

Graver Basis

Francisco Javier Blázquez Martínez



Department of discrete optimization
November 2020

Integer Linear Programming

The underlying question is how to solve the integer linear problem (IP).

$$(IP) \equiv \max\{c^t x : Ax = b, l \leq x \leq u, x \in \mathbb{Z}^n\}$$

Where $A \in \mathbb{Z}^{m \times n}$, $b \in \mathbb{Z}^m$, $c \in \mathbb{Z}^n$, l and u lower and upper bounds for x .

Integer Linear Programming

The underlying question is how to solve the integer linear problem (IP).

$$(IP) \equiv \max\{c^t x : Ax = b, l \leq x \leq u, x \in \mathbb{Z}^n\}$$

Where $A \in \mathbb{Z}^{m \times n}$, $b \in \mathbb{Z}^m$, $c \in \mathbb{Z}^n$, l and u lower and upper bounds for x .

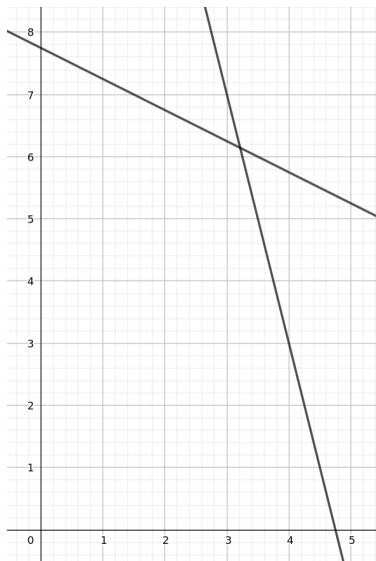
IP is NP-Hard. There are algorithms with polynomial complexity for certain IPs and there are algorithms for the general IPs based on cutting plane methods, dynamic programming, lattice-basis reduction...

Thanks to the study of **Graver basis** new algorithms have appeared improving classic techniques in certain cases.

Integer Linear Programming

$$\begin{aligned} & \max 6x + y \\ (P1) \equiv & \text{ s.t : } 4x + y \leq 19 \\ & x + 2y \leq 31 \\ & x, y \geq 0 \end{aligned}$$

- Feasible region
- Extreme points
- Directions
- Extreme directions



- **Definition 1:** Two vectors $u, v \in \mathbb{R}^n$ are said to be **sign compatible** if $u_i \cdot v_i \geq 0$ for all $i \in \{1, \dots, n\}$
- **Definition 2:** A vector $u \in \ker(A)$ is **indecomposable** if it is not the sum of two sign compatible and non zero elements in $\ker(A)$.

- **Definition 1:** Two vectors $u, v \in \mathbb{R}^n$ are said to be **sign compatible** if $u_i \cdot v_i \geq 0$ for all $i \in \{1, \dots, n\}$
- **Definition 2:** A vector $u \in \ker(A)$ is **indecomposable** if it is not the sum of two sign compatible and non zero elements in $\ker(A)$.

Graver Basis $\equiv Gr(A)$

The Graver Basis of a given matrix A is defined as the set of integral indecomposable elements in the kernel of A .

A key concept in linear programming are the extreme directions. They are a set of directions which allow to represent every other as positive linear combination of themselves. The following property lets us think of Graver basis as the set of extreme directions for the integral case.

- **Property 1:** Every integral element in $\ker(A)$ can be expressed as positive integral linear combination of elements in $Gr(A)$.

A key concept in linear programming are the extreme directions. They are a set of directions which allow to represent every other as positive linear combination of themselves. The following property lets us think of Graver basis as the set of extreme directions for the integral case.

- **Property 1:** Every integral element in $\ker(A)$ can be expressed as positive integral linear combination of elements in $Gr(A)$.

As in simplex, where we checked optimality only by analysing extreme directions, we have the analogous for integral case with Graver basis.

- **Property 2:** Given z in the feasible region of an IP, z is not optimum if and only if there exists $g \in Gr(A)$ s.t. $c^t g > 0$ and $l \leq z + g \leq u$

From the **property 2** we know that if there is no improvement direction in the Graver basis then there is no improvement direction. This allows us to restrict our search to the elements of the Graver basis. However, as computing Graver basis is a complex task, other approaches as bounding the norm of Graver basis elements have proved useful.

- **Property 3:** Given $A \in \mathbb{Z}^{m \times n}$ and Δ an upper bound for the absolute value of each component of A , for every $g \in Gr(A)$:
 - $\|g\|_1 \leq m^{m/2} \Delta^m \cdot (n - m)$
 - $\|g\|_1 \leq (2m\Delta + 1)^m$

From the **property 2** we know that if there is no improvement direction in the Graver basis then there is no improvement direction. This allows us to restrict our search to the elements of the Graver basis. However, as computing Graver basis is a complex task, other approaches as bounding the norm of Graver basis elements have proved useful.

- **Property 3:** Given $A \in \mathbb{Z}^{m \times n}$ and Δ an upper bound for the absolute value of each component of A , for every $g \in Gr(A)$:
 - $\|g\|_1 \leq m^{m/2} \Delta^m \cdot (n - m)$
 - $\|g\|_1 \leq (2m\Delta + 1)^m$

Both bounds are exponential in the general case, however, for certain matrices we'll have a much better bound which will allow us to reduce much more our problem. We denote as $\|Gr(A)\|$ an upper bound to the l_1 - norm of A 's Graver basis elements.

Algorithm

The following is a general IP algorithm. At first sight it may not seem an improvement, however, in the sub-problem the feasible region is being restricted to elements bounded by Graver basis norm, which reduces the problem considerably in certain cases.

The correctness of this algorithm is given by **property 2**.

General IP algorithm using Graver basis norm

- 1 From a feasible solution z_i
- 2 Find g^* optimum for the sub-problem:

$$\max\{c^t g : Ag = 0, l - z_i \leq g \leq u - z_i, g \in \mathbb{Z}^n, \|g\|_1 \leq \|Gr(A)\|\}$$

- $g^* = 0 \implies z_i$ optimal solution.
- $g^* \neq 0 \implies g^*$ improvement direction, loop back to 1 with $z_{i+1} = z_i + \lambda \cdot g^*$ with the biggest λ respecting the bounds.

N-Fold, a success example

A N-Fold IP has constriction matrix A of the form ($A_i \in \mathbb{Z}^{r \times t}$, $B_i \in \mathbb{Z}^{s \times t}$):

$$N = \begin{pmatrix} A_1 & A_2 & \cdots & A_n \\ B_1 & 0 & \cdots & 0 \\ 0 & B_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & B_n \end{pmatrix}$$

N-Fold, a success example

A N-Fold IP has constriction matrix A of the form ($A_i \in \mathbb{Z}^{r \times t}$, $B_i \in \mathbb{Z}^{s \times t}$):

$$N = \begin{pmatrix} A_1 & A_2 & \cdots & A_n \\ B_1 & 0 & \cdots & 0 \\ 0 & B_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & B_n \end{pmatrix}$$

It's possible (using Steinitz Lemma) to obtain a much tighter bound for the norm of the elements in the Graver basis than the ones mentioned before.

N-Fold Graver basis bound

For all $g \in Gr(N)$ $\|g\|_1 \leq L_B(2r\Delta L_B + 1)^r =: L_A$ where $L_B = (2s\Delta + 1)^s$

N-Fold, a success example

The previous bound given is independent of the number of blocks of the N-Fold matrix as well as the number of columns of A_i and B_i . This results in a much more restricted sub-problem for the algorithm presented before which leads to having a polynomial time (in the number of variables) for solving N-Fold IP once the block sizes are fixed.

N-Fold complexity

The N-Fold IP can be solved in time $n^2 t^2 \varphi \log^2 nt \cdot (rs\Delta)^{O(r^2s+rs^2)} + LP$

The proof of this can be found in [1].

- [1] Kim-Manuel Klein Friedrich Eisenbrand Christoph Hunkenschröder. “Faster Algorithms for Integer Programs with Block Structure”. In: *École polytechnique fédérale de Lausanne, Switzerland*, 2018.
- [2] Jack E. Graver. “On the foundations of linear and integer linear programming I”. In: *Syracuse University, New York, U.S.A.*, 1975.
- [3] Bernd Sturmfels. “Algebraic Recipes for Integer Programming”. In: *Mathematical sciences research institute, Berkeley, U.S.A.*, 2003.
- [4] Raymond Hemmecke Elisabeth Finhold. “Lower bounds on the Graver complexity of M-fold matrices”. In: *Technische Universität München, Germany*, 2013.