XLVI CILAMCE
VITÓRIA - ES - BRAZIL - 2025
XLVI IBERO-LATIN AMERICAN CONGRESS ON COMPUTATIONAL METHODS IN ENGENEERING

Benchmark Instances for the Bilevel Optimization of the Toll Pricing Problem

Karen J. Oliveira[1], Jaqueline S. Angelo[2], Heder S. Bernardino[1]

[1]*Universidade Federal de Juiz de Fora (UFJF)*
karen.oliveira@estudante.ufjf.br, heder.bernardino@ufjf.br
[2]*Fundação Oswaldo Cruz (Fiocruz)*
jaqueline.angelo@fiocruz.br

Abstract. The Toll Pricing Problem (TPP) seeks to optimize tolls in a network by maximizing profit while minimizing the travel cost of users. Bilevel optimization emerges as a suitable way for modeling this problem, as it allows the regulatory authority to anticipate users' responses to the toll policy. Due to the complexity of the problem, metaheuristics are frequently employed. The development of new metaheuristics requires a rigorous investigation process, involving the testing of different search strategies and parameter settings. Moreover, comparative evaluations between methods are essential to validate progress and guide improvements in bilevel optimization techniques. We propose a benchmark suite comprising 192 instances for a bilevel toll optimization problem. In this setting, the upper-level objective is to maximize the toll manager's profit, while the lower level seeks to minimize the total travel cost for routing a set of commodities across a network. The benchmark includes instances ranging from 10 to 60 nodes and 5 to 40 commodities, with tolls applied to 20%-40% of edges in smaller instances (up to 15 nodes) and 5%-20% in larger instances (up to 60 nodes). A bilevel algorithm with multiple followers was used to generate reference results, which are provided for future benchmarking. The dataset includes the best-obtained toll values and their corresponding routing costs.

Keywords: Toll Optimization, Bilevel Optimization, Benchmark Instances, Differential Evolution

1 Introduction

Toll pricing in transportation networks has been increasingly adopted by governments, as public funding alone is often insufficient to cover the costs of highway construction, maintenance, and modernization [1]. Due to the high investment required in such projects, it is common for private companies to set high toll rates in an effort to recover their investment over a short period. However, network users tend to avoid high-cost toll roads by diverting to local routes, which can lead to negative externalities, such as increased emissions, congestion, accidents, and deterioration of infrastructure [2]. Therefore, the objective of the Toll Pricing Problem (TPP) is to determine toll rates that simultaneously maximize the profit of road authorities and minimize travel costs for network users.

This problem is often modeled as a Bilevel Optimization Problem (BLP), in which the upper level sets the toll prices, while the lower level determines the optimal routing of the users in response to those tolls. Given the complexity of this type of application, Computational Intelligence techniques have been widely investigated to support more accurate and efficient decision-making processes. In this context, benchmarking plays a crucial role as a catalyst for research in the field, as it establishes reference standards that guide and evaluate the pursuit of better solutions [3]. An effective way to promote the development of benchmarks is through competitions organized at major events such as the Congress on Evolutionary Computation (CEC) and the Genetic and Evolutionary Computation Conference (GECCO). These competitions define specific challenges and themes to stimulate new research directions and benchmarking creation.

Several authors have contributed to the development of algorithms for solving TPP [4–6]. Although some studies describe the characteristics of the instances used, there is a notable limitation in the variety and availability of these datasets; the instances are not public, making it almost impossible to reproduce experiments and perform fair comparisons. To address this gap and promote further research in this area, there is a clear need for publicly available and well-structured benchmark instances that reflect diverse and realistic scenarios. Motivated by this need, we propose a benchmark set of 192 instances for the bilevel TPP, covering networks ranging from 10 to 60 nodes and from 5 to 40 commodities. The instances were designed to capture a variety of practical configurations,

including different proportions of tolled arcs, ranging from 20% to 40% in smaller networks and from 5% to 20% in larger ones.

In addition to the benchmark contribution, we also propose a search strategy for this problem. We employ (i) a Bilevel Differential Evolution approach with Multiple Independent Followers (BLDE-MIF) [7, 8], and (ii) a permutation-based DE variant with random keys [9] to solve the lower-level problem of the bilevel formulation. Moreover, we provide reference results generated by the proposed method, named here as BLDE-MIF-RK, including the best toll values obtained and their corresponding routing costs. These data are intended to facilitate the validation of new approaches and to promote advances in bilevel optimization applied to transportation and logistics problems.

2 Bilevel toll optimization problem

A BLP consists of two hierarchically related optimization tasks. The leader, at the upper level, aims to optimize an objective function while considering its constraints and the reaction of the follower at the lower level. In turn, the follower seeks to optimize its own objective function and constraints, taking the leader's decisions as given. Compared to single-level optimization, the inherently nested structure of BLPs makes them significantly more complex and computationally demanding [10].

For the toll pricing optimization problem [4], consider a multi-commodity network $G = \{K, N, C\}$ defined by a set of commodities K representing origin–destination pairs, a node set N, and a set of arcs $C = \{A \cup B\}$ where the sets A and B represent, respectively, the sets of toll and toll-free arcs. The TPP can be formulated as BLP in the following way:

$$
\begin{aligned}
\max_{t \geq 0} \quad & \sum_{k \in K} \sum_{(i,j) \in A} t_{ij} n^k x_{ij}^k \\
\text{s.t.} \quad & \forall k \in K : \\
& \min_{x^k, y^k \geq 0} \quad \sum_{(i,j) \in A} (c_{ij} + t_{ij}) x_{ij}^k + \sum_{(i,j) \in B} c_{ij} y_{ij}^k \\
& \text{s.t.} \quad \sum_{j:(j,i) \in A} x_{ji}^k + \sum_{j:(j,i) \in B} y_{ji}^k - \sum_{j:(i,j) \in A} x_{ij}^k - \sum_{j:(i,j) \in B} y_{ij}^k = b_i^k \quad \forall i \in N
\end{aligned}
\tag{1}
$$

where c_{ij} is the travel time associated with arc $(i, j) \in C$ and t_{ij} is the toll associated with arc $(i, j) \in A$. For each commodity $k \in K$, n^k represents the demand for travel between the origin node $o(k)$ and the destination node $d(k)$, and the node demand b_i^k is equal to -1 if $i = o(k)$, to $+1$ if $i = d(k)$, and to 0 otherwise. The binary variable x_{ij}^k (respectively, y_{ij}^k) indicates whether commodity k is routed through the toll arc (respectively, toll-free arc) (i, j). The upper-level seeks to maximize the total amount of revenue, assuming that the users in the lower-level seek to minimize the total travel cost.

3 Differential evolution for bilevel programming

Differential Evolution (DE) [11, 12] is a population-based stochastic method originally designed for continuous optimization. The algorithm starts with a random set of candidate solutions. At each generation, all members of the population undergoes mutation, crossover and selection. In DE/rand/1/bin variant, for each individual $x_i \in \mathbb{R}^{\mathtt{n}}$, with $i = 1, \ldots, \mathtt{NP}$ a vector v is generated

$$
v_i = x_{r_1} + \mathtt{F} \times (x_{r_2} - x_{r_3}) \tag{2}
$$

where $r_1 \neq r_2 \neq r_3$ are distinct and randomly selected indexes (different from i), NP is the population size, and F is the scale parameter that controls the magnitude of the vector differences. After mutation, the trial vector u_i is created by the crossover operation, which uses elements from both the target vector x_i and the donor vector v_i, as

$$
u_{i,j} = \begin{cases} v_{i,j}, \text{ if } \mathtt{rand}(0,1) < \mathtt{CR} \text{ or } j = jRand \\ x_{i,j}, \text{ otherwise} \end{cases} \tag{3}
$$

The crossover rate CR is a user defined parameter, $\mathtt{rand}(0, 1)$ is a uniformly distributed random number in the interval $[0, 1]$, and $jRand$ is a randomly generated integer from 1 to n so that, $u_i \neq x_i$. The fitness of the trial vector u_i is calculated and is compared to that of the target vector x_i. The best solutions are selected for the next generation. This evolutionary process continues until a stopping criterion is reached.

The most straightforward way to apply DE in BLPs is to construct a nested procedure, in which two DE algorithms are employed at each level of the optimization process. In the Bilevel Differential Evolution method (BLDE) [7], the lower-level problem is solved for each upper-level solution. At the upper level, BLDE iteratively evolves two populations: the upper-level population $\mathbf{X}_{ul}$ of size $\texttt{NP}_{ul}$, and a corresponding $\mathbf{X}_{ll}^{*}$ optimum lower-level solutions obtained so far. For each upper-level solution x_i^{ul}, there is an associated optimum lower-level solution x^{*ll}, which is the best response of the lower-level procedure. In the lower level, for a fixed upper-level solution, another DE is employed to evolve the population $\mathbf{X}_{ll}$ of size $\texttt{NP}_{ll}$, returning the best solution of the lower-level problem. A detailed description of this process can be found in [7].

4 Generation of benchmark instances

In this section, we present the characteristics of the benchmark instances, including the number of nodes, the range of arc costs, the variation in the number of commodities, and the percentage of tolled arcs. These parameters were defined based on preliminary studies, such as [4], and expanded to broaden the scope of the evaluation. In addition, we present the results obtained from the generated instances to validate the effectiveness of the proposed algorithm in optimizing toll pricing in road networks.

In [4], the authors used instances with 60 nodes and 208 arcs, where the arc traversal costs were randomly generated within the interval $[2, 20]$. The number of commodities varied in the set $\{20, 30, 40\}$, and the percentage of tolled arcs was defined as $5\%, 10\%, 15\%, 20\%$. Following a similar approach, we propose the generation of new instances by expanding the number of nodes to the range $[10, 60]$, in increments of 5. For a given number of nodes, the number of arcs was generated randomly, proportionally to the reference instance in [4]. The arcs are undirected and a random cost from the interval $[2, 20]$ was assigned to each arc. The number of origin-destination pairs (commodities) varies from 5 to 40 and was generated by randomly selecting node pairs from the network. For all instances, each commodity was assigned a demand weight of 1. Additionally, we ensured that at least one feasible path exists between the origin and destination nodes of each commodity by applying Dijkstra's algorithm [13], thereby guaranteeing network connectivity. The percentage of tolled arcs is selected from the set $5\%, 10\%, 15\%, 20\%$, consistent with the values used in [4].

In practical situations, the selection of routes for implanting tolls is typically based on traffic demand forecasts [14]. To simulate this process, we employ a Differential Evolution (DE) algorithm [11] to identifying the paths most frequently used by users when no tolls exist. Although DE was originally developed to solve optimization problems in continuous search spaces, several studies have proposed strategies to adapt this algorithm to routing optimization problems, such as the Traveling Salesman Problem (TSP) [15] and the Vehicle Routing Problem (VRP) [16], where DE has proven effective in generating high-quality solutions.

The arcs with the highest frequency of occurrence in the routing paths were ranked according to their recurrence, and the top k arcs were selected, where k corresponds to the predefined proportion of tolled arcs relative to the number of nodes in the instance under analysis. In cases where the number of high-frequency arcs was lower than the established tolling proportion, the set was completed by randomly selecting additional arcs from the network, thus ensuring the required proportion for each instance. This resulting set constituted the tolled arcs for each generated instance. This approach also aligns with the methodology adopted by Dewez et al. [4], where tolls were assigned randomly, adding a degree of realism to the generated instances.

Accordingly, for each network size, six combinations were considered for instances with 10 and 15 nodes, and twenty possible combinations for instances with 20 to 60 nodes. These combinations result from variations in the number of commodities and the percentage of tolled arcs. As a result, a total of 192 instances were generated, each containing the number of nodes, a list of origin-destination pairs of the commodities, a list of arcs, and a list of tolled arcs. The definitions of these instances and a supplementary material are available publicly[1].

5 Solution method

To address the TPP described in (1), we use a BLDE variant with multiple followers with the random keys strategy to solve the lower-level combinatorial optimization problem. Figure 1 presents an illustrative scheme of the proposed approach. The following sections describe the encoding strategy adopted here for solving the combinatorial optimization problem of the lower level and the bilevel approach with multiple followers.

[1] https://github.com/ciml/BLDE-MIF-RK

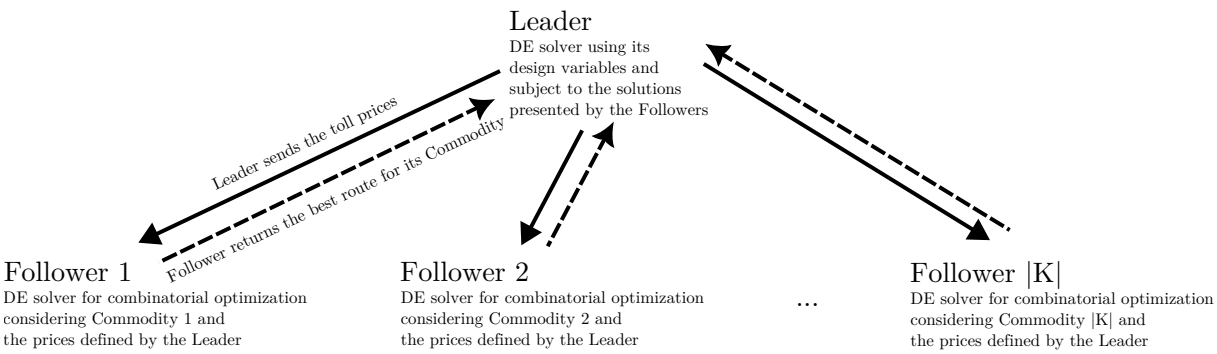


Figure 1. Flowchart of the proposed DE for bilevel optimization of toll pricing using multiple followers. Each follower solves a combinatorial optimization problem for each commodity.

5.1 DE for combinatorial optimization

Although originally designed for optimization in continuous spaces, DE can be adapted to combinatorial problems through the use of the random keys strategy [9]. In this approach, a vector of real numbers is transformed into a permutation of integers: each real value is associated with an integer, and the ascending order of the real values determines the resulting integer sequence. This sequence is then used to construct the routing path for each commodity. For instance, considering a candidate solution for a 10-dimensional optimization problem as $\{0.89, 0.18, 0.95, 0.34, 0.59, 0.03, 0.78, 0.42, 0.68, 0.23\}$, its correspondent permutation representation is $\{5, 1, 9, 3, 7, 4, 8, 6, 0, 2\}$ as the smallest value is in position 5, the second smallest one is position 1, and so on.

To ensure that only existing edges are considered during the path construction, the algorithm starts from the source node of each commodity and retrieves the list of adjacent nodes from the current one. The selected edge is determined by the n-th position in the adjacency list, where n is computed as the modulus between the size of the adjacency list and the current integer value of the sequence. Following the previous example, assuming that the source node of a given commodity has a list of adjacent nodes $\{3, 8, 0, 6\}$, then one has $5 \mod 4 = 1$ and node 8 is select. This process is repeated iteratively until the destination node of the commodity is reached.

5.2 Bilevel optimization with multiple independent followers

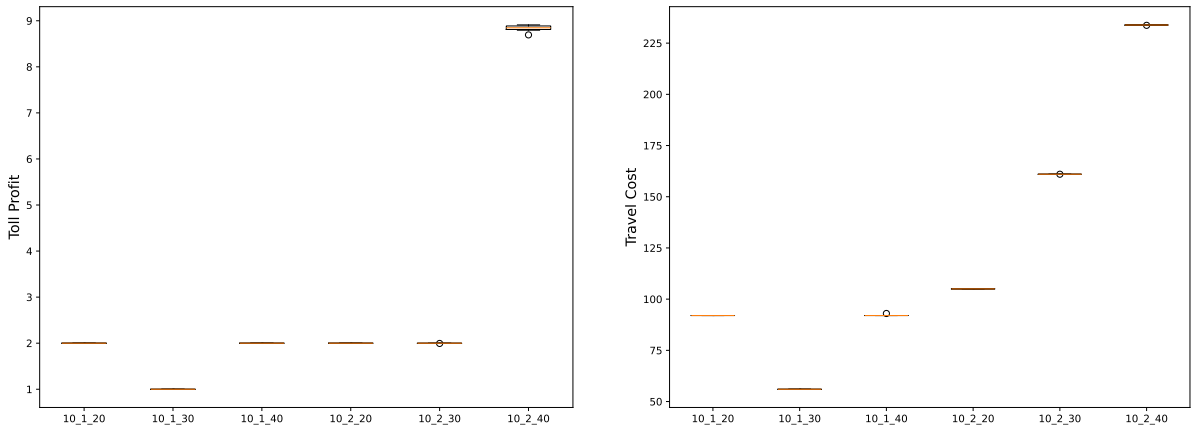
To solve the BLP presented in (1), we adopt a bilevel procedure considering multiple independent followers [17, 18]. In this model, many followers are acting independently, meaning that each follower has its own objective function and constraints and does not share any information with others. Accordingly, the objective function and the constraint set of each follower involve only the leader's variables and the follower's own decision variables. In contrast, the leader's objective function and constraint set depend on both its own variables and the decision variables of all followers.

The proposed approach is based on BLDE, under the assumption that multiple independent followers are present [8]. In this context, for each toll arc configuration defined by the leader, the followers are represented by the commodities. Specifically, for each commodity (follower) k, a DE procedure is executed sequentially, and the best solution obtained for each follower is returned to the leader.

6 Computational experiments

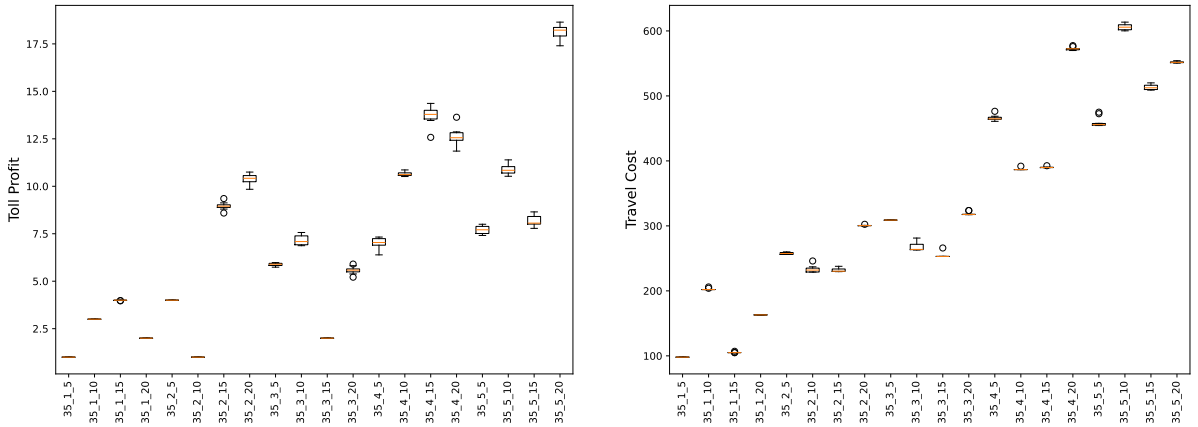
For the experiments, the formulation presented in (1) was adopted. The demand n associated with each commodity was fixed at 1 across all commodities and all evaluated instances. The toll value t was restricted to the interval $[0.1, 1]$. The parameters used for DE were: mutation factor $\texttt{F} = 0.9$, crossover rate $\texttt{CR} = 0.7$, leader population size $\texttt{NP}_{ul} = 20$, follower population size $\texttt{NP}_{ll} = 30$, number of generations of the leader DE $= 30$, number of generations of the follower DE $= 100$, and the DE/rand/1/bin variant. To ensure that the follower selects the shortest path, a penalty cost of 1 was applied for each repeated node in the follower's route, thereby preventing cycles. For both the leader and follower DE approaches, the stopping criterion was set to the maximum number of generations.

The experiments were conducted with 10 independent runs for each instance to ensure the statistical reliability of the results. Figures 2, 3, and 4 respectively, show the evolution of the leader's total profit and the follower's cost, the sum of the commodities' travel costs under different network configurations. The results for the remaining



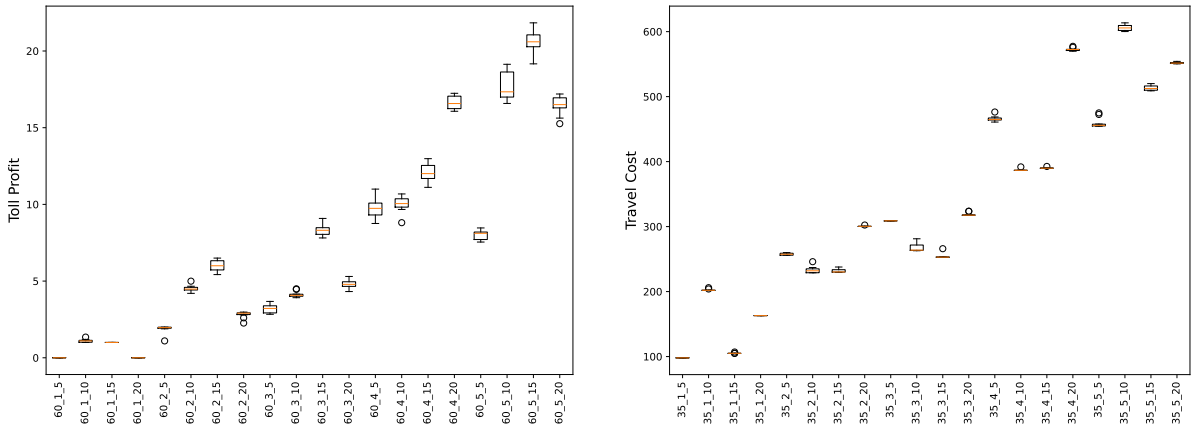
(a) Leader for instances with 10 customers (b) Follower for instances with 10 customers

Figure 2. Resultados obtidos para o Leader e para o Follower nas instâncias com 10 nós.



(a) Leader for instances with 35 customers (b) Follower for instances with 35 customers

Figure 3. Resultados obtidos para o Leader e para o Follower nas instâncias com 35 nós.



(a) Leader for instances with 60 customers (b) Follower for instances with 60 customers

Figure 4. Resultados obtidos para o Leader e para o Follower nas instâncias com 60 nós.

instances are provided in the supplementary material.

Data analysis revealed that the number of commodities emerges as the most influential variable on the leader's profit, dominating the impact of other factors. This phenomenon can be attributed to the increased likelihood of users utilizing tolled arcs, which is further amplified by a higher toll density within the network.

The instances were generated to cover a broad spectrum of complexity by systematically varying the network scale, the number of commodities and tolled arcs, and the configuration settings. Analysis of Figure 2 shows that for smaller networks (10 nodes), the results exhibited significant stability. In contrast, Figures 3 and 4, which correspond to medium-sized (35 nodes) and large-scale networks (60 nodes), reveal greater variability in the solutions as new elements are introduced into the network. This behavior can be attributed to the exponential increase in the optimization problem's complexity as the dimensionality of the search space grows. The variation in instance complexity enables a more robust comparative evaluation of the search mechanisms, allowing for an assessment of their performance across different scenarios.

To ensure the robustness of the conclusions, instances with identical numbers of nodes and edges were generated incorporating variations in network topology, commodity origin-destination distributions, and definitions of priced edges. This structural diversification not only facilitates the evaluation of performance at varying levels of complexity of the problem, but also enables the identification of algorithmic specializations tailored to specific problem configurations, thus contributing to a more comprehensive understanding of the behavior of optimization mechanisms in diverse scenarios.

The plots presented here illustrate the quality of the solutions obtained and their variability. However, for future use of these instances in comparative studies, it would be advisable for researchers to: (i) also present the results in tabular form, allowing for greater clarity regarding the values involved; (ii) perform statistical significance analyses; and (iii) adopt visual comparison strategies that combine the results across all instances, such as performance profiles (see [19, 20]).

7 Concluding remarks and future works

In this work, new instances for the Toll Pricing Problem were proposed with the aim of enabling reproducibility and facilitating comparisons between bilevel optimization approaches. The diversity of structural characteristics across the instances allows for the identification of strengths and limitations in optimization techniques, as well as providing guidance for the development of more efficient strategies. Additionally, a bilevel optimization method with multiple followers, employing Differential Evolution as the search mechanism, was presented. The proposed technique, named BLDE-MIF-RK, was used to generate benchmark results for the instance set defined herein. It successfully solved the instances while fully satisfying the problem's constraints, and the obtained results offer a solid starting point for future studies and research in this field.

For future work, several directions can be explored to expand both the scope and the practical applicability of the contributions presented here. A first approach involves extending the model and the instance generation process to encompass vehicle routing scenarios with simultaneous pickups and deliveries and time window constraints, thereby bringing the problem closer to more realistic logistics contexts and, consequently, to greater operational complexity. Another promising line of research is the investigation of multi-objective approaches that incorporate metrics for reducing environmental impact, improving traffic flow, and assessing social and economic indicators. Such perspectives would allow the problem to be analyzed from a broader perspective, aligned with sustainable public policy objectives.

Acknowledgements. The authors thank the support provided by CNPq (grants 316801/2021-6, 380656/2024-8, and 313452/2025-3), FAPEMIG (grants CEX APQ 03313/22 and 01832/22), Capes, UFJF, and Fiocruz. This work has also been supported by UFJF's High-Speed Integrated Research Network (RePesq)[2].

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] Y. Wang and Z. J. Zhao. Motivations, obstacles, and resources. *Public Performance & Management Review*, vol. 37, n. 4, pp. 679–704, 2014.

[2] `https://repesq.ufjf.br`

[2] L. Perera, R. G. Thompson, and W. Wu. Toll and subsidy for freight vehicles on urban roads: A policy decision for city logistics. *Research in Transportation Economics*, vol. 90, pp. 101132, 2021.
[3] T. Weise, M. Wagner, B. Li, X. Zhang, and J. Lässig. Special issue on benchmarking of computational intelligence algorithms in the applied soft computing journal. *Applied Soft Computing*, vol. 93, pp. 106502, 2020.
[4] S. Dewez, M. Labbé, P. Marcotte, and Gilles Savard. New formulations and valid inequalities for a bilevel pricing problem. *Operations Research Letters*, vol. 36, n. 2, pp. 141–149, 2008.
[5] V. Kalashnikov, F. Camacho, R. Herrera-Maldonado, and N. Kalashnykova. A heuristic algorithm solving bilevel toll optimization problems. *The International Journal of Logistics Management*, vol. 27, 2016.
[6] 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, n. 8, pp. 3529–3549, 2010.
[7] J. S. Angelo, E. Krempser, and H. J. C. Barbosa. Differential evolution for bilevel programming. In *IEEE Congress on Evolutionary Computation*, pp. 470–477, 2013.
[8] J. S. Angelo. *Metaheurísticas para Problemas de Otimização em Dois Níveis*. Thesis, Laboratório Nacional de Computação Científica (LNCC). (In Portuguese)), 2014.
[9] J. C. Bean. Genetic algorithms and random keys for sequencing and optimization. *ORSA Journal on Computing*, vol. 6, n. 2, pp. 154–160, 1994.
[10] J. F. Bard. *Practical Bilevel Optimization*. Kluwer Academic Publisher, 1998.
[11] R. Storn and K. Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, vol. 11, n. 4, pp. 341–359, 1997.
[12] K. V. Price. An introduction to Differential Evolution. *New Ideas in Optimization*, vol. , pp. 79–108, 1999.
[13] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, vol. 1, n. 1, pp. 269–271, 1959.
[14] B. Crowe. Toll road traffic forecasts, their failures and how to fix them. Technical report, JDI Executive Programs, 2014.
[15] M. Mi, X. Huifeng, Z. Ming, and G. Yu. An improved differential evolution algorithm for tsp problem. In *2010 International Conference on Intelligent Computation Technology and Automation*, volume 1, pp. 544–547, 2010.
[16] L. Mingyong and C. Erbao. An improved differential evolution algorithm for vehicle routing problem with simultaneous pickups and deliveries and time windows. *Engineering Applications of Artificial Intelligence*, vol. 23, n. 2, pp. 188–195, 2010.
[17] H. Calvete and C. Galé. Linear bilevel multi-follower programming with independent followers. *Journal of Global Optimization*, vol. 39, n. 3, pp. 409–417, 2007.
[18] S. R. Arora and R. Narang. 0-1 bilevel fractional programming problem with independent followers. *International Journal of Optimization: Theory, Methods and Applications*, vol. 1, n. 2, pp. 225–238, 2009.
[19] E. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Math. Programming*, vol. 91, n. 2, pp. 201–213, 2002.
[20] H. J. C. Barbosa, H. S. Bernardino, and A. M. S. Barreto. Using performance profiles to analyze the results of the 2006 CEC constrained optimization competition. In *IEEE Congress on Evolutionary Computation*, pp. 1–8, 2010.