume 8 of *University Lecture Series*. American Mathematical Society, 1996.

[2] Bernd Sturmfels. Algebraic recipes for integer programming, 2003.

[3] Rekha R. Thomas. *Gröbner Bases in Integer Programming*, pages 533–572. Springer US, Boston, MA, 1998.

[4] Günter M. Ziegler. *Gröbner Bases and Integer Programming*, pages 168–183. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.