A Differential Evolution Algorithm to Semivectorial Bilevel Problems

Maria João Alves[1,3], Carlos Henggeler Antunes[2,3]

[1] CEBER - Faculty of Economics, University of Coimbra, Portugal
[2] DEEC, University of Coimbra, Polo 2, Coimbra, Portugal
[3] INESC Coimbra, Coimbra, Portugal

mjalves@fe.uc.pt; ch@deec.uc.pt

Abstract. Semivectorial bilevel problems (SVBLP) deal with the optimization of a single function at the upper level and multiple objective functions at the lower level of hierarchical decisions. Therefore, a set of nondominated solutions to the lower level decision maker (the follower) exists and should be exploited for each setting of decision variables controlled by the upper level decision maker (the leader). This paper presents a new algorithmic approach based on differential evolution to compute a set of four *extreme* solutions to the SVBLP. These solutions capture not just the optimistic vs. pessimistic leader's attitude but also possible follower's reactions more or less favorable to the leader within the lower level nondominated solution set. The differential evolution approach is compared with a particle swarm optimization algorithm. In this experimental comparison we draw attention to pitfalls associated with the interpretation of results and assessment of the performance of algorithms in SVBLP.

Keywords: semivectorial bilevel problems; differential evolution; particle swarm optimization; optimistic/pessimistic frontiers; optimistic/deceiving solutions; pessimistic/rewarding solutions

1 Introduction

A semivectorial bilevel problem (SVBLP) is an optimization problem with a single objective function at the upper (leader's) level and multiple objective functions at the lower (follower's) level of hierarchical non-cooperative decisions. Hence, a multiobjective (MO) optimization problem contributes to define the feasible region to the leader's problem, in the sense that a lower level nondominated region exists for each setting of upper level variables. Thus, when solving his/her optimization problem, the leader must anticipate the follower's choice of a nondominated solution embodying a trade-off between the lower level multiple objectives. The follower's reaction may strongly affect the leader's optimal solution, depending on the follower's preference structure vis-à-vis the nondominated region established by the instantiation of the leader's decision variables. Therefore, it is useful for the leader to have an overview of possible optimal solutions resulting from different attitudes (optimistic or pessimistic) in face of his/her expectation of the more or less favorable follower's choice. In addition to the intrinsic theoretical and computational difficulty in computing solutions to the SVBLP, the leader does not have a-priori information about the nondominated solution the follower will choose according to his/her (unknown) preferences.

In this setting, this paper presents an algorithmic approach intertwining single and MO versions of Differential Evolution (DE) for the upper level and lower level problems, which is aimed at computing a set of extreme solutions to the SVBLP. These extreme solutions are: the *optimistic* solution offering the leader the best objective function value when the follower's decision for each setting of upper level variables is the best for the leader; the *deceiving* solution when the leader adopts an optimistic approach but the follower's reaction is the worst for the leader; the *pessimistic* solution offering the best objective function value for the leader when the follower's decision for each setting of upper level variables is the worst for the leader; and the *rewarding* solution when the leader adopts a pessimistic approach but the follower's reaction is the most favorable to the leader.

The algorithmic approach introduces new concepts of optimistic and pessimistic frontiers and adapts DE mechanisms to combine the search at both levels with the population split between orientations to each frontier. This approach is compared with a Particle Swarm Optimization (PSO) algorithm we have previously developed [1], which has been extended herein to compute these four extreme solutions. The algorithms are tested on a set of benchmark problems for multiobjective bilevel (MOBL) optimization (considering only one of the objective functions in the upper level). We were able to determine analytically the exact solutions to these problems, which enable to assess the quality of the solutions obtained by the algorithms. A thorough analysis of the computational results allowed us to unveil pitfalls associated with the interpretation of results and assessment of the algorithm performance in SVBL and MOBL optimization. This paper also aims at drawing attention to these pitfalls.

In section 2, the SVBLP is presented and the definitions of the extreme (*optimistic*, *deceiving*, *pessimistic* and *rewarding*) solutions are introduced. Algorithmic approaches to deal with the SVBLP are also briefly reviewed in this section. The concepts of optimistic and pessimistic frontiers are presented and illustrated in section 3. In section 4, the Semivectorial Bilevel Differential Evolution (SVBLDE) algorithm is proposed. Computational results are presented and discussed in section 5. Concluding remarks are presented in section 6.

2 The SVBLP: Optimistic vs. Pessimistic Approaches

The SVBLP is a bilevel optimization problem with a single objective function at the upper level $F(x, y)$ and multiple objective functions $f_k(x, y), k = 1, \ldots, m$ at the lower level.

$$
\begin{aligned}
&\text{'}\min_{x \in X}\text{'} \quad F(x, y) \\
&s.t. \quad G(x, y) \le 0 \qquad\qquad (1) \\
&\qquad y \in \arg\min_{y \in Y} \{ (f_1(x, y), \ldots, f_m(x, y)) : g(x, y) \le 0 \}
\end{aligned}
$$

with $X \subseteq \Re^{n_1}$ and $Y \subseteq \Re^{n_2}$, which impose bounds (box constraints) on the upper level variables x (which are controlled by the leader) and on the lower level variables y (which are controlled by the follower), respectively. $G(x,y) \leq 0$ and $g(x,y) \leq 0$ are general constraints, respectively in the upper and the lower level problems.

Since the decision process is sequential and the leader decides first, x assumes a constant vector in the optimization of $f_k(x,y), k=1,...,m$. For each $x \in X$ there is a set of efficient (Pareto optimal or nondominated) solutions to the lower level problem represented by $\Psi_{Ef}(x)$. Let $Y(x)=\{y \in Y : g(x,y) \leq 0\}$.

Thus, $\Psi_{Ef}(x)=\{y \in Y : (\text{there is no } y' \in Y(x) \mid f(x,y') \prec f(x,y))\}$ where $\prec$ denotes the dominance relation, i.e., $f(x,y') \prec f(x,y)$ iff $f_j(x,y') \leq f_j(x,y)$ for all j=1,…,m, and $f_j(x,y') < f_j(x,y)$ for at least one j.

Since there is not, in general, a single efficient solution to the lower level problem for each x, problem (1) is ambiguous. This is the reason for the quotation marks in the upper level objective function. Two main approaches have been suggested in the literature to address the problem – the optimistic and the pessimistic approaches – leading to two reformulations of (1). As in the single objective bilevel problem with non-unique optimal solutions to the lower level problem, the optimistic formulation of the SVBLP is much simpler to tackle and has therefore been the most investigated.

The optimistic approach assumes that the leader is able to influence the choice of the follower. Thus, the upper level optimization can be taken with respect to x and y to determine the optimal optimistic solution. This means that, for a given upper level decision x, the lower level decision y is the one that presents the minimum $F(x,y)$ among the efficient solutions to the lower level problem for that x, which also satisfy upper level constraints (if there are upper level constraints coupled with lower level variables, i.e. $G(x,y) \leq 0$). The optimal optimistic solution will be called just *optimistic* solution and is defined as follows:

- The *optimistic* solution, (x^o,y^o), is given by

$$\min_{x \in X, y \in Y} \{F(x,y) : y \in \Psi_{Ef}(x), G(x,y) \leq 0\}$$

In the pessimistic approach the leader prepares for the worst case. The leader chooses the x that leads to a feasible solution with minimum F in view of the follower's decisions y worst for the leader. The optimal pessimistic solution will be called just *pessimistic* solution and is defined as follows:

- the *pessimistic* solution, (x^p,y^p), is given by

$$\min_{x \in X} \left\{ \max_{y \in Y} \{F(x,y) : y \in \Psi_{Ef}(x)\} : G(x,y) \leq 0 \right\}$$

A failed optimistic approach leads to the deceiving solution. This means that the leader chooses x according to the optimistic approach but the follower does not react accordingly and takes the decision with worst value for the leader's objective function. Thus, given the optimistic upper level decision x^o,

- the *deceiving* solution is $(x^d, y^d) = (x^o, y^d)$ where y^d is given by
  $\max_{y \in Y} \left\{ F(x^o, y) : y \in \Psi_{Ef}(x^o) \right\}$.

According to the above definition, the *deceiving* solution may be infeasible to the leader, i.e. infeasible for the SVBLP. Knowing whether the deceiving follower's reaction is feasible or infeasible to the upper level problem is also a useful information to the leader.

A successful pessimistic approach leads to the *rewarding* solution. Thus, given the pessimistic upper level decision x^p, the *rewarding* solution can be defined as the feasible $(x^r, y^r) = (x^p, y^r)$ such that y^r is given by

$$\min_{y \in Y} \left\{ F(x^p, y) : y \in \Psi_{Ef}(x^p), G(x^p, y) \le 0 \right\}$$

Bonnel [2] and Bonnel and Morgan [3] firstly addressed the SVBLP by providing necessary optimality conditions [2] and a penalty function method [3] for determining the *optimistic* solution. Other methods based on penalty functions to compute the *optimistic* solution were developed by Ankhili and Mansouri [4], Zheng and Wan [5] and Ren and Wang [6] for the SVBLP with a MO linear problem in the lower level. Calvete and Galé [7] focused on the same problem and proposed an exact method and a genetic algorithm, considering the optimistic approach. Liu et al. [8] developed necessary optimality conditions for the *pessimistic* solution and Lv and Chen [9] proposed a discretization iterative algorithm to compute the *pessimistic* solution to a SVBLP without upper level variables in the lower level constraints. Alves et al. [1] firstly introduced the concept of *deceiving* solution and proposed an algorithm based on PSO to approximate the *optimistic*, *pessimistic* and *deceiving* solutions to the SVBLP. The *rewarding* solution was introduced in [10], where illustrative examples of these four types of extreme solutions were presented. In the present paper we propose a new algorithm based on DE to compute these four extreme solutions and extend the algorithm in [1] to compute also the *rewarding* solution.

3 Optimistic and Pessimistic Frontiers

Let us now define two new concepts to be used in the algorithm proposed in the next section, which are the *Optimistic* and the *Pessimistic frontiers*.

The *Optimistic frontier* (O) consists of the feasible solutions (x, y'), such that y' is the follower's efficient solution $y' \in \Psi_{Ef}(x), G(x, y') \le 0$, that provides the minimum (best) F for that $x \in X$:

$$O = \left\{ (x, y') : x \in X, y' \in \arg\min_{y \in Y} \left\{ F(x, y) : y \in \Psi_{Ef}(x), G(x, y) \le 0 \right\} \right\}$$

The *optimistic* solution (x^o, y^o) to the SVBLP is the solution $(x, y') \in O$ with minimum F.

The *Pessimistic frontier* (P) consists of the solutions (x, y'') such that y'' is the follower's efficient solution $y'' \in \Psi_{Ef}(x)$ that provides the maximum (worst) F for that $x \in X$:

$$P = \left\{ (x, y''): x \in X, y'' \in \arg\max_{y \in Y} \left\{ F(x, y): y \in \Psi_{Ef}(x) \right\} \right\}$$

The *pessimistic* solution (x^p, y^p) to the SVBLP is the feasible solution $(x, y'') \in P$, $G(x, y'') \leq 0$, with minimum F.

The *deceiving* solution (x^d, y^d) is the solution in P with $x^d = x^o$.

The *rewarding* solution (x^r, y^r) is the solution in O with $x^r = x^p$.

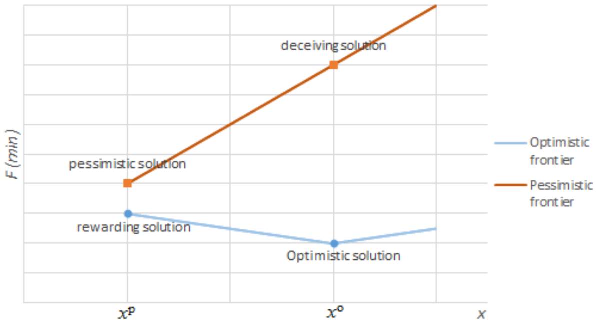

Fig. 1. *F* values in the Optimistic and Pessimistic efficient frontiers of a SVBL linear problem with one upper level variable (*x*) and two objective functions at the lower level.

In the example in Fig. 1, there is a significant difference between the *optimistic* and the *deceiving* solutions for the leader's objective function. Therefore, if the leader opts for an optimistic approach he/she takes a high risk, since the *deceiving* solution is very bad. Conversely, there is a small difference between the *pessimistic F* and the *rewarding* one, being the *F* value in the *rewarding* solution close to the *optimistic F*.

Since the *deceiving* solution is obtained from the *Pessimistic frontier* using an optimistic approach and the *rewarding* solution is obtained from the *Optimistic frontier* using a pessimistic approach, both frontiers should be simultaneously explored by an algorithm aimed at computing these four extreme solutions.

4 A Differential Evolution Algorithm for the SVBLP

In the SVBLDE algorithm proposed below, the population *Pop* of individuals is divided into two sub-populations Pop' and Pop'' which share the upper level *x* vectors. Let *Nu* be the number of upper level individuals. $Pop = Pop' \cup Pop''$ where $Pop' = \{(x_1, y'_1), (x_2, y'_2), ..., (x_{Nu}, y'_{Nu})\}$ and $Pop'' = \{(x_1, y''_1), (x_2, y''_2), ..., (x_{Nu}, y''_{Nu})\}$. The individuals of Pop' aim at approximating the *Optimistic frontier* while the individuals of Pop'' aim at approximating the *Pessimistic frontier*. DE operations are employed to evolve the population of the upper level problem and, for each upper level vector *x*, a lower level DE algorithm (DE_LOWERLEVEL_O_P) is used to determine (x,y') and (x,y''). Below, $\mathcal{F}$ denotes the mutation scaling factor and CR the crossover rate in the DE operations. Let

Tu be the number of upper level generations. The DE upper level search is described in Algorithm 1. We have used $\mathcal{F}$ =0.7 and CR=0.9.

Algorithm 1: Upper Level Search

---

1. $t \leftarrow 1$
2. Create a random initial population of *Nu* upper level vectors: $x_{i,t} \in X,\ i = 1,...,Nu$
3. **For** i =1 **to** *Nu* **do**
4. DE_LOWERLEVEL_O_P($x_{i,t}$, $y'_{i,t}$, $y''_{i,t}$)
5. Insert $(x_{i,t}, y'_{i,t})$ into Pop'_t and $(x_{i,t}, y''_{i,t})$ into Pop''_t
6. **End For** i

   Initialize the incumbent solutions:
7. $(x^o,y^o) \leftarrow$ arg min $\{F(x,y)$: $(x,y) \in Pop'_t$, $G(x,y) \le 0\}$ *//optimistic solution*

   Let $i1$ be the index of (x^o,y^o) in Pop'_t such that $(x^o,y^o) = (x_{i1,t}, y'_{i1,t})$
8. $(x^p,y^p) \leftarrow$ arg min $\{F(x,y)$: $(x,y) \in Pop''_t$, $G(x,y) \le 0\}$ *//pessimistic solution*

   Let $i2$ be the index of (x^p,y^p) in Pop''_t such that $(x^p,y^p) = (x_{i2,t}, y''_{i2,t})$
9. $(x^d,y^d) \leftarrow (x_{i1,t}, y''_{i1,t}) \in Pop''_t$ *//deceiving solution*
10. $(x^r,y^r) \leftarrow (x_{i2,t}, y'_{i2,t}) \in Pop'_t$ *//rewarding solution*
11. **For** t =1 **to** *Tu* **do**
12. **For** i =1 **to** *Nu* **do**
13. ♦ Select r_1 , r_2 and r_3 *// selection dependent on the DE variant*
14. j_{rand} = randint(1,n_1) *//n_1 is the number of UL variables*
15. **For** j =1 **to** n_1 **do**
16. ♦ $u_{i,j,t+1} = \begin{cases} x_{r3,j,t} + \mathcal{F}(x_{r1,j,t} - x_{r2,j,t}) & \text{if rand(0,1)} < \text{CR or } j = j_{rand} \\ x_{i,j,t} & \text{otherwise} \end{cases}$
17. **End For** j
18. DE_LOWERLEVEL_O_P ($u_{i,t+1}$, $w'_{i,t+1}$, $w''_{i,t+1}$)

    Update the incumbent solutions:
19. **If** $F(u_{i,t+1}, w'_{i,t+1}) < F(x^o,y^o)$ **and** $G(u_{i,t+1}, w'_{i,t+1}) \le 0$ **then**
20. $(x^o,y^o) \leftarrow (u_{i,t+1}, w'_{i,t+1})$ *//update the optimistic*
21. $(x^d,y^d) \leftarrow (u_{i,t+1}, w''_{i,t+1})$ *// and deceiving solutions*
22. **If** $F(u_{i,t+1}, w''_{i,t+1}) < F(x^p,y^p)$ **and** $G(u_{i,t+1}, w''_{i,t+1}) \le 0$ **then**
23. $(x^p,y^p) \leftarrow (u_{i,t+1}, w''_{i,t+1})$ *//update the pessimistic*
24. $(x^r,y^r) \leftarrow (u_{i,t+1}, w'_{i,t+1})$ *// and rewarding solutions*
25. ♦ **If** ACCEPT $u_{i,t+1}$ **then** *// criterion dependent on the DE variant*
26. $x_{i,t+1} = u_{i,t+1}$
27. Insert $(u_{i,t+1}, w'_{i,t+1})$ into Pop'_{t+1} and $(u_{i,t+1}, w''_{i,t+1})$ into Pop''_{t+1}
28. **Else**

29. $x_{i,t+1} = x_{i,t}$

30. Insert ($x_{i,t}$, $y'_{i,t}$) into Pop'_{t+1} and $(x_{i,t}, y''_{i,t})$ into Pop''_{t+1}

31. **End For** *i*

32. **End For** *t*

Output: (x^o,y^o), (x^d,y^d), (x^p,y^p), (x^r,y^r)

In Step 16, if $u_{i,j,t+1}$ does not satisfy the bounds defined by *X*, then it is projected into the closest bound.

We consider two DE variants: *DE/rand/1/bin* (the original version, which obtained good results in the comparative study of DE variants for global optimization in [11]) and *DE/best/1/bin* (the variant with highest performance in the same study). The steps marked with ♦ change from one variant to the other. In Step 13, *DE/rand/1/bin* randomly selects indexes $r_1 \neq r_2 \neq r_3$ from $\{1,\ldots,Nu\}$, while *DE/best/1/bin* randomly selects indexes $r_1 \neq r_2$ for x_{r1} and x_{r2} but an x_{best} is used in Step 16 to replace x_{r3}. The *DE/best/1/bin* variant divides the population into two equal parts: the first half is mainly oriented towards the *optimistic* solution, so $x_{best} = x^o$, and the second half of the population is mainly oriented towards the *pessimistic* solution, so $x_{best} = x^p$. In addition, r_1 and r_2 are randomly selected from $\{1,\ldots,Nu/2\}$ for $i \leq Nu/2$ and from $\{Nu/2+1,\ldots,Nu\}$ otherwise.

The criterion to decide whether $u_{i,t+1}$ is accepted or not in Step 25 (ACCEPT) also depends on the DE variant. Steps 25-30 define the population for the next generation. In *DE/rand/1/bin*, if ***(a)*** the new individual obtained for approximating the *Optimistic* frontier $(u_{i,t+1}, w'_{i,t+1})$ improves the current one in Pop'_t, i.e. $F(u_{i,t+1}, w'_{i,t+1}) < F(x_{i,t}, y'_{i,t})$, **or** ***(b)*** the new individual obtained for approximating the *Pessimistic* frontier $(u_{i,t+1}, w''_{i,t+1})$ improves the current one in Pop''_t, i.e. $F(u_{i,t+1}, w''_{i,t+1}) < F(x_{i,t}, y''_{i,t})$, then the new upper level individual $u_{i,t+1}$ is accepted and Steps 26-27 are performed. Otherwise, the previous individual is kept and Steps 29-30 are performed. In *DE/best/1/bin*, the acceptance criterion in the first half of the population (oriented to the *optimistic* solution) only considers condition ***(a)*** to decide whether $u_{i,t+1}$ is accepted or not, whereas in the second half of the population only condition ***(b)*** is considered.

The DE_LOWERLEVEL_O_P algorithm aims at computing two extreme efficient solutions to the lower level problem for a given *x*, one belonging to the *Optimistic* frontier and the other belonging to the *Pessimistic* frontier: $(x, y') \in O$ and $(x, y'') \in P$.

Let *Tl* be the number of lower level generations and *Nl* (an even number) the size of the lower level population. The algorithm attempts to converge to a population *Popy* of efficient solutions to the lower level problem polarized to the extreme values of the upper level objective function *F* (the maximum and the minimum). The first *Nl*/2 individuals of *Popy* are oriented to converge to y' while the remaining *Nl*/2 individuals are oriented to converge to y''.

Algorithm 2: DE_LOWERLEVEL_O_P $(x\downarrow,\ y'\uparrow,\ y''\uparrow)$

1. $t \leftarrow 1$
2. Create a random initial population of Nl lower level vectors $Popy_t = \{y_{i,t} \in Y, i = 1,..., Nl\}$ and sort $Popy_t$ by increasing order of F.
3. Define Eff with the solutions in $Popy_t$ that satisfy $g(x,y_{it}) \le 0$ and are not dominated by any other solution regarding the lower level objective functions $f_1,\dots f_m$.
4. Initialize y' and y'':
   $y' \leftarrow \arg\min_y \{F(x,y) : y \in Eff, G(x,y) \le 0\}$ //y': *solution on the Optimistic frontier*
   $y'' \leftarrow \arg\max_y \{F(x,y) : y \in Eff\}$ // y'': *solution on the Pessimistic frontier*
5. **For** t =1 **to** Tl **do**
6. **For** i =1 **to** Nl **do**
7. **If** $i \le Nl/2$ **then**
8. ♦ Randomly select $r_1 \ne r_2 \in \{1,\dots,Nl/2\}$ and select r_3 *//depends on the DE variant*
9. **Else**
10. ♦ Randomly select $r_1 \ne r_2 \in \{Nl/2+1,\dots,Nl\}$ and select r_3
11. j_{rand} = randint(1,n_2) *//n_2 is the number of LL variables*
12. **For** j =1 **to** n_2 **do**
13. ♦ $v_{i,j,t+1} = \begin{cases} y_{r3,j,t} + \mathcal{F}(y_{r1,j,t} - y_{r2,j,t}) & \text{if rand(0,1) < CR or } j = j_{rand} \\ y_{i,j,t} & \text{otherwise} \end{cases}$
14. **End For** j
15. **End For** i
16. Insert in Eff the mutually nondominated $(x, v_{i,t+1})$, $i = 1,..., Nl$ that satisfy $g(x, v_{i,t+1}) \le 0$ and are not dominated by any member of Eff. Delete solutions that become dominated in Eff
17. **For** i =1 **to** Nl **do**
18. **If** ACCEPT_LL $v_{i,t+1}$ **then** *//depends on x, $v_{i,t+1}$, $y_{i,t}$, i , Eff*
19. $y_{i,t+1} \leftarrow v_{i,t+1}$
20. **Else**
21. $y_{i,t+1} \leftarrow y_{i,t}$
22. Insert $y_{i,t+1}$ into $Popy_{t+1}$
23. Update the incumbent solutions y' and y'' as in Step 4
24. **End For** t

Output: y' and y''

In step 2, for each new $y_{i,t} \in Y$ randomly generated, the lower level constraints $g(x, y_{i,t}) \le 0$ are checked; if the constraints are violated then another $y_{i,t} \in Y$ is drawn. If the first and the second trials are infeasible, the solution with smaller overall violation of constraints g is selected. In step 13, if $v_{i,j,t+1}$ does not satisfy the bounds defined by Y, then it is projected into the closest bound.

As in Algorithm 1, the steps marked with ♦ change from one DE variant to the other. In Steps 8 and 10, *DE/rand/1/bin* randomly selects $r_3 \in \{1,\ldots,Nl/2\}$ for $i \leq Nl/2$ and $r_3 \in \{Nl/2+1,\ldots,Nl\}$ for $i > Nl/2$. The *DE/best/1/bin* variant defines $y_{best} = y'$ for $i \leq Nl/2$ and $y_{best} = y''$ for $i > Nl/2$; y_{best} is used in Step 13 to replace y_{r3}.

The acceptance criterion in step 18 (ACCEPT_LL) determines whether the new individual $v_{i,t+1}$ is accepted or not to replace $y_{i,t}$ in the next population. The acceptance criterion firstly observes whether the solutions $(x, v_{i,t+1})$ and $(x, y_{i,t})$ satisfy the lower level constraints $g(x,y) \leq 0$ (g-feasibility), privileging the feasible solution if one of them is infeasible. If both are g-feasible, then it is checked whether they are nondominated w.r.t. to the current set of solutions *Eff*. If one of the solutions $v_{i,t+1}$ or $y_{i,t}$ is nondominated (i.e., it belongs to *Eff*) and the other is dominated, the nondominated solution is selected. If both solutions have the same status, the selection is based upon the upper level objective function value: for $i \leq Nl/2$ (sub-population oriented to the *Optimistic* frontier) the individual with lowest F is selected; for $i > Nl/2$ (sub-population oriented to the *Pessimistic* frontier) the individual with highest F is selected. It is worthwhile to note that, in an initial version of the algorithm, we did not use the set *Eff* in the acceptance criterion of $v_{i,t+1}$. The algorithm only compared the two candidate solutions, $v_{i,t+1}$ and $y_{i,t}$, checking whether one dominated the other or both were nondominated w.r.t. to each other. However, the algorithm revealed a very poor convergence of the population to nondominated solutions, which was overcome with the current strategy.

5 Computational Experiment

The SVBLDE algorithm has been compared with the PSO algorithm in [1], which was extended to compute also the *rewarding* solution as this algorithm had been originally designed to determine the other three extreme solutions. Below we shortly designate the *optimistic*, *pessimistic*, *deceiving* and *rewarding* solutions by *sol.o*, *sol.p*, *sol.d* and *sol.r*, respectively (with F^o, F^p, F^d and F^r being the respective upper level objective values).

To test and compare the algorithms we have considered two sets of problems. The first set includes 4 problems – *Prob.1* to *Prob.4* – whose formulations and *sol.o*, *sol.p* and *sol.d* are presented in [1]; these problems were adapted from the MOBL problems in [12] by considering only one upper level objective function. All the problems have one upper level variable and two lower level objective functions. Below we briefly describe these problems by indicating the number of lower level variables (n_2) and showing the values of F^o, F^p, F^d and F^r.

Prob.1 – $n_2 = 2$; *sol.o* ≠ *sol.d* ≠ *sol.p* with F^o=0.5, F^d=1.25, F^p=1; *sol.r* = *sol.p*, so F^r=1.

Prob.2 – generalization of *Prob.1* with $n_2 = k$. We consider k =14. The extreme solutions have the same characteristics as in *Prob.1* and the same upper/lower level objective values.

Prob.3 and *Prob.4* have $n_2 = 2$ and differ from each other in the upper level objective function. They include an upper level constraint *G* depending on lower level variables, which increases their difficulty. *Prob.3*: *sol.o* = *sol.r* with $F = -2$ and *sol.d* = *sol.p* with $F = -1$. *Prob.4*: this problem admits alternative *pessimistic* solutions (all with $F^p = 0$) but with different outcomes for the corresponding *rewarding* solution (with $F = -\alpha$, $0 \leq \alpha \leq 1$). The best *rewarding* solution corresponds to *sol.p* = *sol.d*, $F^p = F^d = 0$, being the *rewarding* solution *sol.r* = *sol.o* with $F^r = F^o = -1$.

The second group of test problems are the MOBL problems *DS1* to *DS5* in [13], originally proposed in [14]. We consider only F_1 for the upper level objective function. This is a set of scalable problems with a variety of complex features to the algorithms. Problems *DS1*-*DS3* have *k* upper level and *k* lower level variables – we consider k=5. Problems *DS4* and *DS5* have one upper level variable and $k+l$ lower level variables – we consider k=3 and l=2. All the other parameters were set as in [13]. The corresponding values of F^o, F^p, F^d and F^r are presented in Table 1.

We have considered the following parameters for both algorithms, which were tuned through experimentation: Nu -N_l -Tu -T_l equal to 20-60-50-100 for the first set of problems except *Prob.2*; 20-100-50-100 for *Prob.2*, *DS4* and *DS5*, which also have one upper level variable but more than 2 lower level variables; 100-100-100-100 for *DS1* to *DS3*, which have a higher number of upper level variables. Specific parameters of the PSO algorithm were set as in [1]. We performed 30 independent runs of each algorithm in each problem.

Concerning the DE variants of the SVBLDE algorithm, we observed that the results of *DE/rand/1/bin* were not statistically different from the results of *DE/best/1/bin* in about half of the cases; however, *DE/rand/1/bin* provided very poor results in a few other cases. Therefore, and due to space reasons, we omit herein the results of that variant. Table 1 presents the median and the interquartile range IQR of the *F* values obtained for the four extreme solutions over the 30 runs using the variant *DE/best/1/bin* of SVBLDE and the PSO algorithm. We also include the exact values of *F* (obtained analytically), which are very useful to assess the quality of the results obtained. The non-parametric Mann-Whitney test has been applied to assess whether the differences of the *F* values obtained with the two algorithms are statistically significant, considering a confidence level of 95%. The best result for each solution is highlighted in bold if the difference is statistically significant ('+' in the last column).

Table 1. Median and interquartile range of *F* in 30 independent runs for each algorithm.

		SVBLDE		PSO algorithm			M-W
		Median *F*	IQR *F*	Median *F*	IQR *F*	**Exact *F***	test
Prob.1	*Sol.o*	**0.497384**	0.000708	0.496248	0.001489	**0.5**	+
	Sol.p	0.993762	7.04E-05	0.993742	9.36E-05	**1**	-
	Sol.d	1.246284	0.01238	1.246038	0.016857	**1.25**	-
	Sol.r	0.993713	0.004252	0.988769	0.021896	**1**	-
Prob.2	*Sol.o*	**0.487397**	0.002296	0.407539	0.026634	**0.5**	+
	Sol.p	**0.999220**	0.011627	0.991885	0.000691	**1**	+
	Sol.d	**1.250138**	0.035337	1.202307	0.069874	**1.25**	+
	Sol.r	0.908306	0.057767	**0.98603**	0.007036	**1**	+
Prob.3	*Sol.o*	-2	0.006734	-1.99995	4.01E-05	**-2**	-
	Sol.p	-0.99985	0.001206	-0.99984	0.000215	**-1**	-
	Sol.d	-1.00214	0.003443	-1.00296	0.001645	**-2**	-
	Sol.r	-1.95307	0.101921	**-1.99036**	0.002566	**-1**	+

Prob.4	*Sol.o*	-0.99694	0.007655	-0.99995	5.12E-05	**-1**	-
	Sol.p	**-0.00356**	0.001833	-0.00606	0.000695	**0**	+
	Sol.d	-0.00334	0.001618	-0.00391	0.001138	**0**	-
	Sol.r	**-0.96020**	0.092859	-0.89689	0.111896	**-1**	+
DS1	*Sol.o*	**2,51E-05**	3,1E-05	5,61E-05	3,11E-05	**0**	+
	Sol.p	0,07746	0,056167	**0,099769**	0,000128	**0.1**	+
	Sol.d	0,092602	0,061951	**0,099981**	0,000168	**0.1**	+
	Sol.r	**3,23E-05**	0,000107	0,000179	0,000193	**0**	+
DS2	*Sol.o*	**-0,25977**	0,013058	-0,34826	0,035846	**-0.238773**	+
	Sol.p	-0,23876	8,07E-06	**-0,23877**	2,68E-06	**-0.238773**	+
	Sol.d	-0,23873	0,000247	**-0,23877**	9,51E-07	**-0.238773**	+
	Sol.r	-0,23876	1,36E-05	**-0,23878**	0,091723	**-0.238773**	+
DS3	*Sol.o*	**1.85E-07**	2.64E-07	5.34E-05	9.29E-05	**0**	+
	Sol.p	1.84E-07	1.03E-07	**0.200086**	1.65E-05	**0.2**	+
	Sol.d	1.86E-07	1.83E-07	**0.200299**	0.000192	**0.2**	+
	Sol.r	**1.99E-07**	2.65E-07	0.001461	0.004658	**0**	+
DS4	*Sol.o*	**0**	0	0	0.845635	**0**	+
	Sol.p	102	0	102	0	**102**	-
	Sol.d	**204**	0	204	100.5451	**204**	+
	Sol.r	**1.000245**	0.000347	2.388914	0.519674	**1**	+
DS5	*Sol.o*	**0.760132**	0.000174	2.01667	0.021934	**0.76**	+
	Sol.p	102	0	102	0	**102**	-
	Sol.d	**188.9164**	27.86941	102	0	**167.3**	+
	Sol.r	**1.000139**	0.000134	2.268107	0.36943	**1**	+

It is noteworthy that there are several difficulties in evaluating results to SVBLP. These difficulties can easily lead to pitfalls in the interpretation of results, which may be very difficult to avoid in general problems for which the exact solutions are not known. We draw attention to some of these pitfalls:

- Only efficient (Pareto optimal) solutions to the lower level problem are feasible to the SVBLP. Therefore, an algorithm may yield apparently better solutions (for any of the four extreme solutions), i.e. with lower *F* values, but the solutions are invalid because they are not efficient to the lower level problem.
- Even if only efficient (Pareto optimal) solutions to the lower level problem are obtained, other difficulties arise in assessing the *pessimistic* and *deceiving* solutions. Solutions with lower *F* values (i.e., which seem to be better) may be false because they are not in the *Pessimistic* frontier, i.e., they are not the worst for the leader for that setting of *x*. We can observe this situation in Table 1 for several *sol.d* and *sol.p* (e.g., *Prob.2*, *Prob.4, DS1, DS3* or *DS5*).

From Table 1, we observe that SVBLDE outperformed the PSO algorithm in 17 out of the 36 cases (4 extreme solutions to 9 problems) while the PSO algorithm outperformed SVBLDE in 9 cases (the differences in the other 10 cases were not statistically significant). Therefore, SVBLDE seems to perform slightly better than the PSO algorithm. We can also observe that SVBLDE is very effective in approximating the *optimistic* solution, being always better or equal to the PSO algorithm, but SVBLDE reveals more difficulty in attaining the real *pessimistic* and *deceiving* solutions in several cases.

6 Conclusions

We presented a new DE algorithm to compute the optimistic/deceiving and pessimistic/rewarding solutions to the SVBLP. These four extreme solutions capture the optimistic vs. pessimistic leader's attitude and possible follower's reactions more or less favorable to the leader. The DE approach seems to perform slightly better than the PSO-based approach, but the results do not evidence a clear performance advantage of the SVBLPDE algorithm with respect to PSO. The experiments unveiled some pitfalls associated with the interpretation of results and assessment of the algorithm performance in SVBLP. These pitfalls could be avoided because we were able to determine analytically the exact solutions to the problems tested. Research is underway on techniques to mitigate these pitfalls in general problems, which are nevertheless intrinsic to this kind of problems and cannot be entirely avoided.

Acknowledgment. This work was supported by projects UID/MULTI/00308/2013 and SAICTPAC/0004/2015-POCI-01-0145-FEDER-016434.

7 References

[1] Alves MJ, Antunes CH, Carrasqueira P. A PSO Approach to Semivectorial Bilevel Programming: Pessimistic, Optimistic and Deceiving Solutions. Proc. Genet. Evol. Comput. Conf. (GECCO 2015), 2015, p. 599–606.

[2] Bonnel H. Optimality conditions for the semivectorial bilevel optimization problem. Pacific J Optim 2006;2:447–68.

[3] Bonnel H, Morgan J. Semivectorial bilevel optimization problem: penalty approach. J Optim Theory Appl 2006;131:365–82.

[4] Ankhili Z, Mansouri A. An exact penalty on bilevel programs with linear vector optimization lower level. Eur J Oper Res 2009;197:36–41.

[5] Zheng Y, Wan Z. A solution method for semivectorial bilevel programming problem via penalty method. J Appl Math Comput 2011;37:207–19.

[6] Ren A, Wang Y. A novel penalty function method for semivectorial bilevel programming problem. Appl Math Model 2016;40:135–49.

[7] Calvete H, Galé C. On linear bilevel problems with multiple objectives at the lower level. Omega 2011;39:33–40.

[8] Liu B, Wan Z, Chen J, Wang G. Optimality conditions for pessimistic semivectorial bilevel programming problems. J Inequalities Appl 2014;2014:41.

[9] Lv Y, Chen J. A Discretization Iteration Approach for Solving a Class of Semivectorial Bilevel Programming Problem. J Nonlinear Sci Appl 2016;9:2888–99.

[10] Alves, M. J., Antunes CH. An illustration of different concepts of solutions in semivectorial bilevel programming. 2016 IEEE Symp. Comput. Intell. (SSCI); 2016.

[11] Mezura-Montes E, Velázquez-Reyes J, Coello Coello CA. A comparative study of differential evolution variants for global optimization. Proc. 8th Annu. Conf. Genet. Evol. Comput., 2006, p. 485–92.

[12] Deb K, Sinha A. Solving bilevel multi-objective optimization problems using evolutionary algorithms. Proc. EMO 2009, LNCS 5467, Springer; 2009, p. 110–24.

[13] Sinha A, Malo P, Deb K. Approximated set-valued mapping approach for handling multiobjective bilevel problems. Comput Oper Res 2017;77:194–209.

[14] Deb K, Sinha A. Constructing test problems for bilevel evolutionary multi-objective optimization. 2009 IEEE Congr. Evol. Comput., 2009, p. 1153–60.