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

aFac. de Economia, Universidade de Coimbra / INESC, Coimbra, Portugal;  
bFreiberg University, Freiberg, Germany;  
cDep. 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. We examine some properties of this 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. An illustrative numerical example is presented 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, 11, 12]. 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 cost construction 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 on the bilevel problem, little research work has been done thus far on multiobjective bilevel problems.

* Corresponding author. Email: mjalves@fe.uc.pt
Yin [32] and Erkut and Gzara [17] 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. 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 [26, 27] present an interactive algorithm for nonlinear bilevel multiobjective problems. 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 in [1] with some modifications in assigning satisfactoriness to each objective function at all levels of the problem.

Zhang et al. [33] have developed an approximation branch-and-bound algorithm to deal with multiobjective bilevel decision problems with fuzzy demands. Eichfelder [14, 16] 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 [15] these results have been extended to problems with upper level constraints that depend on the lower level variables.

Nishizaki and Sakawa [25] have 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 [25] 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.

Nishizaki and Sakawa [25] 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 [21], 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 [11].

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 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 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 give particular attention to the bi-objective case.

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. An interactive reference point procedure developed by Alves and Climaco [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 relationship 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 presenting the interactive procedure [2] for multiobjective mixed-integer linear programming and proposes a generating algorithm for the bi-objective case. An illustrative example of the application of the proposed algorithm to a bi-objective bilevel problem is shown 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:

\[
\begin{align*}
\max_x & \quad F_1(x, y) = c_1^1 x + d_1^1 y \\
\vdots \\
\max_x & \quad F_k(x, y) = c_k^1 x + d_k^1 y \\
\text{s.t.} & \quad A^1 x \leq b^1 \\
& \quad x \geq 0 \\
& \quad \max_y f(y) = d^2 y \\
\text{s.t.} & \quad A^2 x + B^2 y \leq b^2 \\
& \quad y \geq 0
\end{align*}
\]  
(1)

where \( x \in \mathbb{R}^n \) and \( y \in \mathbb{R}^m \) are the upper level and lower level decision variables, respectively, \( k \) is the number of objective functions of the leader, \( c_i^1 \in \mathbb{R}^n \), \( d_i^1 \in \mathbb{R}^m \), \( i = 1, \ldots, k \), \( d^2 \in \mathbb{R}^m \), \( A^1 \in \mathbb{R}^{m \times n} \), \( b^1 \in \mathbb{R}^m \), \( A^2 \in \mathbb{R}^{m_2 \times n_2} \), \( B^2 \in \mathbb{R}^{m_2 \times n_2} \), \( b^2 \in \mathbb{R}^{m_2} \) and \( c \) 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 by selecting an optimal \( x \), and the lower level decision maker (follower) responds by selecting an optimal \( y \). 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:

\[
S = \{ (x, y) : A^1 x \leq b^1, A^2 x + B^2 y \leq b^2, x \geq 0, y \geq 0 \}
\]

\( P(x) \) is the follower’s rational reaction set to a given \( x \):

\[
P(x) = \{ y : \arg \max \{ f(y') : B^2 y' \leq b^2 - A^2 x, y' \geq 0 \} \}
\]

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

\[
IR = \{ (x, y) : (x, y) \in S, y \in P(x) \}
\]

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

\[
\begin{align*}
\max & \quad F_1(x, y) = c_1^1 x + d_1^1 y \\
\vdots \\
\max & \quad F_k(x, y) = c_k^1 x + d_k^1 y \\
\text{s.t.} & \quad (x, y) \in IR
\end{align*}
\]  
(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, 22]. 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. The general theory for multicriteria (multiobjective) optimization can be found in [13] or [28]. To facilitate the exposition of the concepts, consider a multiobjective optimization problem defined generically as follows:

$$\begin{align*}
\max & \quad z_1(x) \\
\vdots \\
\max & \quad z_k(x) \\
\text{s.t.} & \quad x \in X
\end{align*}$$

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

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 $Z$ [28]. We do not make any particular distinction herein, adopting the ‘Pareto optimal’ designation for almost cases, 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 $j = 1, \ldots, k$ 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:

$z \in \mathbb{R}^k$ dominates $z' \in \mathbb{R}^k$ if and only if $z_j \geq z'_j$ for all $j = 1, \ldots, k$ 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 $j = 1, \ldots, k$.

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 [23, 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 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 [11] 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 $F_1(x, y) = -2x$

max $F_2(x, y) = -x + 5y$

s.t. max $f(y) = -y$

s.t. $x - 2y \leq 4$ (1)

$2x - y \leq 24$ (2)

$3x + 4y \leq 96$ (3)

$x + 7y \leq 126$ (4)

$-4x + 5y \leq 65$ (5)

$x + 4y \geq 8$ (6)

$x, y \geq 0$

The problem is depicted in Figure 1.
Figure 1. Graphical representation of example 1.

The induced region, \( IR \), is \([DE] \cup [EB] \cup [BA]\).

Graphically, we can also determine the whole Pareto optimal set (a subset of \( IR \)) for this bi-objective bilevel problem, which is \( \{D\} \cup [CB] \cup [BA]\). 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.

<table>
<thead>
<tr>
<th></th>
<th>( x )</th>
<th>( y )</th>
<th>( F_1 )</th>
<th>( F_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>17.45455</td>
<td>10.90909</td>
<td>-34.9091</td>
<td>37.09091</td>
</tr>
<tr>
<td>B</td>
<td>14.66667</td>
<td>5.333333</td>
<td>-29.3333</td>
<td>12</td>
</tr>
<tr>
<td>C</td>
<td>13.3333</td>
<td>4.666667</td>
<td>-26.6667</td>
<td>10</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>

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 \( F_2 \).
- All solutions from C to D (exclusive) of the induced region are non-efficient 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 MOLP problem

In this section we establish a relationship between the induced region, IR, and the Pareto optimal set of a multiobjective linear programming (MOLP) problem. This property holds for the single-objective bilevel linear programming problem and for the MOBLLP (1) as we have only considered multiple objectives at the upper level which do not affect the induced region.

Proposition 1. 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 problem with \( n_1+2 \) objective functions:

\[
\begin{align*}
\max & \quad f(y) = d^2 y \\
\max & \quad x_i, \quad i = 1, \ldots, n_1 \\
\max & \quad -\sum_{i=1}^{n_1} x_i \\
\text{s.t.} & \quad (x, y) \in S 
\end{align*}
\] (4)

Proof. Let \( E_1 \) denote the set of Pareto optimal solutions of (4) and recall that \( IR = \{(x, y) : (x, y) \in S, y \in P(x)\} \).

a) \((\bar{x}, \bar{y}) \in IR \Rightarrow (\bar{x}, \bar{y}) \in E_1\)
As \((\bar{x}, \bar{y})\in IR\), then \((\bar{x}, \bar{y})\in S\) and \(\bar{y}\in P(\bar{x})\), which means that \(\bar{y}\) maximizes \(f(y)\) over \(S(\bar{x})=\{x: B^2 y \leq b^2 - A^2 \bar{x}, y \geq 0\}\). Suppose that \((\bar{x}, \bar{y})\notin E_1\), i.e. there exists \((x, y)\in S\) such that the criterion vector of \((x, y)\) dominates the criterion vector of \((\bar{x}, \bar{y})\). This is equivalent to say that \(f(y) \geq f(\bar{y})\), \(x_i \geq \bar{x}_i\) for \(\forall i\) and \(-\sum_{i=1}^{n} x_i \geq -\sum_{i=1}^{n} \bar{x}_i\), with at least one strict inequality. Thus, the strict inequality must be the first one, that is \(f(y) > f(\bar{y})\). However, this contradicts the fact that \(\bar{y}\in P(\bar{x})\). Hence, \((\bar{x}, \bar{y})\in E_1\).

\[ b) (\bar{x}, \bar{y})\in E_1 \Rightarrow (\bar{x}, \bar{y})\in IR \]

Suppose that \((\bar{x}, \bar{y})\notin IR\). As \((\bar{x}, \bar{y})\in E_1\), 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, 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 of \((4)\) and equal in the others. This contradicts the hypothesis that \((\bar{x}, \bar{y})\in E_1\). Hence, \((\bar{x}, \bar{y})\in IR\).

It should be noticed that this proposition results from a different formulation of a related proposition stated in [14, 16]. While Proposition 1 makes an explicit representation of the surrogate MOLP problem \((4)\), the proposition in [14, 16] uses the concept of \(K\)-efficiency with respect to a pointed convex cone \(K\) which can be different from \(\mathcal{R}_+^p\) (where \(p\) denotes the number of objective functions in the surrogate multiobjective problem). Note that the concept of Pareto optimality or efficiency defined above for a multiobjective problem with \(k\) objectives is equivalent to the \(K\)-efficiency only if \(K=\mathcal{R}_+^k\).

Although Proposition 1 can provide an interesting theoretical result, its use in practice is at least doubtful due to the large number of objectives of \((4)\).

Suppose that we wish to exploit this result to solve a MOBLLP. Therefore, several Pareto optimal solutions to \((4)\) are computed, which are then evaluated by the upper level objective functions of the MOBLLP, and the nondominated points are selected. However, even if we generate all the Pareto optimal extreme points of \((4)\) (or a larger set of 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:

\[
\begin{align*}
\text{max } & \quad F_1(x, y) = 2x_1 - 4x_2 + y_1 - y_2 \\
\text{max } & \quad F_2(x, y) = -x + 2x_2 - y_1 + 5y_2 \\
\text{s.t. } & \quad \text{max } \quad f(y) = 3y_1 + y_2 \\
\text{s.t. } & \quad 4x_1 + 3x_2 + 2y_1 + y_2 \leq 60 \\
& \quad 2x_1 + x_2 + 3y_1 + 4y_2 \leq 60 \\
& \quad x_1, x_2, y_1, y_2 \geq 0
\end{align*}
\]

The formulation \((4)\) with respect to this MOBLLP is the following:
\[ \begin{align*} 
\text{max } f_1 &= 3y_1 + y_2 \\
\text{max } f_2 &= x_1 \\
\text{max } f_3 &= x_2 \\
\text{max } f_4 &= -x_1 - x_2 \\
\text{s.t. } & 4x_1 + 3x_2 + 2y_1 + y_2 \leq 60 \\
& 2x_1 + x_2 + 3y_1 + 4y_2 \leq 60 \\
& x_1, x_2, y_1, y_2 \geq 0 
\end{align*} \]

Using a *Vector Maximum Algorithm* [28] 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>
<thead>
<tr>
<th>Solution</th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( y_1 )</th>
<th>( y_2 )</th>
<th>( f(y) )</th>
<th>( F_1(x, y) )</th>
<th>( F_2(x, y) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>60</td>
<td>20</td>
<td>-20</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>-15</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-80</td>
<td>40</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>8.5714</td>
<td>17.1429</td>
<td>0</td>
<td>51.429</td>
<td>-17.143</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>7.5</td>
<td>0</td>
<td>15</td>
<td>0</td>
<td>45</td>
<td>30</td>
<td>-22.5</td>
</tr>
</tbody>
</table>

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) enabled 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 problem 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 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, 19]. In this section we follow these transformations to reduce MOBLLP to a multiobjective mixed-integer (0-1) linear programming 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, 11, 20]:

\[
\begin{align*}
\text{max } & \quad F_1(x, y) = c_1^T x + d_1^T y \\
\text{s.t. } & \quad A_1 x \leq b_1 \\
& \quad A_2 x + B_2 y \leq b_2 \\
& \quad \lambda (B_2^T - A_2^T x - A_1^T y) = 0 \\
& \quad y (\lambda B_2^T - d_2^T y) = 0 \\
& \quad \lambda B_2^T \geq d_2^T \\
& \quad y \geq 0, \ x \geq 0, \ \lambda \geq 0
\end{align*}
\]

where \( \lambda \in \mathbb{R}^{m_2} \).

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:

\[
\begin{align*}
\text{max } & \quad F_1(x, y) = c_1^T x + d_1^T y \\
\text{s.t. } & \quad A_1 x \leq b_1 \\
& \quad A_2 x + B_2 y \leq b_2 \\
& \quad \lambda (B_2^T - A_2^T x - A_1^T y) = 0 \\
& \quad y (\lambda B_2^T - d_2^T y) = 0 \\
& \quad \lambda B_2^T \geq d_2^T \\
& \quad y \geq 0, \ x \geq 0, \ \lambda \geq 0 \\
& \quad \lambda + \mu \leq \mu \leq \mu \\
& \quad -A_2 x - A_1 y - \mu \leq -b_2 \\
& \quad y + \mu \leq \mu \\
& \quad \lambda B_2^T - \mu \leq d_2^T \\
& \quad u \in \{0,1\}^{m_2} \quad v \in \{0,1\}^{m_2}
\end{align*}
\]

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

5. A methodology based on a MOMILP procedure

Methods for computing Pareto optimal solutions to a multiobjective programming problem 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 [13] and [18], 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 [30] can reach both supported and unsupported Pareto optimal solutions, thus being more adequate to deal with MOBLLP problems.

Reference point approaches [30] 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 [31].

Consider the general formulation (3) of a multiobjective optimization problem. Let \( q \in \mathbb{R}^k \) 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 \( \min_{x \in X} \left\{ \max_{i=1}^{k} (q_i - z_i(x)) \right\} \) 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 \( -\rho \sum_{i=1}^{k} z_i(x) \) is usually added to the scalarizing function to ensure the Pareto optimality condition (where \( \rho > 0 \) is a constant small enough).

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

\[
\begin{align*}
\min & \quad \alpha - \rho \sum_{i=1}^{k} z_i(x) \\
\text{s.t.} & \quad z_i(x) + \alpha \geq q_i, \quad i = 1, \ldots, k \\
& \quad x \in X \\
& \quad \alpha \text{ free}
\end{align*}
\]

(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, 29] 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 only 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 methods, sensitivity information on the variation of parameters can be very useful to avoid computing the same Pareto optimal solution more than once. This type 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 Climaco [2, 3] developed an interactive reference point procedure and software for MOMILP problems, 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 multiobjective mixed-integer programming problem (MOMILP). Although the procedure has been developed to be an interactive method, it can be further used to generate the whole Pareto frontier in bi-objective bilevel linear problems, thus providing a generating method.

We start by describing the interactive procedure in [2]. Firstly, the payoff table of the MOMILP problem 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. Special measures are taken to guarantee that the row criterion vectors are nondominated. Let $x^i$ be a Pareto optimal solution that maximizes $z_i(x)$ over $X$, $i=1,\ldots,k$. So, $z^i = z(x^i) = (z_1^i, z_2^i, \ldots, z_k^i)$, where $z_i^* = z_i^1$ is the maximum of $z_i(x)$, constitutes the $i$th row of the payoff table.

### Table 3. Payoff table.

<table>
<thead>
<tr>
<th>$z^1$</th>
<th>$z^2$</th>
<th>$\ldots$</th>
<th>$z^k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$z_1^*$</td>
<td>$z_2^*$</td>
<td>$\ldots$</td>
<td>$z_k^*$</td>
</tr>
<tr>
<td>$z_1^1$</td>
<td>$z_2^1$</td>
<td>$\ldots$</td>
<td>$z_k^1$</td>
</tr>
<tr>
<td>$z_1^2$</td>
<td>$z_2^2$</td>
<td>$\ldots$</td>
<td>$z_k^2$</td>
</tr>
<tr>
<td>$\ldots$</td>
<td>$\ldots$</td>
<td>$\ldots$</td>
<td>$\ldots$</td>
</tr>
<tr>
<td>$z_1^k$</td>
<td>$z_2^k$</td>
<td>$\ldots$</td>
<td>$z_k^*$</td>
</tr>
</tbody>
</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.

The main cycle of the algorithm is as follows:

- Define a new reference point $\bar{q}$.
- Compute a Pareto optimal solution by solving the mixed-integer program (7) with $q=\bar{q}$.
- 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 ($\bar{q}_j$) keeping the other components equal. The amount by which $\bar{q}_j$ 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 problem.

**Step 1.** Ask the DM to specify a reference point $\bar{q} \in \mathbb{R}^k$.

At the first interaction it is proposed by default the ideal point of the MOMILP problem (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 $q = \bar{q}$ 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 $\left(\bar{q}_1, \ldots, \bar{q}_j + \theta_j, \ldots, \bar{q}_k\right)$ 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 $\bar{\theta}_j > 0$ such that the structure of the previous branch-and-bound tree remains unchanged for variations in $\bar{q}_j$ up to $\bar{q}_j + \bar{\theta}_j$.

This means that reference points $\left(\bar{q}_1, \ldots, \bar{q}_j + \theta_j, \ldots, \bar{q}_k\right)$ with $\theta_j \leq \bar{\theta}_j$ lead to the same Pareto optimal solution or they lead to different Pareto optimal solutions that are easily computed. The branch-and-bound tree is then updated (ii) for $\theta_j$ slightly over $\bar{\theta}_j$.

Since the $\bar{\theta}_j$ returned by the sensitivity analysis can be only a lower bound for the true maximum value of the parameter, the same Pareto optimal solution can be obtained in (ii), a situation that occurs more often in all-integer programs than in mixed-integer models. In this case the process automatically returns to (i).
Let us now briefly explain how $\bar{\theta}_j$ is determined in phase $(i)$. Let $SP(\bar{q}, \theta_j)$ denote the scalarizing program (7) with $q=\bar{q}$ and parameterized on $q_j$:

$$f(\theta_j) = \min \left\{ \alpha - \rho \sum_{i=1}^{k} z_i(x) \right\}$$

s.t. $z_i(x) + \alpha \geq \bar{q}_i \quad i = 1, \ldots, k, i \neq j$

$$z_j(x) + \alpha \geq \bar{q}_j + \theta_j$$

$x \in X$

where $X = \{x: Ax = b, \ x \geq 0, \ x_l \text{ integer for } l \in I \subseteq \{1, \ldots, n\} \}$ and $z_i(x) = c^i x$ as this concerns a multiobjective mixed-integer linear programming problem. $I$ denotes the set of indices of the integer variables. It is worthwhile to note that the outcome $x$ of this program does not change if a constant is added to all components of $\bar{q}$. Therefore, if necessary the vector $\bar{q}$ is shifted in order that $\bar{q} > z^*$, which enables to consider $\alpha \geq 0$.

The first Pareto optimal solution of the directional search, say $\bar{x}$, has been obtained by solving $SP(\bar{q}, 0)$ using a branch-and-bound algorithm, for a given $\bar{q}$. Regarding the parametric mixed-integer program $SP(\bar{q}, \theta_j)$, the linear sub-problem associated with a node $p$ of the branch-and-bound tree can be generically formulated as follows. This program is denoted by $LP^p(\bar{q}, \theta_j)$:

$$f^p(\theta_j) = \min \left\{ \alpha - \rho \sum_{i=1}^{k} z_i(x) \right\}$$

s.t. $z_i(x) + \alpha \geq \bar{q}_i \quad i = 1, \ldots, k, i \neq j$

$$z_j(x) + \alpha \geq \bar{q}_j + \theta_j$$

$x \in X^p$

where $X^p = \{x: Ax = b, \ x \geq 0, \ L_l^p \leq x_l \leq U_l^p, \ l \in I\}$. Some $L_l^p$ may be zero and some $U_l^p$ infinite.

The following properties hold in any $LP^p(\bar{q}, \theta_j)$ when $\theta_j$ is increased:

Property 1 – If $LP^p(\bar{q}, 0)$ is infeasible then $LP^p(\bar{q}, \theta_j)$ is also infeasible for any $\theta_j > 0$.

Property 2 – The function $f^p(\theta_j)$ is a nondecreasing piecewise linear convex function of the parameter $\theta_j$ and its slope in the last interval of $\theta_j$ is equal to $1$.

Let $\pi^p_i, \ i=1, \ldots, k,$ be the dual variables associated with the first $k$ constraints of $LP^p(\bar{q}, \theta_j)$. In the last interval of $\theta_j$, $\pi^p_j = 1$ which means that the solution that maximizes $z_j(x)$ over $X^p$ has been reached (see [2]).

The purpose of the sensitivity analysis is to provide a range $[0, \bar{\theta}_j]$ for the parameter $\theta_j$ such that the optimal solutions of $SP(\bar{q}, \theta_j)$ with $\theta_j \in [0, \bar{\theta}_j]$ are given by the node of the branch-and-bound tree that solved $SP(\bar{q}, 0)$. Let us denote this particular node by node * and the corresponding linear program by $LP^*$.

If the surplus variable of the $j^{th}$ constraint is basic in the optimal solution of $LP^*(\bar{q}, 0)$, then $\bar{\theta}_j$ is given by the value of this surplus variable. Moreover, there is no need to
explore variations of $\theta_j$ under this value because they lead to the same Pareto optimal solution. If the surplus variable is nonbasic, then the function $f^*(\theta_j)$ is compared with lower bound estimates for the piecewise functions $f^p(\theta_j)$ of some other tree leaves of the branch-and-bound tree, thus defining “intersection” parameter values $\theta_j^p$ with those nodes (for details, see [2]). $\theta_{j}$ is then defined as the minimum value among all $\theta_j^* p$ and $\theta_{j}^{\text{max}}$, where $\theta_{j}^{\text{max}}$ is the maximum value of $\theta_j$ that preserves the current optimal basis of $LP^*$ and gives a feasible solution to $SP(\overline{q}, \theta_j)$.

The nodes to be compared with node * are the terminal nodes (tree leaves) that are neither infeasible nor inactive. A node $p$ (different from *) is classified as inactive if the current value of the dual variable $\pi_j^p$ is 1. Note that this node will be fathomed for any $SP(\overline{q}, \theta_j)$ with $\theta_j > 0$. An inactive node will be re-activated if the DM changes the direction of search and chooses another objective to be improved.

As a consequence of property 2 stated above, if $\pi_j^* = 1$ and all the other terminal nodes are either infeasible or inactive, then the current Pareto optimal solution maximizes the objective function $z_j$ of the MOMILP problem.

This sensitivity analysis phase and the next optimization stage require that some information about the branch-and-bound tree must be preserved. This information includes the structure of the tree (node indices, links and branching constraints) and the following data for each terminal node $p$ that has a feasible $LP^p$: a codification of the basis (a sum of powers of 2 whose exponents are the indices of the basic variables), $f^p$ (the current optimal value of $LP^p$), $\pi_i^p$ and $\theta_i^{\text{max}}$, $i = 1, \ldots, k$, and the current status of the node (indicating whether the current solution of $LP^p$ is feasible to the mixed-integer program or if the node is inactive).

For $0 \leq \theta_{j} \leq \overline{\theta}_j$ the structure of the branch-and-bound tree does not change and $SP(\overline{q}, \theta_j)$ for each $\theta_j$ is optimized in the node * of the tree. These optimal solutions to the scalarizing program may correspond to the same Pareto optimal solution or to different continuous Pareto optimal solutions that can be obtained straightforwardly by applying classic linear programming sensitivity analysis to $LP^*$. The parameter is then changed to $\tilde{\theta}_j = \overline{\theta}_j + \epsilon$ with $\epsilon$ small and positive in order to continue searching for Pareto optimal solutions throughout the same direction. Hence, the next reference point is $\left(\overline{q}_1, \ldots, \tilde{\theta}_j, \ldots, \overline{q}_k\right)$.

Then, the branch-and-bound tree is updated (phase ii) to find the optimal solution of $SP(\overline{q}, \tilde{\theta}_j)$. Although the updating process varies according to different situations that may result from the sensitivity analysis phase, three main steps summarize this process:

1. simplify the tree by cutting off parts that are linked by branching constraints no longer active;
2. update the information on the remaining terminal nodes;
3. proceed with the branch-and-bound method as usual, expanding the tree by further branching if necessary, until the optimum of the new scalarizing program \( SP(\bar{q}, \bar{\theta}_j) \) is reached.

A full description of the updating process 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, some features may be highlighted.

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

- The procedure recognizes when it reaches a Pareto optimal solution maximizing one of the objectives of the multiobjective problem. Therefore, if a direction 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 \( \mu \). 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 \( z_1(x) \).

Let \( z^1=(z_1^*, z_2^1) \) be the first row vector of the payoff table of the bi-objective problem, i.e., the nondominated criterion vector corresponding to a Pareto optimal solution \( x^1 \) that maximizes \( z_1(x) \) in the feasible region \( X \). The solution \( x^1 \) also optimizes the achievement scalarizing program (7) for the reference point \( q = z^1 \) provided that the constant \( \rho \) in (7) is set small enough, i.e. it satisfies \( \rho < \rho' \) for a certain \( \rho' \). In fact the following result holds:

**Proposition 2.** Consider the bi-objective programming problem

\[
\begin{align*}
\max & \quad z_1(x) \\
\max & \quad z_2(x) \\
\text{s.t.} & \quad x \in X
\end{align*}
\]

Let \( Z \) denote the feasible region in the criterion space 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 \( z^1 \in Z_{nd} \) its criterion vector. Then, \( x^1 \) or any other Pareto optimal solution whose criterion vector is \( z^1 \) are the unique optimal solutions of the scalarizing program

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

with \( q = z^1 \) and
\[
0 < \rho < \min_{\bar{z} \in Z_{nd} \setminus \{z^1\}} \left\{ \frac{z^1_1 - \bar{z}_1}{\sum_{i=1}^{2} (\bar{z}_i - z^1_i)} : \sum_{i=1}^{2} (\bar{z}_i - z^1_i) > 0 \right\}
\]

**Proof.** Consider a feasible solution \( \bar{x} \in X \) whose criterion vector is \( \bar{z} \in Z \) such that \( \bar{z} = (z^1_1(x), z^1_2(x)) \neq z^1 \).

Since \( z^1_1 \) is the maximum of \( z_1(x) \) and \( z^1 \) is a nondominated vector, then

\[
\max_{i=1,2} \left\{ \mathcal{U}_i - z_1(x) \right\} = \max_{i=1,2} \left\{ \mathcal{E}_i - z_1, z^1_2 - z^1_2 \right\}
\]

is strictly positive for all \( \bar{z} \in Z \setminus \{ z^1 \} \).

On the other hand, \( \alpha^1 = \max_{i=1,2} \left\{ \mathcal{U}_i - z_1(x) \right\} = 0 \) for \( q = z^1 \).

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

\[
-\rho \sum_{i=1}^{2} z^1_i < \bar{\alpha} - \rho \sum_{i=1}^{2} \bar{z}_i , \forall \bar{z} \in Z \setminus \{ z^1 \}
\]

This means that we must have \( \rho < \frac{-\alpha^1}{\sum_{i=1}^{2} (\bar{z}_i - z^1_i)} \) whenever \( \sum_{i=1}^{2} (\bar{z}_i - z^1_i) > 0 \),

for all \( \bar{z} \in Z \setminus \{ z^1 \} \).

If \( \bar{z} \) is nondominated, then \( z^1_1 > \bar{z}_1, z^1_2 < \bar{z}_2 \) and \( \alpha^1 = z^1_2 - z^1_1 \).

Hence, it suffices for \( \rho \) to be defined as in this proposition for \( z^1 \) uniquely optimize the scalarizing program with \( q = z^1 \).

Consequently, 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 \( z_i(x) \). Let \( z^1 = (z^1_1, z^1_2) \) 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 \( z_2 \) is reached.

In this algorithm, \( \mu \) represents the maximum value that is allowed for the ratio \( (z^\text{new}_2 - z^\text{prev}_2) / (\bar{z}^* - z^1_2) \), where \( z^\text{new} \) and \( z^\text{prev} \) are the criterion vectors of two continuous Pareto optimal solutions, the new and the previous one, respectively; \( \bar{z}^*_2 \) 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, \( \mu \) is used for restricting differences in \( z_1 \).

6. An example of the application of the MOMILP procedure to a bi-objective bilevel linear problem

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.

\[
\begin{align*}
\text{max } F_1(x, y) &= -2x \\
\text{max } F_2(x, y) &= -x + 5y \\
\text{s.t. } & \quad x - 2y \leq 4 \\
& \quad 2x - y \leq 24 \\
& \quad 3x + 4y \leq 96 \\
& \quad x + 7y \leq 126 \\
& \quad -4x + 5y \leq 65 \\
& \quad x + 4y \geq 8 \\
& \quad 2\lambda_1 + \lambda_2 - 4\lambda_3 - 7\lambda_4 - 5\lambda_5 + 4\lambda_6 \leq 1 \\
& \quad (x - 2y - 4)\lambda_1 = 0 \\
& \quad (2x - y - 24)\lambda_2 = 0 \\
& \quad (3x + 4y - 96)\lambda_3 = 0 \\
& \quad (x + 7y - 126)\lambda_4 = 0 \\
& \quad (-4x + 5y - 65)\lambda_5 = 0 \\
& \quad (-x - 4y + 8)\lambda_6 = 0 \\
& \quad (2\lambda_1 + \lambda_2 - 4\lambda_3 - 7\lambda_4 - 5\lambda_5 + 4\lambda_6 - 1).y = 0 \\
& \quad x, y \geq 0 \\
& \quad \lambda_i \geq 0, \quad i = 1, \ldots, 6
\end{align*}
\]

Next, the problem is reformulated as the following MOMILP problem.

\[
\begin{align*}
\text{max } F_1(x, y) &= -2x \\
\text{max } F_2(x, y) &= -x + 5y \\
\text{s.t. } & \quad x - 2y \leq 4 \\
& \quad 2x - y \leq 24 \\
& \quad 3x + 4y \leq 96 \\
& \quad x + 7y \leq 126 \\
& \quad -4x + 5y \leq 65 \\
& \quad x + 4y \geq 8 \\
& \quad 2\lambda_1 + \lambda_2 - 4\lambda_3 - 7\lambda_4 - 5\lambda_5 + 4\lambda_6 \leq 1 \\
& \quad x - 2y + Mu_{11} \geq 4 \\
& \quad \lambda_1 + Mu_{12} \leq M \\
& \quad 2x - y + Mu_{21} \geq 24 \\
& \quad \lambda_2 + Mu_{22} \leq M \\
& \quad 3x + 4y + Mu_{31} \geq 96 \\
& \quad \lambda_3 + Mu_{32} \leq M \\
& \quad x + 7y + Mu_{41} \geq 126 \\
& \quad \lambda_4 + Mu_{42} \leq M \\
& \quad -4x + 5y + Mu_{51} \geq 65 \\
& \quad x, y \geq 0 \\
& \quad \lambda_i \geq 0, \quad i = 1, \ldots, 6
\end{align*}
\]
\[
\begin{align*}
\lambda_5 + Mu_5 & \leq M \\
-x - 4y + Mu_6 & \geq -8 \\
\lambda_6 + Mu_6 & \leq M \\
2\lambda_1 + \lambda_2 - 4\lambda_3 - 7\lambda_4 - 5\lambda_5 + 4\lambda_6 + Mv_1 & \geq 1 \\
y + Mv_1 & \leq M \\
u_i & \in \{0,1\}, \ i = 1, \ldots, 6 \\
v_1 & \in \{0,1\}
\end{align*}
\]

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 the example.

<table>
<thead>
<tr>
<th></th>
<th>F_1</th>
<th>F_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-34.9091</td>
<td>37.09091</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>

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 stepwise \( \mu = 0.5\% \) 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 \( \mu \) 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 \( q=(0, 72.1) \). Figure 3 shows the Pareto frontier of the problem, i.e. 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 (which has been produced by a graphical analysis).
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 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


