

PARTIAL GRÖBNER BASES FOR MULTIOBJECTIVE INTEGER LINEAR OPTIMIZATION*

VÍCTOR BLANCO[†] AND JUSTO PUERTO[†]

Abstract. This paper presents a new methodology for solving multiobjective integer linear programs (MOILP) using tools from algebraic geometry. We introduce the concept of partial Gröbner basis for a family of multiobjective programs where the right-hand side varies. This new structure extends the notion of Gröbner basis for the single objective case to the case of multiple objectives, i.e., when there is a partial ordering instead of a total ordering over the feasible vectors. The main property of these bases is that the partial reduction of the integer elements in the kernel of the constraint matrix by the different blocks of the basis is zero. This property allows us to prove that this new construction is a test family for a family of multiobjective programs. An algorithm “á la Buchberger” is developed to compute partial Gröbner bases, and two different approaches are derived, using this methodology, for computing the entire set of Pareto-optimal solutions of any MOILP problem. Some examples illustrate the application of the algorithm, and computational experiments are reported on several families of problems.

Key words. multiple objective optimization, integer programming, Gröbner bases, test sets

AMS subject classifications. 90C29, 90C10, 13P10

DOI. 10.1137/070698051

1. Introduction. The multiobjective paradigm appeared in economic theory in the nineteenth century in the seminal works by Edgeworth [14] and Pareto [30] to define an economic equilibrium. Mathematically, the multiobjective optimization approach consists of determining the maximal (minimal) elements of a partially ordered set. This problem was already addressed by Cantor [7], Cayley [8], and Hausdorff [21] at the end of the nineteenth century. Since then, multiobjective programming (including multicriteria optimization) has been a fruitful research field within the areas of applied mathematics, operations research, and economic theory. Excellent textbooks and survey papers are available in the literature; the interested reader is referred to the books by Sawaragi, Nakayama, and Tanino [32], Chankong and Haimes [9], Yu [45], Miettinen [28], or Ehrgott, Figueira, and Gandibleux [19], and to the surveys in [17, 18].

The importance of multiobjective optimization is not due only to its theoretical implications but also to its many applications. Witnesses of that are the large number of real-world decision problems that appear in the literature formulated as multiobjective programs. These include flowshop scheduling [24], analysis in finance [17], railway network infrastructure capacity [13], vehicle routing problems [25, 34], or trajectory optimization [36], among many others.

Multiobjective programs are formulated as optimization (without loss of generality, we restrict ourselves to the minimization case) problems over feasible regions with at least two objective functions. Usually, it is not possible to minimize all of the objective functions simultaneously, since the objective functions induce a partial

*Received by the editors July 23, 2007; accepted for publication (in revised form) October 17, 2008; published electronically February 6, 2009. This research was partially supported by Ministerio de Educación y Ciencia under grant MTM2007-67433-C02-01.

<http://www.siam.org/journals/sidma/23-2/69805.html>

[†]Departamento de Estadística e Investigación Operativa, Universidad de Sevilla, 41012 Sevilla, Spain (vblanco@us.es, puerto@us.es).

order over the vectors in the feasible region, so a different notion of solution is needed. A feasible vector is said to be Pareto-optimal (or nondominated) if no other feasible vector has componentwise smaller objective values, with at least one strict inequality.

This paper studies multiobjective integer linear programs (MOILP). Thus, we assume that all objective functions and constraints that define the feasible region are linear and that the feasible vectors have nonnegative integer components.

There are nowadays several exact methods to solve MOILP problems. Two of them, multiobjective implicit enumeration [46, 47] and multiobjective dynamic programming [26], claim to be of general use and have attracted the attention of researchers in the last several years. Nevertheless, although, in principle, they may be applied to any number of objectives, one can mainly find, in the literature, applications to biobjective problems. Moreover, some other methods even do not provide the entire set of Pareto-optimal solutions, but provide the supported ones (those that can be obtained as solutions of linearly scalarized programs).

On the other hand, there are several methods that apply to biobjective problems but that do not extend to the general case. Thus, one can see that there are two thresholds in multiobjective programming: a first step from one to two objectives, and a second, and deeper one, from two to more than two objectives.

In terms of complexity, it is worth noting that most MOILP problems are NP-hard [16]. Even when the single-objective problem is polynomially solvable, the multiobjective version may become NP-hard. This is the case of spanning tree [20] and minimum-cost flow problems [15], among others. Therefore, computational efficiency is not an issue when analyzing MOILP. The important point is to develop tools that can handle these problems and that give insights into their intrinsic nature. The goal of this paper is to present a new general methodology for solving MOILP using tools borrowed from algebraic geometry. The usage of algebraic geometry tools in integer programming (single criterion) is not new (see [10], [22], [41], [23], [44], [43]). The main idea is to compute a Gröbner basis for certain toric ideals (related to the constraints matrix) with a monomial order induced by the objective function.

Gröbner bases were introduced by Buchberger in 1965 in his Ph.D. thesis [6]. He named them Gröbner bases paying tribute to his advisor Wolfgang Gröbner. This theory emerged as a generalization, from the one variable case to the multivariate polynomial case, of the greatest common divisor. One of the outcomes of Gröbner bases theory was its application to integer programming, first published by Conti and Traverso [10]. This paper opened a new research line, followed by a number of authors, consisting of the application of algebraic geometry tools for solving integer programs.

In [22], Hoşten and Sturmfels gave two ways to implement the Conti and Traverso algorithm that improve in some cases the branch-and-bound algorithm to exactly solve integer programs. Thomas presented in [41] a geometric point of view of the Buchberger algorithm as a method to obtain solutions of an integer program. Later, Thomas and Weismantel [43] improved the Buchberger algorithm in its application to solve integer programs introducing truncated Gröbner bases. At the same time, Urbaniak, Weismantel, and Ziegler [44] published a clear geometric interpretation of the reduction steps of this kind of algorithm in the original space (decision space). The interested reader can find excellent descriptions of this methodology in the books by Adams and Loustaunau [2], Sturmfels [37], Cox, Little, and O'Shea [12], or Bertsimas and Weismantel [5], and in the papers by Aardal, Weismantel, and Wolsey [1], Sturmfels [38, 39], Sturmfels and Thomas [40], and Thomas [42].

Our main contribution is to adapt some of the above-mentioned tools from algebraic geometry to solve MOILP problems. We present an algorithm to exactly solve multiobjective problems, i.e., providing the whole set of Pareto-optimal solutions (supported and nonsupported ones). One of the main advantages of our approach is that the number of objective functions does not significantly increase the difficulty. A new geometric approach of the notion of reduction based on a partial ordering is given. This reduction allows us to extend the concept of Gröbner basis when a partial ordering rather than a total ordering is considered over \mathbb{N}^n . We call these new structures partial Gröbner bases or p-Gröbner bases. We prove that p-Gröbner bases can be generated in a finite number of steps by a variation of the Buchberger algorithm. The main property of a p-Gröbner basis being that, for each pair in $\mathbb{Z}^n \times \mathbb{Z}_+^n$ with first component in $\text{Ker}(A)$, the reduction by maximal chains in the basis is the zero set.

We propose two versions of the same algorithm to solve multiobjective integer programs based on this new construction. Our first approach consists of three stages. The first one uses only the constraint matrix of the problem, and it produces a system of generators for the toric ideal \mathfrak{S}_A (or its geometric representation I_A). In the second step, a p-Gröbner basis is built using the initial basis given by the system of generators computed in the first step. This step requires us to fix the objective matrix, since it induces the partial order used in the reduction steps. Once the right-hand side vector is fixed, in the third step, the Pareto-optimal solutions are obtained. This computation uses the new concept of partial reduction of an initial feasible solution by the p-Gröbner basis.

This algorithm extends, to some extent, Hoşten and Sturmfels' algorithm [22] for integer programs because if we apply our method to single-objective problems, partial reductions and p-Gröbner bases coincide with the standard notions of reductions and Gröbner bases, respectively.

Our second approach is based on the original idea by Conti and Traverso [10]. It consists of using the big-M method that results in an increasing number of variables, in order to have an initial system of generators. Moreover, this approach also provides an initial feasible solution. Therefore, the first step in the former variant of the algorithm can be ignored, and the third step is highly simplified. In any case, our first version (the one extending the Hoşten and Sturmfels approach) has proved to be more efficient than this second one, since the computation of a p-Gröbner basis is highly sensitive to the number of variables.

Both algorithms have been implemented in MAPLE 10. We report on some computational experiments based on the first version of the algorithm and on two different families of problems with different number of objective functions.

The rest of the paper is organized as follows. In section 2 we give the notation, the formulation of the problem, and its algebraic codification. We also introduce here the notion of test family and its geometric description. Section 3 presents the definition of p-Gröbner basis, based on the notion of partial reduction. Here, we also state the relationship between test families and p-Gröbner bases: the reduced p-Gröbner basis for a family of multiobjective programs varying the right-hand side coincides with the minimal test family for that family. At the end of the section, an example illustrates all of the above concepts. Section 4 is devoted to the results of the computational experiments and its analysis. Here, we solve several families of MOILP, report on the performance of the algorithms, and draw some conclusions on its results and their implications.

2. The problem and its translation. The goal is to solve the MOILP in its standard form:

$$(1) \quad \begin{aligned} & \min (c_1 x, \dots, c_k x) \\ & \text{subject to (s.t.) } \sum_{j=1}^n a_{ij} x_j = b_i, i = 1, \dots, m, \\ & \quad \quad \quad x_j \in \mathbb{Z}_+, \quad j = 1, \dots, n, \end{aligned}$$

with b_i nonnegative integers for $i = 1, \dots, m$, $c_l \in \mathbb{Z}_+^n$ for $l = 1, \dots, k$, $x = (x_1, \dots, x_n)$, and the constraints define a polytope (bounded). For the sake of simplicity, at times, we use a vector notation and denote $A = (a_{ij}) \in \mathbb{Z}^{m \times n}$, $b = (b_i) \in \mathbb{Z}_+^m$, and $C = (c_{ij}) \in \mathbb{Z}_+^{k \times n}$. In the following, problem (1) is referred to as $MIP_{A,C}(b)$, and we denote by $MIP_{A,C}$ the family of multiobjective problems where the right-hand side varies.

The reader may note that there is no loss of generality in our approach to multiobjective integer linear programming, since any general MOILP problem with inequality constraints and rational components in A , b , and C can be transformed to a problem in the above standard form.

It is clear that the problem $MIP_{A,C}(b)$ is not a usual optimization problem since the objective function is a vector, thus inducing a partial order among its feasible solutions. Hence, solving the above problem requires an alternative concept of solution, namely, the set of nondominated or Pareto-optimal points (vectors).

A feasible vector $\hat{x} \in \mathbb{R}^n$ is said to be a *Pareto-optimal* or *nondominated solution* of $MIP_{A,C}(b)$ if there is no other feasible vector y such that

$$c_j y \leq c_j \hat{x} \quad \text{for all } j = 1, \dots, k$$

with at least one strict inequality for some j .

If x is a Pareto-optimal solution, the vector $(c_1 x, \dots, c_k x) \in \mathbb{R}^k$ is called *efficient*.

We say that a feasible point y is *dominated* by a feasible point x if $c_i x \leq c_i y$ for all $i = 1, \dots, k$, with at least one strict inequality. According to the above concept, solving a multiobjective problem consists of finding its entire set of Pareto-optimal solutions, including those that have the same objective values.

From the objective function C , we obtain a partial order over \mathbb{Z}^n as follows:

$$x \prec_C y : \iff Cx \not\leq Cy \quad \text{or} \quad x = y,$$

where $Cx \not\leq Cy$ stands for $Cx \leq Cy$ and $Cx \neq Cy$.

Observe that since $C \in \mathbb{Z}_+^{k \times n}$, the above relation is not complete. Hence, there may exist incomparable vectors (those $x, y \in \mathbb{Z}_+^n$ such that neither $x \prec_C y$ nor $y \prec_C x$). We use this partial order induced by the objective function of problem $MIP_{A,C}$ as the input for the multiobjective integer programming algorithm developed in this paper.

Remark 2.1. Note that distinct solutions with the same objective values are incomparable under \prec_C . This order can be refined so that those solutions with the same objective values are comparable. Consider the binary relation

$$x \preceq_C y : \iff \begin{cases} Cx \not\leq Cy, & \text{or} \\ Cx = Cy \text{ and } x \prec_{lex} y. \end{cases}$$

This alternative order allows us to rank those solutions that have the same objective values using the lexicographical order of their components.

The above partial order \preceq_C permits us to solve a simplified version of the multi-objective problem after introducing the following equivalence relation in \mathbb{Z}^n :

$$x \sim_C y : \iff Cx = Cy.$$

In this version, we obtain solutions in \mathbb{Z}^n / \sim_C . The reader may note that when solving the problem with the order \preceq_C , one would obtain only a representative element of each class of Pareto-optimal solutions (the lexicographically smallest). With those efficient values $\{v_1, \dots, v_t\}$, the remaining solutions can be obtained solving the following system of diophantine equations, in x , for each $v_i, i = 1, \dots, t$:

$$\begin{cases} Cx = v_i, \\ Ax = b, \\ x \in \mathbb{Z}_+^n. \end{cases}$$

Remark 2.2. In some cases, the order \prec_C can be refined to be adapted to specific problems. This is the case when slack variables appear in mathematical programs. Two feasible solutions (x, s_1) and (x, s_2) , where s_1 and s_2 are the slack components, have the same objective values. The order \prec_C considers both solutions as incomparable, although they are the same because we are looking just for the x -component of the solution. In these cases, we consider the following refined partial order in $\mathbb{Z}^n \times \mathbb{Z}^r$:

$$(x, s) \prec_C^s (y, s') : \iff \begin{cases} Cx \preceq_C Cy, & \text{or} \\ Cx = Cy \text{ and } s \prec_{lex} s', \end{cases}$$

where $x, y \in \mathbb{Z}_+^n$ are the actual decision variables and $s, s' \in \mathbb{Z}_+^r$ are the slack variables of our problem.

In the following, we will use partial order \prec_C unless it is explicitly specified.

Our matrix A is encoded in the set

$$(2) \quad J_A = \{\{u, v\} : u, v \in \mathbb{N}^n, u - v \in \text{Ker}(A)\}.$$

Let $\pi : \mathbb{N}^n \rightarrow \mathbb{Z}^n$ denote the map $x \mapsto Ax$. Given a right-hand side vector b in \mathbb{Z}^n , the set of feasible solutions to $MIP_{A,C}(b)$ constitutes $\pi^{-1}(b)$, the preimage of b under this map. In the rest of the paper, we identify the discrete set of points $\pi^{-1}(b)$ with its convex hull, and we call it the b -fiber of $MIP_{A,C}$. Thus, $\pi^{-1}(b)$ or the b -fiber of $MIP_{A,C}$ is the polyhedron defined by the convex hull of all feasible solutions to $MIP_{A,C}(b)$.

For any pair $\{u, v\}$, with $u, v \in \mathbb{N}^n$, we define the set $setlm(u, v)$ as follows:

$$setlm(u, v) = \begin{cases} \{u\} & \text{if } v \prec_C u, \\ \{v\} & \text{if } u \prec_C v, \\ \{u, v\} & \text{if } u \text{ and } v \text{ are incomparable by } \prec_C. \end{cases}$$

The reader may note that $setlm(u, v)$ is the set of degrees of the leading monomials according to identification $\{u, v\} \mapsto x^u - x^v \in \mathbb{R}[x_1, \dots, x_n]$, induced by partial order \prec_C .

From the above definition, $setlm(u, v)$ may have more than one leading term, since \prec_C is only a partial order. To account for all this information, we denote by $\mathcal{F}(u, v)$ the set of triplets

$$\mathcal{F}(u, v) = \{(u, v, w) : w \in setlm(u, v)\}.$$

The above concept extends to any finite set of pairs of vectors in \mathbb{N}^n , accordingly. For a pair of sets $\mathbf{u} = \{u_1, \dots, u_t\}$ and $\mathbf{v} = \{v_1, \dots, v_t\}$, the corresponding set of ordered pairs is

$$\mathcal{F}(\mathbf{u}, \mathbf{v}) = \{(u_i, v_i, w) : w \in \text{setlm}(u_i, v_i), i = 1, \dots, t\}.$$

$\mathcal{F}(\mathbf{u}, \mathbf{v})$ can be partially ordered based on the third component of its elements. Therefore, we can see $\mathcal{F}(\mathbf{u}, \mathbf{v})$ as a directed graph $G(E, V)$, where V is identified with the elements of $\mathcal{F}(\mathbf{u}, \mathbf{v})$ and $((u_i, v_i, w'), (u_j, v_j, w)) \in E$ if $(u_i, v_i, w), (u_j, v_j, w') \in V$ and $w' \prec_C w$. We are interested in the maximal ordered chains of G . Note that they can be efficiently computed by different methods, e.g., [4], [33].

The above concepts are clarified in the following example.

Example 2.1. Let $\mathbf{u} = \{(2, 3), (0, 2), (3, 0), (2, 1), (1, 1)\}$, $\mathbf{v} = \{(1, 4), (1, 3), (4, 2), (1, 2), (1, 0)\}$, and \prec_C be the partial order induced by the matrix

$$C = \begin{bmatrix} 2 & 1 \\ 3 & 5 \end{bmatrix},$$

then, $\text{setlm}((2, 3), (1, 4)) = \{(2, 3), (1, 4)\}$, $\text{setlm}((0, 2), (1, 3)) = \{(1, 3)\}$, $\text{setlm}((3, 0), (4, 2)) = \{(4, 2)\}$, $\text{setlm}((2, 1), (1, 2)) = \{(2, 1), (1, 2)\}$, and $\text{setlm}((1, 1), (1, 0)) = \{(1, 1)\}$. Now, by definition, we have

$$\begin{aligned} \mathcal{F}(\mathbf{u}, \mathbf{v}) = & \{((2, 3), (1, 4), (2, 3)), ((2, 3), (1, 4), (1, 4)), ((0, 2), (1, 3), (1, 3)), \\ & ((3, 0), (4, 2), (4, 2)), ((2, 1), (1, 2), (2, 1)), ((2, 1), (1, 2), (1, 2)), \\ & ((1, 1), (1, 0), (1, 1))\}. \end{aligned}$$

Figure 1 corresponds to the directed graph associated with $\mathcal{F}(\mathbf{u}, \mathbf{v})$, according to the partial ordering induced by C . There are four maximal chains:

$$M_1 = \{((3, 0), (4, 2), (4, 2)), ((2, 3), (1, 4), (2, 3)), ((0, 2), (1, 3), (1, 3)), ((2, 1), (1, 2), (2, 1)), ((1, 1), (1, 0), (1, 1))\},$$

$$M_2 = \{((3, 0), (4, 2), (4, 2)), ((2, 3), (1, 4), (2, 3)), ((0, 2), (1, 3), (1, 3)), ((2, 1), (1, 2), (1, 2)), ((1, 1), (1, 0), (1, 1))\},$$

$$M_3 = \{((2, 3), (1, 4), (1, 4)), ((0, 2), (1, 3), (1, 3)), ((2, 1), (1, 2), (2, 1)), ((1, 1), (1, 0), (1, 1))\},$$

$$M_4 = \{((2, 3), (1, 4), (1, 4)), ((0, 2), (1, 3), (1, 3)), ((2, 1), (1, 2), (1, 2)), ((1, 1), (1, 0), (1, 1))\}.$$

For any pair of sets $\mathbf{u} = \{u_1, \dots, u_t\}$ and $\mathbf{v} = \{v_1, \dots, v_t\}$, with $\{u_i, v_i\} \in J_A$, for all $i = 1, \dots, t$, the corresponding set $\mathcal{F}(\mathbf{u}, \mathbf{v})$ may also be seen as a set of pairs in $\mathbb{Z}^n \times \mathbb{Z}_+^n$ through the following map:

$$\begin{aligned} \phi: \mathbb{N}^n \times \mathbb{N}^n \times \mathbb{N}^n & \longrightarrow \mathbb{Z}^n \times \mathbb{Z}_+^n \\ (u, v, w) & \longmapsto (u - v, w). \end{aligned}$$

We denote by $I_A = \phi(\mathcal{F}(J_A))$, i.e.,

$$I_A = \{(u - v, w) : u - v \in \text{Ker}(A), w = \text{setlm}(u, v)\}.$$

It is clear that the maximal chains F_1, \dots, F_r of the image of $\mathcal{F}(\mathbf{u}, \mathbf{v})$ under ϕ with respect to the order \prec_C over the second components satisfy the following properties:

1. F_i is totally ordered by the second components with respect to \prec_C for $i = 1, \dots, r$.

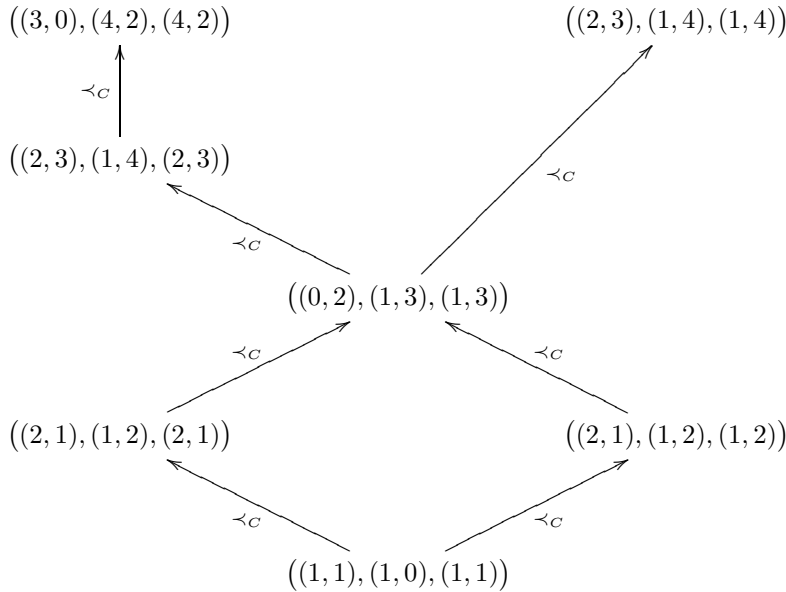


FIG. 1. Hasse diagram of the graph associated with the data in Example 2.1.

2. For all $(\alpha, \beta) \in F_i, i = 1, \dots, r, A(\beta - \alpha) = A\beta$.

The map ϕ and the above properties allow us to define the notion of test family for $MIP_{A,C}$. This notion is analogous to the concept of test set for a family of single objective integer programs when we have a partial order rather than a total order over \mathbb{N}^n [41]. Test families are instrumental for finding the Pareto-optimal set of each member $MIP_{A,C}(b)$ of the family of MOILP.

DEFINITION 2.1 (test family). A finite collection $\mathcal{G} = \{\mathcal{G}_C^1, \dots, \mathcal{G}_C^r\}$ of sets in $\mathbb{Z}^n \times \mathbb{Z}_+^n$ is a test family for $MIP_{A,C}$ if and only if

- (1) \mathcal{G}_C^j is totally ordered by the second component with respect to \prec_C for $j = 1, \dots, r$.
- (2) For all $(g, h) \in \mathcal{G}_C^j, j = 1, \dots, r, A(h - g) = Ah$.
- (3) If $x \in \mathbb{N}^n$ is a dominated solution for $MIP_{A,C}(b)$, with $b \in \mathbb{Z}_+^n$, there is some \mathcal{G}_C^j in the collection, and $(g, h) \in \mathcal{G}_C^j$ such that $x - g \prec_C x$.
- (4) If $x \in \mathbb{N}^n$ is a Pareto-optimal solution for $MIP_{A,C}(b)$, with $b \in \mathbb{Z}_+^n$, then for all $(g, h) \in \mathcal{G}_C^j$ and for all $j = 1, \dots, r$, either $x - g$ is infeasible or $x - g$ is incomparable to x .

Given a test family for $MIP_{A,C}$, there is a natural approach for finding the entire Pareto-optimal set. Suppose we wish to solve $MIP_{A,C}(b)$ for which x^* is a feasible solution.

If x^* is dominated, then there is some j and $(g, h) \in \mathcal{G}_C^j$ such that $x^* - g$ is feasible and $x^* - g \prec_C x^*$, whereas, for the remaining chains, there may exist some (g, h) such that $x^* - g$ is feasible but incomparable to x^* . We keep track of all of them.

If x^* is nondominated, we have to keep it as an element in our current solution set. Then, reducing x^* by the chains in the test family, we can only obtain either incomparable feasible solutions, that we maintain in our structure, or infeasible solutions that are discarded.

The above two cases lead us to generate the following set. From x^* , we compute the set of incumbent solutions:

$$IS(x^*) := \{y^* : y^* = x^* - g_{j_i}, (g_{j_i}, h_{j_i}) \text{ is the largest element } (g, h) \text{ in the chain } \mathcal{G}_C^i \text{ such that } x^* - g \text{ is feasible}, i = 1, \dots, r\}.$$

Now, the scheme proceeds recursively on each element of the set $IS(x^*)$. Finiteness of the above scheme is clear, since we are generating a search tree with bounded depth (cardinality of the test family) and bounded width, each element in the tree has at most r (number of chains) followers. The correctness of this approach is ensured, since any pair of Pareto-optimal solutions must be connected by a reduction chain through elements in the test family (see Theorem 2.1 and Corollary 2.1).

The above approach assumes that a feasible solution to $MIP_{A,C}(b)$ is known (thus implying that the problem is feasible). Methods to detect infeasibility and to get an initial feasible solution are connected to solving diophantine systems of linear equations; the interested reader is referred to [31] for further details.

The following lemmas help us in describing the geometric structure of a test family for multiobjective integer linear problems.

LEMMA 2.1 (Gordan–Dickson lemma, Theorem 5 in [11]). *If $P \subseteq \mathbb{N}^n$, $P \neq \emptyset$, then there exists a minimal subset $\{p_1, \dots, p_m\} \subseteq P$ that is finite and unique such that $p \in P$ implies $p_j \leq p$ (componentwise) for at least one $j = 1, \dots, m$.*

LEMMA 2.2. *There exists a unique, minimal, finite set of vectors $\alpha_1, \dots, \alpha_k \in \mathbb{N}^n$ such that the set \mathcal{L}_C of all dominated solutions in all fibers of $MIP_{A,C}$ is a subset of \mathbb{N}^n of the form*

$$\mathcal{L}_C = \bigcup_{j=1}^k (\alpha_j + \mathbb{N}^n).$$

Proof. The set of dominated solutions of all problems $MIP_{A,C}$ is

$$\mathcal{L}_C = \{\alpha \in \mathbb{N}^n : \exists \beta \in \mathbb{N}^n, \text{ with } A\beta = A\alpha \text{ and } \beta \prec_C \alpha\}.$$

Let α be an element in \mathcal{L}_C and β a Pareto-optimal point in the fiber $\pi^{-1}(A\alpha)$ that satisfies $\beta \prec_C \alpha$. Then, for any $\gamma \in \mathbb{N}^n$, $A(\alpha + \gamma) = A(\beta + \gamma)$, $\alpha + \gamma, \beta + \gamma \in \mathbb{N}^n$, and $\beta + \gamma \prec_C \alpha + \gamma$, because the cost matrix C has only nonnegative coefficients. Therefore, $\alpha + \gamma$ is a feasible solution dominated by $\beta + \gamma$ in the fiber $\pi^{-1}(A(\alpha + \gamma))$. Then, $\alpha + \gamma \in \mathcal{L}_C$ for all $\gamma \in \mathbb{N}^n$, so $\alpha + \mathbb{N}^n \subseteq \mathcal{L}_C$. By Lemma 2.1, we conclude that there exists a minimal set of elements $\alpha_1, \dots, \alpha_k \in \mathbb{N}^n$ such that $\mathcal{L}_C = \bigcup_{j=1}^k (\alpha_j + \mathbb{N}^n)$. \square

Once elements $\alpha_1, \dots, \alpha_k$ generating \mathcal{L}_C (in the sense of the above result) have been obtained, one can compute the maximal chains of the set $\{\alpha_1, \dots, \alpha_k\}$ with respect to the partial order \prec_C . We denote by $\mathcal{C}_C^1, \dots, \mathcal{C}_C^\mu$ these maximal chains and set $\mathcal{L}_C^i = \bigcup_{t=1}^{k_i} (\alpha_t^i + \mathbb{N}^n)$, where $\alpha_t^i \in \mathcal{C}_C^i$ for $t = 1, \dots, k_i$ and $i = 1, \dots, \mu$. For details about maximal chains, upper bounds on its cardinality and algorithms to compute them for a partially ordered set, the reader is referred to [4].

It is clear that, with this construction, we have $\mathcal{L}_C = \bigcup_{i=1}^\mu \mathcal{L}_C^i$.

Next, we describe a finite family of sets $\mathcal{G}_{\prec_C} \subseteq \text{Ker}(A) \cap \mathbb{Z}^n$ and prove that it is indeed a test family for $MIP_{A,C}$.

Let $\mathcal{G}_{\prec_C} = \{\mathcal{G}_{\prec_C}^i\}_{i=1}^\mu$, being

$$(3) \quad \mathcal{G}_{\prec_C}^i = \{(g_{ij}^k, h_{ij}^k) = (\alpha_j^i - \beta_{ij}^k, \alpha_j^i), j = 1, \dots, k_i, k = 1, \dots, m_{ij}, i = 1, \dots, \mu\}$$

the maximal chains of $\mathcal{G}_{<_C}$ (with respect to the order $<_C$ over the second components) and where $\alpha_1^i, \dots, \alpha_{k_i}^i$ are the unique minimal elements of $\mathcal{L}^i_{<_C}$ and $\beta_{ij}^1, \dots, \beta_{ij}^{m_{ij}}$ are the Pareto-optimal solutions to the problem $MIP_{A,C}(A\alpha_j^i)$.

In the next section, we give an algorithm that explicitly constructs $\mathcal{G}_{<_C}$. Notice that for fixed i, j and k , $g_{ij}^k = (\alpha_j^i - \beta_{ij}^k)$ is a point in the subspace $S = \{x \in \mathbb{Q}^n : Ax = 0\}$, i.e., in the 0-fiber of $MIP_{A,C}$. Geometrically we think of $(\alpha_j^i - \beta_{ij}^k, \alpha_j^i)$ as the oriented vector $\vec{g}_{ij}^k = [\overrightarrow{\beta_{ij}^k}, \alpha_j^i]$ in the $A\alpha_j^i$ -fiber of $MIP_{A,C}$. The vector is directed from the Pareto-optimal point β_{ij}^k to the nonoptimal point α_j^i due to the minimization criterion in $MIP_{A,C}$, which requires us to move away from expensive points. Subtracting the point $\vec{g}_{ij}^k = \alpha_j^i - \beta_{ij}^k$ from the feasible solution γ gives the new solution $\gamma - \alpha_j^i + \beta_{ij}^k$, which is equivalent to translating \vec{g}_{ij}^k by a nonnegative integer vector.

Consider an arbitrary fiber of $MIP_{A,C}$ and a feasible lattice point γ in this fiber. For each vector \vec{g}_{ij}^k in $\mathcal{G}_{<_C}$, check whether $\gamma - g_{ij}^k$ is in \mathbb{N}^n . At γ , draw all such possible translations of vectors from $\mathcal{G}_{<_C}$. The head of the translated vector is also incident at a feasible point in the same fiber as γ , since g_{ij}^k is in the 0-fiber of $MIP_{A,C}$. We do this construction for all feasible points in all fibers of $MIP_{A,C}$. From Lemma 2.2 and the definition of $\mathcal{G}_{<_C}$, it follows that no vector $(\alpha_j^i - \beta_{ij}^k, \alpha_j^i)$ in $\mathcal{G}_{<_C}$ can be translated by a ν in \mathbb{N}^n such that its tail meets a Pareto-optimal solution on a fiber unless the obtained vector is incomparable to the Pareto-optimal point β_{ij}^k .

THEOREM 2.1. *The above construction builds a connected directed graph in every fiber of $MIP_{A,C}$. The nodes of the graph are all the lattice points in the fiber, and (γ, γ') is an edge of the directed graph if $\gamma' = \gamma - g_{ij}^k$ for some i, j , and k . Any directed path of this graph is nonincreasing with respect to the partial order $<_C$.*

Proof. Pick a fiber of $MIP_{A,C}$ and, at each feasible lattice point, construct all possible translations of the vector \vec{g}_{ij}^k from the set $\mathcal{G}_{<_C}$ as described above. Let α be a lattice point in this fiber. By Lemma 2.2, $\alpha = \alpha_j^i + \nu$ for some $i \in \{1, \dots, t\}$ and $\nu \in \mathbb{Z}_+^n$. Now, since the point α'_k defined as $\alpha'_k = \beta_{ij}^k + \nu$ also lies in the same fiber as α , then $\alpha'_k <_C \alpha$ or α'_k and α are incomparable. Therefore, \vec{g}_{ij}^k translated by $\nu \in \mathbb{N}^n$ is an edge of this graph, and we can move along it from α to a point α' in the same fiber such that $\alpha' <_C \alpha$ or α and α' are incomparable. This proves that, from every dominated point in the fiber, we can reach an improved or incomparable point (with respect to $<_C$) in the same fiber by moving along an edge of the graph. \square

We call the graph in the b -fiber of $MIP_{A,C}$ built from elements in $\mathcal{G}_{<_C}$ the $<_C$ -skeleton of that fiber.

The reader may note that, from each dominated solution α , one can easily build paths to its comparable Pareto-optimal solutions subtracting elements in $\mathcal{G}_{<_C}$. Indeed, let β a Pareto-optimal solution in the $A\alpha$ -fiber such that β dominates α . Then, let α_i be a minimal element of \mathcal{L}_C such that $\alpha = \alpha_i + \gamma$, with $\gamma \in \mathbb{N}^n$, and let β_i be the Pareto-optimal solution in the $A\alpha_i$ -fiber that is comparable to α_i and such that $\beta_i + \gamma$ is comparable to β . Then $\alpha' = \beta_i + \gamma$ is a solution in the $A\alpha$ -fiber with $\beta <_C \alpha' <_C \alpha$. Now, one repeats this process but starting with α' and β , until $\alpha' = \beta$. Moreover, the case where α and β are incomparable reduces to the previous one by finding a path from α to any intermediate point β' that compares with β . This analysis leads us to the following result.

COROLLARY 2.1. *In the $<_C$ -skeleton of a fiber, there exists a directed path from every feasible point α to each Pareto-optimal point β in the same fiber. The vectors of objective function values of successive points in the path do not increase componentwise from α to β .*

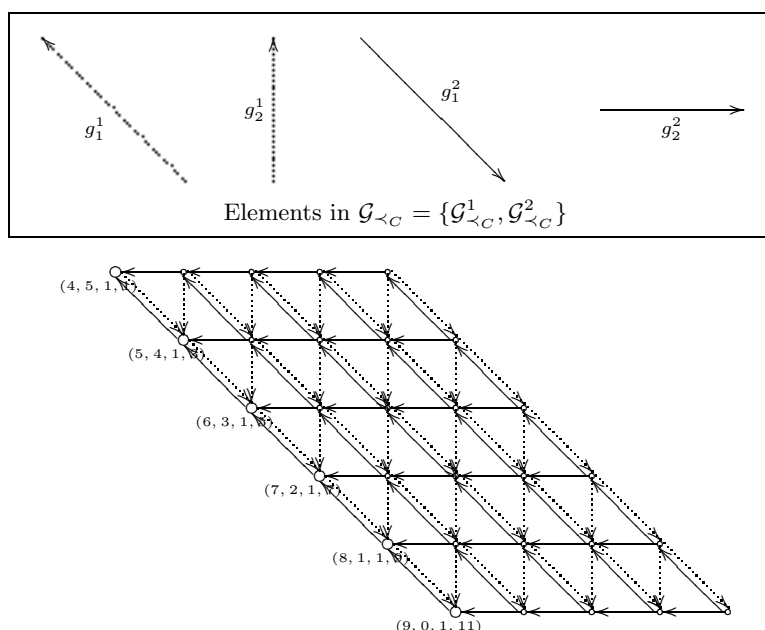


FIG. 2. The \prec_C -skeleton of the $(17, 11)^t$ -fiber of $MIP_{A,C}$ projected on the (x_1, x_2) -plane.

COROLLARY 2.2. The family \mathcal{G}_{\prec_C} is the unique minimal test family for $MIP_{A,C}$. It depends only on the matrix A and the cost matrix C .

Proof. By the definition of \mathcal{G}_{\prec_C} , conditions 1 and 2 of Definition 2.1 are satisfied. From Theorem 2.1, it follows that conditions 3 and 4 are also satisfied, so \mathcal{G}_{\prec_C} is a test family for $MIP_{A,C}$. Minimality is due to the fact that removing any element (g_{ij}^k, h_{ij}^k) from \mathcal{G}_{\prec_C} results in $\mathcal{G}_{\prec_C} \setminus \{(g_{ij}^k, h_{ij}^k)\}$. However, this new set is not a test family, since no oriented vector in $\mathcal{G}_{\prec_C} \setminus \{(g_{ij}^k, h_{ij}^k)\}$ can be translated through a nonnegative vector in \mathbb{N}^n such that its tail meets α_j^i . It is clear by definition that \mathcal{G}_{\prec_C} depends only on A and C . \square

Example 2.2. Let $MIP_{A,C}$ be the family of multiobjective problems, with the following constraints and objective function matrices:

$$A = \begin{bmatrix} 2 & 2 & -1 & 0 \\ 0 & 2 & 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 10 & 1 & 0 & 0 \\ 1 & 10 & 0 & 0 \end{bmatrix}.$$

Let (x_1, x_2, s_1, s_2) be the vector of variables, where s_1 and s_2 are slack variables. In this example, using order \prec_C^s (see Remark 2.2), $\mathcal{G}_{\prec_C} = \{G_{\prec_C}^1, G_{\prec_C}^2\}$, where $G_{\prec_C}^1 = \{\vec{g}_1^1 = ((0, 1, 2, -1), (0, 1, 2, 0)), \vec{g}_2^1 = ((-1, 1, 0, -2), (0, 1, 0, 0))\}$, and $G_{\prec_C}^2 = \{\vec{g}_1^2 = ((1, 0, 2, 0), (1, 0, 2, 0)), \vec{g}_2^2 = ((1, -1, 0, 2), (1, 0, 0, 2))\}$.

Figure 2 shows, on the (x_1, x_2) -plane, the \prec_C -skeleton of the fiber corresponding to the right-hand side vector $(17, 11)^t$. In the box over the graph of the \prec_C -skeleton, we show the second components of the elements of \mathcal{G}_{\prec_C} . The reader may note that, in the graph, the arrows have opposite directions due to the fact that the directed paths (improving solutions) are built subtracting the elements in \mathcal{G}_{\prec_C} . We describe how to compute the sets $G_{\prec_C}^1$ and $G_{\prec_C}^2$ in section 3.

Given \mathcal{G}_{\prec_C} , there are several ways to build a path from each feasible point in a fixed fiber to any Pareto-optimal solution. However, there is a canonical way to do

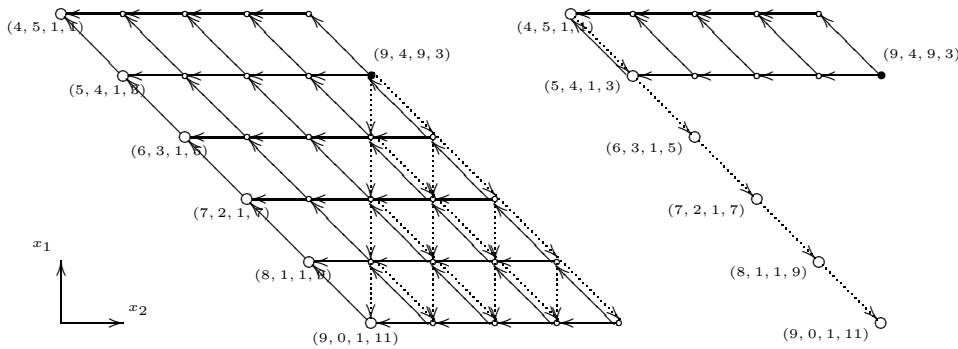


FIG. 3. Different ways to compute paths from $(9, 4, 9, 3)$ to the Pareto-optimal solutions in its fiber.

it: Fix σ a permutation of the set $\{1, \dots, \mu\}$ and subtract from the initial point the elements of $\mathcal{G}_{<_C}^{\sigma(i)}$, for $i = 1, \dots, \mu$. Add this element to an empty list. After each subtraction by elements in $\mathcal{G}_{<_C}^{\sigma(i)}$, $i = 1, \dots, \mu$, remove from the list those elements dominated by the new element. We prove in Section 3 that this result does not depend on the permutation σ .

Example 2.2 (continuation). This example shows the above-mentioned different ways to compute paths from dominated solutions to any Pareto-optimal solution. The vector $(9, 4, 9, 3)$ is a feasible solution for $MIP_{A,C}$ in the $(17, 11)^t$ -fiber. Figure 3 shows the sequence of Pareto-optimal points obtained from the feasible point $(9, 4, 9, 3)$ using the permutation $\sigma_1 = (1, 2)$ (on the left) and using $\sigma_2 = (2, 1)$ (on the right).

Remark 2.3. Let $<_C$ be the partial order induced by C . Then, a directed path from a dominated point α to each Pareto-optimal point β in a fiber, applying the above method, cannot pass through any lattice point in this fiber more than μ times (recall that μ is the number of maximal chains in $\mathcal{G}_{<_C}$). This implies that obtaining the Pareto-optimal solutions of a given $MIP_{A,C}$ using $\mathcal{G}_{<_C}$ cannot cycle.

3. Test families and partial Gröbner bases. In the previous section, we motivated the importance of having a test family for $MIP_{A,C}$, since this structure allows us to obtain the entire set of Pareto-optimal solutions of the above family of multiobjective integer programs (when the right-hand side varies). Our goal in this section is to provide the necessary tools to construct test families for any multiobjective integer problem. Our construction builds upon an extension of Gröbner bases on partial orders.

In order to introduce this structure, we define the reduction of a pair $(g, h) \in \mathbb{Z}^n \times \mathbb{Z}_+^n$ by a finite set of ordered pairs in $\mathbb{Z}^n \times \mathbb{Z}_+^n$. Given is a collection $\mathcal{G}_C \subseteq \mathbb{Z}^n \times \mathbb{Z}_+^n$, where $\mathcal{G}_C = \{(g_1, h_1), \dots, (g_l, h_l) : h_{k+1} <_C h_k, k = 1, \dots, l - 1\}$.

The reduction of (g, h) by \mathcal{G}_C consists of the process described in Algorithm 1. The above reduction process extends to the case of a finite collection of ordered sets of pairs in $\mathbb{Z}^n \times \mathbb{Z}_+^n$ by establishing the sequence in which the sets of pairs are considered. We denote by $pRem((g, h), \mathcal{G})_\sigma$ the reduction of the pair (g, h) by the family $\mathcal{G} = \{\mathcal{G}_i\}_{i=1}^t$ for a fixed sequence of indices σ .

From now on, we denote by $pRem((g, h), \mathcal{G})$ the set of remainders of (g, h) by the family $\mathcal{G} = \{\mathcal{G}_i\}_{i=1}^t$ for the natural sequence of indices $(1, \dots, t)$, i.e., when σ is the identity.

ALGORITHM 1: PARTIAL REDUCTION ALGORITHM.

input : $R = \{(g, h)\}$, $S = \{(g, h)\}$,
 $\mathcal{G}_C = \{(g_1, h_1), \dots, (g_l, h_l) : h_{k+1} \prec_C h_k, k = 1, \dots, l - 1\}$.
 Set $i := 1$, $S_o = \{\}$.
repeat
 for $(\tilde{g}, \tilde{h}) \in S \setminus S_o$ **do**
 while $\tilde{h} - h_i \geq 0$ **do**
 if $\tilde{h} - g_i$ and $\tilde{h} - \tilde{g}$ are comparable by \prec_C **then**
 $R_o = \{(\tilde{g} - g_i, \max_{\prec_C} \{\tilde{h} - g_i, \tilde{h} - \tilde{g}\})\}$
 else
 $R_o = \{(\tilde{g} - g_i, \tilde{h} - g_i), (\tilde{g} - g_i, \tilde{h} - \tilde{g})\}$
 end
 For each $r \in R_o$ and $s \in R$:
 if $r \prec_C s$ **then**
 $R = R \setminus \{s\}$;
 end
 $S = R_o$.
 $R = R \cup R_o$.
 $S_o = S_o \cup \{(\tilde{g}, \tilde{h})\}$.
 end
 end
 $i = i + 1$.
until $i \leq t$;
output: R , the partial reduction set of (g, h) by \mathcal{G}_C .

The reduction of a pair that represents a feasible solution by a test family gives the entire set of Pareto-optimal solutions. In order to obtain that test family, we introduce the notion of p-Gröbner basis. This name has been motivated by the fact that when the ordering in \mathbb{N}^n is induced by a single cost vector, a Gröbner basis is a test set for the family of integer programs $IP_{A,c}$ (see [10] or [41] for extended details). In the single objective case, the Buchberger algorithm computes a Gröbner basis. However, in the multiobjective case, the cost matrix induces a partial order, so division or the Buchberger algorithm are not applicable. Using the above reduction algorithm (Algorithm 1), we present an “à la Buchberger” algorithm to compute the so called p-Gröbner basis to solve MOILP problems.

DEFINITION 3.1 (partial Gröbner basis). *A family $\mathcal{G} = \{\mathcal{G}_1, \dots, \mathcal{G}_t\} \subseteq I_A$ is a partial Gröbner basis (p-Gröbner basis) for the family of problems $MIP_{A,C}$ if $\mathcal{G}_1, \dots, \mathcal{G}_t$ are the maximal chains for the partially ordered set $\bigcup_{i=1}^t \mathcal{G}_i$ and for any $(g, h) \in \mathbb{Z}^n \times \mathbb{Z}_+^n$, with $h - g \geq 0$*

$$g \in \text{Ker}(A) \iff p\text{Rem}((g, h), \mathcal{G})_\sigma = \{0\}$$

for any sequence σ .

A p-Gröbner basis is said to be reduced if every element in each maximal chain cannot be obtained by reducing any other element of the same chain.

Given a p-Gröbner basis, computing a reduced p-Gröbner basis is done by deleting the elements that can be reduced by other elements in the basis. After the removing process, the family is a p-Gröbner basis having only nonredundant elements. It is easy to see that the reduced p-Gröbner basis for $MIP_{A,C}$ is unique and minimal, in the sense that no element can be removed from it, maintaining the p-Gröbner basis structure.

This definition clearly extends to p-Gröbner bases for the ideal I_A induced by A , once we fix the partial order \prec_C induced by C .

In the following, we present algorithms to solve multiobjective problems analogous to the methods that solve the single objective case using usual Gröbner basis. These methods are based on computing the reduction of a feasible solution by the basis. The key for that result is the fact that the reduction of any pair of feasible solutions is the same, therefore, the algorithm is valid for any initial feasible solution. After the following theorem, Lemma 3.1 ensures the same statement for the multiobjective case and p-Gröbner bases.

THEOREM 3.1. *Let \mathcal{G} be the reduced p-Gröbner basis for $MIP_{A,C}$ and α a feasible solution for $MIP_{A,C}(A\alpha)$. Then, $pRem((\alpha, \alpha), \mathcal{G})_\sigma = pRem((\alpha, \alpha), \mathcal{G})_{\sigma'}$ for any sequences σ and σ' .*

Proof. We first observe that the elements in $pRem((\alpha, \alpha), \mathcal{G})_\sigma$ are of the form (β, β) . Indeed, since the first step of Algorithm 1 reduces the element (α, α) , then $\tilde{h} - \tilde{g} = \alpha - \alpha = 0$. Therefore, $\tilde{h} - \tilde{g}$ is always dominated by $\tilde{h} - g_i$ because $0 \prec_C \tilde{h} - g_i$, so that the remainders are of the form $(\alpha - g_i, \alpha - g_i)$.

On other hand, let (β, β) be an element in $pRem((\alpha, \alpha), \mathcal{G})_\sigma$, then $\alpha - \beta \in \text{Ker}(A)$ and by Definition 3.1, $pRem((\alpha - \beta, \alpha), \mathcal{G})_{\sigma'} = pRem((\alpha - \beta, \beta), \mathcal{G})_{\sigma'} = \{0\}$ for any σ' . \square

The above result ensures that without loss of generality reductions of elements of the form (α, α) by p-Gröbner bases are independent of the permutation of indices used. Therefore, we do not make reference to σ in the notation, referring always to the natural sequence $\sigma = (1, \dots, t)$.

LEMMA 3.1. *Let \mathcal{G} be the reduced p-Gröbner basis for $MIP_{A,C}$ and α_1, α_2 two different feasible solutions in the same fiber of $MIP_{A,C}$. Then, $pRem((\alpha_1, \alpha_1), \mathcal{G}) = pRem((\alpha_2, \alpha_2), \mathcal{G})$.*

Proof. Let $(\beta, \beta) \in pRem((\alpha_1, \alpha_1), \mathcal{G})$, then since $A\alpha_1 = A\alpha_2$, β is in the same fiber that α_2 . Next, since β cannot be reduced, then $(\beta, \beta) \in pRem((\alpha_2, \alpha_2), \mathcal{G})$. \square

The following theorem states the relationship between the three structures introduced before: test families, reduced p-Gröbner bases, and the family \mathcal{G}_{\prec_C} .

THEOREM 3.2. *The reduced p-Gröbner basis for $MIP_{A,C}$ is the unique minimal test family for $MIP_{A,C}$. Moreover, \mathcal{G}_{\prec_C} , introduced in (3), is the reduced p-Gröbner basis for $MIP_{A,C}$.*

Proof. Let $\mathcal{G} = \{\mathcal{G}_1, \dots, \mathcal{G}_t\}$ be the reduced p-Gröbner basis for $MIP_{A,C}$. We have to prove that \mathcal{G} satisfies the four conditions in Definition 2.1. By the definition of p-Gröbner basis, it is clear that each \mathcal{G}_i is totally ordered by its second component with respect to \prec_C (condition 1). Condition 2 follows because, for each i and for each $(g, h) \in \mathcal{G}_i \subseteq \mathbb{Z}^n \times \mathbb{Z}_+^n$, clearly $pRem((g, h), \mathcal{G}) = \{0\}$, so $g \in \text{Ker}(A)$ and then $A(h - g) = Ah$.

Now, let $x \in \mathbb{N}^n$ be a dominated solution for $MIP_{A,C}(b)$. Then, there is a Pareto-optimal solution β such that $\beta \prec_C x$. By Lemma 3.1, $pRem((x, x), \mathcal{G}) = pRem((\beta, \beta), \mathcal{G})$ and by construction of the set of partial remainders, $\beta \in pRem((\beta, \beta), \mathcal{G})$, thus $x \notin pRem((x, x), \mathcal{G})$. This implies that there exists $(g, h) \in \mathcal{G}_i$, for some $i = 1, \dots, t$ such that $x - g \prec_C x$. This proves condition 3 of Definition 2.1.

On the other hand, if x is a Pareto-optimal solution for $MIP_{A,C}(b)$, $x \in pRem((x, x), \mathcal{G})$, then there exists no (g, h) in any \mathcal{G}_i such that $x - g \prec_C x$. Therefore, for every i and for each $(g, h) \in \mathcal{G}_i$, either $x - g$ is infeasible or incomparable to x , which proves condition 4 of Definition 2.1.

Minimality is due to the fact that removing an element from the reduced p-Gröbner basis, that is, the minimal partial Gröbner basis that can be built for $MIP_{A,C}$,

we cannot guarantee to have a test family because there may exist a pair $(g, h) \in \mathbb{Z}^n \times \mathbb{Z}_+^n$, with $g \in \text{Ker}(A)$ that cannot be reduced to the zero set.

Finally, the second statement of the theorem follows from Corollary 2.2. \square

Next, we describe an extended algorithm to compute a p-Gröbner basis for I_A , with respect to the partial order induced by C . First, for any $(g, h), (g', h')$ in $\mathbb{Z}^n \times \mathbb{Z}_+^n$, we denote by $S^1((g, h), (g', h'))$ and $S^2((g, h), (g', h'))$ the pairs

$$S^1((g, h), (g', h')) = \begin{cases} (g - g' - 2(h - h'), \gamma + g - 2h) & \text{if } \gamma + g - 2h \prec_C \gamma + g' - 2h', \\ (g' - g - 2(h' - h), \gamma + g' - 2h') & \text{if } \gamma + g' - 2h' \prec_C \gamma + g - 2h, \\ (g - g' - 2(h - h'), \gamma + g - 2h) & \text{if } \gamma + g' - 2h' \text{ and } \gamma + g - 2h, \\ & \text{are incomparable,} \end{cases}$$

and

$$S^2((g, h), (g', h')) = \begin{cases} (g - g' - 2(h - h'), \gamma + g - 2h) & \text{if } \gamma + g - 2h \prec_C \gamma + g' - 2h', \\ (g' - g - 2(h' - h), \gamma + g' - 2h') & \text{if } \gamma + g' - 2h' \prec_C \gamma + g - 2h, \\ (g' - g - 2(h' - h), \gamma + g' - 2h') & \text{if } \gamma + g' - 2h' \text{ and } \gamma + g - 2h, \\ & \text{are incomparable,} \end{cases}$$

where $\gamma \in \mathbb{N}^n$ and $\gamma_i = \max\{h_i, h'_i\}$, $i = 1, \dots, n$.

The pairs $S^1((g, h), (g', h'))$ and $S^2((g, h), (g', h'))$ are called 1-Svector and 2-Svector of (g, h) and (g', h') , respectively. The reader may note that $S^1((g, h), (g', h'))$ and $S^2((g, h), (g', h'))$ coincide, provided that the resulting pairs are comparable under \prec_C , whereas they correspond with the two possible choices of the new pair in the case when the vectors $\gamma + g' - 2h'$ and $\gamma + g - 2h$ are incomparable.

The name is due to the analogy with the algebraic-geometrical notion of S-polynomial for a pair of polynomials with a given term order. Since we consider a partial order, it may happen that in the standard construction of an Svector [41], we cannot decide which is the leading term. Therefore, in our definitions of Sectors, we must consider all possible combinations of leading terms, with respect to the partial order \prec_C .

The original Buchberger criterion was stated in a polynomial language. Therefore, we adapt our notation to follow the line of that proof. Let $\text{leadmon}_C(f)$ denote the set of leading monomials with respect to the order induced by C for any multivariate polynomial $f \in \mathbb{R}[x_1, \dots, x_n]$. We identify the set J_A introduced in (2), with $\mathfrak{S}_A = \langle x^u - x^v : u - v \in \text{Ker}(A) \rangle$, and therefore, the set $\text{setlm}(u, v)$ is identified with the elements in $\text{leadmon}_C(x^u - x^v)$. Moreover, each pair $(g, h) \in \mathbb{Z}^n \times \mathbb{Z}_+^n$, with $g \in \text{Ker}(A)$ and $h - g \geq 0$ is identified with the binomial $x^h - x^{h-g}$. Then, we associate with $\mathcal{G} = \{\mathcal{G}_1, \dots, \mathcal{G}_t\}$ the polynomial set $\mathcal{G}^* = \{\mathcal{G}_1^*, \dots, \mathcal{G}_t^*\}$ identifying one-to-one each pair in \mathcal{G} with the corresponding binomial in \mathcal{G}^* . In this way, we adapt accordingly the definition of $p\text{Rem}((f, p), \mathcal{G}^*)$, the set of partial remainders of $f \in \mathbb{R}[x_1, \dots, x_n]$ with leading monomial p and with respect to \mathcal{G}^* .

Moreover, we define the 1-Spolynomial and 2-Spolynomial as the binomial transcriptions of the 1-Svector and 2-Svector. For any two binomials $x^{\alpha_1} - x^{\beta_1}$ and $x^{\alpha_2} - x^{\beta_2}$, the k -Spolynomial with respect to the leading monomials $x^{\alpha_1}, x^{\alpha_2}$ is

$$\mathbb{S}^k((x^{\alpha_1} - x^{\beta_1}, x^{\alpha_1}), (x^{\alpha_2} - x^{\beta_2}, x^{\alpha_2})) = x^{\gamma - \alpha_2 + \beta_2} - x^{\gamma - \alpha_1 + \beta_1}, \quad k = 1, 2,$$

where $\gamma \in \mathbb{N}^n$ and $\gamma_i = \max\{(\alpha_1)_i, (\alpha_2)_i\}$, $i = 1, \dots, n$. The difference between the 1-Spolynomial and the 2-Spolynomial is the choice of the leading term: They coincide when the monomials are comparable and differ when the monomials are incomparable,

and in this case, each k -Spolynomial corresponds with the two possible choices of the leading term.

The following lemma is used in the proof of our extended criterion, and it is an adaptation of the analogous result for total orders and usual S-polynomials.

LEMMA 3.2. *Let $f_1, \dots, f_s \in \mathbb{R}[x_1, \dots, x_n]$ be such that there exists $p \in \bigcap_{i=1}^s \text{leadmon}_C(f_i)$. Let $f = \sum_{i=1}^s c_i f_i$, with $c_i \in \mathbb{R}$. If there exists $q \in \text{leadmon}_C(f)$ such that $q \prec_C p$, then f is a linear combination with coefficients in \mathbb{R} of the k -Spolynomial, $k = 1, 2$, of f_i and f_j , $1 \leq i < j \leq s$.*

Proof. By hypothesis, $f_i = a_i p + \text{other smaller or incomparable terms}$, with $a_i \in \mathbb{R}$ for all i . Then, f can be rewritten as $f = \sum_{i=1}^s c_i f_i = \sum_{i=1}^s c_i a_i p + \text{other smaller or incomparable terms}$. Since $q \prec_C p$, then $\sum_{i=1}^s c_i a_i = 0$.

By definition, for $k = 1, 2$, $\mathbb{S}^k((f_i, p), (f_j, p)) = \frac{1}{a_i} f_i - \frac{1}{a_j} f_j$, thus,

$$\begin{aligned} f &= c_1 f_1 + \dots + c_s f_s = c_1 a_1 \left(\frac{1}{a_1} f_1 \right) + \dots + c_s a_s \left(\frac{1}{a_s} f_s \right) \\ &= c_1 a_1 \left(\frac{1}{a_1} f_1 - \frac{1}{a_2} f_2 \right) + (c_1 a_1 + c_2 a_2) \left(\frac{1}{a_2} f_2 - \frac{1}{a_3} f_3 \right) + \dots \\ &+ (c_1 a_1 + \dots + c_{s-1} a_{s-1}) \left(\frac{1}{a_{s-1}} f_{s-1} - \frac{1}{a_s} f_s \right) + (c_1 a_1 + \dots + c_s a_s) \frac{1}{a_s} f_s \\ &= d_1^k \mathbb{S}^k((f_1, p), (f_2, p)) + \dots + d_{s-1}^k \mathbb{S}^k((f_{s-1}, p), (f_s, p)) + \left(\frac{1}{a_s} \sum_{i=1}^s c_i a_i \right) f_s \\ &= \sum_{i=1}^{s-1} d_i^k \mathbb{S}^k((f_i, p), (f_{i+1}, p)), \end{aligned}$$

where $d_i^k = \sum_{j=1}^i c_j a_j$ for $i = 1, \dots, s$ and $k = 1, 2$. This proves the lemma. \square

The algorithm to compute standard Gröbner bases is based on the Buchberger criterion. Its analogous for a partial order states that it suffices to check that the partial remainders are zero for Svectors and for any fixed sequence of indices.

THEOREM 3.3 (extended Buchberger’s criterion). *Let $\mathcal{G} = \{\mathcal{G}_1, \dots, \mathcal{G}_t\}$, with $\mathcal{G}_i \subseteq I_A$ for all $i = 1, \dots, t$, be the maximal chains of the partially ordered set $\{g_i : g_i \in \mathcal{G}_i \text{ for some } i = 1, \dots, t\}$ and such that \mathcal{G}^* , the polynomial transcription of \mathcal{G} , is a system of generators of \mathfrak{S}_A . Then the following statements are equivalent:*

- (1) \mathcal{G} is a p -Gröbner basis for the family $\text{MIP}_{A,C}$.
- (2) For each $i, j = 1, \dots, t$ and $(g, h) \in \mathcal{G}_i$, $(g', h') \in \mathcal{G}_j$, $p\text{Rem}(\mathbb{S}^k((g, h), (g', h')), \mathcal{G}) = \{0\}$ for $k = 1, 2$.

Proof. Let \mathcal{G} be a p -Gröbner basis for I_A and $(g, h) \in \mathcal{G}_i$, $(g', h') \in \mathcal{G}_j$ for any $i, j \in \{1, \dots, t\}$. Then, $\mathbb{S}^k((g, h), (g', h'))$, for $k = 1, 2$, is in I_A so by the definition of p -Gröbner basis, $p\text{Rem}(\mathbb{S}^k((g, h), (g', h')), \mathcal{G})_\sigma = \{0\}$, for any sequence σ , in particular, for $\sigma = (1, \dots, t)$.

Conversely, assume that for each $(g, h) \in \mathcal{G}_i$ and $(g', h') \in \mathcal{G}_j$ for any $i, j \in \{1, \dots, t\}$, $p\text{Rem}(\mathbb{S}^k((g, h), (g', h')), \mathcal{G}) = \{0\}$ for $k = 1, 2$. Let $(\tilde{g}, \tilde{h}) \in \mathbb{Z}^n \times \mathbb{Z}_+^n$, with $\tilde{g} \in \text{Ker}(A)$ and $\tilde{h} - \tilde{g} \geq 0$. We define $f = x^{\tilde{h}} - x^{\tilde{h} - \tilde{g}} \in \mathbb{Z}[x_1, \dots, x_n]$, and we denote by $\mathcal{G}^* = \{g_1^*, \dots, g_d^*\}$ the polynomial set associated with \mathcal{G} .

Then, by hypothesis f , can be written as a linear combinations of the elements in \mathcal{G}^* (this representation is not unique):

$$f = \sum_{i=1}^d p_i g_i^*$$

for some $p_i \in \mathbb{R}[x_1, \dots, x_n]$ for $i = 1, \dots, d$.

Let $X = \{X_1, \dots, X_N\}$ be the set of maximal elements of the set $\{P_i R_i : P_i \in \text{leadmon}_C(p_i), R_i \in \text{leadmon}_C(g_i^*)\}$ with respect to \prec_C .

If $X \supseteq \text{leadmon}_C(f)$, the polynomial f can be partially reduced by the elements in \mathcal{G}^* . This proves the result.

Otherwise, there must exist $l \in \text{leadmon}_C(f) \setminus X$. We will prove by contradiction that this case is not possible. Indeed, if $l \in \text{leadmon}_C(f)$, it must come from some simplification (reduction) of the linear combination defining f . Then, the construction ensures that there must exist at least one element $X_i \in X$ such that $l \prec_C X_i$.

Set $J(X_i) = \{j : P_j R_j = X_i, \text{ with } P_j \in \text{leadmon}_C(p_j), R_j \in \text{leadmon}_C(g_j^*)\}$. For any $j \in J(X_i)$, we can write $p_j = P_j + \text{other terms}$ and define $q = \sum_{j \in J(X_i)} P_j g_j^*$. Then, $X_i \in \text{leadmon}_C(P_j g_j^*)$ for all $j \in J(X_i)$. However, by hypothesis, there exists $Q \in \text{leadmon}_C(q)$, with $Q \prec_C X_i$.

Hence, by Lemma 3.2, there exist $d_{s,r}^k \in \mathbb{R}$, $k = 1, 2$ such that

$$q = \sum_{r,s \in J(X_i), r \neq s, g_r^*, g_s^* \in \mathcal{G}^*} d_{s,r}^k \mathbb{S}^k((P_s g_s^*, L_s), (P_r g_r^*, L_s)), \quad k = 1, 2$$

for some $L_j \in \text{leadmon}_C(P_j g_j^*)$ for all $g_j^* \in \mathcal{G}^*$.

Now, for any $r, s \in J(X_i)$, we have that $X_i = \text{lcm}(L_r, L_s)$ for some $L_r \in \text{leadmon}_C(P_r g_r^*)$ and $L_s \in \text{leadmon}_C(P_s g_s^*)$, and therefore, we can write

$$\begin{aligned} \mathbb{S}^k((P_r g_r^*, L_r), (P_s g_s^*, L_s)) &= \frac{X_i}{L_r} P_r g_r^* - \frac{X_i}{L_s} P_s g_s^* \\ &= \frac{X_i}{l_r} g_r^* - \frac{X_i}{l_s} g_s^* = \frac{X_i}{P_{r,s}} \mathbb{S}^k((g_r^*, l_r), (g_s^*, l_s)), \end{aligned}$$

where $l_r = \frac{L_r}{P_r}$, $l_s = \frac{L_s}{P_s}$, $P_{r,s} = \text{lcm}(l_r, l_s)$, and $k = 1, 2$.

By hypothesis, $p\text{Rem}(\mathbb{S}^k((g_r^*, l_r), (g_s^*, l_s)), \mathcal{G}^*) = \{0\}$. Thus, from the last equation we deduce that

$$p\text{Rem}(\mathbb{S}^k((P_r g_r^*, L_r), (P_s g_s^*, L_s)), \mathcal{G}) = \{0\}.$$

This gives a representation:

$$\mathbb{S}^k((P_r g_r^*, L_r), (P_s g_s^*, L_s)) = \sum_{g_\nu^* \in \mathcal{G}^*} p_{r,s}^{k,\nu} g_\nu^*$$

with $p_{r,s}^{k,\nu} \in \mathbb{R}[x_1, \dots, x_n]$ and $k = 1, 2$.

Then, $\{P_{r,s}^{k,\nu} R^\nu : g_\nu^* \in \mathcal{G}^*, P_{r,s}^{k,\nu} \in \text{leadmon}_C(p_{r,s}^{k,\nu}), R^\nu \in \text{leadmon}_C(g_\nu^*)\}$ and do not exist $P_{r,s}^{k,\tilde{\nu}}$ and $R^{\tilde{\nu}}$ satisfying $P_{r,s}^{k,\tilde{\nu}} \in \text{leadmon}_C(p_{r,s}^{k,\tilde{\nu}})$, $R^{\tilde{\nu}} \in \text{leadmon}_C(g_{\tilde{\nu}}^*)$ such that $P_{r,s}^{k,\nu} R^\nu \prec_C P_{r,s}^{k,\tilde{\nu}} R^{\tilde{\nu}} \} = \text{leadmon}_C(\mathbb{S}^k(P_r g_r^*, P_s g_s^*))$.

To simplify the notation, denote $S_{r,s}^k = \text{leadmon}_C(\mathbb{S}^k(P_r g_r^*, P_s g_s^*))$.

By construction of S-polynomials, we have that there exists $p \in S_{r,s}^k$ such that $p \prec_C X_i$, so, substituting these expressions into q above, we have

$$\begin{aligned} f &= \sum_{j \notin J(X_i)} p_j g_j^* + \sum_{j \in J(X_i)} p_j g_j^* = \sum_{j \notin J(X_i)} p_j g_j^* + q \\ &= \sum_{j \notin J(X_i)} p_j g_j^* + \sum_{r,s} d_{r,s}^k \mathbb{S}^k((P_s g_s^*, L_s), (P_r g_r^*, L_r)) = \sum_{j \notin J(X_i)} p_j g_j^* + \sum_{r,s} \sum_{\nu} p_{r,s}^{k,\nu} g_{\nu}^*. \end{aligned}$$

Thus, we have expressed f as

$$f = \sum_{i=1}^d p'_i g_i^*,$$

with one leading term p smaller than X_i . However, this is a contradiction, and the theorem is proved. \square

This criterion (the one in Theorem 3.3) allows us to describe a geometric algorithm which constructs a p-Gröbner basis \mathcal{G}_C for $MIP_{A,C}$, and therefore, a test family for that family of multiobjective problems.

The first approach to compute a p-Gröbner basis for a family of multiobjective programs is based on the Conti and Traverso method for the single objective case [10]. For this algorithm, the key is transforming the given multiobjective program into another one where computations are easier and so that an initial set of generators for I_A is known.

Notice that finding an initial set of generators for I_A can be done by a straightforward modification of the Big-M method [3].

Given the program $MIP_{A,C}(b)$, we consider the associated extended multiobjective program $EMIP_{A,C}(b)$ as the problem $MIP_{\tilde{A},\tilde{C}}(b)$, where

$$\tilde{A} = \left(\begin{array}{c|c} & \begin{array}{c} -1 \\ \vdots \\ -1 \end{array} \\ \hline Id_m & A \end{array} \right) \in \mathbb{Z}^{m \times (m+1+n)},$$

$\tilde{C} = (M \cdot \mathbf{1}|C) \in \mathbb{Z}^{(m+1+n) \times k}$, Id_m stands for the $m \times m$ identity matrix, M is a large constant, and $\mathbf{1}$ is the $(m+1) \times k$ matrix whose components are all 1. This problem adds $m+1$ new variables, whose weights in the multiobjective function are big, and so solving this extended minimization program allows us to solve directly the initial program $MIP_{A,C}$. Indeed, any feasible solution to the original problem is a feasible solution to the extended problem with the first m components equal to zero, so any feasible solution of the form $(0, \overset{m+1}{0}, 0, \alpha_1, \dots, \alpha_n)$ is nondominated, upon the order $\prec_{\tilde{C}}$, by any solution without zeros in the first m components. Then, computing a p-Gröbner basis for the extended program using the partial Buchberger Algorithm (Algorithm 2) allows detecting infeasibility of the original problem. Furthermore, a trivial feasible solution $\tilde{\mathbf{x}}_0 = (b_1, \dots, b_m, 0, \overset{m+1}{0})$ is known, and the initial set of generators for I_A is given by $\{\{M_i - P_i, M_i\} : i = 0 \dots, n\}$, where $M_i = (a_{1i} - \min\{0, \min_j\{a_{ji}\}\}, \dots, a_{mi} - \min\{0, \min_j\{a_{ji}\}\}, -\min\{0, \min_j\{a_{ji}\}\}, 0, \dots, 0)$, $P_i = (0, \overset{m+1}{0}|e_i)$, for all $i = 1, \dots, n$, $M_0 = (1, \overset{m+1}{1}, 0, \dots, 0)$, and $P_0 = \mathbf{0}$, $M_i, P_i, M_0, P_0 \in \mathbb{Z}_+^{n+m+1}$ (see [2] for further details). Then, we can state the following result.

THEOREM 3.4. *Let $\mathcal{G} = \{\mathcal{G}_i\}_{i=1}^t$ be a p-Gröbner basis for $EMIP_{A,C}$ and $\mathbf{b} = (b_1, \dots, b_m)$. The entire set of Pareto-optimal solutions for $MIP_{A,C}(\mathbf{b})$ consists*

ALGORITHM 2: PARTIAL BUCHBERGER ALGORITHM I.

input : $F_1 = \{M_0, M_1, \dots, M_n\}$ and $F_2 = \{P_0, P_1, \dots, P_n\}$,
 $M_i = (a_{1i} - \min\{0, \min_j\{a_{ji}\}\}, \dots, a_{mi} - \min\{0, \min_j\{a_{ji}\}\}, -\min\{0, \min_j\{a_{ji}\}\}, 0, \dots, 0)$ ($i > 0$),
 $P_i = (0, \overset{m+1}{m+1}, 0|e_i) \in \mathbb{N}^{m+n+1}$ ($i > 0$),
 $M_0 = (1, \overset{m+1}{m+1}, 1, 0, \dots, 0)$,
 $P_0 = (0, \overset{n+m+1}{n+m+1}, 0)$.
repeat
 Compute, $\mathcal{G}_1, \dots, \mathcal{G}_t$, the maximal chains for $\mathcal{G} = \phi(\mathcal{F}(F_1, F_2))$.
 for $i, j \in \{1, \dots, t\}$, $i \neq j$, and each pair $(g, h) \in \mathcal{G}_i$, $(g', h') \in \mathcal{G}_j$ **do**
 Compute $R^k = pRem(S^k((g, h), (g', h')), \mathcal{G})$, $k = 1, 2$.
 if $R^k = \{0\}$ **then**
 | Continue with other pair.
 else
 | Add $\phi(\mathcal{F}(r))$ to \mathcal{G} for each $r \in R^k$.
 end
 end
until $R^k = \{0\}$ for every pairs;
output: $\mathcal{G} = \{\mathcal{G}_1, \dots, \mathcal{G}_Q\}$ p-Gröbner basis for I_A with respect to \prec_C .

of the vectors $\alpha = (\alpha_1, \dots, \alpha_n)$ such that $(0, \overset{m+1}{m+1}, 0, \alpha_1, \dots, \alpha_n) \in pRem(((\mathbf{b}, 0, \overset{n+1}{n+1}, 0), (\mathbf{b}, 0, \overset{n+1}{n+1}, 0)), \mathcal{G})$. Moreover, if there is no α' in the set $pRem(((\mathbf{b}, 0, \overset{n+1}{n+1}, 0), (\mathbf{b}, 0, \overset{n+1}{n+1}, 0)), \mathcal{G})$ whose $m+1$ first components are zero MIP $_{A,C}(b)$ is infeasible.

Proof. Let α be a vector obtained by successive reductions over \mathcal{G} . It is clear that α is feasible because $((\mathbf{0}, \alpha), (\mathbf{0}, \alpha))$ is in the set of remainders of $((\mathbf{b}, \mathbf{0}), (\mathbf{b}, \mathbf{0}))$ by \mathcal{G} and then in the same fiber. Besides, α is a Pareto-optimal solution because \mathcal{G} is a test family for the problem (Theorem 3.2).

Now, if β^* is a Pareto-optimal solution, by Lemma 3.1 $pRem(((\mathbf{0}, \beta^*), (\mathbf{0}, \beta^*)), \mathcal{G}) = pRem(((\mathbf{0}, \mathbf{b}), (\mathbf{0}, \mathbf{b})), \mathcal{G})$, but since β^* is a Pareto-optimal solution, it cannot be reduced so $((\mathbf{0}, \beta^*), (\mathbf{0}, \beta^*)) \in pRem(((\mathbf{0}, \beta^*), (\mathbf{0}, \beta^*)), \mathcal{G})$ and then $((\mathbf{0}, \beta^*), (\mathbf{0}, \beta^*))$ also belongs to the list of partial remainders of $((\mathbf{b}, \mathbf{0}), (\mathbf{b}, \mathbf{0}))$ by \mathcal{G} . \square

Hoşten and Sturmfels [22] improved the method by Conti and Traverso to solve single-objective programs using standard Gröbner bases. Their improvement is due to the fact that it is not necessary to increase the number of variables in the problem, as Conti and Traverso's algorithm does. Hoşten and Sturmfels's algorithm allows decreasing the number of steps in the computation of the Gröbner basis, but, on the other hand, it needs an algorithm to compute an initial feasible solution, which was trivial in the Conti and Traverso algorithm. We have modified this alternative algorithm to compute the entire set of Pareto-optimal solutions. The first step in the algorithm is computing an initial basis for the polynomial toric ideal $\mathfrak{S}_A = \langle x^u - x^v : u - v \in \text{Ker}(A) \rangle$ that we can identify with J_A . This step does not depend on the order induced by the objective function, so it can be used to solve multiobjective problems. Details can be seen in [22]. Algorithm 3 implements the computation of the set of generators of \mathfrak{S}_A . This procedure uses the notion of Lenstra–Lenstra–Lovász (LLL)-reduced basis (see [27] for further details). In addition, we use a ω -graded reverse lexicographic term order $\prec_{\omega}^{gr_i}$ induced by $x_{i+1} > \dots > x_{i-1} > x_i$ (with $x_{n+1} := x_1$) that is defined as follows:

$$\alpha \prec_{\omega}^{gr_i} \beta \iff \sum_{j=1}^n \omega_j \alpha_j < \sum_{j=1}^n \omega_j \beta_j \text{ or } \sum_{j=1}^n \omega_j \alpha_j = \sum_{j=1}^n \omega_j \beta_j \text{ and } \alpha \prec_{lex} \beta,$$

ALGORITHM 3: `setofgenerators(A)`.

input : $A \in \mathbb{Z}^{m \times n}$

1. Find a lattice basis \mathcal{B} for $\text{Ker}(A)$ (using Hermite normal form).
2. Replace \mathcal{B} by the LLL-reduced lattice basis \mathcal{B}_{red} .
 Let $J_0 := \langle x^{u_+} - x^{u_-} : u \in \mathcal{B}_{red} \rangle$.

for $i = 1, \dots, n$ **do**

Compute $J_i = (J_{i-1} : x_i^\infty)$ as

- (a) Compute \mathcal{G}_{i-1} the reduced Gröbner basis for J_{i-1} with respect to $\prec_\omega^{gr_i}$.
- (b) Divide each element $f \in \mathcal{G}_{i-1}$ by the highest power of x_i that divides f .

output: $\mathfrak{S}_A := J_n = \{x^{u_1} - x^{v_1}, \dots, x^{u_s} - x^{v_s}\}$ system of generators for I_A .

ALGORITHM 4: `pgrobner(F1, F2)`.

input : $F_1 = \{M_1, \dots, M_s\}$ and $F_2 = \{P_1, \dots, P_s\}$.

repeat

Compute $\mathcal{G}_1, \dots, \mathcal{G}_t$ the maximal chains for $\mathcal{G} = \phi(\mathcal{F}(F_1, F_2))$.

for $i, j \in \{1, \dots, t\}$, $i \neq j$, and each pair $(g, h) \in \mathcal{G}_i$, $(g', h') \in \mathcal{G}_j$ **do**

Compute $R^k = pRem(S^k((g, h), (g', h')), \mathcal{G})$, $k = 1, 2$.

if $R^k = \{0\}$ **then**

Continue with other pair.

else

Add $\phi(\mathcal{F}(r))$ to \mathcal{G} for each $r \in R^k$.

end

end

until $R^k = \{0\}$ for every pairs;

output: $\mathcal{G} = \{\mathcal{G}_1, \dots, \mathcal{G}_Q\}$ p-Gröbner basis for $MIP_{A,C}$.

where $\omega \in \mathbb{R}_+^n$ is chosen such that $x_{i+1} > \dots > x_{i-1} > x_i$. Finally, for any $a \in \mathbb{R}$, we denote by $a_+ = \max\{a, 0\}$ and $a_- = -\min\{a, 0\}$.

\mathfrak{S}_A consists of binomials $x^{u_i} - x^{v_i}$ with $u_i - v_i \in \text{Ker}(A)$ for $i = 1, \dots, s$. Coming back to our notation, each binomial $x^u - x^v$ in \mathfrak{S}_A is identified with $\{u, v\} \in J_A$, so computing a set of generators for \mathfrak{S}_A gives us, in some sense, a finite number of generators for the set that represents the constraints matrix. We compute in the next step a partial Gröbner basis from initial sets $F_1 = \{u_1, \dots, u_s\}$ and $F_2 = \{v_1, \dots, v_s\}$ using our extended Buchberger algorithm (Algorithm 4).

Once we have obtained the partial Gröbner basis using the above algorithm, we can compute the entire set of Pareto-optimal solutions for $MIP_{A,C}(b)$ by Algorithm 5.

There are some interesting cases where our methodology is highly simplified due to the structure of the set of constraints. One of these cases is when the dimension of the set of constraints is $n - 1$. The next remark explains how the algorithm simplifies in this case.

Remark 3.1. Let A be an $m \times n$ integer matrix with rank $n - 1$. Then, since $\dim(\text{Ker}(A)) = 1$, the system of generators for I_A (Step 2) has just one element, (g, h) and the p-Gröbner basis (Step 3) is the family $\mathcal{G} = \{(g, h)\}$ because no Svector appears during the computation of the Buchberger algorithm. In this case, Pareto-optimal solutions are obtained as partial remainders of an initial feasible solution (α, α) by (g, h) , i.e., the entire set of Pareto-optimal solutions is a subset of $\Gamma = \{\alpha - \lambda g : \lambda \in \mathbb{Z}_+\}$. More explicitly, the set of Pareto-optimal solutions for $MIP_{A,C}(b)$ is the set of minimal elements (with respect to \prec_C) of Γ .

ALGORITHM 5: PARETO-OPTIMAL SOLUTIONS COMPUTATION FOR $MIP_{A,C}(b)$.

input : $MIP_{A,C}(b)$.

Step 1. Compute an initial feasible solution α_o for $MIP_{A,C}(b)$ (a solution for the diophantine system of equations $Ax = b$, $x \in \mathbb{Z}^n$).

Step 2. Compute a system of generators for I_A : $\{\{u_i, v_i\} : i = 1, \dots, s\}$ using **setofgenerators**(A).

Step 3. Compute the partial reduced Gröbner basis for $MIP_{A,C}$, $\mathcal{G}_C = \{\mathcal{G}_1, \dots, \mathcal{G}_t\}$ using **pgrobner**(F_1, F_2), where $F_1 = \{u_i : i = 1, \dots, r\}$ and $F_2 = \{v_i : i = 1, \dots, r\}$.

Step 4. Calculate the set of partial remainders: $R := pRem(\alpha_o, \mathcal{G}_C)$.

output: Pareto-optimal Solutions : R .

To illustrate the above approach, we present an example of MOILP with two objectives where all the computations are done in detail.

Example 3.1.

$$(4) \quad \begin{aligned} \min \quad & \{10x + y, x + 10y\} \\ \text{s.t.} \quad & \\ & 2x + 2y \geq 17, \\ & 2y \leq 11, \\ & x \leq 10, \\ & x, y \in \mathbb{Z}_+. \end{aligned}$$

Transforming the problem to the standard form results in

$$(5) \quad \begin{aligned} \min \quad & \{10x + y + 0z + 0t + 0q, x + 10y + 0z + 0t + 0q\} \\ \text{s.t.} \quad & \\ & 2x + 2y - z = 17, \\ & 2y + t = 11, \\ & x + q = 10, \\ & x, y, z, t, q \in \mathbb{Z}_+. \end{aligned}$$

Step 1. Feasible solution for $MIP_{A,C}(b)$: $u = (9, 4, 9, 3, 1)$.

Step 2. Following the steps of Algorithm 3:

1. Basis for $\text{Ker}(A)$: $\mathcal{B} := \{(0, 1, 2, -2, 0), (-1, 0, -2, 0, 1)\}$.
2. LLL-reduced basis for \mathcal{B} : $\mathcal{B}_{red} := \mathcal{B} := \{(-1, 0, -2, 0, 1), (-1, 1, 0, -2, 1)\}$.
3. $J_0 := \langle x^{u^+} - x^{u^-} : u \in \mathcal{B}_{red} \rangle = \langle x_5 - x_1x_3^2, x_2x_5 - x_1x_4^2 \rangle$.
4. $J_{i+1} := (J_i : x_i^\infty)$.
 - (a) $\tilde{\mathcal{G}}_0 := \{x_5 - x_1x_3^2, x_2x_5 - x_1x_4^2, x_2x_3^2 - x_4^2\} \Rightarrow J_1 := \langle x_5 - x_1x_3^2, x_2x_5 - x_1x_4^2, x_2x_3^2 - x_4^2 \rangle$.
 - (b) $\tilde{\mathcal{G}}_1 := \{x_5 - x_1x_3^2, x_2x_5 - x_1x_4^2, x_2x_3^2 - x_4^2\} \Rightarrow J_2 := \langle x_5 - x_1x_3^2, x_2x_5 - x_1x_4^2, x_2x_3^2 - x_4^2 \rangle$.
 - (c) $\tilde{\mathcal{G}}_2 := \{x_5 - x_1x_3^2, x_2x_5 - x_1x_4^2, x_2x_3^2 - x_4^2\} \Rightarrow J_3 := \langle x_5 - x_1x_3^2, x_2x_5 - x_1x_4^2, x_2x_3^2 - x_4^2 \rangle$.
 - (d) $\tilde{\mathcal{G}}_3 := \{x_5 - x_1x_3^2, x_2x_5 - x_1x_4^2, x_2x_3^2 - x_4^2\} \Rightarrow J_4 := \langle x_5 - x_1x_3^2, x_2x_5 - x_1x_4^2, x_2x_3^2 - x_4^2 \rangle$.
5. $\mathfrak{S}_A = \langle x_5 - x_1x_3^2, x_2x_5 - x_1x_4^2, x_2x_3^2 - x_4^2, x_1x_3^2 - 1 \rangle \mapsto$
 $I_A = \langle \{(1, 0, 0, 0, 1), (0, 1, 0, 2, 0)\}, \{(1, 0, 2, 0, 0), (0, 0, 0, 0, 1)\},$
 $\{(0, 1, 2, 0, 0), (0, 0, 0, 2, 0)\} \rangle$.

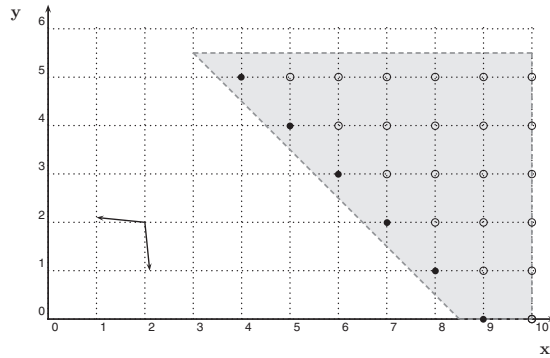


FIG. 4. Feasible region, Pareto-optimal solutions, and improvement cone for Example 3.1.

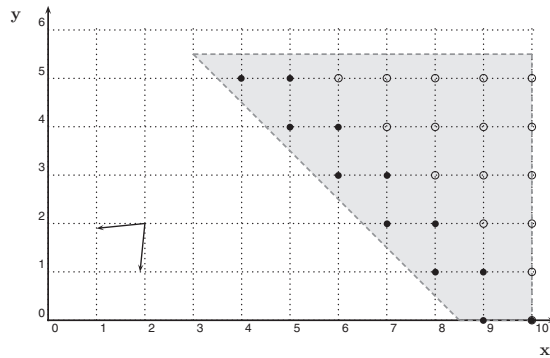


FIG. 5. Feasible region, Pareto-optimal solutions, and improvement cone for Example 3.1 with $C = \begin{bmatrix} 10 & -1 \\ -1 & 10 \end{bmatrix}$.

Step 3. Computing a p-Gröbner basis for I_A , using order \prec_C^s (Remark 2.2) and following Algorithm 4, we obtain \mathcal{G} , whose maximal chains are $\mathcal{G}_1 = \{((0, 1, 2, 0, 0), (0, 0, 0, 2, 0), (0, 1, 2, 0, 0)), ((0, 1, 0, 0, 2), (2, 0, 2, 2, 0), (0, 1, 0, 0, 2)), ((0, 1, 0, 0, 1), (1, 0, 0, 2, 0), (0, 1, 0, 0, 1))\}$ and $\mathcal{G}_2 = \{((1, 0, 0, 4, 0), (0, 2, 2, 0, 1), (1, 0, 0, 4, 0)), ((1, 0, 2, 0, 0), (0, 0, 0, 0, 1), (1, 0, 2, 0, 0)), ((1, 0, 0, 2, 0), (0, 1, 0, 0, 1), (1, 0, 0, 2, 0))\}$.

Step 4. Partial remainders. Reducing first by \mathcal{G}_1 ,
 $pRem((9, 4, 9, 3, 1), \mathcal{G}_1) = \{(9, 0, 1, 11, 1)\}$.
 Then, reducing each remainder by \mathcal{G}_2 ,
 $pRem((9, 0, 1, 11, 1), \mathcal{G}_2) = \{(9, 0, 1, 11, 1), (8, 2, 3, 7, 2), (7, 2, 1, 9, 3), (6, 3, 1, 5, 4), (5, 4, 1, 3, 5), (4, 5, 1, 1, 6)\}$.
 The entire set of Pareto-optimal solutions is
 $\{(9, 0, 1, 11, 1), (8, 1, 1, 9, 2), (7, 2, 1, 7, 3), (6, 3, 1, 5, 4), (5, 4, 1, 3, 5), (4, 5, 1, 1, 6)\}$.

Figure 4 shows the feasible region and the Pareto-optimal solutions of the example above. In addition, we have evaluated the problem with the same feasible region but choosing a cost matrix such that the respective normal vectors of each of the rows in the matrix form an acute angle. Then, nonsupported solutions appear in the set of Pareto-optimal solutions. Figure 5 shows the Pareto-optimal solutions for the same feasible region and $C = \begin{bmatrix} 10 & -1 \\ -1 & 10 \end{bmatrix}$.

TABLE 1
Summary of computational experiments for knapsack problems.

problem	sogt	pgbt	post	tott	pos	maxch	steps	act_pGB
knap4_2	0.063	249.369	1.265	250.697	11	20	2	164.920
knap4_3	0.063	1002.689	2.012	1004.704	5	46	2	772.772
knap4_4	0.063	1148.574	2.374	1151.011	16	98	2.4	763.686
knap5_2	0.125	1608.892	0.875	609.892	3	29	2	1187.201
knap5_3	0.125	3500.831	2.035	3503.963	2	30	2.2	2204.123
knap5_4	0.125	3956.534	2.114	3958.773	9	45.4	3	3044.157
knap6_2	0.185	2780.856	2.124	2783.165	18	156	2.4	2241.091
knap6_3	0.185	3869.156	2.018	3871.359	16.4	189	2.4	2790.822
knap6_4	0.185	4598.258	3.006	4601.449	26	298	3.2	3096.466

4. Computational results. A series of computational experiments have been performed in order to evaluate the behavior of the proposed solution method. Programs have been coded in MAPLE 10 and executed in a PC with an Intel Pentium 4 processor at 2.66GHz and 1 GB of RAM. In the implementation of Algorithm 4 to obtain the p-Gröbner basis, the package *poset* for Maple [35] has been used to compute, at each iteration, the maximal chains for the p-Gröbner basis. The implementation has been done in a symbolic programming language, available upon request, in order to make the access easy to both optimizers and algebraic geometers.

The performance of the algorithm was tested on randomly generated instances for knapsack and transportation [29] multiobjective problems for 2, 3, and 4 objectives. For the knapsack problems, 4, 5, and 6 variables, programs have been considered, and, for each group, the coefficients of the constraint were randomly generated in $[0, 20]$, whereas the coefficients of the objective matrices range in $[0, 20]$. Once the constraint vector (a_1, \dots, a_n) is generated, the right-hand side is fixed as $b = \lceil \frac{1}{2} \sum_{i=1}^n a_i \rceil$ to ensure feasibility.

The computational tests for each number of variables have been done in the following way: (1) Generate five constraint vectors and compute the initial system of generators for each of them using Algorithm 3; (2) Generate five random objective matrices for each number of objectives (2, 3, and 4) and compute the corresponding p-Gröbner basis using Algorithm 4; and (3) with $b = \lceil \frac{1}{2} \sum_{i=1}^n a_i \rceil$ and for each objective matrix, compute the Pareto-optimal solutions using Algorithm 5.

Table 1 contains a summary of the average results obtained for the considered knapsack multiobjective problems. The second, third, and fourth columns show the average CPU times for each stage in the algorithm: *sogt* is the CPU time for computing the system of generators, *pgbt* is the CPU time for computing a p-Gröbner basis, and *post* is the time for computing a feasible solution and partially reducing it to obtain the set of Pareto-optimal solutions. The fifth column shows the total time for computing the set of Pareto-optimal solutions for the problem. Finally, the sixth and seventh columns show the average number of Pareto-optimal solutions and the number of maximal chains in the p-Gröbner basis for the problem, respectively. The problems have been named as *knapN_0*, where N is the number of variables and 0 is the number of objectives. For the transportation problems, instances with 3 origins \times 2 destinations, 3 origins \times 3 destinations, and 4 origins \times 2 destinations have been considered. In this case, for each fixed numbers of origins s and destinations d , the constraint matrix $A \in \mathcal{Z}^{(s+d) \times (sd)}$ is fixed. Then, we have generated five instances for each problem of size $s \times d$. Each of these instances is combined with five different right-hand side vectors. The procedure is analogous to the knapsack computational

TABLE 2

Summary of computational experiments for the battery of multiobjective transportation problems.

problem	sogt	pgbt	post	tott	pos	maxch	steps	act_pGB
tr3x2_2	0.015	11.813	0.000	11.828	5.2	6	2	7.547
tr3x2_3	0.015	7.218	13.108	30.341	12	2.6	2	6.207
tr3x2_4	0.015	6.708	15.791	21.931	6	5	2.2	4.561
tr3x3_2	0.047	1545.916	1.718	1547.681	5	92	2	928.222
tr3x3_3	0.047	3194.333	11.235	3205.615	9	122	2.4	2172.146
tr3x3_4	0.047	3724.657	7.823	3732.527	24	187.4	2.2	2112.287
tr4x2_2	0.046	675.138	2.122	677.306	3.4	35.2	2	398.093
tr4x2_3	0.046	1499.294	6.288	1505.628	5.8	42.4	2.2	119.519
tr4x2_4	0.046	2285.365	7.025	2292.436	12	59	2.2	1654.048

test: a first step where a system of generators is computed, a second one where the p-Gröbner basis is built, and in the last step, the set of Pareto-optimal solutions is computed using partial reductions. Table 2 shows the average CPU times and the average number of Pareto-optimal solutions and maximal chains in the p-Gröbner basis for each problem. The **steps** column shows the average number of steps in the p-Gröbner computation, and **act_pGB** is the average CPU time in the computation of the p-Gröbner basis elapsed since the last element was added to the basis until the end of the process. The problems have been named as $\text{tr}N \times M_0$, where N is the number of origins, M is the number of destinations, and 0 is the number of objectives. As can be seen in Tables 1 and 2, the overall CPU times are clearly divided into three steps, the most costly being the computation of the p-Gröbner basis. In all of the cases, more than 99% of the total time is spent computing the p-Gröbner basis. Once this structure is computed, obtaining the Pareto-optimal solutions is done very efficiently.

The CPU times and sizes in the different steps of the algorithm are highly sensitive to the number of variables. However, our algorithm is not very sensitive to the number of objectives, since the increment of CPU times with respect to the number of objectives is much smaller than the one with respect to the number of variables.

It is clear that one can not expect fast algorithms for solving MOILP, since all these problems are NP-hard. Nevertheless, our approach provides exact tools that, apart from solving these problems, give insights into the geometric and algebraic nature of the problem.

As mentioned above, using our methodology one can identify the common algebraic structure within any MOILP problem. This connection allows us to improve the efficiency of our algorithm, making use of any advance that improves the computation of Gröbner bases. In fact, any improvements of the standard Gröbner bases theory may have an impact in improving the performance of this algorithm. In particular, one can expect improvements in the efficiency of our algorithm based on the special structure of the integer program (see, for instance, Remark 3.1). In addition, we have to mention another important issue in our methodology. As shown in Theorem 3.2, solving MOILP with the same constraint and objective matrices requires computing only once the p-Gröbner basis. Therefore, once this is done, we can solve different instances varying the right-hand side very quickly.

Finally, we have observed from our computational tests that a significant amount of the time, more than 60% (see column **act_pGB**) for the computation of the p-Gröbner basis is spent checking that no new elements are needed in this structure. This implies that the actual p-Gröbner basis is obtained much earlier than when the final test is finished. A different truncation strategy may be based on the number of steps required to obtain the p-Gröbner basis. According to the exact method, the

algorithm stops once in a step; no new elements are added to the structure. Our tables show that, in most cases, the number of steps is two, actually, only one step is required to generate the entire p -Gröbner basis (see column `steps`). These facts can be used to accelerate the computational times at the price of obtaining only heuristic Pareto-optimal solutions. This idea may be considered an alternative primal heuristic in MOILP and will be the subject of further research.

REFERENCES

- [1] K. AARDAL, R. WEISMANTEL, AND L. WOLSEY, *Non-standard approaches to integer programming*, Discrete Appl. Math., 123 (2002), pp. 5–74.
- [2] W. ADAMS AND P. LOUSTAUNAU, *An Introduction to Gröbner Bases*, Grad. Stud. Math. 3, AMS, Providence, RI, 1994.
- [3] M.A. BAZARAA, H.D. SHERALI, AND C.M. SHETTY, *Nonlinear Programming: Theory and Algorithms*, John Wiley and Sons, New York, 1993.
- [4] R.M. BAER AND O. ØSTERBY, *Algorithms over partially ordered sets*, J. BIT Numer. Math., 9 (1969), pp. 97–118.
- [5] D. BERTSIMAS AND R. WEISMANTEL, *Optimization Over Integers*, Dynamic Ideas, Belmont, MA, 2005.
- [6] B. BUCHBERGER, *An Algorithm for finding the basis elements of the residue class ring of a zero-dimensional polynomial ideal*, J. Symb. Comp., 4 (2005), pp. 475–511.
- [7] G. CANTOR, *Beiträge zur Begründung der transfiniten Mengenlehre (Zweiter Artikel)*, Math. Ann., 49 (1897), pp. 207–246.
- [8] A. CAYLEY, *A theorem on trees*, Q. J. Math., 23 (1889), pp. 376–378.
- [9] V. CHANKONG AND Y.Y. HAIMES, *Multiobjective Decision Making Theory and Methodology*. Elsevier Science, New York, 1983.
- [10] P. CONTI AND C. TRAVERSO, *Buchberger algorithm and integer programming*, in Proceedings of the AAEECC-9, New Orleans, Lect. Notes Comput. Sci. 539, H. F. Mattson, T. Mora, and T. R. N. Rao, eds., Springer, New York, 1991, pp. 130–139.
- [11] D. COX, J. LITTLE, AND D. O’SHEA, *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*, 1st ed., Springer, New York, 1992.
- [12] D. COX, J. LITTLE, AND D. O’SHEA, *Using Algebraic Geometry*, 1st ed., Springer, New York, 1998.
- [13] X. DELORME, X. GANDIBLEUX, AND F. DEGOUTIN, *Resolution approché du probleme de set packing bi-objectifs*, in Proceedings of de L’ecole d’Automne de Recherche Operationnelle de Tours (EARO), 2003, pp. 74–80.
- [14] F.Y. EDGEWORTH, *Mathematical Psychics*, P. Keagan, London, 1881.
- [15] M. EHRGOTT, *Multicriteria Optimization*, Lecture Notes in Econom. Math. Systems 491, Springer, Berlin, 2000.
- [16] M. EHRGOTT AND X. GANDIBLEUX, *A survey and annotated bibliography of multicriteria combinatorial optimization*, OR Spectrum, 22 (2000), pp. 425–460.
- [17] M. EHRGOTT AND X. GANDIBLEUX, EDS., *Multiple Criteria Optimization. State of the Art Annotated Bibliographic Surveys*, Kluwer, Boston, 2002.
- [18] M. EHRGOTT, J. FIGUEIRA, AND S. GRECO, EDS., *Multiple Criteria Decision Analysis. State of the Art Surveys*, Springer, New York, 2005.
- [19] M. EHRGOTT, J. FIGUEIRA, AND X. GANDIBLEUX, EDS., *Multiobjective discrete and combinatorial optimization*, Ann. Oper. Res., 147 (2006), pp. 1–3.
- [20] H. HAMACHER AND G. RUHE, *On spanning tree problems with multiple objectives*, Ann. Oper. Res., 52 (1994), pp. 209–230.
- [21] F. HAUSDORFF, *Untersuchungen über Ordnungstypen, Berichte über die Verhandlungen der königlich sächsischen Gesellschaft der Wissenschaften zu Leipzig, Mathematisch - Physische Klasse*, 58 (1906), pp. 106–169.
- [22] S. HOŞTEN AND B. STURMFELS, *GRIN: An implementation of Gröbner bases for integer programming*, in Proceedings of the 4th International IPCO Conference, Integer Programming and Combinatorial Optimization, Lect. Notes Comput. Sci. 920, E. Balas and J. Clausen, eds., Springer, Berlin, 1995, pp. 267–276.
- [23] S. HOŞTEN, *Degrees of Gröbner bases of integer programs*, Ph.D. thesis, Cornell University, Ithaca, NY, 1997.

- [24] H. ISHIBUCHI AND T. MURATA, *A multi-objective genetic local search algorithm and its application to flowshop scheduling*, IEEE Trans. Syst. Man Cybern. C, 28 (1998), pp. 392–403.
- [25] N. JOZEFOWIEZ, F. SEMET, AND E-G. TALBI, *A multi-objective evolutionary algorithm for the covering tour problem*, Applications of Multi-Objective Evolutionary Algorithms, C. A. Coello and G. B. Lamont, eds., World Scientific, River Edge, NJ, 2004, pp. 247–267.
- [26] M.H. KARWAN AND B. VILLARREAL, *Multicriteria dynamic programming with an application to the integer case*, J. Optim. Theory Appl., 31 (1982), pp. 43–69.
- [27] A.K. LENSTRA, H.W. LENSTRA, AND L. LOVÁSZ, *Factoring polynomials with rational coefficients*, Math. Ann., 261 (1982), pp. 515–534.
- [28] K. MIETTINEN, *Nonlinear Multiobjective Optimization*, Kluwer, Boston, 1999.
- [29] G.L. NEMHAUSER AND L.A. WOLSEY, *Integer and Combinatorial Optimization*, John Wiley and Sons, New York, 1988.
- [30] V. PARETO, *Manual d'Economie Politique*, F. Rouge, Lausanne, 1896.
- [31] L. POTTIER, *Minimal solutions of linear diophantine systems: Bounds and algorithms*, in Proceedings of the Fourth International Conference on Rewriting Techniques and Applications, 1991, Como, Italy, pp. 162–173.
- [32] Y. SAWARAGI, H. NAKAYAMA, AND T. TANINO, *Theory of Multiobjective Optimization*, Academic Press, New York, 1985.
- [33] A. SCHRIJVER, *Combinatorial Optimization: Polyhedra and Efficiency*, Springer, New York, 2003.
- [34] N. EL-SHERBENY, *Resolution of a Vehicle Routing Problem with Multiobjective Simulated Annealing Method*, Ph.D. thesis, Faculte Polytechnique de Mons, Mons, Belgium, 2001.
- [35] J.R. STEMBRIDGE, *A Maple Package for Posets*, <http://www.math.lsa.umich.edu/~jrs>.
- [36] R.E. STEUER, *Multiple Criteria Optimization: Theory, Computation and Application*, John Wiley and Sons, New York, 1985.
- [37] B. STURMFELS, *Gröbner Bases and Convex Polytopes*, AMS, Univ. Lectures Ser. 8, Providence, RI, 1996.
- [38] B. STURMFELS, *Solving Systems of Polynomial Equations*, CBMS Reg. Conf. Ser. 97, AMS, Providence, RI, 2002.
- [39] B. STURMFELS, *Algebraic Recipes for Integer Programming*, Proc. Sympos. Appl. Math., 61, AMS, Providence, RI, 2004.
- [40] B. STURMFELS AND R.R. THOMAS, *Variation of cost functions in integer programming*, Math. Program., 77 (1997), pp. 357–387.
- [41] R.R. THOMAS, *A geometric Buchberger algorithm for integer programming*, Math. Oper. Res., 20 (1995), pp. 864–884.
- [42] R.R. THOMAS, *Applications to integer programming*, in Applications of Computational Algebraic Geometry, D. A. Cox and B. Sturmfels, eds., Proceedings of the 53rd Symposium in Applied Mathematics, AMS, 1997, pp. 119–142.
- [43] R.R. THOMAS AND R. WEISMANTEL, *Truncated Groebner bases for integer programming*, Appl. Algebra Engrg., Comm. Comput., 8 (1997), pp. 241–257.
- [44] R. URBANIAK, R. WEISMANTEL, AND G.M. ZIEGLER, *A variant of the Buchberger algorithm for integer programming*, SIAM J. Discrete Math., 10 (1997), pp. 96–108.
- [45] P.L. YU, *Cone convexity, cone extreme points and nondominated solutions in decision problems with multiobjectives*, J. Optim. Theory Appl., 14 (1974), pp. 319–377.
- [46] S. ZIONTS, *A survey of multiple criteria integer programming methods*, Ann. Discrete Math., 5 (1979), pp. 389–398.
- [47] S. ZIONTS AND J. WALLENIUS, *Identifying efficient vectors: Some theory and computational results*, Oper. Res., 23 (1980), pp. 785–793.