Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 820747, 16 pages
http://dx.doi.org/10.1155/2014/820747

*Research Article*

# A Differential Evolution with Two Mutation Strategies and a Selection Based on an Improved Constraint-Handling Technique for Bilevel Programming Problems

**Hong Li and Li Zhang**

*School of Mathematics and Statistics, Xidian University, Xi'an 710071, China*

Correspondence should be addressed to Hong Li; lihong@mail.xidian.edu.cn

Received 29 April 2014; Revised 1 July 2014; Accepted 1 July 2014; Published 23 July 2014

Academic Editor: Yuping Wang

Two mutation operators are used in the differential evolution algorithm to improve the diversity of population. An improved constraint-handling technique based on a comparison mechanism is presented, and then it is combined with the selection operator in the differential evolution algorithm to fulfill constraint handling and selection simultaneously. A differential evolution with two mutation strategies and a selection based on this improved constraint-handling technique is developed to solve bilevel programming problems. The simulation results on some linear and nonlinear bilevel programming problems show the effectiveness and efficiency of the proposed algorithm.

## 1. Introduction

Bilevel programming problem (BLPP) deals with mathematical programming whose feasible set is implicitly determined by another embedded optimization problem [1]. As a hierarchical system, BLPP is widely used to lots of fields such as economy, control, engineering, and management (see [2–4]). In such a problem, there is a leader that makes the decision first and a follower that makes its decision based on the leader's. More and more practical applications can be formulated as bilevel programming models; so it is important to design some effective algorithms to solve different types of BLPPs in order to satisfy the needs of real-world problems.

Various approaches have been developed in different branches of numerical optimization to solve BLPPs under various assumptions, including extreme point methods, branch and bound algorithms, cutting plane techniques, mixed integer programming, parametric complementary pivoting methods, penalty function methods, trust region methods, and heuristic algorithms. A detailed list of references can be found in [2–4].

As it is known, BLPPs being intrinsically difficult, it is not surprising that most algorithmic research to date has focused on the simplest cases of BLPPs, that is, problems having nice properties such as linear, quadratic or convex objective, and/or constraint functions. In particular, the most studied instance of BLPPs has been for a long time the linear version [5]. Linear BLPP was shown to be a strongly NP-hard problem [6]. Furthermore, it is a NP-hard problem to find the local optimal solution of BLPP [7]. Linear BLPP has the important property that at least one optimal solution is attained at an extreme point of the constraint region. Based on this property, the $K$th-best approach was used successfully to compute global solutions of linear BLPPs by enumerating the extreme points of the constraint region [1]. A heuristic algorithm was proposed, in which classical extreme point enumeration techniques were combined with genetic algorithms by associating chromosomes with extreme points of the polyhedron [8]. Another customary way to solve linear BLPPs is to use the Karush-Kuhn-Tucker conditions to reduce the problem to a one-level complementary slackness problem. To solve this nonlinear slackness problem, many distinct approaches were proposed, for example, the branch and bound algorithms [6, 9, 10], the hybrid neural network approach [11], the penalty function method [10], the branch and cut algorithm [12], genetic algorithms [13, 14], and so forth.

Due to their inherent nonconvexity and nondifferentiability, the general nonlinear BLPPs are more intractable than linear version. Moreover, there exist some types of BLPPs which have nondifferentiable upper-level objective functions and can not be solved by traditional optimization algorithms based on gradient methods. In this case, heuristic algorithms such as evolutionary algorithm [20], differential evolution [21], and particle swarm optimization [22] are alternative methods. For solving nonlinear BLPPs, a variety of solution techniques based on heuristic algorithms can been found in the literature. For example, Liu [23] presented a genetic algorithm for Stackelberg-Nash equilibrium. Oduguwa and Roy [24] proposed a genetic algorithm for the bilevel optimization problems. Wang et al. [25] proposed an evolutionary algorithm for nonlinear BLPPs based on a new constraint-handing scheme. Li et al. [26] proposed a hierarchical particle swarm optimization for solving general BLPPs. The BLPP is transformed to solve the upper-level and lower-level problems iteratively by two variants of PSO. Wang et al. [27] developed a new evolutionary algorithm for solving a class of nonlinear BLPPs. Angelo et al. [17] proposed an algorithm which uses differential evolution to solve both the upper- and lower-level optimization problems. Sinha et al. [28] presented a standard test suite of 12 problems and have solved the proposed test problems using a nested bilevel evolutionary algorithm.

In our previous work [29], a linear BLPP is transformed into a linear single-level implicit programming problem. A differential evolution (DE) algorithm with two mutation strategies and a selection based on an improved constraint-handling technique is developed to solve the linear implicit programming problem. Preliminary results were obtained for linear BLPPs, but this algorithm is considered to have a general applicability. In this paper, this algorithm is extended from linear version of BLPPs to nonlinear bilevel programming problems. Two types of bilevel problems including linear and nonlinear versions are used to test the effectiveness of the proposed algorithm. The results obtained are compared with other available algorithms in the literature.

This paper is organized as follows. The general BLPP is described in the next section. A DE algorithm with two mutation strategies and a selection based on an improved constraint-handling method is presented in Section 3. Computational results and comparison are provided in Section 4. Section 5 concludes this paper.

## 2. Statement of General BLPP

When two independent decision makers, the leader and follower, have conflicting objectives, this bilevel programming model arises. Without loss of generality, we consider the minimization problems in our discussion. The general BLPP under discussion is expressed as

$$\begin{aligned} \min_{x\in X} \quad & F(x,y) \\ \text{s.t.} \quad & G(x,y) \le 0 \\ \min_{y\in Y} \quad & f(x,y) \\ \text{s.t.} \quad & g(x,y) \le 0, \end{aligned} \tag{1}$$

where $X \subset R^n$ and $Y \subset R^m$ are box constraint sets of leader's variable $x$ and follower's variable $y$, respectively. $F : R^{n\times m} \to R$, $f : R^{n\times m} \to R$, $G : R^{n\times m} \to R^p$, and $g : R^{n\times m} \to R^q$.

The constraint region of problem (1) is denoted by $\Omega$:

$$\Omega = \{(x,y) \in X \times Y : G(x,y) \le 0,\ g(x,y) \le 0\}. \tag{2}$$

Denote the projection of $\Omega$ onto the leader's decision space by

$$I = \{x : \exists y, \text{ such that } (x,y) \in \Omega\}. \tag{3}$$

The follower's rational reaction set for $x \in I$ is defined as

$$R(x) = \{y : y \in \arg\min\{f(x,\overline{y}) : g(x,\overline{y}) \le 0\}\}. \tag{4}$$

Denote the inducible region (feasible set) by

$$IR = \{(x,y) : (x,y) \in \Omega,\ y \in R(x)\}. \tag{5}$$

The main feature of BLPP is that it involves two mathematical programming, the solution of one being part of the constraints of the other. That is to say, the follower's solution depends implicitly on the leader's variable; in return, the leader's objective optimal value is restricted by the follower's solution. So it is quite evident that problem (1) can be converted into the following equivalent single-level implicit programming problem:

$$\begin{aligned} \min_{x\in X} \quad & F(x,y(x)) \\ \text{s.t.} \quad & G(x,y(x)) \le 0 \\ & y(x) = \arg\min_{\overline{y}\in Y}\{f(x,\overline{y}) : g(x,\overline{y}) \le 0\}. \end{aligned} \tag{6}$$

It is also rewritten as follows:

$$\begin{aligned} \min_{x\in X} \quad & F(x,y(x)) \\ \text{s.t.} \quad & y(x) \in R(x), \quad x \in I. \end{aligned} \tag{7}$$

We suppose that $F(x,y)$ and $G(x,y)$ may be continuous, yet nonconvex, even nondifferentiable, and $f(x,y)$ and $g(x,y)$ are differentiable and convex in $y$ for $x$ fixed. Under these assumptions, the constraint region $\Omega$ of problem (1) is bounded and closed set.

If the lower-level problem is linear programming, we can use available linear programming solver to obtain $y(x)$ for the given $x$; if the lower-level problem is quadratic programming, we can use available quadratic programming solver to obtain $y(x)$ for the given $x$; for the generalized lower-level problem, we must turn to some available effective algorithms for solving this constrained optimization problem, such as trust region algorithm, interior algorithm, and active-set algorithm. In this paper, we adopt the optimization toolbox of MATLAB to solve lower-level problems. Certainly, the lower-level problem can be solved by evolutionary algorithms, but it is over time-consuming.

## 3. The Proposed Differential Evolution Algorithm

Differential evolution (DE), introduced by [21], is one of the most prominent evolutionary algorithms (EAs) for solving real-valued optimization problems. DE has exclusive advantages, such as a simple and easy-to-understand concept, ease of implementation, powerful search capability, and robustness. Using only a few control parameters, DE has gradually become more popular and has been used in many practical cases [33].

*3.1. Individual Coding.* For solving implicit programming problem (7), a vector $x = (x_1, x_2, \ldots, x_n)$ is used to express an individual, which represents a decision variable of leader, where $x_i \in [lb_i, ub_i]$, $i = 1, 2, \ldots, n$.

If the bounds of each $x_i$ $(i = 1, 2, \ldots, n)$ of upper-level variable $x$ are not given, the following method is used to establish the bounds of upper-level variable $x$; that is, $lb_i \le x_i \le ub_i$, $i = 1, 2, \ldots, n$:

$$\begin{aligned} lb_i &= \min\{x_i : (x, y) \in \Omega\}, \\ ub_i &= \max\{x_i : (x, y) \in \Omega\}. \end{aligned} \tag{8}$$

*3.2. Fitness Function.* The fitness function $\text{fit}(x)$ is expressed as follows:

$$\text{fit}(x) = \begin{cases} F(x, y(x)), & \text{if } y(x) \text{ exists} \\ +\infty, & \text{otherwise.} \end{cases} \tag{9}$$

*3.3. A Constraint-Handling Technique Based on a Comparison Mechanism.* A constraint-handling technique is developed by improving Deb's feasibility-based comparison method [34], which is based on the following criteria.

(i) Between two feasible solutions, the one with the lowest objective function value wins for minimization problem.

(ii) If one solution is feasible and the other one is infeasible, the feasible solution wins.

(iii) If both solutions are infeasible, the one with the lowest sum of constraint violations is preferred.

There are two changes in our constraint-handling method. (1) Sum of constraint violations is replaced by maximum of constraint violations. (2) The infeasible individuals in certain range (close to the boundary of the feasible region) with the good values of the objective function are selected for the next generation.

This improved constraint-handling technique is presented detailedly as follows.

*Definition 1.* An individual $x$ is called a feasible individual, if and only if $(x, y) \in IR$.

*Definition 2.* For a given $x$, obtain $y \in R(x)$. Let $\overline{G}_i(x) = \max\{G_i(x, y), 0\}$ for $i = 1, \ldots, p$, $\overline{g}_j(x) = \max\{g_j(x, y), 0\}$ for $j = 1, \ldots, q$. $C(x) = \max\{\overline{G}_i(x), \overline{g}_j(x)$ for all $i, j\}$ is called constraint violation of an individual $x$.

*Definition 3.* An individual $x^1$ is better than another individual $x^2$, if one of the following criteria is satisfied.

(1) If $x^1$ and $x^2$ are two feasible individuals, that is, $C(x^1) = 0$ and $C(x^2) = 0$, as well as $\text{fit}(x^1) < \text{fit}(x^2)$.

(2) If both individuals $x^1$ and $x^2$ are infeasible, that is, $C(x^1) > 0$ and $C(x^2) > 0$, as well as $C(x^1) < C(x^2)$.

(3) If one individual is feasible and another one is infeasible, the following cases are considered.

   (a) If $0 < C(x^1) \le \epsilon(T)$ and $C(x^2) = 0$, as well as $\text{fit}(x^1) < \text{fit}(x^2)$;

   (b) if $C(x^1) = 0$ and $0 < C(x^2) \le \epsilon(T)$, as well as $\text{fit}(x^1) < \text{fit}(x^2)$;

   (c) if $C(x^1) = 0$ and $C(x^2) > \epsilon(T)$,

where $\epsilon(T)$ is a given small positive parameter related to the evolutionary generation $T$. In this paper, for a given maximum evolutionary generation $\text{Max}\, T$, the values of $\epsilon(T)$ are listed as follows:

$$\epsilon(T) = \begin{cases} 0.1, & \text{if } T < \dfrac{\text{Max}\, T}{4} \\ 0.01, & \text{if } \dfrac{\text{Max}\, T}{4} \le T < \dfrac{\text{Max}\, T}{3} \\ 0.001, & \text{if } \dfrac{\text{Max}\, T}{3} \le T < \dfrac{\text{Max}\, T}{2} \\ 0.0001, & \text{if } T \ge \dfrac{\text{Max}\, T}{2}. \end{cases} \tag{10}$$

*Definition 4.* The best individual in the current generation is determined by the constraint violations and fitness values of all individuals in the current population according to the following rules.

(1) If the constraint violations of all individuals are less than 0.0001, one individual with the lowest fitness value is the best individual in the current generation.

(2) If the constraint violations of all individuals are greater than 0.0001, one individual with the lowest constraint violation is the best individual in the current generation.

(3) If there is only part of all individuals whose constraint violations are less than 0.0001, one with the lowest fitness value in this part is the best individual in the current generation.

*3.4. Hybrid Mutation Operation.* Two mutation operators are adopted to generate two subpopulations, each of whom comprises NP/2 individuals (NP denotes population size and is even); then combine them together to form a population with NP individuals.

According to Definition 4, determine the best individual in the $T$th generation denoted by $x^b(T) = (x_1^b(T), x_2^b(T), \ldots, x_n^b(T))$.

The first subpopulation with NP/2 individuals is generated by the following mutation operator:

$$\begin{aligned} v^t(T) = {} & x^{r_1}(T) + \text{SF} \cdot \left(x^b(T) - x^{r_1}(T)\right) \\ & + \text{SF} \cdot \left(x^{r_2}(T) - x^{r_3}(T)\right), \end{aligned} \tag{11}$$

where $t = 1, 2, \ldots, \mathrm{NP}/2$, $r_1$, $r_2$, and $r_3$ are mutually distinct indexes in $[1, \mathrm{NP}]$ not equal to $t$. $\mathrm{SF} \in (0, 1)$ denotes scaling factor.

The second subpopulation with NP/2 individuals is generated by the following mutation operator:

$$v^t(T) = x^{r_1}(T) + \mathrm{SF} \cdot \left(x^{r_2}(T) - x^{r_3}(T)\right), \tag{12}$$

where $t = \mathrm{NP}/2 + 1, \mathrm{NP}/2 + 2, \ldots, \mathrm{NP}$, $r_1$, $r_2$, and $r_3$ are mutually distinct indexes in $[1, \mathrm{NP}]$ not equal to $t$. $\mathrm{SF} \in (0, 1)$ is scaling factor.

If a mutant individual $v^t(T) = (v_1^t(T), v_2^t(T), \ldots, v_n^t(T))$, $t = 1, 2, \ldots, \mathrm{NP}$, is not in box constraints, that is, $v_i^t(T) \notin [lb_i, ub_i]$, $i = 1, 2, \ldots, n$, then randomly select a real number in $[lb_i, ub_i]$ to substitute it.

*3.5. Crossover Operation.* For each pair of individuals constituted by the parent individual $x^t(T) = (x_1^t(T), \ldots, x_k^t(T), \ldots, x_n^t(T))$ and mutation individual $v^t(T) = (v_1^t(T), \ldots, v_k^t(T), \ldots, v_n^t(T))$, the crossover individual $u^t(T) = (u_1^t(T), \ldots, u_k^t(T), \ldots, u_n^t(T))$ is generated by the following strategy:

$$u_k^t(T) = \begin{cases} v_k^t(T), & \text{if } R_k \le \mathrm{CR} \text{ or } k = rn(t) \\ x_k^t(T), & \text{otherwise,} \end{cases} \tag{13}$$

where $R_k \in (0, 1)$ is a uniform random number, $rn(t) \in \{1, 2, \ldots, n\}$ is the randomly selected index chosen once for each $t$, and CR is a real-valued crossover rate constant in the range $(0, 1)$.

*3.6. Selection Operation Based on the Constraint-Handling Technique.* The above-mentioned constraint-handling technique is introduced to the selection operator to decide the population for the next generation.

For $t = 1, 2, \ldots, \mathrm{NP}$, the trial vector $u^t(T)$ and the parent vector $x^t(T)$ are compared. The individual of the next generation $x^t(T + 1)$ is decided according to Definition 3:

$$x^t(T + 1) = \begin{cases} u^t(T), & \text{if } u^t(T) \text{ is better than } x^t(T) \\ x^t(T), & \text{otherwise.} \end{cases} \tag{14}$$

*3.7. The Proposed Algorithm.* Now we present the differential evolution algorithm with two mutation strategies and a selection based on an improved constraint-handling technique as follows.

*Step 1* (parameter setting). Set population size NP, scaling factor SF, crossover rate CR, and maximum number of generations Max $T$.

*Step 2* (initialization). Let $T = 0$. Generate randomly an initial population $\{x^t(T) = (x_1^t(T), x_2^t(T), \ldots, x_n^t(T)) : x_i^t(T) \in [lb_i, ub_i],\ i = 1, 2, \ldots, n,\ t = 1, 2, \ldots, \mathrm{NP}\}$. For each individual $x^t(T)$, obtain lower-level optimal solution by available optimization solver. Evaluate the constraint violations and fitness values of all individuals and determine the best individual according to Definition 4.

*Step 3* (mutation operation). For $t = 1$ to NP/2, select randomly $r_1 \ne r_2 \ne r_3 \ne t$ and $r_1, r_2, r_3 \in [1, \mathrm{NP}]$. The first subpopulation is generated according to (11).

For $t = \mathrm{NP}/2 + 1$ to NP, select randomly $r_1 \ne r_2 \ne r_3 \ne t$ and $r_1, r_2, r_3 \in [1, \mathrm{NP}]$. The second subpopulation is generated according to (12).

*Step 4* (crossover operation). Crossover individuals are produced by (13). For each crossover individual $u^t(T)$, obtain lower-level optimal solution by using available optimization solver. Evaluate the constraint violations and fitness values of all crossover individuals and determine the best individual according to Definition 4.

*Step 5* (selection operation). According to selection rule, form the next population.

*Step 6* (stopping criterion). If the stopping criterion is met, then stop and output the result. Otherwise, set $T = T + 1$ and go to Step 3.

## 4. Computational Results and Comparison

To evaluate the performance of the proposed algorithm, we test two types of benchmark problems: one type is linear BLPP and another type is nonlinear BLPP, which includes linear-quadratic programming, quadratic-quadratic programming, linear-linear fractional programming, general nonlinear BLPP, and complex nonlinear BLPP with nondifferentiable leader's objective function. The results of the proposed algorithm are compared with other algorithms in the literature.

During the simulations, we adopt the following parameter suite:

(i) population size: $\mathrm{NP} = 40$;

(ii) scaling factor: $\mathrm{SF} = 0.5 + \mathrm{rand}(0, 1) \cdot 0.3$, where $\mathrm{rand}(0, 1)$ is a random number in the range $(0, 1)$;

(iii) crossover rate $\mathrm{CR} = 0.9$;

(iv) maximum number of generations $\mathrm{Max}\, T = 500$.

All experiments are performed on a notebook computer with 2.53 GHz Dual-core Processor and 1.94 GB of RAM in MATLAB software. Each test problem implemented 50 independent runs. For solving the lower-level optimization problems, we use corresponding function calls in the optimization toolbox of MATLAB to solve linear programming, quadratic programming, or other nonlinear programming.

Performance of the proposed algorithm is evaluated by the following criteria:

(i) success rate (SR), that is, the percentage of convergence to the optimal solution in 50 independent runs, which can be used to evaluate the reliability of the proposed algorithm;

(ii) mean number of fitness evaluations (MNFE) when the optimal solution is reached first in 50 independent runs, which is assessed the computational cost;

Table 1: Comparative results of two constraint-handling methods for linear BLPPs over 50 independent runs.

| Problem | The improved method | | | Deb's method | | |
|---|---|---|---|---|---|---|
| | SR (%) | MNFE | SD | SR (%) | MNFE | SD |
| A.1 | 100 | 470 | 0.00 | 100 | 650 | 0.00 |
| A.2 | 100 | 638 | 0.00 | 100 | 758 | 0.00 |
| A.3 | 100 | 755 | 0.00 | 100 | 894 | 0.00 |
| A.4 | 100 | 3,684 | 0.00 | 100 | 3,457 | 0.00 |
| A.5 | 100 | 266 | 0.00 | 100 | 266 | 0.00 |
| A.6 | 100 | 169 | 0.00 | 100 | 146 | 0.00 |
| A.7 | 98 | 2,652 | 0.09 | 86 | 2,981 | 0.73 |
| A.8 | 100 | 5,070 | 0.00 | 100 | 5,047 | 0.00 |
| A.9 | 98 | 17,614 | 1.93 | 96 | 19,685 | 4.48 |

(iii) standard deviation of fitness values (SD), which is used to measure the quality of the solution obtained;

(iv) best values of upper-level objective functions $F(x^*, y^*)$ and lower-level objective functions $f(x^*, y^*)$, as well as best solution obtained $(x^*, y^*)$.

In order to demonstrate the effectiveness of the improved constraint-handling technique, we compare it with Deb's feasibility-based method [34] in the same framework of algorithm and same parameters. Also, we compare the obtained results with those presented in the corresponding references.

*4.1. Computational Results for Linear BLPPs.* To examine the effectiveness of the proposed algorithm for linear BLPPs, we first test 9 linear bilevel programming problems and compare the results of the proposed algorithm with other algorithms in the literature. The linear test problems are provided in Appendix A.

The comparative results obtained by two constraint-handling methods are shown in Table 1. As shown in Table 1, the improved constraint-handling technique is slightly better than Deb's method in terms of SR, MNFE, and SD. Furthermore, the proposed algorithm has a good performance according to stability, efficiency, and quality of solution obtained. Therefore it can be concluded that the improved constraint-handling technique is effective.

Table 2 presents the best results obtained by the proposed algorithm for linear BLPPs, including best solution, upper-level and lower-level objective function values.

In order to demonstrate the improvement in algorithm performance by the use of hybrid mutation operator, we compare hybrid mutation operator with single mutation operator in the same framework of algorithm and same parameters. The comparative results are shown in Table 3. From this table, it can be seen that the reliability of algorithm is enhanced by using hybrid mutation operator in terms of SR. Although hybrid mutation operator requires a larger computational cost than single mutation operator (11) except for Problems A.4 and A.7, the stability of single mutation operator (11) is worse than that of hybrid mutation operator according to SR and SD. The computational cost of single mutation operator (12) is larger than that of hybrid mutation operator, but the quality of solution obtained is better than hybrid mutation operator for Problem A.7 in terms of SR and SD. Therefore the algorithmic performance is improved by the use of those two mutation operators.

The proposed algorithm is compared with the algorithms in the literature according to the best values of upper-level and lower-level objective functions. The comparative results are shown in Table 4. From this table, the proposed algorithm exhibits the better results than other algorithms. For Problems A.1, A.8, and A.9, the best results obtained by the proposed algorithm are better than those of other algorithms in the literature. For remaining problems, they have same optimization results.

*4.2. Computational Results for Nonlinear BLPPs.* To validate the effectiveness of the proposed algorithm for nonlinear BLPPs, we test two sets of nonlinear bilevel programming problems. The first set has 10 examples selected from different sources of the literature, and the second set consists in solving the test collection (SMD1 to SMD12) proposed in [28]. Finally, we compare the results of the proposed algorithm with the other algorithms in the literature. The first set of nonlinear BLPPs is provided in Appendix B, and the second set of nonlinear BLPPs can be obtained in [28].

With same algorithm's framework and parameter setting, the improved constraint-handling method is compared with Deb's feasibility-based method [34] for first set of nonlinear BLPPs to illuminate the effectiveness of the improved constraint-handling method again. Comparative results of two constraint-handling methods are shown in Table 5. From this table, it can be seen that the improved constraint-handling method is better than the Deb's method in terms of success rate, mean number of fitness evaluations, and standard deviation of fitness values. Moreover, this table shows that the proposed algorithm with the improved constraint-handling method is effective and can find the global optimal solutions with a 100% success rate, except for Problem B.4 with a 76% success rate.

In order to display the computational results obtained by the proposed algorithm for first set of nonlinear BLPPs, the best solution $(x^*, y^*)$, the upper-level objective value $F(x^*, y^*)$, and the lower-level objective value $f(x^*, y^*)$ for all problems are provided in Table 6. This table shows that the best solutions obtained by the proposed algorithm are same as the global optimal solutions of all problems.

Table 7 illustrates the comparative results of the proposed algorithm with available algorithms in the literature for the first set of nonlinear BLPPs. As shown in Table 7, the proposed algorithm is able to find the global optimal solutions for the first set of nonlinear BLPPs. Compared against the other algorithms in the literature, the proposed algorithm can obtain equal optimal results for all problems.

In order to investigate further the performance of the proposed algorithm, the second set of nonlinear BLPPs is tested, which is a complex test collection with different dimensions in [28]. We performed 50 runs for each of the test problems with 5 and 10 dimensions and 11 runs for each of the test problems with 20 dimensions. In the case with five dimensions, $p = 1$, $q = 2$, and $r = 1$ are used in SMD1 to

Table 2: Best results obtained by the proposed algorithm for linear BLPPs.

| Problem | $(x^*, y^*)$ | $F(x^*, y^*)$ | $f(x^*, y^*)$ |
|---|---|---|---|
| A.1 | (19.00, 14.00) | −37.00 | 14.00 |
| A.2 | (16.00, 11.00) | −49.00 | 17.00 |
| A.3 | (17.45, 10.91) | −85.09 | 50.18 |
| A.4 | (0.00, 0.90, 0, 0.60, 0.40) | −29.20 | 3.20 |
| A.5 | (1.00, 9.00, 0.00) | −19.00 | −9.00 |
| A.6 | (2.00, 0.00, 1.50, 0.00) | −3.25 | −6.00 |
| A.7 | (0.50, 0.80, 0, 0.20, 0.80) | −18.40 | 1.80 |
| A.8 | (1.55, 0.78, 0.16, 2.21, 1.89, 0) | 14.99 | −16.99 |
| A.9 | (0.00, 2.44, 10.00, 0.00, 10.00, 8.74, 5.25, 10.00, 0.00, 10.00, 3.73, 10.00, 10.00, 10.00, 0.00, 0.00) | −453.61 | −68.81 |

Table 3: Comparative results of hybrid mutation operator and single mutation operator for linear BLPPs over 50 independent runs.

| Problem | Hybrid mutation operator | | | Mutation operator (11) | | | Mutation operator (12) | | |
|---|---|---|---|---|---|---|---|---|---|
| | SR (%) | MNFE | SD | SR (%) | MNFE | SD | SR (%) | MNFE | SD |
| A.1 | 100 | 470 | 0.00 | 100 | 438 | 0.00 | 100 | 593 | 0.00 |
| A.2 | 100 | 638 | 0.00 | 100 | 529 | 0.00 | 100 | 710 | 0.00 |
| A.3 | 100 | 755 | 0.00 | 100 | 703 | 0.00 | 100 | 966 | 0.00 |
| A.4 | 100 | 3,684 | 0.00 | 100 | 3,777 | 0.00 | 52 | 7,056 | 0.01 |
| A.5 | 100 | 266 | 0.00 | 100 | 242 | 0.00 | 100 | 314 | 0.00 |
| A.6 | 100 | 169 | 0.00 | 100 | 129 | 0.00 | 100 | 217 | 0.00 |
| A.7 | 98 | 2,652 | 0.09 | 70 | 4,068 | 1.11 | 100 | 3,454 | 0.00 |
| A.8 | 100 | 5,070 | 0.00 | 100 | 3,532 | 0.00 | 64 | 8,008 | 0.01 |
| A.9 | 98 | 17,614 | 1.93 | 86 | 13,678 | 4.38 | 90 | 20,040 | 0.14 |

SMD5 and SMD7 to SMD12, and $p = 1$, $q = 0$, $r = 1$, and $s = 2$ are used in SMD6. In the case with ten dimensions, $p = 3$, $q = 3$, and $r = 2$ are used in SMD1 to SMD5 and SMD7 to SMD12, and $p = 3$, $q = 1$, $r = 2$, and $s = 2$ are used in SMD6. In the case with twenty dimensions, $p = 5$, $q = 5$, and $r = 5$ are used in SMD1 to SMD5 and SMD7 to SMD12, and $p = 5$, $q = 3$, $r = 5$, and $s = 2$ are used in SMD6, where $p$, $q$, $r$, and $s$ are parameters of test problems in [28]. Abovementioned parameters of the proposed algorithm continue to be used here. When the accuracy of best upper-level function value becomes less than $10^{-4}$, the algorithm terminates.

The statistical results for 5-, 10-, and 20-dimensional test problems of second set are provided in Table 8, including success rate (SR), mean number of fitness evaluations (MNFE), and standard deviation of fitness values (SD). This table shows that the proposed algorithm can locate the optimal solution for 5- and 10-dimensional test problems. Except for SMD10–SMD12 with 20 dimensions, it can also obtain the optimal solution for other 20-dimensional test problems. The proposed algorithm fails to find the optimal solution within a given accuracy (less than $10^{-4}$) for SMD10-SMD12 with 20 dimensions.

Table 9 presents the best results obtained by the proposed algorithm for 5- and 10-dimensional test problems of second set, including the best solution $(x^*, y^*)$, the best upper-level objective value $F(x^*, y^*)$, and the best lower-level objective value $f(x^*, y^*)$.

The comparison of median accuracy achieved by the proposed algorithm and those reported in [28] is provided in Table 10. For 20-dimensional test problems, the accuracy for the upper level (UL) and lower level (LL) is not given in [28]. From this table, it is obvious that the proposed algorithm is better than the nested bilevel evolutionary algorithm in [28] according to median value of the accuracy at the upper level and lower level for 5- and 10-dimensional test problems, except for SMD5. Furthermore, the proposed algorithm can obtain the good accuracy for 20-dimensional test problems except for SMD10–SMD12. The nested bilevel evolutionary algorithm in [28] cannot find feasible solutions for 10-dimensional SMD9–SMD12, which are constrained problems, and 20-dimensional SMD7–SMD12. This also shows that the improved constraint-handling method in this paper performs better than the constraint handling in [28], which belongs to Deb's method [34].

Tables 11, 12, and 13 provide the best, median, and worst numbers of function evaluations at upper and lower levels for SMD1–SMD12 with 5, 10, and 20 dimensions, respectively. In these tables, we also compare the proposed differential evolution algorithm (DE) with the nested bilevel evolutionary algorithm (NBEA) in [28] for SMD1–SMD12 with 5, 10, and 20 dimensions, respectively.

From Table 11, it can be seen that for 5-dimensional test problems, DE has smaller "best FE" for the upper level than NBEA for 8 out of 12 test problems (SMD2–SMD4, SMD6–SMD9, and SMD11) and smaller "worst FE" for the upper level than NBEA for 6 out of 12 test problems (SMD1, SMD2, SMD5, SMD7, SMD8, and SMD11). But DE has larger "median FE" for the upper level than NBEA except for SMD4,

Table 4: Comparison of the proposed algorithm with the algorithms in the literature for linear BLPPs.

| Problem/literature | The result by the proposed algorithm | | The result in the literature | |
|---|---|---|---|---|
| | $F(x^*, y^*)$ | $f(x^*, y^*)$ | $F(x^*, y^*)$ | $f(x^*, y^*)$ |
| A.1/[15] | −37.00 | 14.00 | −36.88 | 13.95 |
| A.2/[16] | −49.00 | 17.00 | −49 | 17 |
| A.3/[11] | −85.09 | 50.18 | −85.09 | 50.17 |
| A.4/[17] | −29.20 | 3.20 | −29.2 | 3.2 |
| A.5/[18] | −19.00 | −9.00 | −19 | −9 |
| A.6/[15] | −3.25 | −6.00 | −3.25 | −6 |
| A.7/[6] | −18.40 | 1.80 | −18.4 | 1.8 |
| A.8/[19] | 14.99 | −16.99 | 15.40 | −16.99 |
| A.9/[15] | −453.61 | −68.81 | −447.46 | −11.62 |

Table 5: Comparative results of two constraint-handling methods for first set of nonlinear BLPPs over 50 independent runs.

| Problem | The improved method | | | Deb's method | | |
|---|---|---|---|---|---|---|
| | SR (%) | MNFE | SD | SR (%) | MNFE | SD |
| B.1 | 100 | 1,761 | 0.00 | 98 | 2,107 | 0.71 |
| B.2 | 100 | 1,966 | 0.00 | 100 | 1,610 | 0.00 |
| B.3 | 100 | 3,923 | 0.00 | 100 | 7,085 | 0.00 |
| B.4 | 76 | 7,363 | 2.23 | 66 | 8,903 | 2.50 |
| B.5 | 100 | 379 | 0.00 | 100 | 379 | 0.00 |
| B.6 | 100 | 2,793 | 0.00 | 100 | 2,726 | 0.00 |
| B.7 | 100 | 1,761 | 0.00 | 98 | 2,107 | 0.71 |
| B.8 | 100 | 2,269 | 0.00 | 100 | 2,448 | 0.00 |
| B.9 | 100 | 404 | 0.00 | 100 | 432 | 0.00 |
| B.10 | 100 | 2,748 | 0.00 | 100 | 2,875 | 0.00 |

SMD5, and SMD11. It notes that DE has much smaller FE for lower level than NBEA in terms of best FE, median FE, and worst FE.

From Table 12, it is obvious that for 10-dimensional test problems, DE requires smaller "best FE" for the upper level than NBEA for 4 out of 8 test problems (SMD2, SMD4, SMD5, and SMD7), smaller "median FE" for the upper level than NBEA for 7 out of 8 test problems (SMD1–SMD3 and SMD5–SMD8), and smaller "worst FE" for the upper level than NBEA for 6 out of 8 test problems (SMD1–SMD3, SMD5, SMD6, and SMD8). DE has much smaller FE for lower level than NBEA in terms of best FE, median FE, and worst FE for SMD1–SMD8. In addition, NBEA cannot obtain a feasible solution for each of 10-dimensional SMD9–SMD12.

From Table 13, it can be seen that for 20-dimensional test problems, DE has smaller "median FE" for the upper level than NBEA for 5 out of 6 test problems (SMD1–SMD3, SMD5, and SMD6), and smaller "worst FE" for the upper level than NBEA for 6 test problems (SMD1–SMD6). But DE has larger "best FE" for upper level than NBEA except for SMD2 and SMD5. DE has much smaller FE for lower level than NBEA in terms of best FE, median FE, and worst FE for SMD1–SMD6. In addition, NBEA cannot obtain a feasible solution for each of 20-dimensional SMD7–SMD12.

From Tables 11, 12, and 13, we can conclude that it is expensive that the lower-level programming is solved by evolutionary algorithms.

## 5. Conclusion

This paper presents a differential evolution with two mutation strategies and a selection operator based on an improved constraint-handling technique for BLPPs. This algorithm avoids the use of penalty function to deal with the constraints, because the right penalty parameter is hard to select. Especially, this algorithm can solve general nonlinear BLPP which may have a nondifferentiable upper-level objective function. The BLPP's natural structure is considered in this paper to develop algorithm in order to solve a wide class of BLPPs with weak features. The numerical results on linear and nonlinear BLPPs show the proposed algorithm is effective and robust.

## Appendices

## A. Linear Test Problems

*Problem A.1* (see [15]). Consider

$$\begin{aligned} \min_{x,y} \quad & F(x,y) = x - 4y \\ \text{s.t.} \quad & x \geq 0 \\ \min_{y} \quad & f(x,y) = y \\ \text{s.t.} \quad & -2x + y \leq 0, \\ & 2x + 5y \leq 108, \\ & 2x - 3y \leq -4, \\ & y \geq 0. \end{aligned} \tag{A.1}$$

*Problem A.2* (see [16]). Consider

$$\begin{aligned} \min_{x,y} \quad & F(x,y) = -x - 3y \\ \min_{y} \quad & f(x,y) = -x + 3y \end{aligned}$$

Table 6: Best results obtained by the proposed algorithm for first set of nonlinear BLPPs.

| Problem | $(x^*, y^*)$ | $F(x^*, y^*)$ | $f(x^*, y^*)$ |
|---|---|---|---|
| B.1 | (0.00, 30.00, −10.00, 10.00) | 0.00 | 100.00 |
| B.2 | (0.00, 2.00, 1.88, 0.91) | −12.68 | −1.01 |
| B.3 | (7.02, 3.03, 11.98, 17.97, 0.05, 10.00, 29.95, 0) | 6600.00 | 24.58, 29.54 |
| B.4 | (1.95, 8.05, 0.00, 0.97, 0.97, 1.31, 6.74, 0, 0) | 9.56 | 1.61, 7.09, 0 |
| B.5 | (7.07, 7.07, 7.07, 7.07) | 1.98 | −1.98 |
| B.6 | (0.00, 0.90, 0.00, 0.60, 0.40) | −29.20 | 0.31 |
| B.7 | (0.00, 30.00, −10.00, 10.00) | 0.00 | 100.00 |
| B.8 | (20.00, 5.00, 10.00, 5.00) | 0.00 | 100.00 |
| B.9 | (1.89, 0.89, 0.00) | 0.00 | 7.61 |
| B.10 | (0.00, 0.90, 0.00, 0.60, 0.40) | 0.00 | 0.31 |

Table 7: Comparison of the proposed algorithm with the algorithms in the literature for the first set of nonlinear BLPPs.

| Problem/literature | The result by the proposed algorithm | | The result in the literature | |
|---|---|---|---|---|
| | $F(x^*, y^*)$ | $f(x^*, y^*)$ | $F(x^*, y^*)$ | $f(x^*, y^*)$ |
| B.1/[25] | 0.00 | 100.00 | 0 | 100 |
| B.2/[30] | −12.68 | −1.01 | −12.68 | −1.02 |
| B.3/[31] | 6600.00 | 24.58, 29.54 | 6600 | 11.92, 64.68 |
| B.4/[23] | 9.56 | 1.61, 7.09, 0 | 9.57 | 1.61, 7.10, 0 |
| B.5/[23] | 1.98 | −1.98 | 1.98 | −1.95 |
| B.6/[32] | −29.20 | 0.31 | −29.20 | 0.31 |
| B.7/[25] | 0.00 | 100.00 | 0 | 100 |
| B.8/[25] | 0.00 | 100.00 | 0 | 100 |
| B.9/[25] | 0.00 | 7.61 | 0.00 | 7.62 |
| B.10/[25] | 0.00 | 0.31 | 0.00 | 0.31 |

$$\begin{aligned}
\text{s.t.}\quad & -x-2y \le -10,\\
& x-2y \le 6,\\
& 2x-y \le 21,\\
& x+2y \le 38,\\
& -x+2y \le 18,\\
& x \ge 0,\quad y \ge 0.
\end{aligned} \tag{A.2}$$

*Problem A.3* (see [11]). Consider

$$\begin{aligned}
\min_{x,y}\quad & F(x,y) = 2x-11y\\
\min_{y}\quad & f(x,y) = x+3y\\
\text{s.t.}\quad & x-2y \le 4,\\
& 2x-y \le 24,\\
& 3x+4y \le 96,\\
& x+7y \le 126,\\
& -4x+5y \le 65,\\
& -x-4y \le -8,\\
& x \ge 0,\quad y \ge 0.
\end{aligned} \tag{A.3}$$

*Problem A.4* (see [1]). Consider

$$\begin{aligned}
\min_{x,y}\quad & F(x,y) = -8x_1-4x_2+4y_1-40y_2-4y_3\\
\text{s.t.}\quad & x_1 \ge 0,\quad x_2 \ge 0\\
\min_{y}\quad & f(x,y) = x_1+2x_2+y_1+y_2+2y_3\\
\text{s.t.}\quad & -y_1+y_2+y_3 \le 1,\\
& 2x_1-y_1+2y_2-0.5y_3 \le 1,\\
& 2x_2+2y_1-y_2-0.5y_3 \le 1,\\
& y_1 \ge 0,\quad y_2 \ge 0,\quad y_3 \ge 0.
\end{aligned} \tag{A.4}$$

*Problem A.5* (see [18]). Consider

$$\begin{aligned}
\min_{x,y}\quad & F(x,y) = -x-2y_1-3y_2\\
\text{s.t.}\quad & 0 \le x \le 8
\end{aligned}$$

Table 8: Statistical results obtained by the proposed algorithm for second set of nonlinear BLPPs.

| Problem | 5 dimensions | | | 10 dimensions | | | 20 dimensions | | |
|---|---|---|---|---|---|---|---|---|---|
| | SR (%) | MNFE | SD | SR (%) | MNFE | SD | SR (%) | MNFE | SD |
| SMD1 | 100 | 750 | $2.45e-5$ | 100 | 2,069 | $2.24e-5$ | 100 | 3,996 | $2.02e-5$ |
| SMD2 | 100 | 673 | $2.90e-5$ | 100 | 1,635 | $2.72e-5$ | 100 | 3,756 | $1.49e-5$ |
| SMD3 | 100 | 748 | $2.86e-5$ | 100 | 2,086 | $2.43e-5$ | 100 | 3,920 | $2.30e-5$ |
| SMD4 | 100 | 606 | $2.93e-5$ | 100 | 1,554 | $2.48e-5$ | 100 | 3,436 | $1.32e-5$ |
| SMD5 | 100 | 712 | $3.17e-5$ | 100 | 1,953 | $2.51e-5$ | 100 | 3,996 | $2.02e-5$ |
| SMD6 | 100 | 750 | $2.46e-5$ | 100 | 2,069 | $2.24e-5$ | 100 | 4,091 | $2.17e-5$ |
| SMD7 | 100 | 739 | $2.85e-5$ | 100 | 2,162 | $2.18e-5$ | 82 | 5,175 | $3.50e-2$ |
| SMD8 | 100 | 1,181 | $2.81e-5$ | 100 | 3,447 | $2.07e-5$ | 100 | 6,024 | $1.12e-5$ |
| SMD9 | 100 | 838 | $2.96e-5$ | 100 | 2,432 | $2.50e-5$ | 100 | 6,604 | $1.96e-5$ |
| SMD10 | 92 | 2,513 | $4.38e+0$ | 100 | 6,680 | $5.26e-5$ | — | — | — |
| SMD11 | 100 | 771 | $2.40e-5$ | 100 | 10,429 | $2.88e-4$ | — | — | — |
| SMD12 | 80 | 2,951 | $1.59e+1$ | 100 | 7,625 | $4.75e-5$ | — | — | — |

$$\begin{aligned}\min_{y}\quad & f(x,y) = -y_1 + y_2\\ \text{s.t.}\quad & x + y_1 + y_2 \le 10,\\ & 0 \le y_1 \le 9,\quad 0 \le y_2 \le 7.\end{aligned}\tag{A.5}$$

*Problem A.6* (see [15]). Consider

$$\begin{aligned}\min_{x,y}\quad & F(x,y) = -2x_1 + x_2 + 0.5y_1\\ \text{s.t.}\quad & x_1 + x_2 \le 2,\quad x \ge 0\\ \min_{y}\quad & f(x,y) = -4y_1 + y_2\\ \text{s.t.}\quad & 2x_1 - y_1 + y_2 \ge 2.5,\\ & -x_1 + 3x_2 - y_2 \ge -2,\\ & y_1 \ge 0,\quad y_2 \ge 0.\end{aligned}\tag{A.6}$$

*Problem A.7* (see [6]). Consider

$$\begin{aligned}\min_{x,y}\quad & F(x,y) = -8x_1 - 4x_2 + 4y_1 - 40y_2 - 4y_3\\ \text{s.t.}\quad & x_1 + 2x_2 - y_3 \le 1.3,\\ & x_1 \ge 0,\quad x_2 \ge 0\\ \min_{y}\quad & f(x,y) = 2y_1 + y_2 + 2y_3\\ \text{s.t.}\quad & -y_1 + y_2 + y_3 \le 1,\\ & 4x_1 - 2y_1 + 4y_2 - y_3 \le 2,\\ & 4x_2 + 4y_1 - 2y_2 - y_3 \le 2,\\ & y_1 \ge 0,\quad y_2 \ge 0,\quad y_3 \ge 0.\end{aligned}\tag{A.7}$$

*Problem A.8* (see [19]). Consider

$$\begin{aligned}\min_{x,y}\quad & F(x,y) = -4x_1 + 8x_2 + x_3 - x_4 + 9y_1 - 9y_2\\ \text{s.t.}\quad & -9x_1 + 3x_2 - 8x_3 + 3x_4 + 3y_1 \le 1,\\ & 4x_1 - 10x_2 + 3x_3 + 5x_4 + 8y_1 + 8y_2 \le 25,\\ & 4x_1 - 2x_2 - 2x_3 + 10x_4 - 5y_1 + 8y_2 \le 21,\\ & 9x_1 - 9x_2 + 4x_3 - 3x_4 - y_1 - 9y_2 \le -1,\\ & -2x_1 - 2x_2 + 8x_3 - 5x_4 + 5y_1 + 8y_2 \le 20,\\ & 7x_1 + 2x_2 - 5x_3 + 4x_4 - 5y_1 \le 11,\\ & x_1 \ge 0,\quad x_2 \ge 0,\quad x_3 \ge 0,\quad x_4 \ge 0\\ \min_{y}\quad & f(x,y) = -9y_1 + 9y_2\\ \text{s.t.}\quad & -6x_1 + x_2 + x_3 - 3x_4 - 9y_1 - 7y_2 \le -15,\\ & 4x_2 + 5x_3 + 10x_4 \le 26,\\ & -9x_1 + 9x_2 - 9x_3 + 5x_4 - 5y_1 - 4y_2 \le -5,\\ & 5x_1 + 3x_2 + x_3 + 9x_4 + y_1 + 5y_2 \le 32,\\ & y_1 \ge 0,\quad y_2 \ge 0.\end{aligned}\tag{A.8}$$

*Problem A.9* (see [15]). Consider

$$\begin{aligned}\min_{x,y}\quad & F(x,y) = 12x_1 - x_2 - 12x_3 + 13x_4\\ & \qquad + 2x_6 - 5x_8 + 6x_9 - 11x_{10}\\ & \qquad - 5y_1 - 6y_2 - 4y_3 - 7y_4,\\ \text{s.t.}\quad & Ax + By \le r_1,\\ & 0 \le x_i \le 10,\quad i = 1, 2, \dots, 10\end{aligned}$$

Table 9: Best results obtained by the proposed algorithm for 5 and 10 dimensions in second set of nonlinear BLPPs.

| Problem | Dimension | $(x^*, y^*)$ | $F(x^*, y^*)$ | $f(x^*, y^*)$ |
|---|---|---|---|---|
| SMD1 | 5 | (−1.33e − 3, −4.22e − 4, −2.38e − 9, −2.38e − 9, −4.22e − 4) | 1.94e − 6 | 1.77e − 6 |
| | 10 | (9.56e − 5, 1.87e − 3, 4.81e − 4, 2.35e − 3, −2.89e − 3, −7.45e − 9, −7.45e − 9, −7.45e − 9, 2.35e − 3, −2.89e − 3) | 1.76e − 5 | 3.76e − 6 |
| SMD2 | 5 | (−5.28e − 4, 7.12e − 4, −6.45e − 9, −6.45e − 9, 1.00e + 0) | 7.86e − 7 | 2.79e − 7 |
| | 10 | (0, 0, 0, 0, 0, 0, 0, 0, 1, 1) | 0 | 0 |
| SMD3 | 5 | (−4.04e − 4, −9.92e − 4, −5.54e − 6, −5.54e − 6, 5.26e − 5) | 1.15e − 6 | 1.67e − 7 |
| | 10 | (9.29e − 4, 2.24e − 3, −1.43e − 3, −1.60e − 3, −2.22e − 4, −5.06e − 3, −5.06e − 3, −5.06e − 3, 2.56e − 4, 2.29e − 4) | 1.33e − 5 | 8.03e − 6 |
| SMD4 | 5 | (−1.25e − 4, 3.17e − 5, 0, 0, 0) | 1.57e − 8 | 1.67e − 8 |
| | 10 | (0, 0, 0, 0, 0, 0, 0, 0, 0, 0) | 0 | 0 |
| SMD5 | 5 | (5.83e − 4, −1.33e − 3, 1.00e + 0, 1.00e − 0, −2.66e − 3) | 3.58e − 7 | 2.09e − 6 |
| | 10 | (4.53e − 4, 1.91e − 3, 1.75e − 4, 2.55e − 3, −3.64e − 3, 1.00e + 0, 1.00e + 0, 1.00e − 0, −2.63e − 4, −1.22e − 2) | 4.94e − 6 | 2.26e − 5 |
| SMD6 | 5 | (−1.33e − 3, −4.22e − 4, 0, 0, 0) | 1.77e − 6 | 1.94e − 6 |
| | 10 | (9.56e − 5, 1.87e − 3, 4.81e − 4, 2.35e − 3, −2.89e − 3, −7.45e − 9, −7.45e − 9, −7.45e − 9, 2.35e − 3, −2.89e − 3) | 1.76e − 5 | 3.76e − 6 |
| SMD7 | 5 | (3.88e − 4, 1.98e − 3, 7.69e − 9, 7.69e − 9, 1.00e + 0) | 4.01e − 6 | 2.45e − 10 |
| | 10 | (0, 0, 0, 0, 0, 0, 0, 0, 1, 1) | 0 | 0 |
| SMD8 | 5 | (−2.12e − 7, −8.26e − 3, 1.00e + 0, 1.00e + 0, −2.02e − 1) | 6.91e − 5 | 2.12e − 7 |
| | 10 | (−6.49e − 6, 6.99e − 8, −1.87e − 6, 7.29e − 4, 3.28e − 3, 1.00e − 0, 1.00e − 0, 1.00e − 0, 9.00e − 2, 1.49e − 1) | 2.69e − 5 | 8.42e − 6 |
| SMD9 | 5 | (−4.09e − 4, −3.62e − 4, 2.69e − 8, 2.69e − 8, −3.62e − 4) | 2.98e − 7 | 1.67e − 7 |
| | 10 | (0, 0, 0, 0,3.13e − 8, 3.13e − 8, 3.13e − 8, 1.90e − 8, 2.17e − 8) | 0 | 0 |
| SMD10 | 5 | (1.00e + 0, 1.00e + 0, 1, 1, 7.85e − 1) | 4.00e − 0 | 3.00e + 0 |
| | 10 | (5.00e − 1, 5.00e − 1, 5.00e − 1, 5.00e − 1, 5.00e − 1, 7.07e − 1, 7.07e − 1, 7.07e − 1, 4.64e − 1, 4.64e − 1) | 1.275e + 1 | 5.76e + 0 |
| SMD11 | 5 | (−1.77e − 3, 4.02e − 3, −1.33e − 6, −1.33e − 6, 3.69e − 1) | −1.00e + 0 | 1.00e + 0 |
| | 10 | (2.23e − 3, 1.18e − 3, 1.97e − 3, 5.74e − 3, 5.74e − 3, −4.69e − 7, −4.69e − 7, −4.69e − 7, 4.94e − 1, 4.98e − 1) | −1.00e + 0 | 1.00e + 0 |
| SMD12 | 5 | (1.00e + 0, 1.00e + 0, 1.00e − 0, 1.00e − 0, 3.10e − 5) | 3.00e − 0 | 4.00e + 0 |
| | 10 | (5.00e − 1, 5.00e − 1, 5.00e − 1, 5.00e − 1, 5.00e − 1, 7.07e − 1, 7.07e − 1, 7.07e − 1, −2.04e − 1, −2.04e − 1) | 1.216e + 1 | 6.76e + 0 |

Table 10: Comparison of accuracy for the upper level (UL) and lower level (LL) by the proposed algorithm and the algorithm in [28] for the test problems of the second set. A dash denotes that a feasible solution could not be obtained for the test problem. NA means the result is not available.

| Problem | Dimension | Median value by the proposed algorithm | | Median value in [28] | |
|---|---|---|---|---|---|
| | | UL accuracy | LL accuracy | UL accuracy | LL accuracy |
| SMD1 | 5 | $3.98e-5$ | $1.19e-5$ | $1.14e-4$ | $8.70e-5$ |
| | 10 | $6.61e-5$ | $3.66e-5$ | $3.32e-4$ | $1.80e-5$ |
| | 20 | $7.54e-5$ | $3.79e-5$ | NA | NA |
| SMD2 | 5 | $5.61e-5$ | $2.19e-5$ | $7.30e-5$ | $1.60e-5$ |
| | 10 | $7.26e-5$ | $3.86e-5$ | $6.60e-5$ | $1.10e-5$ |
| | 20 | $8.47e-5$ | $3.24e-5$ | NA | NA |
| SMD3 | 5 | $4.54e-5$ | $1.44e-5$ | $5.40e-5$ | $5.50e-5$ |
| | 10 | $6.35e-5$ | $3.09e-5$ | $3.59e-4$ | $3.30e-5$ |
| | 20 | $6.97e-5$ | $2.81e-5$ | NA | NA |
| SMD4 | 5 | $3.98e-5$ | $1.02e-5$ | $2.30e-5$ | $5.70e-5$ |
| | 10 | $6.55e-5$ | $2.68e-5$ | $2.86e-4$ | $2.70e-5$ |
| | 20 | $7.62e-5$ | $4.70e-5$ | NA | NA |
| SMD5 | 5 | $2.92e-5$ | $9.39e-5$ | $2.00e-6$ | $9.00e-6$ |
| | 10 | $6.23e-5$ | $9.72e-5$ | $5.20e-5$ | $9.00e-6$ |
| | 20 | $7.54e-5$ | $3.79e-5$ | NA | NA |
| SMD6 | 5 | $3.98e-5$ | $1.19e-5$ | $1.08e-4$ | $6.10e-5$ |
| | 10 | $6.61e-5$ | $3.66e-5$ | $1.44e-3$ | $8.20e-5$ |
| | 20 | $6.73e-5$ | $2.33e-5$ | NA | NA |
| SMD7 | 5 | $4.55e-5$ | $1.17e-9$ | $1.60e-5$ | $1.77e-4$ |
| | 10 | $6.55e-5$ | $3.15e-8$ | $6.26e-3$ | $1.27e-4$ |
| | 20 | $7.62e-5$ | $4.62e-7$ | NA | NA |
| SMD8 | 5 | $6.16e-5$ | $4.38e-2$ | $1.74e-4$ | $2.70e-5$ |
| | 10 | $7.96e-5$ | $4.55e-5$ | $3.12e-3$ | $1.57e-4$ |
| | 20 | $8.61e-5$ | $6.77e-5$ | NA | NA |
| SMD9 | 5 | $4.23e-5$ | $1.72e-5$ | $1.70e-5$ | $5.40e-5$ |
| | 10 | $5.90e-5$ | $3.17e-5$ | — | — |
| | 20 | $8.80e-5$ | $4.53e-5$ | NA | NA |
| SMD10 | 5 | $1.58e-5$ | $9.49e-6$ | $3.48e-2$ | $1.85e-2$ |
| | 10 | $3.36e-5$ | $1.67e-4$ | — | — |
| | 20 | — | — | NA | NA |
| SMD11 | 5 | $5.70e-5$ | $1.32e-5$ | $1.32e-2$ | $1.30e-2$ |
| | 10 | $9.66e-5$ | $1.40e-5$ | — | — |
| | 20 | — | — | NA | NA |
| SMD12 | 5 | $5.98e-5$ | $2.02e-5$ | $3.24e-2$ | $2.06e-4$ |
| | 10 | $1.33e-4$ | $5.58e-5$ | — | — |
| | 20 | — | — | NA | NA |

$$\begin{aligned} \min_{y} \quad & f(x,y) = 3y_1 - 2y_2 - 3y_3 - 3y_4 + y_5 + 6y_6 \\ \text{s.t.} \quad & Cx + Dy \le r_2, \\ & 0 \le y_i \le 10, \quad i = 1,2,\ldots,6, \end{aligned} \tag{A.9}$$

where

$$A = \begin{bmatrix} -2 & -3 & 14 & -2 & -9 & 2 & 1 & -4 & 0 & 2 \\ 1 & -7 & 13 & 0 & -15 & 2 & -8 & -4 & 4 & -7 \end{bmatrix},$$

$$B = \begin{bmatrix} -3 & 9 & -2 & -8 & 1 & -8 \\ -6 & -2 & 6 & 2 & 8 & -4 \end{bmatrix},$$

Table 11: Comparison of function evaluations (FE) for the upper level (UL) and lower level (LL) by the proposed differential evolution algorithm (DE) and the nested bilevel evolutionary algorithm (NBEA) in [28] for 5-dimensional test problems of the second set.

| Problem | Algorithm | Best FE | | Median FE | | Worst FE | |
|---|---|---|---|---|---|---|---|
| | | UL | LL | UL | LL | UL | LL |
| SMD1 | DE | 440 | 19,257 | 760 | 30,510 | 960 | 35,661 |
| | NBEA | 438 | 256,858 | 668 | 375,488 | 1,008 | 582,770 |
| SMD2 | DE | 360 | 16,583 | 680 | 28,759 | 880 | 35,672 |
| | NBEA | 380 | 196,744 | 628 | 332,197 | 1,102 | 613,221 |
| SMD3 | DE | 440 | 18,070 | 760 | 28,386 | 960 | 33,556 |
| | NBEA | 488 | 262,703 | 604 | 315,598 | 844 | 439,316 |
| SMD4 | DE | 400 | 12,284 | 600 | 17,694 | 800 | 21,990 |
| | NBEA | 420 | 259,486 | 608 | 366,294 | 796 | 480,675 |
| SMD5 | DE | 520 | 14,673 | 720 | 21,356 | 920 | 27,649 |
| | NBEA | 444 | 222,078 | 930 | 457,265 | 1,232 | 610,108 |
| SMD6 | DE | 440 | 4,789 | 760 | 8,157 | 960 | 10,210 |
| | NBEA | 540 | 334,763 | 696 | 427,114 | 936 | 585,358 |
| SMD7 | DE | 400 | 10,572 | 740 | 17,428 | 920 | 21,213 |
| | NBEA | 468 | 246,375 | 652 | 333,629 | 1,342 | 685,029 |
| SMD8 | DE | 560 | 36,486 | 1,200 | 87,965 | 1,800 | 134,656 |
| | NBEA | 812 | 443,430 | 1,008 | 582,583 | 2,076 | 1,218,196 |
| SMD9 | DE | 120 | 7,430 | 840 | 41,365 | 1,080 | 52,429 |
| | NBEA | 330 | 183,231 | 514 | 284,648 | 696 | 395,735 |
| SMD10 | DE | 1,840 | 69,110 | 2,380 | 89,266 | 4,040 | 165,797 |
| | NBEA | 480 | 179,986 | 758 | 277,696 | 1,316 | 501,639 |
| SMD11 | DE | 280 | 11,769 | 800 | 37,108 | 960 | 44,865 |
| | NBEA | 4,348 | 11,489,609 | 5,086 | 13,408,524 | 7,764 | 20,540,610 |
| SMD12 | DE | 2,120 | 117,423 | 2,740 | 149,102 | 4,040 | 297,407 |
| | NBEA | 354 | 6,211,173 | 738 | 12,950,512 | 1,196 | 20,983,708 |

$$C=\begin{bmatrix} -5 & 7 & 4 & -2 & 3 & -9 & 9 & -1 & -3 & 11\\ 6 & -5 & -3 & -2 & 8 & 5 & 8 & -3 & 7 & 3\\ -6 & -4 & 2 & 0 & -2 & 3 & -3 & 2 & 2 & 4\\ 5 & 6 & 0 & -4 & 3 & -8 & 1 & 0 & 2 & -3\\ 11 & -11 & 4 & 5 & -10 & -6 & 14 & -7 & -11 & -3\\ 9 & -12 & -4 & -10 & 2 & 8 & 5 & -11 & -4 & 1\\ 7 & -2 & -6 & 0 & -11 & 1 & -2 & -2 & -1 & -2 \end{bmatrix},$$

$$D=\begin{bmatrix} 10 & -9 & -6 & 4 & 6 & -3\\ -5 & -7 & 1 & 1 & -6 & 4\\ 10 & 5 & 6 & -4 & 3 & -1\\ -4 & -3 & -4 & -4 & 1 & 1\\ -10 & -7 & 7 & 7 & 2 & 7\\ 2 & -5 & 10 & 1 & 4 & 5\\ -5 & -5 & -6 & -5 & 1 & -12 \end{bmatrix},$$

$$r_1=(30,-134)^\top,\quad r_2=(83,92,168,-96,-133,89,-192)^\top. \tag{A.10}$$

## B. Nonlinear Test Problems

*Problem B.1* (see [35]). Consider the following

$$\begin{aligned} \min_{0\le x\le 50}\quad & F(x,y)=2x_1+2x_2-3y_1-3y_2-60\\ \text{s.t.}\quad & x_1+x_2+y_1-2y_2\le 40\\ \min_{-10\le y\le 20}\quad & f(x,y)=(y_1-x_1+20)^2+(y_2-x_2+20)^2\\ \text{s.t.}\quad & 2y_1-x_1+10\le 0,\\ & 2y_2-x_2+10\le 0. \end{aligned} \tag{B.1}$$

*Problem B.2* (see [30]). Consider the following

$$\begin{aligned} \min_{x\ge 0}\quad & F(x,y)=-x_1^2-3x_2^2-4y_1+y_2^2\\ \text{s.t.}\quad & x_1^2+2x_2\le 4\\ \min_{y\ge 0}\quad & f(x,y)=2x_1^2+y_1^2-5y_2\\ \text{s.t.}\quad & x_1^2-2x_1+x_2^2-2y_1+y_2\ge -3,\\ & x_2+3y_1-4y_2\ge 4. \end{aligned} \tag{B.2}$$

Table 12: Comparison of function evaluations (FE) for the upper level (UL) and lower level (LL) by the proposed differential evolution algorithm (DE) and the nested bilevel evolutionary algorithm (NBEA) in [28] for 10-dimensional test problems of the second set. A dash denotes that a feasible solution could not be obtained for the test problem. A notation of $(x)$ denotes that the algorithm terminated far away ($\Delta F \geq 0.1$) from the optimal solution.

| Problem | Algorithm | Best FE | | Median FE | | Worst FE | |
|---|---|---|---|---|---|---|---|
| | | UL | LL | UL | LL | UL | LL |
| SMD1 | DE | 1,800 | 60,209 | 2,040 | 68,569 | 2,320 | 77,968 |
| | NBEA | 1,080 | 862,653 | 2,534 | 1,623,356 | 3,488 | 2,074,334 |
| SMD2 | DE | 120 | 5,987 | 1,800 | 65,604 | 2,240 | 86,422 |
| | NBEA | 1,398 | 1,055,976 | 2,366 | 1,467,246 | 3,418 | 2,114,442 |
| SMD3 | DE | 1,880 | 122,781 | 2,080 | 134,397 | 2,360 | 154,032 |
| | NBEA | 1,210 | 900,358 | 2,278 | 1,383,632 | 2,862 | 1,805,562 |
| SMD4 | DE | 120 | 2,989 | 1,740 | 73,836 | 2,240 | 94,518 |
| | NBEA | 678 | 566,344 | 1,598 | 1,087,632 | 2,028 | 1,314,986 |
| SMD5 | DE | 1,560 | 116,112 | 1,960 | 143,344 | 2,240 | 165,197 |
| | NBEA | 1,620 | 1,226,344 | 2,890 | 1,993,124 | 3,492 | 2,483,442 |
| SMD6 | DE | 1,800 | 31,436 | 2,040 | 35,490 | 2,320 | 39,932 |
| | NBEA | 1,502 | 1,225,742 | 2,936 | 2,224,450 | 4,278$(x)$ | 3,786,498$(x)$ |
| SMD7 | DE | 120 | 6,037 | 2,240 | 79,360 | 3,960 | 142,556 |
| | NBEA | 1,382 | 932,460 | 2,394 | 1,566,481 | 3,858$(x)$ | 2,435,994$(x)$ |
| SMD8 | DE | 3,040 | 485,145 | 3,400 | 557,742 | 4,120 | 671,696 |
| | NBEA | 2,116 | 1,457,480 | 4,188 | 2,710,132 | 5,986$(x)$ | 5,294,734$(x)$ |
| SMD9 | DE | 120 | 9,469 | 2,720 | 220,749 | 3,360 | 293,523 |
| | NBEA | — | — | — | — | — | — |
| SMD10 | DE | 5,840 | 493,079 | 6,600 | 556,981 | 7,880 | 658,663 |
| | NBEA | — | — | — | — | — | — |
| SMD11 | DE | 5,520 | 592,663 | 10,960 | 1,143,648 | 12,040 | 1,257,916 |
| | NBEA | — | — | — | — | — | — |
| SMD12 | DE | 7,000 | 597,165 | 7,320 | 624,658 | 8,320 | 709,605 |
| | NBEA | — | — | — | — | — | — |

*Problem B.3* (see [31]). Consider the following

$$\begin{aligned}
\max_{x}\quad & F(x,y) = (y_1+y_3)(200-y_1-y_3) \\
& \qquad + (y_2+y_4)(160-y_2-y_4) \\
\text{s.t.}\quad & x_1+x_2+x_3+x_4 \leq 40 \\
& 0 \leq x_1 \leq 10, \quad 0 \leq x_2 \leq 5, \\
& 0 \leq x_3 \leq 15, \quad 0 \leq x_4 \leq 20 \\
\min_{0\leq y\leq 20}\quad & f_1(x,y) = (y_1-4)^2 + (y_2-13)^2 \\
\text{s.t.}\quad & 0.4y_1+0.7y_2-x_1 \leq 0, \\
& 0.6y_1+0.3y_2-x_2 \leq 0 \\
\min_{0\leq y\leq 40}\quad & f_2(x,y) = (y_3-35)^2 + (y_4-2)^2 \\
\text{s.t.}\quad & 0.4y_3+0.7y_4-x_3 \leq 0, \\
& 0.6y_3+0.3y_4-x_4 \leq 0.
\end{aligned} \tag{B.3}$$

*Problem B.4* (see [23]). Consider the following

$$\begin{aligned}
\max_{x\geq 0}\quad & F(x,y) = y_{11}y_{12}\sin x_1 + y_{21}y_{22}\sin x_2 \\
& \qquad + y_{31}y_{32}\sin x_3 \\
\text{s.t.}\quad & x_1+x_2+x_3 \leq 10 \\
\max_{y_1\geq 0}\quad & f_1(y_1) = y_{11}\sin y_{12} + y_{12}\sin y_{11} \\
\text{s.t.}\quad & y_{11}+y_{12} \leq x_1 \\
\max_{y_2\geq 0}\quad & f_2(y_2) = y_{21}\sin y_{22} + y_{22}\sin y_{21} \\
\text{s.t.}\quad & y_{21}+y_{22} \leq x_2 \\
\max_{y_3\geq 0}\quad & f_3(y_3) = y_{31}\sin y_{32} + y_{32}\sin y_{31} \\
\text{s.t.}\quad & y_{31}+y_{32} \leq x_3.
\end{aligned} \tag{B.4}$$

Table 13: Comparison of function evaluations (FE) for the upper level (UL) and lower level (LL) by the proposed differential evolution algorithm (DE) and the nested bilevel evolutionary algorithm (NBEA) in [28] for 20-dimensional test problems of the second set. A dash denotes that a feasible solution could not be obtained for the test problem. A notation of $(x)$ denotes that the algorithm terminated far away ($\Delta F \geq 0.1$) from the optimal solution.

| Problem | Algorithm | Best FE | | Median FE | | Worst FE | |
|---|---|---|---|---|---|---|---|
| | | UL | LL | UL | LL | UL | LL |
| SMD1 | DE | 3,600 | 537,221 | 3,960 | 585,595 | 4,520 | 649,558 |
| | NBEA | 3,210 | 3,105,178 | 5,248 | 5,262,456 | 7,378 | 6,868,944 |
| SMD2 | DE | 3,280 | 613,759 | 3,720 | 696,895 | 4,120 | 734,711 |
| | NBEA | 3,326 | 2,166,384 | 4,052 | 4,102,678 | 7,076 | 5,803,812 |
| SMD3 | DE | 3,520 | 890,998 | 3,920 | 981,330 | 4,240 | 1,064,627 |
| | NBEA | 3,220 | 3,696,032 | 4,282 | 4,814,112 | 6,314 | 7,015,724 |
| SMD4 | DE | 3,120 | 1,140,213 | 3,400 | 1,261,720 | 3,960 | 1,462,526 |
| | NBEA | 2,454 | 2,017,734 | 3,110 | 2,755,534 | 5,296 | 4,364,876 |
| SMD5 | DE | 3,600 | 1,175,815 | 3,960 | 1,308,906 | 4,520 | 1,516,729 |
| | NBEA | 4,488 | 4,574,482 | 7,004 | 8,800,232 | 9,290 | 12,064,566 |
| SMD6 | DE | 3,680 | 338,737 | 4,080 | 375,752 | 4,560 | 421,805 |
| | NBEA | 3,530 | 5,026,522 | 6,962 | 8,448,154 | 9,978$(x)$ | 12,448,922$(x)$ |
| SMD7 | DE | 3,840 | 633,706 | 4,560 | 747,605 | 8,040 | 1,161,633 |
| | NBEA | — | — | — | — | — | — |
| SMD8 | DE | 5,560 | 2,738,987 | 6,060 | 3,003,491 | 6,640 | 3,253,497 |
| | NBEA | — | — | — | — | — | — |
| SMD9 | DE | 5,760 | 1,240,993 | 6,840 | 1,504,952 | 7,240 | 1,589,831 |
| | NBEA | — | — | — | — | — | — |
| SMD10 | DE | — | — | — | — | — | — |
| | NBEA | — | — | — | — | — | — |
| SMD11 | DE | — | — | — | — | — | — |
| | NBEA | — | — | — | — | — | — |
| SMD12 | DE | — | — | — | — | — | — |
| | NBEA | — | — | — | — | — | — |

*Problem B.5* (see [23]). Consider the following

$$\begin{aligned}
\max_{x\geq 0}\quad & F(x,y)=\frac{(x_1+y_1)(x_2+y_2)}{1+x_1y_1+x_2y_2}\\
\text{s.t.}\quad & x_1^2+x_2^2\leq 100\\
\max_{y\geq 0}\quad & f(x,y)=-F(x,y)\\
\text{s.t.}\quad & 0\leq y_1\leq x_1,\qquad 0\leq y_2\leq x_2.
\end{aligned}\tag{B.5}$$

*Problem B.6* (see [32]). Consider the following

$$\begin{aligned}
\min_{x\geq 0}\quad & F(x,y)=-8x_1-4x_2+4y_1-40y_2-4y_3\\
\min_{y\geq 0}\quad & f(x,y)=\frac{1+x_1+x_2+2y_1-y_2+y_3}{6+2x_1+y_1+y_2-3y_3}\\
\text{s.t.}\quad & -y_1+y_2+y_3+y_4=1,\\
& 2x_1-y_1+2y_2-0.5y_3+y_5=1,\\
& 2x_2+2y_1-y_2-0.5y_3+y_6=1.
\end{aligned}\tag{B.6}$$

*Problem B.7* (see [25]). Consider the following

$$\begin{aligned}
\min_{0\leq x\leq 50}\quad & F(x,y)=\left|2x_1+2x_2-3y_1-3y_2-60\right|\\
\text{s.t.}\quad & x_1+x_2+y_1-2y_2\leq 40\\
\min_{-10\leq y\leq 20}\quad & f(x,y)=(y_1-x_1+20)^2+(y_2-x_2+20)^2\\
\text{s.t.}\quad & 2y_1-x_1+10\leq 0,\qquad 2y_2-x_2+10\leq 0.
\end{aligned}\tag{B.7}$$

*Problem B.8* (see [25]). Consider the following

$$\begin{aligned}
\min_{x_2\leq 15}\quad & F(x,y)=\Big|(x_1-30)^2+(x_2-20)^2\\
& \qquad -20y_1+20y_2-225\Big|\\
\text{s.t.}\quad & x_1+2x_2\geq 30,\qquad x_1+x_2\leq 25\\
\min_{0\leq y\leq 10}\quad & f(x,y)=(x_1-y_1)^2+(x_2-y_2)^2.
\end{aligned}\tag{B.8}$$

*Problem B.9* (see[25]). Consider the following

$$\begin{aligned}
\min_{x\geq 0}\quad & F(x,y) = \left|(x-1)^2 + 2y_1 - 2x + 1.2097\right| \\
\min_{y\geq 0}\quad & f(x,y) = (2y_1-4)^2 + (2y_2-1)^2 + xy_1 \\
\text{s.t.}\quad & 4x + 5y_1 + 4y_2 \leq 12, \\
& 4y_2 - 4x - 5y_1 \leq -4, \\
& 4x - 4y_1 + 5y_2 \leq 4, \\
& 4y_1 - 4x + 5y_2 \leq 4.
\end{aligned} \tag{B.9}$$

*Problem B.10* (see [25]). Consider the following

$$\begin{aligned}
\min_{x\geq 0}\quad & F(x,y) = \left|-8x_1 - 4x_2 + 4y_1 - 40y_2 - 4y_3 + 29.2\right| \\
\min_{y\geq 0}\quad & f(x,y) = \frac{1 + x_1 + x_2 + 2y_1 - y_2 + y_3}{6 + 2x_1 + y_1 + y_2 - 3y_3} \\
\text{s.t.}\quad & -y_1 + y_2 + y_3 + y_4 = 1, \\
& 2x_1 - y_1 + 2y_2 - 0.5y_3 + y_5 = 1, \\
& 2x_2 + 2y_1 - y_2 - 0.5y_3 + y_6 = 1.
\end{aligned} \tag{B.10}$$

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

The authors would like to thank the anonymous reviewers for their helpful suggestions and comments. The first author gratefully acknowledges the support from China Scholarship Council. This work was supported by Natural Science Basic Research Plan in Shaanxi Province of China (Program no. 2013JM1022) and the Fundamental Research Funds for the Central Universities (Grant no. K50511700004).

## References

[1] J. F. Bard, *Practical Bilevel Optimization: Algorithms and Applications*, vol. 30, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.

[2] L. N. Vicente and P. H. Calamai, "Bilevel and multilevel programming: a bibliography review," *Journal of Global Optimization*, vol. 5, no. 3, pp. 291–306, 1994.

[3] S. Dempe, "Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints," *Optimization*, vol. 52, no. 3, pp. 333–359, 2003.

[4] B. Colson, P. Marcotte, and G. Savard, "Bilevel programming: a survey," *4OR*, vol. 3, no. 2, pp. 87–107, 2005.

[5] B. Colson, P. Marcotte, and G. Savard, "A trust-region method for nonlinear bilevel programming: algorithm and computational experience," *Computational Optimization and Applications*, vol. 30, no. 3, pp. 211–227, 2005.

[6] P. Hansen, B. Jaumard, and G. Savard, "New branch-and-bound rules for linear bilevel programming," *Society for Industrial and Applied Mathematics*, vol. 13, no. 5, pp. 1194–1217, 1992.

[7] 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.

[8] H. I. Calvete, C. Galé, and P. M. Mateo, "A new approach for solving linear bilevel problems using genetic algorithms," *European Journal of Operational Research*, vol. 188, no. 1, pp. 14–28, 2008.

[9] J. F. Bard and J. T. Moore, "A branch and bound algorithm for the bilevel programming problem," *SIAM Journal on Scientific and Statistical Computing*, vol. 11, no. 2, pp. 281–292, 1990.

[10] C. H. M. de Sabóia, M. Campêlo, and S. Scheimberg, "A computational study of global algorithms for linear bilevel programming," *Numerical Algorithms*, vol. 35, no. 2–4, pp. 155–173, 2004.

[11] K. Lan, U. Wen, H. Shih, and E. S. Lee, "A hybrid neural network approach to bilevel programming problems," *Applied Mathematics Letters*, vol. 20, no. 8, pp. 880–884, 2007.

[12] C. Audet, G. Savard, and W. Zghal, "New branch-and-cut algorithm for bilevel linear programming," *Journal of Optimization Theory and Applications*, vol. 134, no. 2, pp. 353–370, 2007.

[13] H. Li, Y. Jiao, F. S. Zhang, and L. Zhang, "An efficient method for linear bilevel programming problems based on the orthogonal genetic algorithm," *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 9, pp. 2837–2846, 2009.

[14] S. R. Hejazi, A. Memariani, G. Jahanshahloo, and M. M. Sepehri, "Linear bilevel programming solution by genetic algorithm," *Computers and Operations Research*, vol. 29, no. 13, pp. 1913–1925, 2002.

[15] H. Tuy, A. Migdalas, and N. T. Hoai-Phuong, "A novel approach to bilevel nonlinear programming," *Journal of Global Optimization*, vol. 38, no. 4, pp. 527–554, 2007.

[16] G. Anandalingam and D. J. White, "A solution method for the linear static Stackelberg problem using penalty functions," *IEEE Transactions on Automatic Control*, vol. 35, no. 10, pp. 1170–1173, 1990.

[17] J. S. Angelo, E. Krempser, and H. J. C. Barbosa, "Differential evolution for bilevel programming," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '13)*, pp. 470–477, June 2013.

[18] Y. H. Liu and T. H. Spencer, "Solving a bilevel linear program when the inner decision maker controls few variables," *European Journal of Operational Research*, vol. 81, no. 3, pp. 644–651, 1995.

[19] K. Moshirvaziri, M. A. Amouzegar, and S. E. Jacobsen, "Test problem construction for linear bilevel programming problems," *Journal of Global Optimization*, vol. 8, no. 3, pp. 235–243, 1996.

[20] Y. Wang and C. Dang, "An evolutionary algorithm for global optimization based on level-set evolution and latin squares," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 5, pp. 579–595, 2007.

[21] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[22] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, Perth, Australia, December 1995.

[23] B. Liu, "Stackelberg-Nash equilibrium for multilevel programming with multiple followers using genetic algorithms,"

*Computers & Mathematics with Applications*, vol. 36, no. 7, pp. 79–89, 1998.

[24] V. Oduguwa and R. Roy, "Bi-level optimization using genetic algorithm," in *Proceedings of the 2002 IEEE International Conference on Artificial Intelligence Systems*, pp. 123–128, 2002.

[25] Y. Wang, Y.-C. Jiao, and H. Li, "An evolutionary algorithm for solving nonlinear bilevel programming based on a new constraint-handling scheme," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 35, no. 2, pp. 221–232, 2005.

[26] X. Li, P. Tian, and X. Min, "A hierarchical particle swarm optimization for solving bilevel programming problems," in *Artificial Intelligence and Soft Computing—ICAISC 2006*, vol. 4029 of *Lecture Notes in Computer Science*, pp. 1169–1178, Springer, Berlin, Germany, 2006.

[27] Y. Wang, H. Li, and C. Dang, "A new evolutionary algorithm for a class of nonlinear bilevel programming problems and its global convergence," *INFORMS Journal on Computing*, vol. 23, no. 4, pp. 618–629, 2011.

[28] A. Sinha, P. Malo, and K. Deb, "Test problem construction for single-objective bilevel optimization," *Evolutionary Computation Journal*, 2014.

[29] H. Li and L. Zhang, "A differential evolution with two mutation strategies for linear bilevel programming problems," in *Proceedings of the 9th International Conference on Computational Intelligence and Security (CIS '13)*, pp. 55–60, 2013.

[30] M. A. Amouzegar, "A global optimization method for nonlinear bilevel programming problems," *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 29, no. 6, pp. 771–777, 1999.

[31] J. F. Bard, "Convex two-level optimization," *Mathematical Programming*, vol. 40, no. 1, pp. 15–27, 1988.

[32] H. I. Calvete and C. Galé, "The bilevel linear/linear fractional programming problem," *European Journal of Operational Research*, vol. 114, no. 1, pp. 188–197, 1999.

[33] S. Das and P. N. Suganthan, "Differential evolution: a survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.

[34] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2–4, pp. 311–338, 2000.

[35] E. Aiyoshi and K. Shimizu, "A solution method for the static constrained Stackelberg problem via penalty method," *IEEE Transactions on Automatic Control*, vol. 29, no. 12, pp. 1111–1114, 1984.