Differential Evolution and Lemke's Algorithm in the Solution of Bilevel Programming Problems

Celio H.N. Larcher Jr. and Helio J. C. Barbosa

Laboratório Nacional de Computação Científica, MCTIC
{clarcher,hcbm}@lncc.br

Abstract. Bilevel optimization problems rise great interest, given their ability to model hierarchical decision structures. However, the challenges in their solution require the development of appropriate computational techniques. This work aims at a class of bilevel problems where the objective function of the follower agent is linear or quadratic in the follower variables. Using the Karush-Kuhn-Tucker conditions, the follower level is solved as a linear complementarity problem, giving rise to a nested approach. The follower response is obtained by Lemke's algorithm, while the leader level is handled by the differential evolution metaheuristic.

Keywords: Bilevel Programming, Differential Evolution, Lemke's Method

1 Introduction

The bilevel programming problem (BLPP) is a variation of the general optimization problem. It includes some characteristics of bi-matrix games, and provides an interesting way to model a large variety of problems including defensive planning [8], facilities location [7], optimal pricing strategy [11], among others.

Due to their various applications, the solution of bilevel programming problems is of great practical relevance. A BLPP can be characterized as an optimization problem which has another optimization problem included in its constraints. Each one of these optimization problems has its own variables, objectives, and constraints. This formulation gives rise to a hierarchical structure where in the upper level an agent (also called the leader) makes its choices while in the lower (inner) level another agent (also called the follower) responds taking into account the leader decision and its own interests. Then, for each leader decision a new optimization problem has to be solved. In fact, the BLPP is NP-Hard [10].

Considering these challenges, a first possibility is to use the Karush-Kuhn-Tucker (KKT) conditions so that the lower level is eliminated transforming the BLPP into a single-level optimization problem. Then, standard techniques can be used, but difficult complementarity constraints are added to the problem [6].

In recent years, metaheuristics became popular in the BLPP context. The most natural way to implement these techniques is through a nested algorithm, where each level is solved using a specific approach. Some proposals employ metaheuristics in both levels [3] while others use a mathematical framework to solve one of them [13].

Due to the large cost for solving multiple optimization problems in the BLPP, the idea of using an approximate solution for the follower level, by means of a metaheuristic or even metamodels, for example, can save valuable resources [3,22]. However, this gives rise to a major risk when the follower level is not solved correctly, which can lead to a deceptive solution, where the leader value is actually lower (thus seems better) than the real minimum [2]. In another alternative for the nested approach, one can combine the good performance and convergence properties of a given mathematical method with the great generality of a metaheuristic. Some works along this line are [13,14].

Another major issue occurs when the follower objective function has multiple minima. In this case, the problem lies in defining the follower response in view of the multiple possibilities and chance of multiple values in the leader function for these responses. A common way to deal with this issue is by means of optimistic/pessimistic strategies: in the optimistic strategy the leader assumes that the follower will respond with the best option for the leader, while in the pessimistic strategy the leader assumes the opposite: the follower will respond with the worst solution for the leader objective. The decision of what strategy should be used depends on the context.

Here, the class of BLPPs where the follower level is linear or quadratic in the follower variables, and only linear constraints are present in this level is considered. Then, a nested approach, referred to here as DE+LEMKE, is proposed where the upper level is solved by the differential evolution (DE) metaheurisc and the lower level is solved by Lemke's method applied to the resulting linear complementarity problem (LCP).

Lemke's method has already been used, combined with a genetic algorithm, in [13], reaching good results in a smaller set of small instances with no major challenges. As a result, a direct comparison between [13] and the proposed DE+LEMKE technique is not performed here, due to the facts that (i) even for the same amount of resources the instances considered in [13] are not sufficient to provide a good comparison between them that would allow for sound and generalizable conclusions, and (ii) there are no results in [13] concerning the cost of Lemke's method.

Besides, although the DE+LEMKE may look similar to the technique proposed in [13], this last work is limited to those cases where for each leader configuration it is always possible to find a follower response, and no mechanisms are provided to treat the cases when this assumption is false, that is, when there are leader choices without a possible response for the follower level. Another issue is that [13] just shows a superficial analysis for Lemke's model associated with bilevel programming, without including any consideration on the cost of Lemke's method, which is another contribution given here.

The paper is organized as follows: Section II presents the mathematical formulation used for the class of BLPPs considered; Section III describes the proposed approach (DE+LEMKE); Section IV presents the computational results obtained, and Section V concludes the paper with some possible future works.

2 Mathematical Formulation

The formulation of the class of BLPPs considered here reads:

$$\begin{array}{ll} \min_x & F(x,y) \\ \text{s.t.} & G(x,y) \leq 0 \\ & \min_y \quad f(x,y) = \frac{1}{2} y^T Q(x) y + c(x)^T y + d(x) \\ & \text{s.t.} \qquad A(x)y + b(x) \leq 0 \end{array} \tag{1}$$

where $x \in \Re^{n_1}$, $y \in \Re^{n_2}$. $F(x,y) : \Re^n \to \Re$ and $f(x,y) : \Re^n \to \Re$, $n = n_1 + n_2$ are the upper and lower level optimization functions, $G(x,y) : \Re^n \to \Re^{m_1}$ are the upper level constraints, $Q(x) \in \Re^{n_2 \times n_2}$ is a symmetric matrix determined by the leader configuration, $c(x) \in \Re^{n_2}$ and $d(x) \in \Re$ are the linear and constant terms in the quadratic function, $A(x) \in \Re^{m_2 \times n_2}$ and $b(x) \in \Re^{m_2}$ are the linear constraint terms induced by the leader configuration.

As in the follower level one has a quadratic (in the follower variables) optimization problem, it can then be replaced by the corresponding KKT optimality conditions leading to

$$\begin{array}{ll} \min_{x,y,\lambda} & F(x,y) \\ \text{s.t.} & G(x,y) \leq 0 \\ & Q(x)y + A(x)^T \lambda_i + c(x) = 0 \\ & A(x)y + b(x) + u = 0 \\ & \lambda_i u_i = 0 \\ & \lambda \geq 0, u \geq 0 \end{array} \tag{2}$$

For the formulation proposed, the follower constraints are convex for every leader configuration. The follower function may be convex if the $Q(x)$ matrix is, at least, positive semi-definite. In addition, if the follower function is strictly convex the KKT conditions have one, and just one, solution.

Adding a non-negativity condition for the follower variable, the KKT conditions can be written as an LCP

$$w - Mz = q \quad w, z \geq 0 \quad w_i z_i = 0 \quad \forall i \tag{3}$$

and the original bilevel problem is transformed into a single-level optimization problem with an LCP among its constraints:

$$\begin{array}{ll} \min_{x,y,\lambda} & F(x,y) \\ \text{s.t.} & G(x,y) \leq 0 \\ & \lambda'' - (Q(x)y + A(x)^T \lambda_i') = c(x) \\ & u - (-A(x)y + 0\lambda_i') = -b(x) \\ & \lambda_i' u_i = 0, \quad \lambda_j'' y_j = 0 \\ & \lambda', \lambda'', u, y \geq 0 \end{array} \tag{4}$$

where

$$w = \begin{pmatrix} \lambda'' \\ u \end{pmatrix}, z = \begin{pmatrix} y \\ \lambda' \end{pmatrix} M = \begin{pmatrix} Q & A^T \\ -A & 0 \end{pmatrix}, q = \begin{pmatrix} c \\ -b \end{pmatrix}$$

In this way, solving the LCP for some leader configuration is equivalent to finding a KKT point for the above system. In addition, if this point is unique it is also the minimal global solution for the follower minimization problem.

An immediate issue about the non-negativity constraint in the follower variables can be easily bypassed replacing a y variable for the relation $y = y' - y''$.

3 The Proposed Approach

A nested approach, referred here to as DE+LEMKE, is proposed for the BLPP: the upper level is solved by the differential evolution (DE) metaheuristic, and the lower level is solved via Lemke's method applied to the LCP problem in (4).

In this way, for each leader configuration, Lemke's method is applied in order to find the correct follower reaction. The Algorithm 1 summarizes the technique. The "rand" function gives a random number, within a specified range, assuming a uniform distribution.

Algorithm 1 DE+LEMKE

Ensure: F, CR , NP
1: $X_0 \leftarrow$ Population_Random_Init(NP);
2: $Y_0 \leftarrow$ Lemke(X_0), $G \leftarrow 0$;
3: **while** unsatisfied stop criteria **do**
4: $G \leftarrow G + 1$;
5: **for** i$\leftarrow 1 \ldots$ NP **do**
6: $\{x_{r_1,G}, x_{r_2,G}, x_{r_3,G}\} \leftarrow$ Select_Individuals(X_G); /*$r_1 \neq r_2 \neq r_3$*/
7: $x_{MUT} \leftarrow x_{r1,G} + F(x_{best,G} - x_{r1,G}) + F(x_{r2,G} - x_{r3,G})$;
8: jrand $\leftarrow$ rand$[1 \ldots N]$; /* N: dimension in parametric vector */
9: **for** $j \leftarrow 1 \ldots N$ **do**
10: $S \leftarrow$ rand$[0 \ldots 1.0]$;
11: **if** $S \leq CR$ or jrand $= j$ **then**
12: $x^j_{REC} \leftarrow x^j_{MUT}$;
13: **else**
14: $x^j_{REC} \leftarrow x^j_{i,G}$;
15: **end if**
16: **end for**
17: $y_{REC} \leftarrow$ Lemke(x_{REC});
18: **if** $f_{DEB}(x_{REC}, y_{REC}) \leq f_{DEB}(x_{i,G}, y_{i,G})$ **then**
19: $x_{i,G+1} \leftarrow x_{REC}$;
20: **else**
21: $x_{i,G+1} \leftarrow x_{i,G}$;
22: **end if**
23: **end for**
24: **end while**
25: **return** best fitness leader evaluation in X_G;

The main steps for the DE+LEMKE approach follow:

Step 1: The population is generated randomly, with leader values given within the bounds assigned in each variable. For each of these initial candidate solutions, Lemke's method computes the correct follower response.

Step 2: For each individual in a new population $G + 1$ three individuals are selected in the current generation G: the best element in this generation $x_{best,G}$ and two distinct randomly select individuals $x_{r1,G}$ and $x_{r2,G}$. These individuals are used in the rand-to-best/1 mutation scheme giving rise to the x_{MUT} solution.

Step 3: For each mutated individual (x_{MUT}) the binomial crossover operation is applied, combining the components in the vector x_{MUT} with those of x_i, a reference individual for the current generation. This operation creates the individual x_{REC}, to whom is attributed the exact response of the follower computed by Lemke's method.

Step 4: The new individual x_{REC} is evaluated and compared with the reference individual x_i; the best individual survives to the next generation $G + 1$.

Step 5: The process ends when the stop criteria are reached. Otherwise it returns to Step 2.

A more detailed view on DE and on Lemke's process follows.

3.1 Lemke's Method

Lemke's algorithm was proposed by Carlton Lemke in [12] for solving the LCP problem (3). The main idea in this method is to find a basic feasible solution for the LCP problem by means of successive pivoting operations on an initial almost feasible LCP solution which is constructed by inserting an artificial variable. This variable has to be driven out by the pivoting operations so that the system solution can be found.

There are some issues in the use of this technique in the BLPP context. The main one is that it is not trivial to define an optimistic/pessimistic strategy when the lower level has multiple global minima.

Another difficulty occurs when the follower level has no response for a given leader configuration. In this case, it is not possible to estimate an approximate solution, preventing any considerations about constraint violations.

Despite those potential difficulties, for all test-problems used in this work Lemke's method was capable of leading the DE+LEMKE approach to converge to the known optimal solution.

3.2 DE implementation

The DE in this work uses the DE/rand-to-best/1/bin variant and is applied only at the leader level, not involving the follower variables.

The rand-to-best/1 mutation and the binary recombination were chosen due to the fact that this was the combination with best performance among the tested variants. The rand-to-best/1 strategy has shown a good performance considering the amount of function evaluations used despite the possibility of premature convergence. While the binary crossover improves to some extent the results reached in these experiments.

In this implementation, the initial population is randomly generated taking into account the bounds in the variables.

For constraint handling, Deb's method [9] is used with some modifications for the BLPP context. In the standard Deb method, feasible solutions are always better evaluated when compared with infeasible solutions. For two feasible solutions, their fitness value is compared and the best one is considered the better. Finally, for two infeasible solutions the sum of its violations is computed and the one with the lowest value is considered the winner.

Considering the infeasibility for the lower level a new comparison step is added to Deb's method. Solutions with infeasible lower level are considered incomplete solutions. Thus, no consideration can be made about violations in the constraints, knowing only that there is no solution for the follower which does not violate any constraint in the lower level. In the adapted version of Deb's method, incomplete solutions are considered the inferior solutions when compared with both infeasible and feasible solutions.

When comparing incomplete solutions, the sum of bound violations for the leader variables is computed, and the one with the smaller value is considered the winner. The following equation summarizes these previous considerations.

$$F_{DEB}(x,y) = \begin{cases} F(x,y) & x \text{ is feasible} \\ M + \sum_{i=1}^{m_1} \max(G_i(x,y),0) & x \text{ is infeasible just for the leader constraints} \\ 2M + \sum_{i=1}^{n_1} \max(x_i - u_i, l_i - x_i, 0) & x \text{ is an incomplete solution} \end{cases}$$

were M is an arbitrarily large constant to guarantee that every infeasible solution is worse than a feasible solution (the same for infeasible versus incomplete solutions), (x, y) are the leader value and the follower response in a given solution, $G_i(x, y)$ are the leader constraints and the terms u_i and l_i are the upper and lower bounds for the variable x_i. Note that the leader objective function evaluation is needed only for feasible solutions. That allows the approach to save resources by avoiding the evaluation of solutions a priori known as infeasible.

As the mutation operation does not have any provision for satisfying the bounds, any bound violation is dealt with as any other constraint violation, by Deb's method. The binomial crossover operation is used in the traditional way, with no adaptations needed.

4 Computational Results

The definition of the parameters used in the DE+LEMKE approach was made using the "irace" framework [15]. For the tuning procedure, a subset of the test functions was selected and fifteen thousand samples were performed to obtain the ideal parameter set.

As a result, the following parameters were used in the experiments (in brackets, the allowed range for each parameter): population size NP=20 $\in [15, 80]$,

weighting coefficient for mutation $F = 0.7 \in [0.5, 1.0]$, and crossover probability $CR = 0.6 \in [0.3, 1.0]$.

For the experiments, 18 test functions were selected. The Table 1 shows where the formulation for each test can be found.

Table 1: Location and reference for the test-problems used.

Test	Reference	Location
Pr1	[20]	Section V – Example 2
Pr2	[6]	Section 5.2 – Example 2
Pr3	[1]	Section III – Example 2
Pr4	[4]	Section 2 – Example 1
Pr5	[4]	Section 4 – Example 3
Pr6	[19]	Section 3 – Example
Pr7	[5]	Section 8.1.1 – Example 8.1.3
Pr8	[18]	Section 3 – Problem 2
Pr9	[5]	Section 5.3.2 – Example 5.3.1
Pr10	[10]	Section 5 – Example
Pr11	[16]	Section 6 – Example
Pr12	[24]	Section 8 – Example 3
Pr13	[4]	Section 4 – Example 2
Pr14	[25]	Section V – Function 20
Pr15	[25]	Section V – Function 26
Pr16	[17]	Section 3 – Example 5
Pr17	[17]	Section 4 – Example 4
Pr18	[17]	Section 4 – Example 6

The first experiment aims at checking the suitability of Lemke's method in the solution of the follower level. For 50 independent runs for all the 18 instances using the DE+LEMKE algorithm, the number of pivot operations was recorded and compared with the dimension and number of constraints at each level. The Table 2 summarizes the results obtained in this experiment. The columns "Dim" and "Ctr" show the dimension and the number of constraints in each level; the column "Lemke's Matrix" indicates the size of the *tableau* built for each problem, while the "Pivot Operation" columns lists the Max, Mean, Min and SD values of this factor.

It may be noted that there is not a well-defined pattern for dimension size or number of constraints in this table. However, considering the small variation in the values for these attributes, some trends can be observed.

The plot in the Figure 1 shows the pivot operations compared with dimension and numbers of constraints in the leader level for each instance. The plot in the Figure 2 shows the same attributes, but now for the follower level.

Observing these plots the problem dimension appears to be related to the number of pivoting operations, as well as the number of constraints in the follower level. With respect to the leader, the number of constraints seems to be of lesser relevance.

Table 2: Pivot operations and problem behavior

Test	Leader Dim	Leader Ctr	Follower Dim	Follower Ctr	Lemke's Matrix	Pivot Operation Max	Pivot Operation Mean	Pivot Operation Min	Pivot Operation SD
Pr1	2	3	2	0	6x14	4	3.00	1	0.08
Pr2	2	0	2	3	9x20	7	4.22	1	0.42
Pr3	2	1	2	2	14x30	6	3.05	2	0.48
Pr4	1	0	1	3	6x14	3	2.07	1	0.28
Pr5	2	1	2	2	8x18	7	3.04	3	0.31
Pr6	1	0	2	4	9x20	5	4.07	3	0.28
Pr7	1	0	1	0	3x8	1	1.00	1	0.00
Pr8	1	0	1	3	7x16	3	2.00	1	0.09
Pr9	2	0	3	3	9x20	5	4.11	2	0.35
Pr10	2	1	3	3	9x20	5	4.19	2	0.41
Pr11	4	6	2	4	8x18	7	2.33	1	2.33
Pr12	10	2	6	7	25x52	38	19.93	7	2.27
Pr13	4	1	4	4	16x34	12	8.69	4	1.10
Pr14	2	1	2	2	10x22	8	3.56	2	0.70
Pr15	1	0	2	4	8x18	6	4.21	2	0.63
Pr16	2	0	2	2	6x14	6	1.29	1	0.48
Pr17	1	0	2	2	7x16	4	3.99	1	0.12
Pr18	1	0	2	2	7x16	4	3.11	1	0.32

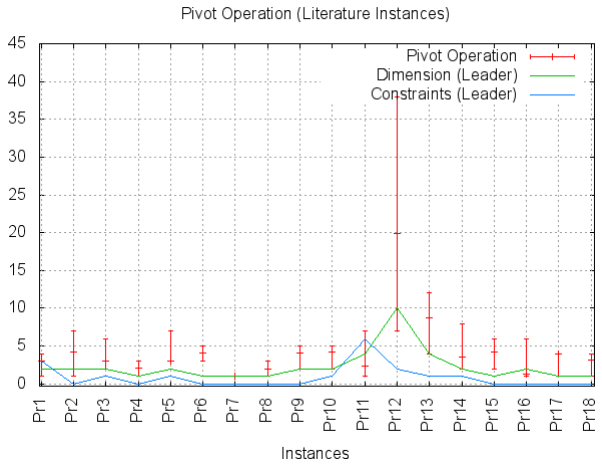

Fig. 1: Comparison of pivot operations with dimension and number of constraints for the leader level

The little correlation shown with the change of the number of leader constraints is somewhat expected, considering that this factor has little influence in the size of Lemke's *tableau*. In a similar sense it is expected that the leader dimension has little or no influence on the number of pivoting operations, but as in

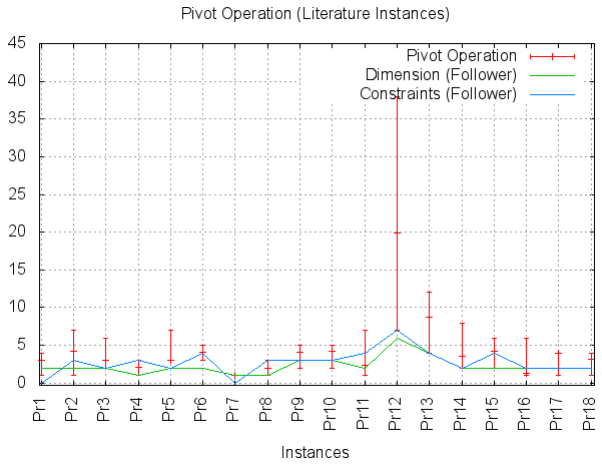

Fig. 2: Comparison of pivot operations with dimension and number of constraints for the follower level

these instances the changes in the leader's dimension factor are linked to changes in the follower's dimension it becomes difficult to verify this factor clearly. About the follower variables and constraints, the growth of pivoting operations according to the addition of these factors occurs as expected considering that they directly influence the size of Lemke's matrix. Another aspect to consider is that there is not a bad behavior in Lemke's method as the problem scales up. Even in the largest problem, the growth in the mean cost (given by pivot operations) is low, considering Lemke's Matrix size, which indicates some level of scalability.

A second experiment presents the performance achieved by the DE+LEMKE strategy comparing with the use of an inexact method in the follower level, the nested technique DE+DE.

For the DE+DE approach, the follower DE was as in the work of [3], with an stopping criterion by population variance:

$$\alpha = \sum_{i} \frac{\sigma^2(X_{i,G})}{\sigma^2(X_{i,init})}$$

with threshold of $\alpha < 10^{-5}$ [21]. The upper DE is implemented in the same way as can be seen in the DE+LEMKE.

The stop criteria in the leader level for both approaches was either 6000 upper level function evaluations or 10000 DE iterations. These two conditions are used considering that not every solution will be evaluated by DEB's criteria and the 10000 DE iterations can occur before the 6000 upper level function evaluations. Again, 50 independent runs are performed to obtain these results.

The Table 3 shows the objective function for the best, mean, median, and worst solutions found by the DE+LEMKE and DE+DE approaches, compared

with the best solution found in the literature ([Pr1-Pr8] [3], [Pr9-Pr15] [14], and [Pr16-Pr18] [13]).

Table 3: Summary of the performance of DE+LEMKE and DE+DE. The "a" superscript means a maximization problem at both levels.

Test	Technique	Best	Mean	Median	Worst	Literature
Pr1	DE+LEMKE	225.00	225.00	225.00	225.00	225.000
	DE+DE	224.93	224.96	224.96	224.98	
Pr2^a	DE+LEMKE	3.25	3.25	3.25	3.25	3.250
	DE+DE	4.00	3.30	3.25	3.25	
Pr3	DE+LEMKE	0.00	0.40	0.00	5.00	0.000
	DE+DE	-0.01	0.79	0.00	4.97	
Pr4	DE+LEMKE	17.00	17.96	17.00	25.00	17.000
	DE+DE	16.50	17.26	16.60	24.98	
Pr5	DE+LEMKE	-12.68	-12.68	-12.68	-12.65	-12.679
	DE+DE	-12.71	-12.69	-12.69	-12.67	
Pr6	DE+LEMKE	-1.21	-1.21	-1.21	-1.21	-1.210
	DE+DE	-1.21	-1.21	-1.21	-1.21	
Pr7	DE+LEMKE	1.00	1.00	1.00	1.00	1.000
	DE+DE	1.00	1.00	1.00	1.00	
Pr8	DE+LEMKE	5.00	5.00	5.00	5.00	5.000
	DE+DE	4.99	4.99	4.99	4.99	
Pr9	DE+LEMKE	-29.20	-29.20	-29.20	-29.20	-29.200
	DE+DE	-29.21	-29.20	-29.20	-29.20	
Pr10	DE+LEMKE	-18.41	-16.29	-16.00	-16.00	-18.400
	DE+DE	-18.73	-16.35	-16.03	-16.01	
Pr11	DE+LEMKE	14.99	14.99	14.99	14.99	14.990
	DE+DE	14.98	14.99	14.99	14.99	
Pr12	DE+LEMKE	-466.83	-454.38	-453.61	-453.41	-453.610
	DE+DE	-467.57	-453.91	-453.56	-448.06	
Pr13^a	DE+LEMKE	6600.01	6600.01	6600.01	6600.01	6600.000
	DE+DE	6600.02	6599.73	6599.87	6595.86	
Pr14	DE+LEMKE	0.00	0.60	0.00	5.00	0.000
	DE+DE	0.00	0.70	0.00	4.97	
Pr15	DE+LEMKE	0.00	0.00	0.00	0.00	0.000
	DE+DE	0.00	0.00	0.00	0.00	
Pr16	DE+LEMKE	-3.92	-3.89	-3.92	-3.79	-3.920
	DE+DE	-3.92	-3.89	-3.92	-3.79	
Pr17	DE+LEMKE	0.85	0.85	0.85	0.85	0.849
	DE+DE	0.59	0.74	0.75	0.77	
Pr18	DE+LEMKE	1.56	1.56	1.56	1.56	1.563
	DE+DE	1.56	1.56	1.56	1.56	

In these experiments it can be seen that DE+LEMKE easily solves half of the test-problems, finding always the best solution of the literature. For other instances, this approach can find the best solution at least one time, with the median being less than the best only in the Pr10 function. In addition, DE+LEMKE can improve the best solution in the case Pr12, which corroborates with its good performance. The DE+DE has a similar performance, but false optimal points can be found, specially in the Pr17 instance, where in all executions a false minimum was found. In fact, this is potentially much worse than finding a sub-optimal viable solution.

Note that, despite these previous analyzes, there is another aspect to consider about the scope for the two techniques, DE+LEMKE and DE+DE. While DE+DE can be applied to all types of functions and constraints in the leader

and follower levels, the DE+LEMKE approach can be applied only when the follower problem is linear or quadratic, with linear constraints, as mentioned before. However, whenever it is possible to choose between these two options, the experiments indicate that DE+LEMKE is a better option.

5 Conclusions

A new approach combing mathematical programming and metaheuristics was proposed for the class of bilevel programming problems with a quadratic follower function including linear constraints.

Rewriting the BLPP as an optimization problem with a linear complementarity problem among its constraints, this algorithm solves the follower level using Lemke's algorithm while the leader is submitted to the differential evolution metaheuristic.

The experiments show a good behavior for Lemke's method with a controlled growth of complexity with increased dimensions. The DE+LEMKE technique shows a good overall performance, more consistent than the DE+DE, in all the functions analyzed coming from the literature, always finding the best solutions recorded so far.

As future work, larger test-problems will be used aiming at verifying the scalability of the algorithm. Another objective is continuing to improve the DE+LEMKE approach with a restart procedure, as well as using other strategies like the adaptive population size used in the L-SHADE DE variant[23].

Acknowledgements

The authors would like to thank CNPq (grant 310778/2013-1).

References

1. Aiyoshi, E., Shimizu, K.: A solution method for the static constrained Stackelberg problem via penalty method. IEEE Trans. on Automatic Control 29(12), 1111–1114 (1984)
2. Angelo, J.S., Barbosa, H.J.C.: A study on the use of heuristics to solve a bilevel programming problem. Intl. Trans. in Operational Research 22(5), 861–882 (sep 2015)
3. Angelo, J.S., Krempser, E., Barbosa, H.J.C.: Differential evolution for bilevel programming. In: 2013 IEEE Congress on Evolutionary Computation. vol. 1, pp. 470–477. IEEE, Cancun (jun 2013)
4. Bard, J.F.: Convex two-level optimization. Mathematical Programming 40-40(1-3), 15–27 (jan 1988)
5. Bard, J.F.: Practical Bilevel Optimization, Nonconvex Optimization and Its Applications, vol. 30. Springer US, Boston, MA (1998)
6. Bard, J.F., Falk, J.E.: An explicit solution to the multi-level programming problem. Computers and Operations Research 9(1), 77–100 (1982)

7. Beresnev, V.: Branch-and-bound algorithm for a competitive facility location problem. Computers & Operations Research 40(8), 2062–2070 (2013)
8. Bracken, J., McGill, J.T.: Defense applications of mathematical programs with optimization problems in the constraints. Operations Research 22, 1086–1096 (1974)
9. Deb, K.: An efficient constraint handling method for genetic algorithms. Computer Methods in Applied Mechanics and Engineering 186(2-4), 311–338 (jun 2000)
10. Hansen, P., Jaumard, B., Savard, G.: New Branch-and-Bound Rules for Linear Bilevel Programming. SIAM Journal on Scientific and Statistical Computing 13(5), 1194–1217 (1992)
11. Labbé, M., Violin, A.: Bilevel programming and price setting problems. 4OR 11(1), 1–30 (mar 2013)
12. Lemke, C.E.: Bimatrix Equilibrium Points and Mathematical Programming. Management Science 11(7), 681–689 (may 1965)
13. Li, H., Wang, Y.: A Hybrid Genetic Algorithm for Solving a Class of Nonlinear Bilevel Programming Problems. In: Simulated Evolution and Learning, pp. 408–415. Springer Berlin Heidelberg (2006)
14. Li, H., Zhang, L.: A Differential Evolution with Two Mutation Strategies and a Selection Based on an Improved Constraint-Handling Technique for Bilevel Programming Problems. Mathematical Problems in Engineering 2014 (2014)
15. López-Ibánez, M., Dubois-Lacoste, J., Stützle, T., Birattari, M.: The irace package, iterated race for automatic algorithm configuration. Tech. Rep. TR/IRIDIA/2011-004, IRIDIA, Université Libre de Bruxelles, Belgium (2011)
16. Moshirvaziri, K., Amouzegar, M.A., Jacobsen, S.E.: Test problem construction for linear bilevel programming problems. J. of Global Optimization 8(3), 235–243 (1996)
17. Outrata, J.V.: On the numerical solution of a class of stackelberg problems. Zeitschrift für Operations Research 34(4), 255–277 (1990)
18. Rajesh, J., Gupta, K., Kusumakar, H.S., Jayaraman, V.K., Kulkarni, B.D.: A tabu search based approach for solving a class of bilevel programming problems in chemical engineering. Journal of Heuristics 9(4), 307–319 (2003)
19. Savard, G., Gauvin, J.: The steepest descent direction for the nonlinear bilevel programming problem. Operations Research Letters 15(5), 265–272 (1994)
20. Shimizu, K., Aiyoshi, E.: A new computational method for stackelberg and min-max problems by use of a penalty method. IEEE Trans. on Automatic Control AC-26(2), 460–466 (1981)
21. Sinha, A., Malo, P., Deb, K.: Unconstrained scalable test problems for single-objective bilevel optimization. In: 2012 IEEE Congress on Evolutionary Computation (CEC). pp. 1–8 (2012)
22. Sinha, A., Malo, P., Deb, K.: Efficient evolutionary algorithm for single-objective bilevel optimization. CoRR abs/1303.3901 (2013)
23. Tanabe, R., Fukunaga, A.S.: Improving the search performance of shade using linear population size reduction. In: 2014 IEEE Congress on Evolutionary Computation (CEC). pp. 1658–1665. IEEE, Beijing, China (July 2014)
24. Tuy, H., Migdalas, A., Hoai-Phuong, N.T.: A novel approach to bilevel nonlinear programming. Journal of Global Optimization 38(4), 527–554 (2007)
25. Wang, Y., Jiao, Y.C., Li, H.: An Evolutionary Algorithm for Solving Nonlinear Bilevel Programming Based on a New Constraint-Handling Scheme. IEEE Trans. on Systems, Man and Cybernetics, Part C 35(2), 221–232 (2005)