# An algebraic approach to the Weighted Sum Method in Multi-objective Integer Programming

J. M. Jiménez-Cobano*[1], H. Jiménez-Tafur†[2], and J.M. Ucha-Enríquez ‡[3]

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

**Abstract**

In this work we present how to use test sets of Linear Integer Programming Problems to apply the classical *Weighted Sum Method* in bi-objective optimization. Although this method does not compute in general the complete set of non-dominated solutions, is one of the most widely used due to its simplicity.

The interest of using test sets computed with Gröbner bases is that these combinatorial tools compute exactly which weights should be considered to obtain the complete set of supported non-dominated solutions. Our approach can be extended to some problems in Multi-objective Non-Linear Integer Programming as well.

## 1 Preliminaries

### 1.1 Multi-objective optimization

Most real-life decision-making activities require more than one objective to be considered. These objectives can be conflicting, and thus some trade-offs are needed. As a result, a set of *Pareto-optimal solutions*, rather than a single solution, must be found.

A general multi-objective optimization problem can be written as

$$\begin{array}{ll} \min & f_1(\mathbf{x}),\ldots,f_r(\mathbf{x}) \\ s.t. & g_j(\mathbf{x}) \leq 0,\ j=1,\ldots,J \\ & h_k(\mathbf{x})=0,\ k=1,\ldots,K \\ & \mathbf{x}\in\mathbb{R}^d \end{array} \tag{1}$$

The space of the vectors of decision variables $\mathbf{x}$ is called the *search space*. The space formed by all the possible values of objective functions is called the *objective space*. Since in general there is no feasible point that minimises all the cost functions, we are interested in the *efficient points*: those feasible points $\mathbf{x}^\star$ such that there is no feasible $\mathbf{x}$ with $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^\star)$ with at least one strict inequality for $i = 1,\ldots,r$. If $\mathbf{x}^\star$ is an efficient point, $(f_1(\mathbf{x}^\star),\ldots,f_r(\mathbf{x}^\star))$ is a *non-dominated point* in the objective space. The set of all non-dominated points is usually called the *Pareto front*.

The Weighted Sum Method (cf. [4]) combines all the multi-objective functions into a single objective function $w_1 f_1 + \cdots + w_r f_r$ with $\sum_{i=1}^r w_i = 1$ to express the preferences of the decision maker. So the aim of this approach is to describe (as accurate as possible) the set of of solutions of the following family of single objective problems:

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

$$\begin{array}{ll} \min & w_1 f_1(\mathbf{x}) + \cdots + w_r f_r(\mathbf{x}) \\ s.t. & g_j(\mathbf{x}) \leq 0,\ j = 1, \ldots, J \\ & h_k(\mathbf{x}) = 0,\ k = 1, \ldots, K \\ & \mathbf{x} \in \mathbb{R}^d \end{array} \tag{2}$$

It is well known that this method only produce the complete set of non-dominated solutions if the Pareto front is convex and it is not always clear how to select properly the $w_i$. The solutions obtained by this method are called *supported points.*

### 1.2 Bi-objective linear integer case

In this work we treat the bi-objective linear case for which objectives and constraints are linear functions and in which the variables are integer, that is

$$\begin{array}{ll} \min & \mathbf{c}_1^t\mathbf{x}, \mathbf{c}_2^t\mathbf{x} \\ \text{s.t.} & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \in \mathbb{Z}^n_{\geq 0}, \end{array} \tag{3}$$

for $\mathbf{b} \in \mathbb{Z}^m, A \in \mathbb{Z}^{m\times n}$. We present a combinatorial description of the classical Weighted Sum method for our problem considering the family

$$\begin{array}{ll} \min & w_1\mathbf{c}_1^t\mathbf{x} + w_2\mathbf{c}_2^t\mathbf{x} \\ \text{s.t.} & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \in \mathbb{Z}^n_{\geq 0}, \end{array} \tag{4}$$

for $w_1 + w_2 = 1$, using *test sets* computed via Gröbner bases. We will show how test sets provides the *exact* values of $w_i$ that has to be considered to not drop any supported point.

In [8] an algebraic approach also based in test sets is proposed to apply the *$\epsilon$-constraint method*, another classical approach that solves a family of several problems of only one objective to manage the multi-objective case. Tests sets in that case shows exactly which single objective problems are required to be solved to obtain all non-dominated solutions without redundant calculations.

### 1.3 Tests sets in linear integer programming

Given a linear integer programming problem with a single objective function

$$\begin{array}{ll} \min & \mathbf{c}^t\mathbf{x} \\ \text{s.t.} & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \in \mathbb{Z}^d_{\geq 0} \end{array} \tag{5}$$

a fundamental tool is the *test set* associated to $\mathbf{c}$ and $A$:

**Definition 1.** *A test set of Problem 5 is a set $T \subset \ker(A) \subset \mathbb{Z}^d$ such that: 1) for any feasible solution $\mathbf{x}$ of Problem 5 that is not optimal, there exists $\mathbf{t} \in T$ such that $\mathbf{x} - \mathbf{t}$ is feasible and $\mathbf{c}^t(\mathbf{x} - \mathbf{t}) < \mathbf{c}^t\mathbf{x}$, and 2) given the optimal solution $\mathbf{x}^\star$ of Problem 5, $\mathbf{x} - \mathbf{t}$ is not feasible for any $\mathbf{t} \in T$.*

Test sets produced a natural way of solving Problem 5: starting from a feasible point, subtract elements of the test set as long as it is possible. Test sets can be obtained computing a Gröbner basis of the ideal associated to Problem 5 (cf. [3]) with respect to a suitable monomial ordering that takes into account the cost function $\mathbf{c}$ and codifying the exponents of the polynomials in vectors (see [11]) in which positive components correspond to the term leader of the polynomial. Test sets can be computed for a fixed $\mathbf{b}$ or, usually, valid for any possible $\mathbf{b}$.(*see*[**?**]

Up to our knowledge, the best implementation to compute test sets is `4ti2` ([5]). Test sets been introduced in [12] to solve nonlinear integer problems, and in [2], [6] or [7] for real size cases of Portfolio Selection and Reliability Redundancy Allocation problems.

## 2 The weighted sum method with test sets

Using test sets computed with Gröbner bases it is possible to compute exactly which values of $w_1, w_2$ produce new potential efficient points. This is possible because the Gröbner bases behind the test sets have the following property: if the exponents with respect to an ordering $\prec_2$ of the polynomials of a given base for another ordering $\prec_1$ are the same, the bases with respect to both orderings are the same one (cf. [3]).

**Example 2.** *Let us consider an illustrative example to get the general idea of our procedure. Given the bi-objective assignment problem*

$$\begin{array}{rl} \min & \mathbf{c}_1^t\mathbf{x}, \mathbf{c}_2^t\mathbf{x} \\ \text{s.t.} & \sum_{j=1}^3 x_{ij} = 1, 1 \leq i \leq 3 \\ & \sum_{i=1}^3 x_{ij} = 1, 1 \leq j \leq 3, \\ & x_{ij} \in \{0,1\}, \end{array} \tag{6}$$

*with costs* $\mathbf{c}_1 = (12, 12, 8, 15, 9, 1, 16, 4, 3)$ *and* $\mathbf{c}_2 = (6, 4, 11, 10, 19, 18, 16, 10, 17)$, *a test set*[1] *for the problem with objective* $\mathbf{c}_1$, *that is* $(1-w)\mathbf{c}_1 + w\mathbf{c}_2$ *with* $w = 0$, *is*

$$\begin{aligned} T = \{ \ & (-1,0,1,0,0,0,1,0,-1), (-1,0,1,1,0,-1,0,0,0), (-1,1,0,0,0,0,1,-1,0), \\ & (-1,1,0,1,-1,0,0,0,0), (0,-1,1,0,1,-1,0,0,0), (0,0,0,-1,1,0,1,-1,0), \\ & (0,0,0,0,1,-1,0,-1,1), (0,0,0,1,0,-1,-1,0,1), (0,1,-1,0,0,0,0,-1,1)\} \end{aligned}$$

*and the optimum solution of the problem is* $P_0 = (1, 0, 0, 0, 0, 1, 0, 1, 0)$. *The combination of costs* $(1-w)\mathbf{c}_1 + w\mathbf{c}_2$ *is*

$$(-6w + 12, -8w + 12, 3w + 8, -5w + 15, 10w + 9, 17w + 1, 16, 6w + 4, 14w + 3)$$

*The cost of the first element* $(1, 0, 1, 0, 0, 0, 1, 0, 1)$ *for this combination is*

$$-(-6w + 12) + (3w + 8) + 16 - (14w + 3) = -5w + 9,$$

*so for every* $w \in [0, 1]$ *the exponent (corresponding to the leading term of the polynomials in the Gröbner basis) does not change: the cost is always positive, that is, the cost of positive components always surpass the cost of negative components and the exponent of the element does not change. On the contrary, if the element* $(0, 0, 0, 1, 0, 1, 1, 0, 1)$ *is considered, its cost is*

$$(-5w + 15) - (17w + 1) - (16) + (14w + 3) = -8w + 1.$$

*For any* $w \in [0, 1/8)$ *the exponent of this element does not change, but for* $w = 1/8$ *(and with respect to a monomial ordering that uses namely* $\mathbf{c}_2$ *to break ties) the exponent does change. Checking which are the* $w$ *for which the exponent changes for every element in* $T$, *and considering the smallest one* $w_0$, *we can assure that*

- $T$ *will be the test set of Problem 6 for* $w \in [0, w_0)$. *In this case* $w_0$ *is precisely* $1/8$.
- *A computation of the test set to solve Problem 6 for* $w = 1/8$ *will provide a new test set (and, eventually, a new optimum solution).*

The general procedure, given some $\mathbf{c}_1, \mathbf{c}_2$ the matrix $A$ of the constraints and a feasible point $P_{\text{feas}}$ (that implies the value of $\mathbf{b}$) is Algorithm 2. First a test set $T$ corresponding to cost $\mathbf{c}_1$ and the associated optimum solution are computed. Then consider all the $w \in (0, 1]$ that produce changes of exponent in the elements of $T$, and take the smallest one $w_0$ for which the test set of the minimisation problem with cost $(1-w)\mathbf{c}_1 + w\mathbf{c}_2$ is different to $T$. Repeat this process until $w_0$ turns out to be 1.

[1]We have implemented an ordering that takes into account $\mathbf{c}_1$ first, and break ties with $\mathbf{c}_2$. In `4ti2` this option is possible introducing matrices of costs, with each row corresponding to a different cost.

**Algorithm 1** Weighted Sum with Test Sets

**input:** $\mathbf{c}_1, \mathbf{c}_2, A, \mathbf{b}, P_{\text{feas}}$ of Problem 3
**output:** Set of all supported points
$c_1 := \mathbf{c}_1$
SupPoints $:= \emptyset$
**while** $c_1 \neq \mathbf{c}_2$ **do**
  T = TestSet$(c_1, A)$ $\{T$ does not depend on $\mathbf{b}\}$
  P := Solve$(P_{\text{feas}}, T)$
  SupPoints := SupPoints $\cup\{P\}$. $\{P$ can be superfluous$\}$
  $w_0 = \min_{\mathbf{t}\in T}\{w \in (0,1] | w \text{ changes exponent of } \mathbf{t}\}$ $\{w_0$ can be equal to $1\}$
  $c_1 := (1 - w_0)c_1 + w_0\mathbf{c}_2$
**end while**
T = TestSet$(\mathbf{c}_2, A)$
P := Solve$(P_{\text{feas}}, T)$
SupPoints := SupPoints $\cup\{P\}$. {Optimum for $\mathbf{c}_2$ computed, just in case}
**return** SupPoints =0

The algorithm is correct because the number of different Gröbner basis for a given ideal is finite (cf. [9]).

Our method describes exactly which $w_i$ are necessary to be selected because they produce a different test set, so potentially a new optimal point. Nevertheless, two different test sets can lead to the same optimal point.

**Example 3.** *In Problem 6, when the test set is computed for $w = 1/8$ a new test set is obtained:*

$$\begin{aligned} T' = \{ \quad & (-1,0,1,0,0,0,1,0,-1), (-1,0,1,1,0,-1,0,0,0), (-1,1,0,0,0,0,1,-1,0), \\ & (-1,1,0,1,-1,0,0,0,0), (0,-1,1,0,1,-1,0,0,0), (0,0,0,-1,0,1,1,0,-1), \\ & (0,0,0,-1,1,0,1,-1,0), (0,0,0,0,1,-1,0,-1,1), (0,1,-1,0,0,0,0,-1,1)\} \end{aligned}$$

*However, the optimum point is the same one.*

## 3 Applications to bi-objective non-linear integer programming

As it is explained in [12] test sets can be exploided to solve problems of type

$$\begin{array}{rl} \min & \mathbf{c}^t\mathbf{x} \\ \text{s.t.} & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \in \Omega \\ & \mathbf{x} \in \mathbb{Z}^n_{\geq 0}, \end{array} \tag{7}$$

for $\Omega$ described with non-linear (computable) conditions. The strategy is to calculate the linear optimum for the problem without the non-linear constraints and walking back (adding elements of the test set that worsens the values of the cost function) until points in $\Omega$ are reached. The best point obtained into $\Omega$ is the optimum of Problem 7.

Instead of $\mathbf{c}$ a family of costs $(1 - w)\mathbf{c}_1 + w\mathbf{c}_2$ can be handled: with Algorithm 2 we can achieve the values of $w$ that produce different linear optima for the whole family. Only walking back from them is required to obtain all supported points for the bi-objective counterpart of Problem 7. We present how to apply this framework to a thoroughly studied example in the literature.

In [1] a method to treat a family of three-objective redundancy allocation problem is presented. The functions to be optimised are the cost, weight and reliability of the system. We propose an alternative way to handle the case example (section 3) of three subsystems. We instead solve the bi-objective problem with respect to cost and weight and add the reliability $f_R$ as an extra constraint (for which we ask a convenient value $\rho$), as in [12]. Additionally, we have rearranged the weights (coefficients of $f_2$) to obtain more supported solutions.

We consider the resulting problem of the form

$$\begin{array}{ll} \min & f_1, f_2 \\ \text{s.t.} & 1 \leq \sum_{i=1}^{3} \sum_{j=1}^{m_i} x_{ij} \leq 7 \\ & f_R(\mathbf{x}) \leq \rho \\ & x_{ij} \in \mathbb{Z}_{\geq 0}^{14} \end{array} \tag{8}$$

with

$$\begin{aligned} f_1 = \; & 4x_{11} + 6x_{12} + 7x_{13} + 8x_{14} + 9x_{15} + \\ & +3x_{21} + 4x_{22} + 5x_{23} + 7x_{24} + \\ & +2x_{13} + 4x_{32} + 4x_{33} + 6x_{34} + 8x_{35}, \\ f_2 = \; & 9x_{11} + 6x_{12} + 6x_{13} + 3x_{14} + 2x_{15} + \\ & +12x_{21} + 3x_{22} + 2x_{23} + 2x_{24} + \\ & +10x_{13} + 6x_{32} + 4x_{33} + 3x_{34} + 2x_{35} \end{aligned}$$

and $m_1 = m_3 = 5, m_2 = 4$.

Applying Algorithn 2 to this problem produces the consecutive values of $w_0 = 1/3, 1/2, 1/2$ and $3/4$, and the following list of 6 optimal points of the linear part:

$$\begin{array}{c} (0,0,0,0,1,0,0,1,0,0,0,0,0,1), (0,0,0,0,1,0,0,1,0,0,0,1,0,0), \\ (0,0,0,1,0,0,1,0,0,0,0,1,0,0), (1,0,0,0,0,0,1,0,0,0,0,1,0,0), \\ (1,0,0,0,0,0,1,0,0,1,0,0,0,0), (1,0,0,0,0,1,0,0,0,1,0,0,0,0) \end{array}$$

For each of these optimal points we solve the non-linear corresponding problems with the strategy of walkback (as it is explained in [6]) for the values of $\rho$ for which we are interested in, and obtain a suitable subset of the Pareto front of the non-linear Problem 8. For $\rho = 0.99$ the complete set of supported non-dominated points obtained is

$$\begin{array}{c} \{(0,0,0,0,5,0,1,3,0,0,0,0,0,5), \\ (1,1,0,0,0,0,3,0,0,2,0,0,0,0), \\ (2,0,0,0,0,1,1,0,0,2,0,0,0,0)\} \end{array}$$

We consider that the approach of solving this problem for different values of $\rho$ is more convenient that the one proposed in [1] in which 6112 non-dominated points are reported. The size of this set is unmanageable for a decision maker.

## 4 Conclusions

We have presented an algebraic description of the weighted sum method to compute the supported non-dominated solutions of a bi-objective linear integer programming problem. A generalization to any number of objective functions is a work in progress and requires a complete understanding of how to calculate the subset of the *Gröbner fan* of the ideal corresponding to the the combinations of the costs with any number of parameters.

In addition we have presented how to extend the algebraic weighted sum method to some multi-objective non-linear integer problems, specifically to a widely studied example of redundancy allocation problem.

## References

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

[2] F. J. Castro, M. J. Gago, M. I. Hartillo, J. Puerto, J. M. Ucha, An algebraic approach to integer portfolio problems. *European J. Oper. Res.* **210** (2011), no. 3, 647–659.

[3] D. A. Cox, J. Little, D. O'Shea, *Using algebraic geometry.* Graduate texts in mathematics: 185 (2nd). Springer, New York, 2005.

[4] M. Ehrgott, *Multicriteria optimization.* (2nd). Berlin: Springer, 2005.

[5] 4ti2 team. *4ti2—a software package for algebraic, geometric and combinatorial problems on linear spaces.* (2015) Available at `www.4ti2.de`.

[6] M. J. Gago, M. I. Hartillo, J. Puerto, J. M. Ucha J. M. Exact cost minimization of a series-parallel reliable system with multiple component choices using an algebraic method. *Comput. Oper. Res.* **40** (2011), no. 11, 2752–2759.

[7] Hartillo-Hermoso, M.I., Jiménez-Cobano, J.M., Ucha-Enríquez, J.M.: Finding multiple solutions in nonlinear integer programming with algebraic test-sets. In: Computer algebra in scientific computing 2018, Lille (France). LNCS, vol. 11077, pp. 230–237. Springer, Heidelberg (2016).

[8] M. I. Hartillo-Hermoso, H. Jiménez-Tafur, Haydee, J. M. Ucha-Enríquez, An exact algebraic $\epsilon$-constraint method for bi-objective linear integer programming based on test sets. *European J. Oper. Res.* **282** (2020), no. 2, 453–463.

[9] T. Mora, L. Robbiano, The Gröbner fan of an ideal. *Journal of Symbolic Computation*, **6**(2–3)(1988), 183–208

[10] A. Schrijver, *Theory of linear and integer programming.* Wiley-Interscience Series in Discrete Mathematics. John Wiley  Sons, Ltd., Chichester. A Wiley-Interscience Publication, 1986.

[11] R. Thomas, A geometric Buchberger algorithm for integer programming. *Mathematics of Operations Research*, **20(4)**(1985), 864–884.

[12] S.R. Tayur, R.R. Thomas, N.R. Natraj, An algebraic geometry algorithm for scheduling in presence of setups and correlated demands. *Mathematics Program*, **69** (1995), 369–401.