Journal of Symbolic Computation 46 (2011) 511–533

Contents lists available at ScienceDirect

Journal of Symbolic Computation

journal homepage: www.elsevier.com/locate/jsc

Some algebraic methods for solving multiobjective polynomial integer programs[☆]

Víctor Blanco[a], Justo Puerto[b]

[a] Departamento de Álgebra, Facultad de Ciencias, Universidad de Granada, 18071 Granada, Spain

[b] Departamento de Estadística e Investigación Operativa, Facultad de Matemáticas, Universidad de Sevilla, 41012 Sevilla, Spain

ARTICLE INFO

Article history:
Received 27 November 2008
Accepted 25 September 2010
Available online 13 October 2010

Keywords:
Multiple objective nonlinear optimization
Integer programming
Gröbner bases

ABSTRACT

Multiobjective discrete programming is a well-known family of optimization problems with a large spectrum of applications. The linear case has been tackled by many authors during the past few years. However, the polynomial case has not been studied in detail due to its theoretical and computational difficulties. This paper presents an algebraic approach for solving these problems. We propose a methodology based on transforming the polynomial optimization problem to the problem of solving one or more systems of polynomial equations and we use certain Gröbner bases to solve these systems. Different transformations give different methodologies that are theoretically stated and compared by some computational tests via the algorithms that they induce.

1. Introduction

A multiobjective polynomial program consists of a finite set of polynomial objective functions and a finite set of polynomial constraints (in inequality or equation form), and solving that problem means obtaining the set of minimal elements in the feasible region defined by the constraints with respect to the partial order induced by the objective functions.

Polynomial programs have a wide spectrum of applications. Examples of them are capital budgeting (Laughhunn, 1970), capacity planning (Bretthauer and Shetty, 1995), optimization problems in graph theory (Beck and Teboulle, 2000), portfolio selection models with discrete features (Beasley et al., 1995; Jobst et al., 2001) or chemical engineering (Ryoo and Sahinidis, 1995), among many others. The reader is referred to Li and Sun (2006) for further applications.

☆ This research was partially supported by Spanish research grant numbers MTM2007-67433-C02-01, MTM2010-19576-C02-01 and P06-FQM-01366.

E-mail addresses: vblanco@ugr.es (V. Blanco), puerto@us.es (J. Puerto).

0747-7171/$ – see front matter
doi:10.1016/j.jsc.2010.10.003

Polynomial programming generalizes linear and quadratic programming and can serve as a tool to model engineering applications that are expressed by polynomial equations. Even those problems with transcendental terms such as sin, log, and radicals can be reformulated by means of Taylor series as a polynomial program. A survey of the publications on general nonlinear integer programming can be found in Cooper (1981).

We study here multiobjective polynomial integer programs (MOPIP). Thus, we assume that the feasible vectors have integer components and that there are more than one objective function to be optimized. This change makes single objective and multiobjective problems to be treated in a totally different manner, since the concept of solution is not the same.

In this paper, we introduce a new algebraic methodology for solving general MOPIP. The main feature of this approach is the use of Gröbner bases for obtaining the solutions of certain systems of diophantine equations related to different optimality conditions of the multiobjective problem. Gröbner bases were introduced by Bruno Buchberger in 1965 in his Ph.D. Thesis (Buchberger, 1965). He named it Gröbner basis 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 Euclidean algorithm, Gaussian elimination and the Sylvester resultant. One of the outcomes of Gröbner Bases Theory was its application to linear integer programming (Conti and Traverso, 1991; Hoşten and Sturmfels, 1995; Thomas, 1998). Later, Blanco and Puerto (2009) introduced a new notion of partial Gröbner basis for toric ideals in order to solve multiobjective linear integer programs. A different approach for solving linear integer programs was developed by Bertsimas et al. (2000) based on the application of Gröbner bases for solving systems of polynomial equations. This alternative use of Gröbner bases is also used in the paper by Hägglöf et al. (1995) for solving continuous polynomial optimization problems. Further details about Gröbner bases can be found in Cox et al. (2005, 2007). Actually, there are alternative algebraic methods, to the triangularization with lexicographic Gröbner bases, that also use Gröbner bases for solving systems of polynomial equations like companion matrices or resultants Sturmfels (2002). Moreover, nowadays there are other methods, different from those based on Gröbner, to solve systems of polynomial equations such as for instance the 'moment matrix' by Lasserre (2008).

This paper describes different approaches for exactly solving MOPIP using Gröbner bases which are based on reducing the problem to finding solutions of a system of polynomial equations induced by optimality conditions: the necessary Karush–Kuhn–Tucker, the Fritz–John and the multiobjective Fritz–John nondominance conditions.

Clearly, no actually efficient algorithm can be developed for the problem in the paper, unless P=NP. Therefore, the goal of this paper is simply to present and theoretically justify different approaches that can be used to exactly solve MOIP as an alternative to pure brute force (full enumeration). As a byproduct our tools can be easily used as certificates of nondominance for any feasible solution of these problems. In general, no statement can be given on whether our methods improve upon full enumeration. (Note that one can design examples where any feasible solution is nondominated.) Nevertheless, in many cases, as shown in our computational tests, our methods need not enumerate all feasible solutions. This is particular true under convexity assumptions since then, our optimality conditions are not only necessary but also sufficient. In these cases, our approaches will fully enumerate all feasible solutions only if they are all nondominated, and thus in most cases solving the proposed equations will reduce significantly the complete enumeration. In fact, there are no general methodologies for solving the multiobjective polynomial integer problem in the literature if no extra hypotheses are added to the problem (linearity or convexity, among others). A few exact methods appear when the problems have two objectives (Ralphs et al., 2006; Scholz, 2010). On the other hand, some heuristic procedures have been proposed when the structure of the problem is fixed, i.e., quadratic assignment problems (Knowles and Corne, 2002; Li and Landa-Silva, 2009) or quadratic knapsack problems (Vianna and Arroyo, 2004). Further details can be found in Ehrgott and Gandibleux (2004) and Jahn (2004). In all these cases, our tools can be used as certificates of nondominance for the solutions provided by the heuristics. This can be done by replacing the values of the given feasible solution into the corresponding system of polynomial equations and testing if they are compatible, i.e. if there are solutions on the remaining variables that satisfy the system.

In this paper we have implemented our algorithms in MAPLE using standard libraries for obtaining Gröbner bases. The only purpose of this implementation was to compare the different approaches that we have introduced. This simple implementation allowed us to solve problems with several objectives and up to 13 integer variables. Needless to say that better implementations that moreover use more powerful tools for obtaining Gröbner bases would augment the size of the instances solved by our algorithms.

The paper is structured as follows. In the next section we give some preliminaries in multiobjective polynomial integer optimization. We present in Section 3 our first algorithm for solving MOPIP using only the triangularization property of lexicographic Gröbner bases. Section 4 is devoted to two different algorithms for solving MOPIP using a Chebyshev like scalarization and the Karush–Kuhn–Tucker or the Fritz–John optimality conditions. The last algorithm, based on the multiobjective Fritz–John optimality condition, is described in Section 5. In Section 6, we compare the algorithms with the results of some computational experiments and its analysis. Finally, in Section 7 we draw some conclusions about the contributions of this paper and further research.

2. The multiobjective integer polynomial problem

The goal of this paper is to the solve multiobjective polynomial integer programs (MOPIP):

$$\begin{array}{lll} \min & (f_1(x), \ldots, f_k(x)) & \\ s.t. & g_j(x) \leq 0 \quad j = 1, \ldots, m & \\ & h_r(x) = 0 \quad r = 1, \ldots, s & \\ & x \in \mathbb{Z}_+^n & \end{array} \qquad (MOPIP_{\mathbf{f},\mathbf{g},\mathbf{h}})$$

with $f_1, \ldots, f_k, g_1, \ldots, g_m, h_1, \ldots, h_s$ polynomials in $\mathbb{R}[x_1, \ldots, x_n]$ and the constraints defining a bounded feasible region. Therefore, from now on we deal with $MOPIP_{\mathbf{f},\mathbf{g},\mathbf{h}}$ and we denote $\mathbf{f} = (f_1, \ldots, f_k)$, $\mathbf{g} = (g_1, \ldots, g_m)$ and $\mathbf{h} = (h_1, \ldots, h_r)$. If the problem had no equality (resp. inequality) constraints, we would denote it by $MOPIP_{\mathbf{f},\mathbf{g}}$ (resp. $MOPIP_{\mathbf{f},\mathbf{h}}$), avoiding the nonexistent term.

However, ($MOPIP_{\mathbf{f},\mathbf{g},\mathbf{h}}$) can be transformed to an equivalent multiobjective polynomial binary problem. Since the feasible region $\{x \in \mathbb{R}_+^n : g_j(x) \leq 0, \; h_r(x) = 0, j = 1, \ldots, m, r = 1, \ldots, s\}$ is assumed to be bounded, it can be always embedded in a hypercube $\prod_{i=1}^n [0, u_i]^n$. Then, every component in x, x_i, has an additional, but redundant, constraint $x_i \leq u_i$. We write x_i in binary form, introducing new binary variables z_{ij} with values in $\{0, 1\}$, $x_i = \sum_{j=0}^{\lfloor \log u_i \rfloor} 2^j z_{ij}$, substituting every x_i in ($MOPIP_{\mathbf{f},\mathbf{g},\mathbf{h}}$) we obtain an equivalent 0–1 problem.

Then, from now on, without loss of generality, we restrict ourselves to multiobjective polynomial binary programs ($MOPBP$) in the form:

$$\begin{array}{lll} \min & (f_1(x), \ldots, f_k(x)) & \\ s.t. & g_j(x) \leq 0 \quad j = 1, \ldots, m & \\ & h_r(x) = 0 \quad r = 1, \ldots, s & \\ & x \in \{0, 1\}^n. & \end{array} \qquad (MOPBP_{\mathbf{f},\mathbf{g},\mathbf{h}})$$

If the problem had no equality (resp. inequality) constraints, we would denote the problem by $MOPBP_{\mathbf{f},\mathbf{g}}$ (resp. $MOPBP_{\mathbf{f},\mathbf{h}}$), avoiding the nonexistent term.

Clearly, the number of solutions of the above problem is finite, since the decision space is finite.

It is clear that $MOPBP_{\mathbf{f},\mathbf{g},\mathbf{h}}$ is not a standard optimization problem since the objective function is a k-coordinate 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.

A feasible vector $\widehat{x} \in \mathbb{R}^n$ is said to be a *nondominated* (or Pareto optimal) solution of $MOPIP_{\mathbf{f},\mathbf{g}}$ if there is no other feasible vector y such that

$$f_j(y) \leq f_j(\widehat{x}) \quad \forall j = 1, \ldots, k$$

with at least one strict inequality for some j. If x is a nondominated solution, the vector $\mathbf{f}(\mathbf{x}) = (f_1(x), \ldots, f_k(x)) \in \mathbb{R}^k$ is called *efficient*.

We say that a feasible solution, y, is dominated by a feasible solution, x, if $f_i(x) \leq f_i(y)$ for all $i = 1, \ldots, k$ and $\mathbf{f}(x) \neq \mathbf{f}(y)$. We denote by X_E the set of all nondominated solutions for ($MOPBP_{\mathbf{f},\mathbf{g},\mathbf{h}}$)

and by Y_E the image under the objective functions of X_E, that is, $Y_E = \{\mathbf{f}(x) : x \in X_E\}$. Note that X_E is a subset of $\mathbb{R}^n$ (*decision space*) and Y_E is a subset of $\mathbb{R}^k$ (*space of objectives*).

From the objective functions $\mathbf{f} = (f_1, \ldots, f_k)$, we obtain a partial order on $\mathbb{Z}^n$ as follows:

$$x \prec_{\mathbf{f}} y :\Longleftrightarrow \mathbf{f}(x) \lneqq \mathbf{f}(y) \text{ or } x = y.$$

Note that since $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^k$, the above relation is not complete. Hence, there may exist incomparable vectors.

In the following sections we describe some algorithms for solving MOPIP using tools from algebraic geometry. In particular, in each of these methods, we transform our problem in a certain system of polynomial equations, and we use Gröbner bases to solve it.

3. Obtaining nondominated solutions by solving systems of polynomial equations

In this section we present the first approach for solving multiobjective polynomial integer programs using Gröbner bases. For this method, we transform the program in a system of polynomial equations that encodes the set of feasible solutions and its objective values. Solving that system in the objective values, and then, selecting the minimal ones in the partial componentwise order, allows us to obtain the associate feasible vectors, thus, the nondominated solutions.

Through this section we solve $MOPBP_{\mathbf{f},\mathbf{h}}$. Without loss of generality, we reduce the general problem to the problem without inequality constraints since we can transform inequality constraints to equality constraints as follows:

$$g(x) \leq 0 \Longleftrightarrow g(x) + z^2 = 0, \quad z \in \mathbb{R}, \tag{1}$$

where the quadratic term, z^2, assures the nonnegativity of the slack variable and then, less than or equal to type inequality. Initially, we suppose that all the variables are binary. In Remark 3.1 we describe how to modify the algorithm to incorporate the above slack variables.

This approach consists of transforming $MOPBP_{\mathbf{f},\mathbf{h}}$ to an equivalent problem such that the objective functions are part of the constraints. For this transformation, we add k new variables, $y_1, \ldots, y_k$ to the problem, encoding the objective values for all feasible solutions. The modified problem is:

$$\begin{array}{lrl} \min & (y_1, \ldots, y_k) & \\ s.t. & h_r(x) = 0 & r = 1, \ldots, s \\ & y_j - f_j(x) = 0 & j = 1, \ldots, k \\ & x_i(x_i - 1) = 0 & i = 1, \ldots, n \\ & y \in \mathbb{R}^k \; x \in \mathbb{R}^n & \end{array} \tag{2}$$

where integrality constraints are encoded as quadratic constraints so, $MOPBP_{\mathbf{f},\mathbf{h}}$ is a polynomial continuous problem.

The algorithm consists of, first, obtaining the set of feasible solutions of Problem (2) in the y-variables; then, selecting from that set those solutions that are minimal with respect to the componentwise order, obtaining the set of efficient solutions of $MOPBP_{\mathbf{f},\mathbf{h}}$. The feasible solutions in the x-variables associated to those efficient solutions correspond with the nondominated solutions of $MOPBP_{\mathbf{f},\mathbf{h}}$.

For the sake of completeness and to enhance readability, for readers from the optimization field and nonspecialist in algebraic geometry, we describe a procedure for solving the system of polynomial equations that encodes the feasible region of Problem (2), i.e. the solutions of

$$\begin{array}{lll} h_r(x) & = 0 & \text{for all } r = 1, \ldots, s \\ y_j - f_j(x) & = 0 & \text{for all } j = 1, \ldots, k \\ x_i(x_i - 1) & = 0 & \text{for all } i = 1, \ldots, n. \end{array} \tag{3}$$

In order to analyze System (3) we use Gröbner bases as a tool for solving systems of polynomial equations. Further details can be found in the book by Sturmfels (2002). Clearly, any method for solving these kinds of systems could be used, in particular, the recent methodology by Lasserre (2008), as well as resultants, companion matrices or Gröbner bases with different elimination orderings.

The set of solutions of (3) coincides with the affine variety of the following polynomial ideal in $\mathbb{R}[y_1, \dots, y_k, x_1, \dots, x_n]$:

$$I = \langle h_1(x), \dots, h_m(x), y_1 - f_1(x), \dots, y_k - f_k(x), x_1(x_1 - 1), \dots, x_n(x_n - 1)\rangle.$$

Note that I is a zero-dimensional ideal since the number of solutions of the equations that define I is finite. Let $V(I)$ denote the affine variety of I. If we restrict I to the family of variables x (resp. y) the variety $V(I \cap \mathbb{R}[x_1, \dots, x_n])$ (resp. $V(I \cap \mathbb{R}[y_1, \dots, y_k])$) encodes the set of feasible solutions (resp. the set of possible objective values) for that problem.

Applying the elimination property, the reduced Gröbner basis for I, $\mathcal{G}$, with respect to the lexicographical ordering with $y_k \prec \cdots \prec y_1 \prec x_n \prec \cdots \prec x_1$ gives us a method for solving system (3) sequentially, i.e., solving in one indeterminate at a time. Explicitly, the shape of $\mathcal{G}$ is:

(1) $\mathcal{G}$ contains one polynomial in $\mathbb{R}[y_k]$: $p_k(y_k)$

(2) $\mathcal{G}$ contains one or several polynomials in $\mathbb{R}[y_{k-1}, y_k] : p^1_{k-1}(y_{k-1}, y_k), \dots, p^{m_{k-1}}_{k-1}(y_{k-1}, y_k)$.

⋮

$(k+1)$ $\mathcal{G}$ contains one or several polynomials in $\mathbb{R}[x_n, y_1, \dots, y_k] : q^1_n(x_n, \mathbf{y}), \dots, q^{s_n}_n(x_n, \mathbf{y})$.

⋮

$(k+n)$ $\mathcal{G}$ contains one or several polynomials in $\mathbb{R}[x_n, y_1, \dots, y_k] : q^1_1(x_1, \dots, x_n, \mathbf{y}), \dots, q^{s_1}_n(x_1, \dots, x_n, \mathbf{y})$.

Then, with this structure of $\mathcal{G}$, we can solve, in a first step, the system in the y-variables using those polynomials in $\mathcal{G}$ that only involve this family of variables as follows: we first solve for y_k in $p_k(y_k) = 0$, obtaining the solutions: $y^1_k, y^2_k, \dots$. Then, for fixed y^r_k, we find the common roots of $p^1_{k-1}, p^2_{k-1}, \dots$ getting solutions $y^1_{k-1,r}, y^2_{k-1,r}, \dots$ and so on, until we have obtained the roots for $p_1(y_1, \dots, y_k)$. Note that at each step we only solve one variable polynomial equations.

We denote by Ω the above set of solutions in vector form

$$\Omega = \{(\hat{y}_1, \dots, \hat{y}_k) : p_k(\hat{y}_k) = 0, p^1_{k-1}(\hat{y}_{k-1}, \hat{y}_k) = 0, \dots, p^{m_{k-1}}_{k-1}(\hat{y}_{k-1}, \hat{y}_k) = 0, \dots \\ p^1_1(\hat{y}_1, \hat{y}_2, \dots, \hat{y}_k) = 0, \dots, p^{m_1}_1(\hat{y}_1, \hat{y}_2, \dots, \hat{y}_k) = 0\}.$$

As we stated above, Ω is the set of all possible values of the objective functions at the feasible solutions of $MOPBP_{\mathbf{f},\mathbf{h}}$. We are looking for the nondominated solutions that are associated with the efficient solutions. From Ω, we can select the efficient solutions as those that are minimal with respect to the componentwise order in $\mathbb{R}^k$. So, we can extract from Ω the set of efficient solutions, Y_E:

$$Y_E = \{(y^*_1, \dots, y^*_k) \in \Omega : \nexists (y'_1, \dots, y'_k) \in \Omega \text{ with } y'_j \le y^*_j \text{ for } \\ j = 1, \dots, k \text{ and } (y'_1, \dots, y'_k) \neq (y^*_1, \dots, y^*_k)\}.$$

Once we have obtained the solutions in the y-variables that are efficient solutions for $MOPBP_{\mathbf{f},\mathbf{h}}$, we compute with an analogous procedure the nondominated solutions associated to the y-values in Y_E. It consists of solving the triangular system given by $\mathcal{G}$ for the polynomial where the x-variables appear once the values for the y-variables are fixed to be each of the vectors in Y_E.

A pseudocode for this procedure is described in Algorithm 1.

Theorem 3.1. *Algorithm 1 either provides all nondominated and efficient solutions or provides a certificate of infeasibility whenever $G = \{1\}$.*

Proof. Suppose that $G \neq \{1\}$. Then, G^y_{k-1} has exactly one element, namely $p(y_k)$. This follows from the observation that $I \cap \mathbb{R}[y_k]$ is a polynomial ideal in one variable, and therefore, needs only one generator.

Solving $p(y_k) = 0$ we obtain every $\hat{y}_k \in V(G^y_{k-1})$. Sequentially we obtain $\hat{y}_{k-1}$ extending $\hat{y}_k$ to the partial solutions $(\hat{y}_{k-1}, \hat{y}_k)$ in $V(G^y_{k-1})$ and so on.

By the Extension Theorem, this is always possible in our case.

Continuing in this way and applying the Extension Theorem, we can obtain all solutions $(\hat{y}_1, \dots, \hat{y}_k)$ in $V(G \cap \mathbb{R}[y_1, \dots, y_k]$. These vectors are all the possible objective values for all feasible

Algorithm 1: Solving MOPIP by solving systems of polynomial equations

Input : $f_1, \ldots, f_k, h_1, \ldots h_s \in \mathbb{R}[x_1, \ldots, x_n]$
Initialization: $I = \langle f_1 - y_1, \ldots, f_k - y_k, h_1, \ldots, h_s, x_1(x_1-1), \ldots, x_n(x_n-1)\rangle$.
Algorithm:
Step 1. Compute a Gröbner basis, G, for I with respect to a lexicographic order with $y_k \prec \cdots \prec y_1 \prec x_n \prec \cdots \prec x_1$.
Step 2. Let $G_l^y = G \cap \mathbb{R}[y_{l+1}, \ldots, y_k]$ be a Gröbner basis for $I_l^y = I \cap \mathbb{R}[y_{l+1}, \ldots, y_k]$, for $l = 0, \ldots, k-1$. (By the Elimination Property).
1. Find all $\hat{y}_k \in V(G_{k-1}^y)$.
2. Extend every $\hat{y}_k$ to $(\hat{y}_{k-1}, \hat{y}_k) \in V(G_{k-2}^y)$.
⋮
$(k-1)$ Extend every $(\hat{y}_3, \ldots, \hat{y}_k)$ to $(\hat{y}_2, \hat{y}_3, \ldots, \hat{y}_k) \in V(G_1^y)$.
(k) Find all $\hat{y}_1$ such that $(\hat{y}_1, \ldots, \hat{y}_k) \in V(G_0^y)$.
Step 3. Select from $V(G_0^y)$ the minimal vectors with respect to the usual componentwise order in $\mathbb{R}^k$. Set Y_E this subset.
Step 4. Let $G_l = G \cap \mathbb{R}[y_1, \ldots, y_k, x_{l+1}, \ldots, x_n]$ be a Gröbner basis for $I_l \cap \mathbb{R}[y_1, \ldots, y_k, x_{l+1}, \ldots, x_n]$, for $l = 0, \ldots, n-1$. (By the Elimination Property). Denote by $S_l = \{(\hat{y}_1, \ldots, \hat{y}_k, \hat{x}_{l+1}, \ldots, \hat{x}_n) : (\hat{y}_1, \ldots, \hat{y}_k) \in Y_E$, and exists $(x_1, \ldots, x_l)$ such that $(x_1, \ldots, x_n)$ is feasible$\}$ for $l = 0, \ldots, n-1$.
1. Find all $\hat{x}_n$ such that $(\hat{y}_1, \ldots, \hat{y}_k, \hat{x}_n) \in V(G_{n-1}) \cap S_{n-1}$.
2. Extend every $\hat{x}_n$ to $((\hat{y}_1, \ldots, \hat{y}_k, \hat{x}_{n-1}, \hat{x}_n) \in V(G_{n-2}) \cap S_{n-2}$.
⋮
$(n-1)$ Extend every $(\hat{y}_1, \ldots, \hat{y}_k, \hat{x}_3, \ldots, \hat{x}_n)$ to $(\hat{y}_1, \ldots, \hat{y}_k, \hat{x}_2, \hat{x}_3, \ldots, x_n) \in V(G_1) \cap S_1$.
(n) Find all $\hat{x}_1$ such that $(\hat{y}_1, \ldots, \hat{y}_k, \hat{x}_1, \ldots, \hat{x}_n) \in V(G_0) \cap S_0$.
Set $X_E = \pi_x(V(G_0) \cap S_0)$, where π_x denotes the projection over the x-variables.

Output: Y_E the set of efficient solutions and X_E the set of nondominated solutions for $MOPBP_{\mathbf{f},\mathbf{h}}$.

solutions of the problem. Selecting from $V(G \cap \mathbb{R}[y_1, \ldots, y_k])$ those solutions that are not dominated in the componentwise order in $\mathbb{R}^k$, we obtain Y_E.

Following a similar scheme in the x-variables, we have the set $V(G_0) \cap S_0^*$ encoding all efficient (in the first k coordinates) and nondominated (in the last n coordinates) solutions.

Finally, if $G = \{1\}$, then, the ideal I coincides with $\mathbb{R}[y_1, \ldots, y_k, x_1, \ldots, x_n]$, indicating that $V(I)$ is empty (it is the set of the common roots of all polynomials in $\mathbb{R}[y_1, \ldots, y_k, x_1, \ldots, x_n]$). Then, we have an infeasible integer problem. □

Remark 3.1. In the case where we add slack variables, as explained in (1), we slightly modify the above algorithm solving first in the slack variables and selecting those solutions that are real numbers. Then we continue with the procedure as described in Algorithm 1.

The following example illustrates how Algorithm 1 works. In this case, the feasible region has two different connected components and one of these components is not convex.

Example 3.1. Consider the following biobjective polynomial integer problem.

$$\begin{aligned} \min \quad & (f_1(x_1,x_2), f_2(x_1,x_2)) = (x_1^2 - x_2, x_1 - x_2^2) \\ s.t. \quad & \\ & g_1(x_1,x_2) = x_2 - x_1^4 + 10x_1^3 - 30x_1^2 + 25x_1 - 7 \geq 0 \\ & g_2(x_1,x_2) = x_2 - x_1^3 + 9x_1^2 - 25x_1 + 12 \leq 0 \\ & x_1, x_2 \in \mathbb{Z}_+. \end{aligned} \tag{4}$$

The feasible region for this problem is shown in Fig. 1.

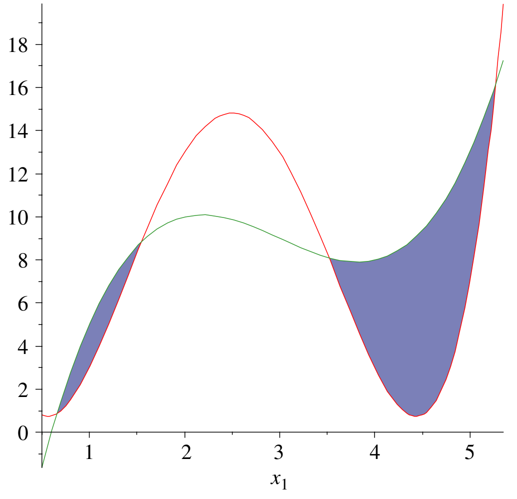

Fig. 1. Feasible region of Example 3.1.

To solve the problem using Algorithm 1, first we need to transform it to a binary problem with equality constraints. For the first task, we substitute each variable, x_1 and x_2 by their binary expressions, taking into account that x_1 is bounded from above by 6 and x_2 by 15 (and then, we need $\lceil \log_2 6 \rceil + 1 = 4$ and $\lceil \log_2 15 \rceil + 1 = 5$ auxiliary z-variables to express x_1 and x_2 in binary code, respectively). For the second task, we add to the problem two new variables w_1 and w_2. Then, the problem is equivalent to the following multiobjective problem. After the change of variables, Problem (4) is re-written as:

$$\begin{array}{lrl}
\min & & (y_1, y_2) \\
s.t. & & \\
& g_1(z_{11}, z_{12}, z_{13}, z_{14}, z_{21}, z_{22}, z_{23}, z_{24}, z_{25}) - w_1^2 & = 0 \\
& g_2(z_{11}, z_{12}, z_{13}, z_{14}, z_{21}, z_{22}, z_{23}, z_{24}, z_{25}) + w_2^2 & = 0 \\
& y_1 - f_1(z_{11}, z_{12}, z_{13}, z_{14}, z_{21}, z_{22}, z_{23}, z_{24}, z_{25}) & = 0 \\
& y_2 - f_2(z_{11}, z_{12}, z_{13}, z_{14}, z_{21}, z_{22}, z_{23}, z_{24}, z_{25}) & = 0 \\
& z_{ij}^2 - z_{ij} & = 0 \\
& z_{ij}, w_k, y_l & \in \mathbb{R}.
\end{array}$$

Now, we compute the reduced Gröbner basis for the set of polynomials that define the feasible region of the above problem with respect to the lexicographic ordering such that $\mathbf{z} \succ \mathbf{y} \succ \mathbf{w}$. Running MAPLE 11 using the package `Groebner` and the procedure `Solve`, we compute the following 16 solutions for the y-variables:

$$\begin{aligned}
&(y_1, y_2) \in \{(16, -76), (9, -45), (17, -59), (14, -116), (11, -21), (13, -139), (-3, -15), \\
&(13, -5), (-2, -8), (12, -12), (12, -164), (15, -95), (10, -32), \\
&(18, -44), (8, -60), (-4, -24)\}.
\end{aligned}$$

The minimal elements (with respect to the componentwise ordering in $\mathbb{R}^2$) are:

$$Y_E = \{(12, -164), (8, -60), (-4, -24)\},$$

whose values in the z-variables are:

$$Z_E = \{(0, 0, 1, 0, 0, 0, 0, 1, 0), (1, 0, 0, 0, 1, 0, 1, 0, 0), (1, 0, 1, 0, 1, 0, 1, 1, 0)\},$$

and translating to values in the original x-variables we have that the set of nondominated solutions for the problem is:

$$X_E = \{(4, 8), (5, 13), (1, 5)\}.$$

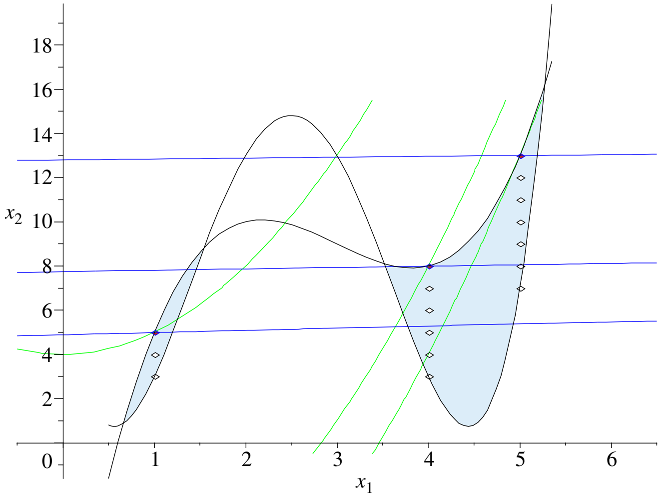

Fig. 2. Feasible region, the nondominated solutions and the level curves of Example 3.1.

Fig. 2 shows these solutions in the feasible region of the problem and the level curves of both objective functions at each of these solutions.

4. Obtaining nondominated solutions by the Chebyshev norm approach

In this section we describe two additional methods for solving MOPIP based on a different rationale, namely scalarizing the multiobjective problem and solving it as a parametric single objective problem. We propose a methodology based on the application of optimality conditions to a family of single objective problems related to our original multiobjective problem. The methods consist of two main steps: a first step where the multiobjective problem is scalarized to a family of single objective problems such that each nondominated solution is an optimal solution for at least one of the single objective problems in that family; and a second step that consists of applying necessary optimality conditions to each one of the problems in the family, to obtain their optimal solutions. Those solutions are only candidates to be nondominated solutions of the multiobjective problem since we just use necessary conditions.

For the first step, the scalarization, we use a weighted Chebyshev norm approach. Other weighted sum approaches could be used to transform the multiobjective problem to a family of single objective problems whose set of solutions contains the set of nondominated solutions of our problem. However, the Chebyshev approach seems to be rather adequate since it does not require to impose extra hypothesis to the problem. This approach can be improved for problems satisfying convexity conditions, where alternative well-known results can be applied (see Jahn (2004) for further details).

For the second step, we use the Fritz–John and Karush–Kuhn–Tucker necessary optimality conditions, giving us two different approaches. In this section we describe both methodologies since each of them has its own advantages over the other.

For applying the Chebyshev norm scalarization, we use the following result that states how to transform our problem to a family of single objective problems, and how to obtain nondominated solutions from the optimal solution of those single objective problems. Further details and proofs of this result can be found in Jahn (2004).

Theorem 4.1 (*Corollary 11.21 in Jahn, 2004*). *Let (MOPBP$_{\mathbf{f,g,h}}$) be feasible. x^* is a nondominated solution of (MOPBP$_{\mathbf{f,g,h}}$) if and only if there are positive real numbers $\omega_1, \ldots, \omega_k > 0$ such that x^* is an image unique solution of the following weighted Chebyshev approximation problem:*

$$\begin{array}{lrl} \min & \gamma & \\ s.t. & \omega_i\,(f_i(x)-\hat{y}_i)-\gamma \leq 0 & i=1,\ldots,k \\ & g_j(x) \leq 0 & j=1,\ldots,m \\ & h_r(x)=0 & r=1,\ldots,s \\ & x_i(x_i-1)=0 & i=1,\ldots,n \\ & \gamma \in \mathbb{R}\; x \in \mathbb{R}^n & \end{array} \qquad (P_\omega)$$

where $\hat{y}=(\hat{y}_1,\ldots,\hat{y}_k)\in\mathbb{R}^k$ is a lower bound of $f=(f_1,\ldots,f_k)$, i.e., $\hat{y}_i \leq f_i(x)$ for all feasible solution x and $i=1,\ldots,k$.

According to the above result, every nondominated solution of ($MOPBP_{\mathbf{f,g,h}}$) is the unique solution of (P_ω) for some $\omega > 0$. We apply, in the second step, necessary optimality conditions for obtaining the optimal solutions for those problems (taking ω as parameters). These solutions are candidates to be nondominated solutions of our original problem. Actually, every nondominated solution is among those candidates.

In the following subsections we describe the above mentioned two methodologies for obtaining the optimal solutions for the scalarized problems (P_ω) for each ω.

4.1. The Chebyshev–Karush–Kuhn–Tucker approach

The first optimality conditions that we apply are the Karush–Kuhn–Tucker (KKT) necessary optimality conditions, that were stated, for the general case, as follows (see e.g. Bazaraa et al. (1993) for further details):

Theorem 4.2 (*KKT Necessary Conditions*)**.** *Consider the problem:*

$$\begin{array}{ll} \min & f(x) \\ s.t. & g_j(x) \leq 0 \quad j=1,\ldots,m \\ & h_r(x)=0 \quad r=1,\ldots,s \\ & x\in\mathbb{R}^n. \end{array} \qquad (5)$$

Let x^* be a feasible solution, and let $J=\{j : g_j(x^*)=0\}$. Suppose that f and g_j, for $j=1,\ldots,m$, are differentiable at x^*, that g_j, for $j\notin J$, is continuous at x^*,and that h_r, for $r=1,\ldots,s$, is continuously differentiable at x^*. Further suppose that $\nabla g_j(x^*)$, for $j\in I$, and $\nabla h_r(x^*)$, for $r=1,\ldots,s$, are linearly independent (regularity conditions). If x^* solves Problem (5) locally, then there exist scalars λ_j, for $j=1,\ldots,m$, and μ_r, for $r=1,\ldots,s$, such that

$$\begin{array}{lr} \nabla f(x^*)+\sum_{j=1}^{m}\lambda_j\nabla g_j(x^*)+\sum_{r=1}^{s}\mu_r\nabla h_r(x^*)=0 & \\ \lambda_j\, g_j(x^*)\ =0 & \text{for } j=1,\ldots,m \\ \lambda_j\ \geq 0 & \text{for } j=1,\ldots,m. \end{array} \qquad \text{(KKT)}$$

From the above theorem the candidates to be optimal solutions for Problem (5) are those that either satisfy the KKT conditions (in the case where all the functions involved in Problem (5) are polynomials, this is a system of polynomial equations) or do not satisfy the regularity conditions. Note that these two sets are, in general, not disjoint.

Regularity conditions can also be formulated as a system of polynomial equations when the involved functions are all polynomials. Let x^* be a feasible solution for Problem (5), x^* does not verify the regularity conditions if there exist scalars λ_j, for $j\in J$, and μ_r, for $r=1,\ldots,s$, not all equal to zero, such that:

$$\sum_{j\in I}\lambda_j\nabla g_j+\sum_{r=1}^{s}\mu_r\nabla h_r=0. \qquad \text{(Non-Regularity)}$$

The above discussion justifies the following result.

Corollary 4.1. *Let x^* be a nondominated solution for (MOPBP$_{\mathbf{f,g,h}}$). Then, x^* is a solution of the systems of polynomial equations (6) or (7), for some $\omega > 0$.*

$$\left.\begin{array}{r} 1 - \sum_{i=1}^{k} \nu_i = 0 \\ \sum_{i=1}^{k} \nu_i\omega_i \nabla f_i(x) + \sum_{j=1}^{m} \lambda_j \nabla g_j(x) + \sum_{r=1}^{s} \mu_j \nabla h_r(x) + \sum_{i=1}^{n} \beta_i e_i\,(2x_i - 1) = 0 \\ \nu_i(\omega_i\,(f_i(x) - \hat{y}_i) - \gamma) = 0, \quad i = 1, \dots, k \\ \lambda_j\, g_j(x) = 0, \quad \text{for } j = 1, \dots, m \end{array}\right\} \tag{6}$$

with $x \in \mathbb{R}^n$ such that $g_j(x) \le 0$, for $j = 1, \dots, m$, $h_r(x) = 0$, for $r = 1, \dots, s$ and for some $\lambda_j \ge 0$, for $j = 1, \dots, m$, $\nu_i \ge 0$, for $i = 1, \dots, k$ and e_i is the ith unit vector in the standard basis of $\mathbb{R}^n$, for $i = 1, \dots, k$.

$$\left.\begin{array}{r} \sum_{i=1}^{k} \nu_i = 0 \\ \sum_{i=1}^{k} \nu_i\omega_i \nabla f_i(x) + \sum_{j=1}^{m} \lambda_j \nabla g_j(x) + \sum_{r=1}^{s} \mu_j \nabla h_r(x) + \sum_{i=1}^{n} \beta_i e_i\,(2x_i - 1) = 0 \\ \omega_i\,(f_i(x) - \hat{y}_i) - \gamma \le 0, \quad i = 1, \dots, k \end{array}\right\} \tag{7}$$

with $x \in \mathbb{R}^n$ such that $g_j(x) \le 0$, for $j = 1, \dots, m$, $h_r(x) = 0$, for $r = 1, \dots, s$ and for some $\lambda_j \ge 0$, for $j = 1, \dots, m$, and $\nu_i \ge 0$, for $i = 1, \dots, k$ with $(\lambda, \mu, \nu, \beta) \neq \mathbf{0}$.

Let X_E^{KKT} denote the set of solutions, in the x-variables, of system (6) and let X_E^{NR} denote the set of solutions, in the x-variables, of system (7) (the problem is solved by avoiding inequality constraints, then every solution is evaluated to check if it satisfies the inequality constraints).

For solving these systems (Chebyshev–KKT and Non-Regularity), we use a Gröbner basis approach. Let I be the ideal generated by the involved equations.

Let us consider a lexicographical order over the monomials in $\mathbb{R}[\mathbf{x}, \gamma, \lambda, \nu, \mu, \beta]$ such that $\mathbf{x} \prec \gamma \prec \lambda \prec \nu \prec \mu \prec \beta$. Then, the Gröbner basis, $\mathcal{G}$, for I with this order has the following triangular shape:

- $\mathcal{G}$ contains one polynomial in $\mathbb{R}[x_n]$: $p_n(x_n)$
- $\mathcal{G}$ contains one or several polynomials in $\mathbb{R}[x_{n-1}, x_n] : p_{n-1}^1(x_{n-1}, x_n), \dots, p_{n-1}^{m_1}(x_{n-1}, x_n)$.

$\vdots$

- $\mathcal{G}$ contains one or several polynomials in $\mathbb{R}[\mathbf{x}] : p_1^1(x_1, \dots, x_n), \dots, p_1^{m_n}(x_1, \dots, x_n)$.
- The remaining polynomials involve variables $\mathbf{x}$ and at least one γ, λ, μ, ν or β.

We are interested in finding only the values for the x-variables, so, we avoid the polynomials in $\mathcal{G}$ that involve any of the other auxiliary variables. In general, we are not able to discuss about the values of the parameters γ, λ, μ, ν and β. Needless to say that in those cases when we can do it, some values of $\mathbf{x}$ may be discarded, simplifying the process. We denote by $\mathcal{G}^x$ the subset of $\mathcal{G}$ that contains only polynomials in the x-variables. By the Extension Theorem, $\mathcal{G}^x$ is a Gröbner basis for $I \cap \mathbb{R}[x_1, \dots, x_n]$.

Solving the system given by $\mathcal{G}^x$ and checking the feasibility of those solutions, we obtain as solutions those of our KKT or Non-Regularity original systems.

It is clear that the set of nondominated solutions of our problem is a subset of $X_E^{KKT} \cup X_E^{NR}$, since either a solution is regular, and then, KKT conditions are applicable or it satisfies the Non-Regularity conditions. However, the set $X_E^{KKT} \cup X_E^{NR}$ may contain dominated solutions, so, at the end we must remove the dominated ones to get only X_E.

The steps to solve Problem (*MOPBP*$_{\mathbf{f,g,h}}$) using the Chebyshev–KKT approach are summarized in Algorithm 2.

Theorem 4.3. *Algorithm 2 solves Problem (MOPBP$_{\mathbf{f,g,h}}$) in a finite number of steps.*

Algorithm 2: Summary of the procedure for solving MOPBP using Chebyshev–KKT approach.

Input : $f_1, \ldots, f_k, g_1, \ldots g_m, h_1, \ldots, h_r \in \mathbb{R}[x_1, \ldots, x_n]$
Algorithm:
Step 1 Formulate the Chebyshev scalarization of ($MOPBP_{\mathbf{f,g,h}}$). (Problem (P_ω))
Step 2 Solve System (6) in the x-variables: X_E^{KKT}.
Step 3 Solve System (7) in the x-variables: X_E^{NR}.
Step 4 Remove from $X_E^{KKT} \cup X_E^{NR}$ the subset of dominated solutions: X_E.

Output: X_E the set of nondominated solutions for ($MOPBP_{\mathbf{f,g,h}}$)

The following example illustrates how the above the algorithm works.

Example 4.1. Consider the following biobjective problem:

$$\begin{array}{ll} \min & (-5x_1 + 8x_2 + 5x_1x_2, 4x_1 + 2x_1x_2) \\ s.t. & -x_1 + x_1^3 + x_2^4 \geq 0 \\ & x_1, x_2 \in \{0, 1, 2\}. \end{array}$$

The Chebyshev scalarization of this problem is:

$$\begin{array}{ll} \min & \gamma \\ s.t. & \omega_1 (-5x_1 + 8x_2 + 5x_1x_2 - (-5)) - \gamma \leq 0 \\ & \omega_2 (4x_1 + 2x_1x_2 - (0)) - \gamma \leq 0 \\ & 32\, x_1 - 32\, x_1^3 - (2x_2 + 1)^4 \leq 0 \\ & \gamma, \omega_1, \omega_2 \in \mathbb{R}, x_1, x_2 \in \{0, 1, 2\}. \end{array}$$

Then, the KKT system is the one given below (after transforming the constraints $x_i \in \{0, 1, 2\}$ in $x_i\,(x_i - 1)\,(x_i - 2) = 0$)

$$\begin{aligned} \beta_1 (3\, x_1^2 - 6x_1 + 2) + \nu_1\, \omega_1 (-5 + 5\, x_2) + \nu_2\, \omega_2 (4 + 2\, x_2) - \lambda_1 (32 - 96\, x_1^2) &= 0 \\ \beta_2 (3\, x_2^2 - 6x_2 + 2) + \nu_1\, \omega_1 (8 + 5\, x_1) + 2\, \nu_2\, \omega_2\, x_1 + 8\, \lambda_1 (2x_2 + 1)^3 &= 0 \\ 1 - \nu_1 - \nu_2 &= 0 \\ x_1^3 - 3x_1^2 + 2x_1 &= 0 \\ x_2^3 - 3x_2^2 + 2x_2 &= 0 \\ \lambda_1 (32\, x_1 - 32\, x_1^3 - (2x_2 + 1)^4) &= 0 \\ \nu_1 (\omega_1 (-5\, x_1 + 8\, x_2 + 5\, x_1\, x_2 + 5) - \gamma) &= 0 \\ \nu_2 (\omega_2 (4\, x_1 + 2\, x_1\, x_2) - \gamma) &= 0. \end{aligned}$$

We solve this system sequentially using a Gröbner basis with respect to a lexicographic ordering with $\nu \prec \lambda \prec x \prec \gamma \prec \beta$.

Sequentially we can solve the above system (the one given by the Gröbner basis). Then, discarding solutions (taking into account that $\lambda_1, \lambda_2, \mu_1, \mu_2, \nu \geq 0$ and $\omega_1, \omega_2 > 0$ and also the requirements of Theorem 4.1), we have as possible nondominated solutions in the x-variables:

$$\{(1, 2), (2, 0), (0, 0), (1, 0), (2, 1)\}.$$

Note that the solutions (0, 1) and (1, 1) are discarded since they do not fulfill the requirement of Theorem 4.1. The reader can check that in all tuples where these solutions appear there exists another tuple in another solution having the same image solution in Problem (P_ω), and then being not unique. For instance, the solutions $\{x_1 = 1, x_2 = 1, \gamma = 13\,\omega_1, \omega_1, \omega_2, \nu_1 = 1, \nu_2 = 0, \lambda_1 = 0, \beta_1 = -13\,\omega_1, \beta_2 = 0\}$ and $\{x_1 = 0, x_2 = 1, \gamma = 13\,\omega_1, \omega_1, \omega_2, \mu_1 = 1, \mu_2 = 0, \lambda_1 = 0, \beta_1 = -8\omega_1, \beta_2 = 0\}$ are not considered since for any pair $\omega_1, \omega_2 > 0$, the feasible solutions (1, 1) and (0, 1) are both optimal solutions for Problem (P_ω). Finally, we point out that the solution (0, 2) is not considered since it only appears as a non-unique optimal solution of Problem P_ω. (Note that one of the tuples with $x_1 = 0$ and $x_2 = 0$ is also an optimal solution for the same problem.)

Now, the Non-Regularity system is:

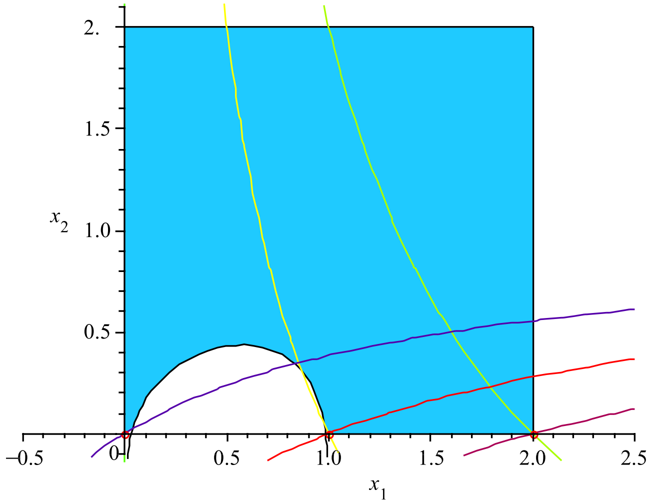

Fig. 3. Feasible region, the nondominated solutions and the level curves of Example 4.1.

$$\begin{aligned}
\beta_1\,(3\,x_1^2-6x_1+2)+\nu_1\,\omega_1\,(-5+5\,x_2)+\nu_2\,\omega_2\,(4+2\,x_2)-\lambda_1\,(32-96\,x_1^2)&=0\\
\beta_2\,(3\,x_2^2-6x_2+2)+\nu_1\,\omega_1\,(8+5\,x_1)+2\,\nu_2\,\omega_2\,x_1+8\,\lambda_1\,(2x_2+1)^3&=0\\
\nu_1+\nu_2&=0\\
x_1^3-3x_1^2+2x_1&=0\\
x_2^3-3x_2^2+2x_2&=0\\
\lambda_1\,(32\,x_1-32\,x_1^3-(2x_2+1)^4)&=0\\
\omega_1\,(-5\,x_1+8\,x_2+5\,x_1\,x_2+5)-\gamma&=0\\
\omega_2\,(4\,x_1+2\,x_1\,x_2)-\gamma&=0.
\end{aligned}$$

The set of solutions projected in the x is the whole set of feasible solutions.

Then, the candidate set of possible solutions of our problem is the set of non-regular solutions fulfilling the requirements of Theorem 4.1:

$$\{(2,2),(2,0),(0,0),(1,0),(2,1)\}.$$

Discarding dominated solutions, the solution set for this example is:

$$\{(2,0),(0,0),(1,0)\}.$$

Fig. 3 shows the feasible region, the nondominated solutions and the level curves for each of them in this example.

4.2. The Chebyshev–Fritz–John approach

Analogously to the previous approach, once we have scalarized the original multiobjective problem to a family of single objective problems, in this section we apply the Fritz–John (FJ) conditions to all the problems in this family. The following well-known result justifies the use of FJ conditions to obtain candidates to optimal solutions for single objective problems. Proofs and further details can be found in Bazaraa et al. (1993).

Theorem 4.4 (*FJ Necessary Conditions*)**.** *Consider the problem:*

$$\begin{array}{lll}
\min & f(x) & \\
s.t. & g_j(x)\le 0 & j=1,\ldots,m\\
 & h_r(x)=0 & r=1,\ldots,s\\
 & x\in\mathbb{R}^n. &
\end{array}\qquad (8)$$

Let x^* be a feasible solution, and let $J = \{j : g_j(x^*) = 0\}$. Suppose that f and g_j, for $j = 1, \ldots, m$, are differentiable at x^*, and that h_r, for $r = 1, \ldots, s$, is continuously differentiable at x^*. If x^* locally solves Problem (8), then there exist scalars λ_j, for $j = 1, \ldots, m$, and μ_r, for $r = 1, \ldots, s$, such that

$$\begin{aligned}
\lambda_0 \nabla f(x^*) + \sum_{j=1}^{m} \lambda_j \nabla g_j(x^*) + \sum_{r=1}^{s} \mu_r \nabla h_r(x^*) &= 0 \\
\lambda_j\, g_j(x^*) &= 0 \qquad \text{for } j = 1, \ldots, m \\
\lambda_j &\geq 0 \qquad \text{for } j = 1, \ldots, m \\
(\lambda_0, \lambda, \mu) &\neq (0, \mathbf{0}, \mathbf{0}).
\end{aligned} \tag{FJ}$$

Note that, in the FJ conditions, regularity conditions are not required to set the result.

Corollary 4.2. *Let x^* be a nondominated solution for ($MOPBP_{\mathbf{f,g,h}}$). Then, x^* is a solution of the system of polynomial equations (9) for some $\nu_i, \lambda_j, \mu_r, \beta_i$, for $i = 1, \ldots, k, j = 1, \ldots, m, r = 1, \ldots, s$ and $\omega > 0$.*

$$\left.\begin{aligned}
\lambda_0 - \sum_{i=1}^{k} \nu_i = 0 \\
\sum_{i=1}^{k} \nu_i \omega_i \nabla f_i(x) + \sum_{j=1}^{m} \lambda_j \nabla g_j(x) + \sum_{r=1}^{s} \mu_j \nabla h_r(x) + \sum_{i=1}^{n} \beta_i e_i\, (2x_i - 1) = 0 \\
\nu_i\, (\omega_i\, (f_i(x) - \hat{y}_i) - \gamma) = 0, \quad i = 1, \ldots, k \\
\lambda_j\, g_j(x) = 0, \quad j = 0, \ldots, m
\end{aligned}\right\} \tag{9}$$

where $\lambda_j \geq 0$, for $j = 1, \ldots, m$, $\nu_i \geq 0$, for $i = 1, \ldots, k$ and not all simultaneously zero.

Let X_E^{FJ} denote the set of solutions, in the x-variables, that are feasible solutions of ($MOPBP_{\mathbf{f,g,h}}$) and solutions of system (9).

The set of nondominated solutions of our problem is a subset of X_E^{FJ}, since every nondominated solution is an optimal solution for some problem in the form of (9), and every solution of this single objective problem is a solution of the FJ system.

However, dominated solutions may appear in the set of solutions of (9), so, a final elimination process is to be performed to select only the nondominated solutions.

The steps to solve ($MOPBP_{\mathbf{f,g,h}}$) using the Chebyshev–FJ approach are summarized in Algorithm 3.

Algorithm 3: Summary of the procedure for solving MOPBP using the Chebyshev–FJ approach.

Input : $f_1, \ldots, f_k, g_1, \ldots g_m, h_1, \ldots, h_r \in \mathbb{R}[x_1, \ldots, x_n]$
Algorithm:
Step 1 Formulate the Chebyshev scalarization of ($MOPBP_{\mathbf{f,g,h}}$). (Problem (P_ω))
Step 2 Solve system (9) in the x-variables for any value of $\omega > 0$: X_E^{FJ}.
Step 3 Remove from X_E^{FJ} the set of dominated solutions: X_E.

Output: X_E the set of nondominated solutions for Problem ($MOPBP_{\mathbf{f,g,h}}$)

Theorem 4.5. *Algorithm 3 solves ($MOPBP_{\mathbf{f,g,h}}$) in a finite number of steps.*

The last part of the section is devoted to showing how to solve the Chebyshev–FJ system using Gröbner bases.

Consider the following polynomial ideal

$$I = \Bigg\langle \lambda_0 - \sum_{i=1}^{k} \nu_i, \sum_{i=1}^{k} \nu_i \omega_i \nabla f_i(x) + \sum_{j=1}^{m} \lambda_j \nabla g_j(x) + \sum_{r=1}^{s} \mu_j \nabla h_r(x) + \sum_{i=1}^{n} \beta_i\, e_i\, (2x_i - 1),$$
$$\nu_1\, (\omega_1\, (f_1(x) - \hat{y}_1) - \gamma), \ldots, \nu_k\, (\omega_k\, (f_k(x) - \hat{y}_k) - \gamma), \lambda_1\, g_1(x), \ldots, \lambda_m\, g_m(x) \Bigg\rangle$$

in the polynomial ring $\mathbb{R}[\mathbf{x}, \gamma, \lambda, \nu, \mu, \beta]$.

Let us consider a lexicographical order over the monomials in $\mathbb{R}[\mathbf{x}, \gamma, \lambda, \nu, \mu, \beta]$ such that $\mathbf{x} \prec \gamma \prec \lambda \prec \nu \prec \mu \prec \beta$. Then, the Gröbner basis, $\mathcal{G}$, for I with this order has the following triangular shape:

- $\mathcal{G}$ contains one polynomial in $\mathbb{R}[x_n]$: $p_n(x_n)$
- $\mathcal{G}$ contains one or several polynomials in $\mathbb{R}[x_{n-1}, x_n] : p^1_{n-1}(x_{n-1}, x_n), \ldots, p^{m_1}_{n-1}(x_{n-1}, x_n)$

...

- $\mathcal{G}$ contains one or several polynomials in $\mathbb{R}[\mathbf{x}] : p^1_1(x_1, \ldots, x_n), \ldots, p^{m_n}_1(x_1, \ldots, x_n)$
- The remainder polynomials involve variables $\mathbf{x}$ and at least one of γ, λ, μ, ν or β.

We are interested in finding only the values for the x-variables, so, we avoid the polynomials in $\mathcal{G}$ that involve any of the other auxiliary variables. We denote by $\mathcal{G}^x$ the subset of $\mathcal{G}$ that contains only all the polynomials in the x-variables. By the Extension Theorem, $\mathcal{G}^x$ is a Gröbner basis for $I \cap \mathbb{R}[x_1, \ldots, x_n]$.

Solving the system given by $\mathcal{G}^x$, we obtain as solutions, those of our FJ original system.

The following example illustrates the how the algorithm works.

Example 4.2. Consider the following biobjective problem:

$$\begin{aligned} \min \quad & (-10x_2 + 3x_1x_2, 2x_1 + 4x_2 - 6x_1x_2) \\ s.t. \quad & -4\,x_2 + 132\,x_1^4 - 143\,x_1^3 + 40\,x_1 \le 0 \\ & \qquad\qquad x_1, x_2 \in \{0, 1, 2\}. \end{aligned}$$

After formulating the Chebyshev problem, the FJ system is:

$$\begin{aligned} \beta_1\,(3\,x_1^2 - 6x_1 + 2) - 4\,\nu_1\,\omega_1 - \nu_2\,\omega_2 - \lambda_1\,(528\,x_1^3 - 429\,x_1^2 + 40) &= 0, \\ \beta_2\,(3\,x_2^2 - 6x_2 + 2) + 10\,\nu_1\,\omega_1 - 4\,\nu_2\,\omega_2 + 4\,\lambda_1 &= 0, \\ \lambda_0 - \nu_1 - \nu_2 &= 0, \\ x_1(x_1 - 1)(x_1 - 2) &= 0, \\ x_2(x_2 - 1)(x_2 - 2) &= 0, \\ \lambda_1\,(-4\,x_2 + 132\,x_1^4 - 143\,x_1^3 + 40\,x_1) &= 0, \\ \nu_1\,(\omega_1\,(-10x_2 + 3x_1x_2 + 10) - \gamma) &= 0, \\ \nu_2\,(\omega_2\,(2x_2 + 4x_2 - 6x_1x_2) - \gamma) &= 0. \end{aligned}$$

Whose solutions using the convenient Gröbner basis are:

$$\{(1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (1, 1), (0, 0), (0, 1)\}.$$

Note that the solution $(0, 2)$ is automatically discarded since it is not a feasible solution of the original problem.

Now, by applying Theorem 4.1, the solutions $(0, 0)$ and $(0, 1)$ are discarded, and then, the candidate set of nondominated solutions is:

$$\{(1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (1, 1)\}.$$

Then, discarding dominated solutions we obtain the set of nondominated solutions:

$$\{(2, 2), (1, 2)\}.$$

Fig. 4 shows feasible region, the nondominated solutions and the level curves of Example 4.2.

Remark 4.1 (*Convex Case*)**.** In the special case where both objective functions and constraints are convex, sufficient KKT conditions can be applied. If the feasible solution x^* satisfies KKT conditions, and all objective and constraint functions are convex, then x^* is a nondominated solution. As a particular case, this situation is applicable to linear problems.

In this case, we may choose a linear scalarization instead of the Chebyshev scalarization. With this alternative approach, the scalarized problem is

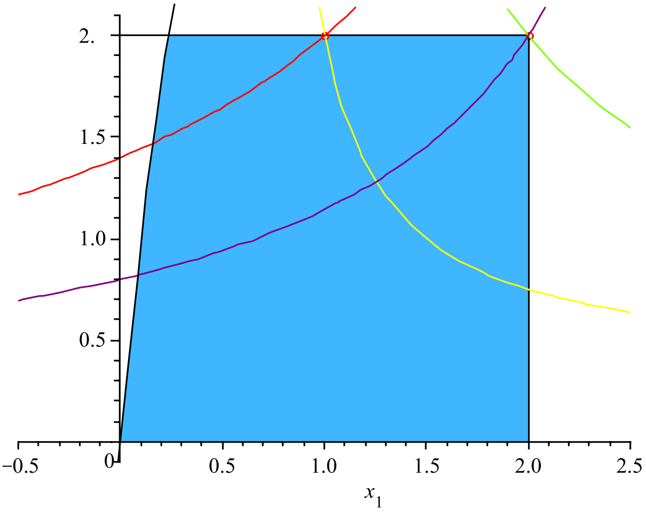

Fig. 4. Feasible region, the nondominated solutions and the level curves of Example 4.2.

$$\begin{array}{lrl} \min & \displaystyle\sum_{s=1}^{k} t_s f_s(x) & \\ s.t. & g_j(x) \leq 0 & j = 1, \ldots, m \\ & h_r(x) = 0 & r = 1, \ldots, s \\ & x_i(x_i - 1) = 0 & i = 1, \ldots, n \end{array}$$

for $t_1, \ldots, t_k > 0$.

Then, by Corollary 11.19 in Jahn (2004) applied to multiobjective binary problems, and denoting by S the feasible region, if $f(S) + \mathbb{R}^k_+$ is convex, then each x^* is a nondominated solution if and only if x^* is a solution of Problem 4.1 for some $t_1, \ldots, t_k > 0$.

Using both results, necessary and sufficient conditions are given for that problem and the removing step is avoided.

Remark 4.2 (*Single Objective Case*)**.** The same approach can be applied to solve single objective problems. In this case, KKT (or FJ) conditions can be applied directly to the original problem, without scalarizations.

5. Obtaining nondominated solutions by multiobjective optimality conditions

In this section, we address the solution of ($MOPBP_{\mathbf{f,g,h}}$) by directly applying necessary conditions for multiobjective problems. With these conditions we do not need to scalarize the problem, as in the above section, avoiding some steps in the process followed in the previous sections.

The following result states the Fritz–John necessary optimality conditions for multiobjective binary problems.

Theorem 5.1 (*Multiobjective FJ Necessary Conditions, Theorem 3.1.1 in Miettinen, 1999*)**.** *Consider the problem:*

$$\begin{array}{lrl} \min & (f_1(x), \ldots, f_k(x)) & \\ s.t. & g_j(x) \leq 0 & j = 1, \ldots, m \\ & h_r(x) = 0 & r = 1, \ldots, s \\ & x \in \{0, 1\}^n. & \end{array} \tag{10}$$

Let x^* be a feasible solution. Suppose that f_i, for $i = 1, \ldots, k$, g_j, for $j = 1, \ldots, m$ and h_r, for $r = 1, \ldots, s$, are continuously differentiable at x^*. If x^* is a nondominated solution for Problem (10) then *there exist scalars ν_i, for $i = 1, \ldots, k$, λ_j, for $j = 1, \ldots, m$, μ_r, for $r = 1, \ldots, s$, and β_i, for $i = 1 \ldots, n$, such that*

$$\begin{aligned}
\sum_{i=1}^{k} \nu_i \nabla f_i(x^*) + \sum_{j=1}^{m} \lambda_j \nabla g_j(x^*) + \sum_{r=1}^{s} \mu_r \nabla h_r(x^*) + \sum_{i=1}^{n} \beta_i e_i (2x_i - 1) &= 0 \\
\lambda_j g_j(x^*) &= 0 \quad \text{for } j = 1, \ldots, m \\
\lambda_j &\geq 0 \quad \text{for } j = 1, \ldots, m \\
\nu_i &\geq 0 \quad \text{for } i = 1, \ldots, k \\
(\nu, \lambda, \mu) &\neq (\mathbf{0}, \mathbf{0}, \mathbf{0}).
\end{aligned} \tag{MO-FJ}$$

We can apply this result directly to any MOPIP problem, once it is written in the form (2). Then, one must solve the system given by the necessary conditions to obtain candidates to be nondominated solutions for the original problem. For solving this system, we use lexicographical Gröbner bases as in the above sections. We summarize the algorithm for solving the multiobjective polynomial problem in Algorithm 4.

Algorithm 4: Summary of the procedure for solving MOPBP using the multiobjective FJ optimality conditions.

Input : $f_1, \ldots, f_k, g_1, \ldots g_m, h_1, \ldots, h_r \in \mathbb{R}[x_1, \ldots, x_n]$
Algorithm:
Step 1 Solve system (MO-FJ): X_E^{MOFJ}.
Step 2 Remove from X_E^{MOFJ} the subset of dominated solutions: X_E.
Output: X_E the set of nondominated solutions for Problem ($MOPBP_{\mathbf{f,g,h}}$)

The following simple example illustrates Algorithm 4.

Example 5.1.

$$\begin{aligned}
\min \quad & (x_1^2 + 10x_2, -4x_1 - x_2^3) \\
s.t. \quad & 16x_2 - 16x_1^3 + 16x_1^2 - 3x_1 - 8 \geq 0 \\
& x_1, x_2 \in \{0, 1\}.
\end{aligned}$$

Solving the problem reduces to solving the following system of polynomial equations (Theorem 5.1):

$$\begin{aligned}
2\nu_1 x_1 - 4\nu_2 + \lambda_1 (2x_1 - 1) - \mu_1 (-3x_1^2 + 2x_1 - 3/16) &= 0 \\
10\nu_1 - 3\nu_2 x_2^2 + \lambda_2 (2x_2 - 1) - \mu_1 &= 0 \\
\mu_1 (x_2 - x_1^3 + x_1^2 - (3/16)x_1 - 1/2) &= 0 \\
x_1^2 - x_1 &= 0 \\
x_2^2 - x_2 &= 0 \\
\lambda_1, \lambda_2 &\geq 0 \\
\nu_1, \nu_2 &\geq 0.
\end{aligned}$$

The solutions of the above system in the original variables, x_1 and x_2, are:

$$\{x_1 = 1, x_2 = 1\}, \quad \{x_1 = 0, x_2 = 1\}.$$

Fig. 5 shows this set of nondominated solutions in the feasible region and the level curves at these points of the problem.

Remark 5.1. In the special case where both objective functions and constraints are convex, Theorem 5.1 gives sufficient nondominance conditions for ($MOPBP_{\mathbf{f,g,h}}$) requiring that $\nu_i > 0$ (see Theorem 3.1.8 in Miettinen (1999)).

6. Computational experiments

A series of computational experiments have been performed in order to evaluate the behavior of the proposed solution methods. Programs have been coded in MAPLE 11 and executed in a PC with

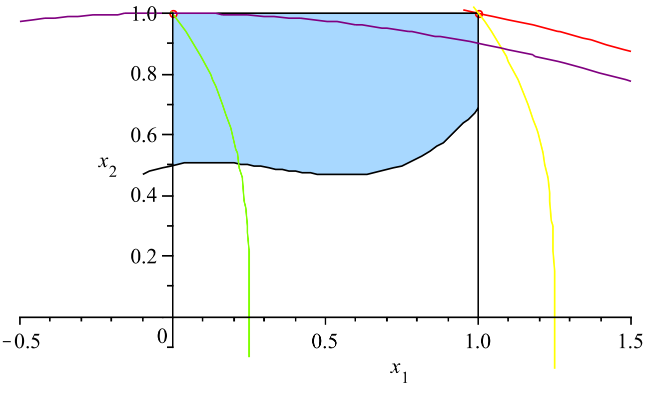

Fig. 5. Nondominated solutions and level curves of Example 5.1.

an Intel Core 2 Quad processor at 2× 2.50 GHz and 4 GB of RAM. The implementation has been done in that symbolic programming language, available upon request, in order to make the access easy to both optimizers and algebraic geometers.

We run the algorithms for three families of binary biobjective and triobjective knapsack problems: linear, quadratic and cubic, and for a biobjective portfolio selection model. For each problem, we obtain the set of nondominated solutions as well as the CPU times for computing the corresponding Gröbner bases associated to the problems, and the total CPU times for obtaining the set of solutions.

We give a short description of the problems where we test the algorithms. In all cases, we use binary variables $x_j, j = 1, \ldots, n$, where $x_j = 1$ means that the item (resp. security) j is selected for the knapsack (resp. portfolio) problem.

1. *Biobjective (linear) knapsack problem* (`biobj_linkn`): Assume that n items are given. Item j has associated costs q_j^1, q_j^2 for two different targets, and a unit profit $a_j, j = 1, \ldots, n$. The biobjective knapsack problem calls for selecting the item subsets whose overall profit ensures a knapsack with value at least b, so as to minimize (in the nondominance sense) the overall costs. The problem may be formulated:

$$\min \left(\sum_{j=1}^{n} q_j^1 x_j, \sum_{j=1}^{n} q_j^2 x_j \right)$$
$$s.t. \quad \sum_{i=1}^{n} a_i x_i \geq b, \quad x \in \{0, 1\}^n.$$

2. *Biobjective cubic knapsack problem* (`biobj_cubkn`): Assume that n items are given where item j has an integer profit a_j. In addition we are given two $n \times n \times n$ matrices $P^1 = (p_{ijk}^1)$ and $P^2 = (p_{ijk}^2)$, where p_{ijk}^1 and p_{ijk}^2 are the costs for each of the targets if the combination of items i, j, k is selected for $i < j < k$; and two additional $n \times n$ matrices $Q^1 = (q_{ij}^1)$ and $Q^2 = (q_{ij}^2)$, where q_{jj}^1 and q_{ij}^2 are the costs for the two different targets if both items i and j are selected for $i < j$. The biobjective cubic knapsack problem calls for selecting the item subsets whose overall profit exceeds the purpose of the knapsack b, so as to minimize the overall costs. The problem may be formulated:

$$\min \left(\sum_{i=1}^{n} \sum_{j=i}^{n} q_{ij}^1 x_i x_j + \sum_{i=1}^{n-2} \sum_{j=i+1}^{n-1} \sum_{l=j+1}^{n} p_{ijl}^1 x_i x_j x_l, \ \sum_{i=1}^{n} \sum_{j=i}^{n} q_{ij}^2 x_i x_j + \sum_{i=1}^{n-2} \sum_{j=i+1}^{n-1} \sum_{l=j+1}^{n} p_{ijl}^2 x_i x_j x_l \right)$$
$$s.t. \quad \sum_{i=1}^{n} a_i x_i \geq b, \quad x \in \{0, 1\}^n.$$

3. *Biobjective quadratic knapsack problem* (`biobj_qkn`): This problem may be seen as a special case of the biobjective cubic knapsack problem when there are no cost correlations between triplets.

4. *Triobjective (linear) knapsack problem* (`triobj_linkn`): Assume that n items are given where item j has an integer profit a_j. In addition, we are given three vectors $q^1 = (q_j^1), q^2 = (q_j^2)$ and $q^3 = (q_j^3)$, where q_j^1, q_j^2 and q_j^3 are the costs for three different targets if j is selected. The triobjective knapsack problem calls for selecting the item subsets whose overall profit ensures a profit for the knapsack at least b, so as to minimize (in the nondominance sense) the overall costs. The problem is:

$$\min \left(\sum_{j=1}^{n} q_j^1 x_j, \sum_{j=1}^{n} q_j^2 x_j, \sum_{j=1}^{n} q_j^3 x_j \right)$$
$$s.t. \sum_{i=1}^{n} a_i x_i \geq b, \quad x \in \{0, 1\}^n.$$

5. *Triobjective cubic knapsack problem* (`triobj_cubkn`): We are given n items where item j has an integer profit a_j. In addition, we are given three $n \times n \times n$ matrices $P^1 = (p_{ijk}^1), P^2 = (p_{ijk}^2)$ and $P^3 = (p_{ijk}^3)$, where p_{ijk}^1, p_{ijk}^2 and p_{ijk}^3 are the costs for each of the targets if the combination of items i, j and k is selected for $i < j < k$; and three additional $n \times n$ matrices $Q^1 = (q_{ij}^1), Q^2 = (q_{ij}^2)$ and $Q^3 = (q_{ij}^2)$, where q_{ij}^1, q_{ij}^2 and q_{ij}^3 are the costs for three different targets if both items i and j are selected for $i < j$. The triobjective cubic knapsack problem calls for selecting the item subsets whose overall profit ensures a value of b, so as to minimize the overall costs. The problem is:

$$\min \left(\sum_{i=1}^{n}\sum_{j=i}^{n} q_{ij}^1 x_i x_j + \sum_{i=1}^{n-2}\sum_{j=i+1}^{n-1}\sum_{l=j+1}^{n} p_{ijl}^1 x_i x_j x_l, \sum_{i=1}^{n}\sum_{j=i}^{n} q_{ij}^2 x_i x_j + \sum_{i=1}^{n-2}\sum_{j=i+1}^{n-1}\sum_{l=j+1}^{n} p_{ijl}^2 x_i x_j x_l, \right.$$
$$\left. \sum_{i=1}^{n}\sum_{j=i}^{n} q_{ij}^3 x_i x_j + \sum_{i=1}^{n-2}\sum_{j=i+1}^{n-1}\sum_{l=j+1}^{n} p_{ijl}^3 x_i x_j x_l \right)$$
$$s.t. \sum_{i=1}^{n} a_i x_i \geq b, \quad x \in \{0, 1\}^n.$$

6. *Triobjective quadratic knapsack problem* (`triobj_qkn`): This problem may be seen as a special case of the triobjective cubic knapsack problem when there are no cost correlations between triplets.
7. *Biobjective portfolio selection* (`portfolio`): Consider a market with n securities. An investor with initial wealth b seeks to improve his wealth status by investing it into these n risky securities. Let X_i be the random return per one lot of the ith security ($i = 1, \ldots, n$). The mean, $\mu_i = E[X_i]$, and the covariance, $\sigma_{ij} = Cov(X_i, X_j), i, j = 1, \ldots, n$, of the returns are assumed to be known. Let x_i be a decision variable that takes value 1 if the decision-maker invests in the ith security and 0 otherwise. Denote the decision vector by $x = (x_1, \ldots, x_n)$. Then, the random return for a inversion vector x from the securities is $\sum_{i=1} x_i X_i$ and the mean and variance of this random variable are $E[\sum_{i=1} x_i X_i] = \sum_{i=1}^{n} \mu_i x_i$ and $Var(\sum_{i=1} x_i X_i) = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j \sigma_{ij}$.
Let a_i be the current price of the ith security. Then, if an investor looks for minimizing his investment risk and simultaneously maximizing the expected return with that investment, the problem can be formulated as:

$$\left. \begin{array}{l} \min \left(Var\left(\sum_{i=1}^{n} x_i X_i\right), -E\left[\sum_{i=1}^{n} x_i X_i\right]\right) \\ s.t. \sum_{i=1}^{n} a_i x_i \leq b, \quad x \in \{0, 1\}^n \end{array} \right\} \Rightarrow \left. \begin{array}{l} \min \left(\sum_{i=1}^{n} \sigma_{ij} x_i x_j, -\sum_{i=1}^{n} \mu_i x_i \right) \\ s.t. \sum_{i=1}^{n} a_i x_i \leq b, \quad x \in \{0, 1\}^n \end{array} \right\}.$$

For each of the above 7 classes of problems, we consider instances randomly generated as follows: a_i is randomly drawn in $[-10, 10]$ and the coefficients of the objective functions, $q_{ij}^k, p_{ijl}^k, \sigma_{ij}$ and μ_i are in the range $[-10, 10]$. Once the constraint vector, $(a_1, \ldots, a_n)$, is generated, the right hand side, b,

Table 1
Computational results for biobjective knapsack problems.

prob	n	alg1			kkt			kkt_sl			fj			fj_sl			mofj			#nd
		gbt	tott	#vars	gbt	tott	#vars	gbt	tott	#vars	gbt	tott	#vars	gbt	tott	#vars	gbt	tott	#vars	
	2	0.02	0.07	5	0.30	0.48	11	0.35	0.61	12	0.24	0.39	10	0.31	0.50	11	0.03	0.06	7	1.8
	3	0.03	0.09	6	0.82	1.31	13	0.95	1.61	14	0.71	1.06	12	0.84	1.39	13	0.10	0.18	9	1.6
	4	0.07	0.23	7	2.45	3.89	15	3.05	4.90	16	2.07	3.10	14	2.54	4.12	15	0.33	0.52	11	3.6
	5	0.32	0.69	8	7.01	10.82	17	8.71	13.90	18	5.92	8.71	16	7.31	11.71	17	0.91	1.40	13	4.6
	6	2.86	4.11	9	25.00	37.88	19	32.29	50.13	20	20.63	30.11	18	26.66	41.70	19	3.01	4.61	15	5.4
biobj_linkn	7	31.15	35.59	10	72.85	107.15	21	82.90	127.08	22	55.68	79.80	20	66.77	104.82	21	7.89	11.60	17	5
	8	342.10	373.72	11	176.94	261.58	23	209.68	322.58	24	133.78	193.54	22	165.97	262.46	23	18.35	27.60	19	4.2
	9	5273.56	6382.43	12	462.14	675.24	25	529.50	813.54	26	333.81	492.89	24	418.75	670.82	25	48.50	79.08	21	8
	10																269.19	404.04	23	7.8
	11																480.26	835.46	25	6.4
	12																1340.31	2004.33	27	5.4
	13																4091.92	19546.05	29	11
	2	0.03	0.07	5	0.28	0.47	11	0.37	0.64	12	0.23	0.36	10	0.31	0.51	11	0.04	0.07	7	1.4
	3	0.04	0.10	6	0.79	1.29	13	1.12	1.86	14	0.68	1.04	12	0.95	1.54	13	0.09	0.16	9	2
	4	0.19	0.37	7	3.27	4.97	15	4.63	7.15	16	2.69	4.01	14	3.89	5.95	15	0.49	0.70	11	2.4
	5	1.76	2.37	8	9.22	13.88	17	11.94	18.06	18	7.54	10.94	16	9.90	15.11	17	1.08	1.74	13	3.8
	6	21.43	23.22	9	25.21	37.33	19	33.74	50.13	20	20.57	29.46	18	27.60	41.86	19	2.92	4.58	15	3.6
biobj_qkn	7	425.13	430.43	10	59.57	91.29	21	85.56	129.88	22	49.38	72.81	20	69.20	107.76	21	5.93	9.71	17	4.4
	8				172.26	255.22	23.00	228.66	337.02	24.00	127.43	188.32	22.00	174.23	272.72	23.00	14.75	25.66	19.00	5.4
	9				463.39	692.25	25	641.29	939.99	26	350.34	515.07	24	489.31	755.88	25	39.90	65.78	21	6.6
	10																138.11	255.42	23.00	7
	11																331.34	643.90	25	9
	12																891.46	1833.75	27	8
	3	0.17	0.21	7	0.72	1.14	13	0.97	1.56	14	0.62	0.92	12	0.82	1.31	13	0.09	0.15	9	1.8
	4	1.90	2.00	8	4.68	6.73	15	6.26	8.98	16	3.86	5.37	14	5.12	7.38	15	0.56	0.83	11	1.8
	5	7.26	7.48	9	9.29	13.43	17	12.49	17.79	18	7.86	10.81	16	10.31	14.91	17	1.11	1.65	13	2.8
	6	205.52	206.25	10													6.11	8.29	15	2.2
biobj_cubkn	7	2067.24	2069.38	11													11.25	15.24	17	6.6
	8																24.98	32.25	19	5.2
	9																81.80	106.90	21	6.2
	10																258.68	389.95	23	5.8
	11																690.43	916.80	25	7.8

Table 2
Information about all the algorithms.

Algorithm	#var	#gen	maxdeg
alg1	$2n+k+m+s$	$n+k+m+s$	$\max\{2, deg(f), deg(g), deg(h)\}$
kkt	$2n+2k+m+s+1$	$2n+k+m+s+1$	$\max\{deg(f)+2, deg(g)+1, deg(h)\}$
nr	$2n+2k+m+s+1$	$2n+m+s$	$\max\{deg(f)+1, deg(g), deg(h)\}$
fj	$2n+2k+m+s+2$	$2n+k+m+s+1$	$\max\{deg(f)+2, deg(g)+1, deg(h)\}$
mojf	$2n+k+m+s$	$2n+m+s$	$\max\{deg(f), deg(g)+1, deg(h)\}$

is randomly generated in $[1, |\sum_{i=1}^{n} a_i|]$. For each type of instances and each value of n in [2, 13] we generated 5 instances.

Tables 1 and 3 contain a summary of the average results over the different instances generated for the above problems. Each algorithm is labeled conveniently: alg1 corresponds to Algorithm 1, kkt is Algorithm 2, kkt_sl is Algorithm 2 where the inequality is transformed to an equation using a slack variable, fj is Algorithm 3, fj_sl is Algorithm 3 where the inequality is transformed to an equation using a slack variable and mofj stands for Algorithm 4. For each of these algorithms we present the CPU time for computing the corresponding Gröbner basis (tgb), the total CPU time for obtaining the set of nondominated solutions (ttot), the number of nondominated solutions (#nd) and the number of variables involved in the resolution of the problem (#vars).

From those tables, the reader may note that Algorithm 1 is faster than the others for the smallest instances, although the CPU times for this algorithm increase faster than those for the others and it is not able to obtain solutions when the size of the problem is around 12 variables. The algorithms based on Chebyshev scalarization (kkt, kkt_sl, fj and fj_sl) are better than alg1 for the largest instances. The differences between these four methods are meaningful, but the algorithms based on the KKT conditions are, in almost all the instances, faster than those based on the FJ conditions. Note that considering slack variables to avoid the inequality constraint is not better, since the CPU times when the slack variable is considered are larger. Finally, the best algorithm, in CPU time, is mofj since except for the small instances it is the fastest and it was able to solve larger instances.

One may think that the last step of our methods, i.e. removing dominated solutions, should be more time consuming in alg1 than in the remaining methods since alg1 does not use optimality conditions. However, from our experiments this conclusion is not clearly supported. Actually, although this process is time consuming, when the dimension of the problem increases this time is rather small compared with the effort necessary to obtain the Gröbner bases.

Table 2 shows some information about each of the presented algorithms. For a multiobjective problem with n variables, m polynomial inequality constraints given by $\mathbf{g} = (g_1, \ldots, g_m)$, s polynomial equality constraints given by $\mathbf{h} = (h_1, \ldots, h_s)$ and k objectives functions given by $\mathbf{f} = (f_1, \ldots, f_k)$, Table 3 shows the number of variables (#var), the number of generators (#gen) and the maximal degrees (maxdeg) of the initial polynomial ideals related to each of the algorithms. These numbers inform us about the theoretical complexity of the algorithms. Here, by theoretical complexity we mean the complexity of computing the lexicographic Gröbner basis, that basically, depends of the number of variables, the number of equations in the system and the maximal degree of the polynomials involved in the system (some complexity bounds for this computation involving #var, #gen and maxdeg can be found in Dubé et al. (1986)).

From the above table the reader may note that both alg1 and mojf have the same number of variables in any case, but the number of initial generators for alg1 is, in general, smaller than the same number for mojf, since the number of objectives is usually smaller than the number of variables. Furthermore, maximal degrees are smaller in alg1 than in mojf . However, in practice, mojf is faster than alg1 since using optimality conditions helps in identifying nondominated solutions.

7. Conclusions

In this paper we present several methodologies, based on the use of Gröbner bases, to solve multiobjective polynomial integer problems based on solving systems of polynomial equations

Table 3
Computational results for triobjective knapsack and biobjective portfolio problems.

prob	n	alg1			kkt			kkt_sl			fj			fj_sl			mofj			#nd
		gbt	tott	#vars	gbt	tott	#vars	gbt	tott	#vars	gbt	tott	#vars	gbt	tott	#vars	gbt	tott	#vars	
triobj_linkn	2	0.03	0.08	6	0.82	1.27	13	0.99	1.61	14	0.75	1.16	12	0.88	1.47	13	0.04	0.06	8	1.6
	3	0.02	0.10	7	2.98	4.33	15	3.63	5.51	16	4.19	5.62	14	4.95	7.00	15	0.14	0.20	10	2
	4	0.06	0.65	8	14.71	20.07	17	18.03	25.22	18	19.93	25.61	16	23.62	31.94	17	0.73	0.90	12	2.8
	5	0.71	1.00	9	40.75	56.40	19	48.20	68.21	20	55.31	72.04	18	65.38	87.07	19	2.70	3.62	14	4.2
	6	2.44	3.32	10	68.69	102.22	21	84.95	129.12	22	107.04	147.24	20	130.13	185.43	21	2.26	3.36	16	6.2
	7	22.35	25.32	11	188.86	276.15	23	219.78	335.08	24	272.32	374.25	22	326.76	470.51	23	5.89	8.59	18	11.4
	8	360.38	367.76	12	501.35	731.46	25	576.07	856.28	26	729.93	993.07	24	830.08	1194.75	25	17.16	23.83	20	4.4
	9																69.68	115.30	22	24.4
	10																193.15	301.04	24	31.4
	11																529.51	1075.93	26	26
	12																1422.70	3145.30	28	85.2
triobj_qkn	2	0.04	0.09	6	0.96	1.52	13	1.28	1.96	14	0.88	1.37	12	1.19	1.81	13	0.05	0.08	8	2.2
	3	0.06	0.18	7	3.20	4.72	15	4.11	6.13	16	4.67	6.25	14	5.21	7.42	15	0.14	0.20	10	2.8
	4	0.09	0.32	8	6.50	9.89	17	9.03	13.82	18	9.59	13.14	16	11.44	16.59	17	0.28	0.45	12	5.2
	5	2.87	3.68	9	21.05	30.87	19	30.61	45.61	20	30.93	41.79	18	39.81	56.92	19	0.79	1.26	14	8.6
	6	31.76	33.69	10	49.11	72.24	21	72.03	106.16	22	69.31	91.93	20	86.83	122.30	21	2.19	3.85	16	9.4
	7	1099.24	1109.07	11	152.63	224.47	23	223.51	326.03	24	232.64	319.26	22	296.38	417.00	23	5.18	8.14	18	12.6
	8				481.73	709.09	25	721.64	1078.26	26	700.63	980.21	24	904.31	1289.78	25	14.81	22.22	20	17.6
	9																44.36	67.14	22	13.6
	10																119.50	213.66	24	17.8
	11																343.53	793.88	26	29.8
	12																1018.35	2445.64	28	40.2
triobj_cubkn	3	0.05	0.15	7	2.74	4.01	15	4.07	5.88	16	4.25	5.55	14	5.33	7.29	15	0.11	0.19	10	3.4
	4	0.18	0.43	8	9.97	13.81	17	13.14	18.50	18	14.32	18.31	16	17.91	23.90	17	0.42	0.56	12	4.4
	5	1.37	1.82	9													1.03	1.43	14	8.4
	6	28.61	29.80	10													2.93	4.19	16	6
	7																9.57	12.67	18	7.6
	8																30.95	37.29	20	13.4
	9																92.52	115.68	22	25.4
	10																371.98	447.80	24	29.4
	11																1006.29	1593.48	26	37.4
portfolio	2	0.03	0.14	5	0.43	0.73	11	0.68	1.10	12	0.40	0.65	10	0.60	0.94	11	0.06	0.08	7	2.0
	3	0.03	0.08	6	1.40	2.09	13	1.65	2.59	14	1.08	1.64	12	1.42	2.28	13	0.18	0.28	9	2.0
	4	0.08	0.38	7	3.50	5.34	15	4.56	7.08	16	2.85	4.17	14	3.74	5.95	15	0.46	0.71	11	4.0
	5	0.82	1.33	8	9.41	14.46	17	11.87	18.16	18	7.85	11.61	16	9.76	15.15	17	1.16	1.84	13	5.4
	6	17.39	18.73	9	25.28	38.45	19	32.75	50.01	20	20.92	30.43	18	26.76	41.96	19	2.90	4.71	15	6.8
	7	179.84	184.29	10	74.90	111.93	21	95.74	143.34	22	59.19	85.91	20	75.66	117.77	21	7.40	12.09	17	6.2
	8	4739.76	4749.91	11	132.97	201.62	23	168.92	255.92	24	108.75	161.05	22	137.98	217.42	23	15.16	25.23	19	8.0
	9				393.58	606.13	25	580.04	886.54	26	311.33	490.41	24	455.34	742.80	25	35.83	76.77	21	11.4
	10				1212.90	1837.36	27	1601.12	2440.86	28	869.72	1379.88	26	1207.49	1982.07	27	101.17	221.92	23	12.2
	11																305.80	724.28	25	15.6

derived from optimality conditions of different transformations of the original problem. The use of Gröbner bases is justified as a instrumental tool to solve the above mentioned systems of equations, although alternative methods for solving those systems would also lead to the same solutions of the multiobjective problem.

The first algorithm enumerates the images, under the objective functions, of the feasible values of the problem, and then, selects the minimal elements (with respect to the componentwise order) among all of them. The last three algorithms use necessary nondominance conditions that transform the original problem to a system of polynomial equations and inequalities. These conditions are, in general, necessary but not sufficient, and then by using them we may obtain solutions that are not nondominated. An additional test is used to discard these solutions. In general, the use of these methods cannot avoid the full enumeration of feasible solutions in the worst case since it is easy to construct instances where any feasible solution is nondominated. Therefore, a worst case analysis lead us to the complete enumeration of the feasible region (and then similar to a brute force approach). Nevertheless, in practice, these systems of equations reduce significantly the complete enumeration.

In addition, the obtained polynomial systems have finite number of solutions in the original variables, which induces zero-dimensional projected ideals and therefore simplifies the computation of Gröbner bases. Actually, it is known that for several classes of functions involved in the problem, the solutions of systems (6), (7) and (9) are directly the sets of nondominated solutions. Also, in those cases where the nondominance conditions are also sufficient, such as for instance under convexity hypothesis, the problem reduce to solving the system of polynomial equations and no additional tests are required.

Furthermore, from a methodological point of view this paper proposes, for the first time, a general methodology for exactly solving multiobjective polynomial integer problems. Scanning the literature in the field, there are only heuristic algorithms for some biobjective versions of specific problems, as quadratic or cubic knapsack or assignments problems (see Jahn (2004) for further details and related problems). Our results can be also used as certificates of nondominance in the above mentioned heuristics which will improve the quality of the solutions provided.

We have implemented our algorithms in MAPLE, using the standard package `Groebner` for obtaining the Gröbner bases, to compare the performance of the different approaches and even with our simple implementation we were able to solve problems up to 13 integer variables. Of course, different implementations using more sophisticated tools for the Gröbner basis computation will enlarge the size of the solved instances.

Finally, this paper also gives a first analysis of the theoretical and practical complexities of the algorithms. Note that theoretical complexity refers to the complexity of computing the lexicographic Gröbner basis. Our Table 2 shows these "indices" of the complexity of the four algorithms. Moreover, in order to analyze the practical complexity of the algorithms, we ran several biobjective and triobjective problems from four different families of well-known problems. From the computational tests we can conclude that Algorithm 4, the one based on the multiobjective Fritz–John nondominance conditions, is the one that is capable of solving largest instances and in less CPU time. That seems to be explained by the smaller number of indeterminates involved in the polynomial transformation of the multiobjective problem. The main drawback of Algorithm 1 is that it needs to select the minimal elements among the images by the objective functions of all the feasible solutions of the problem, which seems to consume a lot of CPU time. On the other hand, the main disadvantage of Algorithms 2 and 3 is the instrumental use of the Chebyshev scalarization. This scalarization adds an additional indeterminate γ, and k extra polynomial equations to the system which increases its difficulty.

References

Bazaraa, M., Sherali, H., Shetty, C., 1993. Nonlinear Programming: Theory and Algorithms. John Wiley and Sons, New York, NY.

Beasley, J., Meade, N., Chang, T., 1995. An evolutionary heuristic for the index tracking problem. European Journal of Operational Research 148, 621–643.

Beck, A., Teboulle, M., 2000. Global optimality conditions for quadratic optimization problems with binary constraints. SIAM Journal on Optimization 11 (1), 179–188.

Bertsimas, D., Perakis, D., Tayur, S., 2000. A new algebraic geometry algorithm for integer programming. Management Science 46 (7), 999–1008.

Blanco, V., Puerto, J., 2009. Partial Gröbner bases for multiobjective integer linear optimization. SIAM Journal on Discrete Mathematics 23 (2), 571–595.

Bretthauer, K., Shetty, B., 1995. The nonlinear resource allocation problem. Operations Research 43, 670–683.

Buchberger, B., 1965. An algorithm for finding a basis for the residue class ring of a zero-dimensional polynomial ideal. Ph.D. Thesis. University of Innsbruck, Institute for Mathematics.

Conti, P., Traverso, C., 1991. Buchberger algorithm and integer programming. In: AAECC.

Cooper, M., 1981. A survey of methods for pure nonlinear integer programming. Management Science 27, 353–361.

Cox, D., Little, J., O'Shea, D., 2005. Using Algebraic Geometry, 2nd ed. Springer.

Cox, D., Little, J., O'Shea, D., 2007. Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, 3/e (Undergraduate Texts in Mathematics). Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Dubé, T., Mishra, B., Yap, C., 1986. Complexity of Buchberger's algorithm for Gröbner bases. Extended Abstract. New York University.

Ehrgott, M., Gandibleux, X., 2004. Approximative solution methods for multiobjective combinatorial optimization. TOP 12 (1), 1–63.

Hägglöf, K., Lindberg, P., Svensson, L., 1995. Computing global minima to polynomial optimization problems using gröbner bases. Journal of Global Optimization (Historical Archive) 7 (2), 115–125.

Hoşten, S., Sturmfels, B., 1995. GRIN: an implementatiion of gröbner bases for integer programming. In: IPCO: 4th Integer Programming and Combinatorial Optimization Conference.

Jahn, J., 2004. Vector Optimization: Theory, Applications and Extensions. Springer.

Jobst, N., Horniman, M., Lucas, C., Mitra, G., 2001. Computational aspects of alternative portfolio selection models in the presence of discrete asset choice constraints. Quantitative Finance 1, 489–501.

Knowles, J., Corne, D., 2002. Towards landscape analyses to inform the design of hybrid local search for the multiobjective quadratic assignment problem. In: Abraham, A., del Solar, J.R., Koppen, M. (Eds.), Soft Computing Systems: Design, Management and Applications. IOS Press, Amsterdam, ISBN: 1-58603-297-6.

Lasserre, J., 2008. A semidefinite programming approach to the generalized problem of moments. Mathematical Programming 112 (1), 62–92.

Laughhunn, D., 1970. Quadratic binary programming with application to capital-budgeting problems. Operations Research 18, 454–461.

Li, D., Sun, X., 2006. Nonlinear Integer Programming. In: International Series in Operations Research & Management Science, Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Li, H., Landa-Silva, D., 2009. An elitist grasp metaheuristic for the multi-objective quadratic assignment problem. Lecture Notes in Computer Science 5647, 481–494.

Miettinen, K., 1999. Nonlinear Multiobjective Optimization. Kluwer Academic Publishers, Boston.

Ralphs, T., Saltzman, M., Wiecek, M., 2006. An improved algorithm for solving biobjective integer programs. Annals of Operations Research 147 (1), 43–70.

Ryoo, H.S., Sahinidis, N.V., 1995. Global optimization of nonconvex NLPs and MINLPs with applications in process design. Computers and Chemical Engineering 19, 551–566.

Scholz, D., 2010. The multicriteria big cube small cube method. TOP 18 (1), 286–302.

Sturmfels, B., 2002. Solving systems of polynomial equations, Tech. Rep. 97, CBMS Regional Conference Series of the AMS, Rhode Island.

Thomas, R., 1998. Applications to integer programming. In: PSAM: Proceedings of the 53th Symposium in Applied Mathematics. American Mathematical Society.

Vianna, D., Arroyo, J., 2004. A grasp algorithm for the multi-objective knapsack problem. In: The First International Conference on Quantitative Evaluation of Systems. IEEE Computer Society, Los Alamitos.