On the Use of DELEqC-III in Bilevel Problems with Linear Equality Constraints

Heder S. Bernardino[1], Jaqueline S. Angelo[2], Helio J. C. Barbosa[1]

[1] *Universidade Federal de Juiz de Fora, Juiz de Fora, MG, Brazil*
heder@ice.ufjf.br, *hcbm@lncc.br*
[2] *Fundação Oswaldo Cruz, Rio de Janeiro, RJ, Brazil*
jaqueline.angelo@fiocruz.br

Abstract. The bilevel problem (BLP) is an optimization problem that has another optimization problem in its constraints. This framework finds utility in modeling decentralized scenarios, which arise in many real-world applications such as traffic management, transportation, and economic policy. Differential Evolution (DE) techniques have emerged in literature for addressing such complex problems. However, handling constraints, particularly linear equality constraints, poses a significant challenge for DE and other metaheuristics. To address this issue, we previously introduced DELEqC algorithm, enhancing DE with a mechanism to deal with the linear equality constraints. A specialized variant, BL-DELEqC, was further proposed to tackle general BLPs. Another variant, DELEqC-III, transforms the original constrained optimization problem into a lower-dimensional unconstrained one, offering applicability to BLPs with linear equality constraints. Thus, we explore in this study the efficacy of DELEqC-III in handling BLPs with linear equality constraints. The proposed BL-DELEqC-III is compared to BL-DELEqC on a selection of benchmark BLPs, demonstrating superior results.

Keywords: Bilevel programming, Linear equality constraints, Differential evolution

1 Introduction

The bilevel programming problem (BLP) models an important class of hierarchical optimization tasks that involve two levels: the upper level (UL) and the lower level (LL). The main characteristic of a BLP is that the UL optimization problem is constrained by a nested LL optimization problem. This nested structure is defined as a mathematical programming problem in which the UL has a subset of its variables constrained to be the optimal solution of the LL optimization problem.

The problem of interest in this paper is the constrained BLP defined as

$$\begin{aligned} &\min_{\mathbf{x}_{ul},\mathbf{x}_{ll}} F(\mathbf{x}_{ul},\mathbf{x}_{ll}) \\ &\text{subject to } \min_{\mathbf{x}_{ll}} f(\mathbf{x}_{ul},\mathbf{x}_{ll}) \\ &\qquad\qquad \text{subject to } h(\mathbf{x}_{ul},\mathbf{x}_{ll}) = 0 \end{aligned} \tag{1}$$

where $\mathbf{x}_{ul} \in \mathbb{R}^n$ and $\mathbf{x}_{ll} \in \mathbb{R}^m$ are the decision variables of the upper- and the lower-level, respectively, $F : \mathbb{R}^{n+m} \to \mathbb{R}$ is the UL objective function, $f : \mathbb{R}^{n+m} \to \mathbb{R}$ is the LL objective function and $h : \mathbb{R}^{n+m} \to \mathbb{R}^q$ denotes the LL linear equality constraints. Given the complexity of these problems, Evolutionary Algorithms (EAs) [1] are a common alternative for solving them, due to their robustness and adaptability to this type of application. While various methods have been developed to tackle unconstrained general BLPs, there is a significant gap in research regarding methods designed for the constrained scenario.

Although EAs are widely used for solving BLPs, dealing with constraints in such methods is not trivial, as they were originally designed for unconstrained search. Consequently, additional mechanisms are required to handle constrained optimization problems effectively. Numerous constraint handling techniques have been proposed, including penalty methods, selection approaches, special representation schemes and move operators, repair techniques, among others [2, 3]. In this paper, we focus on obtaining solutions that automatically satisfy all linear equality constraints. As equality constraints are difficult to be exactly fulfilled, they are typically approximated by inequality constraints (i.e., $|h(\mathbf{x})| \leq \varepsilon$), where a small tolerance value $\varepsilon > 0$ is set by the user. We can find a few examples of this class of problems in the library of test problems that applies such a strategy [4].

While this approach temporarily enlarges the feasible space, it remains challenging to obtain feasible or high-quality candidate solutions [5]. Here, the LL optimization problem is subject to linear equality constraints denoted as $E_{\mathbf{x}}\mathbf{x}_{ul} + E_{\mathbf{y}}\mathbf{x}_{ll} = \mathbf{c}$, where $E_{\mathbf{y}} \in \mathbb{R}^{q\times m}$, $E_{\mathbf{x}} \in \mathbb{R}^{q\times n}$ and $\mathbf{c} \in \mathbb{R}^q$, with $q \leq m$. Note that both the LL objective f and the constraint h are functions of the upper- and lower-level variables, $\mathbf{x}_{ul}$ and $\mathbf{x}_{ll}$, respectively.

To address this issue, the BL-DELEqC method was proposed in [6] using the BL-DE [7] approach coupled with DELEqC [8] for solving BLPs with linear equality constraints in the lower level. Starting with an initial population that satisfies the linear equality constraints, DELEqC ensures feasibility throughout the search process by avoiding the standard DE crossover and using a specialized mutation scheme. DELEqC-II [9] and DELEqC-III [10] were further proposed enhancing DELEqC with new mechanisms to deal with the linear equality constraints for solving single-level optimization problems. Given that DELEqC-III outperforms DELEqC-II in most test problems, we focus on using DELEqC-III. Differently from DELEqC and DELEqC-II, the DELEqC-III reduces the dimension of the problem from n to $n - m$, where n is the number of variables and m denotes the number of linear equality constraints. This transformation converts the problem into an unconstrained one, enabling the application of crossover and mutation operations traditionally used in DE.

In this paper, we propose using the BL-DE approach in which DELEqC-III is used for solving the LL optimization problem. The proposed BL-DELEqC-III, automatically satisfies the linear equality constraints by reducing the dimension of the LL problem transforming it into an unconstrained one.

The remainder of the paper is structured as follows. Sections 2 and 3 provide background material on the DE algorithm and the DELEqC versions for single-level optimization. The proposed BL-DELEqC-III is described in Section 4 and Section 5 presents the experiments. Section 6 presents our conclusions and final remarks.

2 Differential Evolution (DE)

DE [11, 12] is a population-based stochastic method designed to solve optimization problems in continuous search spaces. DE and its variants have been successfully applied to a wide range of real-world problems and are recognized as some of the most competitive and versatile evolutionary methods [13].

The method starts with a random population $\mathbf{X}$ of N candidate solutions within the search space defined by the lower ($x^{(L)}$) and upper bounds ($x^{(U)}$). Each candidate solution $\mathbf{x}^i$ is evaluated by the objective function $f(\mathbf{x}^i)$, assigning a quality measure to that individual (its fitness). At each generation, all individuals undergo mutation and crossover operations, following a predefined DE variant. The DE variant used here is denoted by DE/target-to-best/1/bin (Eqs. 2 and 3). For each vector $\mathbf{x}^i$ of size D, where $i = 1, \ldots, N$, a vector $\mathbf{w}^i$ is generated by mutation according to:

$$\mathbf{w}^i = \mathbf{x}^i + \mathtt{F} \times (\mathbf{x}^{best} - \mathbf{x}^i) + \mathtt{F} \times (\mathbf{x}^{r_1} - \mathbf{x}^{r_2}) \tag{2}$$

where $\mathbf{x}^i$ is the target individual, $\mathbf{x}^{best}$ is the best individual of the current population, r_1-th and r_2-th are distinct and randomly selected individuals, different from the i-th, and $\mathtt{F}$ is the scale parameter used to control the amplitude of the search in the direction of the applied differences. In the crossover operation, the trial vector $\mathbf{v}^i$ is generated using elements from both the target vector $\mathbf{x}^i$ and the donor vector $\mathbf{w}^i$ as:

$$v_j^i = \begin{cases} w_j^i, & \text{if } \mathtt{rand}(0,1) < \mathtt{CR} \text{ or } j = jRand, \\ x_j^i, & \text{otherwise} \end{cases} \tag{3}$$

where $i = 1, \ldots, N$ and $j = 1, ..., D$. The crossover rate $\mathtt{CR}$ is a user-defined parameter, $\mathtt{rand}(0, 1)$ is a uniformly distributed random number in $[0, 1]$, $jRand$ is a randomly generated integer from 1 to D, where D is the dimension of the problem. The fitness of the trial vector $\mathbf{v}^i$, is compared to that of the target vector $\mathbf{x}^i$ and the best solution is selected for the next generation of $\mathbf{X}$. This evolutionary process continues until a stopping criterion is met.

3 DELEqC versions for single-level optimization

In our first attempt to deal with linear equality constraints, DELEqC was proposed in [8] to deal with the following single-level constrained optimization problem

$$\begin{aligned} &\min_{\mathbf{x}} f(\mathbf{x}) \\ &\text{subject to } E\mathbf{x} = \mathbf{c} \end{aligned} \tag{4}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $E \in \mathbb{R}^{m \times n}$ and $\mathbf{c} \in \mathbb{R}^m$. It is assumed that $m < n$ and E has full rank (rank$(E) = m$), that is, the rows of E are linearly independent. A candidate solution $\mathbf{x} \in \mathbb{R}^n$ is said to be feasible if $\mathbf{x} \in S$ where $S = \{\mathbf{x} \in \mathbb{R}^n \ : E\mathbf{x} = \mathbf{c}\}$ denotes the feasible set. A vector $\mathbf{d} \in \mathbb{R}^n$ is a feasible direction at the point $\mathbf{x} \in S$ if $\mathbf{x} + \mathbf{d}$ is feasible, that is, $E(\mathbf{x} + \mathbf{d}) = \mathbf{c}$. Hence, the feasible direction $\mathbf{d}$ must satisfy $E\mathbf{d} = \mathbf{0}$ or that any feasible direction belongs to the null space of the matrix E, denoted as, $\mathcal{N}(E) = \{\mathbf{x} \in \mathbb{R}^n : E\mathbf{x} = 0\}$. In this way, given two feasible vectors $\mathbf{x}^1$ and $\mathbf{x}^2$ one can see that $\mathbf{d} = \mathbf{x^1} - \mathbf{x^2}$ is a feasible direction, as $E(\mathbf{x}^1 - \mathbf{x}^2) = 0$. DELEqC [8] explores this principle. By starting with a feasible initial population w.r.t. the linear equality constraints, the feasibility is maintained by avoiding the standard DE crossover and using an adequate mutation scheme along the search process. As a result, the method consistently generates feasible vectors, provided that the vectors involved in the differences are themselves feasible.

An improved version, called DELEqC-II, was proposed in [9, 14], which employs both mutation and crossover operators. This is accomplished by applying a convex combination of the target vector and the donor vector during the crossover operation. Considering $\texttt{CR} = (0, 1]$, the proposed crossover is a convex combination of the involved vectors: $v_j^i = (1 - \texttt{CR}) \times w_j^i + \texttt{CR} \times x_j^i$, where $i = 1, \ldots, N$, and $j = 1, \ldots, D$. Additionally, a projection procedure was also adopted to correct the solutions that "escape" from the feasible set, due to numerical errors in the floating-point arithmetic operations that may occur during the search.

A different scheme was proposed in [10], where both DE's mutation and crossover operators were applied in their original form. In DELEqC-III, to maintain feasibility with respect to the linear equality constraints, the null-space approach was used. It is based on constructing a matrix $Z \in \mathbb{R}^{n \times (n-m)}$ such that its columns form a basis for $\mathcal{N}(E)$. For any $\mathbf{d} \in \mathcal{N}(E)$, there is $\mathbf{p} \in \mathbb{R}^{(n-m)}$ such that $\mathbf{d} = Z\mathbf{p}$. If $\tilde{\mathbf{x}}$ is a feasible solution of the linear system $E\tilde{\mathbf{x}} = \mathbf{c}$, then the search region can be defined as $S = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} = \tilde{\mathbf{x}} + Z\mathbf{p}, \mathbf{p} \in \mathbb{R}^{(n-m)}\}$ where any $\mathbf{x} \in \mathbb{R}^n$ satisfies the constraints. By defining S as previously, a new function $\phi : \mathbb{R}^{(n-m)} \to \mathbb{R}$ is given by $\phi(\mathbf{p}) = f(\tilde{\mathbf{x}} + Z\mathbf{p})$, which should be minimized. This new optimization problem has no constraints on $\mathbf{p}$, and the number of variables is reduced from n to $(n - m)$. DELEqC-III applies the same initial population procedure from DELEqC and DELEqC-II that generates feasible candidate solutions. By using information of the Z matrix, the original problem (4) is transformed into the lower-dimensional unconstrained problem (min $\phi(\mathbf{p})$).

4 Proposed solution method

The proposed BL-DELEqC-III applies a nested approach, employing two DE algorithms, each responsible for optimizing one level of the BLP. A pseudo-code of the method is given by Alg. 1, 2 and 3.

4.1 The upper-level procedure

The BL-DELEqC-III algorithm (or $\texttt{DE}_{\texttt{UL}}$) iteratively evolves pairs of solutions through successive evolutionary steps. For each UL solution $\mathbf{x}_{ul}^i$, where $i = 1, \ldots, N_{ul}$, there is an associated optimum LL solution $\mathbf{x}_{ll}^{*i}$. The $\texttt{DE}_{\texttt{UL}}$ maintains two populations: the UL population $\mathbf{X}_{ul}$, which contains N_{ul} upper-level candidate solutions, and $\mathbf{X}_{ll}^*$, a collection of the corresponding N_{ul} optimum lower-level solutions obtained so far. The initialization and evaluation of both populations occur between lines 1 and 5 of Alg. 1.

At each generation, each UL target vector $\mathbf{x}_{ul}^i$ undergoes mutation and crossover operations to generate its respective UL trial vector $\mathbf{v}_{ul}^i$ (line 8). For each $\mathbf{v}_{ul}^i$, the lower-level DE is executed (line 9). The $\texttt{DE}_{\texttt{LL}}$ is outlined in Alg. 2, which returns the best LL solution $\mathbf{v}_{ll}^*$ from the LL population $\mathbf{X}_{ll}$ of size N_{ll}, obtained in response to the given $\mathbf{v}_{ul}^i$. In the upper level, the pair of trial vectors ($\mathbf{v}_{ul}^i$, $\mathbf{v}_{ll}^*$) is then evaluated based on the UL objective function (line 10). At line 11, its fitness in both upper and lower levels is compared to that of its respective pair of target vectors ($\mathbf{x}_{ul}^i$, $\mathbf{x}_{ll}^{*i}$). The better pair of solutions is selected for the next generation of populations $\mathbf{X}_{ul}$ and $\mathbf{X}_{ll}^*$. At the upper and lower levels, their respective iterative procedure continues until a stopping criterion, given by a maximum number of upper- and lower-level generations, G_{ul} and G_{ll}, respectively. BL-DELEqC-III concludes by returning the best-found pair of solutions.

4.2 The lower-level procedure

The DELEqC-III method (or $\texttt{DE}_{\texttt{LL}}$) is called for each UL candidate solution $\mathbf{v}_{ul}^i$. The input data related to the LL optimization problem are: $E_{\mathbf{y}}$ (matrix), and $\mathbf{c}$ (vector) associated with the linear equality constraints, and $Z_{\mathbf{x}_{ll}}$ (matrix) containing a basis for the null space of $E_{\mathbf{y}}$.

Differently from the single-level problem, in BLP the linear equality constraints of LL can be written as

$$E_{\mathbf{x}}\mathbf{x}_{ul} + E_{\mathbf{y}}\mathbf{x}_{ll} = \mathbf{c} \tag{5}$$

where $\mathbf{x}_{ul} \in \mathbb{R}^n$, $\mathbf{x}_{ll} \in \mathbb{R}^m$, $E_{\mathbf{x}} \in \mathbb{R}^{q\times n}$, $E_{\mathbf{y}} \in \mathbb{R}^{q\times m}$ and $\mathbf{c} \in \mathbb{R}^q$, with $q \leq m$, where q is the number of linear equality constraints. Following the same principle as in the single-level case, the feasibility concerning the LL linear equality constraints is maintained by constructing a matrix $Z_{\mathbf{x}_{ll}} \in \mathbb{R}^{m\times(m-q)}$ such that its columns form a basis for $\mathcal{N}(E_{\mathbf{y}})$, the null space of the matrix $E_{\mathbf{y}}$. It is noteworthy that the UL variables are passed as parameters to the LL problem, so that, the set of constraints can be rewritten as $E_{\mathbf{y}}\mathbf{x}_{ll} = \mathbf{c} - E_{\mathbf{x}}\mathbf{x}_{ul}$. Hence, the null space of matrix $E_{\mathbf{y}}$ can be obtained by solving the homogeneous system $E_{\mathbf{y}}\mathbf{x}_{ll} = 0$.

For any $\mathbf{d} \in \mathcal{N}(E_{\mathbf{y}})$, there is $\mathbf{p}_{\mathbf{y}} \in \mathbb{R}^{(m-q)}$ such that $\mathbf{d} = Z_{\mathbf{x}_{ll}}\mathbf{p}_{\mathbf{y}}$. If $\tilde{\mathbf{x}}_{ll}$ is a feasible solution of the linear system (5), then the search region can be defined as

$$\hat{S} = \{\mathbf{x}_{ll} \in \mathbb{R}^m : \mathbf{x}_{ll} = \tilde{\mathbf{x}}_{ll} + Z_{\mathbf{x}_{ll}}\mathbf{p}_{\mathbf{y}}, \mathbf{p}_{\mathbf{y}} \in \mathbb{R}^{(m-q)}\} \tag{6}$$

where any $\mathbf{x}_{ll} \in \mathbb{R}^m$ satisfies the constraints. By defining $\hat{S}$ as in (6) a new function $\hat{\phi} : \mathbb{R}^{(m-q)} \to \mathbb{R}$ can be as $\hat{\phi}(\mathbf{p}_{\mathbf{y}}) = f(\tilde{\mathbf{x}}_{ll} + Z_{\mathbf{x}_{ll}}\mathbf{p}_{\mathbf{y}})$ that should be minimized. The BLP is then transformed into

$$\begin{aligned} &\min_{\mathbf{x}_{ul},\mathbf{x}_{ll}} F(\mathbf{x}_{ul}, \mathbf{x}_{ll}) \\ &\text{subject to } \min_{\mathbf{p}_{\mathbf{y}}} \hat{\phi}(\mathbf{p}_{\mathbf{y}}) = f(\tilde{\mathbf{x}}_{ll} + Z_{\mathbf{x}_{ll}}\mathbf{p}_{\mathbf{y}}), \text{ where } \mathbf{x}_{ll} = \tilde{\mathbf{x}}_{ll} + Z_{\mathbf{x}_{ll}}\mathbf{p}_{\mathbf{y}} \end{aligned} \tag{7}$$

The $\texttt{DE}_{\texttt{LL}}$ starts generating a feasible initial population, as presented in Alg. 3. Associated with each member of the population $\mathbf{X}_{ll}$, there are: (i) the vector $\mathbf{x}_{ll} \in \mathbb{R}^m$ (LL decision variables); and (ii) the vector $\mathbf{p}_{\mathbf{y}} \in \mathbb{R}^{(m-q)}$ (decision variables of the transformed problem). In Alg. 2, mutation and crossover operations are performed on vector $\overline{\mathbf{v}}_{ll} \in \mathbb{R}^{m-q}$ (line 4). As individuals are evaluated in the original space, i.e., $\mathbb{R}^m$, the operation $\mathbf{v}_{ll} = \tilde{\mathbf{x}}_{ll} + Z_{\mathbf{x}_{ll}}\overline{\mathbf{v}}_{ll}$ (line 5) has to be performed before fitness evaluation (similarly, $\mathbf{x}_{ll} = \tilde{\mathbf{x}}_{ll} + Z_{\mathbf{x}_{ll}}\mathbf{p}_{\mathbf{y}}$ in line 6 of Alg. 3). After evaluating the individual based on the LL objective function (line 6), the trial vector $\mathbf{v}_{ll}$ is compared to the target vector $\mathbf{x}_{ll}$ (line 7), and the one with the best objective function is selected for the next generation. Next, the population $\mathbf{X}_{ll}$ is updated with the pair $(\mathbf{x}^i_{ll}, \mathbf{p}^i_{\mathbf{y}})$. The $\texttt{DE}_{\texttt{LL}}$ return the best LL solution found for the given $\mathbf{v}_{ul}$.

Algorithm 1: BL-DELEqC-III or $\texttt{DE}_{\texttt{UL}}$

1 **for** $i \leftarrow 1$ **to** N_{ul} **do**
2   $\mathbf{x}^i_{ul} \leftarrow$ `init_random_pop`$(x^{(L)}_{ul}, x^{(U)}_{ul})$;
3   $(\mathbf{x}^*_{ll}, f_{\mathbf{x}^*_{ll}}) \leftarrow \texttt{DE}_{\texttt{LL}}(\mathbf{x}^i_{ul})$;   `// Algorithm 2`
4   $f_{X_{ul}}[i] \leftarrow \texttt{evaluate}_{\texttt{ul}}(\mathbf{x}^i_{ul}, \mathbf{x}^*_{ll})$;
5   $f_{X^*_{ll}}[i] \leftarrow f_{\mathbf{x}^*_{ll}}$;
6 **for** $g \leftarrow 1$ **to** G_{ul} **do**
7   **for** $i \leftarrow 1$ **to** N_{ul} **do**
8     $\mathbf{v}^i_{ul} \leftarrow \texttt{apply\_op}_{\mathbf{v}_{\texttt{ul}}}(\mathsf{CR}, \mathsf{F}, \mathbf{x}^{r_1,r_2,best,i}_{ul})$;
9     $(\mathbf{v}^*_{ll}, f_{\mathbf{v}^*_{ll}}) \leftarrow \texttt{DE}_{\texttt{LL}}(\mathbf{v}^i_{ul})$; `// Algorithm 2`
10     $f_{\mathbf{v}_{ul}} \leftarrow \texttt{evaluate}_{\texttt{ul}}(\mathbf{v}^i_{ul}, \mathbf{v}^*_{ll})$;
11   **if** $f_{\mathbf{v}_{ul}} \leq f_{X_{ul}}[i]$ **and** $f_{\mathbf{v}^*_{ll}} \leq f_{X^*_{ll}}[i]$ **then**
12       $\mathbf{x}^i_{ul} \leftarrow \mathbf{v}^i_{ul}$; $f_{X_{ul}}[i] \leftarrow f_{\mathbf{v}_{ul}}$;
13       $\mathbf{x}^{*i}_{ll} \leftarrow \mathbf{v}^i_{ll}$; $f_{X^*_{ll}}[i] \leftarrow f_{\mathbf{v}_{ll}}$;
14     $\mathbf{X}_{ul}[i] \leftarrow \mathbf{x}^i_{ul}$; $\mathbf{X}^*_{ll}[i] \leftarrow \mathbf{x}^{*i}_{ll}$
15 `select_best`$(f_{X^*_{ul}}, f_{X^*_{ll}})$;
16 **return** the *best pair of solutions* found

Algorithm 2: DELEqC-III or $\texttt{DE}_{\texttt{LL}}$

input : $\mathbf{v}_{ul}$ (UL solution), problem data ($E_{\mathbf{y}}$, $\mathbf{c}$, $Z_{\mathbf{x}_{ll}}$).

1 $(\tilde{\mathbf{x}}_{ll}, \mathbf{X}_{ll}) \leftarrow$ `init_feasible_pop` $(\mathbf{X}_{ll}, \mathbf{v}_{ul}, E_{\mathbf{y}}, \mathbf{c}, Z_{\mathbf{x}_{ll}})$;   `// Algorithm 3`
2 **for** $g \leftarrow 1$ **to** G_{ll} **do**
3   **for** $i \leftarrow 1$ **to** N_{ll} **do**
4     $\overline{\mathbf{v}}^i_{ll} \leftarrow \texttt{apply\_op}_{\mathbf{v}_{\texttt{ll}}}(\mathsf{CR}, \mathsf{F}, \mathbf{p}^{r_1,r_2,best,i}_{\mathbf{y}})$; `//` $\overline{\mathbf{v}}_{ll}, \mathbf{p}_{\mathbf{y}} \in \mathbb{R}^{(m-q)}$
5     $\mathbf{v}^i_{ll} = \tilde{\mathbf{x}}_{ll} + Z_{\mathbf{x}_{ll}}\overline{\mathbf{v}}^i_{ll}$; `//` $\mathbf{v}_{ll}, \tilde{\mathbf{x}}_{ll} \in \mathbb{R}^m; Z_{\mathbf{x}_{ll}} \in \mathbb{R}^{m\times(m-q)}$
6     $f_{\mathbf{v}_{ll}} \leftarrow \texttt{evaluate}_{\texttt{ll}}(\mathbf{v}_{ul}, \mathbf{v}^i_{ll})$;
7   **if** $f_{\mathbf{v}_{ll}} \leq f_{\mathbf{x}_{ll}}[i]$ **then**
8       $\mathbf{x}^i_{ll} \leftarrow \mathbf{v}^i_{ll}$; $\mathbf{p}^i_{\mathbf{y}} \leftarrow \overline{\mathbf{v}}^i_{ll}$; $f_{X_{ll}}[i] \leftarrow f_{\mathbf{v}_{ll}}$;
9     $\mathbf{X}_{ll}[i] \leftarrow (\mathbf{x}^i_{ll}, \mathbf{p}^i_{\mathbf{y}})$
10 `select_best`$(f_{\mathbf{X}^*_{ll}})$;
11 **return** the *best solution* found and its *fitness*

Algorithm 3: `init_feasible_pop`

input : $\mathbf{X}_{ll}$ (LL population matrix), $\mathbf{v}_{ul}$ (UL solution) and problem data ($E_{\mathbf{y}}$, $\mathbf{c}$, $Z_{\mathbf{x}_{ll}}$)

1 $M = E_{\mathbf{y}}E^{\mathrm{T}}_{\mathbf{y}}$;
2 Perform LU Factorization: $M = LU$;
3 Solve $M\mathbf{b} = \mathbf{c}$ ($L\mathbf{w} = \mathbf{c}$ and $U\mathbf{b} = \mathbf{w}$) ;
4 $\tilde{\mathbf{x}}_{ll} = E^{\mathrm{T}}_{\mathbf{y}}\mathbf{b}$;   `//` $\tilde{\mathbf{x}} \in \mathbb{R}^m$
5 **for** $i \leftarrow 1$ **to** N_{ll} **do**
6   $\mathbf{x}^i_{ll} = \tilde{\mathbf{x}}_{ll} + Z_{\mathbf{x}_{ll}}\mathbf{p}^i_{\mathbf{y}}$;   `//` $\mathbf{p}^i \in \mathbb{R}^{(m-q)}$ `is randomly generated`
7   $f^i_{\mathbf{x}_{ll}} \leftarrow \texttt{evaluate}_{\texttt{ll}}(\mathbf{v}_{ul}, \mathbf{x}^i_{ll})$;
8   $\mathbf{X}_{ll}[i] \leftarrow (\mathbf{x}^i_{ll}, \mathbf{p}^i_{\mathbf{y}})$
9 **return** $\tilde{\mathbf{x}}_{ll}$, $\mathbf{P}_{\mathbf{xp}}$

5 Computational Experiments

Computational experiments were conducted to evaluate the performance of the proposed BL-DELEqC-III against BL-DE [7] and BL-DELEqC [6], which is BL-DE equipped with DELEqC [8]. The proposal was implemented in C++ and the codes are publicly available[1]. The matrix $Z_{\mathbf{x}_{ll}}$ was obtained via the command $Z_{\mathbf{x}_{ll}}$ = `null(`$E_{\mathbf{y}}$`,'r')` in Matlab®. Note that any other procedure to solve the homogeneous system $E_{\mathbf{y}}\mathbf{x}_{ll} = 0$ can be used to generate the matrix $Z_{\mathbf{x}_{ll}}$. Supplementary material is also available describing the test problems analyzed. We performed 10 independent runs for all methods.

5.1 Parameters settings

We used Performance Profiles (PPs) [15, 16] for selecting the values of `F`, `CR`, and DE variant of BL-DELEqC-III. PPs are tools for visualizing and analyzing results obtained by a set of methods. They help identify the technique that performs best across a range of problems and the one that produces the most reliable results. Additionally, the area under the PP curves indicates the general performance of the methods when solving a set of different problems. We analyzed the following values: `F` $\in \{0.6, 0.7, 0.8, 0.9, 1\}$, `CR` $\in \{0.5, 0.6, 0.7, 0.8, 0.9\}$, and the variants DE/rand/1/bin, DE/best/1/bin, DE/target-ro-rand/1/bin and DE/target-to-best/1/bin. The remaining parameters are the same as in [6]. As the number of generations used for solving the lower-level problem critically affects the computational budget, we considered in the analyses the combination of the problem (Problems P.1-P.5) and the number of LL generations ($G_{ll} = \{50, 100, 200, 500\}$) as different problems in PPs. This comparative analysis results in $5 \times 4 = 20$ problems. The results obtained by the maximization problems were multiplied by -1 so the smaller values are preferable in all the cases analyzed. The best results for BL-DELEqC-III were obtained using `CR`$= 0.7$, `F`$= 0.7$, and DE/target-to-best/1/bin. A comparative analysis of the DE variants in BL-DELEqC-III can be found in the supplementary material.

The parameters selected for BL-DELEqC and BL-DE were the same as in [6], that is, `F`=0.7, DE/target-to-rand/1/bin variant, upper- and lower-level population size equal to 30, and 200 generations for the UL problem. In addition, `CR`$= 0.9$ and $|h(x)| \leq \epsilon = 10^{-4}$ were used for the standard BL-DE.

5.2 Analysis of the Results

We compare the results obtained by BL-DELEqC-III with those presented in [6]. The results are analyzed in terms of the UL objective function value $F(\mathbf{x}_{ul}, \mathbf{x}_{ll})$ and the number of LL function evaluations #FELL. We also analyzed the performance of the methods considering 50, 100, 200, and 500 generations at the lower level (G_{ll}). The stop criteria for the UL problem is the maximum number of generations $G_{ul} = 200$ for all tested techniques. In addition to the maximum number of generation, we also adopted $\alpha = \sum_{i=1}^{m} \frac{\sigma^2(y_i^t)}{\sigma^2(y_i^{initial})} < \alpha_{stop} = 10^{-4}$ in the stopping criterion at the lower-level problem, where m is the number of LL variables, y_i^t are the LL variables in generation t and $y_i^{initial}$ are the LL variables in the initial population.

Tables 1, 2 and 3 present the results obtained, with median and mean values rounded to 2 decimal places. The "fr" indicates the number of feasible runs, that is, violation of the LL constraints is not superior to 10^{-4} and $|f(\mathbf{x}_{ul}, \mathbf{x}_{ll}) - f^*| \leq 0.1$[2]. Considering the number of feasible runs, BL-DELEqC-III demonstrates greater robustness, indicated by fr=10 in all test problems for any value of G_{ll}. This behavior is attributed to its feasibility-preserving approach. Regarding the final solution obtained, BL-DELEqC-III outperforms the other approaches for every number of G_{ll}, except for Problems P.1 and P.4, both with G_{ll} equal to 200 and 500, in which BL-DE obtained the best mean and median results. The good performance observed for BL-DE in these cases, when G_{ll} is large, is caused by the flexibility of the equality constraints commonly adopted in metaheuristics. The approaches with DELEqC and DELEqC-III handle these constraints without this strategy. Also, BL-DELEqC reached the smallest #FELL in all tested cases, except for Problem P.1, in which BL-DE concluded its search with a small #FELL.

6 Conclusions

In this paper, we propose a bilevel method that employs DE for solving the UL optimization problem, while DELEqC-III is used to handle the linear equality constraints of the LL problem. Computational experiments were conducted to evaluate the performance of our proposal when compared to other methods from the literature. The

[1] `https://github.com/ciml/blde-deleqc`

[2] See the supplementary material for the description of the best solution f^* of each test problem.

Table 1. Results for Problems P.1 and P.2.

Method	G_{ll}	Problem P.1					Problem P.2				
		$F(\mathbf{x}_{ul}, \mathbf{x}_{ll})$		#FELL		fr	$F(\mathbf{x}_{ul}, \mathbf{x}_{ll})$		#FELL		fr
		Mean	Median	Mean	Median		Mean	Median	Mean	Median	
BL-DELEqC-III	50	30.06	30.04	9004107	9007500	10	4.00	4.00	9225900	9225900	10
	100	29.96	29.98	9338049	9341205	10	4.00	4.00	18270900	18270900	10
	200	29.96	29.98	9338049	9341205	10	4.00	4.00	36360900	36360900	10
	500	29.96	29.98	9338049	9341205	10	4.00	4.00	90630900	90630900	10
BL-DELEqC	50	29.77	29.73	7594629	7593315	10	–	–	–	–	0
	100	29.83	29.84	7603632	7603305	10	–	–	–	–	0
	200	29.80	29.79	7602429	7603305	10	–	–	–	–	0
	500	29.80	29.79	7602429	7603305	10	3.24	3.25	90354130	90351390	9
BL-DE	50	30.00	30.00	4904400	4897770	10	–	–	–	–	0
	100	29.82	29.76	5430975	5455800	8	–	–	–	–	0
	200	30.02	29.96	6351488	6335250	8	–	–	–	–	0
	500	30.08	30.14	8769963	8622000	9	–	–	–	–	0

Table 2. Results for Problems P.3 and P.4.

Method	G_{ll}	Problem P.3					Problem P.4				
		$F(\mathbf{x}_{ul}, \mathbf{x}_{ll})$		#FELL		fr	$F(\mathbf{x}_{ul}, \mathbf{x}_{ll})$		#FELL		fr
		Mean	Median	Mean	Median		Mean	Median	Mean	Median	
BL-DELEqC-III	50	-1.46	-1.46	7689102	7688775	10	-30.91	-30.88	9046464	9046695	10
	100	-1.47	-1.47	7748733	7744755	10	-30.63	-30.57	9387645	9389040	10
	200	-1.46	-1.46	7749333	7751460	10	-30.63	-30.57	9387645	9389040	10
	500	-1.46	-1.46	7749333	7751460	10	-30.63	-30.57	9387645	9389040	10
BL-DELEqC	50	-1.45	-1.45	7134744	7133190	10	-30.13	-30.12	7645704	7641975	10
	100	-1.45	-1.45	7205708	7205940	8	-30.07	-30.09	7651236	7651020	10
	200	-1.45	-1.45	7305270	7296570	9	-30.09	-30.10	7651896	7652490	10
	500	-1.45	-1.45	7533994	7521795	8	-30.09	-30.10	7651896	7652490	10
BL-DE	50	–	–	–	–	0	–	–	–	–	0
	100	–	–	–	–	0	-18.03	-21.19	18025310	18002040	3
	200	–	–	–	–	0	-35.78	-35.04	30522942	30712680	10
	500	–	–	–	–	0	-35.43	-35.21	33079937	32559000	9

proposed BL-DELEqC-III achieved the best results in most cases. However, the number of calls to the LL objective function was not the smallest. Thus, we conclude that while the proposed method produces good results and finds feasible solutions in all independent runs using any value of LL generations, it requires more objective function evaluations at the lower level. As DELEqC-III is specifically designed to handle linear equality constraints, additional approaches are required to solve more general constrained optimization problems. We plan to investigate the combination of BL-DELEqC-III with other constraint-handling techniques.

Acknowledgements. The authors thank the support provided by CNPq, FAPEMIG, CAPES, UFJF, and RePesq.

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.

Table 3. Results for Problem P.5.

Method	G_{ll}	$F(\mathbf{x}_{ul}, \mathbf{x}_{ll})$		**#FELL**		**fr**
		Mean	**Median**	**Mean**	**Median**	
BL-DELEqC-III	50	0.00	0.00	9041361	9041430	10
	100	0.00	0.00	9385320	9383175	10
	200	0.00	0.00	9385320	9383175	10
	500	0.00	0.00	9385320	9383175	10
BL-DELEqC	50	0.00	0.00	7637856	7632345	10
	100	0.00	0.00	7645335	7648140	10
	200	0.00	0.00	7648593	7648695	10
	500	0.00	0.00	7648593	7648695	10
BL-DE	50	–	–	–	–	0
	100	11.17	8.01	18025310	18002040	3
	200	0.03	0.03	30127362	30011895	10
	500	0.03	0.03	32938393	33088440	9

References

[1] A. Sinha, P. Malo, and K. Deb. A review on bilevel optimization: From classical to evolutionary approaches and applications. *IEEE Trans. on Evolutionary Computation*, vol. 22, n. 2, pp. 276–295, 2018.
[2] E. Mezura-Montes and C. A. C. Coello. Constraint-handling in nature-inspired numerical optimization: Past, present and future. *Swarm and Evolutionary Computation*, vol. 1, n. 4, pp. 173–194, 2011.
[3] R. Datta and K. Deb, eds. *Evolutionary Constrained Optimization*. Infosys Sci. Fou. Series. Springer, 2015.
[4] S. Zhou, A. B. Zemkoho, and A. Tin. *BOLIB: Bilevel Optimization LIBrary of Test Problems*, pp. 563–580. Springer International Publishing, Cham, 2020.
[5] A. S. B. Ullah, R. Sarker, and C. Lokan. Handling equality constraints in evolutionary optimization. *European Journal of Operational Research*, vol. 221, n. 3, pp. 480 – 490, 2012.
[6] K. A. P. Lagares, J. S. Angelo, H. S. Bernardino, and H. J. C. Barbosa. A differential evolution algorithm for bilevel problems including linear equality constraints. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1885–1892, 2016.
[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] H. J. C. Barbosa, R. L. Araujo, and H. S. Bernardino. A differential evolution algorithm for optimization including linear equality constraints. In *Progress in Artificial Intelligence: 17th Portuguese Conf. on Artificial Intelligence (EPIA), Coimbra, Portugal, September 8-11*, pp. 262–273. Springer Intl. Publishing, Cham, 2015.
[9] H. J. C. Barbosa, H. S. Bernardino, and J. S. Angelo. An improved differential evolution algorithm for optimization including linear equality constraints. *Memetic Computing*, vol. 11, n. 3, pp. 317–329, 2019.
[10] J. S. Angelo, H. J. C. Barbosa, and H. S. Bernardino. Differential evolution for linear equality constraint satisfaction via unconstrained search in the null space. *Evolutionary Intelligence*, vol. 16, n. 2, pp. 565–586, 2023.
[11] R. Storn and K. V. Price. Differential Evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.
[12] K. V. Price. *An Introduction to Differential Evolution*, pp. 79–108. McGraw-Hill, 1999.
[13] S. Das, S. S. Mullick, and P. Suganthan. Recent advances in differential evolution - An updated survey. *Swarm and Evolutionary Computation*, vol. 27, pp. 1–30, 2016.
[14] H. S. Bernardino, H. J. C. Barbosa, and J. S. Angelo. Differential evolution with adaptive penalty and tournament selection for optimization including linear equality constraints. In *2018 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8, 2018.
[15] E. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Math. Programming*, vol. 91, n. 2, pp. 201–213, 2002.
[16] 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 *Congress on Evol. Computation*, pp. 1–8. IEEE, 2010.