

# An EDA for Solving Linear Fractional Bilevel Programming Problems

 Huafei Chen<sup>1,2</sup>, Hecheng Li<sup>1,\*</sup> and Jing Huang<sup>1</sup>
<sup>1</sup>School of Mathematics and Statistics, Qinghai Normal University, Xining 810008, China

<sup>2</sup>School of Mathematics and Statistics, Sichuan University of Science & Engineering, Zigong 643000, China

\*Corresponding author

**Abstract**—Heuristic algorithms have attracted great attention in dealing with bilevel programming problems in recent years. In this manuscript, a novel approach termed estimation of distribution algorithm (EDA) is developed to solve a special class of bilevel programming problem, the linear fractional bilevel programming problem. With this method, individuals are encoded with integer strings whose components are the column indices associated with the bases of the follower. Offspring is created from a distribution via modeling selected superior individuals. Numerical results have illustrated the feasibility and efficiency of the proposed algorithm.

**Keywords**—estimation of distribution algorithm; linear fractional bilevel programming problems; probability distribution; optimal solutions

## I. INTRODUCTION

Bi-level programming (BP) is a class of nested optimization which contains two level problems, the leader and the follower. In this model, the leader and the follower have their own objective and constraints. Considering all possible responses at the follower, the decisive variable first optimizes its own objective function independently at the leader. Afterwards, the follower selects his decision under the given decision of the leader. The general bilevel programming problem is formulated as [1]

$$\min_{(x,y) \in S} F(x,y) \quad (1)$$

where  $y \in \arg \min_{v \in S(x)} f(x,v)$ ,  $x \in R^{n_1}$ ,  $y \in R^{n_2}$ ,

$$F, f : R^n \rightarrow R, n = n_1 + n_2,$$

Common constraint region  $S \subset R^n$  and

$S(x) = \{y \in R^{n_2} : (x,y) \in S\}$  is the follower's feasible region.

Let  $S_1$  be the projection of  $S$  on the  $R^{n_1}$ .

For each  $x \in S_1$ , the follower is reduced to

$$\min f(x,y), s.t. y \in S(x) \quad (2)$$

The feasible region of the leader decision, called inducible region  $IR$ , is implicitly defined by the follower problem

$$IR = \{(x, y^*) : x \in S_1, y^* \in M(x)\} \quad (3)$$

where  $M(x)$  denotes the set of optimal solutions to (2).

In real life, plentiful applications of this type of model can be encountered, involving transportation, management and economics, engineering design, supply chain planning, principal-agent problems, and health insurance, etc. [5]. Thus, the resolution of the problem has received great attention of the interested researchers and they have proposed some efficient methods for the problem in the recent years.

Since the bilevel programming problem is non-convex and non-differentiable, it is intrinsically difficult to solve. Even the linear bilevel programming, the simplest type of BPs, has proved to be NP-hard [6]. More detailed discussion of the complexity issues on the linear bilevel programming problem can refer to [6]. Over the past decades, much progress has been made in the process of developing solution methods for BPs. Approaches can be mainly classified into the following categories [3]:

- Extreme point method for linear BPs
- Replace the follower problem with its K-K-T conditions or duality theories
- Heuristic algorithms

It is worthy of mentioning that heuristic algorithms have attracted much focus for BPs. Estimation of distribution algorithm (EDA) [7], as a novel meta-heuristic merging in recent few years, is one of the stochastic optimization techniques which explores the space of potential solutions via building and sampling explicit probabilistic models of promising candidate solutions.

In this manuscript, the linear fractional bilevel programming problem (LFBP) is considered, in which both objective

functions are linear fractional and the common constraint region is a polyhedron. Using the common notation in bilevel programming, the LFBP problem can be written as follows

$$\begin{cases} \min_{x,y} F(x,y) = \frac{c_{11}x + d_{11}y + e_{11}}{c_{12}x + d_{12}y + e_{12}} \\ \min_y f(x,y) = \frac{c_{21}x + d_{21}y + e_{21}}{c_{22}x + d_{22}y + e_{22}} \\ s.t. Ax + By \leq b \\ x, y \geq 0. \end{cases} \quad (4)$$

where  $F(x,y)$ ,  $f(x,y)$  are objective functions of the upper and lower problems and  $x \in R^{n_1}$ ,  $y \in R^{n_2}$  are decision variables of the upper and the lower level problem, respectively.

Also,  $c_{11}^T, c_{12}^T, c_{21}^T, c_{22}^T \in R^{n_1}$

$d_{11}^T, d_{12}^T, d_{21}^T, d_{22}^T \in R^{n_2}$

$A \in R^{q \times n_1}, B \in R^{q \times n_2}$

$e_{11}, e_{12}, e_{21}, e_{22} \in R, b \in R^q$

Applications of the LFBP cover inventory to sales, actual cost to standard cost, output per employee, debt to equity ratio, return on investment, risk assert to capital, foreign loans to total loans, etc. Up to now, many researchers have studied this kind of problem and obtained some results and interested researchers can refer to literatures [1,8-13].

This paper is organized as follows. We outline some basic definitions and assumptions in Section 2. In Section 3, estimation of distribution algorithm is briefly introduced. We propose the design of the algorithm based on EDA in Section 4 and present our algorithm in Section 5. Computational results are given in Section 6 and we finally conclude our approach in Section 7.

## II. DEFINITIONS AND ASSUMPTIONS

**Definition 1** A pair of  $(x, y)$  is feasible for the LFBP if  $(x, y) \in IR$ .

**Definition 2**  $(x^*, y^*)$  is the optimal solution to the LFBP if  $(x^*, y^*) \in IR$  and  $F(x^*, y^*) \leq F(x, y)$  for  $\forall (x, y) \in IR$ .

To ensure that (4) is well posed, we always assume:

- 1) The common constraint region  $S$  is nonempty and compact;
- 2) For all decisions taken by the leader, the follower has some room to respond;
- 3) The follower has unique  $y(x)$  for each fixed  $x \in S(X)$ , the projection of  $S$  onto the upper's decision space;
- 4) The rank of matrix  $B$  is  $q$ ;

In (4), inequality  $Ax + By \leq b$  can be transformed into an equality by adding slack variables. Without loss of generality,

we take  $Ax + By = b$  consisting of  $q$  equations. In addition, we also suppose denominators  $c_{i2}x + d_{i2}y + e_{i2}, i = 1, 2$ , are positive, it suffices to multiply (-1) by both numerator and denominator if necessary.

## III. ESTIMATION OF DISTRIBUTION ALGORITHM

Estimation of distribution algorithm (EDA), first introduced by Mühlenbein and Paaß [7], is a new class of evolutionary algorithms. Unlike GAs, there are neither crossover nor mutation operators in EDA. Instead, EDA captures global statistical information of the population through constructing a probabilistic model for the selected superior solutions obtained so far and generates new candidates via sampling the established probabilistic distribution. In general, EDA endures the following three phases until the termination criteria are met.

*Step 1* Select superior individuals from a population.

*Step 2* Estimate the probability distribution from the chosen individuals.

*Step 3* Generate new individuals (i.e., offspring) from the estimated distribution.

## IV. DESIGN OF ALGORITHM BASED ON EDA

### A. Initial Population

Choose  $q$  columns from  $B$ .

For any  $x = x_0$ , solve the following linear program iteratively

$$\begin{cases} \min c_{11}x_0 + d_{11}y + e_{11} \\ s.t. Ax_0 + By = b, y \geq 0 \end{cases} \quad (5)$$

which is equivalent to

$$\begin{cases} \min d_{11}y \\ s.t. By = b - Ax_0, y \geq 0 \end{cases} \quad (6)$$

During the process, we reserve the base which is feasible for the equality  $By = b - Ax_0$ . If there are not enough bases,  $x_0$  is replaced until a size of population, *popsize*, is reached. Individuals generated in such a way make sure their fitness small as possible since the denominator is positive.

### B. Evaluation of Fitness

We divide  $B$  into two parts, namely  $B = (\bar{B}, N)$ , in which  $\bar{B}$  is corresponding to the basic variables, while  $N$  to the non-basic ones. Similarly,  $y = (y_B, y_N)$ . For an individual  $l = (i_1, i_2, \dots, i_q)$ , we take  $\bar{B} = (B_{i_1}, B_{i_2}, \dots, B_{i_q})$ , thus,

$$(\bar{B}, N) \begin{pmatrix} y_{\bar{B}} \\ y_N \end{pmatrix} = b - Ax \quad (7)$$

We get  $y_{\bar{B}} = \bar{B}^{-1}(b - Ax)$  via taking  $y_N = 0$ . If

$$\begin{cases} \bar{B}^{-1}(b - Ax) \geq 0 \\ \pi \geq 0 \end{cases} \quad (8)$$

is satisfied, then  $y = (\bar{B}^{-1}(b - Ax), 0)^T$  is the optimal solution function of the lower level problem, which relies on  $\bar{B}$ . Here,

$$\begin{aligned} \pi &= v_0 \mu - u_0 v \\ u_0 &= d_{21\bar{B}} \bar{B}^{-1}(b - Ax) + c_{21} x + e_{21} \\ v_0 &= d_{22\bar{B}} \bar{B}^{-1}(b - Ax) + c_{22} x + e_{22} \\ \mu &= d_{21N} - d_{21\bar{B}} \bar{B}^{-1} N \\ v &= d_{22N} - d_{22\bar{B}} \bar{B}^{-1} N \end{aligned}$$

We substitute  $y$  into the objective function in the leader level,

$$\begin{aligned} \bar{F}(x) &= \frac{c_{11}^T x + d_{11}^T y + e_{11}}{c_{12}^T x + d_{12}^T y + e_{12}} \\ &= \frac{c_{11}^T x + d_{11\bar{B}}^T (\bar{B}^{-1}(b - Ax)) + e_{11}}{c_{12}^T x + d_{12\bar{B}}^T (\bar{B}^{-1}(b - Ax)) + e_{12}} \\ &= \frac{(c_{11}^T - d_{11\bar{B}}^T \bar{B}^{-1} A)x + d_{11\bar{B}}^T \bar{B}^{-1} b + e_{11}}{(c_{12}^T - d_{12\bar{B}}^T \bar{B}^{-1} A)x + d_{12\bar{B}}^T \bar{B}^{-1} b + e_{12}} \end{aligned}$$

Problem (1) can be turned into

$$\begin{cases} \min \bar{F}(x) \\ s.t. \bar{B}^{-1}(b - Ax) \geq 0 \\ \pi \geq 0 \\ x \geq 0 \end{cases} \quad (9)$$

The objective function value through the resolution of the problem (9) is taken as the fitness of the individual  $l$ , as well as a pair of  $(x', y')$  is obtained, where  $y' = (\bar{B}^{-1}(b - Ax), 0)$ . Note that the objective function in (9) is a fractional function, a transformation is needed before getting the objective function value of the problem (9).

### C. Selection of individuals

In this part,  $N$  superior individuals are selected, where  $N = [\text{popsize} * 50\%]$ , whose forms are as follows.

$$\begin{aligned} l^1 &: i_{11}, i_{12}, \dots, i_{1q}, \\ l^2 &: i_{21}, i_{22}, \dots, i_{2q}, \\ &\dots \\ l^N &: i_{N1}, i_{N2}, \dots, i_{Nq} \end{aligned}$$

in which  $i_{st} \in \{1, 2, \dots, m\} = \bar{\Omega}$ . We denote  $pop1$  by the above individuals  $\{l^1, l^2, \dots, l^N\}$ , there are  $N * q$  elements in the  $pop1$ . Taking the frequency of each element appears in the

individual into account, the probability of  $a_i$  is  $\frac{a_i}{N * q}$ , the one

for  $a_j$  not appear is 0. We choose individuals according to roulette wheel selection, when  $i_1$  is gotten for the first time, we discard it from  $\bar{\Omega}$ , its probability is shared by the rest elements in the subsequent selecting process. By parity of reasoning,  $q$  elements are chosen.

## V. PROPOSED ALGORITHM

In the following, an EDA for LFBP is presented.

*Step 1 (Initialization)* Firstly, an initialization population  $pop(0)$  with the size of  $pop\text{-size}$  is generated according to the problem (5), let  $t=0$ .

*Step 2 (Fitness evaluation)* Evaluate the fitness of each individual in  $pop(t)$  through the process of solving problem (9).

*Step 3 (Selection)* Sort the  $pop(t)$  in descending according to the fitness values and select  $N$  "better" individuals as  $O_1$ .

*Step 4 (Create probability distribution)* Compute the probability distribution for the selected individuals in  $O_1$ .

*Step 5 (Sample)* Based on the probability distribution in *Step 4*, a new population  $O_2$  containing  $(pop\text{-size} - N)$  individuals are generated. During this process, when the first element is chosen, the probability of the element is assigned to other elements not appear in each individual to share, repeat the process until  $(pop\text{-size} - N)$  is reached.

*Step 6 (Termination criterion)* Evaluate the fitness values of each individual in  $O_2$ , perform the local search operation on the best individual, if new individual is superior to the primary one, the latter is discarded, or else, retain the original one to the next generation. Let  $O = O_1 \cup O_2$ , if the termination criterion is met, stop, output the optimal solution; Otherwise, let  $t = t+1$ , return to *Step 2*.

Also, a major advantage of this algorithm is that we can deal with the linear fractional bilevel programming even though the leader's variable  $n$  is very large.

## VI. SIMULATION

With these experiments, we aim to show the efficiency of our method in terms of the quality of the solutions. The experiments were performed on a PC having AMD A4-3300M APU with Radeon(tm) HD Graphics 1.9 GHz. The algorithm codes were written in MATLAB R2010a.

**TABLE I. COMPUTATIONAL RESULTS ON EXAMPLES 1-5**

<i>TestProb.</i>	<i>Ind.</i>	<i>Run</i>	<i>Optimal solution</i>	<i>TT(s)</i>	<i>ET(s)</i>	<i>EI</i>
1	10	20	$(x, y)=(3,3)^T, F=1.285714.$	0.20202	0.05850	0.75
2	10	20	$(x,y)=(8,2,0)^T, F=2.666667.$	0.19890	0.02808	0.45
3	10	20	$(x,y)=(0.75,0.75,0,0,1)^T, F=0.081633.$	0.25116	0.03120	0.30
4	10	20	$(x, y)=(5,1)^T, F=2.$	0.24258	0.06396	0.55
5	10	20	$(x,y)=(0,0.9,0,0.6,0.4)^T, F=-29.2.$	0.23120	0.02749	0.35

We take  $N$  as the ceiling of 50% of *pop-size* in our experiments.

Example 1 [12]

$$\begin{cases} \max_{x \geq 0} \frac{x+2y}{x+y+1} \\ \max_{y \geq 0} \frac{2x+y}{2x+3y+1} \\ s.t. -x+2y \leq 3, 2x-y \leq 3, \\ x+y \geq 3. \end{cases}$$

The optimal solution  $(x, y) = (3, 3)^T$  with the value  $F = 1.285714.$

Example 2 [14]

$$\begin{cases} \max_{8 \geq x \geq 0} \frac{x}{y_1+y_2+1} \\ \max_{y \geq 0} \frac{y_1}{y_2+1} \\ s.t. x+y_1+y_2 \leq 10, \\ y_1 \leq 9, y_2 \leq 7. \end{cases}$$

The optimal solution  $(x, y) = (8, 2, 0)^T$  with the value  $F = 8/3.$

Example 3 [1]

$$\begin{cases} \max_{x,y} \frac{1+x_1-x_2+2y_2}{8-x_1-2y_1+y_2+5y_3} \\ \max_y \frac{1+x_1+x_2+2y_1-y_2+y_3}{6+2x_1+y_1+y_2-3y_3} \\ s.t. -y_1+y_2+y_3 \leq 1, \\ 2x_1-y_1+2y_2-0.5y_3 \leq 1, \\ 2x_2+2y_1-y_2-0.5y_3 \leq 1. \\ x, y \geq 0 \end{cases}$$

The optimal solution  $(x, y) = (0.75, 0.75, 0, 0, 1)^T$  with the value  $F = 0.0816.$

Example 4 [15]

$$\begin{cases} \min_{x_1, x_2} \frac{x_1+5x_2+2}{x_1+1} \\ \min_{x_2} \frac{3x_1+2x_2+8}{x_1+x_2+3} \\ s.t. x_1+x_2 \geq 2, -2x_1+x_2 \leq 2, x_1+2x_2 \leq 9, \\ x_1+x_2 \leq 6, x_1-x_2 \leq 4, x_1, x_2 \geq 0. \end{cases}$$

The optimal solution  $(x, y) = (5, 1)^T$  with the value  $F = 2.$

Example 5 [16]

$$\begin{cases} \min_{x,y} -8x_1-4x_2+4y_1-40y_2-y_3 \\ \min_y \frac{1+x_1+x_2+2y_1-y_2+y_3}{6+2x_1+y_1+y_2-3y_3} \\ s.t. -y_1+y_2+y_3 \leq 1, \\ 2x_1+y_1+2y_2-0.5y_3 \leq 1, \\ 2x_2+2y_1-y_2-0.5y_3 \leq 1, \\ x, y \geq 0. \end{cases}$$

The optimal solution  $(x, y)$  with the value  $F = -29.2.$

For all tested problems in Table I (five simple examples), we set pop-size be 10 and the algorithm stops when the iteration reaches 10. ‘Test Prob.’ stands for the tested problems, ‘Ind.’ means new individuals generated in each iteration, defined by the product of pop-size and iteration number, and ‘Run’ refers to the run times on each of problems. We also present the average time, CPU, of obtaining an optimal solution in 20 runs and the average value of the earliest CPU time, ET, finding the best solution, both of the two are in seconds. The last column ‘EI’ describes the average value of the iterations at which the optimal solution appears for the first time.

For each tested problem, 10 individuals are computed and run 20 times independently. Table I displays the indices containing the CPU time consumed in seconds by making all the iteration prescribed (TT), the CPU time in seconds invested in getting the best solution (ET) and the average values of earliest iteration (EI), for the five problems solved.

Table I gives the same optimal solutions as those provided in literatures, and it costs less in terms of the three indices TT, ET and EI.

## VII. CONCLUSION

In this paper, a novel estimation of distribution algorithm based on the optimality of linear fractional programming is developed for linear fractional bilevel programming problems. Experimental results illustrate the proposed algorithm is feasible and efficient for the LFBP.

## ACKNOWLEDGEMENTS

The research work was supported by the National Natural Science Foundation of China under Grant No. 61463045, the Natural Science Foundation of Qinghai Provincial under Grant No. 2018-ZJ-901, the Chunhui Project of Ministry of Education under Grant No. Z2017052 and Qinghai Key Laboratory of Internet of Things Project under Grant No. 2017-ZJ-Y21.

## REFERENCES

- [1] H. I. Calvete and G. Gale. Solving linear fractional bilevel problems [J]. *Operations Research Letters*, 2004, 32: 143-151.
- [2] J. F. Bard. *Practical bilevel optimization: Algorithms and applications* [M]. Dordrecht: Kluwer Academic Publishers, 1998.
- [3] B. Colson, P. Marcotte and G. Savard. An overview of bilevel optimization [J]. *Annals of Operations Research*, 2007, 153: 235-256.
- [4] S. Dempe. Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints [J]. *Optimization*, 2003, 52(3): 333-359.
- [5] G. M. Wang, Z. P. Wan and X. J. Wang. bibliography on bilevel programming [J]. *Advances in Mathematics*, 2007, 36(5): 513-529.
- [6] J. Bard. Some properties of the bilevel linear programming [J]. *Journal of Optimization Theory and Applications*, 1991, 32: 146-164.
- [7] H. Mühlenbein and G. Paaß. From Recombination of Genes to the Estimation of Distribution Part 1, Binary Parameter [C]. *Lecture Notes in Computer Science 1141, Parallel Problem Solving from Nature*, 1996, 178-187.
- [8] Thirwani and S. R. Arora. Bilevel linear fractional programming problem [J]. *Cahiers du zero*, 1993, 35: 135-149.
- [9] N. Malhotra and S. R. Arora. An algorithm to solve linear fractional bilevel programming problem via goal programming [J]. *OPSEARCH*, 2000, 37(1): 1-13.
- [10] Y. H. Liu and S. M. Hart. Charactering an optimal solution to the linear bilevel programming problem [J]. *European Journal of Operational Research*, 1994, 73(1): 164-166.
- [11] H. I. Calvete and G. Gale. On the quasi-concave bilevel programming problem [J]. *Journal of Optimization Theory and Applications*, 1998, 98(3): 613-622.
- [12] S. Mishra. Weighting method for bi-level linear fractional programming problems [J]. *European Journal of Operational Research*, 2007, 183: 296-302.
- [13] M. D. Toksar. Taylor series approach for bilevel linear fractional programming problem [J]. *Seluck Journal on Applied Mathematics*, 2010, 11(1): 63-69.
- [14] S. R. Arora, N. Malhotra and D. Thirwani. An algorithm for bilevel linear fractional program when the follower controls few variables [J]. *Indian Journal of Pure and Applied Mathematics*, 2002, 33(2): 227-235.
- [15] H. I. Calvete, G. Gale and P. M. Mateo. A genetic algorithm for solving linear fractional bilevel problems [J]. *Annals of Operations Research*, 2009, 166: 39-56.
- [16] Y. P. Wang, Y. C. Jiao and H. Li. An evolutionary algorithm for solving nonlinear bilevel programming based on a new constraint handing scheme [J]. *IEEE Transactions on System, Man and Cybernetics, Part C: Applications and reviews*, 2005, 35(2): 221-232.