A Physics-Inspired Algorithm for Bilevel Optimization

Jesús-Adolfo Mejía-de-Dios
Artificial Intelligence Research Center
University of Veracruz
Sebastián Camacho 5, Centro
Xalapa, Veracruz, 91000, México
Email: jesusmejded@gmail.com

Efrén Mezura-Montes
Artificial Intelligence Research Center
University of Veracruz
Sebastián Camacho 5, Centro
Xalapa, Veracruz, 91000, México
Email: emezura@uv.mx

***Abstract*—This paper presents the application of a physics-inspired algorithm based on the center of mass concept, called Bilevel Centers Algorithm (BCA), to deal with bilevel optimization problems. The center of mass is adopted for creating new directions in the bilevel continuous search space considering the objective function values of a set of randomly-chosen solutions in a hierarchical optimization structure. The performance of this approach is assessed by using representative test functions for bilevel optimization. The obtained results are compared against the state-of-the algorithm BLEAQ. The results based on accuracy and number of evaluations are competitive and promising.**

I. Introduction

A new kind of optimization problem has been gaining interest by researchers in recent years. That problem was introduced in 1934 by Von Stackelberg in [1] and it is known nowadays as the Bilevel Optimization (BO) problem. A BO problem can be unconstrained, constrained, single and/or multi-objective, continuous and/or discrete, etc. but in any type it contains a nested optimization problem as a constraint [2], [3]. Such introduced hierarchical structure can be useful to model decision-making processes, where a leader(upper level authority) optimizes their goals restricted to optimal decisions/solutions given by a follower (lower level authority) [4], [5], [6], [7], [8].

In order to formally introduce BO problems, we start by stating the traditional optimization problem. It is well known that, without loss of generality, an optimization problem can be defined as finding the set:

$$\begin{aligned} X^* &= \arg\min_{\vec{x}\in X} f(\vec{x}) \\ &= \{\vec{x}^* \in X \ : \ f(\vec{x}^*) \leq f(\vec{x}),\ \forall \vec{x} \in X\}, \end{aligned} \tag{1}$$

where a bounded below function f , i.e., $f(x^*) > -\infty$ is called objective function. X is a D-dimensional parameter space, usually $X \subset \mathbb{R}^D$ is the domain for $\vec{x}$ representing constraints on allowable values for $\vec{x}$. Eq. 1 may be read as: X^* is the set of values (arguments) $\vec{x} = \vec{x}^*$ that minimizes $f(\vec{x})$ subject to X^* (see Figure 1).

An optimization problem is solved only when a global minimum is found. However, global minimum are, in general, difficult to find. Therefore, in practice, we often have to find at least a local minimum [9], [10].

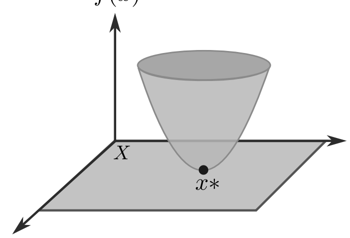

Fig. 1. Single-objective optimization problem.

After the above definitions, a general single-objective BO problem with single-objective functions at both levels can be defined as follows [2], [3]:
Minimize

$$F(\vec{x},\ \vec{y}) \quad \vec{x} \in X,\ \vec{y} \in Y \tag{2}$$

subject to:

$$\vec{y} \in \arg\min\{f(\vec{x},\vec{y}) \ : \ g_j(\vec{x},\vec{y}) \leq 0,\ j = 1,\dots,J\} \tag{3}$$

$$G_k(\vec{x},\vec{y}) \leq 0,\ k = 1,\dots,K \tag{4}$$

where $F : X \times Y \to \mathbb{R}$ and $f : X \times Y \to \mathbb{R}$ are the upper-level objective function (leader) and lower-level objective function (follower), respectively. In this work, $X \subseteq \mathbb{R}^n$ and $Y \subseteq \mathbb{R}^m$ are considered. Here, G and g are the inequality constraints of the upper and lower levels, respectively. Figure 2 shows a schematic diagram of a BO problem.

In 1992, Hansen et al. probed that BO problems are (strongly) NP-hard because evaluating a solution in the simplest BO problem (unimodal linear programming at both levels) is also NP-hard [11], [12]. Moreover, many real-world problems can be naturally modeled as BO problems [13], for example: taxation, border security problems, transportation problems, machine learning algorithms tuning, among others [2], [13], [14].

Due to the importance of BO, many authors have proposed different kind of solutions for those problems e.g. mathematical approaches (mathematical programming, Karush-Kuhn-Tucker condition for single-level reduction) [3], [15], evolutionary computation (genetic algorithms, evolution strategies)

978-1-5386-5935-9/18/$31.00

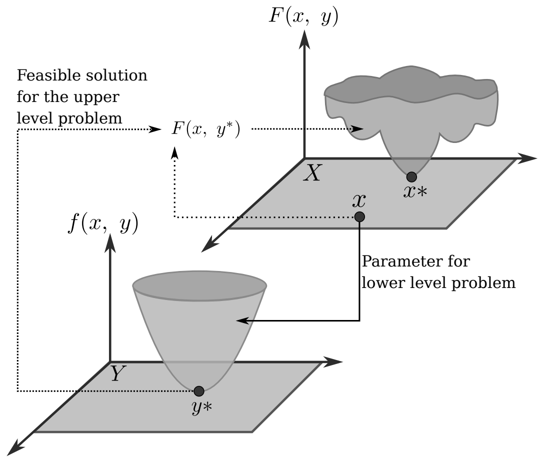

Fig. 2. Diagram of a bilevel optimization problem. Here, y^* is defined as in Eq. (3). Note that $(x,\ y^*)$ is a feasible solution.

and swarm intelligence (particle swarm optimization) [16], [17], [18].

In order to handle BO problems three strategies are identified: (1) nested, (2) single-level reduction and (3) penalty methods. Nested strategies can be effective but with a high computational cost when objective functions are expensive to calculate or when high-dimensionality is present. Single-level reduction is used to transform a BO problem into a single-level problem, usually via Karush-Kuhn-Tucker conditions for smooth lower level objective function. As a consequence, this strategy is restricted to differentiable functions [3], [19]. Penalty methods transform a constrained method into an unconstrained optimization problem by adding some penalties controlled by using parameters. This strategy can be simple to understand but hard to calibrate particularly in large-scale problems [20], [21].

From the literature review above mentioned, regarding population-based metaheuristics for BO problems, most research efforts are focused on evolutionary computation and swarm intelligence algorithms because they have been successfully applied to solve single-level optimization problems. On the other hand, in recent years, physics-inspired algorithms have become popular to solve complex optimization problems and they have provided a competitive performance when solving single-level optimization problems [22].

This is the main motivation of this research work, where a first proposal of a physics-inspired algorithm, originally designed for global optimization [23], is now presented to deal with BO problems. Here, an unconstrained nested scheme is considered.

Our approach is called Bilevel Centers Algorithm (BCA) and it is based on the center of mass concept [23]. Without loss of generality, we assume a maximization problem for non-negative functions f and F for lower and upper objective functions, respectively. The center of mass concept is used in both levels, in order to generate new biased directions towards promising/feasible regions of the search space.

This paper is organized as follows: Section II presents the lower level optimizer which uses the center of mass concept to promote a biased search to promising search space regions. The proposed approach is detailed in Section III. After that, Section IV shows the experimental design and the corresponding results and discussions about the performance of the proposed approach, where a state-of-the-art for BO optimization is used for comparison purposes. The conclusions and future work of this research are described in Section V and Section VI, respectively.

II. Lower Level Optimizer

The center of mass is the unique point $\vec{c}$ at the center of a mass distribution $U = \{\vec{u}_1,\ \vec{u}_2, \ldots, \vec{u}_K\}$ in a space that has the property that the weighted sum of K position vectors relative to this point is zero [23], as indicated in Eq. 5:

$$\sum_{i=1}^{K} m(\vec{u}_i)(\vec{u}_i - \vec{c}) = 0, \text{ implies } \vec{c} = \frac{1}{M}\sum_{i=1}^{K} m(\vec{u}_i)\vec{u}_i, \quad (5)$$

where $m(\vec{u}_i)$ is the mass of $\vec{u}_i$ and M is the sum of the masses of vectors in U. Here, m is a non-negative function.

Thus, the lower level optimizer works as follows: for each solution $\vec{y}_i$ in the population $P = \{\vec{y}_1,\ \vec{y}_2,\ \ldots,\ \vec{y}_N\}$ of N solutions in Y, we generate $U_i \subset P$ with K different solutions such that

$$\bigcup_{i=1}^{N} U_i = P.$$

Next, from U_i the center of mass $\vec{c}_i$ is computed by using Eq. 5 considering $m(\vec{u}) = f(\vec{p}, \vec{u})$, where $\vec{p}$ is given by the upper level optimizer. After that, the worst element $\vec{u}_{\text{worst}}$ in U_i is selected according to the following rule:

$$\vec{u}_{\text{worst}} \in \arg\min\{f(\vec{p},\ \vec{u})\ :\ \vec{u} \in U_i\}.$$

Now, we are able to generate a direction to locate a new solution $\vec{q}_i$ using the already generated center of mass $\vec{c}_i$, the current solution $\vec{y}_i$ and the worst solution $\vec{u}_{\text{worst}}$ by using Eq. 6:

$$\vec{q}_i = \vec{y}_i + \eta_i(\vec{c}_i - \vec{u}_{\text{worst}}), \quad (6)$$

Due to this stochastic strategy, this variation operator can help the exploration-exploitation process to avoid premature convergence because it combines the center of mass as an attractor to promising regions of the search space but using the position of the worst solution as a reference to get far away from it [23]. The replacement operator works as follows: if $\vec{q}_i$ is better than $\vec{y}_i$, then the worst element in P is replaced by $\vec{q}_i$.

A linear reduction of the population size (deleting the worst elements) is applied. The initial population size is $N(0) = K * D$ and the final population size $N(T) = 2 * K$ (for successfully generating the center of mass). Thus, the population size over time is as in Eq. 7:

$$N(t) = KD - \frac{(KD - 2K)t}{T} = K\left(D - \frac{(D-2)t}{T}\right), \quad (7)$$

where $t = 0, 1, 2, \ldots, T$ and T is the maximum number of iterations.

The procedure for the implementation of the lower level optimizer is detailed in Algorithm 1.

Algorithm 1 Lower Level Optimizer pseudocode

Input: upper level parameter $\vec{p}$, $K = 7$, $\eta_{\max} = 2$

$N \leftarrow K * D$
Initialize a population $P \subset Y$ with N elements
while the end criterion is not achieved **do**
  for each $\vec{y}$ in P **do**
    Generate a subset $U \subset P$ with K solutions
    Calculate $\vec{c}$ using U with (5)
    $\eta \leftarrow \text{rand}(0,\ \eta_{\max})$
    Compute $\vec{q}$ using Eq. (6)
    if $f(\vec{p},\ \vec{y}) < f(\vec{p},\ \vec{q})$ **then**
      Replace worst element in P with $\vec{q}$
    end if
  end for
  Resize P with Eq. 7
end while
return the best solution in P

III. BCA

Here, BCA is presented as the upper lower level optimizer. We start describing the variation operator of BCA and its properties. Let F and f be defined as in Eq. (2) and Eq. (3), respectively. Without loss of generality we can assume that both F and f are non-negative and we want to maximize them. Hence, the population for a BO problem can be defined as in Eq. 8:

$$P = \{(\vec{x}_1,\ \vec{y}_1),\ (\vec{x}_2,\ \vec{y}_2), \ldots, (\vec{x}_N,\ \vec{y}_N)\} \subset X \times Y, \tag{8}$$

where $\vec{y}_i \in \arg\min\{f(\vec{x}_i,\ \vec{z}) \ :\ \vec{z} \in Y\}$ for $i = 1, \ldots, N$. Here, $\vec{x}_i$ is generated at random with uniform distribution.

A. Algorithm Description

Now, we are able to describe the BCA procedure: for each iteration and each solution $(\vec{x}_i,\ \vec{y}_i) \in P$, a new upper level parameter is generated using the formulation in Eq. 9:

$$\vec{p}_i = \vec{x}_i + \eta_i(\vec{c}_i - \vec{u}_{\text{worst}}), \tag{9}$$

where the $\vec{c}_i$ is the center of mass computed using:

$$\vec{c}_i = \frac{1}{W} \sum_{(x,y)\in U} Q(\vec{x},\ \vec{y}) \cdot \vec{x}, \qquad W = \sum_{(x,y)\in U} Q(\vec{x},\ \vec{y}), \tag{10}$$

where $Q(\vec{x},\ \vec{y}) = F(\vec{x},\ \vec{y}) + f(\vec{x},\ \vec{y})$, $U \subset P$ such that $\text{card}(U) = K$ and $\vec{u}_{\text{worst}}$ is the first coordinate of the worst element in U, see Eq. 11.

$$\vec{u}_{\text{worst}} \in \arg\min\{Q(\vec{u},\ \vec{y}) \ :\ \vec{u} \in U_i\} \tag{11}$$

Note that the function $Q(\vec{x},\ \vec{y})$ is used to translate the upper level population towards regions where F is maximized while f may control the bias to feasible solutions.

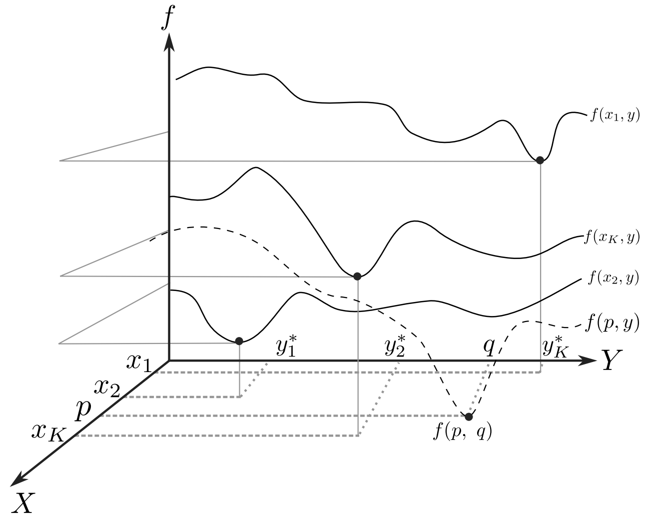

Fig. 3. Diagram of BCA. Here, $x_1, \ldots, x_K$ are used to compute better upper level parameters $\vec{p}$. Note that $(x_i,\ y_i^*)$ and $(\vec{p},\ \vec{q})$ represent feasible solutions.

Finally, the new solution is given by $(\vec{p},\ \vec{q})$ which may replace the worst solution in P if it is better than $(\vec{x}_i,\ \vec{y}_i)$. Here, $\vec{q} = \arg\min_{\vec{z}\in Y} f(\vec{p},\ \vec{z})$ obtained by applying Algorithm 1. Figure 3 shows a representation of BCA solution update.

The BCA approach requires the definition of three parameter values: K, $\eta_{\max}$ and the population size. Here, K is useful to handle the multi-modality, i.e., large K values provide fast convergence to local optima (useful for unimodal functions). When K takes small values, BCA favors an exploratory process. $\eta_{\max}$ mainly controls the exploratory process as the stepsize is controlled by this parameter. The parameter setting recommended by preliminary experiments is $K = 7$ and $\eta_{\max} = 2$.

BCA is summarized in Algorithm 2. This proposed algorithm was used to solve a set of eight test problems to assess the performance of BO algorithms, known as SMD problems. Those test problems are described in [17], [24], [25].

IV. Experiments and Discussion

The configuration used in BCA for the experiments performed was as follows: The number of Function Evaluations (NFEs) was fixed for the upper level: $500D_{UL} = 2500$, and the lower level: $500D_{UL} * 500D_{LL} = 6,250,000$ total NFEs for both levels. The remaining parameters were set as follows:

- Upper level dimension $D_{UL} = 5$
- Lower level dimension $D_{LL} = 5$
- Upper level population size $K * D_{UL}$
- Lower level population size $K * D_{LL}$
- $K = 7$
- $\eta_{\max} = 2$
- Stop condition: BCA stopped when the accuracy (1×10^{-4}) or the maximum NFEs allowed was reached.

BCA was compared against BLEAQ which is a state-of-the-art evolutionary algorithm for BO problems [13], [26]. BLEAQ is based on quadratic approximations of optimal lower

Algorithm 2 BCA pseudocode

$N \leftarrow K * D$
Generate and evaluate start population P with N elements
while the end criterion is not achieved **do**
  for each $(\vec{x},\ \vec{y})$ in P **do**
    Generate a subset $U \subset P$ such that $\text{card}(U) = K$
    Calculate $\vec{c}$ using U with Eq. (10)
    $\eta \leftarrow \text{rand}(0,\ \eta_{\max})$
    Calculate $\vec{p}$ using Eq. (9)
    Find $\vec{q} \in \arg\min_{\vec{z} \in Y} f(\vec{p},\ \vec{z})$ by using Algorithm 1
    if $G(\vec{x},\ \vec{y}) < G(\vec{p},\ \vec{q})$ **then**
      Replace worst element in P with $(\vec{p},\ \vec{q})$.
    end if
  end for
  Resize P with Eq. 7
end while
Report best solution in P

level variables with respect to the upper level variables. Such strategy lets BLEAQ to reduce the NFEs having competitive results. BLEAQ was run with the parameters suggested by the authors [13], [26] and the stop criteria was maintained but accuracy was set at 1×10^{-4} as in BCA. Both algorithms were used to solve independently 31 times each test problem.

TABLE I
UPPER LEVEL ACCURACY STATISTICS BY BCA OBTAINED FROM 31 INDEPENDENT RUNS.

	Best	**Median**	**Mean**	**Worst**	**Std.**
SMD1	1.51E–05	5.35E–05	5.28E–05	8.34E–05	1.90E–05
SMD2	1.50E–05	4.68E–05	5.88E–05	3.11E–04	5.13E–05
SMD3	1.57E–05	5.51E–05	2.40E–04	3.54E–03	6.60E–04
SMD4	4.09E–08	4.90E–05	6.20E–05	3.01E–04	6.70E–05
SMD5	2.30E–05	5.03E–05	4.78E–05	8.33E–05	1.74E–05
SMD6	1.57E–01	1.90E+01	2.59E+01	1.40E+02	2.95E+01
SMD7	9.34E–01	9.75E–01	1.19E+00	3.81E+00	6.00E–01
SMD8	1.58E–05	6.49E–05	1.83E–04	2.23E–03	4.11E–04

TABLE II
LOWER LEVEL ACCURACY STATISTICS BY BCA OBTAINED FROM 31 INDEPENDENT RUNS.

	Best	**Median**	**Mean**	**Worst**	**Std.**
SMD1	2.67E–06	2.06E–05	2.14E–05	4.75E–05	9.77E–06
SMD2	3.56E–06	1.81E–05	2.07E–05	4.60E–05	1.15E–05
SMD3	1.87E–07	2.24E–05	3.12E–04	4.31E–03	9.28E–04
SMD4	6.23E–06	3.65E–05	1.53E–04	7.96E–04	2.33E–04
SMD5	2.50E–06	2.01E–05	2.19E–05	4.55E–05	1.28E–05
SMD6	3.91E–04	2.86E–02	4.23E–02	1.84E–01	4.35E–02
SMD7	3.71E+02	3.74E+02	3.74E+02	3.75E+02	1.01E+00
SMD8	1.18E–06	2.33E–05	6.53E–05	7.93E–04	1.44E–04

Tables I and II show the statistical results of the accuracy values obtained by the proposed BCA in the two levels of each

TABLE III
UPPER LEVEL NFES STATISTICS BY BCA OBTAINED FROM 31 INDEPENDENT RUNS.

	Best	**Median**	**Mean**	**Worst**	**Std.**
SMD1	1244	1526	1539.42	1879	152.119
SMD2	1244	1481	1482.84	2501	220.7
SMD3	1365	1526	1647.84	2501	287.517
SMD4	1169	1435	1437.26	1800	131.393
SMD5	1317	1526	1554.42	1898	137.766
SMD6	2501	2501	2501	2501	0
SMD7	2501	2501	2501	2501	0
SMD8	1780	2219	2234.65	2501	206.959

TABLE IV
LOWER LEVEL NFES STATISTICS BY BCA OBTAINED FROM 31 INDEPENDENT RUNS.

	Best	**Median**	**Mean**	**Worst**	**Std.**
SMD1	3110000	3815000	3848548.4	4697500	380298.6
SMD2	3110000	3702500	3707096.8	6252500	551750.7
SMD3	3412500	3815000	4119596.8	6252500	718793.3
SMD4	2922500	3587500	3593145.2	4500000	328481.3
SMD5	3292500	3815000	3886048.4	4745000	344415.4
SMD6	6252500	6252500	6252500	6252500	0
SMD7	6252500	6252500	6252500	6252500	0
SMD8	4450000	5547500	5586612.9	6252500	517397.6

TABLE V
MEDIAN NFES VALUES BY BCA AND BLEAQ OBTAINED FROM 31 INDEPENDENT RUNS.

	Upper Level		Lower Level	
	BCA	BLEAQ	BCA	BLEAQ
SMD1	**1526**	1600	3815000	**116088**
SMD2	**1481**	1925	3702500	**113504**
SMD3	**1526**	1630	3815000	**122542**
SMD4	**1435**	1750	3587500	**70906**
SMD5	**1526**	3031	3815000	**147289**
SMD6	2501	**1016**	6252500	**7055**
SMD7	2501	**2104**	6252500	**130195**
SMD8	**2219**	5569	5547500	**289886**

BO test problem. Moreover, the best, median, mean, worst and standard deviation values of the NFEs required to solve each BO test problem using BCA are given in Table III and IV for each level, respectively. Tables V and VI compare the median accuracy and NFEs at the upper and lower levels required by BCA against those values obtained by BLEAQ. A result in boldface indicates the best value found.

From Tables I and II it can be observed that BCA is able to consistently reach very competitive results in all test problems for both levels. Just problem SMD7 was difficult to solve by BCA.

A similar robust performance was observed by BCA in the upper level regarding NFEs as indicated in Table III. In contrast, more variation in the number of NFEs was reported

TABLE VI
MEDIAN ACCURACY VALUES BY BCA AND BLEAQ OBTAINED FROM 31 INDEPENDENT RUNS.

	Upper Level		Lower Level	
	BCA	BLEAQ	BCA	BLEAQ
SMD1	**5.35E–05**	9.91E–05	**2.06E–05**	6.72E–05
SMD2	**4.68E–05**	2.82E–04	**1.81E–05**	3.84E–04
SMD3	5.51E–05	**4.96E–06**	2.24E–05	**6.26E–06**
SMD4	**4.90E–05**	1.54E–04	**3.65E–05**	6.12E–04
SMD5	**5.03E–05**	1.62E–04	**2.01E–05**	3.08E–04
SMD6	1.90E+01	**1.46E–13**	2.86E–02	**8.66E–16**
SMD7	9.75E–01	**9.76E–02**	3.74E+02	**1.25E+02**
SMD8	**6.49E–05**	7.46E–03	**2.33E–05**	5.63E–03

in Table IV for the lower level.

Regarding the comparison against the state-of-the-art algorithm, from Table VI it can be concluded that BCA outperformed BLEAQ in five BO test problems in the upper level and also in five BO test problems in the lower level. Furthermore, BCA required less NFEs in the upper level of six BO test problems based on Table V. However, BLEAQ clearly outperformed BCA in the number of lower level NFEs as indicated in the same Table V.

Note that no statistical test was used to compare the algorithms, since the obtained values (from the objective functions) may not represent feasible solutions and that can be an unfair comparison of the performance of the algorithms.

V. CONCLUSIONS

This work presented the adaptation of a physics-inspired algorithm based on the center of mass (BCA) to solve bilevel optimization problems. Both levels used a similar variation operator based on the center of mass to find promising regions of the search space and a greedy replacement based on fitness. BCA is a simple algorithm which requires just three parameters to be fine-tuned by the user (the population size, the size of the subset to compute the center of mass and the stepsize for the variation operator). Eight test problems were solved to assess the performance of the proposed algorithm in terms of upper/lower level accuracy and function evaluations compared against a state-of-the-art evolutionary algorithm for BO. The overall results suggest that BCA was able to provide competitive results in terms of accuracy compared to those obtained by BLEAQ even requiring less evaluations in the upper level. However BLEAQ outperformed BCA with respect to the number of evaluations required at the lower level. Additional resources (code, tutorials, etc.) about bilevel optimization can be found at https://bi-level.org.

VI. FUTURE WORK

The future paths of research are the following: carry out a study in order to approximate feasible solutions at the lower level to decrease number of evaluations required and explore some penalty methods to transform the nested scheme into a single level optimization problem. Study or propose a technique to compare the algorithm performance for bilevel optimization problems.

ACKNOWLEDGMENTS

The first author acknowledges support from the Mexican Council for Science and Technology (CONACyT) for a scholarship to pursue graduate studies at the University of Veracruz. The second author acknowledges support from CONACyT through project No. 220522.

REFERENCES

[1] H. Von-Stackelberg, *Market Structure and Equilibrium*. Springer Science & Business Media, 2010.

[2] J. F. Bard, *Practical bilevel optimization: algorithms and applications*. Springer Science & Business Media, 2013, vol. 30.

[3] S. Dempe, *Foundations of Bilevel Programming*. Springer Science & Business Media, 2002.

[4] L. Brotcorne, M. Labbé, P. Marcotte, and G. Savard, "A bilevel model for toll optimization on a multicommodity transportation network," *Transportation Science*, vol. 35, no. 4, pp. 345–358, 2001.

[5] V. Kalashnikov, F. Camacho, R. Askin, and N. Kalashnykova, "Comparison of algorithms for solving a bi-level toll setting problem," *International Journal of Innovative Computing, Information and Control*, vol. 6, no. 8, pp. 3529–3549, 2010.

[6] A. Sinha, P. Malo, and K. Deb, "Transportation policy formulation as a multi-objective bilevel optimization problem," in *Evolutionary Computation (CEC), 2015 IEEE Congress on*. IEEE, 2015, pp. 1651–1658.

[7] J. Von-Neumann and O. Morgenstern, "Theory of games and economic behavior," *Bull. Amer. Math. Soc*, vol. 51, no. 7, pp. 498–504, 1945.

[8] J. Y. T. Wang, M. Ehrgott, K. N. Dirks, and A. Gupta, "A bilevel multi-objective road pricing model for economic, environmental and health sustainability," *Transportation Research Procedia*, vol. 3, pp. 393–402, 2014.

[9] S. S. Rao and S. S. Rao, *Engineering optimization: theory and practice*. John Wiley & Sons, 2009.

[10] T. Bck, *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, 1996.

[11] 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.

[12] L. Vicente, G. Savard, and J. Júdice, "Descent approaches for quadratic bilevel programming," *Journal of Optimization Theory and Applications*, vol. 81, no. 2, pp. 379–399, 1994.

[13] 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, 2018.

[14] J. M. Arroyo, "Bilevel programming applied to power system vulnerability analysis under multiple contingencies," *IET generation, transmission & distribution*, vol. 4, no. 2, pp. 178–190, 2010.

[15] C. Shi, J. Lu, and G. Zhang, "An extended kuhn–tucker approach for linear bilevel programming," *Applied Mathematics and Computation*, vol. 162, no. 1, pp. 51–63, 2005.

[16] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.

[17] J. S. Angelo, E. Krempser, and H. J. C. Barbosa, "Differential evolution for bilevel programming," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*. IEEE, 2013, pp. 470–477.

[18] X. Li, P. Tian, and X. Min, "A hierarchical particle swarm optimization for solving bilevel programming problems," in *International Conference on Artificial Intelligence and Soft Computing*. Springer, 2006, pp. 1169–1178.

[19] B. Colson, P. Marcotte, and G. Savard, "An overview of bilevel optimization," *Annals of operations research*, vol. 153, no. 1, pp. 235–256, 2007.

[20] G. Savard and J. Gauvin, "The steepest descent direction for the nonlinear bilevel programming problem," *Operations Research Letters*, vol. 15, no. 5, pp. 265–272, 1994.

[21] D. J. White and G. Anandalingam, "A penalty function approach for solving bi-level linear programs," *Journal of Global Optimization*, vol. 3, no. 4, pp. 397–419, 1993.

[22] A. Biswas, K. Mishra, S. Tiwari, and A. Misra, "Physics-inspired optimization algorithms: A survey," *Journal of Optimization*, vol. vol. 2013, 2013.

[23] J. Mejía-de-Dios and E. Mezura-Montes, "A new evolutionary optimization method based on center of mass," in *Decision Science in Action, Asset Analytics*. Springer (in press), 2018, p. 11.

[24] A. Sinha, P. Malo, and K. Deb, "Unconstrained scalable test problems for single-objective bilevel optimization," in *Evolutionary Computation (CEC), 2012 IEEE Congress on*. IEEE, 2012, pp. 1–8.

[25] ——, "Test problem construction for single-objective bilevel optimization," *Evolutionary computation*, vol. 22, no. 3, pp. 439–477, 2014.

[26] ——, "Efficient evolutionary algorithm for single-objective bilevel optimization," *arXiv preprint arXiv:1303.3901*, 2013.