Atlantis Studies in Uncertainty Modelling, volume 3
Joint Proceedings of the 19th World Congress of the International Fuzzy Systems Association (IFSA), the 12th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT), and the 11th International Summer School on Aggregation Operators (AGOP)

A Minimax Criterion Approach to Treat the Inexactness in Feasible Set of a Linear Programming Problem

Zhenzhong Gao[a] and **Masahiro Inuiguchi**[b]
[a,b]Department of Systems Innovation
Graduate School of Engineering Science, Osaka University
Toyonaka, Osaka 560-8531, Japan
[a]zhenzhong@inulab.sys.es.osaka-u.ac.jp
[b]inuiguti@sys.es.osaka-u.ac.jp

Abstract

The robustness analysis investigates the optimality of a solution in a linear programming problem that contains uncertainties. Conventionally, researchers concentrated on the one with a fixed feasible set, while in this paper, we focus on an inexact one. Since an inexact feasible set usually makes a solution infeasible in some situations, we consider it a penalty for the objective function. To accomplish it, we utilise the minimax criterion and propose an updated programming problem, which has an objective function that includes a weighted L^p-norm to represent the penalty. For the application, we only consider L^1-norm (absolute-value norm) and L^2-norm (Euclidean norm). We show that for the L^1-norm, an approach based on linear programming can treat it with low computational complexity. In contrast, we show that an approach based on derivative can treat the case of L^2-norm. Finally, we compare our approach with conventional fuzzy linear programming for the merit and drawback of our approach.

Keywords: Inexact Linear Programming, Interval Linear Programming, Robustness Analysis, Inexact Feasible Set, Minimax Criterion, Weighted L^p-Norm

1 Introduction

Inexact linear programming is to treat the linear programming problem containing uncertainty in the coefficients. In such condition, a decision-maker needs to consider all kinds of situations to have a solution that gains most and loses least simultaneously. For example, in the problem of production planning, if the supply of raw materials is only known roughly with an approximated amount, a decision-maker has to consider any possible situation to guarantee a maximum profit with the slightest regret. However, the result usually varies as different decision-makers have different preferences, which one can regard as trade-offs in the problem.

By reviewing the past researches, one can find numerous methods based on different areas to treat the inexactness, such as probabilistic model [12, 11, 10], fuzzy linear programming [6, 7], stochastic linear programming [9] and interval linear programming [13, 5]. The key that a decision-maker holds is the way to gain more with little suffering. Namely, they need to know what is the preferred trade-off.

At dawn of the study, researchers found something interesting in the structure of a basic optimal solution, called the *basis*. They separated the solution by it, which derived an initial and intuitive approach called *sensitivity analysis* [1]. It utilised a factor called *shadow prices* to analyse the maximum perturbation on a single coefficient such that the basis would remain. Moreover, a unique convex cone [13] was found, which guarantee the basis to be unchanged. Such a result set the foundation for further study of multiple perturbations.

Trying to treat the single restriction, Wendell [13] proposed the *tolerance approach* that allows multiple ones by founding an interval hyper-box in the convex cone. Based on his contribution, Wondolowski [14] and Filippi [2] modified the original approach and gave excellent results. Then, by assuming the feasible set of a linear programming problem to be fixed, researchers have done enormous research. For example, Hladík [4, 5] came up with an approach to calculate the tolerance for the condition of multi-objective linear programming. Inuiguchi and Sakawa [7] made up the concept of *possible and necessary optimality* to describe an optimal solution.

However, the feasible set of a problem cannot be known precisely in reality, which causes infeasibility of a solution. To treat such problem, Garajová and Hladík [3] analysed the possible optimal solution set and gave their approach to approximate it. Inuiguchi [6] utilised fuzzy number and h-level set to level the optimality of a solution and its final objective value. Nevertheless, their approaches are too complicated for application, and the results are usually a solution set instead of a specific solution. It may puzzle a decision-maker in picking the choice.

In this paper, we concentrate on the linear programming problem with an inexact feasible set. To treat it, we consider an approach based on the minimax criterion. Like the original minimax regret criterion in interval linear programming [8], we modified the objective function by including the penalty caused by infeasibility.

To represent the penalty, we utilise the concept of L^p-norm and add weights to coefficients for representing different penalties. For the application, we focus on the study of L^1 and L^2 weighted norm and give the approaches, respectively. Moreover, we compare our approach with conventional fuzzy linear programming. Finally, we would give numerical examples with both conditions.

2 Preliminaries

Before proposing our approach, it is essential to give a brief and relevant introduction concerning with inexact linear programming.

2.1 Linear programming

The general linear programming problem that we address is in the following form.

Problem 1 *A linear programming problem in standard form is expressed as:*

$$\min\ \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x},\ \ s.t.\ A\boldsymbol{x}=\boldsymbol{b},\ \boldsymbol{x}\geq\boldsymbol{0}, \tag{1}$$

where $A\in\mathbb{R}^{m\times n},\boldsymbol{b}\in\mathbb{R}^m,\boldsymbol{c}\in\mathbb{R}^n$ denote the coefficients. $\boldsymbol{x}\in\mathbb{R}^n$ is a solution which needs to be solved.

According to the Simplex method, we have the following proposition to describe a *basic optimal* solution for Problem 1:

Proposition 1 *A feasible solution $\boldsymbol{x}^*\in\mathbb{R}^n$ is a* basic optimal *one if and only if it satisfies the following conditions:*

$$\boldsymbol{c}_N^*-N^{\mathrm{T}}B^{-\mathrm{T}}\boldsymbol{c}_B^*\geq\boldsymbol{0}\ \textit{and}\ B^{-1}\boldsymbol{b}\geq\boldsymbol{0}, \tag{2}$$

where $B\in\mathbb{R}^{m\times m}$ and $N\in\mathbb{R}^{m\times(n-m)}$ refer to the basic and non-basic sub-matrix of A, respectively. B should be non-singular to guarantee the existence of the inverse. Subsequently, $\boldsymbol{c}_B^*\in\mathbb{R}^m$ and $\boldsymbol{c}_N^*\in\mathbb{R}^{n-m}$ refer to the basic and non-basic part of $\boldsymbol{x}^*$, respectively. As the result, the optimal solution is $\boldsymbol{x}_B^*=B^{-1}\boldsymbol{b}$, $\boldsymbol{x}_N^*=\boldsymbol{0}$ and the optimal value equals to $\boldsymbol{c}_B^{\mathrm{T}}B^{-1}\boldsymbol{b}$.

We can find that Proposition 1 implies the existence of a *basis* in a basic optimal solution. Hence it is possible to give a compact definition as:

Definition 1 (Basis) *The* basis *of a basic optimal solution, denoted as $\mathbb{I}_B$, is an index set such that $\mathbb{I}_B\in\{1,2,\ldots,n\}$ and* $\mathrm{Card}(\mathbb{I}_B)=m$*, separated by which A, $\boldsymbol{c}^*$ and $\boldsymbol{x}^*$ satisfy the condition in Proposition 1.* $\mathrm{Card}(\cdot)$ *denotes the cardinality of a set.*

After having the preliminaries concerned with linear programming and basic optimality, we can introduce the inexact linear programming.

2.2 Inexact Linear programming

Intuitively, an inexact linear programming problem is the one containing uncertainty. Namely, the coefficients in the problem have perturbations and cannot be determined precisely.

Due to the inverse calculation of the sub-matrix B derived by A in Proposition 1, we understand that if there exists uncertainty in the matrix A, the problem would become too complicated to solve. Hence, in our research, we assume that A maintains to be precise and contains no uncertainty, which derives the an inexact linear programming problem as:

Problem 2 *The linear programming problem containing inexactness is expressed in the following form:*

$$\min\ \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{x},\ \ s.t.\ A\boldsymbol{x}=\Psi,\ \boldsymbol{x}\geq\boldsymbol{0}, \tag{3}$$

where $\Psi\subseteq\mathbb{R}^m$ and $\Phi\subseteq\mathbb{R}^n$ denote the subsets that contain all possible $\boldsymbol{b}\in\Psi$ and $\boldsymbol{c}\in\Phi$, respectively.

Consequently, Problem 2 is no more a single problem but a combination of multiple problems. Hence, a solution $\boldsymbol{x}^*$ can be defined as there exist $\boldsymbol{b}\in\Psi$ and $\boldsymbol{c}\in\Phi$ such that $\boldsymbol{x}^*$ is optimal, which obviously forms a solution set as $\mathscr{S}(\Psi,\Phi):=\{\boldsymbol{x}^*\in R^n:\exists\boldsymbol{b}\in\Psi$ and $\boldsymbol{c}\in\Phi$ such that x^* is optimal$\}$.

By Proposition 1, we know that if the feasible set is inexact, a basic optimal solution changes its value even its basis maintains to be unchanged. However, since the solution value changes and an inexact linear programming problem is composed of multiple problems, a basic optimal solution may not be optimal, or even not be feasible in some sub-problems. Hence, it is necessary to have another *optimality* definition for Problem 2, which is given as *B*-stable [3].

Definition 2 (B-stable) *Let $\mathbb{I}_B \in \{1,\dots,n\}$ denote an index set. If for all $\boldsymbol{b} \in \Psi$ and $\boldsymbol{c} \in \Phi$, $\mathbb{I}_B$ is the* basis *of Problem 2, then the problem is a* B-stable *problem.*

Specifically, denoting the inexact coefficients Ψ and Φ as *intervals* for an inexact linear programming problem is extraordinarily welcomed and popular. An interval hyper-box can simply represent the inexactness, and because it is bounded and convex, one can check whether it is *B-stable* without too many difficulties.

The interval one is called *interval linear programming* [5], and we would utilise it to propose our approach.

3 Minimax Criterion Method

In reality, it is widely acknowledged that when the feasibility is violated, there would be a penalty. The penalty causes loss, which forms a trade-off with the optimal value of the problem. Therefore, a decision-maker proposes to find the equilibrium in the trade-off and make an ideal decision in a specific preferrence. Such a methodology is prevalent in a series of techniques, such as model predictive control and minimax regret criterion.

Hence, we prefer adding a penalty to the objective function to represent the loss from infeasibility. To evaluate the penalty from infeasibility, we utilised L^p-norm, which is common in the distance analysis in mathematics. Furthermore, as the penalties in each entry of $\boldsymbol{b}$ and $\boldsymbol{c}$ should be different, we attach weights on them.

For the strategy in solving a solution, we prefer the method of the minimax criterion. Moreover, for simplifications, we assume the problem satisfies the following two conditions:

(i) The objective function of the problem has *no inexactness*. Namely, Φ degenerates to a singleton equalling to $\{\boldsymbol{c}\}$.

(ii) The problem is *B-stable*.

Therefore, we can update the interval linear programming problem as:

Problem 3 *Let the following interval linear programming denote the one with inexact feasible set.*

$$\begin{aligned} \min \quad & \left[\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} + \max_{\boldsymbol{b}\in\Psi} p(\boldsymbol{b}^*,\boldsymbol{b})\right] \\ s.t. \quad & A\boldsymbol{x} = \boldsymbol{b}^*,\ \boldsymbol{x} \geq \boldsymbol{0},\ \boldsymbol{b}^* \in \Psi, \end{aligned} \tag{4}$$

where $p(\cdot)$ denotes the penalty function with two right-hand-side vectors and Ψ denotes the interval set that contains all possible $\boldsymbol{b}$.

Since we utilise weighted L^p-norm to represent $p(\cdot)$, it is defined as

$$p(\boldsymbol{b}^*,\boldsymbol{b}) := ||\boldsymbol{b}-\boldsymbol{b}^*||_l^\Delta = \sqrt[l]{\sum_{i=1}^{m} \delta_i |b_i - b_i^*|^l} \tag{5}$$

where $1 \leq l \in \mathbb{R}$ is the order of the L^p-norm and $\Delta = \operatorname{diag}(\delta_1, \delta_1, \dots, \delta_m)$ denotes the weighted diagonal matrix. Typically, when $l = 1$ the L^1-norm is called weighted absolute-value norm*, while L^2-norm is called* weighted Euclidean norm*.*

In application, we find that $l = 1,2$ is enough to represent most of conditions. Therefore, we would concentrate on these two conditions and give our algorithms.

3.1 Weighted Absolute-value Case

At first, let's consider the weighted absolute-value case. By Problem 3 with the weighted L^1-norm, we have the following problem.

Problem 4 *Consider an interval linear programming problem that only has interval right-hand-side vector. If the penalty function is the weighted absolute-value case, then the problem could be re-defined as*

$$\begin{aligned} \min \quad & \left[\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} + \max_{\boldsymbol{b}\in\Psi} \boldsymbol{p}^{\mathrm{T}}|\boldsymbol{b}^* - \boldsymbol{b}|\right] \\ & s.t.\ A\boldsymbol{x} = \boldsymbol{b}^*,\ \boldsymbol{x} \geq \boldsymbol{0},\ \boldsymbol{b}^* \in \Psi \end{aligned} \tag{6}$$

where $\boldsymbol{0} \leq \boldsymbol{p} \in \mathbb{R}^m$ is the weighted vector with $\exists i \in \{1,2,\dots,m\}$, $p_i > 0$.

To solve Problem 4, we have the following lemma by assuming that it is B-stable.

Lemma 1 *If Problem 4 is B-stable, then it is equivalent to*

$$\min_{\boldsymbol{b}^*\in\Psi} \left[\boldsymbol{c}_B^{\mathrm{T}} B^{-1} \boldsymbol{b}^* + \max_{\boldsymbol{b}\in\Psi} \boldsymbol{p}^{\mathrm{T}}|\boldsymbol{b}^* - \boldsymbol{b}|\right] \tag{7}$$

This result is obvious because in linear programming, if $\boldsymbol{x}^*$ is the optimal solution, then $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}^* = \boldsymbol{c}^{\mathrm{T}}B^{-1}\boldsymbol{b}$.

Then, we need to remove either the maximum operator or the absolute operator because both of them would increase the complexity of calculation. Fortunately, we are able to remove the maximum operator by the following theorem.

Theorem 1 *If $\boldsymbol{0} \leq \boldsymbol{p} \in \mathbb{R}^m$, then the following equation is valid.*

$$\max_{\boldsymbol{b}\in\Psi} \boldsymbol{p}^{\mathrm{T}}|\boldsymbol{b}^* - \boldsymbol{b}| = \boldsymbol{p}^{\mathrm{T}}|\boldsymbol{b}^* - \dot{\boldsymbol{b}}| + \boldsymbol{p}^{\mathrm{T}}\tilde{\boldsymbol{b}} \tag{8}$$

where $\dot{\boldsymbol{b}}$ and $\tilde{\boldsymbol{b}}$ denote the centre and the radius of Ψ, respectively. Mathematically, if $\Psi := [\underline{\boldsymbol{b}}, \overline{\boldsymbol{b}}]$, then

$$\dot{\boldsymbol{b}} := (\overline{\boldsymbol{b}} + \underline{\boldsymbol{b}})/2 \text{ and } \tilde{\boldsymbol{b}} := (\overline{\boldsymbol{b}} - \underline{\boldsymbol{b}})/2$$

Proof: As is known that for a constant $p_i > 0$, we have

$$\underset{\underline{b}_i \le b_i \le \overline{b}_i}{\operatorname{argmin}} \, p_i(b_i^* - b_i) = \begin{cases} \underline{b}_i, \text{ if } b_i^* > \dot{b}_i, \\ \overline{b}_i, \text{ if } b_i^* < \dot{b}_i, \\ \underline{b}_i \text{ or } \overline{b}_i, \text{ otherwise.} \end{cases}$$

It is equivalent to $p_i(|b_i^* - \dot{b}_i| + \tilde{b}_i)$, which derives the final equation. ■

As the calculation of the entry of $\boldsymbol{b}^*$ in Equation (7) is independent, we can convert the problem from vector calculation to element one. Moreover, by denoting $\boldsymbol{s} := B^{-\mathrm{T}}\boldsymbol{c}_B$, we have the equivalent problem in a new form:

$$\min_{\boldsymbol{b}^* \in \Psi} \left[\sum_{i=1}^{m} s_i b_i^* + p_i |b_i^* - \dot{b}_i| \right] + \boldsymbol{p}^{\mathrm{T}} \tilde{\boldsymbol{b}} \tag{9}$$

$$= \left[\sum_{i=1}^{m} \min_{\underline{b}_i \le b_i^* \le \overline{b}_i} s_i b_i^* + p_i |b_i^* - \dot{b}_i| \right] + \boldsymbol{p}^{\mathrm{T}} \tilde{\boldsymbol{b}} \tag{10}$$

If we separate the part $s_i b_i^* + p_i |b_i^* - \dot{b}_i|$ into two functions, which are $f_1(b_i^*) = s_i b_i^*$ and $f_2(b_i^*) = p_i |b_i^* - \dot{b}_i|$, we can regard it as the sum of two functions. Hence, as Figure 1 shows, we can have the following result that $\forall i \in \{1, 2, \ldots, m\}$,

$$b_i^* = \begin{cases} \dot{b}_i, & \text{if } |s_i| \le +p_i, \\ \underline{b}_i, & \text{if } s_i > +p_i, \\ \overline{b}_i, & \text{if } s_i < -p_i \end{cases} \tag{11}$$

and the optimal solution is $\boldsymbol{x}^* = B^{-1}\boldsymbol{b}^*$.

3.2 Euclidean Case

After having the result of the weighted absolute-value case, we can talk about the Euclidean case. For simplification, we use the square form to avoid the root calculation. Namely, we prefer the following function as penalty.

$$p(\boldsymbol{b}^*, \boldsymbol{b}) = (\boldsymbol{b} - \boldsymbol{b}^*)^{\mathrm{T}} \Delta (\boldsymbol{b} - \boldsymbol{b}^*) \tag{12}$$

where $\Delta = \operatorname{diag}(\delta_1, \delta_1, \ldots, \delta_m)$ is a positive diagonal matrix to represent the weight.

Similar to the absolute-value one, we still assume the interval linear programming problem is B-stable, which means we can utilise the result from Equation (7) directly. Hence, we can reconstruct the problem as

Problem 5 *Consider an ILP problem that only has interval right-hand-side vector. If the penalty function is the weighted Euclidean case and it is also B-stable, then the problem could be re-defined as*

$$\min_{\boldsymbol{b}^* \in \Psi} \left[\boldsymbol{c}_B^{\mathrm{T}} B^{-1} \boldsymbol{b}^* + \max_{\boldsymbol{b} \in \Psi} [(\boldsymbol{b} - \boldsymbol{b}^*)^{\mathrm{T}} \Delta (\boldsymbol{b} - \boldsymbol{b}^*)] \right] \tag{13}$$

Similar to Theorem 1, we can also have the following proposition.

Theorem 2 *If $\Delta = \operatorname{diag}(\delta_1, \delta_1, \ldots, \delta_m)$ is a positive diagonal matrix, the following equation is valid.*

$$\begin{aligned} &\max_{\boldsymbol{b} \in \Psi} [(\boldsymbol{b} - \boldsymbol{b}^*)^{\mathrm{T}} \Delta (\boldsymbol{b} - \boldsymbol{b}^*)] \\ &\quad = (\boldsymbol{b}^* - \dot{\boldsymbol{b}})^{\mathrm{T}} \Delta (\boldsymbol{b}^* - \dot{\boldsymbol{b}}) + \tilde{\boldsymbol{b}}^{\mathrm{T}} \Delta \tilde{\boldsymbol{b}} \end{aligned} \tag{14}$$

The proof is similar to the one of Theorem 1. Therefore, we can re-write the problem as

$$\min_{\boldsymbol{b}^* \in \Psi} \left[\sum_{i=1}^{m} s_i b_i^* + \delta_i (b_i^* - \dot{b}_i)^2 \right] + \tilde{\boldsymbol{b}}^{\mathrm{T}} \Delta \tilde{\boldsymbol{b}} \tag{15}$$

$$= \left[\sum_{i=1}^{m} \min_{\underline{b}_i \le b_i^* \overline{b}_i} s_i b_i^* + \delta_i (b_i^* - \dot{b}_i)^2 \right] + \tilde{\boldsymbol{b}}^{\mathrm{T}} \Delta \tilde{\boldsymbol{b}} \tag{16}$$

where $\dot{\boldsymbol{b}} := (\overline{\boldsymbol{b}} + \underline{\boldsymbol{b}})/2$ and $\tilde{\boldsymbol{b}} := (\overline{\boldsymbol{b}} - \underline{\boldsymbol{b}})/2$.

Much Simpler than the absolute-value one, the absolute operate is removed in Euclidean case. Therefore, $\forall i \in \{1, 2, \ldots, m\}$, $s_i b_i^* + \delta_i (b_i^* - \dot{b}_i)^2 = \delta_i (b_i^*)^2 + (s_i - 2\delta_i \dot{b}_i) b_i^* + \delta_i d_i^2$. $\because$ $\forall i \in \{1, 2, \ldots, m\}$, $\delta_i > 0$, $\therefore$ the minimum value should be accomplished at

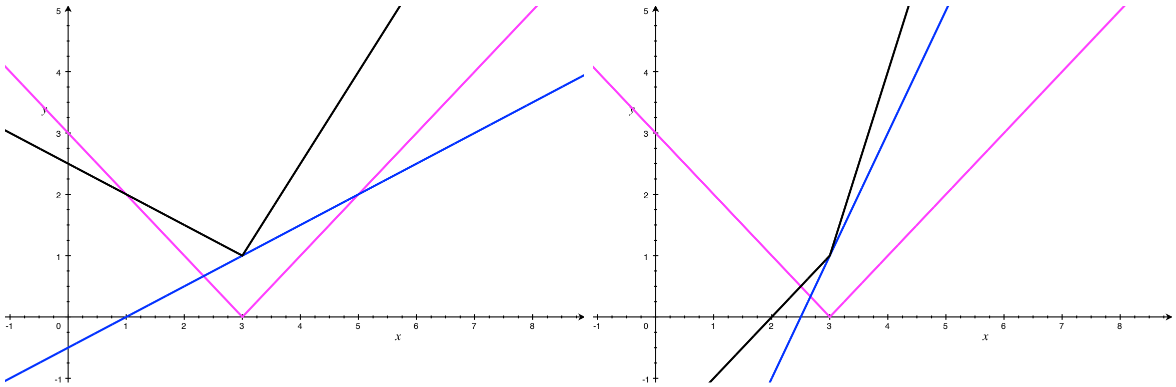
$$b_i^* = \frac{2\delta_i \dot{b}_i - s_i}{2\delta_i}$$

if b_i^* is unbounded. In the case that $\delta_i = 0$, we can simply remove the i_{th} part from objective function. Therefore, we have the final result as $\forall i \in \{1, 2, \ldots, m\}$ and $\delta_i > 0$,

$$b_i^* = \begin{cases} t_i, & \text{if } \underline{b}_i \le t_i \le \overline{b}_i, \\ \underline{b}_i, & \text{if } t_i < \underline{b}_i, \\ \overline{b}_i, & \text{if } t_i > \overline{b}_i \end{cases} \tag{17}$$

where $t_i = (2\delta_i \dot{b}_i - s_i)/(2\delta_i)$, and the optimal solution is $\boldsymbol{x}^* = B^{-1}\boldsymbol{b}^*$.

A huge difference between the Euclidean case with the absolute-value one is that, the final solution of $\boldsymbol{b}^*$ may vary at any point in the interval. However, in the absolute-value case, it usually lays at centre or the boundary.



(a) If $s_i < p_i$, the minimum point of $f_1 + f_2$ is the one that f_2 gets minimum.

(b) If $s_i > p_i$, the minimum point of $f_1 + f_2$ will go the boundary because it becomes monotonic.

Figure 1: the figure to show the different conditions with different s_i and p_i, where f_1 is the blue line, f_2 is the magenta one and $f_1 + f_2$ is black one.

4 Comparison with Fuzzy Linear Programming

Since fuzzy linear programming is also a conventional tool to treat the inexactness in a linear programming problem, we prefer making a comparison with it.

Let us re-consider Problem 2 in the fuzzy way. Similar to the premise in Problem 3, we assume that the fuzzy linear programming problem has a fixed objective function. For the inexactness in the right-hand-side vector $\boldsymbol{b}$, we assume each entry of it is a symmetric triangular fuzzy number. By the fuzzy subset definition [6], the left and right membership functions of a symmetric triangular fuzzy number should be linear and equal, which can be expressed as the following figure as an example:

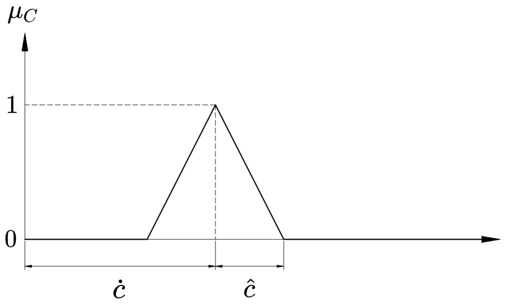


Figure 2: A symmetric triangular fuzzy number c

Moreover, if we consider expressing a symmetric triangular fuzzy number by its h-level set [6], we have the result as:

$$\forall h \in (0,1],\ [c]_h = [\dot{c} - (1-h)\hat{c}, \dot{c} + (1-h)\hat{c}],$$

where $\dot{c}$ and $\hat{c}$, as Figure 2 shows, denote the centre and radius of $[c]_h$ for $h \to 0^+$, respectively. For simplification, we denote a symmetric triangular fuzzy number as $c := (\dot{c}, \hat{c})_{\text{STF}}$.

Hence, we have the following fuzzy problem:

Problem 6 *A fuzzy linear programming problem can be expressed in the following form:*

$$\min\ \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x},\ \ s.t.\ A\boldsymbol{x} = \tilde{\boldsymbol{b}},\ \boldsymbol{x} \geq \boldsymbol{0}, \tag{18}$$

where $\tilde{\boldsymbol{b}}$ denotes the right-hand-side vector of the constraints, in which each entry is a symmetric triangular fuzzy number.

Unlike the minimax criterion that take the regret into the objective, fuzzy linear programming generally only focuses on the *reliability* of a decision, which in the view of information system is related to the *correctness* of an event or the *quality* of an information.

For example, for Problem 6, fuzzy linear programming prefers looking for a solution to satisfy the constraints with the h-level for $\tilde{\boldsymbol{b}}$ as high as possible. Alternatively, we can also specify a particular h-level for the constraints and solve an acceptable solution.

However, by reviewing the past works, we can still find some materials that consider penalty. For example, Zadeh L.A. [15] considered a multiplication of the h-level with the original objective function $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$, which hence gives the following problem as:

$$\min f(\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}, h),\ s.t.\ A\boldsymbol{x} \subseteq [\boldsymbol{b}]_h,\ \boldsymbol{x} \geq \boldsymbol{0}, \tag{19}$$

where $f(\cdot)$ is a function being *negative* correlated to h and *positive* correlated to $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$, where both variables usually have a multiplication relation, such as $(1-h+\varepsilon)\cdot(\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x})$, $\varepsilon > 0$ when $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} > 0$.

Alternatively, it is also possible to consider the penalty as an additive part instead of the multiplicative one, like Problem 3. However, such conversion usually makes the linear programming problem no more linear, which causes extraordinary puzzles for calculation.

By comparing the interval case with the fuzzy one, it is not hard to have the following conclusion.

Interval linear programming takes all possible situations into consideration, which usually focuses on satisfying the trade-off in the worst case. However,

Fuzzy linear programming concentrates on the reliability of a solution, where the higher reliability a solution has, the better it is to be accepted as a candidate.

5 Numerical Example

To illustrate the difference of two approaches, we give examples here. As for the numerical examples, let's modify the one from Wendell [13].

Example 1 *Let us consider a production planning problem. Assume a factory produce 4 products a, b, c and d, and each profit of them are 12, 20, 18 and 40 units, respectively. Any product consumes both row materials A and B, where a, b, c and d consume A as 4, 9, 7 and 10 and consume B as 1, 1, 4, 40, respectively. However, for the supply of A and B, we only know that A is* approximately *6000 and B is* approximately *4000. It is noted that if the raw materials is not enough, the factory needs to buy them temporarily, which causes loss to the profit. So how should a decision-maker make a production plan to have a maximum profit?*

5.1 Proposed Approach

By converting the production planning problem into a linear programming problem, which assumes the approximate values of A and B to be the exact ones, we have the one below:

$$\begin{aligned} \min \quad & -12x_1 - 20x_2 - 18x_3 - 40x_4 \\ s.t. \quad & 4x_1 + 9x_2 + 7x_3 + 10x_4 + x_5 = 6000 \\ & x_1 + x_2 + 3x_3 + 40x_4 + x_6 = 4000 \\ & x_1, x_2, x_3, x_4, x_5, x_6 \geq 0 \end{aligned}$$

which is equivalent to the following tabular:

	x_1	x_2	x_3	x_4	x_5	x_6	RHS
x_5	4	9	7	10	1		6000
x_6	1	1	3	40		1	4000
$(-z)$	-12	-20	-18	-40	0	0	0

Solved by the simplex method, we have the following tabular.

	x_1	x_2	x_3	x_4	x_5	x_6	RHS
x_1	1	$\frac{7}{3}$	$\frac{5}{3}$		$\frac{4}{15}$	$-\frac{1}{15}$	$1333\frac{1}{3}$
x_4		$-\frac{1}{30}$	$\frac{1}{30}$	1	$-\frac{1}{150}$	$\frac{2}{75}$	$66\frac{2}{3}$
$(-z)$	0	$\frac{20}{3}$	$\frac{10}{3}$	0	$\frac{44}{15}$	$\frac{4}{15}$	$-18666\frac{2}{3}$

By the tolerance approach [13], it is not hard to construct a hyper-box for the right-hand-side values to make the problem B-stable. Hence, we solve the problem by assuming multiplicative perturbations with $[6000(1 \pm \lambda), 4000(1 \pm \lambda)]^{\mathrm{T}}$, and the result turns out to be $\lambda \leq 5/11$.

To avoid the difficulty caused by fraction and degeneracy, we choose $\lambda_1 = 0.45$ to test both proposed approaches, which means Ψ is

$$\dot{\boldsymbol{b}} = [6000, 4000]^{\mathrm{T}} \text{ and } \tilde{\boldsymbol{b}} = [2700, 1800]^{\mathrm{T}}$$

Firstly, we define the infeasibility penalty weight on the right-hand-side vector as $[5,1]^{\mathrm{T}}$, which means $\boldsymbol{p} = [5,1]^{\mathrm{T}}$ in the absolute-value case and $\Delta = \mathrm{diag}(5,1)$ in the Euclidean case. Then, we calculate the $\boldsymbol{s} := B^{-\mathrm{T}}\boldsymbol{c}_B$, which is $\boldsymbol{s} = [-44/5, -4/15]^{\mathrm{T}}$.

For the absolute-value case, we have the result by Equation (11).

$$\begin{aligned} & s_1(-44/5) < -p_1(-5) \\ & \qquad \Rightarrow b_1^* = \dot{b}_1 + \tilde{b}_1 = 8700 \\ & -p_2(-1) < s_2(-4/15) < p_2(1) \\ & \qquad \Rightarrow b_2^* = \dot{b}_2 = 4000 \end{aligned}$$

So $\boldsymbol{b}^* = [8700, 4000]^{\mathrm{T}}$, which means the optimal solution that we choose is $\boldsymbol{x} = [2053.333, 0, 0, 48.667, 0, 0]^{\mathrm{T}}$.

In the Euclidean case, it is necessary to solve $\boldsymbol{t}$ in Equation (17), which are $t_1 = 6000.88$, $t_2 = 4000.13$. Therefore, $b_1^* = 6000.88$, $b_2^* = 4000.13$, which derive the optimal solution to be $\boldsymbol{x} = [1333.559, 0, 0, 66.664, 0, 0]^{\mathrm{T}}$.

5.2 Fuzzy Approach

However, if we consider the example in the fuzzy way, we need to concentrate on the h-level of the right-hand-side $\tilde{\boldsymbol{b}}$. To have a better comparison, we denote $\tilde{\boldsymbol{b}}$ as a symmetric triangular fuzzy vector similar to Ψ in the previous approach, which means $\tilde{\boldsymbol{b}} = (\dot{\boldsymbol{b}}, \hat{\boldsymbol{b}})_{\mathrm{STF}}$ such that $\dot{\boldsymbol{b}} = [6000, 4000]^{\mathrm{T}}$ and $\tilde{\boldsymbol{b}} = [2700, 1800]^{\mathrm{T}}$. Therefore, by utilising the h-level set of $\tilde{\boldsymbol{b}}$, we can have the following converted problem as:

$$
\begin{aligned}
\min\ & -12x_1-20x_2-18x_3-40x_4\\
s.t.\ & 4x_1+9x_2+7x_3+10x_4+x_5\geq 6000-2700(1-h)\\
& 4x_1+9x_2+7x_3+10x_4+x_5\leq 6000+2700(1-h)\\
& x_1+x_2+3x_3+40x_4+x_6\geq 4000-1800(1-h)\\
& x_1+x_2+3x_3+40x_4+x_6\leq 4000+1800(1-h)\\
& x_1,x_2,x_3,x_4,x_5,x_6\geq 0
\end{aligned}
$$

where we specify $h\in(0,1]$ before solving the problem. Hence we have the solutions and the optimal values with different h as the following tabular shows:

h	0.6	0.7	0.8	0.9	1.0
$x_1(\times 10^3)$	1.57	1.51	1.45	1.39	1.33
x_2	0	0	0	0	0
x_3	0	0	0	0	0
$x_4(\times 10^1)$	7.87	7.57	7.27	6.97	6.67
res$(\times 10^4)$	-2.20	-2.12	-2.03	-1.95	-1.87

Table 1: The optimal solutions and values with different h-levels for Example 1

From Table 1, it is obvious that if we decrease the reliability of the solution, we would potentially have gain more, which shows a typical trade-off mentioned in the previous content.

6 Conclusion

In this paper, we propose an approach based on the minimax criterion to manage the flexibility in a problem's feasible set. Regarding flexibility as the interval, we convert the problem into an interval linear programming one and utilise the L^P-norm to represent the loss from infeasibility. To have the most robust solution, we aim to minimise the combination of the objective function and the penalty from infeasibility, while maximising the penalty as much as possible simultaneously.

As for the evaluation of the penalty, we consider two models. One is the weighted absolute-value case, which means the penalty is linear to the level of infeasibility, and the other one is the weighted Euclidean case, which means the suffering from the penalty is quadratic to the infeasibility.

For both models, we propose the corresponding algorithms and get excellent results. Treating some parts of the problem as the shadow price shows that if the penalty is large enough, the optimised solution prefers the interval's centre in both conditions. However, if the penalty is little, it may move to the interval's boundary.

Mathematically, if one would rather avoid larger infeasibility, the weighted Euclidean case may be a better choice. When the extent of the infeasibility becomes more extensive, the penalty from the quadratic function will increase more than the linear one. In the application, nevertheless, the penalty is more likely linear to the infeasibility.

Furthermore, to illustrate the difference of our proposed approach with the fuzzy one, we briefly review the knowledge of fuzzy theory and then state the conventional method in fuzzy linear programming. Finally, we also compare them for a given production planning example.

However, the drawback of our approach is also transparent. In our models, we assume the problem to be B-stable with a fixed objective function. However, it is usually hard to have such an ideal situation in applications. Therefore, our study in the future would concentrate on two situations. One is that the inexact linear programming problem is not B-stable. The other is that there exists inexactness in the objective function.

References

[1] S. P. Bradley, A. C. Hax, T. L. Magnanti, Applied mathematical programming, Addison-Wesley Publishing Company, 1977.

[2] C. Filippi, A fresh view on the tolerance approach to sensitivity analysis in linear programming, European journal of operational research 167 (1) (2005) 1–19.

[3] E. Garajová, M. Hladík, On the optimal solution set in interval linear programming, Computational Optimization and Applications 72 (1) (2019) 269–292.

[4] M. Hladík, Additive and multiplicative tolerance in multiobjective linear programming, Operations Research Letters 36 (3) (2008) 393–396.

[5] M. Hladík, Computing the tolerances in multiobjective linear programming, Optimization Methods & Software 23 (5) (2008) 731–739.

[6] M. Inuiguchi, Necessity measure optimization in linear programming problems with fuzzy polytopes, Fuzzy Sets and Systems 158 (17) (2007) 1882–1891.

[7] M. Inuiguchi, M. Sakawa, Possible and necessary optimality tests in possibilistic linear programming problems, Fuzzy Sets and Systems 67 (1) (1994) 29–46.

[8] M. Inuiguchi, M. Sakawa, Minimax regret solution to linear programming problems with an

interval objective function, European Journal of Operational Research 86 (3) (1995) 526–536.

[9] P. Kall, J. Mayer, et al., Stochastic linear programming, Vol. 7, Springer, 1976.

[10] I. Kondor, S. Pafka, G. Nagy, Noise sensitivity of portfolio selection under various risk measures, Journal of Banking & Finance 31 (5) (2007) 1545–1573.

[11] R. Shamir, Probabilistic analysis in linear programming, Statistical Science (1993) 57–64.

[12] M. J. Todd, Probabilistic models for linear programming, Mathematics of Operations Research 16 (4) (1991) 671–693.

[13] R. E. Wendell, The tolerance approach to sensitivity analysis in linear programming, Management Science 31 (5) (1985) 564–578.

[14] F. R. Wondolowski Jr, A generalization of wendell's tolerance approach to sensitivity analysis in linear programming, Decision Sciences 22 (4) (1991) 792–811.

[15] L. Zadeh, Fuzzy sets and systems, International Journal of General Systems 17 (2-3) (1990) 129–138.