ELSEVIER

Fuzzy Sets and Systems 109 (2000) 35–53

FUZZY sets and systems

www.elsevier.com/locate/fss

Evolutionary algorithm solution to fuzzy problems: Fuzzy linear programming

James J. Buckley[a,*], Thomas Feuring[b,1]

[a]*Department of Mathematics, University of Alabama at Birmingham, Birmingham, AL 35294, USA*
[b]*Institut für Informatik, Westfälische Wilhelms-Universität Münster, Einsteinstraße 62, 48149 Münster, Germany*

Received July 1997; received in revised form January 1998

Abstract

In this paper we wish to find solutions to the fully fuzzified linear program where all the parameters and variables are fuzzy numbers. We first change the problem of maximizing a fuzzy number, the value of the objective function, into a multi-objective fuzzy linear programming problem. We prove that fuzzy flexible programming can be used to explore the whole undominated set to the multi-objective fuzzy linear program. An evolutionary algorithm is designed to solve the fuzzy flexible program and we apply this program to two applications to generate good solutions.

Keywords: Mathematical programming; Fuzzy linear programming; Evolutionary algorithms

1. Introduction

In this paper we wish to construct solutions to the fully fuzzified linear program (written FFLP)

$$\max \quad \overline{Z} = \overline{C}_1\overline{X}_1 + \cdots + \overline{C}_n\overline{X}_n \tag{1}$$

$$\text{s.t.} \quad \overline{A}_{i1}\overline{X}_1 + \cdots + \overline{A}_{in}\overline{X}_n \leqslant \overline{B}_i, \quad 1 \leqslant i \leqslant m, \tag{2}$$

$$\overline{X}_i \geqslant 0 \quad \text{for all } i, \tag{3}$$

where the $\overline{C}_i$, $\overline{A}_{ij}$ and $\overline{B}_i$ are all triangular fuzzy numbers and the $\overline{X}_i$ are also triangular fuzzy numbers. We will code the FFLP as

$$\max \quad \overline{Z} = \overline{C}\,\overline{X} \tag{4}$$

$$\text{s.t.} \quad \overline{\boldsymbol{A}}\overline{X} \leqslant \overline{B}, \quad \overline{X} \geqslant 0, \tag{5}$$

where $\overline{C} = (\overline{C}_1, \ldots, \overline{C}_n)$, $\overline{X}^t = (\overline{X}_1, \ldots, \overline{X}_n)$, $\overline{B}^t = (\overline{B}_1, \ldots, \overline{B}_m)$ and $\overline{\boldsymbol{A}} = [\overline{A}_{ij}]$ a $m \times n$ matrix of fuzzy numbers. Before we discuss the contents of the paper, let us introduce the basic notation to be employed. We place a bar over a capital letter to denote a fuzzy subset of the real numbers. So $\overline{A}$, $\overline{B}$, $\overline{C}$, $\overline{X}$, etc., are all fuzzy subsets of the real numbers. We write $\overline{A}(x)$, a number in $[0, 1]$, for the membership function of $\overline{A}$ evaluated as x. An α-cut of $\overline{A}$, written as $\overline{A}[\alpha]$, is defined as $\{x \mid \overline{A}(x) \geqslant \alpha\}$, for $0 < \alpha \leqslant 1$. We separately

* Corresponding author. Tel.: +1-205-934-2154; fax: +1-205-934-9025.
E-mail address: buckley@math.uab.edu (J.J. Buckley)

[1] This work was supported with a grant from the German Academic Exchange Service (DAAD) based on the Hochschulsonderprogramm III of Bund and Länder while staying at the University of Alabama at Birmingham.

0165-0114/00/$ – see front matter
PII: S0165-0114(98)00022-0

specify $\overline{A}[0]$ as the closure of the union of all the $\overline{A}[\alpha]$, $0 < \alpha \leqslant 1$.

A triangular fuzzy number $\overline{N}$ is defined by three numbers $a_1 < a_2 < a_3$ where the graph of $\overline{N}(x)$ is a triangle with base on the interval $[a_1, a_3]$ and vertex at $x = a_2$. We specify $\overline{N}$ as $(a_1/a_2/a_3)$. The degenerate cases of $a_1 = a_2 < a_3$ and $a_1 < a_2 = a_3$, where we get half of a triangular fuzzy number, are allowed and we still call them triangular fuzzy numbers. A triangular-shaped fuzzy number $\overline{M}$ is partially defined by three number $a_1 < a_2 < a_3$ where: (1) the graph of $\overline{M}(x)$ is continuous and monotonically increasing from zero to one on $[a_1, a_2]$; (2) $\overline{M}(a_2) = 1$; and (3) the graph of $\overline{M}(x)$ is continuous and monotonically decreasing from one to zero on $[a_2, a_3]$. We write $\overline{M} \geqslant 0$ if $a_1 \geqslant 0$, $\overline{M} > 0$ if $a_1 > 0$, $\overline{M} \leqslant 0$, if $a_3 \leqslant 0$ and $\overline{M} < 0$ for $a_3 < 0$. We will use standard fuzzy arithmetic, from the extension principle, to evaluate sums, products, etc. of fuzzy numbers. The α-cut of a fuzzy number is always a closed and bounded interval.

To properly define the FFLP we must do two things: (1) define what we mean by $\max \overline{Z}$, or finding the "maximum" of a fuzzy number; and (2) explain what is meant by $\overline{E}_i \leqslant \overline{B}_i$, where $\overline{E}_i = \overline{A}_{i1}\overline{X}_1 + \cdots + \overline{A}_{in}\overline{X}_n$. How we are going to handle the problem of "maximizing" a fuzzy number is the topic of the next section and in Section 3 we discuss methods of evaluating fuzzy inequalities.

Searching for the optimal $\overline{X}$ to the FFLP is a very difficult problem. Let us assume that the meaning of "optimal" has been defined and the FFLP has an optimal solution. Then we do not know of an algorithm that will compute the exact value of the optimal $\overline{X}$. In the next section we turn the search for the optimal $\overline{X}$ into finding undominated solutions to a multi-objective fuzzy linear program. We also do not know of an algorithm that will produce undominated solutions to a multi-objective fuzzy linear program. So, we will employ a directed search method, called an evolutionary algorithm, to find (approximate) undominated solutions. Details about the evolutionary algorithm we used are in Section 4.

We consider two applications of the FFLP in Section 5. The first is the classical max problem of finding the best product mix to obtain the highest revenue. The second is the well-known min problem of finding the least cost diet which satisfies certain minimum requirements. In both cases we demonstrate that our evolutionary algorithm produces good approximate solutions.

The last section contains a brief summary, our conclusions, and directions for future research.

In [4] the authors advocated using fuzzy chaos to search for an optimal solution to the FFLP. However, this procedure is very inefficient being similar to pure random search. We decided instead to employ a directed search technique.

To the authors knowledge no research has treated the FFLP. That is, no one has discussed what it means for $\overline{X}$ to be a "solution" to a fuzzy linear program where all the parameters and variables are fuzzy. Many papers have been written where the $\overline{C}_i$ are fuzzy numbers, or the $\overline{A}_{ij}$ and $\overline{B}_j$ are fuzzy numbers. See [13,15] for a survey of fuzzy linear programming. We believe our method of changing the FFLP into a multi-objective fuzzy linear program is an important step in analyzing the FFLP.

2. Maximizing $\overline{Z}$

$\overline{Z}$ will be a triangular-shaped fuzzy number like the one shown in Fig. 1. Let A_1 be the area under the graph from z_1 to z_2 and A_2 the area under the graph from z_2 to z_3. In a max problem we will max z_2 and A_2 and min A_1. The reason for this decision is: (1) we wish to make $\overline{Z}$ as large as possible so we make z_2 (the most probable value) as large as possible; (2) we want the possibility of exceeding z_2 to be large so we max A_2; and (3) we wish the possibility of obtaining less than z_2 to be small so we min A_1. One cannot directly maximize $\overline{Z}$ since it is a fuzzy number. The analogy is from risk theory where $\overline{Z}$ would be the probability density function of a random variable Z we wish to maximize. There one may [7,16] maximize the expected value of Z, minimize the variance of Z and maximize the skewness to the right of the expected value.

So, for $\max \overline{Z}$ we have a multi-objective optimization problem

$$[\sup z_2, \sup A_2, \inf A_1], \tag{6}$$

for feasible $\overline{X}$, where sup = supremum and inf = infimum. We must use sup and inf because there is no guarantee that $\max z_2$, $\max A_2$ or $\min A_1$ exist.

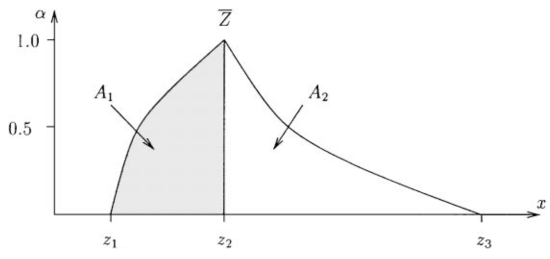


Fig. 1. Graphical description of z_2, A_2 and A_1.

There are a number of ways to change the multi-objective problem into a single objective problem [9,16]. We will consider, in this paper, only two methods both described in detail below. But first we need to introduce more notation and define the concepts of dominance and the undominated set.

Let $\mathscr{F}$ be the set of feasible $\overline{X} = (\overline{X}_1, \dots, \overline{X}_n)$ for the FFLP. That is, $\mathscr{F}$ contains all $\overline{X}$ such that $\overline{X}_i \geqslant 0$; and $\sum_{j=1}^{n} \overline{A}_{ij}\overline{X}_j \leqslant \overline{B}_i$ for $1 \leqslant i \leqslant m$. We will discuss inequalities between fuzzy numbers in the next section, so for now we assume we have defined the meaning of $\leqslant$ between fuzzy numbers. We assume that $\overline{C}_i \geqslant 0$, $\overline{A}_{ij} \geqslant 0$ and $\overline{B}_i \geqslant 0$ for all i, j so that $\overline{X} = 0$ (real number zero) is feasible. That is, $\mathscr{F}$ is always non-empty. Possible extensions to other values of $\overline{C}_i$, $\overline{A}_{ij}$ or $\overline{B}_i$ will be discussed in the last section of the paper. We must, at this time, make one more assumption about the $\leqslant$ we will use. If $\overline{X} \in \mathscr{F}$, then we assume that $\overline{Z} = \sum_{i=1}^{n} \overline{C}_i \overline{X}_i$ is bounded. What this means is that there is a $M > 0$ so that $\overline{Z}$ is a fuzzy subset of $[0, M]$. The $\leqslant$ we will use will assure us that $\overline{Z}$ is bounded.

Let $\sup z_2 = b_1$ for $\overline{X} \in \mathscr{F}$, $\sup A_2 = b_2$ for feasible $\overline{X}$ and $\sup A_1 = b_3$ for $\overline{X}$ in $\mathscr{F}$. All these supremums exist since $\overline{Z}$ is in $[0, M]$. In fact, z_2, A_1 and A_2 are all bounded above by M for $\overline{X}$ in $\mathscr{F}$. Since $\overline{X} = 0$ is feasible, then a possible value for $\overline{Z}$ is also zero. Hence, $\min z_2 = \min A_1 = \min A_2 = 0$ for feasible $\overline{X}$. It will be convenient, for the rest of this paper, to change the objective on A_1 to be supremum instead of infimum. Set $A'_1 = b_3 - A_1$ and now we wish to obtain $\sup A'_1$ in place of $\inf A_1$. For $\max \overline{Z}$ we have the multi-objective problem

$$[\sup z_2, \sup A_2, \sup A'_1], \tag{7}$$

for $\overline{X} \in \mathscr{F}$.

If $\overline{X}_a$ and $\overline{X}_b$ are in $\mathscr{F}$, let $\overline{Z}_a$ ($\overline{Z}_b$) be the value of the objective function using $\overline{X}_a$ ($\overline{X}_b$). Next set $(z_{2a}, A_{2a}, A'_{1a})$ (and $(z_{2b}, A_{2b}, A'_{1b})$) to be the value of (z_2, A_2, A'_1) for $\overline{Z}_a$ ($\overline{Z}_b$). We say that $\overline{X}_a$ dominates $\overline{X}_b$ if $z_{2a} \geqslant z_{2b}$, $A_{2a} \geqslant A_{2b}$, $A'_{1a} \geqslant A'_{1b}$ and at least one of these inequalities is a strict inequality. For example, if $z_{2a} = z_{2b}$, $A_{2a} > A_{2b}$, $A'_{1a} = A'_{1b}$, then $\overline{X}_a$ dominates $\overline{X}_b$. Let $\overline{UD}$ be all the undominated $\overline{X}$ in $\mathscr{F}$. $\overline{UD}$ could be empty.

Define the mapping $\Gamma : \mathscr{F} \to R^3$ as $\Gamma(\overline{X}) = (z_2, A_2, A'_1)$. For notational convenience set $e = (e_1, e_2, e_3) = (z_2, A_2, A'_1)$. Define $\mathscr{R} = \Gamma(\mathscr{F})$. $\mathscr{R}$ is called the objective function space. For the rest of this section we will be primarily interested in the undominated points in $\mathscr{R}$ which can be easily translated back to $\overline{UD}$.

$\overline{Z}$ is in $[0, M]$ so we see that $\mathscr{R}$ is a subset of $[0, M]^3$, or that $\mathscr{R}$ is bounded. Define $\overline{\mathscr{R}} = \text{closure}(\mathscr{R})$. Now $\overline{\mathscr{R}}$ is closed and bounded (a subset of $[0, M]^3$) so $\overline{\mathscr{R}}$ is compact.

For two vectors v and w in R^3 we define $v \leqslant w$ if $v_1 \leqslant w_i$, $i = 1, 2, 3$. We say v dominates w if $w_i \leqslant v_i$ for all i and at least one of the inequalities is a strict inequality. If D is a non-empty subset of R^3, set $UD(D)$ to be the set of undominated vectors in D. Of course, in general, $UD(D)$ could be empty.

Theorem 1. *$UD(\overline{\mathscr{R}})$ is not empty.*

Proof. Let $\odot$ be a chain in $\overline{\mathscr{R}}$ [10, p. 15]. $\odot$ is partially ordered by $\leqslant$ and then, by Kuratowski's lemma [10, p. 33], $\odot$ is contained in a maximal chain η. Let $w = \sup\{v \mid v \in \eta\}$, which exists because η in $\overline{\mathscr{R}}$ and $\overline{\mathscr{R}}$ is bounded. Also, w is in $\overline{\mathscr{R}}$ because $\overline{\mathscr{R}}$ is closed. The vector w is undominated because any vector in $\overline{\mathscr{R}}$ dominating w could be added to η to produce a larger chain. This would contradict η being maximal.

If we knew $\mathscr{R}$ was closed, then we could prove that $UD(\mathscr{R})$ is non-empty. However, we have very little structure available in the FFLP, so we are unable to argue that $\mathscr{R}$ is closed. Since we have not defined the meaning of $\leqslant$ between fuzzy numbers we cannot say very much about $\mathscr{F}$. If we could show that $\mathscr{F}$ is compact and Γ is continuous, then surely $\mathscr{R}$ would be closed.

However, if $UD(\mathscr{R}) \neq \emptyset$, then we easily see that $\overline{UD} \neq \emptyset$. Let $e \in UD(\mathscr{R})$. Then choose any $\overline{X}$ in $\mathscr{F}$ so that $\Gamma(\overline{X}) = e$. We see that $\overline{X} \in \overline{UD}$.

$\overline{UD}$ is called the Pareto optimal set, or the efficient set, for the FFLP [9,16]. If $\overline{UD} \neq \emptyset$, then surely we would want to consider only $\overline{X} \in \overline{UD}$ for possible solutions to the FFLP. If $\overline{UD} = \emptyset$ then as we will explain below, we will look for $\overline{X} \in \mathscr{F}$ that produce $\Gamma(\overline{X}) = e$ which approximates certain v in $\overline{\mathscr{R}}$, as possible solutions.

We assume that $b = (b_1, b_2, b_3)$ is not in $\mathscr{R}$ because if $b \in \mathscr{R}$ there is no need for a compromise solution to the FFLP. Now we will consider two methods of exploring $\overline{UD}$, when it is non-empty. Both methods generate compromise solutions since we cannot obtain the ideal point $b = (b_1, b_2, b_3)$ for $\overline{X}$ in $\mathscr{F}$.

2.1. First method (parametric programming)

Let $\lambda_i \geqslant 0$, $i = 1,2,3$ and $\lambda_1 + \lambda_2 + \lambda_3 = 1$. We change the multi-objective problem into a single objective as follows

$$\sup \quad [\lambda_1 z_2 + \lambda_2 A_2 + \lambda_3 A_1'] \tag{8}$$

$$\text{s.t.} \quad \overline{A}\,\overline{X} \leqslant \overline{B}, \quad \overline{X} \geqslant 0. \tag{9}$$

We call this problem a fuzzy parametric programming problem (FPP). If there are only crisp numbers in the problem, it is called a parametric programming problem [16].

Let $\Omega = \{\lambda = (\lambda_1, \lambda_2, \lambda_3) \mid \lambda_i \geqslant 0,\ \lambda_1 + \lambda_2 + \lambda_3 = 1\}$. Define $\Psi : \overline{\mathscr{R}} \to R$ as $\Psi(v) = \lambda_1 v_1 + \lambda_2 v_2 + \lambda_3 v_3$ for $\lambda \in \Omega$. Given $\lambda \in \Omega$, the FPP becomes sup $\Psi(e)$ for $\overline{X}$ in $\mathscr{F}$, or sup $\Psi(e)$, $e \in \mathscr{R}$. Recall, $e_1 = z_2$, $e_2 = A_2$ and $e_3 = A_1'$.

Theorem 2. *Let $\lambda \in \Omega$. There is a $v^* \in \overline{\mathscr{R}}$ so that $\sup_{e \in \mathscr{R}} \Psi(e) = \Psi(v^*)$.*

Proof. First, we know the supremum exists because $\mathscr{R}$ is bounded and Ψ is continuous. Let $\sup_{e \in \mathscr{R}} \Psi(e) = \tau$. From the definition of supremum, there is a sequence $e^{(k)} \in \mathscr{R}$ so that $\Psi(e^{(k)}) \to \tau$. This sequence is in compact $\overline{\mathscr{R}}$ so there is a subsequence $e^{(k_i)} \to v^* \in \overline{\mathscr{R}}$. Since Ψ is continuous we have

$$\Psi(e^{(k_i)}) \to \tau = \Psi(v^*). \tag{10}$$

Hence, the supremum equals $\Psi(v^*)$ with v^* in $\overline{\mathscr{R}}$.

Now let $\Omega^* = \{\lambda \mid \lambda_i > 0$ for all i and $\lambda_1 + \lambda_2 + \lambda_3 = 1\}$.

Theorem 3. *Let $\lambda \in \Omega^*$ and $\sup_{e \in \mathscr{R}} \Psi(e) = \Psi(v^*)$. If $v^* \in \mathscr{R}$, then $v^* \in UD(\mathscr{R})$.*

Proof. Suppose v^* is dominated by w in $\mathscr{R}$. For example, let $v_1^* = w_1$, $v_2^* < w_2$ and $v_3^* = w_3$. Then, since all the $\lambda_i > 0$, we get $\Psi(v^*) < \Psi(w)$, contradicting $\Psi(v^*)$ was the supremum over $\mathscr{R}$.

So, given $\lambda \in \Omega^*$ the FFP has a solution v^* in $\overline{\mathscr{R}}$. If v^* is in $\mathscr{R}$, then any $\overline{X}$ in $\mathscr{F}$, $\Gamma(\overline{X}) = v^*$ solves the FPP and $\overline{X}$ is undominated. If v^* is not in $\mathscr{R}$, or v^* is in $\overline{\mathscr{R}} - \mathscr{R}$, there is a sequence $e^{(k)}$ in $\mathscr{R}$ converging to v^*. Let $\overline{X}^{(k)}$ be a sequence in $\mathscr{F}$ so that $\Gamma(\overline{X}^{(k)}) = e^{(k)}$. Then we call $\overline{X}^{(k)}$ a *solution sequence* because $\Psi(\Gamma(\overline{X}^{(k)})) \to \sup_{e \in \mathscr{R}} \Psi(e)$. When v^* is not in $\mathscr{R}$ we want our evolutionary algorithm to produce a *solution sequence*.

If $UD(\mathscr{R})$ is not empty, can we use this method, fuzzy parametric programming, to explore $\overline{UD}$? Probably not because even for a crisp multi-objective nonlinear programming problem one may not obtain the whole undominated set by varying λ over Ω^* [8]. But, one does get the whole Pareto optimal set to a crisp multi-objective linear programming problem by varying λ throughout Ω^* [14]. However, our second method may be used to generate all of $\overline{UD}$.

2.2. Second method

This method we call flexible programming [15] which is similar to goal programming [16]. For each objective we first set up a fuzzy goal $\overline{G}_i$. This section is modeled after the methods in [2,3]. We set up the $\overline{G}_1$ for z_2 as follows: if $0 \leqslant c_1 < b_1$, then

$$\overline{G}_1(z_2) = \begin{cases} 0, & z_2 < c_1, \\ 0.1 + 0.9\left(\dfrac{z_2 - c_1}{b_1 - c_1}\right), & c_1 \leqslant z_2 \leqslant b_1, \\ 1, & z_2 \geqslant b_1, \end{cases} \tag{11}$$

and if $c_1 = b_1$

$$\overline{G}_1(z_2) = \begin{cases} 0, & z_2 < b_2, \\ 1, & z_2 \geqslant b_1. \end{cases} \tag{12}$$

The graph of $\overline{G}_1$, for $0 \leqslant c_1 < b_1$, is shown in Fig. 2. The reason for the discontinuity at c_1 is explained later in the proof of Theorem 5.

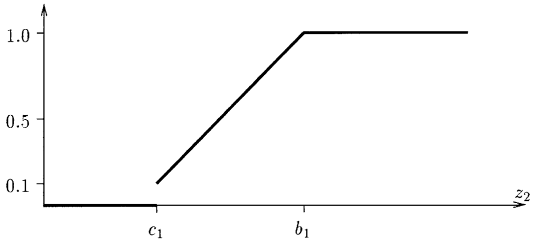


Fig. 2. Fuzzy goal $\overline{G}_1$ for z_2.

The structure of $\overline{G}_2$, for A_2, is similar to $\overline{G}_1$: if $0 \leqslant c_2 < b_2$, then

$$\overline{G}_2(A_2) = \begin{cases} 0, & A_2 < c_2, \\ 0.1 + 0.9\left(\dfrac{A_2 - c_2}{b_2 - c_2}\right), & c_2 \leqslant A_2 \leqslant b_2, \\ 1, & A_2 \geqslant b_2, \end{cases} \tag{13}$$

and if $c_2 = b_2$

$$\overline{G}_2(A_2) = \begin{cases} 0, & A_2 < b_2, \\ 1, & A_2 \geqslant b_2. \end{cases} \tag{14}$$

$\overline{G}_3$ for A'_1 is the same as $\overline{G}_1$ and $\overline{G}_2$. If $0 \leqslant c_3 < b_3$, then

$$\overline{G}_3(A'_1) = \begin{cases} 0, & A'_1 < c_3, \\ 0.1 + 0.9\left(\dfrac{A'_1 - c_3}{b_3 - c_3}\right), & c_3 \leqslant A'_1 \leqslant b_3, \\ 1, & A'_1 \geqslant b_3, \end{cases} \tag{15}$$

and if $c_3 = b_3$

$$\overline{G}_3(A'_1) = \begin{cases} 0, & A'_1 < b_3, \\ 1, & A'_1 \geqslant b_3. \end{cases} \tag{16}$$

Let $c = (c_1, c_2, c_3)$.

Now we can form the fuzzy flexible programming problem (FFP). Let $G(z_2, A_2, A'_1) = \overline{G}_1(z_2)\overline{G}_2(A_2)\overline{G}(A'_1)$. Then we wish to solve

$$\sup \quad G(z_2, A_2, A'_1) \tag{17}$$

$$\text{s.t.} \quad \overline{\boldsymbol{A}}\,\overline{X} \leqslant \overline{B}, \quad \overline{X} \geqslant 0. \tag{18}$$

In order to avoid non-practical solutions we need to restrict the choice of c. Let $\mathscr{C} = \{c \mid 0 \leqslant c_i \leqslant b_i$, there is a $\overline{X}$ in $\mathscr{F}$ so that $G(z_2, A_2, A'_1) \neq 0\}$. Or alternatively, $\mathscr{C} = \{c \mid 0 \leqslant c_i \leqslant b_i$, there is an $e \in \mathscr{R}$ so that $G(e) \neq 0\}$. If G is identically zero on $\mathscr{F}$, then all of $\mathscr{F}$ solves the FFP problem. Having the solution the whole feasible set does not, in general, seem to have any practical use, so we exclude this case.

Theorem 4. *Let $c \in \mathscr{C}$. There is a v^* in $\overline{\mathscr{R}}$ so that $\sup_{e \in \mathscr{R}} G(e) = G(v^*)$.*

Proof. The supremum exists since $G \leqslant 1$. Let $\sup_{e \in \mathscr{R}} G(e) = \tau$. We know $\tau > 0$ from the choice of c in $\mathscr{C}$. By the definition of the supremum there is a sequence $e^{(k)}$ in $\mathscr{R}$ so that $G(e^{(k)}) \to \tau$.

If $\Pi = \{w \mid w \geqslant c\}$, then G is continuous on Π. The sequence $e^{(k)}$ is in compact $\overline{R}$ so there is a subsequence $e^{(k_i)}$ which converges to some $v^* \in \overline{\mathscr{R}}$. Also, $G(e^{(k_i)}) \to \tau > 0$.

Because $\tau > 0$ there is an N so that $e^{(k_i)} \in \Pi$, $i \geqslant N$. G being continuous on Π implies $G(e^{(k_i)}) \to G(v^*)$. Hence $\tau = G(v^*)$.

Theorem 5. (a) *Let $c \in \mathscr{C}$. If v^* solves the FFP (Theorem 4) and if $v^* \in \mathscr{R}$, then v^* is undominated in $\mathscr{R}$.*

(b) *Let $e^* \in UD(\mathscr{R})$. Then we can choose $c \in \mathscr{C}$ so that e^* solves the FFP.*

Proof. (a) Suppose $\sup_{e \in \mathscr{R}} G(e) = G(v^*)$. By the choice c in $\mathscr{C}$ we have $G(v^*) > 0$. If there is a $w \in \mathscr{R}$ which dominates v^*, then $G(v^*) < G(w)$, a contradiction. So, v^* is not dominated by any w in $\mathscr{R}$.

(b) If $e^* \in UD(\mathscr{R})$, then set $c = e^*$. Since $e^* \in \mathscr{R}$ we have $c \in \mathscr{C}$. Now let $v^* \in \overline{\mathscr{R}}$ solve the FFP: $\sup_{e \in \mathscr{R}} G(e) = G(v^*)$. We argue that $v^* = e^*$ so e^* solves the FFP.

For every e in $\mathscr{R}$, $e \neq e^*$, we must have $e_i < e^*_i$, for some i, because e^* is undominated. Hence, $G(e) = 0$ for all $e \in \mathscr{R}$, $e \neq e^*$, and $G(e^*) = (0.1)^3$. Hence $\sup_{e \in \mathscr{R}} G(e) = (0.1)^3 = G(e^*)$. This means e^* solves the FFP.

Given $c \in \mathscr{C}$ let v^* solve the FFP. If $v^* \in \mathscr{R}$, then any $\overline{X}$ in $\mathscr{F}$ so that $\Gamma(\overline{X}) = v^*$ solves the FFP and $\overline{X} \in \overline{UD}$. When $v^* \in \overline{\mathscr{R}} - \mathscr{R}$, then (as in the previous subsection) we have a solution sequence $\overline{X}^{(k)}$ in $\mathscr{F}$ so that $G(\Gamma(\overline{X}^{(k)})) \to \sup_{e \in \mathscr{R}}(G(e))$.

This method may be used to explore all of $\overline{UD}$. However, it has a few drawbacks: (1) finding $\mathscr{C}$; and (2) determining values for the b_i. To make this procedure operational, and to be employed in Section 4, is to (1) estimate the b_i with $b_i' > b_i$; and (2) choose the c_i in $[0, b_i']$ and reduce a value of c_i when $G(z_2, A_2, A_1') = 0$ for all $\overline{X} \in \mathscr{F}$. Theorem 5 is still true for $b_i' > b_i$ if we have a c so that $G(z_2, A_2, A_1') > 0$.

Even after estimating $b_i' > b_i$ and obtaining a c in $\mathscr{C}$ we still have the problem of showing the results of the evolutionary algorithm converge to v^*, a solution to the FFP. The evolutionary algorithm will produce a feasible sequence $\overline{X}^{(k)}$. We need to make sure we do get $e^{(k)} = \Gamma(\overline{X}^{(k)})$ converging to (approximating) v^*, where $G(v^*) = \sup_{e\in\mathscr{R}} G(e)$, for the given $c \in \mathscr{C}$. We discuss the problem of premature convergence again in Section 4.

3. Fuzzy inequality

In this section we will discuss some methods of evaluating $\overline{M} \leqslant \overline{N}$, where $\overline{M}$ and $\overline{N}$ are fuzzy numbers, that we may use in our evolutionary algorithm. $\overline{M}$ will be $\sum_{j=1}^{n} \overline{A}_{ij}\overline{X}_j = \overline{E}_i$, the left-hand side of a fuzzy constraint, and $\overline{N}$ will be $\overline{B}_i$ the right-hand side of the fuzzy constraint, in a FFLP.

This section is not intended to be a survey of all methods presented in the literature of evaluating $\overline{M} \leqslant \overline{N}$. See [6] for an excellent review of this literature up to 1992. Our criteria for choosing a method of evaluating $\overline{M} \leqslant \overline{N}$ is: (1) $\overline{Z}$ must be bounded; (2) it is fairly easily incorporated into our evolutionary algorithm; and (3) it seems "reasonable" to the authors. We want procedures that will produce a "true" or "false" for $\overline{M} \leqslant \overline{N}$ given two fuzzy numbers. In this way we can test a given $\overline{X}$, $\overline{X}_i \geqslant 0$, as being feasible or not feasible.

Twenty-three different ranking methods were presented in [6]. Almost all of these procedures had to be rejected because they did not guarantee that $\overline{Z}$ would be bounded. The problem was that all of these methods assumed that in evaluating $\overline{M} \leqslant \overline{N}$, $\overline{M}$ and $\overline{N}$ are triangular fuzzy numbers. But in evaluating $\overline{E}_i \leqslant \overline{B}$ the fuzzy number $\overline{E}_i$ is not a triangular fuzzy number. Fig. 3 shows the shape of $\overline{E}_i$ (discussed below). Because the graph of $\overline{E}_i$ is concave up on the interval $[e_{i1}, e_{i2}]$ (see Fig. 3) almost all of the methods of evaluating fuzzy inequality in [6] would say that $\overline{E}_i \leqslant \overline{B}_i$ is true when, holding everything else fixed, e_{i3} grows larger and larger. This means that $\overline{E}_i$ can be unbounded and still have $\overline{E}_i \leqslant \overline{B}_i$ true. It follows that the $\overline{X}_i$ can be unbounded which implies that $\overline{Z}$ is unbounded.

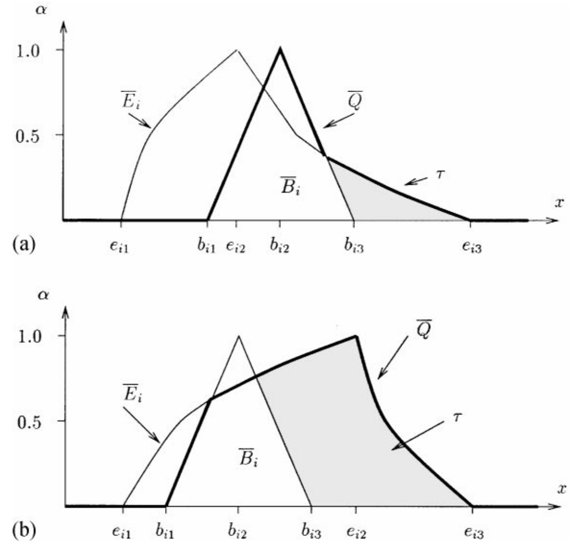


Fig. 3. Fuzzy max and $d(\overline{B}_i, \overline{Q})$. (a) Fuzzy max when $e_{i2} < b_{i2}$. (b) Fuzzy max when $e_{i2} > b_{i2}$.

3.1. First method

We first need to present the fuzzy max (written $\overline{\max}$) of two fuzzy numbers. If $\overline{O} = \overline{\max}(\overline{M}, \overline{N})$, then

$$\overline{O}(z) = \sup\left\{\min(\overline{M}(x), \overline{N}(y)) \mid \max(x, y) = z\right\}. \tag{19}$$

The authors in [12] give a detailed study of the properties of $\overline{\max}$ and $\overline{\min}$ (fuzzy min).

Next we define the Hamming distance between $\overline{M}$ and $\overline{N}$. The Hamming distance, $d(\overline{M}, \overline{N})$ is defined as

$$d(\overline{M}, \overline{N}) = \int_{-\infty}^{\infty} |\overline{M}(x) - \overline{N}(x)|\, dx. \tag{20}$$

Clearly, d is a metric on the space of continuous fuzzy numbers (those whose membership function is continuous).

Then we say $\overline{M} \leqslant \overline{N}$ is true whenever [11,12]

$$d(\overline{N}, \overline{\max}(\overline{M}, \overline{N})) \leqslant d(\overline{M}, \overline{\max}(\overline{M}, \overline{N})). \tag{21}$$

This is simply a fuzzification of $x \leqslant y$ if and only if $\max(x, y) = y$ for real x, y. A numerical example showing $\overline{M} \leqslant \overline{N}$ by this method is in [12, pp. 407–408]. We will call this procedure for evaluating fuzzy inequalities Kerre's method.

We now argue, using this procedure of evaluation $\overline{M} \leqslant \overline{N}$, $\overline{Z}$ will be bounded. That is, there is a positive number M so that $\overline{Z}$ will be a fuzzy number in $[0, M]$. Consider the situation shown in Fig. 3. The support of $\overline{X}_i$ is $[x_{i1}, x_{i3}]$, the support of $\overline{E}_i = [e_{i1}, e_{i3}]$, and the support of $\overline{B}_i = [b_{i1}, b_{i3}]$. Set $\overline{Q}_i = \overline{\max}(\overline{E}_i, \overline{B}_i)$. The heavy line segment defines $\overline{Q}_i$. In computing $d(\overline{B}_i, \overline{Q}_i)$ we need to compute τ, the area of the shaded region shown in Figs. 3(a) and (b), since $d(\overline{B}_i, \overline{Q}_i) = \tau$. The way we have drawn $\overline{E}_i$ in Fig. 3 is correct since the graph of $y = \overline{E}_i(x)$ will be decreasing, and concave up, on $[e_{i2}, e_{i3}]$. This follows from the fact that the $\overline{A}_{ij}$ and $\overline{X}_j$ are all triangular fuzzy numbers. We see that as e_{i3} grows, holding $\overline{B}_i$, e_{i1} and e_{i2} fixed, τ will get larger, and eventually $\overline{E}_i \leqslant \overline{B}_i$ will not be true. As e_{i3} gets larger $d(\overline{E}_i, \overline{Q}_i)$ will also change but its value is bounded and cannot grow without bound. So e_{i3} must be bounded for $\overline{X}$ to be feasible. But e_{i3} is a positive linear combination of the x_{i3} which implies that the x_{i3} must be bounded. But, if the support of $\overline{Z}$ is $[z_1, z_3]$, z_3 is bounded since it is a positive linear combination of the x_{i3}. Hence, $\overline{Z}$ is bounded. In Appendix A we show how we estimate the b_i with $b_i' > b_i$ when we use this method of evaluating $\overline{E}_i \leqslant \overline{B}_i$.

3.2. Second method

An obvious procedure for evaluating $\overline{M} \leqslant \overline{N}$ is to fuzzify the crisp $\leqslant$. Let

$$\tau = \sup \left\{ \min(\overline{M}(x), \overline{N}(y)) \mid x \leqslant y \right\}. \tag{22}$$

Then, give some α^* in $(0, 1]$, we say $\overline{M} \leqslant \overline{N}$ is true if $\tau \geqslant \alpha^*$. The value of α^* is up to the user and $\alpha^* = 0.8$ could be employed.

We now argue that using this method of evaluating $\overline{M} \leqslant \overline{N}$, $\overline{Z}$ can be unbounded. Hence, we will not employ this procedure in our evolutionary algorithm. Let us assume that $\alpha^* = 0.8$.

We switch to $\overline{E}_i = \overline{M}$ and $\overline{B}_i = \overline{N}$. We see that as long as $e_{i2} \leqslant b_{i2}$ we get $\tau = 1$ and $\overline{E}_i \leqslant \overline{B}_i$. This puts no constraint on e_{i3}. As e_{i3} grows larger and larger we have $\overline{E}_i \leqslant \overline{B}_i$ as long as $e_{i2} \leqslant b_{i2}$. Since e_{i3} is not bounded, the x_{i3} are not bounded, and $\overline{Z}$ is not bounded.

3.3. Third method

The third method of ranking fuzzy numbers we focused on is presented by Chen in [5]. Like in Section 3.1 a score is computed for each fuzzy number which is needed for ranking. The fuzzy set with the highest score is the largest fuzzy number. In order to rank triangular fuzzy numbers $\overline{N} = (n_1/n_2/n_3)$ and $\overline{M} = (m_1/m_2/m_3)$ Chen defined a fuzzy max and a fuzzy min where the supports of fuzzy max and min is $[x_{\min}, x_{\max}]$ where

$$x_{\min} = \min(n_1, m_1), \tag{23}$$

$$x_{\max} = \max(n_3, m_3). \tag{24}$$

Fuzzy min and fuzzy max are triangular fuzzy numbers with membership degree one at the left and the right limit of the support, respectively (see Fig. 4). The membership functions are

$$\mu_{\min}(x) = \begin{cases} \dfrac{x - x_{\max}}{x_{\min} - x_{\max}}, & x_{\min} \leqslant x \leqslant x_{\max}, \\ 0, & \text{otherwise}, \end{cases} \tag{25}$$

$$\mu_{\max}(x) = \begin{cases} \dfrac{x - x_{\min}}{x_{\max} - x_{\min}}, & x_{\min} \leqslant x \leqslant x_{\max}, \\ 0, & \text{otherwise}. \end{cases} \tag{26}$$

The intersection points between fuzzy max and $\overline{M}$ and $\overline{N}$ as well as the intersection points between fuzzy min and $\overline{M}$ and $\overline{N}$ are needed for computing the final scores. We compute

$$\mu_{\mathrm{R}}(\overline{M}) = \sup_x(\min(\mu_{\max}(x), \overline{M}(x))), \tag{27}$$

$$\mu_{\mathrm{L}}(\overline{M}) = \sup_x(\min(\mu_{\min}(x), \overline{M}(x))), \tag{28}$$

$$\tag{29}$$

where $\mu_{\mathrm{R}}(\overline{M})$ indicates the max of the intersection point between fuzzy max and $\overline{M}$ and $\mu_{\mathrm{L}}(\overline{M})$ stands for the left score which is given by the max intersection point with fuzzy min. The larger $\mu_{\mathrm{R}}(\overline{M})$ is, the higher $\overline{M}$ should be ranked. On the other hand, a high value of $\mu_{\mathrm{L}}(\overline{M})$ and $\overline{M}$ is close to the fuzzy min, and therefore should be ranked lower. By combining both scores we get the final rating

$$\mu_{\mathrm{T}}(\overline{M}) = \tfrac{1}{2}\left(\mu_{\mathrm{R}}(\overline{M}) + (1 - \mu_{\mathrm{L}}(\overline{M})\right). \tag{30}$$

Similarly, we get $\mu_{\mathrm{T}}(\overline{N})$. We then say that $\overline{M} \leqslant \overline{N}$ is true if $\mu_{\mathrm{T}}(\overline{M}) \leqslant \mu_{\mathrm{T}}(\overline{N})$.

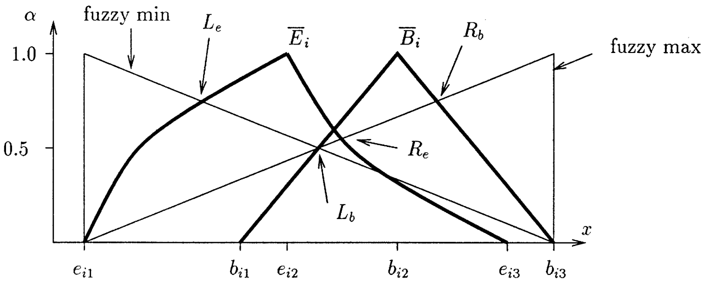


Fig. 4. Ranking fuzzy numbers based on Chen's method.

In Fig. 4 we used the notation $L_e = \mu_L(\overline{E}_i)$, $R_e = \mu_R(\overline{E}_i)$, $L_b = \mu_L(\overline{B}_i)$ and $R_b = \mu_R(\overline{B}_i)$.

In Appendix B we argue that using this method of evaluating fuzzy inequalities $\overline{Z}$ is bounded. So we will employ this procedure in our evolutionary algorithm. Also in Appendix B we show how to estimate the b_i with b_i', so that $b_i' > b_i$, and these b_i' are to be used in the fuzzy goals defined in Section 2.2.

4. Evolutionary algorithm

We will now discuss the evolutionary algorithm used to explore the set of undominated solutions in the application problems presented in the next section. As we mentioned earlier we think that using a directed search tool like evolutionary algorithms is efficient for finding good solutions to the FFLP. We used self implemented evolution strategies, which works on real numbers. A description of the algorithm is presented below, because some modifications need to be made in order to let evolutionary algorithms find an approximate solution to the FFP.

The main processes of evolutionary algorithms are recombination, mutation and selection. Unlike genetic algorithms no binary coding of the population members are necessary. Each element of the population is described by a set of triangular fuzzy numbers $\overline{X}_i$, $i = 1,\dots,n$, which is a feasible solution to the FFLP. In order to minimize the computational expense, each fuzzy number is represented by three real numbers $r_{i0}, r_{i1}, r_{i2} \in [x_{i1}, x_{i3}]$, where $r_{i0} = x_{i1}$ and $r_{i2} = x_{i3}$ are the limits of the support of $\overline{X}_i$ and r_{i1} is the central value of $\overline{X}_i$ (where the membership value is one). In order to compute the sum and product of fuzzy numbers α-cuts for each fuzzy number are calculated. The α-levels are used to produce the corresponding sum or product. Because the sum and product of triangular fuzzy numbers may not be triangular, but instead triangular-shaped fuzzy numbers, we need to store all these α-cuts so that we can use these triangular-shaped fuzzy numbers in further calculations. An additional value is added for each population member which represents their mutation rate. This value is self-adapted during the evolution process. Therefore, the population members are vectors $\pi \in R^{3n+1}$. Hence, an element of the population looks like

$$\pi = (p_0, p_1, p_2, p_3, \dots, p_5, \dots, p_{3(n-1)}, p_{3n-2}, p_{3n-1}, \sigma), \tag{31}$$

where σ stands for the mutation rate of the corresponding member. In the above equation we set $p_{3i+j} = r_{i+1,j}$ for $0 \leqslant i < n$ and $0 \leqslant j \leqslant 2$.

Let us assume that the population consists of λ individuals. Throughout the applications presented below we used a population size of 2000 elements. First, all elements of the population are randomly chosen according to the constraints. In order to get a suitable starting population, we randomly choose the central values of $\overline{X}_i$, p_{3i+1}, close to the solutions of the corresponding crisp linear programming problem. But a generated individual is only taken into the starting population if it satisfies the constraints.

The mutation rates of all members are set to an initial value. Bäck proposed in [1] to initialize $\sigma \approx 0.3$. However a much larger setting is required in our experiments in order to get better fitness values.

After initializing the evolution starts with the selection process. During selection, the fitness of the individuals is computed. The fitness of an individual is given by the objective function $G(z_2, A_2, A_1')$

(see Section 2.2). Therefore, the closer the value of $G(z_2, A_2, A_1')$ is to 1, the greater is the fitness of this member. Because selection is a deterministic process in evolution strategies the μ fittest individuals are chosen to build the offspring of the next generation by using recombination and mutation. In our experiments μ was set to 300.

The recombination process builds a temporary generation by applying a crossover operator to the $\mu \leqslant \lambda$ fittest members of the previous generation. For each individual of the temporary generation two "parents" π^{old_1} and π^{old_2} are randomly chosen from the μ fittest elements of the previous generation. In order to guarantee that the generated individual consists of triangular fuzzy numbers we have to ensure that the recombination operator produces members according to our coding scheme. Therefore, we used a multi-point crossover operator which only can choose points between two encoded fuzzy numbers as crossover points.

According to a user-defined crossover probability q a number $p = \lceil n * q \rceil$ is computed. Then, p crossover points c_i $(1 \leqslant i \leqslant p)$ are randomly generated from $\{0, 3, \ldots, 3(n-1)\}$ so that some of the crossover points c_i can be equal. Furthermore, $c_0 = 0$ and $c_{p+1} = 3n$ are set.

After ordering the crossover points we get $0 = c_0 \leqslant c_1 \leqslant c_2 \leqslant \cdots \leqslant c_p < c_{p+1} = 3n$. Now a temporary element is produced by changing the π^{old_1} and π^{old_2} according to the crossover points. Between two crossover points c_i and c_{i+1} $(0 \leqslant i \leqslant p)$ randomly a real value $r \in [0,1]$ is chosen so that the positions $c_i \leqslant k < c_{i+1}$ of the temporary individual are built by using equation

$$p_k^{\text{temp}} = p_k^{\text{old}_1} + r * (p_k^{\text{old}_2} - p_k^{\text{old}_1}) \tag{32}$$

for $0 \leqslant i \leqslant p$. The mutation rate σ^{temp} is also adjusted during the recombination process by using the same operator and we get

$$\sigma^{\text{temp}} = \sigma^{\text{old}_1} + r * (\sigma^{\text{old}_2} - \sigma^{\text{old}_1}) \tag{33}$$

for a randomly chosen $r \in [0,1]$. The new element (π^{temp}) is only taken into the temporary generation if it satisfies the constraints of the FFLP. By applying equations Eqs. (32) and (33) and verifying the constraints at least λ-times to randomly chosen parents we get the temporary generation.

The mutation process now generates the new generation by randomly changing the members $\pi^{\text{temp}} = (p_0^{\text{temp}}, \ldots, p_{3n-1}^{\text{temp}}, \sigma^{\text{temp}})$ of the temporary generation. One by one each individual is changed according to its own mutation rate. The mutation rate σ^{temp} of each member is first modified. Here we used equation

$$\sigma^{\text{new}} = \sigma^{\text{temp}} + \exp(\tau * \mathrm{N}(0,1)), \tag{34}$$

where $\mathrm{N}(0,1)$ stands for a normally distributed random variable having expectation value zero and standard deviation one. τ is an additional parameter which is set to $n^{-0.5}$ (see [1]). The global factor $\exp(\tau \mathrm{N}(0,1))$ allows for an overall change of the mutability for each individual. We need to guarantee that the mutated individual still consists of triangular fuzzy numbers. For each $i = 0, 3, 6, \ldots, 3(n-1)$ we do the following:

$$p_{3i+1}^{\text{new}} = p_{3i+1}^{\text{temp}} + \sigma^{\text{new}} * \mathrm{N}(0,1), \tag{35}$$

constrained so that p_{3i+1}^{new} lies in $(0, M_i)$,

$$p_{3i}^{\text{new}} = p_{3i}^{\text{temp}} + \sigma^{\text{new}} * \mathrm{N}(0,1), \tag{36}$$

constrained so that p_{3i}^{new} lies in $[0, p_{3i+1}^{\text{new}})$, and

$$p_{3i+2}^{\text{new}} = p_{3i+2}^{\text{temp}} + \sigma^{\text{new}} * \mathrm{N}(0,1), \tag{37}$$

constrained so that p_{3i+2}^{new} lies in $(p_{3i+1}^{\text{new}}, M_i]$, where M_i the upper bound we get with the procedure described in Appendix A or B. If the new individual satisfies the FFLP constraints it is taken into the new generation, otherwise the mutation process is applied on the same temporary population member until the constraints are satisfied. By repeating this process for each temporary individual we get the entire new generation.

Now the selection process finds the μ fittest individuals of the new generation. These three processes continue until the fitness of a population member is greater than a positive value or a user defined generation number is reached.

Before applications are presented in the next section we want to present an argument on why we think that the above described evolutionary algorithm will lead to near optimal solutions. Recall that in the fuzzy goal functions we are using b_i', with $b_i' > b_i$, so we cannot recognize an optimal solution by the value of the objective function. In fact, we experienced very small values of the objective function as the algorithm converged to an undominated solution. If one used $b_i' < b_i$ in the fuzzy goal functions, then we would get premature convergence.

First, an evolutionary algorithm is a directed search method, where the direction is given by the fitness function. In order to increase the fitness a new generation is produced by using the above described operators (crossover, mutation, selection). The mutation process however is the most important module in an evolutionary algorithm, because according to the mutation rate the individuals of a population are randomly modified. The higher the mutation rate, the greater are the changes of a population member and the steps of exploring the feasible set become larger. In this sense the mutation rate describes the step size of the directed search tool. Good results can be expected when the evolutionary algorithm is initialized with several randomly chosen populations as well as different settings for the mutation rate. In our experiments we used different starting values for σ and multiple random number initializations in order to find the best setting of the parameters which lead to the highest fitness values. Therefore, we believe we reduced the risk of premature convergence.

5. Applications

In this section we present two applications of our evolutionary algorithm solution to FFLP. In both cases the evolutionary algorithm is used to solve the FFP discussed in Section 2.2. The first application is the standard product mix problem and the second is the standard diet problem. The second application is a minimization problem so we have to adjust the goals defined in Section 2.2.

5.1. Product mix problem

A company produces three products P_1, P_2 and P_3 each of which must be processed through three departments D_1, D_2 and D_3. The approximate time, in hours, each P_i spends in each D_j is given in Table 1.

Table 1
Approximate times product P_i is in department D_j

Product	Department		
	D_1	D_2	D_3
P_1	6	12	2
P_2	8	8	4
P_3	3	6	1

Each department has only so much time available each week. These times can vary slightly from week to week so the following numbers are estimates of the maximum time available per week, in hours, for each department: (1) for D_1 288 h; (2) 312 h for D_2; and (3) D_3 has 124 h. Finally, the selling price for each product can vary a little due to small discounts to certain customers but we have the following average selling prices: (1) \$6 for P_1; (2) \$8 per unit for P_2, and (3) for P_3 \$6/unit. The company wants to determine the number of units to produce for each product per week to maximize its revenue.

Since all the numbers given are uncertain, we will model the problem as a FFLP. We substitute a triangular fuzzy number for each value given where the peak of the fuzzy number is at the number given. So, we have the following FFLP:

$$\max \quad \overline{Z} = (5.8/6/6.2)\overline{X}_1 + (7.5/8/8.5)\overline{X}_2 + (5.6/6/6.4)\overline{X}_3 \tag{38}$$

$$\text{s.t.} \quad (5.6/6/6.4)\overline{X}_1 + (7.5/8/8.5)\overline{X}_2 + (2.8/3/3.2)\overline{X}_3 \leqslant (283/288/293), \tag{39}$$

$$(11.4/12/12.6)\overline{X}_1 + (7.6/8/8.4)\overline{X}_2 + (5.7/6/6.3)\overline{X}_3 \leqslant (306/312/318), \tag{40}$$

$$(1.8/2/2.2)\overline{X}_1 + (3.8/4/4.2)\overline{X}_2 + (0.9/1/1.1)\overline{X}_3 \leqslant (121/124/127), \tag{41}$$

$$\overline{X}_1, \overline{X}_2, \overline{X}_3 \geqslant 0, \tag{42}$$

where the $\overline{X}_i$ are triangular fuzzy numbers for the amount to produce for P_i per week. We change this to a FFP

$$\sup G(z_2, A_2, A_1') \tag{43}$$

subject to the same constraints.

Next we need to find estimates of b_i (see Section 2.2) to construct the fuzzy goals (Appendices A and B). This depends on which inequality relation $\leqslant$ (see Section 3) we will use.

Using b_i' we next select values for c_i. We decide on three separate selections for the c_i in order to minimally explore the undominated set for the FFLP.

Table 2
Results for application 1 using different values for c_i (i=1, 2, 3)

	Chen's inequality	Kerre's inequality
b'	2936	2637
$c_1/c_2/c_3$	100/100/400	
Best fitness	0.156	0.028
z_2	$\overline{G}_1(1409.93) = 0.516$	$\overline{G}_1(392.33) = 0.204$
A_2	$\overline{G}_2(750.06) = 0.306$	$\overline{G}_2(235.57) = 0.148$
A_1	$\overline{G}_3(39.31) = 0.986$	$\overline{G}_3(200.15) = 0.919$
$c_1/c_2/c_3$	100/200/400	
Best fitness	0.142	0.019
z_2	$\overline{G}_1(1307.54) = 0.483$	$\overline{G}_1(163.05) = 0.122$
A_2	$\overline{G}_2(800.39) = 0.297$	$\overline{G}_2(352.82) = 0.156$
A_1	$\overline{G}_3(36.59) = 0.987$	$\overline{G}_3(83.21) = 0.967$
$c_1/c_2/c_3$	150/100/300	
Best fitness	0.154	0.026
z_2	$\overline{G}_1(1445.78) = 0.519$	$\overline{G}_1(392.32) = 0.188$
A_2	$\overline{G}_2(732.45) = 0.301$	$\overline{G}_2(235.58) = 0.148$
A_1	$\overline{G}_3(40.26) = 0.986$	$\overline{G}_3(200.12) = 0.923$

We used the solution to the crisp linear program to randomly generate the initial values of the $\overline{X}_i$ in the evolutionary algorithm. The crisp linear program is the one obtained by using the peak values (where the membership function is one) of all the $\overline{C}_i$, $\overline{A}_{ij}$ and $\overline{B}_i$. The crisp solution is $x_1 = 0$ for (P_1), $x_2 = 27$ (for P_2) and $x_3 = 16$ (for P_3) with $\max z = 312$. The results of the evolutionary algorithm, for the different selections of $\leqslant$ and the different values of the c_i are presented in Table 2.

The results in Table 2 were surprising in that Chen's method of evaluating $\leqslant$ produced much better results for z_2, A_2, and A_1. It looks like the feasible set is much larger using Chen's method.

Fig. 5 shows the "optimal" $\overline{X}_i$, for both methods, for one value of $c = (c_1, c_2, c_3)$. In Fig. 5(a) $\overline{X}_1$ was not shown because it was very close to zero.

Fig. 6 shows the value of the objective function $\overline{Z}$, corresponding to the values of the $\overline{X}_i$ given in Fig. 5. The pictures imply that $\overline{Z}$ is a triangular fuzzy number, but in fact it is a triangular-shaped fuzzy number very close to being a triangle.

5.2. Diet problem

A farmer has three products P_1, P_2 and P_3 which he plans to mix together to feed his pigs. He knows the pigs require a certain amount of food F_1 and F_2 per day. Table 3 presents estimates of the units of F_1 and F_2 available, per gram of P_1, P_2 and P_3.

Also, each pig should have approximately at least 54 units of F_1 and approximately at least 60 units of F_2, per day. The costs of P_1, P_2 and P_3 vary slightly from day to day but the average costs are: (1) 8¢ per gram of P_1; (2) P_2 is 9¢ per gram; and (3) 10¢/gram for P_3.

The farmer wants to know how many grams of P_1, P_2 and P_3 he should mix together each day, so his pigs will get the approximate minimum, to minimize his costs. Since all the numbers are uncertain we substitute triangular fuzzy numbers, whose peak is at the given value, for all the parameters to give a FFLP. The problem becomes

$$\min \quad \overline{Z} = (7/8/9)\overline{X}_1 + (8/9/10)\overline{X}_2 + (9/10/11)\overline{X}_3 \tag{44}$$

$$\text{s.t.} \quad (2/2.5/3)\overline{X}_1 + (4/4.5/5)\overline{X}_2 + (4.5/5/5.5)\overline{X}_3 \geqslant (50/54/58), \tag{45}$$

$$(4.5/5/5.5)\overline{X}_1 + (2.5/3/3.5)\overline{X}_2 + (9/10/11)\overline{X}_3 \geqslant (56/60/64), \tag{46}$$

$$\overline{X}_1, \overline{X}_2, \overline{X}_3 \geqslant 0, \tag{47}$$

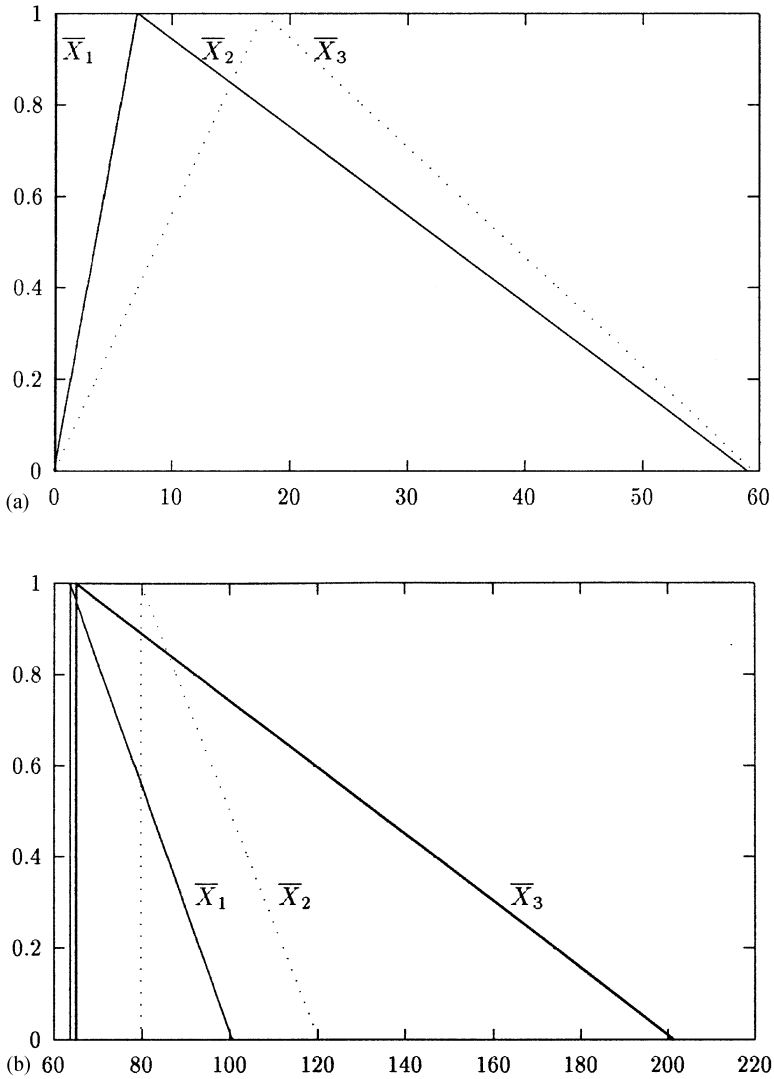


Fig. 5. Shape of the optimal $\overline{X}_1, \overline{X}_2, \overline{X}_3$ for Application 1. (a) $\overline{X}_1$, $\overline{X}_2$ and $\overline{X}_3$ obtained with Kerre's inequality ($c_1 = 100$, $c_2 = 200$, $c_3 = 400$). (b) $\overline{X}_1$, $\overline{X}_2$ and $\overline{X}_3$ obtained with Chen's inequality ($c_1 = 150$, $c_2 = 100$, $c_3 = 300$).

where $\overline{X}_i$ is the amount of P_i, in grams, to use per day, for each pig.

Changing to a multi-objective fuzzy linear program we obtain

$$(\inf z_2, \inf A_2, \inf A_1'), \tag{48}$$

subject to $\overline{X}$ in $\mathscr{F}$. The ideal point $(0,0,0)$ is not in $\mathscr{R}$ because $z_2 = 0$ and $\overline{Z} \geqslant 0$ implies $A_1 = 0$ so that $A_1' = b_3 > 0$. We define the fuzzy goals $\overline{G}_i$ as shown in Figs. 7–9. As before $G(z_2, A_2, A_1') = \overline{G}_1(z_2)\overline{G}_2(A_2)\overline{G}_3(A_1')$ so the FFP is $\sup G(z_2, A_2, A_1')$, $\overline{X} \in \mathscr{F}$.

There are a number of other changes to be made, in the discussion in Section 2, to handle the min problem. We will briefly summarize these changes. If $\overline{X}_a$ and $\overline{X}_b$ are in $\mathscr{F}$, we say $\overline{X}_a$ dominates $\overline{X}_b$ if $z_{2a} \leqslant z_{2b}$, $A_{2a} \leqslant A_{2b}$, $A_{1a}' \leqslant A_{1b}'$ with one inequality a strict inequality. We define $\overline{UD}$, $\mathscr{R}$, $\overline{\mathscr{R}}$ as before. For two vectors v, w in R^3 we say v dominates w if $v_i \leqslant w_i$ all i

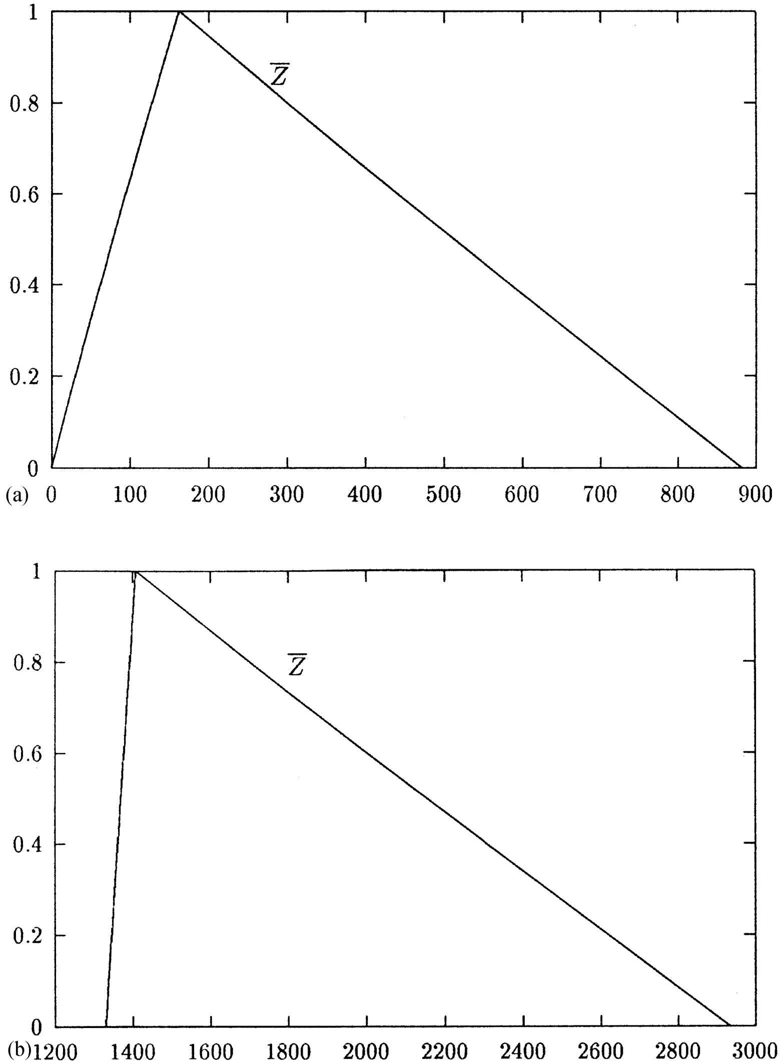


Fig. 6. Shape of $\overline{Z}$ for the optimal $\overline{X}_i$ ($i=1,2,3$) of Application 1. (a) $\overline{Z}$ obtained with Kerre's inequality ($c_1 = 100,\ c_2 = 200,\ c_3 = 400$). (b) $\overline{Z}$ obtained with Chen's inequality ($c_1 = 150,\ c_2 = 100,\ c_3 = 300$).

with at least one inequality a strict inequality. $UD(\overline{D})$ is defined as before and $UD(\overline{\mathscr{R}})$ is non-empty (Theorem 1) using $w = \inf\{v \mid v \in \eta\}$ in the proof. $\mathscr{C}$ is defined as before and Theorem 4 is true with changing Ω to be $\{w \mid w \leqslant c\}$ in the proof. Finally, Theorem 5 is true, simply change $e_i < e_i^*$ some i to $e_i > e_i^*$ some i in the proof.

We again used the solution to the crisp linear program to randomly generate the initial values of the $\overline{X}_i$ in the evolutionary algorithm. The crisp solution is $x_1 = 0$ (for P_1), $x_2 = 8$ (for P_2) and $x_3 = 3.6$ (for P_3) with minimum $z = 108$ (in cents).

The results are given in Table 4.

It appears that for this min problem Kerre's method gives slightly better results. We wanted to use the same c_i values for both methods, but in the first row of results in Table 4 a value of $c_3 = 900$ in Chen's method produced $G(z_2, A_2, A_1') = 0$, so we had to increase the value of c_3 until G was not zero. This explains the different c_3 values for the two methods.

Table 4
Results for Application 2 using different values for c_i $(i = 1, 2, 3)$

	Chen's inequality	Kerre's inequality
b'	1111	951
$c_1/c_2/c_3$	400/400/1050	400/400/900
Best fitness	0.069	0.083
z_2	$\overline{G}_1(209.41) = 0.529$	$\overline{G}_1(173.45) = 0.610$
A_2	$\overline{G}_2(32.34) = 0.927$	$\overline{G}_2(9.93) = 0.978$
A_1	$\overline{G}_3(108.40) = 0.141$	$\overline{G}_3(89.87) = 0.139$
$c_1/c_2/c_3$	300/450/1050	300/450/900
Best fitness	0.038	0.070
z_2	$\overline{G}_1(150.34) = 0.549$	$\overline{G}_1(119.74) = 0.641$
A_2	$\overline{G}_2(197.73) = 0.604$	$\overline{G}_2(6.91) = 0.986$
A_1	$\overline{G}_3(77.82) = 0.117$	$\overline{G}_3(62.05) = 0.111$
$c_1/c_2/c_3$	450/300/1050	450/300/900
Best fitness	0.077	0.088
z_2	$\overline{G}_1(246.40) = 0.507$	$\overline{G}_1(192.61) = 0.615$
A_2	$\overline{G}_2(13.52) = 0.960$	$\overline{G}_2(10.95) = 0.967$
A_1	$\overline{G}_3(127.51) = 0.157$	$\overline{G}_3(99.80) = 0.149$

Table 3
Approximate units of food F_j and product P_i

Product	Food	
	F_1	F_2
P_1	2.5	5
P_2	4.5	3
P_3	5	10

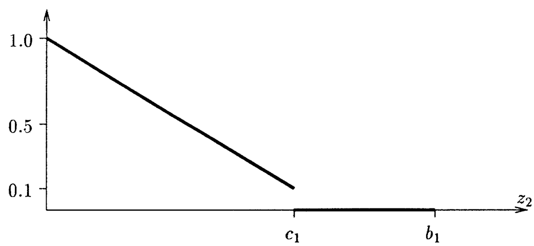


Fig. 7. Fuzzy goal $\overline{G}_1$ for Application 2.

Fig. 10 displays the "optimal" $\overline{X}_i$, for both methods, for one value of c. It is interesting to note that $\overline{X}_1$ and $\overline{X}_3$ are reversed between Figs. 10(a) and (b).

Fig. 11 presents the value of the objective function $\overline{Z}$ for the values of the $\overline{X}_i$ shown in Fig. 10. Again,

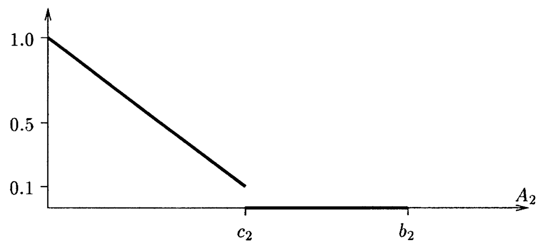


Fig. 8. Fuzzy goal $\overline{G}_2$ for Application 2.

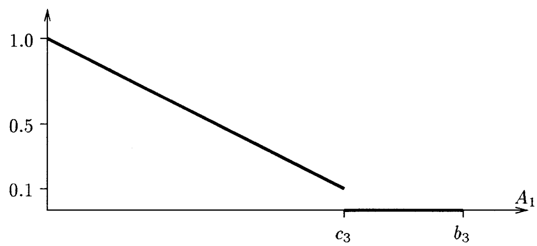
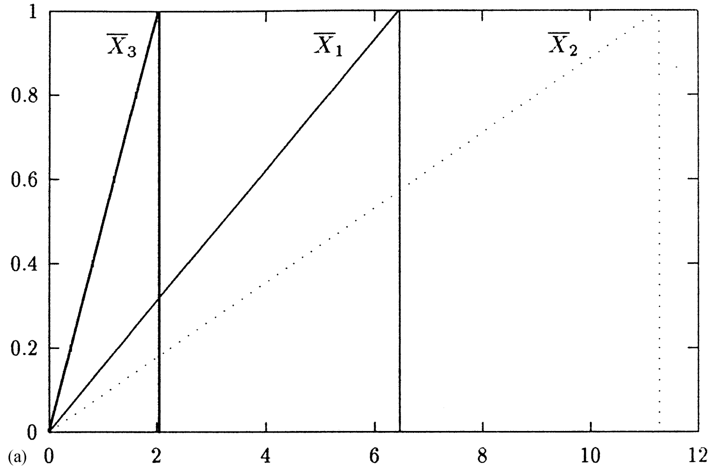


Fig. 9. Fuzzy goal $\overline{G}_3$ for Application 2.

$\overline{Z}$ is a triangular-shaped fuzzy number very close to being an exact triangle.

6. Summary and conclusions

In this paper we discussed a solution to the fully fuzzified linear programming problem (FFLP) where



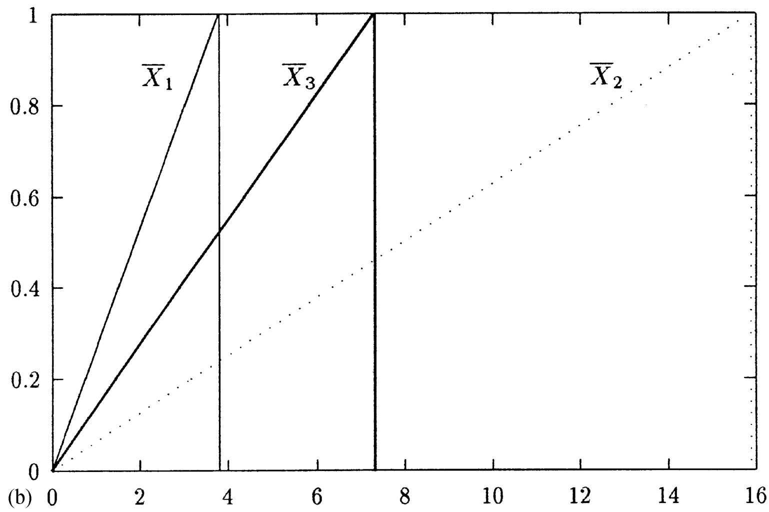


Fig. 10. Shape of the optimal $\overline{X}_1, \overline{X}_2, \overline{X}_3$ for Application 2. (a) $\overline{X}_1, \overline{X}_2, \overline{X}_3$ obtained with Kerre's inequality ($c_1 = 400$, $c_2 = 400$, $c_3 = 900$). (b) $\overline{X}_1, \overline{X}_2, \overline{X}_3$ obtained with Chen's inequality ($c_1 = 450$, $c_2 = 300$, $c_3 = 1050$).

all the parameters and the variables are triangular fuzzy numbers. We first changed the problem of $\max(\overline{Z})$, the fuzzy number value of the objective function, into a multiple objective problem in Section 2. The method of fuzzy flexible programming, discussed in Section 2.2, was selected to explore the undominated set for the FFLP. We also discussed methods of evaluating fuzzy inequalities in Section 3 and picked two procedures, that insured that $\overline{Z}$ was bounded, to be used in the two examples.

Searching for an undominated solution to the FFLP is sufficiently complex that we employed a directed search technique called an evolutionary algorithm, which was discussed in Section 4. We applied the evolutionary algorithm to two classical fuzzified linear programs to show that it can produce good approximate solutions.

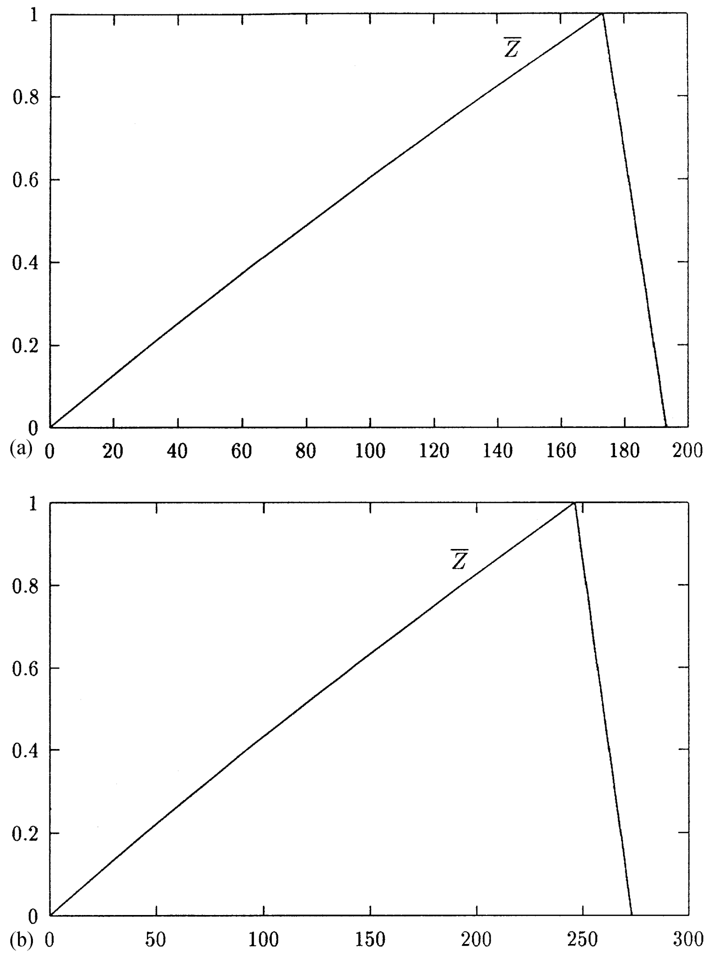


Fig. 11. Shape of $\overline{Z}$ for the optimal $\overline{X}_i$ ($i = 1, 2, 3$) of Application 2. (a) $\overline{Z}$ obtained with Kerre's inequality ($c_1 = 400,\ c_2 = 400,\ c_3 = 900$). (b) $\overline{Z}$ obtained with Chen's inequality ($c_1 = 450,\ c_2 = 300,\ c_3 = 1050$).

Our methods may be extended in a couple of directions. First, some or all of the parameters and variables can be changed to be trapezoidal fuzzy numbers (fuzzy intervals). Originally, we wanted the variables $\overline{X}_i$ to be triangular-shaped fuzzy numbers but ran into the problem of showing that $\overline{Z}$ was bounded. So we restricted the variables to be triangular fuzzy numbers and it was much easier to obtain $\overline{Z}$ bounded.

One can use other methods of evaluating fuzzy inequalities as long as $\overline{Z}$ is bounded. If you cannot show that $\overline{Z}$ is bounded, then you can add constraints to the FFLP such as the $\overline{X}_i$ must be in the interval $[0, M_i]$ for some $M_i > 0$. This will make $\overline{Z}$ bounded. We did not choose to add these extra constraints.

We had assumed that all the parameters were non-negative triangular fuzzy numbers. If you allow some of the parameters to be negative you may have a problem showing $\overline{Z}$ bounded. If $\overline{Z}$ is not bounded you may add those extra constraints mentioned above to get it bounded.

Based on our success in employing an evolutionary algorithm to solve the FFLP we plan to apply it to solve other fuzzy problems such as non-linear fuzzy regression.

Appendix A

In this appendix we compute the b_i', estimates of the b_i, to be used in the fuzzy goals defined in Section 2.2, if we are using the first method of evaluating fuzzy inequalities discussed in Section 3.1. This also gives a second argument that $\overline{Z}$ is bounded.

We first argue for $\overline{X}$ to be feasible there are $M_i > 0$ so that each $\overline{X}_i$ must be in the interval $[0, M_i]$. What this means is that if some $\overline{X}_i$ is not in $[0, M_i]$, then $\overline{X}$ is not feasible.

We will show how to find M_1 for $\overline{X}_1$. Determining the other M_i is similar and therefore omitted. Let $\overline{X}_1 = (x_{11}/x_{12}/x_{13})$ a triangular fuzzy number. We consider the extreme case where $\overline{X}_2 = \cdots = \overline{X}_n = 0$ and $x_{11} = x_{12} = 0 < x_{13}$ in order to isolate x_{13}. Let $\overline{A}_{i1} = (a_{i11}/a_{i12}/a_{i13})$. Then $\overline{E}_i = \overline{A}_{i1}\overline{X}_1$. $\overline{E}_i$ is a triangular-shaped fuzzy number partially specified by $(e_{i1}/e_{i2}/e_{i3})$. In this case we have $e_{i1} = e_{i2} = 0$ and $e_{i3} = a_{i13}x_{13}$. Fig. 12 shows the graph of $\overline{E}_i$ and $\overline{B}_i$. The graph of $\overline{E}_i$ is concave up since $\overline{A}_{i1}$ and $\overline{X}_1$ are triangular fuzzy numbers. In Fig. 12 the a_i are the areas of the regions that contain the a_i. If we set $\overline{Q} = \overline{\max}(\overline{E}_i, \overline{B}_i)$, then $d(\overline{E}_i, \overline{Q}) = a_1 + a_2$ and $d(\overline{B}_i, \overline{Q}) = a_3$. The bold (heavy) line in Fig. 12 is $\overline{Q}$.

Now consider having x_{13} grow without bound. Then e_{i3} and a_3 will also grow without bound. The values of a_1 and a_2 will vary but $a_1 + a_2$ is bounded. So, eventually we get $a_1 + a_2 < a_3$ and $\overline{E}_i \leqslant \overline{B}_i$ is false. Let M_{i1} be a (the smallest) value of x_{13} so that if $x_{13} > M_{i1}$ we get $a_1 + a_2 < a_3$. Define M_1 to be the minimum of the M_{i1}. Then if $\overline{X}_1$ is not in $[0, M_1]$ $\overline{X}$ is not feasible. Similarly we get the M_i, $2 \leqslant i \leqslant n$.

In the applications in Section 5 we need numerical values for the M_i. We wrote a computer program to compute these values. For a given value of x_{13} we found the intersection points of $\overline{E}_i$ and $\overline{B}_i$ in Fig. 12 and then used numerical integration to determine the a_i. Then we increased x_{13} until we could determine M_{i1}. In this way we obtained the M_i for both application problems in Section 5.

Next we consider $\overline{Z} = \sum_{i=1}^{n} \overline{C}_i \overline{X}_i$ with $\overline{C}_i = (c_{i1}/c_{i2}/c_{i3})$ and $\overline{Z}$ a triangular-shaped fuzzy number partially given by $(z_1/z_2/z_3)$. We see that

$$z_3 = \sum_{i=1}^{n} c_{i3}x_{i3} \leqslant M \tag{49}$$

with

$$M = \sum_{i=1}^{n} c_{i3}M_i. \tag{50}$$

Hence, $\overline{Z}$ is in $[0, M]$. It follows that $b_1' = M$ when $z_2 = z_3 = M$, $b_2' = M$ if $z_1 = z_2 = 0$, $z_3 = M$, and $b_3' = M$ when $z_1 = 0$, $z_2 = z_3 = M$.

In this way we make the FFP method of exploring the undominated set operational since surely $b_i < b_i'$, for $i = 1, 2, 3$ and we may now choose the c_i in $[0, b_i']$, $i = 1, 2, 3$.

For the product mix problem in Section 5 we found $b_1' = b_2' = b_3' = 2637$. In the diet problem $b_1' = b_2' = b_3' = 951$.

Appendix B

In this appendix we show how to obtain the b_i' so that $b_i' > b_i$ when using the third method of evaluating fuzzy inequalities described in Section 3.3. In doing this we also show that $\overline{Z}$ is bounded.

We will employ the same methods used in Appendix A. So set $\overline{X}_2 = \cdots = \overline{X}_n = 0$ and $x_{11} = x_{12} = 0 < x_{13}$. We show how to get the M_{i1} and then M_1 is the minimum of the M_{i1}. For notational convenience, let $R_e = \mu_R(\overline{E}_i)$, $L_e = \mu_L(\overline{E}_i)$, $R_b = \mu_R(\overline{B}_i)$ and $L_b = \mu_L(\overline{B}_i)$. In this case $L_e = 1$. Fig. 13 shows the graph of $\overline{E}_i = \overline{A}_{i1}\overline{X}_1$ with fuzzy max, or $\mu_{\max}(x)$. Recall that $e_{i3} = a_{i13}x_{13}$ and $e_{i1} = e_{i2} = 0$. So we have $\overline{E}_i \leqslant \overline{B}_i$ if $R_e \leqslant R_b + (1 - L_b)$.

In Fig. 13 line H is through $(0, 1)$ and $(s, 0.5)$. The point on the graph of $\overline{E}_i$ where the membership value is 0.5 is $(s, 0.5)$ with $s = (a_{i12} + a_{i13})x_{13}/4$. Let γ be the y-coordinate of the intersection of the line H and the line through $(0, 0)$ and $(e_{i3}, 1)$, or the fuzzy max line. We find that

$$\gamma = \frac{a_{i12} + a_{i13}}{a_{i12} + 3a_{i13}}. \tag{51}$$

Notice that γ is independent of x_{13}. Obviously, from Fig. 13 we have $R_e > \gamma$.

We will assume that the $\overline{B}_i$ are symmetric. This is true in the two applications in Section 5. Let $\tau_i = b_{i3} - b_{i2} = b_{i2} - b_{i1}$. From Fig. 4 we find that

$$R_b = \frac{b_{i3}}{e_{i3} + \tau_i} \tag{52}$$

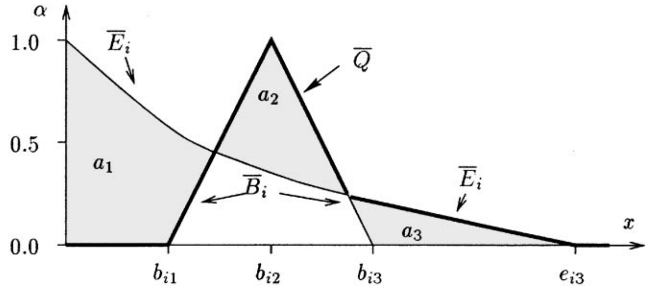


Fig. 12. Graphical description of the estimation of the b_i', for the first method.

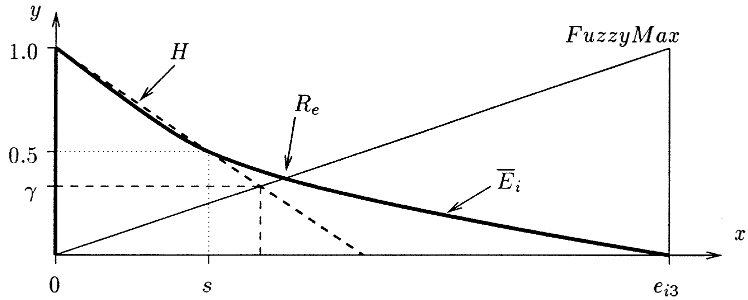


Fig. 13. Graphical description of the estimation of the b_i', for the third method.

and

$$L_b = \frac{e_{i3} - b_{i1}}{e_{i3} + \tau_i}. \tag{53}$$

We now set γ equal to $R_b + (1 - L_b)$ and solve for x_{13}. That is

$$R_e > \gamma = R_b + (1 - L_b) \tag{54}$$

or

$$R_e > \gamma = \frac{b_{i3} + b_{i2}}{e_{i3} + \tau_i}. \tag{55}$$

We now solve this last equation for x_{13} after substituting $a_{i13}x_{13}$ for e_{i3} and the expression in Eq. (51) for γ and obtain

$$x_{13}^* = 2\left[\frac{b_{i2}}{a_{i13}} + \frac{b_{i2} + b_{i3}}{a_{i12} + a_{i13}}\right]. \tag{56}$$

If $x_{13} > x_{13}^*$ then $R_e > R_b + (1 - L_b)$ and $\overline{E}_i \leqslant \overline{B}_i$ is false. Therefore, we set $M_{i1} = x_{13}^*$ and we do this for each constraint.

In a similar way we obtain the M_i for $2 \leqslant i \leqslant n$. Following Appendix A we get M so that $\overline{Z}$ is in $[0, M]$ and $\overline{Z}$ is bounded. We define the b_i' as in Appendix A.

For the product mix problem these values are $b_1' = b_2' = b_3' = 2936$. In the diet problem we calculated $b_1' = b_2' = b_3' = 1111$.

References

[1] T. Bäck, Evolutionary Algorithms in Theory and Practice: Evolutionary Strategies, Evolutionary Programming, Genetic Algorithms, Oxford University Press, New York, 1996.

[2] J.J. Buckley, Fuzzy programming and the Pareto optimal set, Fuzzy Sets and Systems 10 (1983) 57–63.

[3] J.J. Buckley, A reply to 'a note on fuzzy programming and Pareto optimal set', Fuzzy Sets and Systems 14 (1984) 81–82.

[4] J.J. Buckley, Y. Hayashi, Applications of fuzzy chaos to fuzzy simulation, Fuzzy Sets and Systems, to appear.

[5] S.H. Chen, Ranking fuzzy numbers with maximizing set and minimizing set, Fuzzy Sets and Systems 17 (1985) 113–129.

[6] S.J. Chen, C.-L. Hwang, Fuzzy Multiple Attribute Decision Making: Methods and Applications, Springer, New York, 1992.

[7] E.J. Elton, M.J. Gruber, Modern Portfolio Theory and Investment Analysis, 3rd ed., Wiley, New York, 1987.

[8] A.M. Geoffrion, Proper efficiency and theory of vector maximization, J. Math. Anal. Appl. 22 (1968) 618–630.

[9] C.-L. Hwang, A.S.M. Masud, Multiple Objective Decision Making: Methods and Applications, Springer, Heidelberg, 1979.

[10] J.L. Kelley, General Topology, Van Nostrand, New York, 1964.

[11] E.E. Kerre, The use of fuzzy set theory in electrocardiological diagnostics, in: M.M. Gupta, E. Sanchez (Eds.), Approximate Reasoning in Decision Analysis, North-Holland, Amsterdam, 1982, pp. 277–282.

[12] G.J. Klir, B. Yuan, Fuzzy Sets and Fuzzy Logic, Prentice-Hall, Upper Saddle River, NJ, 1995.

[13] Y.-J. Lai, C.-L. Hwang, Fuzzy Mathematical Programming, Lecture Notes in Economics and Mathematical Systems, vol. 394, Springer, Berlin, 1992.

[14] F.A. Lootsma, Optimization with multiple objectives, in: M. Iri, K. Tanabe (Eds.), Mathematical Programming, Kluwer Academic Publishers, Dordrecht, 1989, pp. 333–364.

[15] M.K. Luhandjula, Fuzzy optimization: an appraisal, Fuzzy Sets and Systems 30 (1989) 257–282.

[16] M. Zeleny, Multiple Criteria Decision Making, McGraw Hill, New York, 1982.