A Multi-objective Method for Solving Fuzzy Linear Programming Based on Semi-infinite Model

S. H. Nasseri and H. Zavieh

Department of Mathematical Sciences, University of Mazandaran, Babolsar, Iran

ABSTRACT

In this paper, we present a new method to solve a fuzzy linear programming problem with fuzzy coefficients in the constraints and the objective function based on solving an associated multi-objective model. In particular, we present a weighted method for linear semi-infinite programming (LSIP) model to solve the original problem. Finally, a numerical example is included to illustrate the suggested solving process.

1. Introduction

After successful applications of fuzzy sets theory in the various fields, this theory has been applied in the optimisation area. In particular, fuzzy linear programming (FLP) has a long history as well as fuzzy sets theory. As a pioneering study, Delgado et al. [1] studied a general model for solving FLP problems which involve fuzziness both in the coefficients and in the accomplishment of the constraints. Wu et al. [2] used an analytic centre based on a cutting plane method to solve linear semi-infinite programming (LSIP) problems. It is shown that a near optimal solution can be obtained by generating a polynomial number of cuts. Goberna et al. [3] analyse the effect on the optimal value of a given LSIP problem of the kind of perturbations which more frequently arise in practical applications. Goberna and López [4] present a state-of-the-art survey on LSIP theory and its extensions (in particular, convex semi-infinite programming). Recently, Nasseri et al. [5] shown that such problems can be reduced to an LSIP problem with fuzzy cost coefficients.

This paper is organised into five sections. In Section 2, we give some necessary concepts and definitions which is useful throughout the paper. In Section 3, the mentioned novel solving approach is given to solve the FLP model. Section 4 is assigned to the illustrative example and finally, we will give the conclusions in Section 5.

2. Preliminaries

Here, we first give some fundamental concepts of fuzzy sets which is necessary to the other sections and which is taken from [6–8].

CONTACT S. H. Nasseri nhadi57@gmail.com
© 2018 The Author(s). Published by Taylor & Francis Group on behalf of the Fuzzy Information and Engineering Branch of the Operations Research Society of China & Operations Research Society of Guangdong Province. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.
Definition 2.1: Let $\mathbb{R}$ be the real line. A fuzzy set $\tilde{A}$ in $\mathbb{R}$ is defined to be a set of ordered pairs $\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) : x \in \mathbb{R}\}$, where $\mu_{\tilde{A}}(x)$ is called the membership function for the fuzzy set. The membership function maps each element of $\mathbb{R}$ to a membership value between 0 and 1. A general form of a membership function $\mu_{\tilde{A}}(x)$ is given by

$$
\mu_{\tilde{A}}(x) = \begin{cases} 
L(x), & m - l \leq x \leq m, \\
R(x), & m \leq x \leq m + u, \\
0, & \text{o.w,}
\end{cases}
$$

where $L(x)$ is a function from $(m - l, m)$ to $[0, 1]$ that is monotonically increasing, continuous from the right and such that $L(x) = 0$, for $x \in (-\infty, m - l]$; $R(x)$ is a function from $[m, m + u]$ to $[0, 1]$ that is monotonically decreasing, continuous from the left and such that $R(x) = 0$ for $x \in [m + u, +\infty)$.

We symbolically show every fuzzy number $\tilde{A}$ by $\tilde{A} = (m - l, m, m + u)_{LR}$ based on its membership function.

We named every fuzzy set as a triangular fuzzy number, where $L(x)$ an $R(x)$ are linear functions. We also show the set of all triangular fuzzy number by $F(\mathbb{R})$.

Definition 2.2: Assume $\tilde{A}$ is a fuzzy number and $\alpha \in (0, 1]$, then an $\alpha$-cut of $\tilde{A}$ is defined as $\{x | x \in \mathbb{R}, \mu_{\tilde{A}}(x) \geq \alpha\}$, and we briefly denote it as $A_{\alpha}$.

Definition 2.3: A fuzzy number $\tilde{A}$ shall be called a (fuzzy triangular) zero number, symbolised by $\tilde{0}$, if it is written in the form $\tilde{A} = \tilde{0} = (0, 0, 0)$.

Definition 2.4: If $\tilde{A}_1, \tilde{A}_2 \in F(\tilde{A})$ and $\alpha \in (0, 1]$, then $\tilde{A}_1 \geq_{\alpha} \tilde{A}_2$ if and only if

$$
L_{\tilde{A}_1}(t) \geq_{\alpha} L_{\tilde{A}_2}(t), \\
R_{\tilde{A}_1}(t) \geq_{\alpha} R_{\tilde{A}_2}(t), \quad \forall t \in [\alpha, 1].
$$

An FLP problem is defined as follows [7]:

$$(\text{FLP}) \quad \text{Min} \quad \sum_{j=1}^{n} \tilde{c}_j x_j$$

s.t. \quad \sum_{j=1}^{n} \tilde{a}_{ij} x_j \geq_{\alpha} \tilde{b}_i, \quad i = 1, \ldots, m,

\quad x_j \geq 0, \quad j = 1, \ldots, n,

where $\tilde{c}_j, \tilde{a}_{ij}, \tilde{b}_i \in F(\mathbb{R}), i = 1, \ldots, m, j = 1, \ldots, n$.

3. A Proposed Method

In this section, we will present a new approach for solving an FLP problem which is defined in (1). In the process of solving an FLP problem, we first need to solve an LSIP problem.
with multi-objectives. In this way, we show that this problem will reduce to one target by a weighted method for objective function. Finally, we will use a cutting plan method for its constraints.

The constraint \( \sum_{j=1}^{n} \tilde{a}_{ij} x_j \geq \tilde{b}_i, \ i = 1, \ldots, m \) based on Definition 2.4 is equivalent to the following constraints:

\[
\sum_{j=1}^{n} L_{\tilde{a}_{ij}}(t)x_j \geq L_{\tilde{b}_i}(t), \quad \forall t \in [\alpha, 1].
\]

\[
\sum_{j=1}^{n} R_{\tilde{a}_{ij}}(t)x_j \geq R_{\tilde{b}_i}(t), \quad \forall t \in [\alpha, 1].
\]

We define

\[
\begin{align*}
& f_{ij}(t) \triangleq L_{\tilde{a}_{ij}}(t), \ i = 1, \ldots, q, \ f_{ij}(t) \triangleq R_{\tilde{a}_{i-j}}(t) \ i = q + 1, \ldots, 2q \text{ and } j = 1, \ldots, n, \\
& b_i(t) \triangleq L_{\tilde{b}_i(t)}, \ i = 1, \ldots, q, \ b_i(t) \triangleq R_{\tilde{b}_i(t)} \ i = q + 1, \ldots, 2q \text{ and } j = 1, \ldots, n, \\
& m \triangleq 2q, \ t \in [\alpha, 1].
\end{align*}
\]

The objective function of Problem (1) can be reduced as follows:

\[
\begin{align*}
\text{Max } Z_1 &= \sum_{j=1}^{n} \left( c_j^M - c_j^L \right) x_j, \\
\text{Min } Z_2 &= \sum_{j=1}^{n} c_j^M x_j, \\
\text{Min } Z_3 &= \sum_{j=1}^{n} (c_j^U - c_j^M) x_j.
\end{align*}
\]

Then, we define

\[
\text{Min } Z = W_2 Z_2 + W_3 Z_3 - W_1 Z_1,
\]

where \( 0 \leq W_1, W_2, W_3 \leq 1 \) such that \( W_1 + W_2 + W_3 = 1 \) are the weights of the mentioned functions, which are determined by the decision maker. Hence, Problem (1) can be reduced to Problem (2),

\[
\begin{align*}
\text{Min } Z &= W_2 \sum_{j=1}^{n} c_j^M x_j + W_3 \sum_{j=1}^{n} (c_j^U - c_j^M) x_j - W_1 \sum_{j=1}^{n} (c_j^M - c_j^L) x_j \\
\text{s.t. } &\begin{pmatrix} f_{11}(t) & \cdots & f_{1n}(t) \\ \vdots & \ddots & \vdots \\ f_{m1}(t) & \cdots & f_{mn}(t) \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \geq \begin{pmatrix} b_1(t) \\ \vdots \\ b_m(t) \end{pmatrix}, \quad \forall t \in T, \\
x_j &\geq 0, j = 1, \ldots, n,
\end{align*}
\]

where \( T \) is a compact metric space, \( f_{ij}(t), b_i(t), i = 1, \ldots, m, j = 1, \ldots, n \) are real-valued continuous functions on \( T \).
Problem (2) is a linear semi-infinite programming with weighted function (WFLSIP) problem with \( n \) variables and infinitely many constraints. In the sequel, we denote its feasible region and its optimal objective value as FP and \( v(\text{LSIP}) \), respectively. Here, for solving the following LSIP problem, we reduce it to the form of Problem (2). We will use a ‘cutting plane approach’ to solve LSIP problems as well as used in [9]. By using a cutting plan approach, we can design an iterative algorithm that adds \( m \) constraints at a time until an optimal solution is identified. To be more specific, at the \( r \)th iteration, given \( T_r = \{ t_1, t_2, \ldots, t_r \} \), where \( t_r = (t_1', \ldots, t_m') \in T^m \), and \( r \geq 1 \). We consider the following model:

\[
\begin{align*}
\text{Min} & \quad Z = W_2 \sum_{j=1}^{n} c_j^M x_j + W_3 \sum_{j=1}^{n} (c_j^u - c_j^M) x_j - W_1 \sum_{j=1}^{n} (c_j^M - c_j^l) x_j \\
\text{s.t.} & \quad \begin{pmatrix} f_1(t_1) & \cdots & f_n(t_1) \\ \vdots & \ddots & \vdots \\ f_m(t_m) & \cdots & f_m(t_m) \\ \vdots & \ddots & \vdots \\ f_1(t_1') & \cdots & f_n(t_1') \\ \vdots & \ddots & \vdots \\ f_m(t_m') & \cdots & f_m(t_m') \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \geq \begin{pmatrix} b_1(t_1) \\ \vdots \\ b_m(t_m) \end{pmatrix}, \\
& \quad x_j \geq 0, \quad j = 1, \ldots, n.
\end{align*}
\]

Let \( F' \) is the feasible region of Problem (3). If \( x' = (x_1', x_2', \ldots, x_n') \) is an optimal solution for Problem (3), then consider the ‘constraint violation functions’ as follows:

\[
v_j^{i+1}(t) \triangleq \sum_{j=1}^{n} f_j(t)x_j' - b_i(t), \quad \forall t \in T, \ i = 1, \ldots, m.
\]

Since \( f_j(t) \) and \( b_i(t) \) are continuous over \( T \), and also \( T \) is a compact set, then the function \( v_j^{i+1}(t) \) achieves its minimum over \( T \), for \( i = 1, \ldots, m \). Let \( t_j^{i+1}(t) \) be such a minimiser and consider the value of \( v_j^{i+1}(t_j^{i+1}) \), for \( i = 1, \ldots, m \). If the value is greater or equal than to zero, for \( i = 1, \ldots, m \), then \( x' = (x_1', x_2', \ldots, x_n') \) becomes a feasible solution of Problem (3) and hence, \( x' = (x_1', x_2', \ldots, x_n') \) is optimal for Problem (2), because the feasible region \( F' \) of Problem (3) is not smaller than the feasible region of Problem (2).

Below, we present the main steps of the suggested approach.

**Algorithm 3.1:** For conclude of optimise \( Z \) do:

**Step 1:** Set \( r = 1 \); Choose any \( t_1' \in T \), Set \( T_1 = \{ t_1' \} \).

**Step 2:** Solve \((\text{LP})'\) and obtain an optimal solution \( x' \).

**Step 3:** Find a minimiser \( t_j^{i+1} \in v_j^{i+1}(t) \) over \( T \), for \( i = 1, \ldots, m \).

**Step 4:** If \( v_j^{i+1}(t_j^{i+1}) \geq 0 \), for \( i = 1, \ldots, m \), then stop with \( x' \) being an optimal solution of LSIP.

Otherwise, set \( T_{r+1} = T_r \cup \{ t_r^{i+1} \} \) and \( r \leftarrow r + 1 \), go to Step 1.
4. Numerical Example

In this section, the solving process of an FLP problem, which is proposed in the last section, will be illustrated.

We consider a company which produces two products $A_1$ and $A_2$. These products are processed on two different machines $A_1$ and $A_2$. The time required for the production of one unit of each product on both machines is uncertain which is given together with the daily capacities of the machines in Table 1.

Note that the time availability can vary from day to day due to the breakdown of the machines, overtime work, etc. Finally, the profit for each product can also vary due to variations in price. At the same time, the company wants to keep the profit somehow close to 8 for $P_1$ and 4 for $P_2$. The company wants to determine the range of each product to be produced per day to maximise its profit. It is assumed that all the amounts produced are consumed in the market.

Since the profit from each product and the availability time of each machine are uncertain, the number of units to be produced on each product will also be uncertain. So, we may formulate this problem as an FLP problem. Note that, we will use triangular fuzzy numbers for each uncertain value.

The profit of $P_1$ which is close to 8 is modelled as $(6, 8, 9)$ and that of $P_2$ which is close to 4 is modelled as $(3, 4, 7)$. Thus, we formulate the given FLP problem as

$$\text{Max } Z = (6, 8, 9)x_1 + (3, 4, 7)x_2$$

s.t. $(0.6, 1.5, 0.6)x_1 + (1.6, 2, 2.6)x_2 \leq_{\alpha} (3, 5, 9),$ 

$(1.2, 2, 2.3)x_1 + (0.6, 1, 1.3)x_2 \leq_{\alpha} (4, 5, 8),$ 

$x_1, x_2 \geq 0.$

Now, this FLP will be reduced to the following from:

$$\begin{align*}
\text{Min } & Z_1 = 2x_1 + x_2 \\
\text{Max } & Z_2 = 8x_1 + 4x_2 \\
\text{Max } & Z_3 = x_1 + 3x_2 \\
\text{s.t. } & \begin{pmatrix}
0.9t_1 + 0.6 & 0.4t_1 + 1.6 \\
0.8t_2 + 1.2 & 0.5t_2 + 0.5 \\
-0.1t_3 + 1.6 & -0.6t_3 + 2.6 \\
-2t_4 + 4 & -0.3t_4 + 1.3
\end{pmatrix} \begin{pmatrix}
x_1 \\
x_2
\end{pmatrix} \leq \begin{pmatrix}
2t_1 + 3 \\
t_2 + 4 \\
-4t_3 + 9 \\
-3t_4 + 8
\end{pmatrix}, \quad \forall t_i \in [\alpha, 1], \\
x_1, x_2 \geq 0.
\end{align*}$$

Table 1. Capacities of the machines.

| Machines | Time per unit (min) | Machine capacity (h/day) |
|----------|---------------------|--------------------------|
| $M_1$    | $(0.6, 1.5, 0.6)$   | $(1.6, 2, 2.6)$          | $(3, 5, 9)$          |
| $M_2$    | $(1.3, 2, 4)$       | $(0.6, 1, 1.3)$          | $(4, 5, 8)$          |
An expert decision maker has been decided to give $W_1 = \frac{1}{4}$ weight for first objective function, $W_2 = \frac{1}{2}$ weight for second objective function and $W_3 = \frac{1}{4}$ weight for third objective function, where $\sum_{i=1}^{3} W_i = 1$. We assume in this example that $\alpha = 0.6$ and $t^1 = (t^1_1, t^1_2, t^1_3, t^1_4) = (0.7, 0.8, 0.7, 0.8)$ as an arbitrary point, then we have the regular linear programme,

$$\begin{align*}
\text{Max} & \quad Z = -\frac{1}{4} (2x_1 + x_2) + \frac{1}{2} (8x_1 + 4x_2) + \frac{1}{4} (x_1 + 3x_2) = 3.75x_1 + 2.5x_2 \\
\text{s.t.} & \quad \begin{pmatrix} 0.9t^1_1 + 0.6 & 0.4t^1_1 + 1.6 \\ 0.7t^1_2 + 1.3 & 0.4t^1_2 + 0.6 \\ -0.1t^1_3 + 1.6 & -0.6t^1_3 + 2.6 \\ -2t^1_4 + 4 & -0.3t^1_4 + 1.3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \leq \begin{pmatrix} 2t^1_1 + 3 \\ t^1_2 + 4 \\ -4t^1_3 + 9 \\ -3t^1_4 + 8 \end{pmatrix}, \quad \forall t_i \in [\alpha, 1],
\end{align*}$$

$x_1, x_2 \geq 0$.

By solving the current linear programming problem, we get

$$Z^* = 9.716 \quad \text{and} \quad x^1 = (x^1_1, x^1_2) = (1.828, 1.145).$$

Define,

$$\begin{align*}
v^2_1(t_1) &= (-0.9t_1 - 0.6)x^1_1 + (-0.4t_1 - 1.6)x^1_2 - (-2t_1 - 3), \\
v^2_2(t_2) &= (-0.7t_2 - 1.3)x^1_1 + (-0.4t_2 - 0.6)x^1_2 - (-t_2 - 4), \\
v^2_3(t_3) &= (0.1t_3 - 1.6)x^1_1 + (0.6t_3 - 2.6)x^1_2 - (4t_3 - 9), \\
v^2_4(t_4) &= (2t_4 - 4)x^1_1 + (0.3t_4 - 1.3)x^1_2 - (3t_4 - 8).
\end{align*}$$

So, we have

$$v^2_1(t_1) = -0.103t_1 + 0.071, \quad v^2_2(t_2) = -0.738t_2 + 0.937, \quad v^2_3(t_3) = -3.13t_3 + 3.07$$

and

$$v^2_4(t_4) = t_4 - 0.8.$$

The minimisers of $v^2_1(t_1), v^2_2(t_2), v^2_3(t_3), v^2_4(t_4)$ over $[\alpha, 1]$ are $(1, 1, 1, 0.6)$, respectively.
Hence, we choose \( t^2 = (t_1^2, t_2^2, t_3^2, t_4^2) = (1, 1, 1, 0.6) \). Since \( v_1^3(t_1), v_2^3(t_2) \geq 0 \) and \( v_3^3(t_3), v_4^3(t_4) \leq 0 \), the algorithm iterates with a new following linear mathematical programme:

\[
\begin{align*}
\text{Max} & \quad Z_2 = 3.75x_1 + 2.5x_2 \\
\text{s.t.} & \quad \left( \begin{array}{cccc}
0.9t_1^2 + 0.6 & 0.4t_1^2 + 1.6 \\
0.7t_2^2 + 1.3 & 0.4t_2^2 + 0.6 \\
-0.1t_3^2 + 1.6 & -0.6t_3^2 + 2.6 \\
-2t_4^2 + 4 & -0.3t_4^2 + 1.3 \\
\vdots & \vdots & \vdots & \vdots \\
0.9t_1^2 + 0.6 & 0.4t_1^2 + 1.6 \\
0.7t_2^2 + 1.3 & 0.4t_2^2 + 0.6 \\
-0.1t_3^2 + 1.6 & -0.6t_3^2 + 2.6 \\
-2t_4^2 + 4 & -0.3t_4^2 + 1.3 \\
\end{array} \right) \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix}
2t_1^3 + 3 \\
t_2^3 + 4 \\
-4t_3^3 + 9 \\
-3t_4^3 + 8 \\
\end{bmatrix}
\end{align*}
\]

\[\begin{align*}
x_1, x_2 & \geq 0.
\end{align*}\]

Solving \((LP^2)\) results in an optimal solution

\[Z^* = 20 \text{ and } x^2 = (x_1^2, x_2^2) = (1.838, 1.121)\].

Define,

\[
\begin{align*}
v_1^3(t_1) &= (-0.9t_1 - 0.6)x_1^2 + (-0.4t_1 - 1.6)x_2^2 - (-2t_1 - 3) = -0.102t_1 + 0.103, \\
v_2^3(t_2) &= (-0.7t_2 - 1.3)x_1^2 + (-0.4t_2 - 0.6)x_2^2 - (-t_2 - 4) = -0.728t_2 + 0.938, \\
v_3^3(t_3) &= (0.1t_3 - 1.6)x_1^2 + (0.6t_3 - 2.6)x_2^2 - (4t_3 - 9) = -3.143t_3 + 3.145, \\
v_4^3(t_4) &= (2t_4 - 4)x_1^2 + (0.3t_4 - 1.3)x_2^2 - (3t_4 - 8) = 1.32t_4 - 0.809.
\end{align*}\]

The minimisers of \( v_1^3(t_1), v_2^3(t_2), v_3^3(t_3), v_4^3(t_4) \) over \([\alpha, 1]\) are \((1, 1, 1, 0.6)\), respectively.

Hence, we choose \( t = (t_1^3, t_2^3, t_3^3, t_4^3) = (1, 1, 1, 0.6). \) Since \( v_1^2(t_1), v_2^2(t_2), v_3^2(t_3), v_4^2(t_4) \geq 0 \), the algorithm stops and \( x^* = x^2 = (1.838, 1.121) \) results to be optimal solution with \( \alpha = 0.6 \).

5. Conclusion

In this paper, we investigated an FLP problem. We proposed a new model denoted as ‘Weighted Method’ for the concluded LSIP problem. By using the \( \alpha \)-preference, we used the cutting plane algorithm to solve the reduced semi-infinite linear programming problem. Finally, we illustrated the proposed solving approach by a numerical example. We emphasise that this approach can be useful for an extended version of this model too. We also suggest to an interested reader to investigate the sensitivity analysis discussion as well as [10] and extend the studied model in this study in stochastic cases.

Acknowledgements

Authors would like to greatly appreciate Prof. Bing-Yuan Cao, the editor of Fuzzy Information and Engineering journal for assigning a special issue on occasion of Prof. Lotfi Askar Zadeh who is the founder of fuzzy sets theory.
Disclosure statement
No potential conflict of interest was reported by the authors.

Funding
The authors did not receive any funding for this research.

References
[1] Delgado M, Verdegay JL, Vila MA. A general model for fuzzy linear programming. Fuzzy Sets Syst. 1989;29:21–29.
[2] Wu SY, Fang SC, Lin CH. Analytic center based cutting plane method for linear semi-infinite programming. In: López MA, editor. Semi-infinite programming: recent advances. Dordrecht: Kluwer; 2001. p. 221–233.
[3] Adamo JM. Fuzzy decision trees. Fuzzy Sets Syst. 1980;4:207–219.
[4] Goberna MA, López MA. Linear semi-infinite optimization theory: an updated survey. Eur J Oper Res. 2002;143:390–405.
[5] Nasseri S, Behmanesh E, Faraji P, et al. Semi-infinite programming to solve linear programming with triangular fuzzy coefficients. Ann Fuzzy Math Inf. 2013;1:213–226.
[6] Attari H, Nasseri SH. New concepts of feasibility and efficiency of solutions in fuzzy mathematical programming problems. Fuzzy Inf Eng. 2014;6(2):203–221.
[7] Cao BY. Optimal models and methods with fuzzy quantities. Berlin: Springer-Verlag; 2010. (Studies in fuzziness and soft computing; Vol. 248).
[8] Fang SC, Hu CF, Wang HF, et al. Linear programming with fuzzy coefficients in constraints. Comput Math Appl. 1999;37:63–76.
[9] Hettich R, Kortanek KO. Semi-infinite programming: theory, methods and applications. SIAM Rev. 1993;35:380–429.
[10] Goberna MA, Gómez S, Guerra F, et al. Sensitivity analysis in linear semi-infinite programming: perturbing cost and right-hand-side coefficients. Eur J Oper Res. 2007;181:1069–1085.