Computing the Pareto frontier of a bi-objective bilevel linear problem using a multiobjective mixed-integer programming algorithm

Maria João Alves [a*], Stephan Dempe [b], Joaquim J. Júdice [c]

[a] *Fac. de Economia, Universidade de Coimbra / INESCC, Coimbra, Portugal;*
[b] *Freiberg University, Freiberg, Germany;*
[c] *Dep. de Matemática, Universidade de Coimbra / Instituto de Telecomunicações, Coimbra, Portugal*

In this paper we study the bilevel linear programming problem with multiple objective functions at the upper level (with particular focus on the bi-objective case) and a single objective function at the lower level. We have restricted our attention to this problem type because the consideration of several objectives at the lower level raises additional issues to the bilevel decision process resulting from the difficulty of anticipating a decision from the lower level decision maker. We examine some properties of the problem and propose a methodological approach based on the reformulation of the problem as a multiobjective mixed 0-1 linear programming problem. The basic idea consists in applying a reference point algorithm that has been originally developed as an interactive procedure for multiobjective mixed-integer programming. This approach further enables to characterize the whole Pareto frontier in the bi-objective case. Two illustrative numerical examples are included to show the viability of the proposed methodology.

Keywords: bilevel programming; multiobjective; mixed-integer programming

2000 Mathematics Subject Classifications: 91A65; 90C29; 90C11

1. Introduction

Bilevel mathematical programs model hierarchical optimization problems in which there are two decision makers that have different objective functions, variables and constraints. The decision process is sequential as the upper level decision maker – the leader – makes his/her decisions first, anticipating those of the lower level decision maker – the follower. The bilevel programming problem has been widely studied and most of this research has been devoted to the linear case. For comprehensive references on bilevel programming we refer to [6, 12, 13]. In addition, several applications are described in [10].

The bilevel programming problem considering multiple objectives has great interest for many applications, in particular in transportation system planning and traffic management. For instance, the manager may want to minimize the total travel time of all travellers, to minimize gasoline consumption (by varying the cycle time of traffic lights) and to minimize the construction cost of road improvements, so he/she must take into account several distinct objective functions. In addition, since the options of the manager affect the travel choices of the users, he/she must also accommodate the traffic behaviour that results from the individual decisions of the travellers (the lower-level problem). A situation of this type can be modelled as a bilevel programming problem with multiple objectives at the upper-level. However, in contrast with the vast literature

[*] Corresponding author. Email: mjalves@fe.uc.pt

on the bilevel problem, little research work has been done thus far on multiobjective bilevel problems.

Yin [35] and Erkut and Gzara [18] have recognized the importance of considering multiple objectives in their bilevel applications for planning transportation systems. Yin points out that a multiobjective bilevel modelling approach can be a powerful decision tool, and proposes a solution based on genetic algorithms. A numerical experiment was conducted on a transportation network design bi-objective problem with a traffic assignment lower level problem. Erkut and Gzara deal with a problem of network design for hazardous material transportation in which the government designates a network and the carriers choose the routes on the network. After a first approach to the problem using a bilevel integer formulation, the authors felt the need to extend the model to incorporate two objectives at the upper level, the transportation cost and the risk. A heuristic was then used to compute nearly Pareto optimal solutions.

A few more methodological studies can be found in literature on multiobjective bilevel problems. Shi and Xia [29, 30] present an interactive algorithm for nonlinear bilevel problems with multiple objectives in both levels. The algorithm simplifies the problem by transforming it into separate multiobjective decision-making problems at each level, using in addition a *satisfactoriness* concept to model the preferences of the upper level decision maker. This work has been extended to three-level multiobjective problems by Abo-Sinna and Baky [1] with some modifications in assigning *satisfactoriness* to each objective function at all levels of the problem.

Zhang et al. [36] have developed an approximation branch-and-bound algorithm to deal with bilevel linear problems with fuzzy parameters when the leader or the follower or both have multiple objectives.

Eichfelder [15, 17] has studied the nonlinear multiobjective bilevel programming problem with upper level constraints uncoupled from the lower level variables, and shows that the constraint set of the upper level problem can be expressed as the set of K-minimal solutions of a multiobjective problem with respect to a certain closed pointed convex cone K. Based on this result, the author proposes an algorithm for problems with two objectives at each level and one upper level variable. In [16] these results have been extended to problems with upper level constraints that depend on the lower level variables.

Deb and Sinha [11] suggest an evolutionary multi-objective optimization algorithm for solving bilevel problems with multiple objectives (in both levels). The basic idea of the proposed procedure is to keep two interacting populations in a coevolutionary algorithm so that, instead of a serial and complete optimization of the lower level problem for every upper level solution, both upper and lower level optimization tasks can be pursued simultaneously through iterations. The algorithm was tested using a couple of nonlinear problems described in [16].

Nishizaki and Sakawa [28] have also addressed the multiobjective bilevel linear programming problem with multiple objectives at both levels. Since the leader must take into consideration an infinite number of responses of the follower with respect to each one of his/her decisions, the authors assume that the leader has some subjective anticipation or belief, which can be optimistic, pessimistic or an anticipation arising from the past behaviour of the follower. Optimistic anticipation means that the leader anticipates that the follower will take a decision desirable for the leader, and pessimistic anticipation is the reverse. The solution procedure presented in [28] is based on solving interactively a reference point scalarizing program, for which the leader is asked to update the reference point. Given a reference point to the upper level objectives, the optimistic (pessimistic) anticipation approach assumes that the follower returns the

Pareto optimal solution of his/her problem that best (worst) fits the reference point of the leader. The procedure stops when the leader is satisfied with the obtained solution.

The optimistic and the pessimistic anticipation introduced by Nishizaki and Sakawa [28] for multiobjective bilevel problems clearly show the difficulties of a bilevel decision process when multiple lower level objectives are considered. These difficulties naturally have serious implications on the development of an effective solution procedure.

In case the follower has a single objective, it is often assumed that the rational response of the follower for a decision of the leader is deterministic. Whenever it is not a singleton, an approach consists in assuming that the leader is free to select the solution that suits him/her best. This interpretation is legitimate in case side payments are allowed; it is the so-called optimistic modelling approach for single-objective bilevel problems. When cooperation between the leader and the follower is not allowed, or if the leader is risk-averse and wishes to limit the "damage" resulting from an undesirable selection, a pessimistic approach can be admitted [10]. Furthermore, intermediate approaches between the optimistic and the pessimistic approaches have been discussed [24], and a partial cooperation model was proposed in [9], which includes a cooperation index reflecting the degree of the follower's partial cooperation. The discussion of optimistic and pessimistic approaches can also be found in [12].

There is, however, a major difference between this case and the multiobjective one. In the single-objective case, the reaction solutions are alternative optima to the follower with respect to a decision of the leader, i.e., they all attain the same value of the follower's objective. In case of multiple objectives at the lower level, there is no single optimal objective value to the follower, but rather a set of nondominated objective vectors in which a better value for one objective can only be obtained if at least one of the other objectives is worsened. Therefore, a compromise solution taking into account the multiple objective functions must be selected but, unless a scalar-valued utility function is assumed *a priori* (which turns the lower level problem into a single-objective one), the follower's decision may be very difficult to anticipate. This uncertainty on the behaviour of the follower puts additional difficulties for the development of a procedure that can provide effective decision aid in multiobjective bilevel problems.

These considerations have led us to restrict our attention to the bilevel linear programming problem with multiple objectives at the upper level and a single objective at the lower level. In addition, we have assumed that the problem has no lower level variables in the upper level constraints. In this paper we examine some properties of this type of problem and we propose a methodological approach based on its reformulation as a multiobjective mixed 0-1 linear programming problem. Particular attention is given to the bi-objective case.

An interactive reference point procedure developed by Alves and Clímaco [2] for multiobjective mixed-integer linear programming is used to compute Pareto optimal solutions to the multiobjective bilevel problem. This procedure exploits the use of branch-and-bound techniques for solving successive reference point scalarizing programs in which the reference point is automatically updated to perform a directional search for Pareto optimal solutions. It is shown that this approach can be further used to fully determine the Pareto region of a bi-objective problem (i.e., to act as a generating method) except for a gap between continuous solutions that can be set as small as the user wishes.

The remainder of this paper is organized as follows. In section 2, the problem is formulated and basic concepts of bilevel and multiobjective programming are introduced. Some characteristics of the multiobjective bilevel linear problem with multiple objectives at the upper level are also examined in this section. Section 3 shows a relation between the induced region of the bilevel linear problem and the set of Pareto optimal solutions of a multiobjective linear program and discusses the difficulties of profiting from that result to develop an effective procedure for multiobjective bilevel linear problems. In section 4 the problem is reformulated as a multiobjective mixed 0-1 linear problem. Section 5 introduces the methodological approach by introducing the interactive reference point procedure in [2] for multiobjective mixed-integer linear programming and proposing a generating algorithm for the bi-objective case. Two illustrative examples of the application of this algorithm to bi-objective bilevel problems are included in section 6 and some conclusions and perspectives on future work are included in section 7.

2. The bilevel linear programming problem with multiple objectives at the upper level

2.1 Problem definition

The Multi-Objective Bi-Level Linear Problem (MOBLLP) can be expressed as follows:

$$\begin{array}{ll}
\max\limits_{x,y} & F_1(x,y) = c_1^1 x + d_1^1 y \\
\ldots & \\
\max\limits_{x,y} & F_k(x,y) = c_k^1 x + d_k^1 y \\
\text{s.t.} & A^1 x \le b^1 \\
& x \ge 0 \\
& y \in \arg\max\limits_{y} \left\{ f(y) = d^2 y :\ A^2 x + B^2 y \le b^2,\ \ y \ge 0 \right\}
\end{array} \tag{1}$$

where $x \in \Re^{n_1}$ and $y \in \Re^{n_2}$ are the upper level and lower level decision variables, respectively, k is the number of objective functions of the leader, $c_i^1 \in \Re^{n_1}$, $d_i^1 \in \Re^{n_2}$, i=1,...,k, $d^2 \in \Re^{n_2}$, $A^1 \in \Re^{m_1 \times n_1}$, $b^1 \in \Re^{m_1}$, $A^2 \in \Re^{m_2 \times n_1}$, $B^2 \in \Re^{m_2 \times n_2}$, $b^2 \in \Re^{m_2}$ and cx represents the inner product of two vectors c and x.

In a bilevel problem, the upper level decision maker (*leader*), makes his/her decision first and through his choice of $x \in \Re^{n_1}$ reduces the set of feasible choices available to the lower level decision maker (*follower*). Next, the follower reacts to the leader's decision by choosing an $y \in \Re^{n_2}$ that optimizes his/her objective function. Hence, the follower indirectly affects the leader's solution space and outcomes for his/her single or multiple objective functions.

The following sets should be considered.
S is the *constraint region* of the MOBLLP, which includes all the constraints of the leader and of the follower. We assume that S is non-empty and compact and it is defined as follows:

$$S = \left\{ (x,y) : A^1 x \le b^1,\ A^2 x + B^2 y \le b^2, x \ge 0, y \ge 0 \right\}$$

$P(x)$ is the follower's *rational reaction set* to a given x:

$$P(x) = \arg\max_{y} \left\{ f(y) : B^2 y \le b^2 - A^2 x,\ \ y \ge 0 \ \right\}$$

The feasible set for the leader, which is called the *induced region*, is defined as

$$IR = \left\{ (x,y) : (x,y) \in S,\ y \in P(x) \right\}$$

In terms of the above notation, the MOBLLP can be written as

$$
\begin{array}{ll}
\max\limits_{x,y} & F_1(x,y) = c_1^1 x + d_1^1 y \\
\ldots & \\
\max\limits_{x,y} & F_k(x,y) = c_k^1 x + d_k^1 y \\
\text{s. t.} & (x,y) \in IR
\end{array}
\qquad (2)
$$

It should be noted that the MOBLLP formulation presented in (1) considers upper level constraints uncoupled from the lower level variables. Actually, many authors define the bilevel problem without upper level constraints while others consider upper level constraints involving both upper and lower level variables. It has been shown that *IR* is not necessarily a connected set when there exist upper level constraints containing some lower level variables *y*. On the other hand, *IR* is always connected when the variables *y* are not included in the upper-level constraints. For a discussion on this topic we refer to [4, 25]. Consequently, the multiobjective problem to be studied in this paper has a connected feasible region.

2.2 Basic concepts in multiobjective optimization

In this section we only present a few basic concepts on multiobjective optimization that are used in the rest of the paper. Mathematical foundations and methods of multicriteria (multiobjective) optimization can be found in the books [14], [26] and [31].
To facilitate the exposition of the concepts, consider a multiobjective optimization problem defined generically as follows:

$$
\begin{array}{ll}
\max\limits_{x} & z_1(x) \\
\ldots & \\
\max\limits_{x} & z_k(x) \\
\text{s.t.} & x \in X
\end{array}
\qquad (3)
$$

Let Z denote the image of X in the objective function (criterion) space: $Z = \{z \in \Re^k : z = z(x) = (z_1(x),\ldots,z_k(x)),\ x \in X\}$

In multiobjective optimization there is not, in general, a feasible solution that optimizes simultaneously all objective functions. Thus, the concept of optimal solution is replaced by *Pareto optimal*, *efficient* or *nondominated* solution. Although these designations can be considered interchangeable, some authors prefer to use 'Pareto optimal' or 'efficient' for decision vectors *x* and 'nondominated' for criterion vectors *z* belonging to *Z* [31]. We do not make any particular distinction, adopting herein the 'Pareto optimal' designation for the solutions and referring to the set of nondominated criterion vectors as the *Pareto frontier*.

A solution $x' \in X$ ($z' \in Z$) is *Pareto optimal*, *efficient* or *nondominated* if and only if there is no other $x \in X$ such that $z_j(x) \geq z_j(x')$ for all j=1,...,k and $z_j(x) > z_j(x')$ for at least one j.
In other words, $x' \in X$ ($z' \in Z$) is Pareto optimal iff there is no other $x \in X$ such that $z = z(x)$ dominates $z' = z(x')$, according to the following definition of dominance:
$z \in \Re^k$ *dominates* $z' \in \Re^k$ if and only if $z_j \geq z'_j$ for all j=1,...,k and $z_j > z'_j$ for at least one j.

A solution $x' \in X$ ($z' \in Z$) is said to be a *weakly Pareto optimal* solution if and only if there is no other $x \in X$ such that $z_j(x) > z_j(x')$ for all j=1,...,k.

Although the set of weakly Pareto optimal solutions includes the set of Pareto optimal solutions, for the sake of simplicity we only refer to 'weakly Pareto optimal' a solution for which the Pareto optimality condition does not hold.

The Pareto optimal set is, in general, nonconvex (even in multiobjective linear programming) and may be not connected. According to Miettinen [26, p.20], the connectedness of the sets of Pareto optimal solutions and weakly Pareto optimal solutions is an important feature because it is often useful to know how well we can move continuously from one (weakly) Pareto optimal solution to another one. Several results have been established for the connectedness of the Pareto optimal set. In particular, this set is connected if the feasible region is convex and compact and the maximizing objective functions are concave or strictly quasiconcave (see, e.g. [7]). As it is shown later, the Pareto optimal set of a MOBLLP may be not connected.

Another fundamental concept for the study of the MOBLLP is the distinction between *supported* and *unsupported* Pareto optimal solutions. A nondominated criterion vector $z' \in Z$ is called *unsupported* if it is dominated by any infeasible convex combination (i.e. not belonging to *Z*) of other nondominated criterion vectors. Otherwise, z' is a *supported* nondominated criterion vector. Inverse images, $x' \in X$, of supported (unsupported) nondominated criterion vectors $z' \in Z$ are supported (unsupported) Pareto optimal solutions.

Unsupported Pareto optimal solutions cannot be obtained by optimizing scalar surrogate functions consisting of weighted-sums of the objective functions. As is shown next, a MOBLLP may admit not only supported but also unsupported Pareto optimal solutions.

It should also be remarked that for the linear bilevel programming problem an optimal solution can be found at a vertex of the set *S* (the *constraint region*) − see e.g. [6] or [12] for a proof. In MOBLLP the set of Pareto optimal solutions (or even weakly Pareto optimal solutions) may be not equal to the union of faces of this set, as is shown in the next example.

2.3 An example of MOBLLP

Let us now illustrate the concepts previously defined using a MOBLLP example with two objective functions.

Example 1.

$$\max_{x,y} F_1(x,y) = -2x$$

$$\max_{x,y} F_2(x,y) = -x + 5y$$

$$\text{s.t.} \quad \max_{y} f(y) = -y$$

$$\begin{array}{lll} \text{s.t.} & x - 2y \le 4 & (1) \\ & 2x - y \le 24 & (2) \\ & 3x + 4y \le 96 & (3) \\ & x + 7y \le 126 & (4) \\ & -4x + 5y \le 65 & (5) \\ & x + 4y \ge 8 & (6) \\ & x, y \ge 0 & \end{array}$$

The problem is depicted in Figure 1.

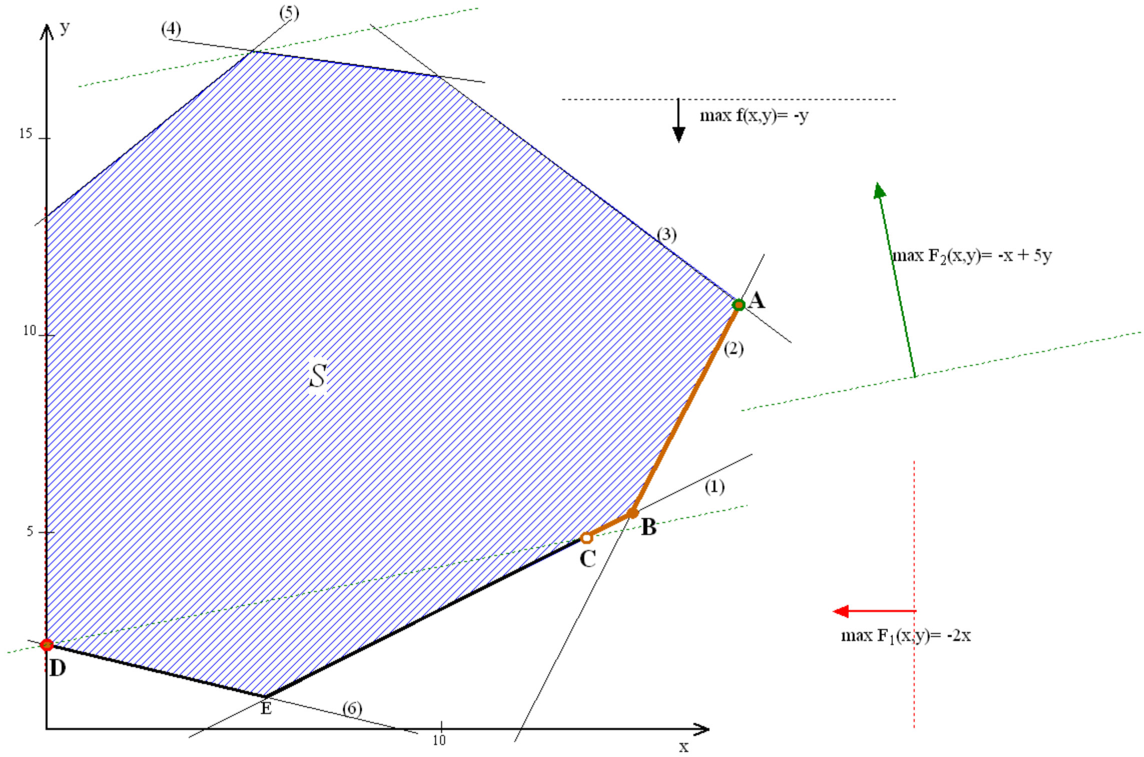

Figure 1. Graphical representation of e*xample 1*.

The induced region, *IR*, is [DE]∪[EB]∪[BA].
Graphically, we can also determine the whole Pareto optimal set (a subset of *IR*) for this bi-objective bilevel problem, which is {D}∪]CB]∪[BA]. The values of the decision variables and the upper level objective functions in the points A, B, C and D are shown in Table 1.

Table 1. Values of the (weakly) Pareto optimal extreme points of *example 1*.

	x	y	F_1	F_2
A	17.45455	10.90909	-34.9091	37.09091
B	14.66667	5.333333	-29.3333	12
C	*13.33333*	*4.666667*	*-26.6667*	*10*
D	0	2	0	10

The following issues can be observed:

- D is the Pareto optimal solution that maximizes F_1 and A is the Pareto optimal solution that maximizes F_2.
- Solutions from *C* to D (exclusive) of the induced region are not Pareto optimal as they are dominated by D. Comparing with *C*, D is superior only in F_1 being equal in F_2. Hence, *C* is a *weakly Pareto optimal* solution.
- Only A and D are *supported* Pareto optimal solutions. All the others are *unsupported*, because there are convex combinations of A and D that would dominate them if they were feasible, i.e., if they belonged to *IR*.
- The Pareto optimal set is *not connected*.

Figure 2 shows the Pareto optimal points in the objective space (*Pareto frontier).*

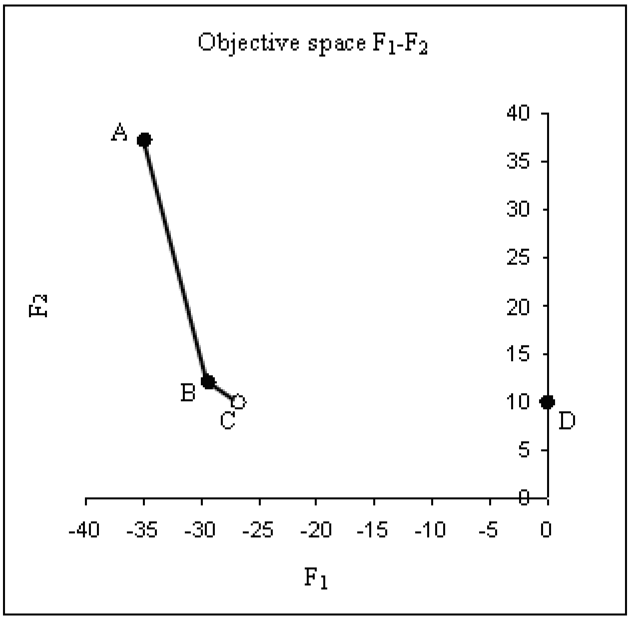

Figure 2. The Pareto frontier of *example 1*.

This example shows that the Pareto optimal set of a MOBLLP may be *not connected* and may have *unsupported* solutions. Furthermore, unsupported solutions may constitute the major part of the Pareto optimal set. Hence, they should not be disregarded.

3. Reduction of the induced region to a multiobjective linear problem

The equivalence between a linear bilevel programming problem and linear optimization over the Pareto optimal set of a multiobjective linear problem (MOLP) was first presented by Fülöp [21] and this is also summarized in the book by Dempe [12, pp.33-34]. Recently, this result has been exploited by Glackin et al. [22], who has proposed an algorithm based on MOLP for solving the linear bilevel problem.

Basically, this result establishes a relation between the induced region, *IR*, and the Pareto optimal set of a MOLP. This property holds for both the single-objective bilevel linear problem and for the MOBLLP (1) as we have only considered multiple objectives at the upper level which do not affect the induced region. This relation can be stated as follows.

Consider the MOBLLP defined in (1) with any $k \geq 1$, and the definitions above for S and *IR*. Then, *IR* coincides with the Pareto optimal set of the following MOLP with n_1+2 objective functions:

$$\begin{aligned} &\max_{x,y} \quad f(y) = d^2 y \\ &\max_{x,y} \quad x_i \ , \quad i = 1,\dots, n_1 \\ &\max_{x,y} \quad -\sum_{i=1}^{n_1} x_i \\ &\text{s.t.} \quad (x,y) \in S \end{aligned} \tag{4}$$

Moreover, different formulations of the MOLP (4) can be considered provided that the coefficient vectors of the last n_1+1 objective functions constitute a set of generators of the cone $\Re^{n_1}$. The proof of this result is included in the Appendix (*Proposition 1*) for a more general MOLP formulation.

Although this result is theoretically interesting, its use in practice is at least doubtful due to the large number of objective functions in the MOLP.
Suppose that we wish to exploit this result to solve a MOBLLP by using a procedure which computes several Pareto optimal solutions to (4) that are evaluated by the upper level objective functions of the MOBLLP and then selects the nondominated points. Even if we generate all the Pareto optimal extreme points of (4) or a more extended set of Pareto optimal solutions, attempting to have a representative set of the induced region of the MOBLLP, we have no guarantee that the selected solutions are Pareto optimal solutions to the MOBLLP. The following example illustrates this drawback.

Example 2.

Consider the following MOBLLP:

$$\begin{aligned}
&\max_{x,y} \quad F_1(x,y) = 2x_1 - 4x_2 + y_1 - y_2 \\
&\max_{x,y} \quad F_2(x,y) = -x_1 + 2x_2 - y_1 + 5y_2 \\
&\text{s.t.} \quad \max_{y} \quad f(y) = 3y_1 + y_2 \\
&\qquad\qquad \text{s.t.} \quad 4x_1 + 3x_2 + 2y_1 + y_2 \le 60 \\
&\qquad\qquad\qquad\;\; 2x_1 + x_2 + 3y_1 + 4y_2 \le 60 \\
&\qquad\qquad\qquad\;\; x_1, x_2, y_1, y_2 \ge 0
\end{aligned}$$

The formulation (4) with respect to this MOBLLP is the following:

$$\begin{aligned}
&\max_{x,y} \quad f(y) = 3y_1 + y_2 \\
&\max_{x,y} \quad x_1 \\
&\max_{x,y} \quad x_2 \\
&\max_{x,y} \quad -x_1 - x_2 \\
&\text{s.t.} \quad 4x_1 + 3x_2 + 2y_1 + y_2 \le 60 \\
&\qquad\;\; 2x_1 + x_2 + 3y_1 + 4y_2 \le 60 \\
&\qquad\;\; x_1, x_2, y_1, y_2 \ge 0
\end{aligned}$$

Using a *Vector Maximum Algorithm* [31] for computing all the Pareto optimal basic solutions of problem (4) we find 5 solutions which are shown in Table 2. These solutions form the set of extreme points of the induced region of the MOBLLP.

Table 2. Pareto optimal extreme points of formulation (4) w.r.t. the MOBLLP of *example 2*.

	x_1	x_2	y_1	y_2	f(y)	F$_1$(x, y)	F$_2$(x, y)
solution 1	0	0	20	0	60	20	–20
solution 2	15	0	0	0	0	30	–15
solution 3	0	20	0	0	0	–80	40
solution 4	0	8.5714	17.1429	0	51.429	–17.143	0
solution 5	7.5	0	15	0	45	30	–22.5

The evaluation of these solutions by F_1 and F_2, whose values are also shown in Table 2, indicates that *solution 1* and *solution 5* are dominated by *solution 2*, while *solutions 2*, *3*

and *4* are nondominated within this set. *Solutions 2* and *3* are definitely Pareto optimal solutions to the MOBLLP, because these are the single extreme points of the induced region that optimize individually F_1 and F_2, respectively. However, no such guarantee exists in what concerns the Pareto optimality of *solution 4*.
Actually, a further study of this bi-objective bilevel problem (using the procedure presented in section 5) enables us to conclude that *solution 4* is not a Pareto optimal solution to the MOBLLP, as it is dominated by feasible convex combinations of *solutions 2* and *3*, e.g. $(x_1, x_2, y_1, y_2) = (9.351, 7.532, 0, 0)$ where $(F_1, F_2) = (-11.428, 5.714)$, among others.

This example shows that an effective approach to deal with the MOBLLP based on the reduction of the induced region to a MOLP may be difficult to implement. Therefore, other type of procedures must be designed to address the problem and, in particular, to compute Pareto optimal solutions. A possible strategy may be the transformation of the MOBLLP into another problem that can be efficiently solved using a suitable procedure. This is the approach followed in this work, which is based on the reformulation of the MOBLLP as a multiobjective mixed integer linear programming problem.

4. Reformulation of the MOBLLP as a multiobjective mixed 0-1 linear problem

A bilevel linear programming problem can be reformulated as a mathematical program with complementarity constraints, which in turn is equivalent to a mixed integer (0-1) linear programming problem [5, 20]. In this section we follow these transformations to reduce MOBLLP to a multiobjective mixed-integer (0-1) linear problem (MOMILP).
Consider the MOBLLP formulation stated in (1). This problem can be first reformulated as a multiobjective linear program with complementarity constraints (5), which contains the primal and dual constraints associated to the follower's problem and the corresponding complementarity slackness conditions [6, 12, 23]:

$$\begin{array}{lll}
\max\limits_{x,y} & F_1(x,y) = c_1^1 x + d_1^1 y & \\
\ldots & & \\
\max\limits_{x,y} & F_k(x,y) = c_k^1 x + d_k^1 y & \\
\text{s.t.} & A^1 x \le b^1 & \lambda\left(b^2 - A^2 x - B^2 y\right) = 0 \\
& A^2 x + B^2 y \le b^2 & y(\lambda B^2 - d^2) = 0 \\
& \lambda B^2 \ge d^2 & \\
& y \ge 0,\ x \ge 0,\ \lambda \ge 0 &
\end{array} \tag{5}$$

where $\lambda \in \Re^{m_2}$.

Using the transformations discussed in [5], the complementarity constraints can be replaced by linear constraints with binary variables and problem (5) is reformulated as the following MOMILP:

$$\begin{array}{lll}
\max\limits_{x,y} & F_1(x,y) = c_1^1 x + d_1^1 y & \\
\ldots & & \\
\max\limits_{x,y} & F_k(x,y) = c_k^1 x + d_k^1 y & \\
\text{s.t.} & A^1 x \le b^1 & \lambda + \mathrm{M}u \le \mathrm{M}e \\
& A^2 x + B^2 y \le b^2 & -A^2 x - B^2 y - \mathrm{M}u \le -b^2
\end{array} \tag{6}$$

$$\begin{array}{ll} \lambda B^2 \geq d^2 & y + \mathrm{M}v \leq \mathrm{M}e \\ y \geq 0\,,\ x \geq 0\,,\ \lambda \geq 0 & \lambda B^2 - \mathrm{M}v \leq d^2 \\ & u \in \{0,1\}^{m_2}\,,\ v \in \{0,1\}^{n_2} \end{array}$$

where M represents a large finite positive constant and e a vector of appropriate dimension and all elements equal to one.

As a direct consequence of the results proved by Audet et al. [5] for the single-objective case, the following result can be stated.

Proposition 2 − Suppose that (x', y') is a Pareto optimal solution of the MOBLLP (1). Then, there exist a large finite constant M>0 and $\lambda' \in \Re^{m_2}$, $u' \in \{0,1\}^{m_2}$, $v' \in \{0,1\}^{n_2}$ such that $(x', y', \lambda', u', v')$ is a Pareto optimal solution of (6). Also, for such an M, if $(x', y', \lambda', u', v')$ is a Pareto optimal of the last problem, then (x', y') is a Pareto optimal solution of the MOBLLP (1).

5. A methodology based on a MOMILP procedure

Methods for computing Pareto optimal solutions to a multiobjective programming problem work in general by transforming the multiobjective problem into a parameterized single-objective problem − a *scalarizing* program − such that the optimum of the scalarizing program for a set of parameters corresponds to a Pareto optimal solution, or at least a weakly Pareto optimal solution, to the multiobjective problem. Different scalarization techniques can be used, e.g. optimization of weighted-sums of the objective functions, constraint techniques or reference point techniques. Discussions on this topic can be found in [14] and [19], among others.
Multiobjective bilevel linear problems admit not only supported but also unsupported Pareto optimal solutions and the latter type of solutions should not be disregarded as it may constitute the major part of the Pareto optimal set. Unsupported Pareto optimal solutions cannot be reached by optimizing simple weighted-sums of the objective functions even if a complete parameterization on the weights is attempted. In contrast to the weighted-sum scalarization, reference point techniques [33] can reach both supported and unsupported Pareto optimal solutions, thus being more adequate to deal with the MOBLLP.

Reference point approaches [33] can be seen as a generalization of goal programming. The reference point can be interpreted as a goal but the sense of “coming close” changes to “coming close or better”, which does not mean minimization of a distance but rather the optimization of an *achievement scalarizing function* [34].
Consider the general formulation (3) of a multiobjective optimization problem. Let $q \in \Re^k$ denotes a criterion reference point, which may represent aspiration levels that the decision maker would like to attain for the objective functions. Let us consider the min-max scalarizing program $\min_{x \in X} \left\{ \max_{i=1 \dots k} \left(q_i - z_i(x) \right) \right\}$ which projects q onto the (weakly) Pareto frontier. Since the optimal solution to this scalarizing program may be only a weakly Pareto optimal solution, the term $\left(-\rho \sum_{i=1}^{k} z_i(x) \right)$ is usually added to the scalarizing function to ensure the Pareto optimality condition (where ρ>0 is a constant small enough).

The augmented scalarizing program is thus, $\min_{x\in X}\left\{\max_{i=1\ldots k}\left(q_i - z_i(x)\right)-\rho\sum_{i=1}^{k} z_i(x)\right\}$, which is equivalent to:

$$\begin{array}{lll} \min\limits_{x,\alpha} & \left(\alpha-\rho\sum\limits_{i=1}^{k} z_i(x)\right) & \\ \text{s.t.} & z_i(x)+\alpha \geq q_i & i=1,\ldots,k \\ & x\in X & \\ & \alpha\in\Re & \end{array} \tag{7}$$

If q is a non-attainable point then the optimal solution to (7) is the Pareto optimal solution closest to q according to the (augmented) Tchebycheff metric. If q is attainable, the scalarizing program (7) does not minimize a distance. Instead, it tries to improve the reference point and consequently a Pareto optimal solution is produced. Actually, this is an *achievement scalarizing program* and the outcome is always a Pareto optimal solution.

Several other related scalarizing programs have been proposed in the literature, in particular, the weighted Tchebycheff scalarizing program [8, 32] which has been widely used. In general, a fixed reference point (which must be non-attainable) is used and the weights are the controlling parameters. So, the main difference between the achievement scalarizing program (7) and the weighted Tchebycheff scalarizing program is the dependence on controlling parameters, the reference levels in the former case and the weights in the latter one.

Whatever the controlling parameters are (weights, reference levels or both), there might exist ranges of parameter values that lead to the same Pareto optimal solution. Therefore, not only the effectiveness of a multiobjective method relies on the availability of a suitable single-objective optimization algorithm, but also depends on the way the parameters are changed. In *generating* methods, which aim to generate the whole set or a representative subset of Pareto optimal solutions, the variation of parameters is controlled by the algorithm. In *interactive* methods, which alternate computation phases with decision making phases, the variation of parameters results from preference information provided by the decision maker. In both types of methods, sensitivity information on the variation of parameters can be very useful to avoid computing the same Pareto optimal solution more than once. This kind of information is especially relevant in problems with discrete variables or discontinuities in the Pareto region, which is the case of the MOBLLP.

Alves and Clímaco [2, 3] developed an interactive reference point procedure and software for the multiobjective mixed-integer linear problem (MOMILP), which uses the scalarizing program (7) to compute Pareto optimal solutions. The procedure is mainly devoted to perform *directional searches* by solving the parametric optimization problem (7) with the parameter vector q. The mixed-integer scalarizing programs are successively solved by a branch-and-bound method using a single tree. Sensitivity analysis and post-optimality techniques have been developed to change automatically the reference point throughout a directional search and to use the previous branch-and-bound tree as a starting structure to solve the next scalarizing programs.

This approach can be applied to the MOBLLP after reformulating the problem as a MOMILP. Although the procedure has been developed to be an *interactive* method, it can be further used to generate the whole Pareto frontier of bi-objective bilevel linear problems, thus providing a *generating* method. The steps of this algorithm are presented below after a brief description of the interactive procedure.

So, let us start by introducing the interactive procedure proposed in [2]. Firstly, the *payoff* table of the MOMILP may be computed. This is an optional step that aims at providing some initial useful information for the decision maker (DM) that helps him/her to choose a first reference point. The *payoff* table is of the form of Table 3 where the rows are criterion vectors resulting from individually maximizing each one of the objectives. A two phase optimization process is used to avoid weakly Pareto optimal solutions in the payoff table. This process is also referred to as a lexicographic optimization approach and consists of first solving for each $i = 1,\ldots,k$, $\max_x \{z_i(x) : x \in X\}$ and then the program $\max_x \{\sum_{j \neq i} z_j(x) : x \in X, z_i(x) \geq z_i^*\}$, where z_i^* is the maximum of $z_i(x)$ obtained in the first optimization phase. Let x^i be the computed Pareto optimal solution for the objective $z_i(x)$. Its criterion vector is $z^i = z(x^i) = (z_1^i,\ldots,z_i^i,\ldots,z_k^i)$, where $z_i^i = z_i^*$, and it constitutes the i^{th} row of the payoff table.
The main diagonal of the *payoff* table is formed by the so-called ideal point $z^* = (z_1^*, z_2^*,\ldots, z_k^*)$, which is suggested to the DM to be the first reference point.

Table 3. *Payoff* table.

	z_1	z_2	…	z_k
z^1	z_1^*	z_2^1	…	z_k^1
z^2	z_1^2	z_2^*	…	z_k^2
…	…	…	…	…
z^k	z_1^k	z_2^k	…	z_k^*

The main cycle of the algorithm is as follows:

- Construct a new reference point $\overline{q}$.
- Compute a Pareto optimal solution by solving the mixed-integer program (7) with $q = \overline{q}$.
- Terminate when the DM is satisfied and does not want to continue the search for new Pareto optimal solutions.

A new reference point can be chosen by the DM or it is automatically changed by the procedure (except in the first iteration) if the DM wants to perform a *directional search*. In the latter case, the DM just specifies an objective function, say z_j, he/she wants to improve with respect to the previous Pareto optimal solution. Then, the procedure increases the j^{th} component of the reference point ($\overline{q}_j$) keeping the other components equal. The amount by which $\overline{q}_j$ is increased is determined by sensitivity analysis using information provided by the previous branch-and-bound tree. This process ensures that the next Pareto optimal solution is close to, but different from, the previous Pareto optimal solution. The next computing phase does not start a new branch-and-bound tree, but rather uses the previous one to proceed to the optimization of the new scalarizing program.

The algorithm of this *interactive* reference point procedure can be stated as follows.

Step 0 [optional]. Compute the payoff table of the MOMILP.

Step 1. Ask the DM to specify a reference point $\overline{q} \in \Re^k$.

At the first interaction it is proposed by default the ideal point of the MOMILP (or the ideal point of the linear relaxation of the problem if the Step 0 has not been performed).

Solve the mixed-integer program (7) with $q = \overline{q}$ using a branch-and-bound method to obtain a Pareto optimal solution.

Step 2. If the DM does not want to compute more Pareto optimal solutions, stop.

Otherwise, if the DM is willing to indicate explicitly a new reference point, return to *Step 1*.

Else, go to *Step 3*.

Step 3. Ask the DM to choose one of the objectives he/she wishes to improve in relation to the previous Pareto optimal solution. Let z_j be the objective specified by the DM.

A *directional search* is carried out by considering reference points of the form $(\overline{q}_1,...,\overline{q}_j + \theta_j,...,\overline{q}_k)$ with $\theta_j > 0$ to produce a sequence of Pareto optimal solutions that successively improve z_j. The computation of new solutions throughout this direction stops when the DM wishes or a Pareto optimal solution that optimizes z_j has been reached.

Return to *Step 2*.

The core of this algorithm is *Step 3* and the way a *directional search* is performed. It consists of optimizing successive scalarizing programs (7) that only differ in the right-hand side of the j^{th} constraint (a special constraint that results from the integration of the j^{th} objective into the scalarizing program). Postoptimality techniques have been developed to perform this task. This is an iterative process with two main phases:

(*i*) sensitivity analysis,

(*ii*) updating the branch-and-bound tree.

The sensitivity analysis (*i*) returns a parameter value $\overline{\theta}_j > 0$ such that the structure of the previous branch-and-bound tree remains unchanged for variations in $\overline{q}_j$ up to $\overline{q}_j + \overline{\theta}_j$. This means that reference points $(\overline{q}_1,...,\overline{q}_j + \theta_j,...,\overline{q}_k)$ with $\theta_j \leq \overline{\theta}_j$ either lead to the same Pareto optimal solution or lead to different Pareto optimal solutions that are easily computed because they come from the same node of the branch-and-bound tree. In the latter case, distinct Pareto optimal solutions may be computed for different specific parameter values under $\overline{\theta}_j$ and these solutions are presented to the DM (who can interactively control the proximity of solutions he/she wants to visualize). In both cases, the branch-and-bound tree is then updated (*ii*) for θ_j slightly over $\overline{\theta}_j$ and a Pareto optimal solution is produced. It may happen that this solution is the same as the last one because the $\overline{\theta}_j$ returned by the sensitivity analysis can be only a lower bound for the true maximum value of the parameter. In that case (which occurs more often in all-integer programs than in mixed-integer models) the process automatically returns to (*i*), and the iterative process finishes when a new Pareto optimal solution is obtained in (*ii*), which is then presented to the DM.

A detailed description of the sensitivity analysis as well as the updating process of the branch-and-bound tree can be found in [2].

Concerning the application of the algorithm to the MOBLLP and, in particular, to generate the Pareto frontier of a bi-objective problem, the following features may be highlighted.

– The user can define a stepsize μ, which represents the maximum variation (in percentage) that the DM wishes for the value of an objective function when continuous Pareto optimal solutions are computed.

– The procedure recognizes when it reaches a Pareto optimal solution that maximizes one of the objectives of the multiobjective problem (even if the payoff table was not initially computed). Therefore, if a directional search is performed to improve z_j and a Pareto optimal solution maximizing z_j is at hand, the procedure indicates that no more improvement in this objective function is possible and the directional search finishes.

Now suppose that we wish to examine the whole Pareto frontier of a bi-objective problem. Then we can either start in the optimum of the first objective and perform a directional search in order to improve the second objective, or we can do the reverse. Since the increase of an objective function implies the decrease of the other, the Pareto frontier is fully determined using such an approach, except for a gap between continuous solutions which is controlled by the stepsize μ. Therefore, we must only ensure that the initial reference point leads to a Pareto optimal solution that maximizes one of the objectives. Without loss of generality, consider that we start in the optimum of $z_1(x)$.

So, $z_1(x)$ is firstly maximized using a lexicographic optimization approach to ensure that a Pareto optimal solution, say x^1, is obtained. Let $z^1 = (z_1^1, z_2^1)$ be the corresponding nondominated criterion vector, where $z_1^1 = z_1^*$ is the maximum value of $z_1(x)$ over X. Solution x^1 also optimizes the achievement scalarizing program (7) for the reference point $q = z^1$ provided that the constant ρ in (7) is set small enough, i.e. it satisfies $\rho < \rho'$ for a certain ρ'. In fact the following result holds:

Proposition 3 – Consider the bi-objective programming problem

$$\begin{array}{ll} \max\limits_{x} & z_1(x) \\ \max\limits_{x} & z_2(x) \\ \text{s.t.} & x \in X \end{array}$$

Let Z denote the feasible region in the criterion space (i.e. image of X) and Z_{nd} its subset corresponding to the nondominated criterion vectors. Let x^1 be a Pareto optimal solution that maximizes $z_1(x)$ in the feasible region X and $z^1 \in Z_{nd}$ be its criterion vector. Then, x^1 optimizes the following scalarizing program for which the criterion vector z^1 is uniquely determined:

$$\min_{x \in X} \left\{ \max_{i=1,2} \{q_i - z_i(x)\} - \rho \sum_{i=1}^{2} z_i(x) \right\}$$

with $q = z^1$ and

$$0 < \rho < \min_{\bar{z} \in Z_{nd} \setminus \{z^1\}} \left\{ \frac{z_1^1 - \bar{z}_1}{\sum_{i=1}^{2} (\bar{z}_i - z_i^1)} : \sum_{i=1}^{2} (\bar{z}_i - z_i^1) > 0 \right\}$$

Proof. Consider a feasible solution $\bar{x} \in X$ whose criterion vector is $\bar{z} \in Z$ such that $\bar{z} = \left(z_1(\bar{x}), z_2(\bar{x})\right) \neq z^1$.

Since $z_1^1 = z_1^*$ is the maximum of $z_1(x)$ and z^1 is a nondominated vector, then $\bar{\alpha} = \max_{i=1,2} \left\{q_i - z_i(\bar{x})\right\} = \max_{i=1,2} \left\{z_1^1 - \bar{z}_1,\, z_2^1 - \bar{z}_2\right\}$ is strictly positive for $\bar{z} \in Z \backslash \{ z^1 \}$.

On the other hand, $\alpha^1 = \max_{i=1,2} \left\{q_i - z_i(x^1)\right\} = 0$ for $q = z^1$.

Thus, z^1 is the unique criterion vector that optimizes the scalarizing program for $q = z^1$ if

$$-\rho \sum_{i=1}^{2} z_i^1 < \bar{\alpha} - \rho \sum_{i=1}^{2} \bar{z}_i \quad , \forall\, \bar{z} \in Z \backslash \{ z^1 \}$$

This means that we must have $\rho < \dfrac{\bar{\alpha}}{\sum_{i=1}^{2} (\bar{z}_i - z_i^1)}$ whenever $\sum_{i=1}^{2} (\bar{z}_i - z_i^1) > 0$, for all $\bar{z} \in Z \backslash \{ z^1 \}$.

If $\bar{z}$ is nondominated, then $z_1^1 > \bar{z}_1$, $z_2^1 < \bar{z}_2$ and $\bar{\alpha} = z_1^1 - \bar{z}_1$. If $\bar{z}$ is dominated then $\bar{\alpha} \geq z_1^1 - \bar{z}_1$.

Hence, it suffices for ρ to be defined as in this proposition for z^1 uniquely optimizing the scalarizing program with $q = z^1$.

Note that, in practice, a small positive value of ρ, e.g. 10^{-3} or 10^{-4} is normally suitable. Moreover, in this particular case, if ρ was not chosen appropriately because it was set too high, the obtained solution would not be the solution that optimizes $z_1(x)$, i.e. z^1, but rather another nondominated solution close to it. Since z^1 is known in advance, the difference would be detected and the search could then be restarted with a lower ρ.

Once a bi-objective bilevel linear programming problem has been transformed into a bi-objective mixed-integer linear program, the following *generating* algorithm can be used to characterize its Pareto frontier.

Step 0. Compute the *payoff* table of the bi-objective problem or just a Pareto optimal solution that maximizes $z_1(x)$. Let $z^1 = (z_1^1, z_2^1)$ be its criterion vector.

Step 1. Define the first reference point as $q = z^1$.
Solve the mixed-integer program (7) using a branch-and-bound method as in Step 1 of the interactive algorithm.

Step 2. Choose $z_2(x)$ to be improved.
Choose a stepsize $\mu > 0$ that defines an acceptable gap between continuous Pareto optimal solutions.
Perform a directional search as in Step 3 of the interactive algorithm stopping when a Pareto optimal solution that maximizes $z_2(x)$ is reached.

In this algorithm, μ represents the maximum value that is allowed for the ratio $(z_2^{new} - z_2^{prev}) / (\tilde{z}_2^* - z_2^1)$, where z^{new} and z^{prev} are the criterion vectors of two continuous Pareto optimal solutions, the *new* and the *previous* one, respectively; $\tilde{z}_2^*$ is an approximation for the maximum of z_2 (e.g. the maximum of z_2 in the linear relaxation of

the problem) or its true maximum value if the *payoff* table has been fully computed in Step 0.
The algorithm has been stated for starting at the optimum of z_1 and finishing at the optimum of z_2. Naturally, starting at the optimum of z_2 and selecting then the first objective to be improved is another possibility to compute the Pareto frontier of the bi-objective problem. In this case, μ is used for restricting differences in z_1.

6. Two examples of the application of the MOMILP procedure to bi-objective bilevel linear problems

Consider again the bi-objective bilevel linear problem presented in *example 1*, which is graphically depicted in Figure 1. This problem is firstly reformulated as the following bi-objective linear problem with complementarity constraints.

$$
\begin{aligned}
&\max_{x,y} F_1(x,y) = -2x\\
&\max_{x,y} F_2(x,y) = -x + 5y\\
\text{s.t.}\quad & x - 2y \le 4\\
& 2x - y \le 24\\
& 3x + 4y \le 96\\
& x + 7y \le 126\\
& -4x + 5y \le 65\\
& x + 4y \ge 8\\
& 2\lambda_1 + \lambda_2 - 4\lambda_3 - 7\lambda_4 - 5\lambda_5 + 4\lambda_6 \le 1\\
& (x - 2y - 4).\lambda_1 = 0\\
& (2x - y - 24).\lambda_2 = 0\\
& (3x + 4y - 96).\lambda_3 = 0\\
& (x + 7y - 126).\lambda_4 = 0\\
& (-4x + 5y - 65).\lambda_5 = 0\\
& (-x - 4y + 8).\lambda_6 = 0\\
& (2\lambda_1 + \lambda_2 - 4\lambda_3 - 7\lambda_4 - 5\lambda_5 + 4\lambda_6 - 1).\ y = 0\\
& x, y \ge 0\\
& \lambda_i \ge 0,\ i = 1,\ldots,6
\end{aligned}
$$

Next, the problem is reformulated as the following MOMILP.

$$
\begin{aligned}
&\max_{x,y} F_1(x,y) = -2x\\
&\max_{x,y} F_2(x,y) = -x + 5y\\
\text{s.t.}\quad & x - 2y \le 4 && x - 2y + \mathrm{M}u_1 \ge 4\\
& 2x - y \le 24 && \lambda_1 + \mathrm{M}u_1 \le \mathrm{M}\\
& 3x + 4y \le 96 && 2x - y + \mathrm{M}u_2 \ge 24\\
& x + 7y \le 126 && \lambda_2 + \mathrm{M}u_2 \le \mathrm{M}\\
& -4x + 5y \le 65 && 3x + 4y + \mathrm{M}u_3 \ge 96\\
& x + 4y \ge 8 && \lambda_3 + \mathrm{M}u_3 \le \mathrm{M}\\
& 2\lambda_1 + \lambda_2 - 4\lambda_3 - 7\lambda_4 - 5\lambda_5 + 4\lambda_6 \le 1 && x + 7y + \mathrm{M}u_4 \ge 126\\
& x, y \ge 0 && \lambda_4 + \mathrm{M}u_4 \le \mathrm{M}\\
& \lambda_i \ge 0,\ i = 1,\ldots,6 && -4x + 5y + \mathrm{M}u_5 \ge 65\\
& && \lambda_5 + \mathrm{M}u_5 \le \mathrm{M}\\
& && -x - 4y + \mathrm{M}u_6 \ge -8\\
& && \lambda_6 + \mathrm{M}u_6 \le \mathrm{M}
\end{aligned}
$$

$$2\lambda_1 + \lambda_2 - 4\lambda_3 - 7\lambda_4 - 5\lambda_5 + 4\lambda_6 + \mathrm{M}v_1 \geq 1$$
$$y + \mathrm{M}v_1 \leq \mathrm{M}$$
$$u_i \in \{0,1\}\ ,\ i = 1,\ldots,6$$
$$v_1 \in \{0,1\}$$

where M >0 is a suitable large number.

This formulation (considering M=150) has been introduced into the MOMILP software that implements the methodology described in the previous section. This software has been developed in Delphi 2007 for Windows. It upgrades the procedure described above which has been previously implemented within a broader decision support system [3].
The *generating* algorithm is applied to this problem. The *payoff* table is firstly computed (Table 4). It is composed by the criterion vectors of the Pareto optimal solutions that optimize individually each objective function, which have been denoted by D and A in Figure 1, respectively.

Table 4. *Payoff* table of *example 1*.

	F_1	F_2
D	0	10
A	-34.9091	37.09091

The reference point q = (0, 10) is chosen to start the search for Pareto optimal solutions and the corresponding mixed-integer scalarizing program (7) is solved by the branch-and-bound method. Its optimal solution is the Pareto optimal solution that maximizes F_1 whose criterion vector is z=(0,10). Let z denote a criterion vector (F_1, F_2) of any Pareto optimal solution.
The directional search is then selected to compute Pareto optimal solutions that successively improve F_2, and the stepsize μ=0.5% is chosen. Next, the procedure performs a sensitivity analysis on the previous branch-and-bound tree and changes the reference point to q =(0, 36.773). The branch-and-bound tree is updated to find the optimal solution of the scalarizing program for the new q and a new Pareto optimal solution is obtained whose criterion vector is z =(-26.727, 10.045). This Pareto optimal solution is nearby the weakly Pareto optimal solution denoted by C in Figure 1. Note that the procedure needs to make a major change in q_2 in order to "jump" the discontinuity in the Pareto region. In this case the stepwise μ cannot be fulfilled, as the solutions are not continuous.
The next solution throughout the directional search has z =(-27.908, 10.181) and it is found using the reference point q =(0, 37.089). Then, z =(-27.0886, 10.3164) is computed using q =(0, 37.405). The directional search continues in the same way by computing very close Pareto optimal solutions until the optimum of F_2 is reached when q =(0, 72.1). In this directional search a total of 202 Pareto optimal solutions are computed with a total CPU time < 0.001 seconds (on a computer with Core 2 CPU 6700, 2.66 GHz, 2 GB of RAM). Figure 3 shows the criterion points for all the Pareto optimal solutions computed by the algorithm. As was expected, apart from the scale, this graph is similar to the one presented in Figure 2 (Pareto frontier that has been produced by a graphical analysis).

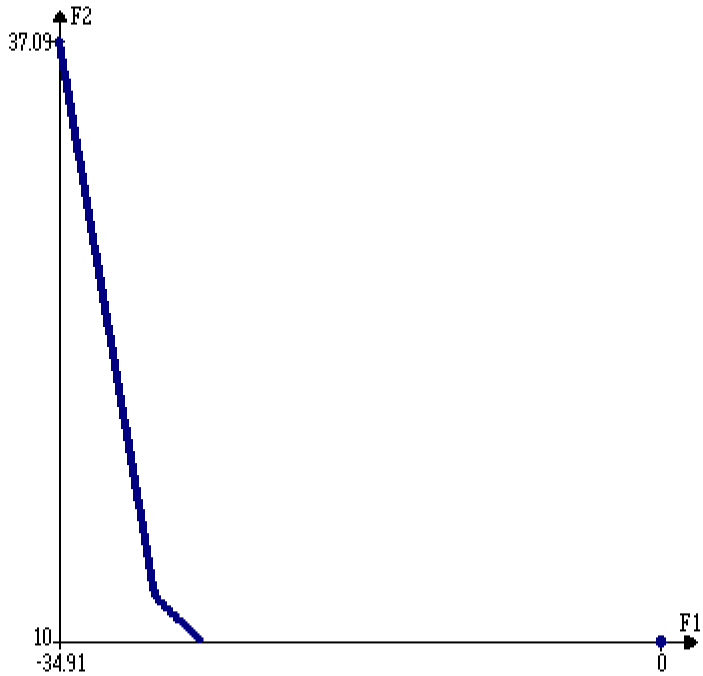

Figure 3. Nondominated criterion points produced by the MOMILP software for the *example 1*.

Let us present another example.

A larger dimensional bi-objective bilevel linear problem (*example 3*) was randomly generated and solved. The total number of variables is 50 where 20 are controlled by the follower (thus, n_1 =30 and n_2=20). The problem was generated in a manner similar to that of Glackin et al. [22]. The number of constraints is 0.4 times the total number of variables (we considered 50% in the upper level and 50% in the lower level, thus m_1 =10 and m_2 =10). The coefficients of the matrices A^1, A^2 and B^2 range from -15 to 45 with a fraction of nonzero entries of 0.4. The right-hand side values of b^1 and b^2 are uniformly distributed between 0 and 50, and we considered that all constraints are inequality constraints of type '$\leq$' (to facilitate feasibility). The coefficients of the upper and lower level objective functions are uniformly distributed between -20 and 20.
In order to get the reformulation of the MOBLLP n_2 constraints and m_2 continuous variables are first added to transform it into a single level problem with complementarity constraints. Then $2(m_2 + n_2)$ constraints and (m_2+n_2) binary variables are further included to obtain a MOMILP. Thus, the bi-objective mixed-integer problem to be addressed has 60 continuous variables, 30 binary variables and 100 constraints. We fixed the constant M to 100 times the largest coefficient of the model (i.e. M=5000).

We performed two directional searches with this problem, one starting at the optimum of F_1 and then searching for Pareto optimal solutions that successively improve F_2, and the other is the reverse search (starting at the optimum of F_2 and then improving F_1). These computations were also useful to test the accuracy of the value assigned to M. We found that all the computed solutions satisfy the complementarity conditions stated in the corresponding bi-objective linear problem with complementarity constraints.
In both directional searches we set the stepsize μ equal to 0.5%. This defines a gap between two consecutive solutions, which is measured by the relative difference in the values of F_2 in the first search and in the values of F_1 in the second search.
In the first directional search, 304 Pareto optimal solutions were computed. The first solution required a CPU time of 0.13 seconds while all the others required a total time

of 2.05 seconds (a mean less than 0.007 seconds per solution). This discrepancy between the first time and the others is due to the computing process which does not start a new branch-and-bound tree in each new optimization phase, but rather updates the previous one.

In the second directional search, a total of 272 Pareto optimal solutions were obtained with a CPU time of 0.30 seconds for computing the first solution and 2.03 seconds for computing all the others.

Figure 4 shows the criterion points of the Pareto optimal solutions obtained in *(a)* the first search (from the optimum of F_1 to the optimum of F_2) and in *(b)* the second search (from the optimum of F_2 to the optimum of F_1). The proximity of the solutions is different in *(a)* and *(b)* although the same stepsize has been used, because this parameter restricts the distance between two solutions in one axis, F_2 or F_1 respectively.

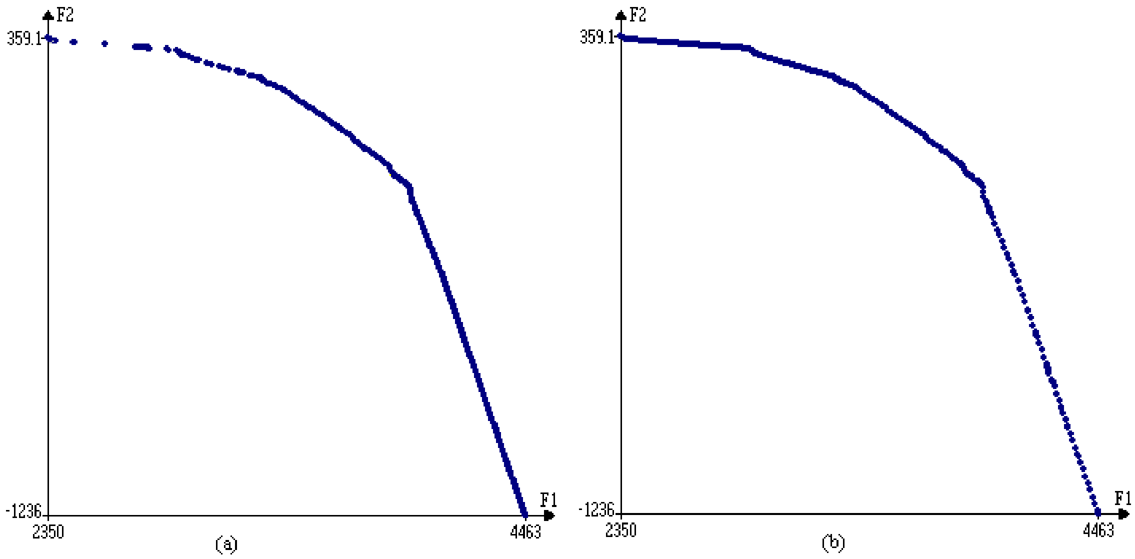

Figure 4. Nondominated criterion points produced by the MOMILP software for the *example 3*. *(a)* Directional search from the optimum of F_1 to the optimum of F_2 *(b)* Directional search from the optimum of F_2 to the optimum of F_1.

7. Conclusions

In this paper we have studied the bilevel linear programming problem with multiple objectives at the upper level (MOBLLP). We have further discussed the potentialities of a reference point algorithm to solve the MOBLLP and its use as a generating method for bi-objective problems. It has been shown that the procedure can fully determine the Pareto frontier of a bi-objective problem except for a gap between continuous Pareto optimal solutions, which can be as small as the user wishes.

Although the proposed approach is viable to solve any MOBLLP, it requires the reformulation of the problem as a multiobjective mixed 0-1 linear programming problem. The MOBLLP is firstly transformed into a multiobjective linear program with complementarity constraints and these constraints are then converted into linear constraints with binary variables. This latter conversion needs the addition of $2(m_2 + n_2)$ constraints and (m_2+n_2) binary variables to the problem, where m_2 and n_2 are the numbers of lower-level constraints and variables, respectively. Furthermore, it may be difficult to define a suitable large number for the constant M in this formulation. Therefore, we aim to develop another procedure which can be applied directly to multiobjective linear problems with complementarity constraints and also exploits the enumerative tree to solve successive reference point scalarizing programs.

References

[1] M.A. Abo-Sinna, and A.I. Baky, *Interactive balance space approach for solving multi-level multi-objective programming problems*, Inf. Sci. 177 (2007), pp. 3397-3410.

[2] M.J. Alves, and J. Clímaco, *An interactive reference point approach for multiobjective mixed-integer programming using branch and bound*, European J. Oper. Res. 124 (2000), pp. 478-494.

[3] M.J. Alves, and J. Clímaco, *A note on a decision support system for multiobjective integer and mixed-integer programming problems*, European J. Oper. Res. 155 (2004), pp. 258-265.

[4] C. Audet, J. Haddad, and G. Savard, *A note on the definition of a linear bilevel programming solution*, Appl. Math. Comput. 181 (2006), pp. 351-355.

[5] C. Audet, P. Hansen, B. Jaumard, and G. Savard. *Links between linear bilevel and mixed 0-1 programming problems*, J. Optim. Theory Appl. 93 (1997), pp. 273-300.

[6] J.F. Bard, *Practical Bilevel Optimization: Algorithms and Applications*, Kluwer Academic Publishers, Dordrecht, 1999.

[7] J. Benoist, *Connectedness of the efficient set for strictly quasiconcave sets*, J. Optim. Theory Appl. 96 (1998), pp. 627-654.

[8] V.J. Bowman, *On the relationship of the Tchebycheff norm and the efficient frontier of multiple-criteria objectives*, in *Multiple Criteria Decision Making*, H. Thiriez and S. Zionts, eds., Lecture Notes in Econom. and Math. Systems 130 (1976), Springer, Berlin, pp. 76-86.

[9] D. Cao, and L.C. Leung, *A partial cooperation model for non-unique linear two-level decision problems*, European J. Oper. Res. 140 (2002), pp. 134-141.

[10] B. Colson, P. Marcotte, and G. Savard, *Bilevel programming: a survey*, 4OR 3 (2005), pp. 87-107.

[11] K. Deb, and A. Sinha, *Solving bilevel muti-objective optimization problems using evolutionary algorithms*, M. Ehrgott et al., eds, Lect. Notes Comput. Sci. 5467 (2009), Springer-Verlag, Berlin, pp. 110-124.

[12] S. Dempe, *Foundations of Bilevel Programming*, Kluwer Academic Publishers, Dordrecht, 2002.

[13] S. Dempe, *Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints*, Optimization 52 (2003), pp. 333-359.

[14] M. Ehrgott, *Multicriteria Optimization*, Springer, Berlin, 2005.

[15] G. Eichfelder, *Multiobjective bilevel optimization*, Math. Program., Ser. A (2008), DOI 10.1007/s10107-008-0259-0.

[16] G. Eichfelder, *Solving nonlinear multiobjective bilevel optimization problems with coupled upper level constraints*, Preprint-series of the Institute of Applied Methematics, no 320, Univ. Erlangen-Nürnberg, Germany, 2007.

[17] G. Eichfelder, *Adaptive Scalarization Methods in Multiobjective Optimization*, Springer, Berlin, 2008.

[18] E. Erkut, and F. Gzara, *Solving the hazmat transport network design problem*, Comput. Oper. Res. 35 (2008), pp. 2234-2247.

[19] J. Fliege, *Gap-free computation of Pareto-points by quadratic scalarizations*, Math. Methods Oper. Res. 59 (2004), pp. 69-89.

[20] J. Fortuny-Amat, and B. McCarl. *A representation and economic interpretation of a two-level programmimg problem*. J. Oper. Res. Society 32 (1981), pp. 783-792.

[21] J. Fülöp, *On the equivalence between a linear bilevel programming problem and linear optimization over the efficient set*, Working paper No. WPO 93-1, Laboratory of Operations Research and Decision Systems, Computer and Automation Institute, Hungarian Academy of Sciences, 1993.

[22] J. Glackin, J. G. Ecker and M. Kupferschmid, *Solving bilevel linear programs using multiple objective linear programming*, J. Optim. Theory Appl. 140 (2009), pp. 197-212.

[23] J. Júdice, and A. Faustino, A *sequential LCP algorithm for bilevel linear programming*, Ann. Oper. Res. 34 (1992), pp. 89-106.

[24] L. Mallozi, and J. Morgan, *Hierarchical systems with weighted reaction set*, in *Nonlinear optimization and Applications*, G. D. Pillo and F. Giannessi, eds., Plenum Press, New York, 1996, pp. 271-282.

[25] A. G. Mersha, and S. Dempe, *Linear bilevel programming with upper level constraints depending on the lower level solution*, Appl. Math. Comput. 180 (2006), pp. 247-254.

[26] K.M. Miettinen, *Nonlinear Multiobjective Optimization,* Kluwer Academic Publishers, 1999.

[27] A. Migdalas, P.M. Pardalos, and P. Varbrand (eds.), *Multi-level Optimization: algorithms and applications*, Kluwer Academic Press, 1998.

[28] I. Nishizaki, and M. Sakawa, *Stakelberg solutions to multiobjective two-level linear programming problems*, J. Optim. Theory Appl. 103 (1999), pp. 161-182.

[29] X. Shi, and H. Xia, *Interactive bilevel multi-objective decision making*, J. Oper. Res. Society 48 (1997), pp. 943-949.

[30] X. Shi, and H. Xia, *Model and interactive algorithm of bi-level multi-objective decision-making with multiple interconnected decision makers*, J. Multi-Criteria Decision Analysis 10 (2001), pp. 27-34.

[31] R. Steuer. *Multiple Criteria Optimization: Theory, Computation and Application*, Wiley, New York, 1986.

[32] R. Steuer, and E.-U. Choo. *An interactive weighted Tchebycheff procedure for multiple objective programming*, Math. Prog. 26 (1983), pp. 326-344.

[33] A. Wierzbicki, *The use of reference objectives in multiobjective optimization*, in *Multiple Criteria Decision Making, Theory and Application*, G. Fandel and T. Gal, eds., Lect. Notes in Econom. and Math. Systems 177 (1980), Springer-Verlag, Berlin, pp. 468-486.

[34] A. Wierzbicki, *Reference points in vector optimization and decision support*. Interim Report, IR-98-017, IIASA, Laxenburg, Austria, 1998.

[35] Y. Yin, *Multiobjective bilevel optimization for transportation planning and management problems*, J. Advanced Transportation 36 (2002), pp. 93-105.

[36] G. Zhang, J. Lu, and T. Dillon, *Decentralized multi-objective bilevel decision making with fuzzy demands*. Knowledge-Based Systems 20 (2007), pp. 495-507.

Appendix

Proposition 1 − Consider the single or multi-objective bilevel linear programming problem MOBLLP defined in (1) with $k \geq 1$ and the definitions above for S and IR. Then $(\bar{x}, \bar{y}) \in IR$ if and only if $(\bar{x}, \bar{y}) \in E$ where E is the set of Pareto optimal solutions of a MOLP with n_1+2 objective functions, defined as:

$$\max_{x,y} \quad f(y) = d^2 y$$

$$\max_{x,y} \quad v^i x \quad , \qquad i = 1, \ldots, n_1+1$$

$$\text{s.t.} \qquad (x,y) \in S$$

with $\{v^1, v^2, \ldots, v^{n_1+1}\}$ a minimal set of generators of the cone $\Re^{n_1}$ (i.e., any point of $\Re^{n_1}$ can be reached by some nonnegative linear combination of the v^i).

Proof.

a) First we assume that $(\bar{x}, \bar{y}) \in IR$. We want to prove that $(\bar{x}, \bar{y}) \in E$.

As $(\bar{x}, \bar{y}) \in IR = \{(x, y) : (x, y) \in S,\ y \in P(x)\}$, then $(\bar{x}, \bar{y}) \in S$ and $\bar{y} \in P(\bar{x})$, which means that $\bar{y}$ maximizes f over $S(\bar{x}) = \{y : B^2 y \leq b^2 - A^2 \bar{x},\ y \geq 0\}$. Suppose that $(\bar{x}, \bar{y}) \notin E$. Then, there exists another $(x, y) \in S$ such that $d^2 y \geq d^2 \bar{y}$ and $v^i x \geq v^i \bar{x}$ for all $i = 1, \ldots, n_1+1$, with at least one strict inequality.

Since $\{v^1, v^2, \ldots, v^{n_1+1}\}$ is a set of generators of the cone $\Re^{n_1}$, for each n_1-dimensional unit vector $e^j = (0,0,\ldots,1,\ldots,0)$ with j^{th} entry equal to 1 and all the others equal to zero, there are constants $\alpha_1^{(j)}, \alpha_2^{(j)}, \ldots, \alpha_{n_1+1}^{(j)} \geq 0$ such that $e^j = \sum_{i=1}^{n_1+1} \alpha_i^{(j)} v^i$. Analogously, there are $\beta_1^{(j)}, \beta_2^{(j)}, \ldots, \beta_{n_1+1}^{(j)} \geq 0$ such that $-e^j = \sum_{i=1}^{n_1+1} \beta_i^{(j)} v^i$. If $v^i x \geq v^i \bar{x}$ for all i, then $\sum_{i=1}^{n_1+1} \alpha_i^{(j)} v^i (x - \bar{x}) = e^j (x - \bar{x}) = x_j - \bar{x}_j \geq 0$ and $\sum_{i=1}^{n_1+1} \beta_i^{(j)} v^i (x - \bar{x}) = -e^j (x - \bar{x}) = -x_j + \bar{x}_j \geq 0$. So, $x_j = \bar{x}_j$ for every $j = 1, \ldots, n_1$. Thus, the strict inequality must be regarding the first objective function, that is, $f(y) > f(\bar{y})$. However, this contradicts the fact that $\bar{y} \in P(\bar{x})$. Hence, $(\bar{x}, \bar{y}) \in E$.

b) Now we assume that $(\bar{x}, \bar{y}) \in E$ and we want to prove that $(\bar{x}, \bar{y}) \in IR$.

Suppose that $(\bar{x}, \bar{y}) \notin IR$. As $(\bar{x}, \bar{y}) \in E$, then $(\bar{x}, \bar{y}) \in S$. Therefore, the condition $(\bar{x}, \bar{y}) \notin IR$ holds only if $\bar{y} \notin P(\bar{x})$, i.e. if there exists another $(\bar{x}, y) \in S$ such that $f(y) > f(\bar{y})$. Under these circumstances, in the MOLP the criterion vector of $(\bar{x}, y)$ dominates the criterion vector of $(\bar{x}, \bar{y})$, because it is superior in the first objective function and equal in the others. This contradicts the hypothesis that $(\bar{x}, \bar{y}) \in E$. Hence, $(\bar{x}, \bar{y}) \in IR$.