Multi-Objective Linear Integer Programming Based in Test Sets

M. I. Hartillo-Hermoso*[1], H. Jiménez-Tafur†[2], and J.M. Ucha-Enríquez ‡[1]

[1]Dpto. Matemática Aplicada I, Universidad de Sevilla, Spain
[2]Dpto. de Matemáticas. Universidad Pedagógica Nacional, Bogotá, Colombia

Abstract

We introduce a new exact algorithm for Multi-objective Linear Integer problems based on the classical ϵ-constraint method and algebraic test sets computed with Gröbner bases. Our method takes advantage of test sets 1) to identify which IPs have to be solved in an ϵ-constraint framework and 2) using reduction with test-sets instead of solving with an optimizer. We show that the computational results are promising in some families of examples.

1 Introduction

Problems in the real world involve multiple objectives. Due to conflict among these objectives, finding a feasible solution that simultaneously optimizes all objectives is often impossible. As decision makers usually need a complete knowledge of the best decisions they can take from those different points of view, generating the set of *efficient solutions* (i.e., solutions for which it is impossible to improve the value of one objective without worsening the value of at least one other objective) is a primary goal in multi-objective optimization. Multi-objective Integer Programming (MOIP) is the branch that deals with this kind of problem in the case of integer variables, and the linear case (MOILP) is the one in which we will concentrate.

Generation methods compute the whole space of Pareto optimal solutions. Among these type of methods, we have the *weighted sum* of objectives approach and the ϵ*-constraint technique*, that generates a grid in the objective space with ranges between the costs of *ideal* and *nadir* points. In ϵ-constraint methods, for each point in the upper bound set (cf. [9, 5]) a single-objective problem is solved, avoiding incremental movements through the grid.

In [14, 10, 15, 19, 13] different approaches to apply this ϵ-constrained setting in MOLIP can be found. Two additional algebraic approaches to MOIP have been presented: the one proposed in [3], that introduces the so called *partial Gröbner bases*, and [4] that generalized for several cost functions the ideas presented in [1] for single-objective problems. Unfortunately these two algebraic proposals can not manage big examples, to the best of our knowledge.

Our approach is based on the so-called *test sets* associated to single-objective Linear Integer Programming problems (LIP), taking advantage of their special characteristics. A test set is a set of directions that guides the movement from any feasible point until the optimum of the LIP is reached. So LIPs are solved by *reduction* with these test sets, instead of passing them to an optimizer. It is proved in [18] that *Gröbner bases* provide the minimal test set for a fixed total ordering compatible with the linear cost function of the considered program. These test sets do not depend on the right hand sides (RHS) of the constraints. Interested readers can consult the references [17, 2].

*Email: hartillo@us.es
†Email: hjimenezt@pedagogica.edu.co
‡Email: ucha@us.es

We will show how our method takes advantage of the features of test sets to manage the ϵ-constraint setting efficiently: most of the typical redundant computations are circumvented and we only provide new efficient solutions. Although the computation of Gröbner bases can be a hard task, very sensitive to the number of variables (cf. [16]) in our experiments the algorithm is fairly competitive in the *unbounded knapsack problem*.

This paper is a generalization of a previous work of the authors ([12]) for the biobjective case.

2 Preliminaries

A multi-objective linear integer optimization problem (MOLIP) in standard form can be stated as

$$\begin{array}{rl} \min & c_1(\mathbf{x}), \ldots, c_p(\mathbf{x}) \\ \text{s.t.} & A\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \in \mathbb{Z}^n_{\geq 0} \end{array} \tag{1}$$

for $A \in \mathbb{Z}^{m \times n}$, $\text{rank}(A) = m$, $\mathbf{b} \in \mathbb{Z}^m$ and $c_1, \ldots, c_p$ with $p \geq 2$ linear functions with integer coefficients. In general there is no feasible point that minimizes all the cost functions, so we are interested in obtaining the *efficient points*, that is those feasible points $\mathbf{x}^\star$ such that there is no feasible $\mathbf{x}$ with $c_k(\mathbf{x}) \leq c_k(\mathbf{x}^\star)$ with at least one strict inequality for $k = 1, \ldots, p$. If $\mathbf{x}^\star$ is an efficient point, $(c_1(\mathbf{x}^\star), \ldots, c_p(\mathbf{x}^\star))$ is a *non-dominated (or Pareto) point* in the decision space. If we replace the condition $c_k(\mathbf{x}) \leq c_k(\mathbf{x}^\star)$ for $c_k(\mathbf{x}) < c_k(\mathbf{x}^\star)$ we obtain *weakly efficient* points. We will denote $\mathcal{X}$ the set of efficient points and $\mathcal{N}$ the set of non-dominated points, the *Pareto frontier*.

We will assume that the feasible region for problem (1) is finite, so the Pareto frontier $\mathcal{N}$ is finite as well. In this paper we present an algorithm to obtain a set $\mathcal{X}^\star \subset \mathcal{X}$ that is a *minimal complete set of efficient points* (that is, if $\mathbf{x}^a, \mathbf{x}^b \in \mathcal{X}^\star$ then $(c_1(\mathbf{x}^a), \ldots, c_p(\mathbf{x}^a)) \neq (c_1(\mathbf{x}^b), \ldots, c_p(\mathbf{x}^b))$ and $|\mathcal{X}^\star| = |\mathcal{N}|$, as in [8])

The *ϵ-constraint technique*, (see [11]), one of the best known techniques to address problem (1), manages many problems of the form

$$\begin{array}{rl} \min & c_k(\mathbf{x}) \\ \text{s.t.} & A\mathbf{x} = \mathbf{b} \\ & c_j(\mathbf{x}) \leq \epsilon_j, \; j = 1, \ldots, p \; (j \neq k) \\ & \mathbf{x} \in \mathbb{Z}^n_{\geq 0} \end{array} \tag{2}$$

for fixed $k = 1, \ldots, p$ and suitable values of ϵ_j in order to solve Problem (1). Optimal points of Problem 2 are always weakly efficient. Furthermore we can identify the efficient solutions, as the following theorem of [8] states:

Theorem 1. *A feasible solution $\mathbf{x}^\star$ of a linear MOIP is efficient if and only if there exists a $(\epsilon_1, \ldots, \epsilon_p) \in \mathbb{R}^p$ such that $\mathbf{x}^\star$ is an optimal solution of the corresponding problems (2) for $k = 1, \ldots, p$.*

Thus we have families of IPs for which only the right hand side (RHS) varies, so it is natural to consider at this point one algebraic tool called the *test set* of a given LIP. Given the family of LIPs in standard form (no inequalities)

$$\begin{array}{rl} \min & c(\mathbf{x}) \\ \text{s.t.} & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \in \mathbb{Z}^n_{\geq 0} \end{array} \tag{3}$$

for $A \in \mathbb{Z}^{m \times n}$, $\text{rank}(A) = m$, $\mathbf{b} \in \mathbb{Z}^m$ and c a linear function with coefficients in $\mathbb{Z}^n$, in general there is not only one optimal point but several ones with the same cost. We can refine the cost function considering a total order $\prec_c$ that first compares two points by the cost c and breaks ties according to a chosen *term order* $\prec$ (see [6]). If we consider problem (3) replacing the cost function by $\prec_c$, it does not affect the optimal value but, as it is a total order, it insures a unique optimum.

Definition 2. *A* test set with respect to $\prec_c$ *of the family of problems (3) for fixed A is a set* $\mathcal{T} \subset \{\mathbf{t} \in \mathbb{Z}^n : A\mathbf{t} = 0\}$ *valid for any RHS, with the following properties:*

1. *For any feasible, non-optimal solution* $\mathbf{x}$ *of (3) for some* $\mathbf{b}$*, there exists* $\mathbf{t} \in \mathcal{T}$ *such that* $\mathbf{x} - \mathbf{t}$ *is feasible and* $\mathbf{x} - \mathbf{t} \prec_c \mathbf{x}$*.*

2. *Given the optimal solution* $\mathbf{x}^\star$ *of (3) for some* $\mathbf{b}$*, we have that* $\mathbf{x}^\star - \mathbf{t}$ *is not feasible for any* $\mathbf{t} \in \mathcal{T}$*.*

There exists a test set for any given LIP that can be computed with Gröbner bases with respect to $\prec_c$ ([18]). The existence of test sets for an LIP implies a straightforward algorithm to find its optimum: we start from any feasible point and subtract elements of the testset as long as we obtain feasible points. We will refer to this process as *reduction* of a feasible point with the test set.

So given the family of problems (2) for a fixed k, using test sets to solve them requires only 1) the computation of *one* test set for *all* the problems and 2) the reduction of a feasible point of each problem with the test set. It is very important to underline that, if the test set is available, the reduction process is very often faster than passing the IP to an optimizer. In addition, we will see that test sets guide us during the task of choosing which values of ϵ_j produce new efficient solutions, avoiding many redundant LIPs to be solved. At last, in contrast with several methods that compute first weakly efficient solutions and filter them in a second step, we will see that using a suitable total order we obtain efficient points directly.

3 Characterization of efficient points using test sets

To solve the problem (1) we will adopt a recursive scheme. We will obtain a minimal set of efficient solution of the problems

$$\begin{array}{ll} \min & c_1(\mathbf{x}), \ldots, c_i(\mathbf{x}) \\ \text{s.t.} & A\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \in \mathbb{Z}^n_{\geq 0} \end{array} \tag{4}$$

for $i = 2, \ldots, p$ and for this purpose we will use the ϵ-constraint method and manage the problems $P_i(\epsilon_1, \ldots, \epsilon_{i-1})$ (in standard form)

$$\begin{array}{ll} \min & c_i(\mathbf{x}) \\ \text{s.t.} & A\mathbf{x} = \mathbf{b} \\ & c_1(\mathbf{x}) + r_1 = \epsilon_1, \\ & \vdots \\ & c_{i-1}(\mathbf{x}) + r_i = \epsilon_{i-1}, \\ & \mathbf{x} \in \mathbb{Z}^n_{\geq 0}, \end{array} \tag{5}$$

for $i = 1, \ldots, p$ and $(\epsilon_1, \ldots, \epsilon_{i-1}) \in \mathbb{R}^{i-1}$.

For a given $i, 2 \leq i \leq p$, let us note $\prec_{\hat{c}_i}$ the total order that first compares two feasible points with respect to c_i and to break ties uses successively $c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_p$ and finally a chosen term order $\prec$ if it were necessary. We will denote $\mathcal{T}_{\hat{i}} \subset \mathbb{Z}^{n+(i-1)}$ the test-set for problem (5) with respect to the total order $\prec_{\hat{c}_i}$ (the elements of this test set have $i-1$ additional variables because of the slack variables added to the problem to put it in standard form).

The following result provides a characterization of the efficient points in this context.

Theorem 3. *A feasible point* $(\mathbf{x}^*, \mathbf{0}) \in \mathbb{Z}^{n+(p-1)}$ *is the optimal solution of* $P_p(c_1(\mathbf{x}^\star), \ldots, c_{p-1}(\mathbf{x}^\star))$ *with respect to the total order* $\prec_{\hat{c}_p}$ *if and only if* $\mathbf{x}^\star$ *is an efficient solution of (4) and among the ones with costs* $(c_1(\mathbf{x}^\star), \ldots, c_{p-1}(\mathbf{x}^\star))$ *is the smallest one with respect to* $\prec_{\hat{c}_p}$

Corollary 4. *If* $(\mathbf{x}^*, \mathbf{t}) \in \mathbb{Z}^{n+(p-1)}$ *con* $\mathbf{t} \geq 0$ *is the optimal solution of* $P_p(\epsilon)$ *for some* $\epsilon \in \mathbb{Z}^{n+(p-1)}$ *with respect to* $\prec_{\hat{c}_p}$ *then* $\mathbf{x}^\star$ *is an efficient solution of (1).*

Theorem 3 provides in particular a way to obtain the first point of our set of representatives of the non-dominated set of points of problem (1), the one with minimum c_1:

Corollary 5. *[8, Lemma 5.2.] Let* $\mathbf{x}_1^\star$ *be the optimal solution of*

$$\min\{c_1(\mathbf{x}) : A\mathbf{x} = \mathbf{b}, \mathbf{x} \in \mathbb{Z}^n_{\geq 0}\}$$

with respect to the ordering $\prec_{\hat{c}_1}$*. Then* $\mathbf{x}_1^\star$ *is an efficient solution of (1) with minimum cost* c_1*.*

4 Recursive construction of a minimal set of efficient solutions

Given the set of efficient solutions $\mathcal{X}$, let us denote $\mathcal{X}^\star \subset \mathcal{X}$ the minimal complete set of efficient points whose elements have the property of being the smallest ones with respect to $\prec_{\hat{c}_p}$ among the points that have the same costs. So by definition there is one efficient solution corresponding to each element in the Pareto frontier $\mathcal{N}$. The next result show how the elements $\mathcal{X}^\star$ can be obtained:

Theorem 6. *Let* $\mathbf{x} \in \mathcal{X}^\star$*. Then one of the following statements is true:*

1. $c'(\mathbf{x}^*) = (c_1(\mathbf{x}^*), \ldots, c_{p-1}(\mathbf{x}^\star))$ *belongs to the Pareto frontier of problem (4) for* $i = p-1$
2. *There exists a solution* $\mathbf{x}'$ *of* $P_p(\epsilon')$ *for some* $\epsilon' \in \mathbb{R}^{n+(p-1)}$ *such that* $c_i(\mathbf{x}') \leq c_i(\mathbf{x}^\star)$ *for* $1 \leq i \leq p-1$ *with at least an strict inequality and there exists* $(\mathbf{t}, \mathbf{r}) \in \mathcal{T}_p$ *such that* $\mathbf{t} \leq \mathbf{x}^\star$ *and* $\mathbf{r} \geq \mathbf{0}, \mathbf{r} \neq \mathbf{0}$ *(componentwise) and* $\mathbf{r} = c'(\mathbf{x}^\star) - c'(\mathbf{x}')$*.*

The theorem above assures, by induction, that the elements of $\mathcal{X}^\star$ come from solving problem (4) for some $i = 1, \ldots, p-1$ (that is, belong to the solution of the problem taking into account only the first i cost functions) or from reducing elements of the form $(\mathbf{x}^\star, \mathbf{r})$ for some $\mathbf{x}^\star$ efficient solution of problem (4) for some $i = 1, \ldots, p-1$ and some $\mathbf{r}$ that produce an element $(\mathbf{x}^\star, \mathbf{r})$ that is reducible and whose reduction with respect to $\mathcal{T}_p$. Its reduction produces a new element in $\mathcal{X}^\star$.

Theorem 7. *Let* $\mathbf{x}^*$ *be an efficient solution of*

$$\begin{array}{ll} \min & c_1(\mathbf{x}), \ldots, c_i(\mathbf{x}) \\ \text{s.t.} & A\mathbf{x} = \mathbf{b} \quad \mathbf{x} \in \mathcal{Z}^n_{\geq 0} \end{array} \tag{6}$$

with respect to $\preceq_{\hat{c}_i}$ *for some* $i, 1 \leq i \leq p-1$ *then* $\mathbf{x}^\star$ *is efficient for*

$$\begin{array}{ll} \min & c_1(\mathbf{x}), \ldots, c_i(\mathbf{x}), c_{i+1}(\mathbf{x}) \\ \text{s.t.} & A\mathbf{x} = \mathbf{b} \quad \mathbf{x} \in \mathbb{Z}^n_{\geq 0} \end{array} \tag{7}$$

with respect to $\preceq_{\hat{c}_{i+1}}$*.*

Algorithm 1 Algorithm to obtain a minimal set of efficient solutions of a MOILP with p objetives ($p \geq 2$)

Require: vector of cost functions $(c_1, c_2, \ldots, c_p)$, A and $\mathbf{b}$ of problem (1)
  Compute $\mathcal{T}_1$
  $\mathbf{e}_1 \leftarrow$ the solution of $\min\{c_1(\mathbf{x})$ s.t. $A\mathbf{x} = \mathbf{b}, \mathbf{x} \in \mathbf{Z}^n_{\geq 0}\}$ with respect to $\prec_{\hat{c}_1}$
  $\mathcal{X}' \leftarrow \{\mathbf{e}_1\}$
  $P \leftarrow \{\mathbf{e}_1\}$
  for $i = 2, \ldots, p$ **do**
    Compute $\mathcal{T}_i$
    for all $\mathbf{x} \in P$ **do**
      $P := P \setminus \{\mathbf{x}\}$
      Compute $\tilde{G}_\mathbf{x}$
      if $\tilde{G}_\mathbf{x} \neq \emptyset$ **then**
        $G_{\text{jumps}} \leftarrow \{(\mathbf{x}, \mathbf{r})$ such that there exists $(\mathbf{t}, \mathbf{r}) \in \tilde{G}_\mathbf{x}\}$
        for all $(\mathbf{x}, \mathbf{r}) \in G_{\text{jumps}}$ **do**
          $(\mathbf{y}, \mathbf{t}) \leftarrow$ optimal solution of $P_i(\epsilon_1, \ldots, \epsilon_{i-1})$ with respect to $\prec_{\hat{c}_i}$ and initial feasible solution $(\mathbf{x}, \mathbf{r})$.
          if $\mathbf{y} \notin \mathcal{X}'$ **then**
            $\mathcal{X}' \leftarrow \mathcal{X}' \cup \{\mathbf{y}\}$
            $P \leftarrow P \cup \{\mathbf{y}\}$
          end if
        end for
      end if
    end for
  end for
  OUPUT: A minimal set of efficient solutions with respect to $\prec_{\hat{c}_p}$

Algorithm 1 takes into account our previous results and produce a minimal set of efficient points for a given problem (1). For a given $\mathbf{x}$ and a given test-set $\mathcal{T}_i$ we will denote $G_\mathbf{x} = \{(\mathbf{t}, \mathbf{r}) \in \mathcal{T} : \mathbf{t} \leq \mathbf{x}, \mathbf{r} \geq \mathbf{0}, \mathbf{t} \neq \mathbf{0}\}$ and $\tilde{G}_\mathbf{x}$ the subset of elements $(\mathbf{t}, \mathbf{r})$ of $G_\mathbf{x}$ with their last $i-1$ components non comparable.

5 Conclusions

We have introduced a new exact algorithm to obtain a minimal set of efficient points for MOLIPs. It is based on the classical ϵ-constraint method and test sets for a family of IPs computed via Gröbner bases with respect to an order that, properly chosen, guides us in the process of obtaining only efficient solutions and avoiding most of unnecessary computations.

Computational experiments are promising for unbounded knapsack problems (that could be hard to treat with the usual techniques of the binary case). We have have been able to solve problems up to 100 variables for 3 objectives and 75 variables for 4 and 5 objectives (as far as we know the biggest examples proposed in the literature). We have treated too some examples of multi-objective redundancy allocation problems (as in [7]) with excellent results.

References

[1] D Bertsimas, G Perakis, and S Tayur. A new algebraic geometry algorithm for integer programming. *Management Science*, 46(7):999–1008, 2000.

[2] D. Bertsimas and R. Weismantel. *Optimization over integers*. Dynamic ideas, 2005.

[3] V. Blanco and J. Puerto. Partial Gröbner bases for multiobjective integer linear optimization. *SIAM J. Discrete Math.*, 23(2):571–595, 2009.

[4] V. Blanco and J. Puerto. Some algebraic methods for solving multiobjective polynomial integer programs. *J. Symbolic Comput.*, 46(5):511–533, 2011.

[5] K. Bringmann, T. Friedrich, C. Igel, and T. Voß. Speeding up many-objective optimization by Monte Carlo approximations. *Artificial Intelligence*, 204:22–29, 2013.

[6] D. A. Cox, J. Little, and D. O'Shea. *Using algebraic geometry*, volume 185 of *Graduate Texts in Mathematics*. Springer, New York, second edition, 2005.

[7] A. Murat D. Cao and R.B. Chinnam. Efficient exact optimization of multi-objective redundancy allocation problems in series-parallel systems. *Reliability Engineering & System Safety*, 111:154–163, 2013.

[8] M. Ehrgott. *Multicriteria Optimization.* Springer, Berlin, second edition, 2005.

[9] M. Ehrgott and X. Gandibleux. Bound sets for biobjective combinatorial optimization problems. *Comput. Oper. Res.*, 34(9):2674–2694, 2007.

[10] M. Ehrgott and S. Ruzika. Improved ϵ-constraint method for multiobjective programming. *J. Optim. Theory Appl.*, 138(3):375–396, 2008.

[11] Y. Haimes, L.S. Lasdon, and D.A. Wismer. On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, 1(3):296–297, 1971.

[12] M. I. Hartillo-Hermoso, Haydee H. Jiménez-Tafur, and J. M. Ucha-Enríquez. An exact algebraic ϵ-constraint method for bi-objective linear integer programming based on test sets. *European J. Oper. Res.*, 282(2):453–463, 2020.

[13] G. Kirlik and S. Sayın. A new algorithm for generating all nondominated solutions of multiobjective discrete optimization problems. *European Journal of Operational Research*, 232:479–488, 2014.

[14] M. Laumanns, L. Thiele, and E. Zitzler. An efficient, adaptative parameter variation scheme for metaheuristics based on the epsilon-constraint method. *European Journal of Operational Research*, 169:932–942, 2006.

[15] G. Mavrotas and K. Florios. An improved version of the augmented ϵ-constraint method (AUGMECON2) for finding the exact pareto set in multi-objective integer programming problems. *Applied Mathematics and Computation*, 219:9652–9669, 2013.

[16] E. W. Mayr and Albert R. A. R: Meyer. The complexity of the word problems for commutative semigroups and polynomial ideals. *Adv. in Math.*, 46(3):305–329, 1982.

[17] B. Sturmfels. *Gröbner bases and convex polytopes*, volume 8 of *University Lecture Series.* American Mathematical Society, Providence, RI, 1996.

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

[19] W. Zhang and M. Reimann. A simple augmented ϵ-constraint method for multi-objective mathematical integer programming problems. *European Journal of Operational Research*, 234:15–24, 2014.