

## **SOLVING PESSIMISTIC BILEVEL OPTIMIZATION PROBLEMS WITH EVOLUTIONARY ALGORITHMS**

### **A PRELIMINARY STUDY**

**Margarita Antoniou<sup>1,2</sup> and Gregor Papa<sup>1,2</sup>**

<sup>1</sup>Jožef Stefan Institute,  
<sup>2</sup>Jožef Stefan International Postgraduate School  
Ljubljana, Slovenia  
{name.surname}@ijs.si

---

**Abstract.** *It is common in practice, an optimal solution of a decision-maker to depend heavily on the response of another decision-maker, formulating a bilevel optimization problem. The optimization of a bilevel problem aims to achieve the optimum solution of the upper-level, taking into consideration the optimal lower-level values too. When the lower-level problem is multi-modal, meaning that it has several global optima, an ambiguity about the optimal upper-level solution appears. The optimistic approach assumes that the follower will respond with an optimal solution, that is favorable by the upper-level as well. In the pessimistic approach, the upper-level is optimising for the worst case. Various evolutionary algorithms have been implemented successfully to solve the optimistic approach of the bilevel problem. To the best of our knowledge, these algorithms have not been extended to the pessimistic approach. In this paper, we use a multi-population nested Differential Evolution to solve the pessimistic bilevel problem when the lower-level has multiple global optima. The performance of the algorithm is examined by solving a test-problem taken from the literature.*

**Keywords:** Pessimistic Bilevel Optimization, Nested Optimization, Evolutionary Algorithms

---

## 1 INTRODUCTION

The bilevel optimization problem is an optimization problem, operating as the upper-level, that has another optimization problem as a constraint, at the lower-level. Bilevel problems present many difficulties in its mathematical formulation, as they typically do not obey simplistic properties such as convexity or continuity. Moreover, it is proven that even the simple case of a linear bilevel optimization problem is NP-hard [1]. The bilevel optimization problem can be also interpreted as a non-cooperative static Stackelberg game, as firstly introduced by von Stackelberg in 1934 in the context of unbalanced economic markets [2].

When the lower-level problem is multimodal, meaning that it has several global optima, an ambiguity about the optimal upper-level solution appears. The community has identified two approaches to tackle this uncertainty, the optimistic and the pessimistic approach [3]. When adopting an optimistic stance, the upper-level assumes that the lower-level will select the most optimal solution that is also optimal for it. In the pessimistic approach, the upper-level is optimising her objective making the worst-case assumptions about the decision of the lower-level.

The methods of solving the bilevel optimization problem can be divided into two main categories [3]. The first one refers to classical approaches, based on mathematical derivation methods, such as branch and bound [1], and is usually applied to well-behaved bilevel problems with specific properties. The second one is based on hybrid and evolutionary approaches, which have become more popular in the last years. Metaheuristic and evolutionary algorithms, in general, do not need to make any assumptions about the objective functions of the problem and can be applied to general bilevel problems [3]. Examples of this kind of Bilevel Evolutionary Algorithms (BLEAs) are BIDE with Differential Evolution (DE) in both levels[4], NBLEA with Genetic Algorithm in both levels[5], a memetic approach in [6], BLEAQ with genetic algorithm and quadratic approximations [7], BL-CMA-ES with CMA-ES in both levels [8], etc. However, all these evolutionary algorithms are adopting the optimistic approach for solving the BOP, as it is easier to track.

While indeed the solution of the optimistic approach is much more tractable, it comes with a drawback: the assumptions in the model, that a cooperation of the lower- with the upper-level exists without any reward considered in her objective, are not realistic. On the contrary, the pessimistic approach, while more conservative, can be interpreted as a bound and minimize the risk of the worst case and give more robust solutions. Therefore, finding solutions for the pessimistic case is of great importance in practice, when optimizing under uncertainty and especially under worst-case scenario optimization.

The pessimistic approach is generally considered much more difficult to solve, and the solution methodologies made for the optimistic approach are not directly applicable to it. A survey about the pessimistic bilevel optimization problem and its optimality conditions can be found in [9, 10, 11]. There are several studies in the classical category, proposing solution methods for pessimistic bilevel problems with specific mathematical properties, for example for the pessimistic bilevel linear problem [12], or a reduction method for the pessimistic quadratic-linear bilevel[13]. From the hybrid and evolutionary approaches, the semivectorial pessimistic bilevel problem has been approached with PSO [14] and Differential Evolution [15], with a multi-objective problem in the lower-level. To the best of our knowledge, there is no Evolutionary Algorithm studied to solve the general pessimistic bilevel optimization problem.

In this study, we aim to solve the pessimistic BOP by adopting EAs at both levels. The algorithm has a nested structure. Therefore, for solving each upper-level candidate solution, a lower-level optimization is conducted. The lower-level optimization algorithm has a multi-

population schema, that can be evaluated in parallel mode to decrease the computational cost. For each upper-level sample point, this generates a number of lower-level solutions-one for each population-, that are further evaluated according to the upper-level fitness function. From these solutions, the ones giving the minimum and maximum upper-level value (optimistic and pessimistic) are then passed to the upper-level. In the upper-level, a 2-ranking system is applied, where the best optimistic and pessimistic solutions are co-evolving. To further improve the accuracy and efficacy of the algorithm, a local search is conducted for every best solution found. The algorithm gives two final best solutions, one for the optimistic and one for the pessimistic approach. To test the method, the algorithm is tested on benchmark test-functions with known global optima.

The rest of the paper is organized as follows. Firstly, the optimistic and pessimistic problem are defined in Section 2. In Section 3 the proposed method and the steps of the algorithm are explained. In section 4 the experimental setup is reported along with the test function used and the relevant results. Finally, in Section 5 we conclude the paper and give some future steps of the research.

## 2 DEFINITIONS OF THE BILEVEL OPTIMIZATION PROBLEM

The bilevel optimization problem consists of two levels of optimization tasks, where two different sets of variables belong to each of these tasks. The mathematical representation is as follows :

$$\begin{aligned} \min_x \quad & F(x, y) \\ \text{s.t.} \quad & G_k(x, y) \leq 0, y \in \Psi(x) \end{aligned} \quad (1)$$

where  $\Psi(x)$  is the set of solutions of the following lower-level problem:

$$\begin{aligned} \min_y \quad & f(x, y) \\ \text{s.t.} \quad & g(x, y) \leq 0 \end{aligned} \quad (2)$$

where  $y$  is the solution of the lower-level problem from the set of solutions  $y \subseteq R^m$ , with regard to solution from upper-level,  $x$  from set of solutions  $x \subseteq R^n$ , where  $F$  represents upper-level' s objective function, while  $f$  represents the lower-level' s objective function. When more than one lower-level optimal solutions for all or some upper-level variables exist, the bilevel problem becomes ill-defined. To tackle this, the community is employing two approaches. In the optimistic approach, the upper-level is optimising expecting that the lower-level reaction will be optimal for her as well. In the pessimistic case, the upper-level is optimising under the worst-case scenario of the lower-level reaction. The general pessimistic bilevel problem can be formulated as follows:

$$\begin{aligned} \min_{x \in X} \max_{y \in \Psi(x)} \quad & F(x, y) \\ \text{s.t.} \quad & G(x, y) \leq 0 \end{aligned} \quad (3)$$

where  $X := x : G(x) \leq 0$ , and  $\Psi(x)$  is the set of solutions of the lower-level problem

$$\begin{aligned} \min_y \quad & f(x, y) \\ \text{s.t.} \quad & g(x, y) \leq 0 \end{aligned} \quad (4)$$

Here  $x \in R^n$  and  $y \in R^m$ . More about the definitions of both the approaches can be found in [16, 3].

### 3 PROPOSED METHOD: P-BLDE

The proposed method is an implementation of a Bilevel Differential Evolution algorithm with multi-population and aims to provide both the optimistic and pessimistic solutions of the bilevel problem. Therefore we call this algorithm Pessimistic Bilevel Differential Evolution (P-BLDE).

In the nested algorithm, for each upper-level individual, a lower-level optimization procedure is conducted. A flowchart of the upper-level and lower-level P-BLDE is presented in the Figures 1 and 2. A co-evolutionary DE is implemented for the upper-level optimization problem, and with the upper-level variables fixed for the population of optimistic and pessimistic individuals, a lower-level optimization DE is conducted, passes its optimistic and pessimistic optimal values to the upper-level, and then the sample points are evaluated. The constraint handling for both levels is done by ranking the individuals with the less violation as the best. More precisely, the steps of the algorithm are the following:

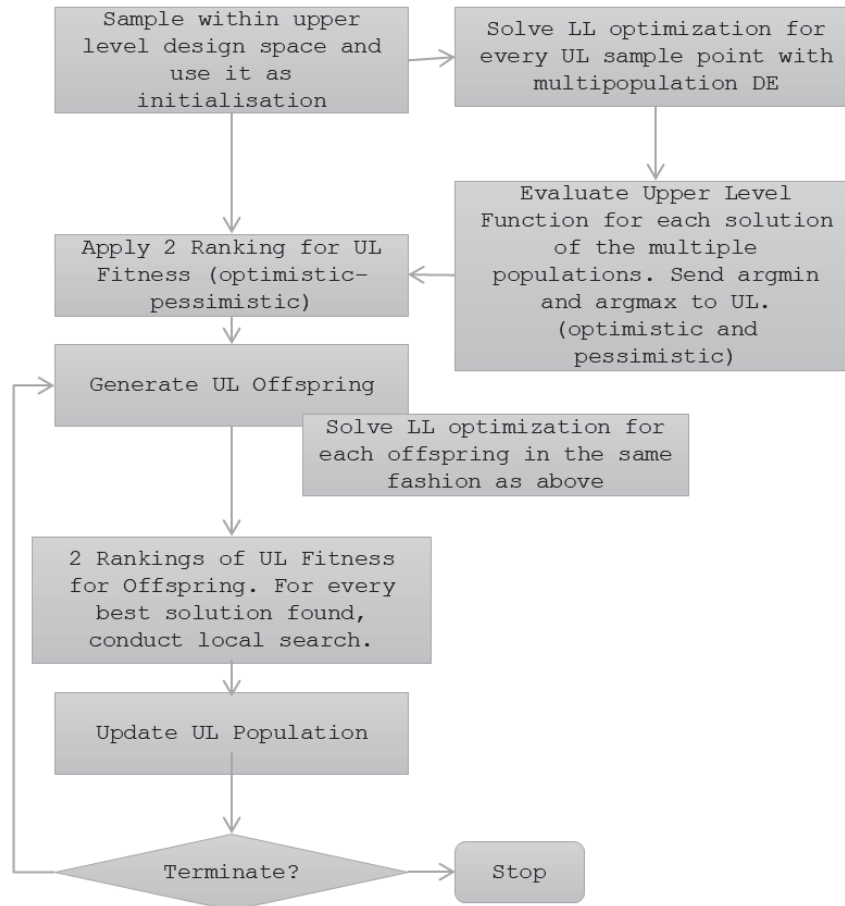


Figure 1: upper-level DE flowchart.

For the upper-level DE:

**Step 1: Initialization:** The algorithm starts by sampling in upper-level feasible space with a population of NP size for the optimistic and pessimistic sub-population respectively. Note that the populations are co-evolving independently. The lower-level variables are then found by the lower-level optimization procedure (the lower-level DE). For each

individual, lower-level DE responds with an optimistic and a pessimistic solution that is used to further evaluate the upper level.

- Step 2: Upper-level mutation and recombination (crossover): The upper-level individuals of each population are evolved with DE/rand/1/bin mutation strategy.
- Step 3: Upper-level fitness evaluation: For each upper-level individual evolved, their fitness is then evaluated. For this to happen, the lower-level optimal variables are needed. Therefore, the lower-level optimization procedure is activated.
- Step 4: Lower-level DE: for the fixed upper-level variables, a DE for lower-level is conducted, including its initialization, evolution, and final lower-level optimistic and pessimistic solutions.
- Step 5: Upper-level termination: After the lower level is conducted, the upper-level fitness is evaluated and then the termination criteria are checked. If they are satisfied, the procedure stops and a final solution is given. Otherwise, the upper-level population is

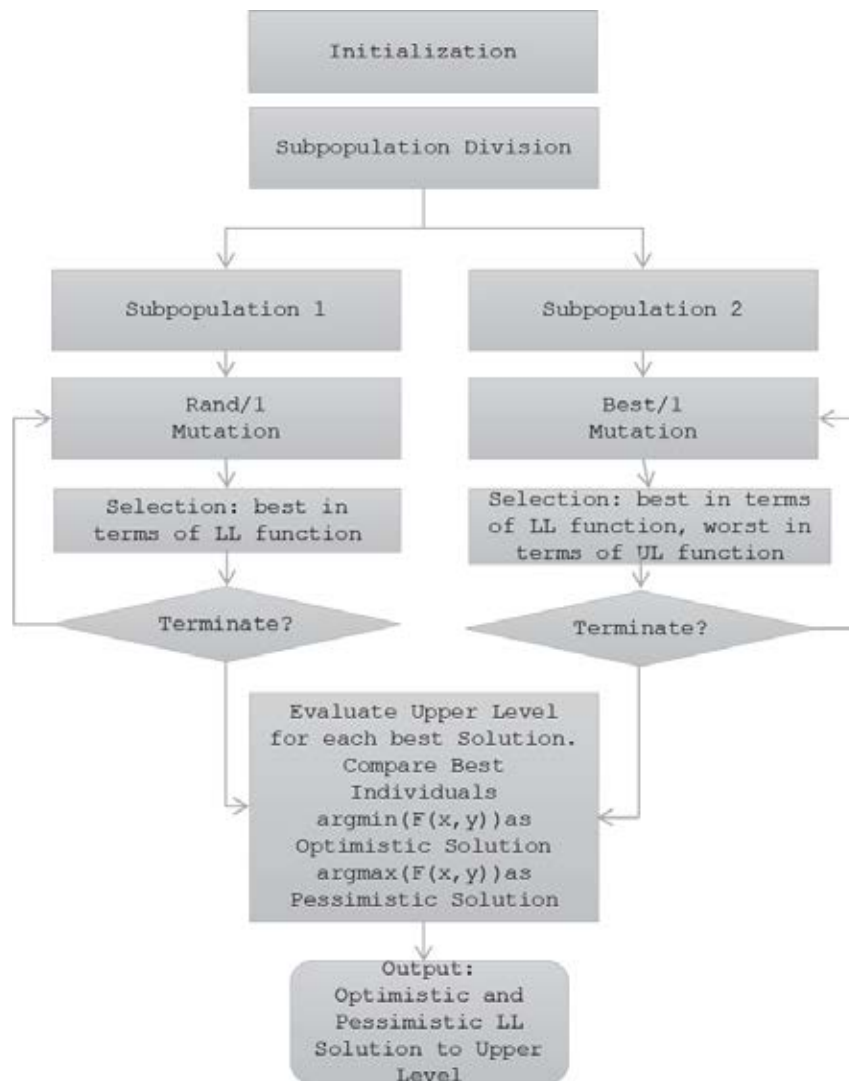


Figure 2: lower-level DE flowchart.

updated with the best solution found so far and we go to step 2 again. The best solution found for the pessimistic popupper levelation is further refined by local search (Interior Point) using the `fmincon` of Matlab.

For the lower-level DE (Input:  $x$  vector from upper level):

- Step 1: Initialization: The algorithm starts by sampling in a lower-level feasible space with a population of NP size. The lower-level variables are then found by the lower-level optimization procedure (the lower-level DE).
- Step 2: Lower-level mutation and recombination (crossover): The optimistic lower-level individuals are evolved with DE/rand/1/bin, while the pessimistic population with DE/best/1/bin.
- Step 3: Lower-level fitness evaluation: For each lower-level individual evolved, their fitness is then evaluated in terms of both lower- and upper-level fitness function.
- Step 4: Lower-level termination: After the lower-level is conducted, the lower-level fitness is evaluated and then the termination criteria are checked. If they are satisfied, the procedure stops, and two final solutions are given. The minimum and maximum solutions are labeled as optimistic or pessimistic respectively and passed to the upper-level. Otherwise, the lower-level population is updated with the best solution found so far and we go to step 2 again. For the optimistic population, the best solution found so far is the one that is minimum for the lower-level function. For the pessimistic population, the best solution found so far is the one that is minimum for the lower-level function and maximum for the upper-level function.

The termination criteria are the maximum number of generations for both levels. The control parameter values used are shown in Table 1. The population size refers to each sub-population for both upper- and lower-level DE, while number of generations is for the whole procedure. The parameter values were selected empirically, and no parameter tuning was conducted.

Table 1: Selected control parameters that are used in all of the reported results.

	upper-level	lower-level
Population size	10	30
Number of Generations	5	10
Crossover	0.9	0.9
Mutation	uniformly(0.2,0.8)	uniformly(0.2,0.8)

## 4 EXPERIMENTAL RESULTS

In this Section, we provide results for one test-problem, to illustrate the ability of the algorithm to reach the optimistic and the pessimistic solutions of the problem. The test function used is called `mb_1_17`, has multiple global minima in the lower-level, and is taken from [11]. The optimistic problem has the following formulation:

$$\begin{aligned}
 & \min_{x,y \in Y(x)} x^2 - y \\
 & \text{s.t. } x \in [0, 1], \quad y \in [0, 3] \\
 & \text{where } Y(x) = \arg \min_{z \in [0,3]} [z - 1 - x/10]^2 - x/2 - 1/2]^2
 \end{aligned} \tag{5}$$



while pessimistic problem is as follows:

$$\begin{aligned}
 & \min_x \quad \max_{y \in Y(x)} x^2 - y \\
 & \text{s.t.} \quad x \in [0, 1] \\
 & \text{where} \quad Y(x) = \arg \min_{z \in [0, 3]} \{[z - 1 - x/10]^2 - x/2 - 1/2\}^2
 \end{aligned} \tag{6}$$

In Table 2 the known global optima for the optimistic and pessimistic approach of the problem are reported. We also mention the pessimistic solution of the optimistic approach, which is different than the global pessimistic solution.

Table 2: Known optima for the test function mb\_1\_1\_17.

Solutions	x	y	F
Optimistic	0.2106	1.799	-1.755
Optimistic-Pessimistic	0.2106	0.243	-0.1987
Pessimistic	0	0.2929	-0.2929

The test function has been independently run 30 times on an Intel(R) Core(TM) i7-7500 CPU @ 2.70GHz, 16 GB of RAM, and the Windows 10 operating system. The algorithm is implemented in Matlab R2018b without any parallelization. In Table 3, the optimal values for the optimistic and pessimistic solutions of the 30 runs of the P-BLDEA are reported. Also, we report the min, median, and max of each solution and the runtime of the run in seconds. We can see, that the solutions found for the optimistic approach, in general, reach the known optimal solution with an accuracy of 0.0002 and for the pessimistic with 0.02. The runtime is approximately 40 seconds per run. This relatively high number is justified if we take into account that we are solving two problems at the same time (the optimistic and the pessimistic) and given that the results are without any parallelization. Moreover, no parameter tuning or adjustment of the population size is done for this experiment, while any tuning might lead to a dramatic decrease in the runtime. In Figure 3 the upper- and lower-level 3D mesh and contour plots are presented. Green dots are the known optimistic and pessimistic optima, while the red stars show the median values of the optima found from P-BLDE. We can see that the algorithm approximates the known solutions, managing to overcome the local optima of the lower-level problem.

## 5 CONCLUSIONS AND FUTURE WORK

In this paper we proposed a nested EA to solve the pessimistic BOP. Firstly, we introduced the topic and the motivation behind our research. Then, we gave the definitions of both the optimistic and the pessimistic bilevel optimization problem. Next, the proposed method was explained. The algorithm has a multi-population DE in each level, where an optimistic and pessimistic population co-evolve. The accuracy of the pessimistic lower-level solution is boosted by a local search for every best solution found. The algorithm gives two final best solutions, one for the optimistic and one for the pessimistic approach. The algorithm is tested in benchmark test-functions with known global optima, where it was shown to approximate well both the optimistic and pessimistic known global optima.

Tuning of the parameters of the algorithm's values, such as the population size and mutation strategy to decrease the number of function evaluations and make the convergence faster is what

Table 3: Numerical results for P-BLDE for the 30 runs for test function mb\_1\_1.17.

	Optimistic			Pessimistic			Runtime
Run	F	x	y	F	x	y	(secs)
1	-1.7548	0.25883	1.8218	-0.23591	0.00064378	0.23591	41.5709
2	-1.7548	0.18484	1.7889	-0.28388	0.00064378	0.28388	40.7871
3	-1.7546	0.22197	1.8039	-0.28627	0.00064378	0.28627	39.6872
4	-1.7492	0.28308	1.8293	-0.27532	0.00054351	0.27532	38.7187
5	-1.7531	0.17097	1.7823	-0.23037	0.00064756	0.23037	39.1808
6	-1.7544	0.18277	1.7878	-0.29213	0.00064378	0.29213	38.2499
7	-1.7554	0.18815	1.7908	-0.2928	0.00064378	0.2928	42.3118
8	-1.7546	0.19378	1.7921	-0.29264	0.00064378	0.29264	37.9264
9	-1.7547	0.22099	1.8035	-0.25513	0.0006468	0.25513	42.5849
10	-1.7526	0.1622	1.7789	-0.28134	0.00053724	0.28134	41.1649
11	-1.7553	0.1681	1.7835	-0.27213	0.00064378	0.27213	41.6821
12	-1.7546	0.18886	1.7903	-0.26977	0.00055272	0.26977	40.2938
13	-1.7538	0.2367	1.8098	-0.27128	0.00054427	0.27128	37.604
14	-1.7548	0.22897	1.8073	-0.29355	0.00064378	0.29355	37.5585
15	-1.7548	0.21446	1.8008	-0.2276	0.00054055	0.2276	37.7102
16	-1.7552	0.2127	1.8004	-0.27017	0.00053614	0.27017	37.4811
17	-1.755	0.22051	1.8037	-0.29301	0.00064378	0.29301	38.0905
18	-1.7561	0.19796	1.7952	-0.28992	0.00054281	0.28992	40.0531
19	-1.755	0.1998	1.7949	-0.1681	0.00053696	0.1681	48.6323
20	-1.7556	0.23253	1.8097	-0.28668	0.00053577	0.28668	46.1897
21	-1.7543	0.22525	1.805	-0.27192	0.00064378	0.27192	43.55
22	-1.7544	0.18446	1.7884	-0.20746	0.00064347	0.20746	49.4864
23	-1.7545	0.19986	1.7944	-0.28558	0.00064378	0.28558	51.1232
24	-1.7545	0.19628	1.7931	-0.29185	0.00064378	0.29185	51.7637
25	-1.7552	0.23102	1.8085	-0.28905	0.00064378	0.28905	57.5057
26	-1.7583	0.19052	1.7946	-0.20317	0.00064378	0.20317	62.2715
27	-1.7542	0.22883	1.8066	-0.23595	0.00053797	0.23595	51.1604
28	-1.7561	0.20439	1.7979	-0.2693	0.00054018	0.2693	49.5794
29	-1.7548	0.20328	1.7962	-0.26064	0.00054826	0.26064	50.1431
30	-1.7552	0.1851	1.7894	-0.29031	0.00053981	0.29031	53.1746
Min	-1.7492	0.1622	1.7789	-0.29355	0.00053577	0.1681	37.481
Median	-1.7548	0.20157	1.7957	-0.27372	0.00064378	0.27372	41.626
Max	-1.7583	0.28308	1.8293	-0.1681	0.00064756	0.29355	62.272

we are working on at the moment. The immediate next step of this research is to take advantage of the high parallelization of EAs, which will dramatically decrease the computational cost and reach higher accuracy. Also, methods known to work well for reducing the computational cost in the optimistic approach state-of-the-art methods, such as the use of surrogates and several mappings between the upper- and the lower-level solutions, are under investigation. Moreover, the strategy can work independently of the EA used, meaning that other evolutionary algorithms more suitable for multimodal problems in the lower-level can be tested, and/or known BLEAs from the literature can be implemented to find the optimistic solution.



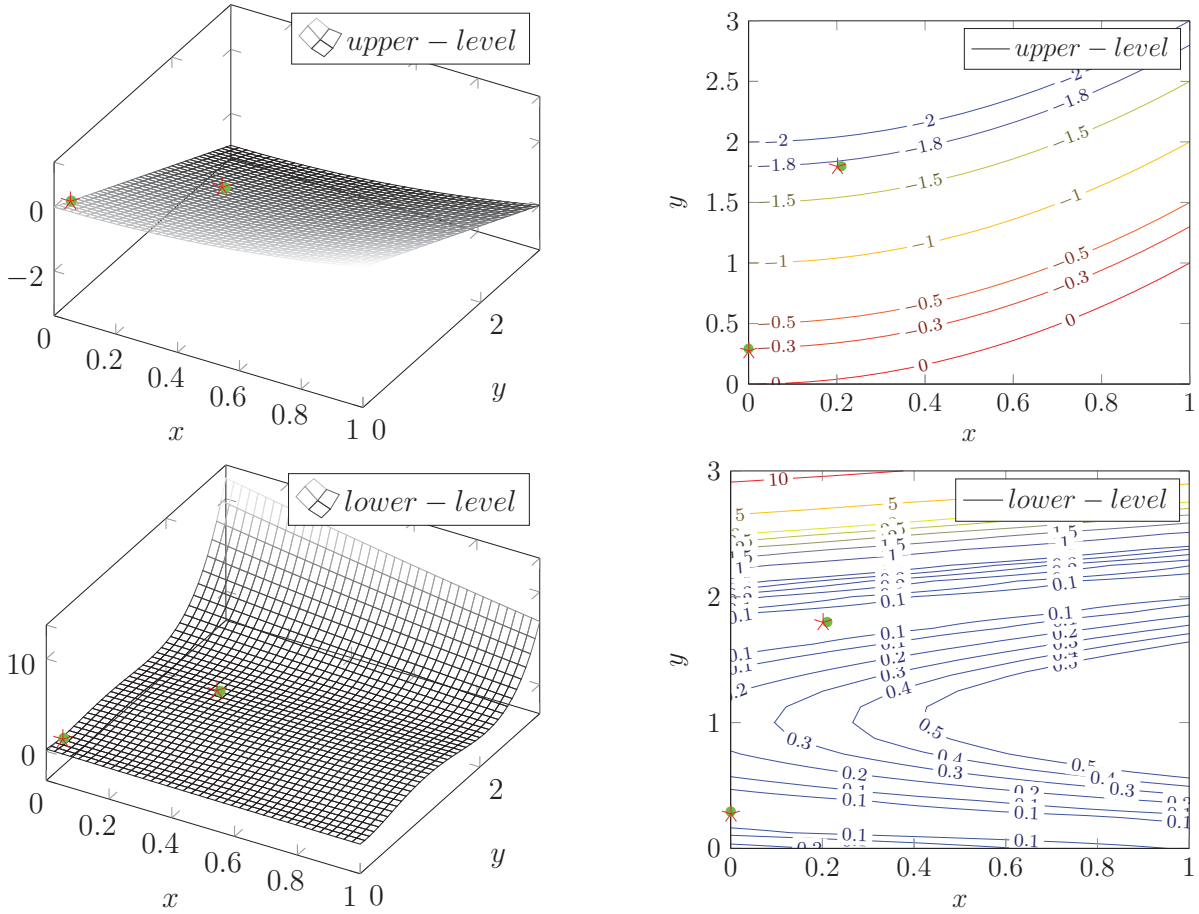


Figure 3: Upper- and lower-level 3D mesh and contour plots. Green dots are the known optimistic and pessimistic optima, while with red stars the median values of the optima found from P-BLDE.

### Acknowledgements

This work was supported by the Slovenian Research Agency (research core funding No. P2-0098), by the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 692286 (SYNERGY), and under the Marie Skłodowska-Curie grant agreement No. 722734 (UTOPIAE).

### REFERENCES

- [1] P. Hansen, B. Jaumard, and G. Savard, “New branch-and-bound rules for linear bilevel programming,” *SIAM Journal on scientific and Statistical Computing*, vol. 13, no. 5, pp. 1194–1217, 1992.
- [2] H. von Stackelberg, “Grundlagen der theoretischen volkswirtschaftslehre (translated as: The theory of the market economy),” *W. Hodge & Co. Ltd., London*, 1952.
- [3] A. Sinha, P. Malo, and K. Deb, “A review on bilevel optimization: from classical to evolutionary approaches and applications,” *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 276–295, 2017.

- [4] J. S. Angelo, E. Krempser, and H. J. Barbosa, “Differential evolution for bilevel programming,” in *2013 IEEE Congress on Evolutionary Computation*. IEEE, 2013, pp. 470–477.
- [5] A. Sinha, P. Malo, and K. Deb, “Test problem construction for single-objective bilevel optimization,” *Evolutionary computation*, vol. 22, no. 3, pp. 439–477, 2014.
- [6] M. M. Islam, H. K. Singh, T. Ray, and A. Sinha, “An enhanced memetic algorithm for single-objective bilevel optimization problems,” *Evolutionary computation*, vol. 25, no. 4, pp. 607–642, 2017.
- [7] A. Sinha, P. Malo, and K. Deb, “Evolutionary algorithm for bilevel optimization using approximations of the lower level optimal solution mapping,” *European Journal of Operational Research*, vol. 257, no. 2, pp. 395–411, 2017.
- [8] X. He, Y. Zhou, and Z. Chen, “Evolutionary bilevel optimization based on covariance matrix adaptation,” *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 258–272, 2018.
- [9] J. Liu, Y. Fan, Z. Chen, and Y. Zheng, “Pessimistic bilevel optimization: a survey,” *International Journal of Computational Intelligence Systems*, vol. 11, no. 1, pp. 725–736, 2018.
- [10] S. Dempe, B. S. Mordukhovich, and A. B. Zemkoho, “Necessary optimality conditions in pessimistic bilevel programming,” *Optimization*, vol. 63, no. 4, pp. 505–533, 2014.
- [11] W. Wiesemann, A. Tsoukalas, P.-M. Kleniati, and B. Rustem, “Pessimistic bilevel optimization,” *SIAM Journal on Optimization*, vol. 23, no. 1, pp. 353–380, 2013.
- [12] S. Dempe, G. Luo, and S. Franke, *Pessimistic bilevel linear optimization*. Technische Universität Bergakademie Freiberg, Fakultät für Mathematik und . . . , 2016.
- [13] A. V. Malyshev and A. S. Strekalovsky, “Global search for pessimistic solution in bilevel problems,” in *Proceedings of the Toulouse global optimization workshop*, 2010, pp. 77–80.
- [14] M. J. Alves, C. H. Antunes, and P. Carrasqueira, “A pso approach to semivectorial bilevel programming: pessimistic, optimistic and deceiving solutions,” in *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, 2015, pp. 599–606.
- [15] M. J. Alves and C. H. Antunes, “A differential evolution algorithm to semivectorial bilevel problems,” in *International Workshop on Machine Learning, Optimization, and Big Data*. Springer, 2017, pp. 172–185.
- [16] J. Liu, Y. Fan, Z. Chen, and Y. Zheng, “Methods for pessimistic bilevel optimization,” in *Bilevel Optimization*. Springer, 2020, pp. 403–420.