An Effective Method for Optimality Test Over Possible Reaction Set for Maximin Solution of Bilevel Linear Programming with Ambiguous Lower-Level Objective Function

Puchit Sariddichainunta(✉) and Masahiro Inuiguchi

Osaka University, 1-3 Kanemachiyama, Toyonaka, Osaka 560-8531, Japan
puchit@inulab.sys.es.osaka-u.ac.jp, inuiguchi@sys.es.osaka-u.ac.jp

Abstract. A bilevel linear optimization problem with ambiguous lower-level objective requires a decision making under uncertainty of rational reaction. With the assumption that the ambiguous coefficient vector of the follower lies in a convex polytope, we apply the maximin solution approach and formulate it as a special kind of three-level programming problem. According to its property that the optimal solution locates on an extreme point, we adopt k-th best method to search the optimal solution equipped with tests for possible optimality, local optimality and global optimality of a solution. In this study, we propose an effective method to verify the rational reaction of the follower which is essential to all steps of optimality test. Our approach uses a relatively small memory to avoid repetition of possible optimality tests. The numerical experiments demonstrate our proposed method significantly accelerates the optimality verification process and eventually computes an optimal solution more efficiently.

Keywords: Bilevel linear optimization · Possibly optimal decision making · Maximin solution

1 Introduction

Bilevel linear programming problem (BLP) is an extension of the linear programming problem that consists of two levels of decision making stage [2]. It is known as a sequential game in non-cooperative game theory or a so-called Stackelberg game. In such game, the leader at the upper level chooses his strategy first, and then the follower at the lower level makes his own decision taking the leader's decision into consideration as given. Its applications are found useful in many research areas where the model used is a hierarchical optimization problem [6]; e.g., principal-agency problem in economics, optimal chemical equilibria, and irrigation water resource management.

In the conventional BLP [2,3,6], each decision maker is assumed to have complete information about the game. The leader and the follower can exactly observe each other's payoff function and strategy space. However, in realistic scenarios, the accurate information of the counterpart is not easily observed and

V.-N. Huynh et al. (Eds.): IUKM 2015, LNAI 9376, pp. 89–101, 2015.
DOI: 10.1007/978-3-319-25135-6_10

often contains ambiguity. For instance, the principal-agent problem between the regulator and the electricity generating company in the monopoly market is the case. The regulator as a policy maker wants to impose an optimal measure for the environmental issue and the energy security policy. While the regulator cannot have the complete information that describes the company motivation for its profit maximizing, the regulator has to develop their policy based on the indeterminate information of the company such as fuel price, demand for electrical power and so on. For the regulator, these information are imprecise and ambiguous due to the uncertainty, but are necessary to foresee the profit which motivates the rational action of the company. This class of principal-agent problem is much discussed in the area of contract theory [4]. Hence, the BLP with ambiguous coefficients has attracted high attentions and has been recently studied [1,5,16].

In our research [11], we propose another approach to solve the BLP problem with an ambiguous follower's objective function based on a maximin decision principle under uncertainty, and a solution algorithm based on k-th best method. We assume that the coefficients in the leader's objective function are precise as well as the coefficients in the constraints are crisp number. Although some research [5,16] set the coefficient vector of leader as imprecise parameters, their models constructed best-case and worst-case scenario of the leader's objective function which eventually fixed the coefficient vector to develop the solution method. Thus, this scenario construction is equivalent to assume that the leader's objective function is precise. On the contrary, the coefficients in the follower's objective function are the most concerned. We assume it to be ambiguous and can be represented by a convex polytope. Since those coefficients often depend on the setting of follower's problem that are not known well by the leader and the objective function reflecting the follower's decision may be unclear to the leader, we think that the BLP problem with an ambiguous follower's objective function coefficients is one of the crucial parts of the BLP problem with ambiguous coefficients.

In this paper, we provide an effective method to possibly conduct the optimality test that helps verifying the possible rational action of the follower. This optimality verification process is essential for both local and global optimality test by the definition. It turns out that the acceleration of our k-th best method is very promising. We have employed the numerical experiments in order to observe the run time efficiency of our proposed methods.

The organization of this paper is as follows. In Section 2, we describe the formulation of the problem. We also discuss some properties related to the formulation. Our proposed solution methods and theoretical backgrounds are provided in Section 3. In Section 4, we show the numerical experiments. Finally, conclusion and potential future works are discussed in Section 5.

2 Problem Formulation

Briefly we introduce our model setting for BLP with ambiguous objective function of the follower as well as the definitions and properties in this model. Our model focus on the maximin decision criteria for the leader's decision when he cannot observe the follower's objective function precisely. We assume the leader just ambiguously knows follower's objective coefficient vector to some extent. This uncertainty is represented by the convex polytope Γ. Moreover, the strategic effect from the sequential decision is incorporated to the model as a two-step rational decision in the constraints. This problem can be written as follows:

$$\begin{aligned} &\underset{\boldsymbol{x}}{\text{maximize}}\ \boldsymbol{c}_1^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{c}_2^{\mathrm{T}}\boldsymbol{y}, \\ &\text{subject to } \boldsymbol{x} \geq \boldsymbol{0}, \\ &\qquad \boldsymbol{y} \text{ is determined by the follower so as to} \\ &\qquad \underset{\boldsymbol{y}}{\text{maximize}} \quad \tilde{\boldsymbol{c}}_3^{\mathrm{T}}\boldsymbol{y},\ \exists \tilde{\boldsymbol{c}}_3 \in \Gamma = \{\boldsymbol{c}_3 \mid G\boldsymbol{c}_3 \leq \boldsymbol{g}\}, \\ &\qquad \text{subject to} \quad A_1\boldsymbol{x} + A_2\boldsymbol{y} \leq \boldsymbol{b}, \\ &\qquad\qquad\qquad\quad \boldsymbol{y} \geq \boldsymbol{0}, \end{aligned} \tag{1}$$

where $\boldsymbol{x} \in \mathbf{R}^p$ and $\boldsymbol{y} \in \mathbf{R}^q$ are the decision variable vectors of the leader and the follower, respectively, $\boldsymbol{c}_1 \in \mathbf{R}^p$, $\boldsymbol{c}_2 \in \mathbf{R}^q$, $\boldsymbol{b} \in \mathbf{R}^m$, $A_1 \in \mathbf{R}^{m\times p}$, and $A_2 \in \mathbf{R}^{m\times q}$ are constant vectors and matrices while $\tilde{\boldsymbol{c}}_3$ is the ambiguous coefficient vector, and $\Gamma \subseteq \mathbf{R}^q$ is a polytope defined by a matrix $G \in \mathbf{R}^{l\times q}$ and a vector $\boldsymbol{g} \in \mathbf{R}^l$.

We assume the feasible solution set $S = \{(\boldsymbol{x}, \boldsymbol{y}) \mid A_1\boldsymbol{x} + A_2\boldsymbol{y} \leq \boldsymbol{b},\ \boldsymbol{x} \geq \boldsymbol{0},\ \boldsymbol{y} \geq \boldsymbol{0}\}$ is bounded and nonempty. The strategy set of the leader and the follower in the feasible region S are defined respectively by $X(S) = \{\boldsymbol{x} \geq \boldsymbol{0} \mid \exists \boldsymbol{y} \geq \boldsymbol{0};\ A_1\boldsymbol{x} + A_2\boldsymbol{y} \leq \boldsymbol{b}\}$, and $S(\boldsymbol{x}) = \{\boldsymbol{y} \geq \boldsymbol{0} \mid\ A_2\boldsymbol{y} \leq \boldsymbol{b} - A_1\boldsymbol{x}\}$. Moreover, we assume that the possible range of ambiguous coefficient vector $\tilde{\boldsymbol{c}}_3$ is known in a bounded convex polyhedron defined by $\Gamma = \{\boldsymbol{c} \in \mathbf{R}^q \mid G\boldsymbol{c} \leq \boldsymbol{g}\}$ where $G \in \mathbf{R}^{r\times q}$ and $\boldsymbol{g} \in \mathbf{R}^r$. When the leader knows exactly the coefficient vector of the follower's objective function, he can understand the follower's rational reaction set, $\mathrm{Opt}(\boldsymbol{c}_3, \boldsymbol{x}) = \left\{\boldsymbol{y} \in S(\boldsymbol{x}) \mid \boldsymbol{c}_3^{\mathrm{T}}\boldsymbol{y} = \max_{\boldsymbol{z}\in S(\boldsymbol{x})} \boldsymbol{c}_3^{\mathrm{T}}\boldsymbol{z}\right\}$. In the case of ambiguous coefficient vector $\tilde{\boldsymbol{c}}_3$, the leader cannot know the follower's rational response exactly, but he can explore the follower's rational response in a larger region. Under the assumption that $\tilde{\boldsymbol{c}}_3 \in \Gamma$, we define the follower's possible reaction set by $\Pi S(\boldsymbol{x}) = \bigcup_{\boldsymbol{c}_3\in\Gamma} \mathrm{Opt}(\boldsymbol{c}_3, \boldsymbol{x})$.

According to the above situation, we further assume that the leader will consider the worst effect of the follower's response to his strategy, and rationalize his solution by adopting maximin criteria. Immediately, linear programming problem (1) is formulated as

$$(OP)\left\{\begin{array}{l} \underset{\boldsymbol{x}}{\text{maximize}}\ \boldsymbol{c}_1^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{c}_2^{\mathrm{T}}\boldsymbol{y} \\ \text{subject to } \boldsymbol{x} \geq \boldsymbol{0} \\ \quad \boldsymbol{y},\ \boldsymbol{c}_3 \text{ solves,} \\ (SP(x))\left\{\begin{array}{l} \underset{\boldsymbol{y},\boldsymbol{c}_3}{\text{minimize}}\ \boldsymbol{c}_2^{\mathrm{T}}\boldsymbol{y} \\ \text{subject to} \\ \quad \boldsymbol{c}_3^{\mathrm{T}}\boldsymbol{y} = \max\limits_{\boldsymbol{z}\geq\boldsymbol{0}}\{\boldsymbol{c}_3^{\mathrm{T}}\boldsymbol{z} \mid A_2\boldsymbol{z} \leq \boldsymbol{b} - A_1\boldsymbol{x}\} \\ \quad G\boldsymbol{c}_3 \leq \boldsymbol{g} \\ \quad A_1\boldsymbol{x} + A_2\boldsymbol{y} \leq \boldsymbol{b} \\ \quad \boldsymbol{y} \geq \boldsymbol{0} \end{array}\right. \end{array}\right.$$

The upper level problem is a maximization problem for the leader to decide his decision variable $\boldsymbol{x}$. The lower level is a minimization problem to obtain the optimal $\boldsymbol{y}$ and $\boldsymbol{c}_3$ which affect on the leader's objective function value at worst. We note that a maximization problem is included in the lower level problem which represents rationality of the follower. Thus, the problem (OP) can be seen as a three-level programming problem. Let us define the inducible region set as $IR = \{(\boldsymbol{x}, \boldsymbol{y}) \mid (\boldsymbol{x}, \boldsymbol{y}) \in S,\ \boldsymbol{y} \in \Pi S(\boldsymbol{x})\}$. The problem (OP) is rewritten as follows:

$$\begin{cases} \underset{\boldsymbol{x},\boldsymbol{y}}{\text{maximize}}\ \boldsymbol{c}_1^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{c}_2^{\mathrm{T}}\boldsymbol{y}, \\ \text{subject to } (\boldsymbol{x}, \boldsymbol{y}) \in IR. \end{cases} \tag{2}$$

3 Proposed Solution Methods

In this section we introduce a solution framework based on k-th best method. It consists of the vertex enumeration process and the optimality test process for possible optimality, local optimality and global optimality. The vertex enumeration is done partly and terminates when the global optimal solution is verified. In addition, we revisit some fundamental ideas of optimality and propose an effective method to reduce computation time for optimality test which consequently improve the overall algorithm efficiency. The basic idea of k-th best method, optimality definitions and test processes are explored in the following subsections.

3.1 K-th Best Method

The k-th best method is a vertex enumeration algorithm. It is a search algorithm starting from the first best solution and sequentially suggests the second best and so on, if the former solution does not satisfy to the optimality conditions. This method is useful when the optimal solution occurs at the vertex; i.e., the extreme point of the feasible region. The following theorem enhances k-th best method applicable to solve our model.

Theorem 1. *[11] The optimal solution of problem (OP) is located on a vertex of feasible region, S.*

Thus we proposed a solution procedure based on the k-th best method by checking the feasibility in the descending sequence of leader's objective function value. The following is the linear problem which ignores the lower-level problem.

$$\max\{\boldsymbol{c}_1^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{c}_2^{\mathrm{T}}\boldsymbol{y} \mid (\boldsymbol{x}, \boldsymbol{y}) \in S\} \tag{3}$$

$(\boldsymbol{x}^1, \boldsymbol{y}^1), \cdots, (\boldsymbol{x}^N, \boldsymbol{y}^N)$ denote the N ordered basic feasible solutions satisfying

$$\boldsymbol{c}_1^{\mathrm{T}}\boldsymbol{x}^k + \boldsymbol{c}_2^{\mathrm{T}}\boldsymbol{y}^k \geq \boldsymbol{c}_1^{\mathrm{T}}\boldsymbol{x}^{k+1} + \boldsymbol{c}_2^{\mathrm{T}}\boldsymbol{y}^{k+1},\ k = 1, \cdots, N-1 \tag{4}$$

According to the fact that there exists an optimal solution at an extreme point, solving problem (OP) is to find k^* which is the smallest index of extreme points in set IR.

$$k^* = \min\{k \in \{1, \cdots, N\} \mid (\boldsymbol{x}^k, \boldsymbol{y}^k) \in IR\} \tag{5}$$

Based on this method, we fix $\boldsymbol{x} = \boldsymbol{x}^k$ and then justify whether $\boldsymbol{y}^k$ is the optimal solution for problem $(SP(x))$. If so, we can conclude $(\boldsymbol{x}^k, \boldsymbol{y}^k) \in IR$. Because we assume S is bounded, we have a finite number of basic feasible solutions (BFS). Therefore, this procedure stops in a finite number of iterations. The procedure is written as follows:

[Algorithm 1]. The solution procedure for the problem (OP) based on k-th best algorithm

Step 1. $k = 1$. Solve problem (3) by simplex method, and get the solution $(\boldsymbol{x}^1, \boldsymbol{y}^1)$. Set $M = \{(\boldsymbol{x}^1, \boldsymbol{y}^1)\},\ T = \emptyset$.

Step 2. Fix $\boldsymbol{x} = \boldsymbol{x}^k$. Justify whether $\boldsymbol{y}^k$ is the optimal solution of $(SP(\boldsymbol{x}^k))$ or not. If it is the optimal solution of $(SP(\boldsymbol{x}^k))$, the solution is also the optimal solution of problem (OP) and terminate the procedure with the optimal solution $(\boldsymbol{x}^k, \boldsymbol{y}^k)$. Otherwise, go to step 3.

Step 3. Generate M^k, a set of adjacent extreme points of $(\boldsymbol{x}^k, \boldsymbol{y}^k)$, which satisfies $\boldsymbol{c}_1^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{c}_2^{\mathrm{T}}\boldsymbol{y} \leq \boldsymbol{c}_1^{\mathrm{T}}\boldsymbol{x}^k + \boldsymbol{c}_2^{\mathrm{T}}\boldsymbol{y}^k$. Then, $T = T \cup \{(\boldsymbol{x}^k, \boldsymbol{y}^k)\}$, $M = (M \cup M^k) \setminus T$.

Step 4. $k = k + 1$. Choose $(\boldsymbol{x}^k, \boldsymbol{y}^k) \in \arg\max_{\boldsymbol{x},\boldsymbol{y}}\{\boldsymbol{c}_1^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{c}_2^{\mathrm{T}}\boldsymbol{y} \mid (\boldsymbol{x}, \boldsymbol{y}) \in M\}$, and go to step 2.

Step 1, 3 and 4 of Algorithm 1 is the usual vertex enumeration process. On the other hand, step 2 is the optimality verification process of $\boldsymbol{y}^k$ to problem $(SP(\boldsymbol{x}^k))$, feasibility, local and global optimality. We set up these tests to avoid computational loads for the global optimality test and to aim at detecting inconsistent solutions to each optimality definition:

Definition 1. *Possible, local and global optimality*

Possible: $\boldsymbol{y} \in \Pi S(\boldsymbol{x}^k)$.

Local: $\boldsymbol{y}^k$ *s.t.* $\nexists\ \boldsymbol{y} \in N(\boldsymbol{y}^k) \cap \Pi S(\boldsymbol{x}^k)$ *and* $\boldsymbol{c}_2^{\mathrm{T}}\boldsymbol{y} < \boldsymbol{c}_2^{\mathrm{T}}\boldsymbol{y}^k$, *where* $N(\boldsymbol{y}^k)$ *is a set of all adjacent basic feasible solutions of* $\boldsymbol{y}^k$.

Global: $\boldsymbol{y}^k$ *s.t.* $\nexists\ \boldsymbol{y} \in \Pi S(\boldsymbol{x}^k)$ *and* $\boldsymbol{c}_2^{\mathrm{T}}\boldsymbol{y} < \boldsymbol{c}_2^{\mathrm{T}}\boldsymbol{y}^k$.

Those specific details of each optimality definition and test procedures are discussed in the next subsections.

3.2 Rational Reactions and Possible Optimality Test

As the leader needs to speculate the act of follower $\boldsymbol{y}^k$, it is necessary to check whether that $\boldsymbol{y}^k$ is a rational choice and a possible reactions of the follower, so-called, possible optimality or the feasibility of rational response. The feasibility of $\boldsymbol{y}^k$ is verified if there is a vector $c_3 \in G$ to advocate it as a rational reaction of the follower in problem $(SP(x^k))$. This is equivalent to that $\boldsymbol{y}^k$ is the member of possibly optimal solution set, noted by $\boldsymbol{y}^k \in \Pi S(\boldsymbol{x})$.

As shown by [9], the possibly optimal solution set equals to a weakly efficient solution set of a multiple objective linear programming problem. Since a weakly efficient solution set is connected and polyhedral [15], $\Pi S(\boldsymbol{x})$ is then connected and polyhedral. However, $\Pi S(\boldsymbol{x})$ is not a convex set, problem $(SP(\boldsymbol{x}))$ cannot be solved easily even when $\boldsymbol{x}$ is determined. Given $(\boldsymbol{x}, \boldsymbol{y}) \in S$, the feasibility of a solution $\boldsymbol{y}$ can be tested by the possibly optimal test proposed by [8]. The characterization of possible optimality test is described in the next theorem.

Theorem 2. *[11] The necessity and sufficient condition for $\boldsymbol{y} \in S(\boldsymbol{x})$ to be a possible reaction, i.e., $\boldsymbol{y} \in \Pi S(\boldsymbol{x})$, is given by the consistency of the following system of linear inequalities:*

$$\sum_{i=1}^{\tilde{m}} u_i \bar{\boldsymbol{a}}_i \in \Gamma, \ u_i \geq 0, \quad \forall i = 1, 2, \ldots, \tilde{m}, \tag{6}$$

where $\bar{\boldsymbol{a}}_i^{\mathrm{T}}$ is a row vector of $\begin{bmatrix} A_2 \\ -E \end{bmatrix}$ corresponding to the i-th active constraint of $A_2\boldsymbol{y} \leq \boldsymbol{b} - A_1\boldsymbol{x}$ and $\boldsymbol{y} \geq 0$, where E is an identity matrix and $\tilde{m}$ is the total number of active constraints.

The test result of the method following to theorem 2 can be reused for other BFSs that have the same set of basic variables. Intuitively, the information required for (6) are just the active normal vectors of binding constraints and $Gc_3 \leq g$. The former can be represented by a set of non-basic variables, and the latter is a necessary condition. To all the other iterations, any extreme points with the same set of basic/non-basic variables correspond to the same active constraints. We thus can identify whether the current BFS belongs to the possibly optimal solution set by considering the set of basic/non-basic variables of each solution.

Corollary 1. *Let I be a set of basic variables for a BFS. Every extreme point in the lower-level problem $(SP(x))$ represented by this set I retain the same result of possibly optimal test.*

Furthermore, some partial information from the test method by theorem 2 can be reused to avoid the repetition of solving the same linear programming. The solution of positive u_i preserves the feasibility of other $\boldsymbol{y}$ of which BFS contains all non-basic variables corresponding to those positive u_i. In other words, if one BFS has been tested and become a member of $\Pi S(\boldsymbol{x})$, the solution of positive u_i generates a subset of non-basic variables that correspond to normal vectors of a cone for c_3. Another BFS with such a subset included in its non-basic variable set is also possibly optimal since the same c_3 maintains the possible optimality for this BFS.

Corollary 2. *Let J be a set of non-basic variables such that it has a corresponding vector $\boldsymbol{u} > 0$ satisfies (6). The BFS that has all of those non-basic variables corresponding to positive u_i is also a possibly optimal solution.*

Subsequently, we found a simple method to test a single non-basic variables that always generate its corresponding positive u_i. The idea is based on the fact that u_i is a positive coefficient extending the normal vector of active constraint to relocate in the convex polyhedron Γ. The requirement is nothing but to find an appropriate scalar value of that normal vector enhancing its feasibility in Γ.

Corollary 3. *There is a positive coefficient u_i for a normal vector $\bar{\boldsymbol{a}}_i$ in matrix A_2 such that $u_i G\bar{\boldsymbol{a}}_i = u_i\hat{\boldsymbol{g}} \leq \boldsymbol{g}$ if and only if there is no such $g_t < 0 < \hat{g}_t$ for $t = 1, \ldots, q$ in $\boldsymbol{g}$ and $\hat{\boldsymbol{g}}$, and there is u_i such that $r^{'} \leq u_i \leq r^{''}$ where*
$r^{'} = \max\left\{\frac{g_t}{\hat{g}_t} > 0 \mid g_t, \hat{g}_t < 0\right\}$ *and* $r^{''} = \min\left\{\frac{g_t}{\hat{g}_t} > 0 \mid g_t, \hat{g}_t > 0\right\}$.

According to corollary 2 and 3, we can adopt the results of non-basic variable with respect to u_i to immediately verify possible optimality of other BFS which have those corresponding non-basic variables. The repetitive computations of solving LP for (6) to check possible optimality are avoidable. Indeed, we reuse the test results that passed the possibly optimal test according to Theorem 2. It requires some memory space only linear order with the upper bound of all possible full-rank combinations in the lower-level problem. The usage of memory storage provide a quick guide for evaluation of unchecked BFSs, and accelerate possible optimality verification in total.

3.3 Local Optimality Test

Local optimal solution is a solution with the property that there is no neighbor BFS which is possibly optimal and improves the objective value of the lower-level problem $(SP(x))$. We define it on the BFS so that the implementation is readily to operate the simplex algorithm. The procedure of local optimality tests is written as follows:

[Algorithm 2]. Local optimality for problem $(SP(\boldsymbol{x}^k))$.

Step 2-(a). Apply simplex method to confirm the existence of a solution satisfying equation (6). If it exists, we know $\boldsymbol{y}^k \in \Pi S(\boldsymbol{x}^k)$ and go to step 2-(b). If it does not exist, go to step 3.

Step 2-(b). For each adjacent basic solution $\boldsymbol{y}$ of the current basic solution corresponding to $\boldsymbol{y}^k$, we check $\boldsymbol{c}_2^{\mathrm{T}}\boldsymbol{y} < \boldsymbol{c}_2^{\mathrm{T}}\boldsymbol{y}^k$ and $\boldsymbol{y} \in \Pi S(\boldsymbol{x}^k)$ by the same way as in step 2-(a). If such an adjacent solution is founded, $\boldsymbol{y}^k$ is not locally optimal and go to step 3. Otherwise, proceed to the global optimality test, i.e., step 2-(c).

Step 2-(c). Test the global optimality of $\boldsymbol{y}^k$ to problem $(SP(\boldsymbol{x}^k))$. If the global optimality is verified, we found that $(\boldsymbol{x}^k, \boldsymbol{y}^k)$ is the optimal solution to problem (OP). Otherwise, go to step 3.

The degenerate BFS causes a serious concern in the verification process since it deters the pivoting process to move to another BFS which improves the objective value in the preferable direction, $-\boldsymbol{c}_2$. It usually occurs when the columns are reduced; in this model the leader's decision variables are eliminated in the lower-level problem. To reach a BFS in the preferable direction, we effectively operate the depth-first search in the BFS search space for local as well as global optimality test and expect for fast detection of non-optimal $\boldsymbol{y}^k$, [14].

3.4 Global Optimality Test

Following to the definition of global optimality, it is necessary to check all feasible solutions of problem $(SP(\boldsymbol{x}^k))$ to assure the maximin decision criteria. In this section, we adopt three algorithms for the global optimality test: the two step k-th best method [13], the enumeration of adjacent possibly optimal solutions [11] and the inner approximation method [7]. The first method is intuitive from the definition and the second method is developed following to the fact that possibly optimal solutions are connected [10]. On the other hand, the third method is an algorithm to approximate the global optimality which somewhat generates unstable results due to zero-rounding errors would probably occur.

First, the two step k-th best method for the global optimality test is a direct implementation, according to the global optimality definition, to find out that no other better solutions than y^k is a possible reaction of the follower. It sequentially enumerates all extreme points in $S(\boldsymbol{x})$. The computation could be costly if there are many extreme points to visit. The Algorithm 3A is a basis of comparison in the numerical experiments.

[Algorithm 3A]. Two step k-th best method

Step (a). $l = 1$ and $\boldsymbol{y}^{[1]} = \underset{\boldsymbol{y} \in S(\boldsymbol{x}^k)}{\operatorname{argmin}} \; \boldsymbol{c}_2^{\mathrm{T}} \boldsymbol{y}$. $\mathcal{E} = \emptyset, \mathcal{T} = \{\boldsymbol{y}^{[1]}\}$.

Step (b). Check $\boldsymbol{y}^{[l]} \in \mathit{\Pi}S(\boldsymbol{x}^k)$. If so, evaluate $\boldsymbol{c}_2^{\mathrm{T}} \boldsymbol{y}^{[l]} = \boldsymbol{c}_2^{\mathrm{T}} \boldsymbol{y}^k$. $\boldsymbol{y}^k$ is global optimal solution for $(SP(\boldsymbol{x}^k))$ when the equality is true, and terminate algorithm. If it is not equal, we conclude $\boldsymbol{y}^k$ is not global optimal solution, and go back to Algorithm 1.

Step (c). Let set $\mathcal{A}$ contain all adjacent BFS $\hat{\boldsymbol{y}}$ such that $\boldsymbol{c}_2^{\mathrm{T}} \hat{\boldsymbol{y}} \geq \boldsymbol{c}_2^{\mathrm{T}} \boldsymbol{y}^{[l]}$. $\mathcal{E} = \mathcal{E} \cup \{\boldsymbol{y}^{[l]}\}$, $\mathcal{T} = \mathcal{T} \cup (\mathcal{A} \setminus \mathcal{E})$.

Step (d). $l = l + 1, \quad \boldsymbol{y}^{[l]} = \underset{\boldsymbol{y}}{\operatorname{argmin}} \{\boldsymbol{c}_2^{\mathrm{T}} \boldsymbol{y} | \boldsymbol{y} \in \mathcal{A}\}$, and go to step (b).

Next, the adjacent enumeration of possibly optimal solution is implemented by using a valid inequality to restrict the feasible region of problem $(SP(\boldsymbol{x}^k))$ which we really concern. It eliminates the non-interesting area where $\boldsymbol{c}_2^{\mathrm{T}} \boldsymbol{y} \geq \boldsymbol{c}_2^{\mathrm{T}} \boldsymbol{y}^k$ holds. We then obtain $S^k(\boldsymbol{x}) = S(\boldsymbol{x}) \cap \{\boldsymbol{y} \mid \boldsymbol{c}_2^{\mathrm{T}} \boldsymbol{y} < \boldsymbol{c}_2^{\mathrm{T}} \boldsymbol{y}^k\}$, and the global optimality condition of $\boldsymbol{y}^k$ becomes $S^k(\boldsymbol{x}) \cap \mathit{\Pi}S(\boldsymbol{x}) = \emptyset$. We set $\varepsilon = 10^{-4}$ for $S^k_\varepsilon(\boldsymbol{x}) = S(\boldsymbol{x}) \cap \{\boldsymbol{y} \mid \boldsymbol{c}_2^{\mathrm{T}} \boldsymbol{y} \leq \boldsymbol{c}_2^{\mathrm{T}} \boldsymbol{y}^k - \varepsilon\}$, instead of $S^k(\boldsymbol{x})$.

[Algorithm 3B]. Enumeration of adjacent possibly optimal solutions

Step (a). Initialize the unchecked basic feasible solution set $\mathcal{U}$ and checked basic feasible solution set is $\mathcal{E}$ as empty sets. Choose a vector $\hat{\boldsymbol{c}}_3 \in \mathit{\Gamma}$, and solve

$\max_{\boldsymbol{y}\in S_\varepsilon^k(\boldsymbol{x})} \hat{\boldsymbol{c}}_3^{\mathrm{T}}\boldsymbol{y}$. If it is infeasible, we terminate the algorithm concluding that $\boldsymbol{y}^k$ is a global optimal solution of problem $(SP(\boldsymbol{x}^k))$. Otherwise, let $\hat{\boldsymbol{y}}$ be the obtained solution and go to step (b).

Step (b). Check the membership of $\hat{\boldsymbol{y}}$ to $\Pi S(\boldsymbol{x})$ by testing the consistency of (6) with respect to $\hat{\boldsymbol{y}}$. If $\hat{\boldsymbol{y}} \in \Pi S(\boldsymbol{x})$, we terminate the algorithm concluding that $\boldsymbol{y}^k$ is not a global optimal solution of problem $(SP(\boldsymbol{x}^k))$. Otherwise, update $\mathcal{E} = \mathcal{E} \cup \{\hat{\boldsymbol{y}}\}$ and go to step (c).

Step (c). Generate all adjacent basic feasible solutions which are members of $\Pi S_\varepsilon^k(\boldsymbol{x})$. Let $\mathcal{D}$ be the set of those solutions. Update $\mathcal{U} = \mathcal{U} \cup (\mathcal{D} \setminus \mathcal{E})$. If $\mathcal{U} = \emptyset$, we terminate the algorithm concluding that $\boldsymbol{y}^k$ is a global optimal solution of problem $(SP(\boldsymbol{x}^k))$.

Step (d). Select a basic feasible solution $\hat{\boldsymbol{y}}$ from $\mathcal{U}$ and update $\mathcal{U} = \mathcal{U} \setminus \{\hat{\boldsymbol{y}}\}$. Go back to step (b).

Finally, the inner approximation method exerts the polyhedral annexation technique [7]. It has been developed for the reverse convex optimization, and then applied to BLP which possesses the similar structure. This method transforms BLP into the quotient space which has lower dimensions, depending mainly on the number of independence constraints of the lower-level problem. It does not directly evolve all the problem $S^k(\boldsymbol{x})$ at once, but gradually approximate the solution from partially a necessary interior set.

The global optimality is to verify whether $\bar{S}^k(\boldsymbol{x}) \subseteq \mathrm{int}\bar{W}(\boldsymbol{x})$, where $\bar{S}^k(\boldsymbol{x}) = \{\boldsymbol{u} \mid \bar{A}^N\boldsymbol{u} \le \tilde{\boldsymbol{b}},\ \boldsymbol{u} \ge \boldsymbol{0}\} \cap \{\boldsymbol{u} \mid \boldsymbol{c}_2^{\mathrm{T}}(\boldsymbol{y}^0 + \xi\boldsymbol{u}) < \boldsymbol{c}_2^{\mathrm{T}}\boldsymbol{y}^k\}$ and $\bar{W}(\boldsymbol{x}) = \{\boldsymbol{u} \mid \boldsymbol{c}_3^{\mathrm{T}}(\boldsymbol{y}^0 + \xi\boldsymbol{u}) \le \max\limits_{\boldsymbol{z}\in S(\boldsymbol{x})} \boldsymbol{c}_3^{\mathrm{T}}\boldsymbol{z},\ \forall \boldsymbol{c}_3 \in \Gamma\}$. In practice, we construct $P_j = \{\boldsymbol{u} \ge \boldsymbol{0} \mid \boldsymbol{t}^{\mathrm{T}}\boldsymbol{u} \le 1,\ \forall \boldsymbol{t} \in V_j\}$ for $V_j \subset \mathbb{R}^q$, and observe whether $P_1 \subset P_2 \subset \cdots \subset \bar{W}(\boldsymbol{x})$. $\boldsymbol{t}^j$ is the normal vector of simplex P_j. It provides the recession direction $\boldsymbol{u}^j$ that updates P_{j+1} when there is $\boldsymbol{t} \in V_j$ such that $\nu(\boldsymbol{t}) \ge 1$. The initial simplex $P_1 = \mathrm{co}\{\boldsymbol{0}, \mu^1\boldsymbol{e}_1, \mu^2\boldsymbol{e}_2, \cdots, \mu^q\boldsymbol{e}_q\}$ is constructed by μ^i the solution of the following problem:

$$\begin{cases} \underset{\boldsymbol{v},\tau}{\text{maximize}}\ \tau \\ \text{subject to } A_2(-G^{\mathrm{T}}\boldsymbol{v} + \boldsymbol{y}^0 + \xi\tau\boldsymbol{u}^j) \le \boldsymbol{b} - A_1\boldsymbol{x} \\ \qquad\qquad -G^{\mathrm{T}}\boldsymbol{v} + \boldsymbol{y}^0 + \xi\tau\boldsymbol{u}^j \ge \boldsymbol{0} \\ \qquad\qquad \boldsymbol{g}^{\mathrm{T}}\boldsymbol{v} \le 0 \\ \qquad\qquad \boldsymbol{v} \ge \boldsymbol{0},\ \tau \ge 0 \end{cases} \tag{7}$$

[Algorithm 3C]. Inner approximation method

Initalization: Solve $\boldsymbol{y}^0 = \underset{\boldsymbol{y}\in S(\boldsymbol{x})}{\mathrm{argmin}}\, \boldsymbol{c}_2^{\mathrm{T}}\boldsymbol{y}$ and apply possible optimality test.

If feasible, $\boldsymbol{y}^k$ is not global optimal. Go back to algorithm 1.

If not, we construct $\bar{S}_\varepsilon^k(\boldsymbol{x})$, $\bar{W}(\boldsymbol{x})$ and solve for $\mu^i, i = 1, \cdots, q$.

If $\mu^i\boldsymbol{e}_i \cap \bar{S}_\varepsilon^k(\boldsymbol{x}) \ne \emptyset$, then $\boldsymbol{y}^k$ is global optimal and go back to algorithm 1.

If pass the above requirements, we find $\boldsymbol{t}^0$ from μ^i. Set $j \leftarrow 1$ to initialize $N_1 = \{\boldsymbol{0}\}$, $V_1 = \emptyset$, $\Lambda_1 = \mathbf{R}_+^q$.

Step (a). For every $\lambda \in N_j$, compute $\nu(\boldsymbol{\lambda})$. $\nu(\boldsymbol{\lambda}) = \max\limits_{\boldsymbol{\lambda}}\{(\boldsymbol{t}^0 - \boldsymbol{\lambda})^{\mathrm{T}}\boldsymbol{u} \mid \boldsymbol{u} \in \bar{S}_\varepsilon^k(\boldsymbol{x})\}$. Remove $\boldsymbol{\lambda}$ from N_j such that $\nu(\boldsymbol{\lambda}) < 1$. If V_j and N_j are both empty, then $\boldsymbol{y}^k$ is global optimal. Terminate the algorithm.

Step (b). Set $V_j \leftarrow V_j \cup N_j$. Find $\boldsymbol{\lambda}^j \in \arg\max_{\boldsymbol{\lambda}}\{\nu(\boldsymbol{\lambda}) \mid \boldsymbol{\lambda} \in V_j\}$ and $\boldsymbol{u}^j \in \arg\max_{\boldsymbol{u}}\{(\boldsymbol{t}^0 - \boldsymbol{\lambda}^j)^{\mathrm{T}}\boldsymbol{u} \mid \boldsymbol{u} \in \bar{S}^k_\varepsilon(\boldsymbol{x})\}$

Step (c). Check $\boldsymbol{u}^j \in \partial\bar{W}(\boldsymbol{x})$. If $\boldsymbol{u}^j \in \partial\bar{W}(\boldsymbol{x})$, $\boldsymbol{y}^k$ is not global optimal. Go back to algorithm 1. If $\boldsymbol{u}^j \notin \partial\bar{W}(\boldsymbol{x})$, solve θ^j which is the optimal value of (7).

Step (d). Use a cutting hyperplane, $\Lambda_{j+1} = \Lambda_j \cap \{\boldsymbol{\lambda} \mid (\boldsymbol{t}^0 - \boldsymbol{\lambda})^{\mathrm{T}}\boldsymbol{u}^j \leq \frac{1}{\theta^j}\}$. For all extreme points in Λ_{j+1}, there are some new extreme points generated by the cut. Put them into N_{j+1}, and separate the others to V_{j+1}.
Update $j \leftarrow j + 1$ and go back to Step (a).

4 Numerical Experiments

We evaluate the performance of our proposed method on randomly generated problems in the following way. m-p-q-r is the parameter of our generated problems. m is the number of constraints of feasible set S. p and q are the numbers of decision variables of the leader and of the follower respectively. r is the number of constraints in Γ.

4.1 Problem Generation

To generate a set of feasible region S, we derive m tangent hyperplanes from the surface of a unit hypersphere in the positive coordinate plane [12]. $\boldsymbol{r}_1 \in \mathbf{R}^{p+q}$ is a uniform random vector for the hypersphere equation, $\frac{\boldsymbol{r}_1^{\mathrm{T}}}{|\boldsymbol{r}_1|}\begin{pmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{pmatrix} \leq 1$. $|\boldsymbol{r}_1|$ means the norm of uniformly random vector $\boldsymbol{r}$. After that, we perturb these hyperplane by using a random vector $\boldsymbol{r}_2 \in \mathbf{R}^{p+q}$ of which each element randomizes between $[1, 3]$. The coordinate is converted as a direct multiplication, $(\hat{x}, \hat{y}) = \boldsymbol{r}_2 \otimes (\boldsymbol{x}, \boldsymbol{y})$. Now we have a convex set S as a feasible region defined by m tangent hyperplanes of a random ellipsoid.

The coefficient vectors $\boldsymbol{c}_1 \in \mathbf{R}^p$, $\boldsymbol{c}_2 \in \mathbf{R}^q$ is a random vector whose elements are uniformly random in $[1, 4]$. The set Γ defined by q constraints is also generated similarly to S. The further process is that we move the center of a hypersphere out of the origin and add another q parallel tangent hyperplanes to create a new convex polytope without the axes.

4.2 Numerical Results

The numerical experiments are set up to compare the effect before and after implementing the reuse of possible optimality test results for each global optimality test method. The base line algorithms are described in subsection global optimality: Algorithm (3A), (3B) and (3C). The programs are developed in Microsoft Visual C/C++ 2013 and run the experiments in a desktop computer (OS: Windows 8.1, CPU:3.4 GHz, RAM:8GB). The epsilon value 10^{-4} is used.

The numerical experiments aim to compare the improvement of efficiency in each method. We measure the CPU run time used in achieving the global optimal

Table 1. Results of experiment (CPU time (s))

Case	5-5-8-20			10-5-8-20			15-5-8-20		
Method	A	B	C	A	B	C	A	B	C
1st	22%	19%	35%	8%	12%	13%	2%	3%	10%
Mean	0.090	0.097	0.067	2.450	2.261	1.866	29.442	23.472	20.438
S.D.	0.105	0.118	0.072	2.735	3.141	3.208	33.671	29.527	26.081
Min	0	0	0	0	0	0	0	0	0
Max	0.562	0.609	0.375	10.125	13.954	27.877	136.775	128.009	113.227
Median	0.054	0.062	0.046	1.062	0.843	0.656	15.423	11.329	10.079
Method	A'	B'	C'	A'	B'	C'	A'	B'	C'
1st	63%	41%	73%	31%	50%	49%	14%	48%	32%
Mean	0.050	0.059	0.050	1.571	1.583	1.673	23.967	21.785	20.235
S.D.	0.049	0.058	0.053	1.753	2.058	3.028	28.876	28.503	26.332
Min	0	0	0	0	0	0	0	0.015	0
Max	0.250	0.250	0.328	6.672	8.891	26.798	113.742	124.228	113.899
Median	0.031	0.046	0.031	0.758	0.625	0.586	10.782	9.774	9.649
Ratio	55%	44%	43%	67%	50%	45%	73%	52%	45%

solution. 100 problems are generated for 3 cases: 5-5-8-20, 10-5-8-20, and 15-5-8-20 to observe the efficiency varied by the number of constraints which represents the size of problem. Algorithm (3A), (3B), and (3C) without the reuse strategy are represented by A, B, and C, otherwise denoted by A', B' and C'.

Table 1 reports the statistics of CPU run time in seconds. The row of 1st is the proportion of problem sets which the method can finish at the lowest CPU run time among 6 methods. It is derived from the number of problems it can finish first over the number of all problems. The sum of those percentage of each method in the same problem set dose not equal to one because there are some problems that more than one method finish at the same time. The ratio at the bottom of the table is a proxy represents the average rate of efficiency improvement in each problem set. It depicts how our proposed method for possible optimality test has saved time from the duplication of possible optimality test. It is calculated by the frequency of reusing test results over the frequency of test inquiries. The higher scale of ratio, the more efficiency gained.

The result in Table 1 illustrates an empirical evidence for the validity of our proposed method which save some computational time in the possible optimality test process. When the problem size increases, the ratio in column A (the two step k-th best method) also positively changes as the larger problem size encounters with more inquiries of the possible optimality test, especially for the two step k-th best method. Although the ratio for method B and C are not very large because these methods are designed to check fewer BFSs at the lower-level problem. However, the proportion is still notable.

Considering before and after implement our reuse strategy, method C (the inner approximation) seems to be the fastest finisher following by B and A according to the mean run time. In general, the computational time of method A' and B' come closed to the computational time of method C' when deploy the reuse strategy. Only the case 10-5-8-20, it seems that method B' win over method C', but the difference is not statistically significant as the dependent sample t-test is applied with the result: t-value 0.412 and two-sided probability 0.367. Following to the percentage of the 1st row, method B' has beaten method C' for the problem set 10-5-8-20 and 15-5-8-20. The result implies that the reuse strategy effectively improve computational time of method B in many types of problems. The further investigation in problem structure is required.

5 Conclusion

After we introduced the minimax decision model and solution method for bilivel linear programming with ambiguous lower-level objective function, we explained some theoretical background to support our effective method for the possible optimality test. The efficiency improvement in each setting in the numerical result is because the utilization of memory follows by our theoretical observations in possible optimality test.

The future work could be in several directions. For example, the improvement for local search in both local and global optimality tests are at our concern. The comparison with the commercial package that can solve BLP as a complementary optimization problem is also of interest.

References

1. Abass, S.A.: An Interval Number Programming Approach for Bilevel Linear Programming Problem. International Journal of Management Science and Engineering Management **5**(6), 461–464 (2010)
2. Bard, J.F.: Practical Bilevel Optimization: Algorithms and Applications. Kluwer Academic Publishers, Dordrecht (1998)
3. Bialas, W., Karwan, M.: On Two-level Optimization. IEEE Transactions on Automatic Control **27**, 211–214 (1982)
4. Bolton, P., Dewatripont, M.: Contract Theory. MIT Press, Cambridge (2005)
5. Calvete, H.I., Galé, C.: Linear Bilevel Programming with Interval Coefficients. Journal of Computational and Applied Mathematics, 3751–3762 (2012)
6. Dempe, S.: Foundations of Bilevel Programming. Kluwer Academic Publishers, Dordrecht (2002)
7. Horst, R., Tuy, H.: Global Optimization: Deterministic Approaches, 3rd edn. Springer-Verlag, Berlin (1995)
8. Inuiguchi, M., Sakawa, M.: Possible and Necessary Optimality Tests in Possibilistic Linear Programming Problems. Fuzzy Sets and Systems **67**, 29–46 (1994)
9. Inuiguchi, M., Kume, Y.: Minimax Regret in Linear Programming Problems with an Interval Objective Function. In: Tzeng, G.H., Wang, H.F., Wen, U.P., Yu, P.L. (eds.) Multiple Criteria Decision Making, pp. 65–74. Springer, New York (1994)
10. Inuiguchi, M., Tanino, T.: Enumeration of All Possibly Optimal Vertices with Possible Optimality Degrees in Linear Programming Problems with A Possibilistic Objective Function. Fuzzy Optimization and Decision Making **3**, 311–326 (2004)

11. Inuiguchi, M., Sariddichainunta, P., Kawase, Y.: Bilevel linear programming with ambiguous objective function of the follower - formulation and algorithm. In: Proceeding of the 8th International Conference on Nonlinear Analysis and Convex Analysis, pp. 207–217. Yokohama Publishers, Yokohama (2013)
12. Muller, M.E.: A Note on A Method for Generating Points Uniformly on N-Dimensional Sheres. Communications of the ACM **2**(4), 19–20 (1959)
13. Nishizaki, I., Sakawa, M.: Solution concepts and their computational methods in multiobjective two-level linear programming problems. In: Proceeding of 1999 IEEE International Conference on Systems, Man and Cybernetics, vol. 3, pp. 985–990. IEEE, Tokyo (1999)
14. Sariddichainunta, P., Inuiguchi, M.: The Improvement of Optimality Test over Possible Reaction Reaction Set in Bilevel Linear Optimization with Ambiguous Objective Function of the Follower. Journal of Advanced Computational Intelligence and Intelligent Informatics **19**(5) (2015) (forthcoming)
15. Steuer, R.E.: Multiple Criteria Optimization: Theory, Computation, and Application. John Wiley and Sons, New York (1986)
16. Ren, A., Wang, Y.: A Cutting Plane Method for Bilevel Linear Programming with Interval Coefficients. Annals of Operations Research, online publication (2014)