I IBERIAN CONFERENCE ON MCDM/A (IMCDM/A)
Universidade de Coimbra, 2025

Computing the Pareto front in Multiobjective Linear Mixed Integer Fractional Programming

João Paulo Costa and Maria João Alves, Univ Coimbra, CeBER, Faculty of Economics, Portugal

Objectives

To present 2 Branch & Bound techniques to compute all the nondominated solutions in multiobjective linear mixed integer fractional programming (MOMILFP).

To compare the techniques through computational experiments.

Notation

$$\max \quad z_k(x) = \frac{\sum_{j=1}^{n} c_{kj}x_j + \alpha_k}{\sum_{j=1}^{n} d_{kj}x_j + \beta_k}, \quad k = 1, \ldots, p$$

$$\text{s. t.: } x \in X = \left\{ \begin{array}{c} x \in R^n | Ax \leq b, A \in R^{m.n} b \in R^m, x \geq 0, \\ x_j \in Z_0^+, j \in J, x_j \in \{0,1\}, j \in I \end{array} \right\}$$

$J, I \subseteq \{1, \ldots, n\}, J \cap I = \emptyset, J \cup I \neq \emptyset$

$c, d \in R^{p.n}; \alpha, \beta \in R^p; d_k x + \beta_k > 0; x \in X, k = 1, \ldots p$

Branch & Bound Techniques

Initialization.

Characterize the nondominated region of the MOMILFP problem by computing its pay-off table.

Branch & Bound process.

i) Select the next nondominated region.

ii) Divide this region into two sub-regions, by imposing constraints on one of the objective functions. 2 approaches:

First technique: Divide by the middle of the objective function biggest range in the pay-off table.

Second technique: Cut a little bit (the predefined error) at the bottom of the region, considering the objective function biggest range in the pay-off table.

iii) Compute the pay-off tables of the two new sub-regions in order to characterize them.

The process is repeated for every sub-region until the remaining sub-regions are ‘smaller’ than a predefined error. One sub-region is ‘smaller’ than a predefined error when the range of values of each objective function in the pay-off table is lower than the predefined error.

An example

$$\begin{array}{l} max \quad z_1(x) = \frac{x_1+3x_2+2}{2x_1+x_2+1} \\ max \quad z_2(x) = \frac{3x_1+x_2+2}{x_1+2x_2+1} \end{array} \text{ s. t.: } \left. \begin{array}{l} x_1, x_2 \in \mathbb{Z}_0^+, \\ 3x_1 + 2x_2 \leq 24 \\ -x_1 + x_2 \leq 4 \\ x_1 - x_2 \leq 4 \end{array} \right\} (x_1, x_2) \in X$$

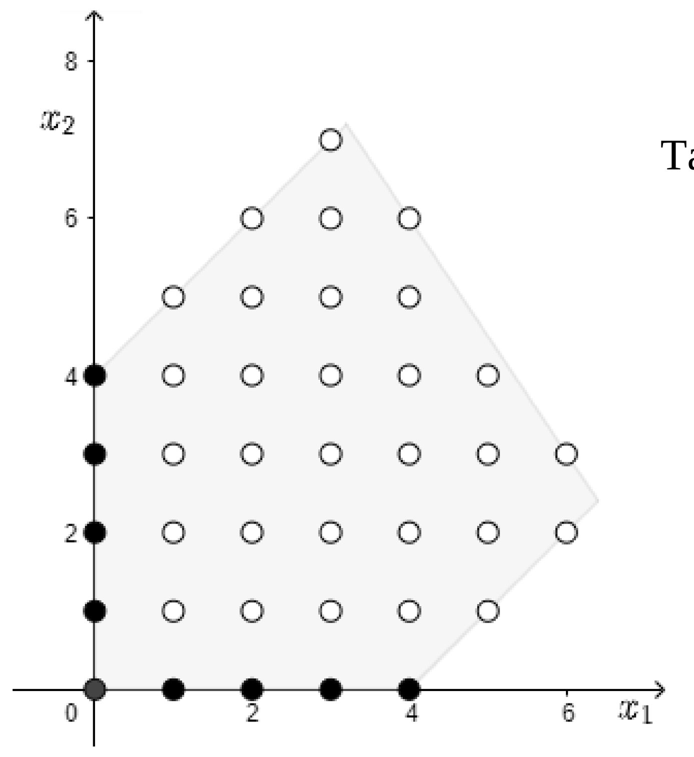


Table 1. All efficient solutions of the example.

x_1	x_2	z_1	z_2	$s(x)$
0	0	2.000	2.000	2.000
0	1	2.500	1.000	1.750
0	2	2.667	0.800	1.733
0	3	2.750	0.714	1.590
0	4	2.800	0.667	1.733
1	0	1.000	2.500	1.750
2	0	0.800	2.667	1.733
3	0	0.714	2.750	1.590
4	0	0.667	2.800	1.733

The techniques provide not only supported but also unsupported nondominated solutions.

It can be observed that the nondominated solution set of a MOLMIFP problem has, in general, a significant part of unsupported solutions.

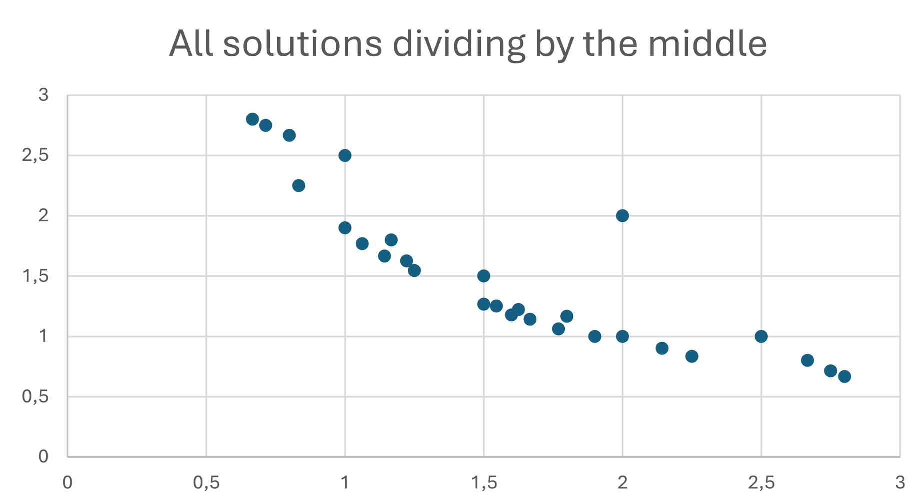
Computational Tests

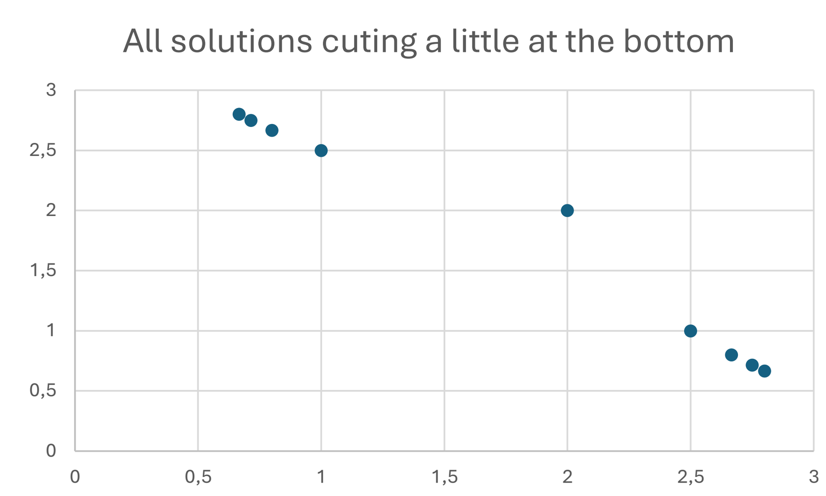
Preliminary tests.

CPLEX™ Optimizer V12.4 for solving the auxiliary linear programming problems.

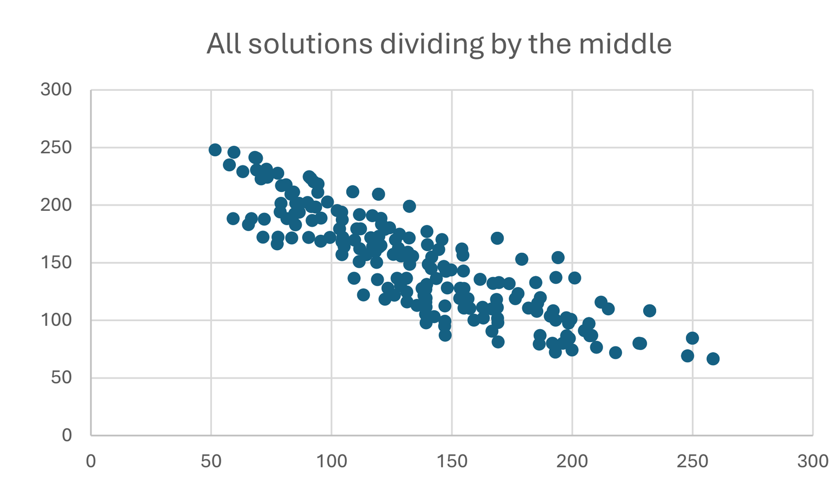
The techniques were coded in Delphi™ XE25 for Win64.

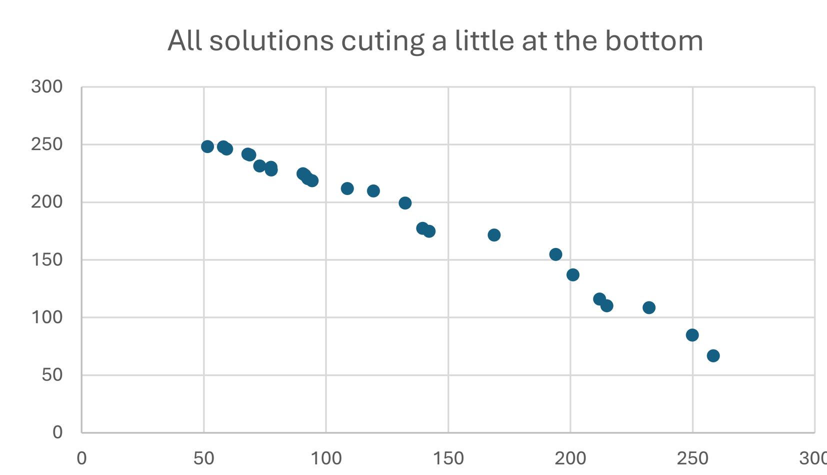
Using the example



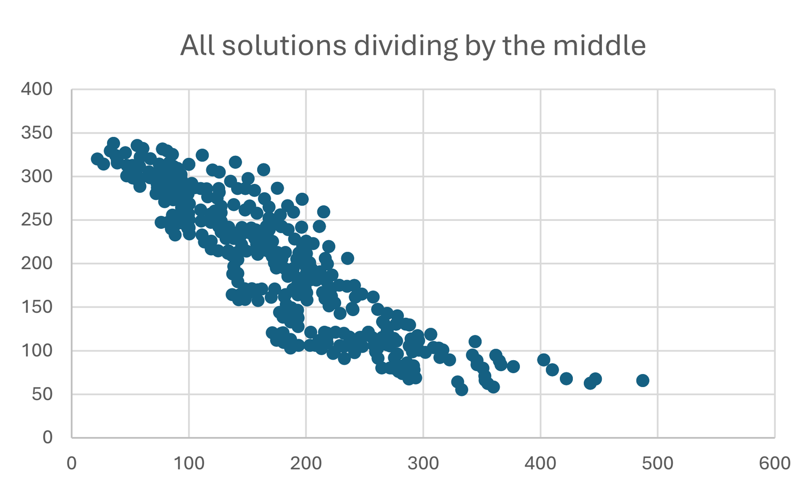


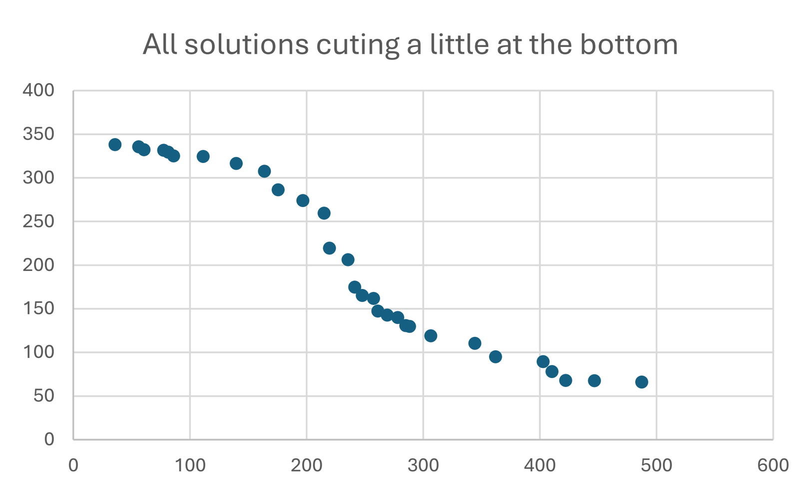
2 objective functions; 5 constraints; 20 binary variables.



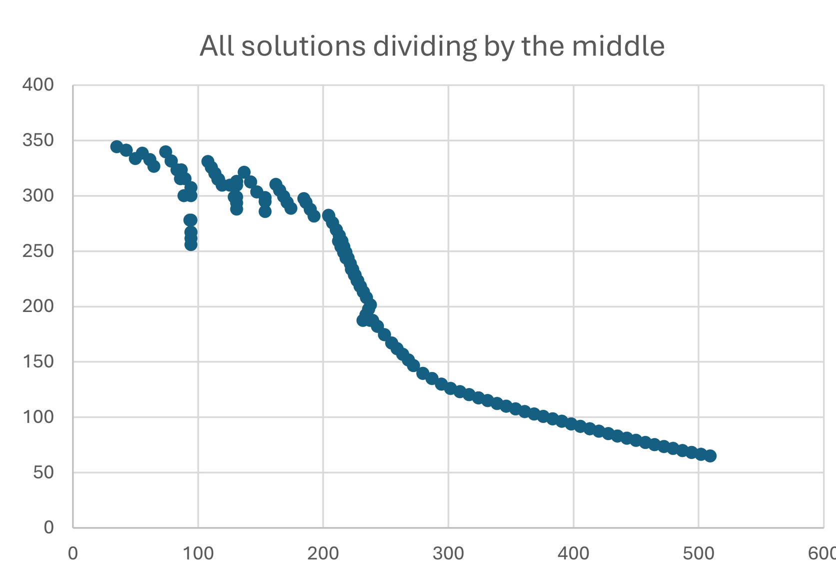


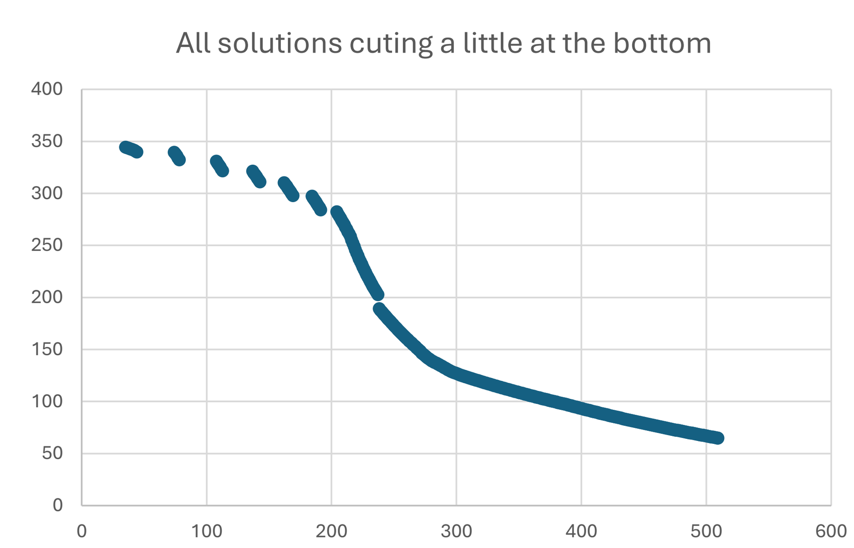
2 objective functions; 5 constraints; 20 integer variables.





2 objective functions; 5 constraints; 10 integer and 10 continuous variables.





Conclusions

Computational experiments show that the algorithms can deal with practical MOMILFP problems with binary, general integer and mixed-integer variables. The second technique seems to be better.

More tests are needed.

The algorithm and the software implementation must be improved to avoid unnecessary computations.