Applications of algebraic-geometric techniques in optimization problems

Aravindan Vethachalam

April 2022

Contents

1 Outline of the report

In this report, we briefly summarize the techniques behind solving optimization problems using abstract algebraic constructs. Initially, we'll go through the Gröbner basis and observe how it's used to solve Integer Programming problems (IPs). Secondly, to better understand the mechanics, we'll dive into the geometric intuition behind the optimization procedure. Finally, we shall explore more robust methods such as the Graver basis to solve a more general family of IPs. Just a note, while the primary focus would be on IPs for the majority of this report, we shall also see some details from the lens of Stochastic Programming problems and observe how all of this fits into the bigger picture.

2 Solving Integer Programming problems using Gröbner basis

2.1 Basic terminology

It is advised that the reader is familiar with some concepts of algebra like Ideals and Varieties. See the textbook by Cox [2] for reference.

- k : Algebraic field
- x^α : $x_1^{\alpha_1} \dots x_n^{\alpha_n}$, basically a monomial in $x_1, \dots, x_n$
- cx^α : A term in the polynomial where $c \in k$
- $f = \sum_{\alpha \in Z_+^n} c_\alpha x^\alpha$: Polynomial in n variables
- $k[x] = k[x_1, \dots, x_n]$: A polynomial ring in n variables.
- $\mathbb{A}^n = \mathbb{A}^n(k)$: Affine space over field k.
- Affine Variety : $V(I) = \{z \in \mathbb{A}^n | f(z) = 0 \forall f \in I\}$

2.2 Solving IPs using Gröbner basis

2.2.1 Building an ideal from the IP constraints

The primary objective of this section is to understand how the ideal is constructed from the optimization problem. Once the ideal is constructed, we can simply use the Buchberger's algorithm to obtain the Gröbner basis. Finally, after we have the Gröbner basis in our arsenal, we reduce a monomial (which has the RHS b in it) using the Gröbner basis to obtain our optimal solution (if it exists).

2.2.2 Case 1: A and b have positive integer values

$$\begin{aligned} \min_x \quad & c^t.x \\ \text{s.t.} \quad & Ax = b, \\ & A \in \mathbb{Z}_+^{m\times n}, b \in \mathbb{Z}_+^m, c \in \mathbb{Z}_+^n \end{aligned}$$

A new variable for the ith constraint is introduced (let's call it z_i) and the ith constraint is represented as follows:

$$z_i^{a_{i1}x_1+\cdots+a_{in}x_n} = z_i^{b_i}$$

Note that the constraints can be formulated as monomials in this manner. Secondly, let us introduce another variable, say w_i for each of the decision variables. **Just note that there are as many w_i's as there are decision variables and as many z_i's as constraints.**

With that in mind, let us define a mapping

$$\phi : k[w_1, \ldots, w_n] \to k[z_1, \ldots, z_m]$$

$$\text{such that} \quad \phi(w_j) = \prod_{i=1}^{m} z_i^{a_{ij}}$$

then, for $g \in k[w]$ we have,

$$\phi(g(w_1, \ldots, w_n)) = g(\phi(w_1), \ldots, \phi(w_n))$$

The main takeaway here is that, through the use of these transformations , we are essentially encoding the constraints as monomials. By doing so, we can leverage algebraic algorithms which take these monomials as inputs and reduce them.

To let the idea sink in, let's see an example of how to form an ideal for the following constraint matrix (it's not an optimization problem):

$$\begin{aligned} 4x_1 + 5x_2 + x_3 &= 37 \\ 2x_1 + 3x_2 + x_4 &= 20 \end{aligned}$$

By using the mapping we defined earlier, we get the following:

$$\begin{aligned}\phi(w_1) &= z_1^4 z_2^2 \\ \phi(w_2) &= z_1^5 z_2^3 \\ \phi(w_3) &= z_1 \\ \phi(w_4) &= z_2\end{aligned}$$

The set of feasible solutions are all the integer points (x_1, x_2, x_3, x_4) such that,

$$\phi(w_1^{x_1} w_2^{x_2} w_3^{x_3} w_4^{x_4}) = z_1^{37} z_2^{20}$$

It should be obvious at this point to see all the constraints taken into account.

By defining $f_j = \phi(w_j) = \prod_{i=1}^{m} z_i^{a_{ij}}$, we can now construct our ideal as given below,

$$I = \langle f_1 - w_1, f_2 - w_2, \ldots, f_n - w_n \rangle \subset k[z, w]$$

2.2.3 Case 2: A and b have integer values

$$\begin{aligned}\min_{x} \quad & c^t.x \\ \text{s.t.} \quad & Ax = b, \\ & A \in \mathbb{Z}^{m \times n}, b \in \mathbb{Z}^m, c \in \mathbb{Z}_+^n\end{aligned}$$

For this variation, only a minor addition is made while constructing the ideal generator. We simply extend the mapping to polynomial rings which can take inverse polynomials as well $i.e(z_i^{-1})$.

Defining the mapping

$$\phi : k[w_1, \ldots, w_n] \to k[z_1, \ldots, z_m, z_1^{-1}, \ldots, z_m^{-1}]$$

$$\text{such that} \quad \phi(w_j) = \prod_{i=1}^{m} z_i^{a_{ij}}$$

The elements in a column j of matrix A is written as $a_j = a_j^+ - a_j^-$ with $a_j^+, a_j^- \geq 0$.

Now, construct the following ideal. Note that the last term $(1 - tz_1 \ldots z_m)$ is included to account for the inverse polynomial terms.

$$I = \langle z^{a_j^-} w_j - z^{a_j^+}, 1 - tz_1 \dots z_m \rangle$$

Do check out: 4ti2. It is a neat package which can be used for algebraic-geometric computations for optimization problems. It can be used with low-level programming languages like C++ or at a higher level using SageMath.

2.2.4 Solving the IP

Just to recap so far, we have constructed the ideal, which in very simple terms has the all the information regarding the constraint matrix A. Now, if you have noticed, we have not factored the RHS b or the cost vector c yet. We shall do so now.

First, now that we have the ideal, let's compute the Gröbner basis using the Buchberger's algorithm. We need a term ordering for this, the intuition behind this term-ordering will be explained in the next section. For now, let's just have the reduced Gröbner basis (we choose the reduced Gröbner basis because it is unique for a given ideal).

The algorithm to solve the IP is as follows:

$G \leftarrow$ `Reduced Gröbner Basis computed for the ideal;`
if $g = NormalForm(\prod_{i=1}^{m} z_i^{b_i}, G) \in k[w]$ **then**
  $g \leftarrow w_1^{x_1^*} \dots w_n^{x_n^*}$;
  `return` $(x_1^*, \dots, x_n^*)$
else
  `There is no feasible solution.`
end

To oversimplify things, the NormalForm algorithm is analogous to doing a division and getting a remainder.

Observe that the major computations do not involve the RHS b. We just need b to obtain the normal form. Since the Gröbner basis does not depend on b, it is all the more convenient to parametrize b now and obtain solutions for IP(b). This allows us to solve a family of IPs easily.

The main takeaway from this section is the above fact. In the next section, the major focus would be to demonstrate the geometric intuition behind the optimization using this method.

2.2.5 Geometric intuition behind the approach

We need a few definitions, before we can dive in.

Definition 2.1 (b-fiber)**.** A b-fiber is defined as the set of all feasible solutions to the $IP_{A,c}(b)$ for a particular RHS b.

Definition 2.2 (Term Order)**.** A term ordering, represented as $>$ on $\mathbb{N}^n$ has the following properties:

1. $>$ is a total order on $\mathbb{N}^n$
2. $\alpha > \beta \implies \alpha + \gamma > \beta + \gamma$
3. $\alpha > 0 \quad \forall \alpha \in \mathbb{N}^n \setminus \{0\}$

To understand the intuition behind solving IP problems with varying RHS, we briefly discuss Rekha Thomas' paper on the Geometric Buchberger's algorithm. She essentially shows that the reduced Gröbner basis forms a unique minimal test set and outlines the algorithm to construct it. But, keeping that aside, we shall take the algorithm as given and pay more attention to the geometric intuitions in her paper.

Each point in $\mathbb{N}^n$ is a feasible solution in a unique fiber of $\mathtt{IP}_{\{A,c\}}$. To elaborate, we can always say that α is a feasible solution in the $A\alpha$-fiber of $\mathtt{IP}_{\{A,c\}}$ right? Let's try to order the points in $\mathbb{N}^n$ in increasing order of cost value (i.e, $c.x$). It is possible for more than one lattice point to have the same cost value, hence this ordering may not be a total order on $\mathbb{N}^n$. By utilising a term ordering $>$ we can break ties amongst lattice points with the same cost values to create a total ordering. Let $>_c$ be the composite total order that first compares the cost of two points and breaks ties as per the term order. We can consider $>_c$ as a refinement of the cost function. Note that this composite term order may not always satisfy point number (3) in 2.2 (costs may be in such a manner that we indeed get 0).

We can replace the objective function by the refinement $>_c$ and not disturb the problems formulation. As we shall soon see, due to the total ordering, we are bound to reach the unique optimum (if it exists) in every fiber.

Lemma 1. *There exists a unique, minimal, finite set of vectors $\alpha(1), \dots, \alpha(t) \in \mathbb{N}^n$ such that the set of all nonoptimal solutions in all fibers of $\mathtt{IP}_{\{A,c\}}$ is a subset of $\mathbb{N}^n$ of the form:*

$$\mathcal{L}_{>_c} = \bigcup_{i=1}^{t} (\alpha(i) + \mathbb{N}^n)$$

Definition 2.3. A set $\mathcal{G} \subset \{x \in \mathbb{Z}^n : Ax = 0\}$ is a test set for $\mathtt{IP}_{\{A,>_c\}}$ if

1. For each non-optimal solution α to each program in $\mathtt{IP}_{\{A,>_c\}}$ there exists $g \in \mathcal{G}$ such that $\alpha - g$ is a feasible solution to the same program with $\alpha >_c \alpha - g$.
2. For the optimal solution, say β to a program in $\mathtt{IP}_{\{A,>_c\}}$, $\beta - g$ is infeasible for every $g \in \mathcal{G}$.

Consider the problem $\mathtt{IP}_{\{A,>_c\}}(b)$ for which α is a feasible solution. If α is non-optimal, using the above definition, we can find an element $g \in \mathcal{G}$ such that $\alpha - g$ is another feasible solution with the property: $\alpha >_c \alpha - g$. In case α is the optimal solution, any further movement, would lead us straight to an infeasible solution (see (2) in 2.3).

Definition 2.4. Let $\mathcal{G}_{>_c} = g_i = (\alpha(i) - \beta(i)), i = 1, \ldots, t$ where $\alpha(1), \ldots, \alpha(t)$ are the unique minimal elements of $\mathcal{L}_c$ and $\beta(i)$ is the unique optimum to the $\mathtt{IP}_{\{A,>_c\}}(A\alpha(i))$.

The set $\mathcal{G}_{>_c}$ is the reduced Gröbner basis with respect to $>_c$ obtained using the Geometric Buchberger's algorithms. Note that, $(\alpha(i) - \beta(i))$ is a lattice point in the kernel. Based on this fact, one should be clearly able to see why this approach is particularly useful with regard to its independence of the right hand sides (b). Geometrically, it is represented as a directed line segment from $\alpha(i)$ to $\beta(i)$.

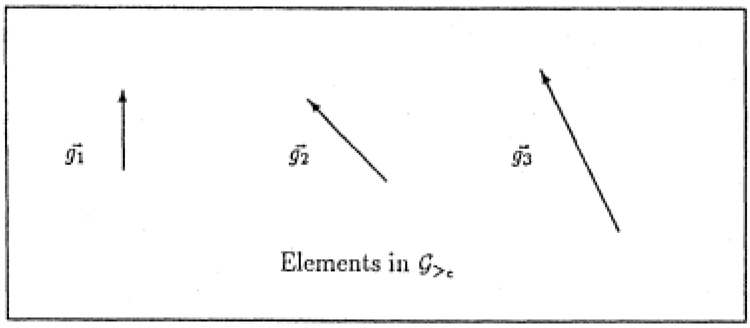
Using the theory above, we can build a connected, directed graph in every fiber of $\mathtt{IP}_{\{A,c\}}$. The nodes of the graph are all the lattice points in the fiber and the edges are the translations of elements in $\mathcal{G}_{>_c}$ by non-negative integral vectors. The graph in a fiber has unique sink at the unique optimum. After building this graph, starting out any non-optimal feasible solution, we can traverse the graph and reach a unique optimal solution. Along the way, our objective function values of the lattice points decrease monotonically from lattice points α to β.

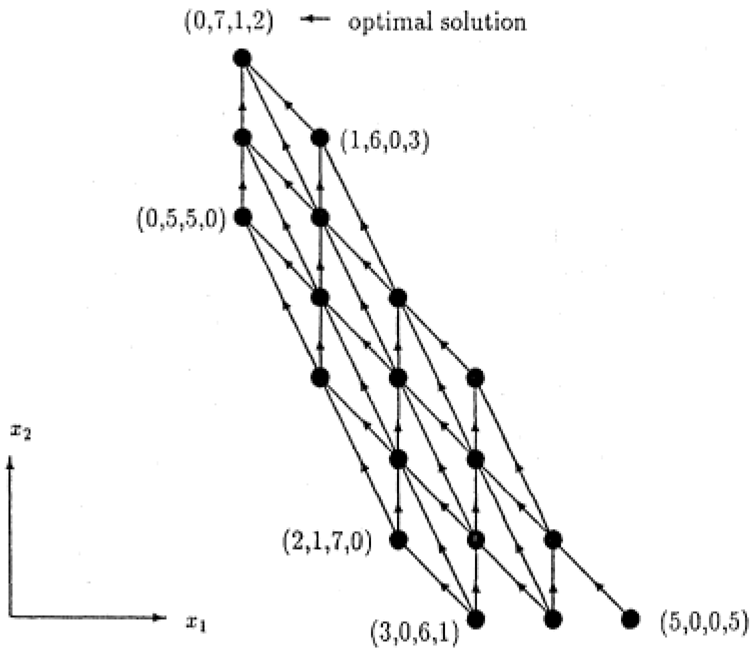
Now to clearly define the role of the reduced Gröbner basis. It is the unique minimal test set for $\mathtt{IP}_{\{A,>_c\}}$. It depends only on the matrix A and the refined objective function $>_c$. Consider the problem.

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \end{bmatrix}, \quad c = \begin{bmatrix} 1 & 13 & 14 & 17 \end{bmatrix}$$

Using the reverse lexicographic ordering as the tie-breaking term-ordering, the elements in the reduced Gröbner basis are given by:

$$\vec{g_1} = \overrightarrow{(0,0,2,0),(0,1,0,1)}$$
$$\vec{g_2} = \overrightarrow{(1,0,0,1),(0,1,1,0)}$$
$$\vec{g_3} = \overrightarrow{(1,0,1,0),(0,2,0,0)}$$





With these vectors, we start at a feasible solution and reach the optimum by traversing the directed graph. But this begs the question: how do we get the initial feasible solution to start with?

To address the above question, we need an extended IP for solving the optimization problem. Consider the below IP which has three extra components M, I, and y. I is the identity matrix and y is the extra dummy variable of our extended IP.

$$
\begin{aligned}
\min_{y,\,x} \quad & My + cx \\
\text{s.t.} \quad & Iy + Ax = b, \\
& (y, x) \in \mathbb{N}^{m+n}
\end{aligned}
$$

We choose the vector M to have a high cost so as to penalize our search by trying to exclude non-zero y lattice points during our search. In this extended IP, we have a starting feasible solution ($y = b$, $x = 0$). By taking a directed path to the optimal value, if we reach an optimal point where y is not equal to 0, then we don't have any feasible solution for our IP.

If interested, please refer to the paper by Rekha Thomas [6] for the Geometric Buchberger's algorithm.

2.3 Applications of the Gröbner basis in solving Stochastic Programs with Integer Recourse

Having seen how the Gröbner basis is useful for a family of IPs with varying RHS constraints, we shall study how to exploit this approach for Stochastic Programs. Referring to the paper by Rüdiger Schultz [4], we first formulate the two-stage stochastic programming problem with integer recourse as follows:

$$\min \{cx + Q(x) : x \in C\} \tag{1}$$

where,

$$Q(x) = \mathbb{E}_{\xi} v(Tx - \xi) \tag{2}$$

$$v(s) = \min (\tilde{q}y : Wy \geq s, y \in \mathbb{Z}_+^m) \tag{3}$$

ξ is a random vector in $\mathbb{R}^p$, and $\mathbb{E}_{\xi}$ is the expectation wrt to ξ.

All matrices/vectors except W have real elements. W can be a rational matrix, but Schultz shows the functioning of the algorithm only when W takes $\mathbb{Z}$ or can be scaled appropriately, which is exactly the case we have seen.
Eq 3 can be rewritten as

$$Wy = s \qquad \text{where,} \quad s = \lceil Tx - \xi \rceil \tag{4}$$

Observe that s is varying now. The primary interest in the second stage optimization (3) is to leverage the use of Gröbner Basis in solving multiple problems with varying RHS and a fixed W, similar to the discussion on [2.2.3].

The overall algorithm presented by Schultz [4] for solving Stochastic Programs with Integer Recourse is as follows.

1. Compute the Gröbner basis for the second stage IP problem using a **"compatible"** term ordering.
2. Solve the continuous relaxation of [1] 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.

To understand the nitty gritties of level sets and the enumeration scheme, refer Schulz's paper [4].

2.4 Term Ordering

Before we jump to the next section, let's briefly discuss what a "compatible" term ordering means. Refer to the paper by Conti and Traverso [1].

Definition 2.5. A term ordering is said to be compatible with a cost function, with respect to a linear map L, if $L(a) = L(b)$ and $c(a) > c(b)$ then $a > b$. (In the paper, the matrix A is taken as the linear map L.)

There are 2 main points to be discussed in this paper:

1. The authors prove a theorem which essentially states that if an optimal solution exists for the IP, an appropriate term ordering exists as well.
2. Futhermore, if the cost function is non-negative, the tie-breaking order we chose does not matter (from a computational perspective, there might some advantages though). However, if the cost is negative, we have to construct the term-ordering carefully as outlined by the steps in the paper.

3 Graver Basis

Having seen the reduced Gröbner basis act as a test set for a family of Integer Programming problems with varying right hand sides, we look more robust approaches which generalize to a greater extent. Specifically, we now turn our attention to the cost vector (c). Is it possible that the Gröbner basis $\mathcal{G}_{A,c}$ remains unchanged for certain values of the cost vector? Strumfels [5] shows that the Gröbner basis would not change when c ranges within the Gröbner cone which is defined by the following linear inequalities in the unknowns $c_1, \ldots, c_n$.

$$c.g \geq 0 \qquad \text{for all } g \text{ in } \mathcal{G}$$

The collection of Gröbner cones in $\mathbb{R}^n$ forms the Gröbner fan of the matrix A which is an important invariant in the study of the solution to the IP when both b and c change.

To understand if we can expand on our knowledge of test sets for such problems, let us first look at the Hilbert basis, and then use it construct the Graver basis.

3.1 Hilbert basis

$$ker_{\mathbb{N}}(A) = \{u \in \mathbb{N}^n : A.u = 0\}$$

$$\mathcal{H}_A = \{u \in ker_N(A) \setminus \{0\} : \text{ no element } v \in ker_N(A) \setminus \{0, u\} \text{ satisfies } v \leq u\}$$

Proposition 1. *The set $\mathcal{H}_A$ is finite. It is the unique minimal set such that every vector in $ker_N(A)$ is an $\mathbb{N}$-linear combination of elements in $\mathcal{H}_A$*

This finite set $\mathcal{H}_A$ is called the Hibert basis of matrix A.

3.2 Graver basis

Let D_σ be a $n \times n$ diagonal matrix with each entry being $\{+1, -1\}$. The Graver basis of A ($\mathcal{GR}_A$) is given as:

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

We are taking the union over the 2^n Hibert bases for the various matrices $A.D_\sigma$. The signs are adjusted so that each Hibert basis lies in the kernel of the original matrix A. [1] ensures that $\mathcal{GR}_A$ is a finite subset of the kernel of A.

Proposition 2. *The Graver basis is a universal Gröbner basis. It contains up to negating vectors, the Gröbner basis of A for all cost functions.*

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

Let $A \in \mathbb{Z}^{m \times n}$, $b \in \mathbb{Z}^m$, $l, u \in \mathbb{Z}^n$ and an objective function $f : \mathbb{R}^n \to \mathbb{R}$. The Graver basis are particularly useful in solving IPs of the type:

$$(IP)_{A,b,l,u,f} : min\{f(z) : Az = b; l \leq z \leq u; z \in \mathrm{Z}^n\}$$

where, $f(.)$ is a separable convex function. A separable convex function can be written as $f(z) = \sum_{i=1}^{n} f_i(z_i)$, where each $f_i : \mathbb{R} \to \mathbb{R}$ is a convex function. It is obvious that $c^T z$ is a separable convex function. The Graver basis forms the optimality certificate or the test set for this family of IPs.

For augmentation algorithms (**Graver's Best Augmentation Algorithm**) and algorithms for the computation of the Graver basis (**Pottier's Algorithm**) see Chapter 3 of [3].

3.3 Graver bases for Two-Stage Stochastic Integer Programs

Having seen the uses of the Graver Basis for IPs, can we also somehow exploit the block-structured property of two-stage Stochastic IPs? Yes, we can.

Consider the N-fold 4-block decomposable program which is denoted as follows:

$$\begin{pmatrix} C & D \\ B & A \end{pmatrix}^{(N)} = \begin{pmatrix} C & D & D & \dots & D \\ B & A & 0 & \dots & 0 \\ B & 0 & A & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ B & 0 & 0 & \dots & A \end{pmatrix} \tag{5}$$

It is called N fold as there are N copies of matrices A, B, and D. 0 denotes a zero matrix.

Specifically, in the case of two-stage Stochastic Integer Programs, we do not consider C and D. Instead, A and B will be of interest to us. B is the matrix associated with the first-stage decision variables and A is associated with the decision variables in the second stage. The number of occurences of A is the same as the number of scenarios that arise once the first-stage decisions have been made.

With a greater ability to parametrize the solution brings with it more computational baggage. However, the Two Stage Stochastic Integer Program has structure that can be exploited. See [3] for more information.

Specifically, consider the discretized formulation of the problem:

$$min\{c^T x + \sum_{i=1}^{N} \pi^{(i)} d^T y^{(i)} : Mx = a,\ x \in \mathbb{Z}_+^m,\ Tx + Wy^{(i)} = \xi^{(i)},\ y^{(i)} \in \mathbb{Z}_+^n\ \forall i\} \tag{6}$$

$$\begin{bmatrix} B & A & 0 & \dots & 0 \\ B & 0 & A & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ B & 0 & 0 & \dots & A \end{bmatrix} \tag{7}$$

The sizes of the Graver bases of $\begin{pmatrix} \cdot & \cdot \\ B & A \end{pmatrix}^{(N)}$ are not polynomially bounded in N. In other words, they increase exponentially in N. However, the structure of the matrix $\begin{pmatrix} \cdot & \cdot \\ B & A \end{pmatrix}^{(N)}$ leads to a structure on the elements in $\mathcal{G}(\begin{pmatrix} \cdot & \cdot \\ B & A \end{pmatrix}^{(N)})$, which we can exploit to construct in $\mathcal{O}(N^2)$ many steps an augmentation vector for any given non-optimal feasible solution that is atleast as good as the Graver best augmentation step.

First, let us observe that $(x, y^{(1)}, \dots, y^{(n)})^T \in ker(\begin{pmatrix} \cdot & \cdot \\ B & A \end{pmatrix}^{(N)})$ if and only if $(x, y^{(i)})^T \in ker(\begin{pmatrix} \cdot & \cdot \\ B & A \end{pmatrix}^{(1)})$ for $i = 1, 2, \dots, N$. The basic idea is that we can construct from one Graver basis element many others by simply permuting the bricks. In other words, we can store exponentially many Graver basis elements by simply remembering the bricks they are composed of.

For further specific details refer [3]. In addition, they also show the time complexity in reaching the optimal solution using the Graver-best augmentation algorithm.

4 Papers to review

1. Thomas, Rekha R., and Robert Weismantel. "Truncated Gröbner bases for integer programming." Applicable Algebra in Engineering, Communication and Computing 8.4 (1997): 241-256.

2. Urbaniak, Regina, Robert Weismantel, and Günter M. Ziegler. "A variant of the Buchberger algorithm for integer programming." SIAM Journal on

Discrete Mathematics 10.1 (1997): 96-108.

3. Weismantel, Robert. "Test sets of integer programs." Mathematical Methods of Operations Research 47.1 (1998): 1-37.

4. Blanco, Víctor, and Justo Puerto. "Partial Gröbner bases for multiobjective integer linear optimization." SIAM Journal on Discrete Mathematics 23.2 (2009): 571-595.

5. Malkin, Peter. Computing Markov bases, Gröbner bases, and extreme rays. Diss. Catholic University of Louvain, Louvain-la-Neuve, Belgium, 2007.

6. Li, Qiang, et al. "Minimised geometric Buchberger algorithm for integer programming." Annals of Operations Research 108.1 (2001): 87-109.

7. Malkin, Peter N. "Truncated Markov bases and Gröbner bases for Integer Programming." arXiv preprint math/0612615 (2006).

8. Sen, S., Higle, J. L. (2005). The C 3 theorem and a D 2 algorithm for large scale stochastic mixed-integer programming: Set convexification. Mathematical Programming, 104(1), 1-20.

9. Sturmfels, Bernd, and Rekha R. Thomas. "Variation of cost functions in integer programming." Mathematical Programming 77, no. 2 (1997): 357-387.

10. Diaconis, P., Eisenbud, D., Sturmfels, B. (1998). Lattice walks and primary decomposition. In Mathematical essays in honor of Gian-Carlo Rota (pp. 173-193). Birkhäuser Boston.

11. Kong, N., Schaefer, A. J., Hunsaker, B. (2006). Two-stage integer programs with stochastic right-hand sides: a superadditive dual approach. Mathematical Programming, 108(2), 275-296.

12. Hemmecke, R., Schultz, R. (2003). Decomposition of test sets in stochastic integer programming. Mathematical Programming, 94(2), 323-341.

13. Hoşten, S., Thomas, R. R. (1999). Standard pairs and group relaxations in integer programming. Journal of Pure and Applied Algebra, 139(1-3), 133-157.

14. Biase, F. D., Urbanke, R. (1995). An algorithm to calculate the kernel of certain polynomial ring homomorphisms. Experimental Mathematics, 4(3), 227-234.

15. Hemmecke, R., Schultz, R. (2001). Decomposition methods for two-stage stochastic integer programs. In Online optimization of large scale systems (pp. 601-622). Springer, Berlin, Heidelberg.

16. Hemmecke, R., Malkin, P. N. (2009). Computing generating sets of lattice ideals and Markov bases of lattices. Journal of Symbolic Computation, 44(10), 1463-1476.

17. Thomas, R. R. (2005). The structure of group relaxations. Handbooks in operations research and management science, 12, 123-170.

18. Haneveld, W. K. K., van der Vlerk, M. H. (1999). Stochastic integer programming: General models and algorithms. Annals of operations research, 85, 39.

19. Sturmfels, B., Weismantel, R., Ziegler, G. M. (1994). Gröbner bases of lattices, corner polyhedra, and integer programming.

20. Weismantel, R. (1998). Test sets of integer programs. Mathematical Methods of Operations Research, 47(1), 1-37.

21. Thomas, R. R. (1998). Applications to integer programming. In Proceedings of Symposia in Applied Mathematics (Vol. 53, pp. 119-142). American Mathematical Society.

22. Blanco, V., Puerto, J. (2012). An application of integer programming to the decomposition of numerical semigroups. SIAM Journal on Discrete Mathematics, 26(3), 1210-1237.

23. Li, Q., Guo, Y. K., Darlington, J., Ida, T. (2001). Minimised geometric Buchberger algorithm for integer programming. Annals of Operations Research, 108(1), 87-109.

24. Briales-Morales, E., Campillo-López, A., Pisón-Casares, P., Vigneron-Tenorio, A. (2003). Minimal resolutions of lattice ideals and integer linear programming. Revista Matemática Iberoamericana, 19(2), 287-306.

25. Bogart, T. (2007). Problems in computational algebra and integer programming. University of Washington.

26. Linfoot, A. J. (2006). A case study of a multithreaded Buchberger normal form algorithm. The University of Arizona.

27. Ito, M., Hirabayashi, R. (2003). Computing test sets for integer programming. Optimization Methods and Software, 18(4), 437-452.

28. Peifer, D. J. (2021). Reinforcement Learning in Buchberger's Algorithm (Doctoral dissertation, Cornell University).

29. Hashemi, A. (2014). Solving linear systems of equations over integers with Gröbner bases. Acta Arithmetica, 163, 261-270.

References

[1] P. Conti and C. Traverso. Buchberger algorithm and integer programming. In *International Symposium on Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes*, pages 130–139. Springer, 1991.

[2] J. L. D. O. Cox, D. *Ideals, Varieties and Algorithms*. 1997.

[3] J. A. De Loera, R. Hemmecke, and M. Köppe. *Algebraic and geometric ideas in the theory of discrete optimization*. SIAM, 2012.

[4] R. Schultz, L. Stougie, and M. H. V. D. Vlerk. Solving stochastic programs with integer recourse by enumeration: a framework using gröbner basis reductions, 1995.

[5] B. Sturmfels. Algebraic recipes for integer programming. In *Proceedings of Symposia in Applied Mathematics*, volume 61, pages 99–113, 2004.

[6] R. R. Thomas. A geometric buchberger algorithm for integer programming. *Mathematics of Operations Research*, 20(4):864–884, 1995.