

Research Article

An Evolutionary Algorithm Using Duality-Base-Enumerating Scheme for Interval Linear Bilevel Programming Problems

Hecheng Li¹ and Lei Fang²

¹ Department of Mathematics, Qinghai Normal University, Xining 810008, China

² Business School, Nankai University, Tianjin 300070, China

Correspondence should be addressed to Lei Fang; fanglei@nankai.edu.cn

Received 2 June 2014; Accepted 29 July 2014; Published 13 August 2014

Academic Editor: Yuping Wang

Copyright © 2014 H. Li and L. Fang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Interval bilevel programming problem is hard to solve due to its hierarchical structure as well as the uncertainty of coefficients. This paper is focused on a class of interval linear bilevel programming problems, and an evolutionary algorithm based on duality bases is proposed. Firstly, the objective coefficients of the lower level and the right-hand-side vector are uniformly encoded as individuals, and the relative intervals are taken as the search space. Secondly, for each encoded individual, based on the duality theorem, the original problem is transformed into a single level program simply involving one nonlinear equality constraint. Further, by enumerating duality bases, this nonlinear equality is deleted, and the single level program is converted into several linear programs. Finally, each individual can be evaluated by solving these linear programs. The computational results of 7 examples show that the algorithm is feasible and robust.

1. Introduction

The bilevel programming problem (BLPP) is a hierarchical optimization problem, involving two optimization problems located at upper and lower levels. The constraints of the upper level problem (the leader's problem) are determined implicitly by the lower level problem (the follower's problem). Mathematically, this is a mathematical program that contains an optimization problem in the constraints and can be formulated as follows:

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

where $x \in R^n$ are the upper level variables, which are controlled by the leader, and $y \in R^m$ are called the lower level variables and are controlled by the follower; $F, f: R^{n+m} \mapsto R$ are the leader and the follower objective functions, respectively, while $G(x, y) \leq 0$ and $g(x, y) \leq 0$ are called the upper

level and the lower level constraints which consist of some inequalities or equalities. In problem (1)

$$\begin{aligned} & \min_{x \in X} F(x, y), \\ & \text{s.t. } G(x, y) \leq 0, \\ & \min_{y \in Y} f(x, y), \\ & \text{s.t. } g(x, y) \leq 0 \end{aligned} \quad (2)$$

are known as the upper level and the lower level problems (the leader's and the follower's problems), respectively. Sets $x \in X$ and $y \in Y$ are additional constraints, such as bound constraints and integer restrictions.

For problem (1), the decision-making is hierarchically executed. The leader moves first to select an x and attempts to minimize their own objective function, and then the follower observes the leader's action and selects a y in a way that is personally optimal. If the point (x, y) also satisfies other constraints at upper level, it is called a feasible point. To solve (1) is to find a pair (x, y) among all feasible points such that the objective $F(x, y)$ is minimized. Unlike single level

mathematical programs, any feasible solution to the bilevel program (1) must satisfy the optimality of the follower's problem when the leader's variables x are fixed.

This kind of hierarchical optimization was first realized in the field of game theory by Stackelberg [1] in his monograph about market economy; as a result, this problem is also known as Stackelberg problem. As mathematical programming, bilevel programs were initially considered by Bracken and McGill in a series of papers [2–4]. By that time, such problems were called mathematical programs with optimization problems in the constraints. The term “bilevel programming” appeared for the first time in a paper by Candler and Norton [5], who considered a multilevel formulation in the context of agricultural economics. Since that time, lots of papers have been dedicated to this topic.

For a bilevel program, when the upper level problem involves some constraints depending on both x and y , this problem is more complex than those not including the leader's constraints. The leader's constraints can cause the feasible region to become unconnected, even empty. To our knowledge, it seems that Aiyoshi and Shimizu [6] first introduced the leader's constraints in bilevel programming models.

If the follower's problem has a unique optimal solution for all leader's variable values, this problem is equivalent to a one-level optimization problem having an implicitly defined objective and constraint functions. For problems having nonunique lower level optimal solutions, the optimistic (or weak) and the pessimistic (or strong) approaches were discussed. In the case of optimistic bilevel programming, it is assumed that, whenever the optimal solution set of the follower is not a singleton, the leader is allowed to select the element in the set that suits them best. When cooperation of the leader and the follower is not allowed, the pessimistic case arises. In this case, the leader wishes to limit the “damage” resulting from an undesirable selection of the follower [7, 8]. In order that the bilevel program can be well posed, we only discuss optimistic model in this paper.

On the one hand, bilevel programming problems are intrinsically hard, and even the “simplest instance,” the linear BLPP, was shown to be strongly NP-hard [7]. On the other hand, a variety of intractable problems in real world have been modeled as bilevel programming problems and solved, such as the investigation of network of oligopolies [9], the traffic planning [10, 11], the network design problem [12], and the terrorist threat problem [13]. The challenge of theoretic researches and applications have been stimulating factors for the development of BLPPs, and the usefulness of these mathematical programs in modelling hierarchical decision processes and engineering design problems prompted researchers to pay close attention to bilevel programs. As a result, over the past 20 years or so, the field of bilevel optimization has received a lot of attention in developing efficient algorithms and optimality conditions [7, 8, 14, 15]. Most of the classical, theoretical, and algorithmic researches are focused on linear bilevel programming problems [16]. An important property of linear bilevel programs is that their solution set, whenever it is nonempty, contains at least one vertex of the constraint region [14], by which some efficient

approaches, such as “ k th best” algorithms, branch and bound approaches, and penalty methods, have been proposed [7, 14, 17].

In the above real-world and theoretical problems, all coefficients involved are exactly known. However, in practice, it is very common for the coefficient values to be only approximately known because relevant data is nonexistent or scarce, difficult to obtain or estimate, and so forth. Therefore, it is very necessary to explicitly consider the intrinsic uncertainty associated with the model coefficients in mathematical programming models. Uncertainty can be handled mainly in three manners: interval programming, stochastic programming, or fuzzy numbers. Since it is not always easy to specify the membership function or probability distribution in an inexact environment, interval programming is the most concerned one among these approaches. For this problem, one always needs to compute the optimal value range between the best and the worst optimal objective values. These two extreme values allow the decision maker to better understand the risk involved.

Mathematical programs with interval coefficients in the objective function and/or in the constraints have been addressed in the literature [18, 19]. For this kind of problems, not only does an algorithm need to search the space of variables, but it also needs to specify the coefficient values in the intervals given. This means that the search procedure is more complex than that for problems with exact coefficients. In view of the inherent complexity of the problem, most of the algorithmic approaches are focused on one-level mathematical programming with interval coefficients, especially on linear case. Ishibuchi and Tanaka [20] investigated the mathematical programming problem with interval coefficients in objective function and reduced the problem to the question of determining the Pareto-optimal solutions of a bicriteria linear programming problem by using the defined order relations. Chanas and Kuchta [19] introduced a family of preference relations and presented a unified way to determine the solution of the problem with interval objective coefficients. Lai et al. [21] discussed a class of linear programming problems with interval coefficients in both the objective function and constraints; the noninferior solutions to such problems were defined based on two order relations between intervals. One can obtain the solution by solving a parametric linear programming problem. Molai and Khorram [22] introduced a satisfaction function and defined a satisfactory solution to the problem. Chinneck and Ramadan [23] proposed an approach for a class of linear programming with interval coefficients by analyzing all kinds of variables and constraints. Hladik [24] presented a general approach to the situation; the feasible set is described by an arbitrary linear interval system. Oliveira and Antunes [25] provided an overview of the different approaches reported in the literature to deal with interval multiple objective linear programming models. For nonlinear case, very few results have been obtained. Hladik [26] considered a generalized linear fractional programming problem with interval data and presented an efficient method which deals with the problem by solving at most four linear fractional programs. Li and Tian [27] investigated interval quadratic programming

and transformed the problem into two exact mathematical programs by using the bounds of intervals and duality.

For bilevel programming problems with interval coefficients, there exist few approaches. Calvete and Galé [18] addressed a linear bilevel problem in which only objective coefficients were assumed to lie between specific bounds and developed two enumerative algorithms to compute the optimal value range based on the k th best method. Ren and Wang [28] transformed the bilevel linear programming problem with fuzzy random coefficients into interval linear bilevel programming problems. In order to obtain the optimal value range, the interval programming problem was further converted into two BLPPs with determined coefficients. However, in these approaches, one has to execute two different algorithms to compute the upper bound and the lower bound of optimal objective values, respectively.

Evolutionary algorithms have been widely adopted in dealing with hard optimization problems, which can efficiently get away from local optima [29]. In this paper, we concentrate on the linear bilevel programming problem in which the coefficients of two objective functions are intervals as well as the right-hand-side values of constraints and present an efficient evolutionary algorithm based on duality bases for solving the problem. In the proposed algorithm, the intervals of the lower level objective coefficients and RHS of constraints are taken as the search space, and the duality conditions of the linear program are adopted to evaluate each individual.

This paper is organized as follows. The discussed problem and some notations are presented in Section 2, and a transformation procedure is investigated in Section 3. A real coded evolutionary algorithm is given based on the duality theorem of the linear program in Section 4, and this is followed by the experimental results in Section 5. We finally conclude our paper in Section 6.

2. Interval Linear Bilevel Programming Problem

In this section an interval linear BLPP (ILBP) is presented, in which the coefficients of the objective functions at two levels are interval numbers, and RHS of constraints are also specified by intervals. This means that all constants/coefficients in the bilevel program are interval numbers except for the coefficient matrix at the left-hand side of constraints. The problem can be formulated as follows:

$$\begin{aligned} \min_{x \geq 0} \quad & F(x, y) = cx + dy, \\ \min_{y \geq 0} \quad & f(x, y) = ey, \\ \text{s.t.} \quad & Ax + By \leq b, \end{aligned} \quad (3)$$

where $c \in [c^l, c^u]$, $d \in [d^l, d^u]$, $e \in [e^l, e^u]$, and $b \in [b^l, b^u]$ are interval vectors; $x \in R^n$, $y \in R^m$; A and B are coefficient matrices, and there are q inequality constraints in the lower level program. In the problem, c^l, d^l, e^l , and b^l are

lower bound vectors of corresponding intervals, whereas c^u, d^u, e^u , and b^u provide upper bounds for these intervals. For simplicity, we denote the interval linear bilevel program by $\text{ILBP}(c, d, e, b)$ when the objective coefficients and the right-hand-side vector are taken as c, d, e , and b , respectively. Next, for exact bilevel programming problem $\text{ILBP}(c, d, e, b)$, we introduce some related definitions and notations.

- (1) Constraint region: $S = \{(x, y) \mid Ax + By \leq b, x, y \geq 0\}$.
- (2) For x fixed, the feasible region of follower's problem: $S(x) = \{y \mid Ax + By \leq b, y \geq 0\}$.
- (3) Projection of S onto the leader's decision space: $S(X) = \{x \mid \exists y, (x, y) \in S\}$.
- (4) The follower's rational reaction set for each $x \in S(X)$: $M(x) = \{y \mid y \in \text{argmin}\{ev, v \in S(x)\}\}$.
- (5) Inducible region: $\text{IR} = \{(x, y) \in S \mid y \in M(x)\}$.

In order to avoid too much theoretical analysis of the existence of solutions and center on the algorithmic approach, we assume that the solution of (3) always exists for any selected coefficients. Some definitions of optimal solutions are presented as follows.

Definition 1 (optimal solution). (x, y) is called an optimal solution to (3) if there exist $c \in [c^l, c^u]$, $d \in [d^l, d^u]$, $e \in [e^l, e^u]$, and $b \in [b^l, b^u]$ such that (x, y) is an optimal solution to $\text{ILBP}(c, d, e, b)$.

Definition 2 (best optimal solution). An optimal solution (x^*, y^*) is called the best optimal solution to (3) if, for any optimal solution (x, y) , the inequality

$$F(x^*, y^*) \leq F(x, y) \quad (4)$$

holds.

Definition 3 (worst optimal solution). An optimal solution (x^*, y^*) is called the worst optimal solution to (3) if, for any optimal solution (x, y) , the inequality

$$F(x^*, y^*) \geq F(x, y) \quad (5)$$

holds.

The purpose of solving (3) is to obtain the best and the worst optimal solutions as well as the corresponding objective coefficient values and the values of RHS of the constraints.

3. Problem Transformation

It is very hard for one to directly solve $\text{ILBP}(3)$; hence, we have to transform the problem into several simpler subproblems than present one. Considering the characteristics of objective functions, we have the following.

Theorem 4. *The best optimal solution to ILBP(3) can be obtained by solving ILBP(c^l, d^l, e, b).*

Proof. Let (x^b, y^b) be the best optimal solution to ILBP(3) and be obtained by solving ILBP($\bar{c}, \bar{d}, \bar{e}, \bar{b}$); here, $\bar{c} \in [c^l, c^u]$, $\bar{d} \in [d^l, d^u]$, $\bar{e} \in [e^l, e^u]$, and $\bar{b} \in [b^l, b^u]$. On the one hand, if $\bar{c} \neq c^l$ or $\bar{d} \neq d^l$, that is, there is at least one component of \bar{c} or \bar{d} that is not at a lower bound, it follows that $c^l x^b + d^l y^b < \bar{c} x^b + \bar{d} y^b$ since $x^b \geq 0$ and $y^b \geq 0$. On the other hand, (x^b, y^b) is also a feasible solution of ILBP($c^l, d^l, \bar{e}, \bar{b}$) because both ILBP($c^l, d^l, \bar{e}, \bar{b}$) and ILBP($\bar{c}, \bar{d}, \bar{e}, \bar{b}$) share a common feasible region (inducible region). One can conclude that ILBP($c^l, d^l, \bar{e}, \bar{b}$) can provide a better optimal objective value than $\bar{c} x^b + \bar{d} y^b$. This is a contradiction. Hence, the proof is completed. \square

Theorem 5. *The worst optimal solution to ILBP(3) can be obtained when ILBP(c^u, d^u, e, b) is solved.*

Proof. For arbitrarily selected $c = \bar{c}, d = \bar{d}, e = \bar{e}$, and $b = \bar{b}$, here, $\bar{c} \in [c^l, c^u]$, $\bar{d} \in [d^l, d^u]$, $\bar{e} \in [e^l, e^u]$, and $\bar{b} \in [b^l, b^u]$. It should be noted that both ILBP($\bar{c}, \bar{d}, \bar{e}, \bar{b}$) and ILBP($c^u, d^u, \bar{e}, \bar{b}$) share a common inducible region, and, for any point (x, y) in this region, inequality $\bar{c}x + \bar{d}y \leq c^u x + d^u y$ always holds. This means that, in this common inducible region, ILBP($c^u, d^u, \bar{e}, \bar{b}$) has no less optimal objective value than ILBP($\bar{c}, \bar{d}, \bar{e}, \bar{b}$). It follows that the worst optimal solution to ILBP(3) can be obtained only by solving ILBP(c^u, d^u, e, b). This completes the proof. \square

On basis of Theorems 4 and 5, the best and the worst optimal solutions to ILBP(3) can be got by solving two interval bilevel programs as follows:

$$\begin{aligned} \min_{x \geq 0} \quad & F(x, y) = c^l x + d^l y, \\ \min_{y \geq 0} \quad & f(x, y) = [e^l, e^u] y, \end{aligned} \quad (6)$$

$$\text{s.t.} \quad Ax + By \leq [b^l, b^u],$$

$$\begin{aligned} \min_{x \geq 0} \quad & F(x, y) = c^u x + d^u y, \\ \min_{y \geq 0} \quad & f(x, y) = [e^l, e^u] y, \end{aligned} \quad (7)$$

$$\text{s.t.} \quad Ax + By \leq [b^l, b^u].$$

It should be noted that problems (6) and (7) share a common structure; that is, the upper objective is a linear function, whereas the lower level is an interval linear programming problem. Without any loss of generality, we only consider the solution method of problem (6), and the other can be dealt with by using the same procedure. At first, we take $\forall t \in [e^l, e^u]$ and $\forall \bar{b} \in [b^l, b^u]$; that is, when t and \bar{b} are

fixed, we can obtain the following linear bilevel programming problem:

$$\begin{aligned} \min_{x \geq 0} \quad & F(x, y) = c^l x + d^l y, \\ \min_{y \geq 0} \quad & f(x, y) = ty, \\ \text{s.t.} \quad & Ax + By \leq \bar{b}. \end{aligned} \quad (8)$$

In this problem, t and \bar{b} , as parameters, are different from variables x and y . In order to efficiently solve the problem (8), we further transform (8) into the following problem based on the duality theorem:

$$\begin{aligned} \min_{x \geq 0} \quad & F(x, y) = c^l x + d^l y, \\ \text{s.t.} \quad & ty = (Ax - \bar{b})^T u, \\ & Ax + By \leq \bar{b}, \quad -B^T u \leq t^T, \\ & u \geq 0, \quad y \geq 0. \end{aligned} \quad (9)$$

After doing so, it follows that one can obtain the solutions to ILBP(6) by solving (9) for all $t \in [e^l, e^u]$ and $\bar{b} \in [b^l, b^u]$. We select evolutionary algorithms to solve the problem. In the proposed algorithm, t and \bar{b} are taken as evolutionary search variables; that is, the values of these two variables are encoded as individuals in populations and the region $[e^l, e^u] \times [b^l, b^u]$ is taken as the search space. For each individual, we solve the problem (9). In all of the obtained optimal solutions, the one with the minimal objective value is the best optimal solution to ILBP(3).

Following a similar procedure, one can obtain the worst optimal solution by solving the following nonlinear program for each $t \in [e^l, e^u]$ and $\bar{b} \in [b^l, b^u]$:

$$\begin{aligned} \min_{x \geq 0} \quad & F(x, y) = c^u x + d^u y, \\ \text{s.t.} \quad & ty = (Ax - \bar{b})^T u, \\ & Ax + By \leq \bar{b}, \quad -B^T u \leq t^T, \\ & u \geq 0, \quad y \geq 0. \end{aligned} \quad (10)$$

In all obtained objective values by solving (10) with different parametric t and \bar{b} , the worst one can be taken as the worst optimal value of ILBP(3).

In fact, problems (9) and (10) are nonlinear programming problems since the first constraint is nonlinear. This implies that it is hard to solve these two problems. In order to obtain precise optimal solutions to these nonlinear programs, we apply a duality-base-enumerating method to solve these problems. At first, we begin with constraint $-B^T u \leq t^T$ and transform the constraint into an equality by adding some slack variables. In order to avoid generating too many variables, we still denote the equality by $-B^T u = t^T$. Then, we enumerate all bases generated by constraint equality group $-B^T u = t^T$ and denote these bases by u^1, u^2, \dots, u^k , which are

actually the bases of the lower level dual problem of ILBP(8). Finally, for each fixed t and \bar{b} , nonlinear programming problem (9) can be divided into k linear programs in terms of these duality bases:

$$\begin{aligned} \min_{x \geq 0} \quad & F(x, y) = c^l x + d^l y, \\ \text{s.t.} \quad & ty = (Ax - \bar{b})^T u^i, \\ & Ax + By \leq \bar{b}, \\ & y \geq 0, \end{aligned} \quad (11)$$

$i = 1, 2, \dots, k$. We solve these k linear programs and take the minimal objective value as the optimal value of problem (9). For problem (10), the same procedure can be adopted for obtaining the optimal solutions.

It should be noted that both t and \bar{b} have the same value region when the best and the worst optimal solutions are searched, and problems (9) and (10) share a common feasible region. Hence, it is computationally expensive if the best and the worst optimal solutions are separately searched by using the same procedure. In the proposed approach, both the best and the worst optimal solutions can be obtained only by executing the algorithm once.

4. The Proposed Solution Method

In this section, we present an evolutionary algorithm using a double-fitness-evaluation technique and a real coding scheme.

4.1. Chromosome Encoding. We encode the values of t and \bar{b} using the real-number-coding scheme; that is, $[e^l, e^u] \times [b^l, b^u]$ is taken as the search space of the evolutionary algorithm. For each individual, the code is composed of two parts; the first part is the value of t , whereas the other is the value of \bar{b} .

4.2. Fitness Evaluation. In order to obtain the best and the worst optimal solution in one run of the algorithm, we design a double-fitness-evaluation scheme. In this process, each individual must endure twice evaluations for searching the best and the worst optimal solutions. First, for each individual $l = (t_1, \dots, t_m; \bar{b}_1, \dots, \bar{b}_q)$, problems (9) and (10) are solved and two objective values can be obtained. These two objective values are taken as the fitness evaluations of the individual. Here, the objective value of (9) is called the first fitness and is used to search the best optimal solutions, whereas that of (10) is called the second fitness which is always applied to find the worst optimal solutions. From the viewpoint of the best optimal solution, the less the first fitness is, the better the individual is. For the worst optimal solution, the larger fitness implies that the individual is better.

If problem (9) has no optimal solution, then the first fitness is taken as M ; here, M is a positive number and is large enough; when problem (10) has no optimal solution, the second fitness is taken as $-M$.

4.3. Crossover and Mutation Operators. In the designed algorithm a heuristic crossover is presented and Gaussian mutations are adopted.

4.3.1. Crossover Operator. Let l_b be the best individual found so far by the first fitness and let l_w be the best individual found so far by the second fitness. Let l be a crossover parent individual. For $\forall \alpha_i$ taken randomly in $[0, 2]$, $i = 1, 2$, crossover offspring can be generated as follows:

$$\begin{aligned} o_1 &= l + \alpha_1 (l_b - l), \\ o_2 &= l + \alpha_2 (l_w - l). \end{aligned} \quad (12)$$

4.3.2. Mutation Operator. \hat{l} is taken for mutation; then the offspring is given as follows:

$$o_m = \hat{l} + N(0, \sigma^2); \quad (13)$$

here, $N(0, \sigma^2)$ is a variable distributed normally with mean 0 and variance σ^2 .

4.4. Algorithm Description. In this subsection we propose an evolutionary algorithm using duality-base-enumerating scheme for interval linear bilevel programs (EA-DBES).

Step 1 (initial population). N initial points are generated randomly in search space $[e^l, e^u] \times [b^l, b^u]$ to form the initial population $\text{pop}(0)$. Let $g = 0$.

Step 2 (fitness). Evaluate each point in $\text{pop}(g)$ by using the first and second fitness evaluations and record the best solutions l_b and l_w regarding two fitness values.

Step 3 (crossover and mutation). The designed heuristical crossover operator and Gaussian mutations are adopted to generate all genetic offspring. The set of all offspring is denoted by O .

Step 4. Evaluate the fitness values of all offspring and update l_b and l_w .

Step 5. For each fitness, the best $[N/3]$ individuals are selected from set $\text{pop}(g) \cup O$, and other $N - 2 \times [N/3]$ individuals are chosen at random from the remaining elements of $\text{pop}(g) \cup O$. A total of N individuals are put into the next generation of population $\text{pop}(g + 1)$.

Step 6. If the termination condition is satisfied, then the algorithm is stopped; otherwise, let $g = g + 1$; go to Step 3.

5. Computational Examples

Since it is very hard to solve interval bilevel programming problems, there are few examples investigated in the literature. In order to illustrate the performance of EA-DBES, we first select two examples with objective interval coefficients from [18] and denote them by Examples 1 and 2, respectively. Further, in order to generate enough computational examples

to test the performance of the algorithm, we transform some existing bilevel programming examples into ILBPs by introducing a coefficient transformation procedure as follows: let a be a real number; then the number is converted into an interval $[a - |a|/2, a + |a|/2]$. Using the procedure, we obtain 5 computational examples of interval bilevel programming problems. First, we convert the right-hand-side vectors in Examples 1 and 2 into interval numbers and denote these two obtained problems by Examples 3 and 4, respectively. In addition, we also select three linear bilevel programming problems with determined coefficients from the literature [14, 15], which are frequently solved as computational examples to estimate the performance of the proposed algorithms, and then convert both the objective coefficients and the right-hand-side vector into interval numbers by using the transformation procedure. Three obtained interval programming problems are denoted by Examples 5–7, respectively. A total of 7 computational examples are shown as follows.

Example 1. Let

$$\begin{aligned} \min_{2 \geq x \geq 0} \quad & x + y_1 + y_2, \\ \min_{y \geq 0} \quad & [-1, 2] y_1 + [1, 3] y_2, \\ \text{s.t.} \quad & -3x - 3y_1 + 2y_2 \leq 1, \\ & x + 2y_1 \leq 4, \quad y_2 \leq 2. \end{aligned} \quad (14)$$

Example 2. Let

$$\begin{aligned} \min_{2 \geq x \geq 0} \quad & [1, 4] x + [1, 3] y_1 + [1, 2] y_2, \\ \min_{y \geq 0} \quad & [2, 5] y_1 + [-3, -1] y_2, \\ \text{s.t.} \quad & -3x - 3y_1 + 2y_2 \leq 1, \\ & x + 2y_1 \leq 4, \quad y_2 \leq 2. \end{aligned} \quad (15)$$

Example 3. Let

$$\begin{aligned} \min_{2 \geq x \geq 0} \quad & x + y_1 + y_2, \\ \min_{y \geq 0} \quad & [-1, 2] y_1 + [1, 3] y_2, \\ \text{s.t.} \quad & -3x - 3y_1 + 2y_2 \leq \left[\frac{1}{2}, \frac{3}{2} \right], \\ & x + 2y_1 \leq [2, 6], \quad y_2 \leq [1, 3]. \end{aligned} \quad (16)$$

Example 4. Let

$$\begin{aligned} \min_{x \geq 0} \quad & [1, 4] x + [1, 3] y_1 + [1, 2] y_2, \\ \min_{y \geq 0} \quad & [2, 5] y_1 + [-3, -1] y_2, \\ \text{s.t.} \quad & -3x - 3y_1 + 2y_2 \leq \left[\frac{1}{2}, \frac{3}{2} \right], \\ & x + 2y_1 \leq [2, 6], \quad y_2 \leq [1, 3]. \end{aligned} \quad (17)$$

Example 5. Let

$$\begin{aligned} \min_{x \geq 0} \quad & [-12, -4] x_1 + [-6, -4] x_2 + [2, 6] y_1 \\ & + [-60, -20] y_2 + [-6, -2] y_3, \\ \min_{y \geq 0} \quad & [0.5, 1.5] y_1 + [0.5, 1.5] y_2 + [1, 3] y_3, \\ \text{s.t.} \quad & -y_1 + y_2 + y_3 \leq \left[\frac{1}{2}, \frac{3}{2} \right], \\ & 2x_1 - y_1 + 2y_2 - 0.5y_3 \leq \left[\frac{1}{2}, \frac{3}{2} \right], \\ & 2x_2 + 2y_1 - y_2 - 0.5y_3 \leq \left[\frac{1}{2}, \frac{3}{2} \right]. \end{aligned} \quad (18)$$

Example 6. Let

$$\begin{aligned} \min_{1 \geq x \geq 0} \quad & [-150, -50] x + [-1500, -500] y_1, \\ \min_{y \geq 0} \quad & [-1.5, -0.5] y_1 + [-1.5, -0.5] y_2, \\ \text{s.t.} \quad & x + y_1 - y_2 \leq \left[\frac{1}{2}, \frac{3}{2} \right], \\ & y_1 + y_2 \leq \left[\frac{1}{2}, \frac{3}{2} \right]. \end{aligned} \quad (19)$$

Example 7. Let

$$\begin{aligned} \min_{x \geq 0} \quad & [0.5, 1.5] x + [-6, -2] y, \\ \min_{y \geq 0} \quad & [0.5, 1.5] y, \\ \text{s.t.} \quad & -x - y \leq [-4.5, -1.5], \\ & -2x + y \leq [0, 0], \quad 2x + y \leq [6, 18], \\ & 3x - 2y \leq [2, 6]. \end{aligned} \quad (20)$$

It is easily seen that the constructed examples are more challenging than the original ones since interval coefficients are involved.

The parameters are taken as follows: $N = 50$, the crossover probability $p_c = 0.8$, the mutation probability $p_m = 0.1$, $\sigma = 1$, and the maximum number of generations $G_m = 20$. EA-DBES is executed for 20 runs on a PC, and the best optimal solutions (x_b, y_b) as well as the worst optimal solutions (x_w, y_w) are recorded. In all 20 runs the best ones (F_b^b, F_w^b) , mean values (F_b^m, F_w^m) , and worst ones (F_b^w, F_w^w) of two classes of solutions are shown in Tables 1 and 2. In Tables 1 and 2, F_b and F_w represent the known best and worst optimal values, respectively. For Examples 1 and 2, F_b and F_w were provided by the existing algorithm [18], whereas for Examples 3–7, only the optimal values of the original bilevel programs are presented, which can provide upper bounds for the best optimal values of Examples 3–7 as well as lower bounds for the worst optimal values.

From Tables 1 and 2, one can see that EA-DBES found the same optimal results as those provided by the literature for

TABLE 1: Best optimal solutions provided by EA-DBES.

Number	F_b^b	F_b^m	F_b^w	(x_b, y_b)	F_b
Example 1	0	0	0	(0, 0, 0)	0
Example 2	0.5	0.5	0.5	(0, 0, 0.5)	0.5
Example 3	0	0	0	(0, 0, 0)	≤ 0
Example 4	0.25	0.25	0.25	(0, 0, 0.25)	≤ 0.5
Example 5	-65.7	-65.7	-65.7	(0, 1.35, 0, 0.9, 0.6)	≤ -29.2
Example 6	-2250	-2250	-2250	(0, 1.5, 0)	≤ -1000
Example 7	-40.14	-40.14	-40.14	(5.4, 7.1)	≤ -12

TABLE 2: Worst optimal solutions provided by EA-DBES.

Number	F_w^b	F_w^m	F_w^w	(x_w, y_w)	F_w
Example 1	2	2	2	(0, 2, 0)	2
Example 2	7	7	7	(0, 1, 2)	7
Example 3	3	3	3	(0, 3, 0)	≥ 0
Example 4	11.5	11.5	11.5	(0, 1.8, 3)	≥ 0.5
Example 5	-7.3	-7.3	-7.3	(0, 0.45, 0, 0.3, 0.2)	≥ -29.2
Example 6	-50	-50	-50	(1, 0, 1.3)	≥ -1000
Example 7	-1.25	-1.25	-1.25	(0.5, 1)	≥ -12

TABLE 3: Values of t and \bar{b} corresponding to the best and the worst results.

Number		Best	Worst
Example 1	t	(1.72, 1.39)	(-0.35, 2.59)
	\bar{b}	(1, 4, 2)	(1, 4, 2)
Example 2	t	(3.13, -1.63)	(2.01, -2.10)
	\bar{b}	(1, 4, 2)	(1, 4, 2)
Example 3	t	(0.58, 1.35)	(-0.32, 1)
	\bar{b}	(1.05, 5.58, 2.10)	(0.63, 6.00, 2.68)
Example 4	t	(2, -1)	(2, -3)
	\bar{b}	(0.5, 6, 1)	(0.5, 6, 3)
Example 5	t	(0.5, 1.5, 3)	(1.5, 0.5, 3)
	\bar{b}	(1.5, 1.5, 1.5)	(0.5, 0.5, 0.5)
Example 6	t	(-1.5, -0.5)	(-0.99, -1.07)
	\bar{b}	(1.5, 1.5)	(1.02, 1.35)
Example 7	t	(1.5)	(0.5)
	\bar{b}	(-1.5, 0, 18, 2)	(-1.5, 0, 6, 2)

Examples 1 and 2. For Examples 3–7, the best optimal values are not larger than the original ones, and the worst optimal values are not less than the original ones. Also, it should be noted that for each class of solutions we obtained the same optimal values in all 20 runs for each problem, which implies that the proposed EA-DBES is robust and feasible.

Consider that the results including the best and the worst optimal solutions can be got in one execution of EA-DBES, which is different from the existing approaches [18, 28].

The values of t and \bar{b} corresponding to the best and the worst optimal solutions are shown in Table 3. From the sensitivity analysis, it follows that both t and \bar{b} are not unique for these optimal solutions.

6. Conclusion

In this paper an interval linear bilevel programming problem is discussed, and a solution approach is presented. In the proposed algorithm, the search space of evolutionary algorithm consists of both the coefficient interval of the lower level objective and the right-hand-side vector interval, and, for each individual, the fitness function value is obtained by solving several linear programs. An advantage of the proposed algorithm is that it can simultaneously obtain the best and the worst optimal values in one run, which makes the proposed EA different from other existing approaches.

Conflict of Interests

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

Acknowledgments

The research work was supported by the National Natural Science Foundation of China under Grant no. 61065009, the National Social Science Fund of China under Grant no. 13BXW037, and the Natural Science Foundation of Qinghai Province under Grant no. 2013-z-937Q.

References

- [1] H. V. Stackelberg, *The Theory of the Market Economy*, Oxford University Press, 1952.
- [2] J. Bracken and J. T. McGill, "Mathematical programs with optimization problems in the constraints," *Operations Research*, vol. 21, pp. 37–44, 1973.
- [3] J. Bracken and J. T. McGill, "Defense applications of mathematical programs with optimization problems in the constraints," *Operations Research*, vol. 22, no. 5, pp. 1086–1096, 1974.

- [4] J. Bracken and J. T. McGill, "Production and marketing decisions with multiple objectives in a competitive environment," *Journal of Optimization Theory and Applications*, vol. 24, no. 3, pp. 449–458, 1978.
- [5] W. Candler and R. Norton, "Multilevel programming," Tech. Rep. 20, World Bank Development Research Center, Washington, DC, USA, 1977.
- [6] E. Aiyoshi and K. Shimizu, "Hierarchical decentralized systems and its new solution by a barrier method," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, no. 6, pp. 444–449, 1981.
- [7] B. Colson, P. Marcotte, and G. Savard, "Bilevel programming: a survey," *A Quarterly Journal of Operations Research (4OR)*, vol. 3, no. 2, pp. 87–107, 2005.
- [8] S. Dempe, *Foundations of Bilevel Programming*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002.
- [9] H. Abou-Kandil and P. Bertrand, "Government-private sector relations as a Stackelberg game: a degenerate case," *Journal of Economic Dynamics & Control*, vol. 11, no. 4, pp. 513–517, 1987.
- [10] H. I. Calvete, C. Galé, and M. Oliveros, "Bilevel model for production distribution planning solved by using ant colony optimization," *Computers & Operations Research*, vol. 38, no. 1, pp. 320–327, 2011.
- [11] A. Migdalas, "Bilevel programming in traffic planning: models, methods and challenge," *Journal of Global Optimization*, vol. 7, no. 4, pp. 381–405, 1995.
- [12] O. Ben-Ayed, D. Boyce, and C. Blair, "A general bilevel linear programming formulation of the network design problem," *Transportation Research B*, vol. 22, no. 4, pp. 311–318, 1988.
- [13] J. M. Arroyo and F. D. Galiana, "On the solution of the bilevel programming formulation of the terrorist threat problem," *IEEE Transactions on Power Systems*, vol. 20, no. 2, pp. 789–797, 2005.
- [14] J. F. Bard, *Practical Bilevel Optimization*, Kluwer Academic, Dordrecht, The Netherlands, 1998.
- [15] 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 C*, vol. 35, no. 2, pp. 221–232, 2005.
- [16] 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.
- [17] A. G. Mersha and S. Dempe, "Linear bilevel programming with upper level constraints depending on the lower level solution," *Applied Mathematics and Computation*, vol. 180, no. 1, pp. 247–254, 2006.
- [18] H. I. Calvete and C. Galé, "Linear bilevel programming with interval coefficients," *Journal of Computational and Applied Mathematics*, vol. 236, no. 15, pp. 3751–3762, 2012.
- [19] S. Chanas and D. Kuchta, "Multiobjective programming in optimization of interval objective functions: a generalized approach," *European Journal of Operational Research*, vol. 94, no. 3, pp. 594–598, 1996.
- [20] H. Ishibuchi and M. Tanaka, "Multiobjective programming in optimization of the interval objective function," *European Journal of Operational Research*, vol. 48, no. 2, pp. 219–225, 1990.
- [21] K. K. Lai, S. Y. Wang, J. P. Xu, S. S. Zhu, and Y. Fang, "A class of linear interval programming problems and its application to portfolio selection," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 6, pp. 698–704, 2002.
- [22] A. A. Molai and E. Khorram, "Linear programming problem with interval coefficients and an interpretation for its constraints," *Iranian Journal of Science and Technology A*, vol. 31, no. 4, pp. 369–390, 2007.
- [23] J. W. Chinneck and K. Ramadan, "Linear programming with interval coefficients," *Journal of the Operational Research Society*, vol. 51, no. 2, pp. 209–220, 2000.
- [24] M. Hladik, "Optimal value range in interval linear programming," *Fuzzy Optimization and Decision Making*, vol. 8, no. 3, pp. 283–294, 2009.
- [25] C. Oliveira and C. H. Antunes, "Multiple objective linear programming models with interval coefficients—an illustrated overview," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1434–1463, 2007.
- [26] M. Hladik, "Generalized linear fractional programming under interval uncertainty," *European Journal of Operational Research*, vol. 205, no. 1, pp. 42–46, 2010.
- [27] W. Li and X. Tian, "Numerical solution method for general interval quadratic programming," *Applied Mathematics and Computation*, vol. 202, no. 2, pp. 589–595, 2008.
- [28] A. Ren and Y. Wang, "An interval programming approach for bilevel linear programming problem with fuzzy random coefficients," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 462–469, June 2013.
- [29] 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.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

