ELSEVIER

Fuzzy Sets and Systems 117 (2001) 81–93

FUZZY sets and systems

www.elsevier.com/locate/fss

Using fuzzy bases to resolve nonlinear programming problems

P.Z. Wang [a,1], Ralf Östermark [b,*], Rajan Alex [c], S.H. Tan [d]

[a] AEI, West Texas A&M University, BOX 60248, Canyon TX 79016, USA

[b] Department of Business Administration, Åbo Akademi University, Henriksgatan 7, FIN-20500 Åbo, Finland

[c] Department of Mathematics, West Texas A&M University, Canyon TX 79016, USA

[d] Beijing Aptronix Fuzzy Techn. Co. Ltd, 19-1 ZHONG CHENG Building, Heidian Road, Beijing 100080, China

Received October 1997; received in revised form June 1998

Abstract

In this paper we introduce the concept of fuzzy bases and its usefulness in solving optimization problems with a nonlinear objective function and linear constraints. We investigate the properties of fuzzy bases and operationalize them in fuzzy interpolation. The NLP can be relaxed into a bilinear program with a simple structure using fuzzy interpolation, irrespective of whether the objective function is convex or not. If the objective function is convex, we prove that the optimization problem can be transformed into an ordinary LP using fuzzy (linear) bases. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Fuzzy bases; Fuzzy interpolation; Transformation of NLPs

1. Introduction

Important research on fuzzy mathematical programming has evolved since the early 1970s following the path-breaking work by Bellman and Zadeh [2] and Zimmermann [11]. Significant contributions have been made especially concerning various aspects of fuzzy multiobjective linear programming (FMOLP) problems (cf., e.g., [6,1] on the stability of multiobjective NLPs, and [5] on temporal interdependence in FMOLP-problems), modelling of nonlinear systems [7,4] and solution methods for fuzzy programming [3]. However, solutions to NLPs by adherence to fuzzy set theory are presently scarce (cf. [8–10]). In this paper, we present some new results that may be used to effectively simplify the solution of NLP-problems.

The paper is organized as follows. In Section 2 we present the general theory of fuzzy bases. In Section 3, we specialize the theory to fuzzy linear bases, to cope with nonconvex optimization problems. In Section 4, we operationalize the concept of fuzzy bases in fuzzy interpolation, showing that any continuous differentiable function on $\mathbb{R}$ can be approximated by fuzzy (linear) interpolation to accuracy ε. In Section 5, we show how to transform general (nonconvex) nonlinear programming (NLP) problems to a mixed integer bilinear form using fuzzy interpolation. In Section 6 we show that, if the objective function is convex, then the NLP can be relaxed to an ordinary LP using fuzzy bases. In Section 7, we suggest simplifying cutting methods for NLPs. Section 8 concludes.

* Corresponding author. Tel.: +358 2 2154669; fax: +358 2 2154806.

E-mail address: ralf.ostermark@abo.fi (R. Östermark).

[1] On leave from Beijing Normal University, China.

0165-0114/01/$ - see front matter
PII: S0165-0114(98)00273-5

2. Fuzzy bases

Fuzzy set theory provides a general framework to describe and analyze real-world phenomena from many areas of application. An ordinary (crisp) system can be represented by the mapping

$$f: X \to Y, \tag{2.1}$$

where X and Y are usually subsets of one-dimensional or high-dimensional real-valued sets. In fuzzy set theory, again, the universe of discourse can be partitioned into a finite group of fuzzy subsets of X,

$$X^* = \{B_0, \ldots, B_K\}, \tag{2.2}$$

constituting a new universe of discourse called the *fuzzy granulation* of X. The fuzzification of the mapping f into $f^*: X^* \to Y$ is the core issue in fuzzy interpolation, as discussed subsequently (Section 4).

Definition 2.1. Let $B = \{B_k\}_{(k=0,\ldots,K)}$ be a fuzzy granulation of $X = [a, b]$. B is called a *fuzzy basis* of X if

(1) B is a fuzzy division, i.e.,

$$\sum_{k=0}^{K} B_k(x) \equiv 1. \tag{2.3}$$

(2) For each $k \in \{0, 1, \ldots, K\}$, B_k has a dromedary membership function, i.e., for any $\lambda \in [0, 1]$, the λ-cut of B_k is an interval on $X = [a, b]$.

(3) For each $k \in \{0, 1, \ldots, K\}, B_k$ is unimodular, i.e., it has one and only one peak point e_k such that

$$B_k(e_k) = \max_x B_k(x). \tag{2.4}$$

Definition 2.2. We call $B = \{B_k\}_{(k=0,\ldots,K)}$ a *fuzzy regular basis* of X if it is a fuzzy basis and for each $k \in \{0, 1, \ldots, K\}$ we have that

$$B_k(e_k) = 1. \tag{2.5}$$

Definition 2.3. We call $B = \{B_k\}_{(k=0,\ldots,K)}$ a *fuzzy binary basis* of X if it is a fuzzy regular basis and for any $x \in X$ there are at most two adjacent subsets of the basis taking non-zero membership degrees:

$$\exists k \in \{1, \ldots, K\}:$$
$$B_{k-1}(x) \geqslant 0,\ B_k(x) \geqslant 0,\ B_j(x) = 0,\ j \notin \{k, k-1\}. \tag{2.6}$$

Definition 2.4. Let

$$a = e_0 < e_1 < \cdots < e_K = b \tag{2.7}$$

be a group of division points on an interval $[a, b]$. The fuzzy binary basis defined below is called the *simplest fuzzy basis* or the fuzzy linear binary basis with respect to Eq. (2.7) on $[a, b]$:

$$B_0(x) = \begin{cases} 1, & x = a, \\ \dfrac{e_1 - x}{e_1 - a}, & a \leqslant x \leqslant e_1, \\ 0, & x > e_1; \end{cases}$$

$$B_k(x) = \begin{cases} 0, & x < e_{k-1}, \\ \dfrac{x - e_{k-1}}{e_k - e_{k-1}}, & e_{k-1} \leqslant x < e_k, \\ 1, & x = e_k, \\ \dfrac{e_{k+1} - x}{e_{k+1} - e_k}, & e_k < x \leqslant e_{k+1}, \\ 0, & x > e_{k+1}; \end{cases} \tag{2.8}$$

$$(k = 1, \ldots, K-1)$$

$$B_K(x) = \begin{cases} 0, & x < e_{K-1}, \\ \dfrac{x - e_{K-1}}{b - e_{K-1}}, & e_{K-1} \leqslant x < b, \\ 1, & x = b. \end{cases}$$

Note that the simplest fuzzy basis is completely determined by the division points. The simplest fuzzy basis defined in (2.8) is denoted by $B(ae_1, \ldots, e_{K-1}b)$. As shown in Fig. 1, the simplest fuzzy basis consists of a group of triangular fuzzy numbers. The membership

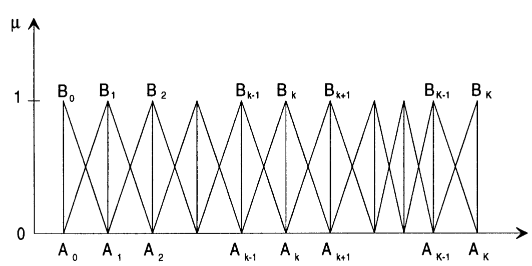

Fig. 1. The simplest fuzzy basis.

functions of them are the broken lines

$$(B_0A_1\dots A_K),(A_0B_1A_2\dots A_K),$$

$$(A_0\dots A_{k-1}B_kA_{k+1}\dots A_K),\dots,(A_0\dots A_{K-1}B_K).$$

Definition 2.5. Let $B=\{B_0,\dots,B_K\}$ be a fuzzy basis of $X=[a,b]$. We call it *bilinearly satisfying* if for any $x\in[a,b]$ there exist

$$\{\delta_k\}_{(k=1,\dots,K)},\quad \delta_k=0 \text{ or } 1 \tag{2.9}$$

$$\{\gamma_k\}_{(k=1,\dots,K)},\quad \gamma_k=0 \text{ or } 1 \tag{2.10}$$

such that

$$\gamma_k(B_{k-1}(x)+B_k(x))=\delta_k\quad (k=1,\dots,K), \tag{2.11}$$

where

$$\sum_{k=1}^{K}\delta_k\geqslant 1 \tag{2.12}$$

and

$$\sum_{k=1}^{K}\gamma_k\geqslant 1. \tag{2.13}$$

We can now operationalize the previous definition as follows

Theorem 2.1. *A fuzzy basis $B=\{B_0,\dots,B_K\}$ is a fuzzy binary basis of X if and only if it is bilinearly satisficing.*

Proof. From Eq. (2.11) we see that there are only three possible cases for $\{\gamma_k\}$ and $\{\delta_k\}$ in Eqs. (2.9) and (2.10):

(1) $\gamma_k=0,\ \delta_k=0$, constraint (2.11) is inactive on $\{B_k(x)\}$;

(2) $\gamma_k=1,\ \delta_k=0$, $B_{k-1}(x)=B_k(x)=0$ by Eq. (2.11);

(3) $\gamma_k=\delta_k=1$, $B_{k-1}(x)+B_k(x)=1$ by Eq. (2.11).

Suppose that $\{B_k(x)\}_{k=0,\dots,K}$ is a fuzzy basis which is bilinearly satisficing. Then, for any $x\in[a,b]$, there is $\{\gamma_k\}_{k=1,\dots,K}$ and $\{\delta_k\}_{k=1,\dots,K}$ satisfying Eqs. (2.9)–(2.13). From Eq. (2.3) there is at least a $k\in\{1,\dots,K\}$ such that $\delta_k=1$. Now, this only occurs in case 3. Hence $\gamma_k=\delta_k=1$, implying that $B_{k-1}(x)+B_k(x)=1$. Since B is a fuzzy basis, by Eq. (2.3), we have that $B_j(x)=0$ $(j\neq k,k-1)$, so that Eq. (2.6) holds. Therefore B is a fuzzy binary basis (Fig. 2).

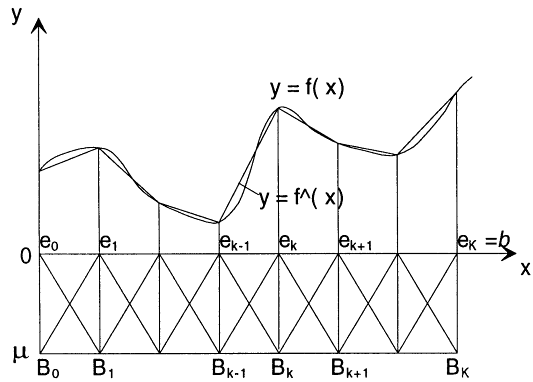

Fig. 2. A fuzzy interpolation.

Conversely, suppose that $B=\{B_k(x)\}_{k=0,\dots,K}$ is a fuzzy binary basis. According to Definition 2.3, for any $x\in[a,b]$ there is a $k\in\{1,\dots,K\}$ such that Eq. (2.6) is true. Then we can set

$$\gamma_1=\cdots=\gamma_{k-1}=\gamma_{k+1}=\cdots=\gamma_K=0;\ \gamma_k=1,$$

$$\delta_1=\cdots=\delta_{k-1}=\delta_{k+1}=\cdots=\delta_K=0;\ \delta_k=1,$$

where $\sum_{k=1}^{K}\gamma_k=\sum_{k=1}^{K}\delta_k=1$, and $\gamma_i=\delta_i=0$ when $i\neq k$. We have that

$$\gamma_i(B_{i-1}(x)+B_i(x))=\delta_i\quad (i\neq k).$$

From Eq. (2.6) we have that $B_{i-1}(x)=B_i(x)=0$ when $i\neq k$, so that

$$\gamma_k(B_{k-1}(x)+B_k(x))=B_{k-1}(x)+B_k(x)=\sum_{i=1}^{K}(B_{i-1}(x)+B_i(x)).$$

Since B is a fuzzy basis, we have that

$$\gamma_k(B_{k-1}(x)+B_k(x))=1=\delta_k,$$

so that (2.11) is true. Since Eqs. (2.9), (2.10), (2.12) and (2.13) are satisfied, B is bilinearly satisficing. □

Remark 2.1. Theorem 2.1 has important meaning when we use fuzzy bases in solving (non-convex) non-linear programming problems in the mixed integer bilinear form.

Let $B=\{B_k\}_{(k=0,\dots,K)}$ be a fuzzy basis on $X=[a,b]$ and consider the mapping

$$\mu:[a,b]\to[0,1]^{K+1}$$
$$\mu(x)=(\mu_0,\dots,\mu_K), \tag{2.14}$$

where

$$\mu_k=B_k(x)\quad(k=0,\dots,K).$$

Let

$$L=\left\{\mu=(\mu_0,\dots,\mu_K)\mid \mu_k\geqslant 0\ (k=0,\dots,K),\ \sum_{k=0}^{K}\mu_k=1\right\}. \tag{2.15}$$

L represents the image of all fuzzy bases.

Let

$$I=\{\mu=(\mu_0,\dots,\mu_K)\mid \mu\in L,\ \mu \text{ satisfies the bilinear conditions (2.9)–(2.13)}\}. \tag{2.16}$$

Eq. (2.16) implies that μ is bilinearly satisficing, i.e.,

$$\gamma_k(\mu_{k-1}+\mu_k)=\delta_k\quad(k=1,\dots,K). \tag{2.17}$$

We then have the following.

Theorem 2.2. *Suppose that $B=\{B_k\}_{(k=0,\dots,K)}$ is a simplest fuzzy basis with respect to the division points* (2.7). *Then the mapping μ defined in* (2.14) *constitutes a one-to-one correspondence between $[a,b]$ and I.*

Proof. For any x in $[a,b]$, there is one and only one point $\mu=(B_0(x),\dots,B_K(x))$, since B is a simplest fuzzy basis, so that $\mu\in I$. Conversely, given a point $\mu^*=(\mu_0^*,\dots,\mu_K^*)$ in I, there is one and only one point x^* in the interval $[a,b]$ such that $B_k(x^*)=\mu_k{}^*$ $(k=0,\dots,K)$. Indeed, set

$$x^*=e_0\mu_0^*+e_1\mu_1^*+\cdots+e_K\mu_K^*. \tag{2.18}$$

Conditions (2.9)–(2.13) then imply that, $\exists k\in\{1,\dots,K\}$, either
(1) $\mu_k^*=1,\ \mu_j^*=0,\ j\neq k$ or
(2) $\mu_{k-1}^*\neq 0,\ \mu_k^*\neq 0,\ \mu_j^*=0,\ j\notin\{k-1,k\}$.

In case (1) $x^*=e_k$ by Eq. (2.18). By Eq. (2.8) we have that $B_k(x^*)=\mu_k^*$ $(k=0,\dots,K)$. In case (2) $\mu_{k-1}^*+\mu_k^*=1$, i.e., $\mu_{k-1}^*=1-\mu_k^*$. Eq. (2.18) then implies that

$$x^*=e_{k-1}(1-\mu_k^*)+e_k\mu_k^*. \tag{2.19}$$

Hence

$$B_{k-1}(x^*)=\frac{e_k-x^*}{e_k-e_{k-1}}=\frac{e_k-[e_{k-1}(1-\mu_k^*)+e_k\mu_k^*]}{e_k-e_{k-1}}=1-\mu_k^*=\mu_{k-1}^*,$$

$$B_k(x^*)=\frac{x^*-e_{k-1}}{e_k-e_{k-1}}=\frac{[e_{k-1}(1-\mu_k^*)+e_k\mu_k^*]-e_{k-1}}{e_k-e_{k-1}}=\mu_k^*,$$

$$B_j(x^*)=\mu_j^*=0\quad(j\notin\{k-1,k\}).\qquad\square$$

3. Fuzzy linear bases

When the optimization problem is nonconvex, the fuzzy basis concept defined above is insufficient. For this situation, some advanced results are provided in the present section.

Definition 3.1. A fuzzy basis $B=\{B_k\}_{(k=0,\dots,K)}$ on $[a,b]$ is called a (K-stage) fuzzy linear basis if for any k, $y=B_k(x)$ is a piecewise linear function with respect to Eq. (2.7) and for any $x\in[a,b]$ we have that

$$e_0B_0(x)+\cdots+e_KB_K(x)=x. \tag{3.1}$$

Denote the set of all K-stage fuzzy linear bases on $[a,b]$ by $C_K[a,b]$.

Any simplest fuzzy basis is a fuzzy linear basis and there is only one 1-stage fuzzy linear basis on $[a,b]:B(ab)$. Indeed, formula (3.1) implies that

$$aB_0(x)+bB_1(x)=x.$$

But $\{B_0, B_1\}$ is the simplest fuzzy basis $B(ab)$ (cf. Eq. (2.8)),

$$B_0(x) = \frac{b-x}{b-a}, \qquad B_1(x) = \frac{x-a}{b-a}. \tag{3.2}$$

Therefore $B_1(x) = 1 - B_0(x)$. Hence,

$$C_1[a,b] = \{B(ab)\}. \tag{3.3}$$

Assume that $B^{(1)} = \{B_k^{(1)}\}_{(k=0,\dots,K_1)}$ and $B^{(2)} = \{B_k^{(2)}\}_{(k=0,\dots,K_2)}$ are two fuzzy linear bases on $[a,b]$ with respect to the divisions

$$\delta_1: a = e_{10} < e_{11} < \cdots < e_{1K_1} = b$$

and

$$\delta_2: a = e_{20} < e_{21} < \cdots < e_{2K_2} = b,$$

respectively. From δ_1 and δ_2 we obtain the union of divisions $\delta = \delta_1 \cup \delta_2$: $a = e_0 < e_1 < \cdots < e_K = b$. Now, any breakpoint in δ_1 or δ_2 is a breakpoint in δ and, conversely, any breakpoint in δ is a breakpoint in either δ_1 or δ_2.

Let

$${}^{\cup}B^{(i)}(x) = \{{}^{\cup}B_k^{(i)}(x)\}_{(k=0,\dots,K)} \quad (i = 1,2), \tag{3.4}$$

where ${}^{\cup}B_k^{(i)}(x) = B_k^{(i)}(x)$, if $e_k \in \delta_i$ and ${}^{\cup}B_k^{(i)}(x) = 0$, if $e_k \notin \delta_i$. For any convex coefficients $\lambda_1 \geqslant 0, \lambda_2 \geqslant 0$, $\lambda_1 + \lambda_2 = 1$, we set

$$\begin{aligned} B_k(x) &= \lambda_1^{\cup} B_k^{(1)}(x) + \lambda_2^{\cup} B_k^{(2)}(x) \\ &= \sum_{i=1}^{2} \lambda_i^{\cup} B_k^{(i)}(x) \quad (k = 0,\dots,K). \end{aligned} \tag{3.5}$$

Definition 3.2. The new fuzzy basis $B = \{B_k\}_{(k=0,\dots,K)}$ defined in Eq. (3.5) is called the convex combination of $B^{(i)} = \{B_k^{(i)}\}_{(k=0,\dots,K_i)}$ $(i = 1,2)$.

The definition extends directly to any convex combination of n fuzzy linear bases.

Example 3.1. Let $B^{(1)} = B(02)$, which is the simplest fuzzy basis on the partition

$$\delta_1:\ 0 = e_{10} < e_{11} = 2 \tag{3.6}$$

and let $B^{(2)} = B(012)$, which is the simplest fuzzy basis on the partition

$$\delta_2:\ 0 = e_{10} < e_{11} = 1 < e_{12} = 2. \tag{3.7}$$

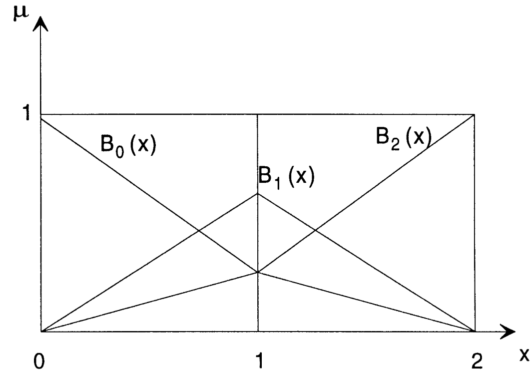

Fig. 3. The (0.4, 0.6)-combination of $B^{(1)}$ and $B^{(2)}$.

As illustrated in Fig. 3, the (0.4, 0.6)-convex combination of $B^{(1)}$ and $B^{(2)}$ is $B = \{B_k(x)\}_{(k=0,1,2)}$, where

$$B_0(x) = \begin{cases} 1 - 0.8x & (0 \leqslant x \leqslant 1), \\ 0.2 - 0.2(x-1) & (1 < x \leqslant 2), \end{cases}$$

$$B_1(x) = \begin{cases} 0.6x & (0 \leqslant x \leqslant 1), \\ 0.6 - 0.6(x-1) & (1 < x \leqslant 2), \end{cases} \tag{3.8}$$

$$B_2(x) = \begin{cases} 0.2x & (0 \leqslant x \leqslant 1), \\ 0.2 + 0.8(x-1) & (1 < x \leqslant 2). \end{cases}$$

Generally, we can prove that $C_2[a,b]$, the set of all two-stage fuzzy linear bases on $[a,b]$ is generated by $B(ab)$ and $B(aeb)$ and has the following general formulae:

$$B_0(x) = \begin{cases} 1 - \left(\dfrac{\lambda_1}{b-a} + \dfrac{\lambda_2}{e-a}\right)(x-a) & \\ \qquad\qquad (a \leqslant x \leqslant e), \\ \dfrac{\lambda_1(b-x)}{b-a} \quad (e < x \leqslant b), \end{cases}$$

$$B_1(x) = \begin{cases} \dfrac{\lambda_2(x-a)}{e-a} & (a \leqslant x \leqslant e), \\ \dfrac{\lambda_2(b-x)}{b-e} & (e < x \leqslant b), \end{cases} \tag{3.9}$$

$$B_2(x) = \begin{cases} \dfrac{\lambda_1(x-a)}{b-a} & (a \leqslant x \leqslant e), \\ \dfrac{\lambda_1(x-a)}{b-a} + \dfrac{\lambda_2(x-e)}{b-e} & (e < x \leqslant b). \end{cases}$$

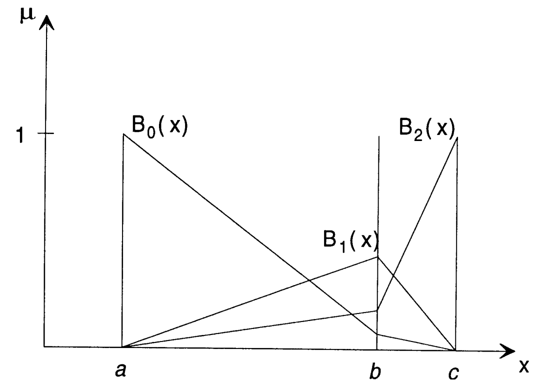

Fig. 4. Two-stage fuzzy linear bases.

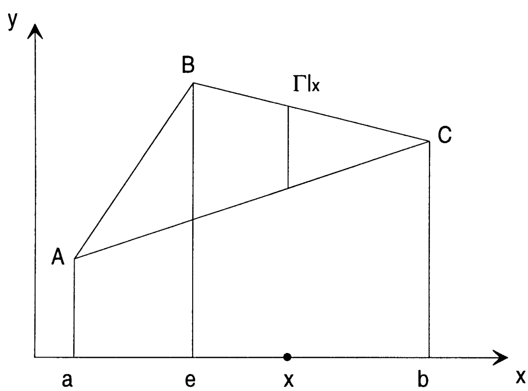

Fig. 5. Sectional cut by fuzzy linear bases.

The main characteristics of this basis are illustrated at $x=e$ (see Fig. 4):

$$B_1(e)=\lambda_2, \qquad B_0(e)+B_2(e)=\lambda_1,$$
$$B_0(e)\colon B_2(e)=\frac{b-e}{e-a} \tag{3.10}$$

As indicated in Fig. 5, for any $x\in[a,b]$, the section of the triangle ΔABC at x can be represented by

$$\Gamma|_x=\{(aB_0(x)+eB_1(x)+bB_2(x),$$
$$y_0B_0(x)+y_1B_1(x)+y_2B_2(x))\mid B\in C_2[a,b]\}.$$

Hence,

$$\Delta ABC=\bigcup\{\Gamma|_x\mid x\in[a,b]\}. \tag{3.11}$$

Denote the class of all fuzzy linear bases on $[a,b]$ by $C[a,b]$. We can then prove the following.

Theorem 3.1. *The class of fuzzy linear bases $C[a,b]$ is closed for the operation of convex combination, i.e., any convex combination of fuzzy linear bases is still a fuzzy linear basis.*

Denote the class of all subsets of $C[a,b]$ that are closed for convex combination by $C^{\#}[a,B]$.

Definition 3.3. Let

$$\sigma[B^{(1)},\ldots,B^{(r)}]$$
$$=\bigcap\{\mathscr{B}\mid B^{(1)},\ldots,B^{(r)}\in\mathscr{B}\in C^{\#}[a,b]\}. \tag{3.12}$$

Then $\sigma[B^{(1)},\ldots,B^{(r)}]$ is called the (rth order) convex closure of the fuzzy linear bases $B^{(1)},\ldots,B^{(r)}$.

$\sigma[B^{(1)},\ldots,B^{(r)}]$ is the smallest set containing $B^{(1)},\ldots,B^{(r)}$, which is closed for the operation of convex combination. In general, we have the following.

Theorem 3.2. *For any positive integer K, we have that*

$$C_K[a,b]=\sigma[\{B(ae_ie_jb)\}\mid 0\leqslant i<j\leqslant K\}], \tag{3.13}$$

where

$$B(ae_0e_jb)=B(ae_ib), \qquad B(ae_ie_Kb)=B(ae_ib),$$
$$B(ae_0e_Kb)=B(ab).$$

For any $\mu=(\mu_0,\ldots,\mu_K)\in L$ (L defined in Eq. (2.15)), there exists a basis $B^*=\{B_k^*(x)\}_{(k=0,\ldots,K)}$ in $C_K[a,b]$ such that

$$\mu_k=B_k^*(a\mu_0+\cdots+e_k\mu_k+\cdots+b\mu_K),$$
$$k=0,\ldots,K. \tag{3.14}$$

From Theorem 3.2, we obtain the following important theorem:

Theorem 3.3. *For any given set of breakpoints*

$$\Phi=\{(e_0,y_0),(e_1,y_1),\ldots,(e_K,y_K)\}, \tag{3.15}$$

the generated convex polygon

$$\Gamma_K = \{(e_0\mu_0 + \cdots + e_K\mu_K,\ y_0\mu_0 + \cdots + y_K\mu_K) \mid \mu \in L\} \quad (3.16)$$

can be represented as a convex combination of K-stage fuzzy linear bases:

$$\Gamma_K = \{(e_0B_0(x) + \cdots + e_KB_K(x),\ y_0B_0(x) + \cdots + y_KB_K(x)) \mid x \in [a,b];\ B \in C_K[a,b]\}. \quad (3.17)$$

In the next theorem we show that there exists an alternative representation of Γ_K utilizing convex combinations of C_h, where h is of arbitrary order. Let k_0 be a fixed index between 1 and K. Let

$$L' = \{\mu = (\mu_0,\ldots,\mu_K) \mid \mu \in L,\ \text{either } \mu_0 + \cdots + \mu_{k_0-1} = 1 \text{ or } \mu_{k_0-1} + \mu_{k_0} = 1 \text{ or } \mu_{k_0} + \cdots + \mu_K = 1\}. \quad (3.18)$$

The convex polygon generated by Φ on L' becomes

$$\Gamma' = \{(e_0\mu_0 + \cdots + e_K\mu_K,\ y_0\mu_0 + \cdots + y_K\mu_K) \mid \mu \in L'\}. \quad (3.19)$$

The polygon Γ' can be represented by means of fuzzy linear bases. Let $[c,d]$ be a sub-interval of $[a,b]$. For $0 \leqslant k_c < k_d \leqslant K$, let $c = e_{k_c},\ d = e_{k_d}$. Denote by $\gamma([c,d])$

$$\{(cB_{k_c}(x) + \cdots + dB_{k_d}(x),\ y_{k_c}B_{k_c}(x) + \cdots + y_{k_d}B_{k_d}(x)) \mid x \in [c,d];\ B \in C_{k_d-k_c}[c,d]\}. \quad (3.20)$$

Then we have the following.

Theorem 3.4.

$$\Gamma' = \gamma[a, e_{k_0-1}] \cup \gamma[e_{k_0-1}, e_{k_0}] \cup \gamma[e_{k_0}, b]. \quad (3.21)$$

Proof. For any point P^* in Γ', we have that

$$P^* = (e_0\mu_0^* + \cdots + e_K\mu_K^*, y_0\mu_0^* + \cdots + y_K\mu_K^*),$$

where $\mu^* = (\mu_0^*,\ldots,\mu_K^*) \in L' \subseteq L$. According to Theorem 3.2, there exists a basis $B^* = \{B_k^*\}_{(k=0,\ldots,K)}$ in $C_K[a,b]$ such that $\mu_k^* = B_k^*(a\mu_0^* + \cdots + e_k\mu_k^* + \cdots + b\mu_K^*)_{(k=0,\ldots,K)}$. But, $\mu^* \in L'$, implying that when $e_{k_0-1} \leqslant x \leqslant e_{k_0}$, we must have

$$B_{k_0-1}^*(x) + B_{k_0}^* = 1; \quad B_k^*(x) = 0\ (k \notin \{k_0-1, k_0\}).$$

This means that B^* must be the simplest basis on $[e_{k_0-1}, e_{k_0}]$: $B^* \in C_1[e_{k_0-1}, e_{k_0}]$. Therefore, the intersection of Γ' and $Y \times [e_{k_0-1}, e_{k_0}]$ must be $\gamma[e_{k_0-1}, e_{k_0}]$. When $x < e_{k_0-1}$, we have $\mu_0 + \cdots + \mu_{k_0-1} = 1$ so that $B_k^*(x) = 0$ $(k \geqslant k_0)$. Hence $\Gamma' \cap Y \times [a, e_{k_0-1}] = \gamma[a, e_{k_0-1}]$. Similarly, $\Gamma' \cap Y \times [e_{k_0}, b] = \gamma[e_{k_0}, b]$. □

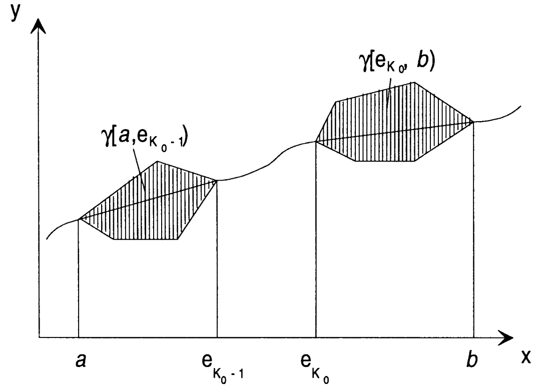

Fig. 6. Representing polygons by fuzzy linear bases (cf. Theorem 3.3).

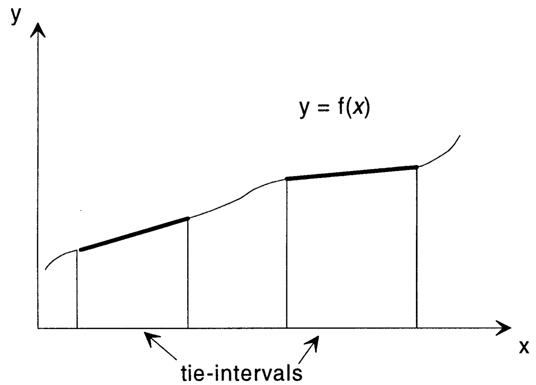

Fig. 7. Tie-intervals with compressed areas of convex combinations.

The meaning of the above theorem is illustrated in Fig. 6. In Fig. 7, the areas of convex combinations of breakpoints have been compressed. The selection of the interval $[e_{k_0-1}, e_{k_0}]$ is the key for the compression. We call $[e_{k_0-1}, e_{k_0}]$ a tie-interval. The theorem simplifies the resolution of mathematical programming problems, as shown below.

Let

$$f^{\#}(x) = \max_y \{(x, y) \in \Gamma'\} \quad (x \in [a, b]), \tag{3.22}$$

which is the top border of Γ'. As stated in Theorem 4.1 below, we can select the tie-intervals such that

$$|f^{\#}(x) - f(x)| < \varepsilon, \quad x \in [a, b]. \tag{3.23}$$

4. Fuzzy interpolation

Definition 4.1 (Wang [7]). Given a mapping $f : [a, b] \to Y$ and a fuzzy basis

$$B = X^* = \{B_k\}_{(k=0,\dots,K)},$$

the mapping

$$\begin{aligned} &f^* : X^* \to Y, \\ &f^*(B_k) = f(e_k)_{(k=0,\dots,K)} \end{aligned} \tag{4.1}$$

is called the fuzzification of the mapping f with respect to the fuzzy basis B. Let

$$f^{\wedge}(x) = f(e_0)B_0(x) + \cdots + f(e_K)B_K(x) \quad (x \in [a, b]). \tag{4.2}$$

$f^{\wedge}$ is called the fuzzy interpolation of the mapping f with respect to the fuzzy basis B (cf. Fig. 2).

Theorem 4.1. *Given a function $y = f(x)$ defined on an interval $[a, b]$ with continuous derivatives, there exists, for any given positive number $\varepsilon > 0$, a fuzzy basis B such that $f^{\wedge}$, the fuzzy interpolation of f based on B, can approximate f to accuracy $\varepsilon > 0$:*

$$|f^{\wedge}(x) - f(x)| \leqslant \varepsilon \quad (x \in [a, b]). \tag{4.3}$$

Proof. Since $y' = f'(x)$ is continuous on $[a, b]$, $f'(x)$ reaches its maximum and minimum on $[a, b]$:

$$M = \max_{a \leqslant x \leqslant b} f'(x), \qquad m = \min_{a \leqslant x \leqslant b} f'(x).$$

Denote $w = \max(|M|, |m|)$. If $w = 0$, then $y \equiv f(a)$. Theorem 4.1 is obviously true. We only consider $w > 0$ and denote δ by ε/w.

Let $e_0 = a$, $y_0 = f(a)$. For any $k \in \{1, 2, \dots, p\}$ where p satisfies the condition $p\delta \leqslant b - a < (p+1)\delta$, set $e_k = a + k\delta$, $y_k = f(a + k\delta)$. If $p\delta < b - a$, then set $e_{p+1} = b$, $y_{p+1} = f(b)$. Define

$$K = \begin{cases} p & \text{if } p\delta = b - a, \\ p + 1 & \text{if } p\delta < b - a. \end{cases}$$

We get the set of $K + 1$ breakpoints $\{(e_k, y_k)\}_{k=0,\dots,K}$.

Let $B = \{B_k(x)\}_{(k=0,\dots,K)}$ be the simplest fuzzy basis with respect to the division $a = e_0 < e_1 < \cdots < e_K = b$ defined in Eq. (2.8). We obtain the fuzzy interpolation of f as follows:

$$\begin{aligned} f^{\wedge}(x) &= f(e_0)B_0(x) + \cdots + f(e_K)B_K(x) \ (x \in [a, b]) \\ &= f(e_{k-1})B_{k-1}(x) + f(e_k)B_k(x) \\ &\qquad (e_{k-1} \leqslant x \leqslant e_k) \quad (k = 1, \dots, K). \end{aligned}$$

From Eq. (2.8) we obtain

$$f^{\wedge}(x) = y_{k-1} + \frac{y_k - y_{k-1}}{e_k - e_{k-1}}(x - e_{k-1}) \qquad (e_{k-1} \leqslant x \leqslant e_k) \quad (k = 1, \dots, K) \tag{4.4}$$

which is the broken line passing through those breakpoints.

We are going to prove that Eq. (4.3) is true. For any $k \in \{1, \dots, K\}$, Eq. (4.4) implies that $f^{\wedge}(x) \in [y_{k-1}, y_k]$ whenever $x \in [e_{k-1}, e_k]$. Therefore

$$|f^{\wedge}(x) - f(x)| \leqslant \max(|f(x) - y_{k-1}|, |f(x) - y_k|).$$

But

$$\begin{aligned} |f(x) - y_{k-1}| &= |f(x) - f(e_{k-1})| = |f'(\theta)(x - e_{k-1})| \\ &\leqslant |f'(\theta)|(e_k - e_{k-1}) \\ &\leqslant w\delta = \varepsilon \quad (e_{k-1} \leqslant \theta \leqslant x \leqslant e_k). \end{aligned}$$

Note: $|f'(\theta)| < w$ by definition and $e_k = a + k\delta$ implies that $\delta = e_k - e_{k-1}$.

Similarly

$$|f(x) - y_k| = |f(x) - f(e_k)| = |f'(\varphi)(x - e_k)|$$
$$\leqslant |f'(\varphi)|(e_k - e_{k-1})$$
$$\leqslant w\delta = \varepsilon \quad (e_{k-1} \leqslant x \leqslant \varphi \leqslant e_k).$$

So Eq. (4.4) holds. □

5. Transforming a general nonlinear program into a bilinear form

Consider the following nonlinear separable programming problem:

Program 1.

$$\text{Maximize} \quad \sum_{j=1}^{n} f_j(x_j) \tag{5.1}$$

$$\text{subject to} \quad \sum_{j=1}^{n} a_{ij}x_j \leqslant b_i \quad (i = 1, \ldots, m), \tag{5.2}$$

$$x_j \geqslant 0 \quad (j = 1, \ldots, n), \tag{5.3}$$

where for any $j = 1, \ldots, n$, $f_j(x)$ has continuous derivatives on the interval $[0, v_j]$. Suppose that, for each $j \in \{1, \ldots, n\}$, $f_j^{\wedge}(x)$ is the fuzzy interpolation of f_j with respect to the simplest (triangular) fuzzy basis $B_j = \{B_{jk}\}_{(k=0,\ldots,K)}$ with break points

$$0 < e_{j0} < e_{j1} < \cdots < e_{jk_j} = v_j \quad (j = 1, \ldots, n). \tag{5.4}$$

Let $\mu_k = B_k(x)$, $k = 0, \ldots, K$, and recall the definitions of I and L' in Eqs. (2.16) and (3.18), respectively. We then have the following

Theorem 5.1. *The nonlinear separable programming* (*NLSP*) *Problem* 1 *is equivalent to the following mixed-integer bilinear programming problem*: *Program* 2.

$$\textit{Maximize} \quad \sum_{j=1}^{n} (f_j(e_{j0})\mu_{j0} + f_j(e_{j1})\mu_{j1} + \cdots + f_j(e_{jK_j})\mu_{jK_j}) \tag{5.5}$$

$$\textit{subject to} \quad \sum_{j=1}^{n} a_{ij}(e_{j0}\mu_{j0} + \cdots + e_{jK_j}\mu_{jK_j}) \leqslant b_i \quad (i = 1, \ldots, m), \tag{5.6}$$

$$\mu_j = (\mu_{j0}, \ldots, \mu_{jK_j}) \in I, \ \forall j. \tag{5.7}$$

Proof. According to the proof of Theorem 2.2, for each j, the mapping

$$\mu_{jk} = B_{jk}(x_j) \quad (k = 0, \ldots, K_j) \quad (j = 1, \ldots, n), \tag{5.8}$$

and the mapping

$$x_j = e_{j0}\mu_{j0} + \cdots + e_{jK}\mu_{jK_j} \quad (j = 1, \ldots, n) \tag{5.9}$$

are inverse transformations. Suppose that there is a feasible solution $\boldsymbol{x}^* = (x_1^*, \ldots, x_n^*)$ to Program 1. From Eq. (5.8) we obtain

$$\mu^* = (\mu_1^*, \ldots, \mu_n^*) = (\mu_{10}^*, \ldots, \mu_{1K_1}^*; \mu_{20}^*, \ldots, \mu_{2K_2}^*; \ldots; \mu_{n0}^*, \ldots, \mu_{nK_n}^*). \tag{5.10}$$

$\mu^* = (\mu_1^*, \ldots, \mu_n^*)$ satisfies Eqs. (5.5)–(5.7) since $\boldsymbol{x}$ and μ are inverse transforms. Thus, $\mu^* = (\mu_1^*, \ldots, \mu_n^*)$ is a feasible solution to Program 2.

Conversely, suppose that there is a feasible solution $\mu^* = (\mu_1^*, \ldots, \mu_n^*)$ for Program 2. From Eq. (5.9) we get $\boldsymbol{x}^* = (x_1^*, \ldots, x_n^*)$. Since $\mu^* = (\mu_1^*, \ldots, \mu_n^*)$ satisfies Eqs. (5.5)–(5.7), $\boldsymbol{x}^* = (x_1^*, \ldots, x_n^*)$ satisfies Eqs. (5.2) and (5.3). Thus, $\boldsymbol{x}^* = (x_1^*, \ldots, x_n^*)$ is a feasible solution to Program 1.

Suppose that the feasible solution $\boldsymbol{x}^* = (x_1^*, \ldots, x_n^*)$ is optimal for Program 1. If the corresponding feasible solution $\mu^* = (\mu_1^*, \ldots, \mu_n^*)$ is not optimal for Program 2, then there is another feasible solution $\mu'^* = (\mu_1'^*, \ldots, \mu_n'^*)$ to Program 2 such that

$$\sum_{j=1}^{n} (f_j(e_{j0}\mu_{j0}'^* + \cdots + f_j(e_{jK_j})\mu_{jK_j}'^*) > \sum_{j=1}^{n} (f_j(e_{j0}\mu_{j0}^* + \cdots + f_j(e_{jK_j})\mu_{jK_j}^*).$$

Then the corresponding feasible solution $\boldsymbol{x}'^* = (x_1'^*, \ldots, x_n'^*)$ of Program 1 will have

$$\left[\sum_{j=1}^{n} f_j^{\wedge}(x_j'^*) > \sum_{j=1}^{n} f_j^{\wedge}(x_j^*)\right]$$

$$\Rightarrow \left[\sum_{j=1}^{n} f_j(x_j'^*) > \sum_{j=1}^{n} f_j(x_j^*)\right]$$

to accuracy ε by Theorem 4.1. This is contradictory to the assumption. Therefore, $\mu^* = (\mu_1^*, \ldots, \mu_n^*)$ is an optimal solution to Program 2. Similarly, we can prove that if the feasible solution $\mu^* = (\mu_1^*, \ldots, \mu_n^*)$ is optimal for Program 2, then the corresponding feasible solution $\boldsymbol{x}^* = (x_1^*, \ldots, x_n^*)$ is optimal for Program 1. □

Theorem 5.1 can be utilized in solving mixed-integer nonlinear programming problems (MINLP), as shown in the following

Theorem 5.2. *The separable MINLP Program* 3.

$$\underset{(x,y)}{\textit{Maximize}} \quad \sum_{j=1}^{n} f_j(x_j) + \sum_{j=1}^{t} g_j y_j,$$

(f_j has continuous derivatives on $[0, v_j]$ for each j)

$$\tag{5.11}$$

$$\textit{subject to} \quad \sum_{j=1}^{n} a_{ij} x_j \leqslant b_i, \quad i = 1, \ldots, m_1, \tag{5.12}$$

$$\sum_{j=1}^{t} g_j y_j \leqslant d_i, \quad i = 1, \ldots, m_2, \tag{5.13}$$

$$x_j \in [0, v_j] \quad (j = 1, \ldots, n),$$

$$y = (y_1, \ldots, y_t) \in Z_+^t$$

can be approximated by Program 4.

$$\underset{(x,y)}{\textit{Maximize}} \quad \sum_{j=1}^{n} \sum_{i=0}^{K_j} f_j(e_{ji}) \mu_{ji} + \sum_{j=1}^{t} g_j y_j \tag{5.14}$$

$$\textit{subject to} \quad \sum_{j=1}^{n} a_{ij} \sum_{i=0}^{K_j} e_{ji} \mu_{ji} \leqslant b_i, \quad i = 1, \ldots, m_1, \tag{5.15}$$

$$\sum_{j=1}^{t} g_j y_j \leqslant d_i, \quad i = 1, \ldots, m_2, \tag{5.16}$$

$$\mu_j = (\mu_{j0}, \ldots, \mu_{jK_j}) \in I \quad (j = 1, \ldots, n),$$

$$y = (y_1, \ldots, y_t) \in Z_+^t$$

to accuracy ε.

Proof. By Theorem 4.1, f_j can be approximated by $f_j^{\wedge}$ $\forall j$ to accuracy ε. By Theorem 5.1, the continuous real-valued subspace of Program 3 can be relaxed into a mixed-integer bilinear form using fuzzy bases. □

6. Convex objective functions

Program 2 is a bilinear mixed-integer programming problem since, for each component, the conditions (2.9)–(2.13) hold. We will now show that Program 2 reduces to an ordinary LP, if the objective function is convex.

Consider the range of the objective function of Program 2,

$$y = f(e_0)\mu_0 + f(e_1)\mu_1 + \cdots + f(e_K)\mu_K \tag{6.1}$$

on the domain D, given the following set of $K + 1$ breakpoints on the plane $X \times Y$:

$$\Phi = \{(e_0, f(e_0)), (e_1, f(e_1)), \ldots, (e_K, f(e_K))\}. \tag{6.2}$$

Applying Theorem 3.3, the set

$$\Gamma_K = \{(e_0\mu_0 + \cdots + e_K\mu_K, f(e_0)\mu_0 + \cdots + f(e_K)\mu_K) \mid \mu \in L\} \tag{6.3}$$

is the convex combination of Φ defined by the simplest (triangular) fuzzy basis (cf. Fig. 3). Let

$$f^{\#}(x) = \max_{y}\{(x, y),\ x \in [a, b],\ y \in I\}, \tag{6.4}$$

which is the top marginal broken line. Theorem 4.1 implies that, if $y = f(x)$ is convex, then

$$f^{\#}(x) = f^{\wedge}(x) \tag{6.5}$$

(cf. Fig. 4). We can then use the following.

Theorem 6.1. *Assume that for each j, $j = 1, \ldots, n$, $y = f_j(x)$ is convex on $[0, b_j]$. Then Program 1 is equivalent to the ordinary LP.*

Program 5.

$$\text{Maximize} \quad \sum_{j=1}^{n} (f_j(e_{j0})\mu_{j0} + f_j(e_{j1})\mu_{j1} + \cdots + f_j(e_{jK_j})\mu_{jK_j}) \tag{6.6}$$

$$\text{subject to} \quad \sum_{j=1}^{n} a_{ij}(e_{j1}\mu_{j1} + \cdots + e_{jK_j}\mu_{jK_j}) \leqslant b_i \quad (i=1,\ldots,m), \tag{6.7}$$

$$e_{j1}\mu_{j1} + \cdots + e_{jK_j}\mu_{iK_j} \geqslant 0 \quad (j=1,\ldots,n), \tag{6.8}$$

$$\sum_{k=1}^{K_j} \mu_{jk} = 1 \quad (j=1,\ldots,n), \tag{6.9}$$

$$\mu_{jk} \geqslant 0 \quad (k=0,\ldots,K_j) \quad (j=1,\ldots,n). \tag{6.10}$$

7. Cutting methods

In some special situations, the membership functions of the constraints of the programming problem are monotone decreasing (e.g., reflecting various costs and volumes of production) from the viewpoint of fuzzy programs. In these cases, the following simplifying assumption is valid:

Assumption 7.1. Suppose that $\boldsymbol{x}' = (x_1', \ldots, x_n')$ is an arbitrary feasible solution to Program 1, i.e., $x_j' \geqslant 0$ $(j=1,\ldots,n)$, $\sum_{j=1}^{n} a_{ij}x_j' \leqslant b_i$ $(i=1,\ldots,m)$. Then, for any $j=1,\ldots,n$ and for any $0 \leqslant x_j < x_j'$, $\boldsymbol{x} = (x_1', \ldots, x_j, \ldots, x_n')$ is also a feasible solution to Program 1, i.e., $a_{i_1}x_1' + \cdots + a_{i,j-1}x_{j-1}' + a_{ij}x_j + a_{i,j+1}x_{j+1}' + \cdots + a_{in}x_n' \leqslant b_i$ $(i=1,\ldots,m)$.

Let

$$\Delta^{f_j} = \{x_j \mid x_j \in [0, b_j];\ (\forall u \in [0, x_j]);\ f_j(u) \leqslant f_j(x_j)\}, \quad j=1,\ldots,n. \tag{7.1}$$

Theorem 7.1. *Suppose that Program* 1 *satisfies Assumption* 7.1. *Then the program is equivalent to*

Program 6.

$$\text{Maximize} \quad \sum_{j=1}^{n} f_j(x_j) \tag{7.2}$$

$$\text{subject to} \quad x_j \in \Delta^{f_j} \quad (j=1,\ldots,n), \tag{7.3}$$

$$\sum_{j=1}^{n} a_{ij}x_j \leqslant b_i \quad (i=1,\ldots,m). \tag{7.4}$$

Proof. Since the feasible domain of Program 6 is included in that of Program 1, we need to prove that any optimal solution to Program 1 must be a feasible solution to Program 6. Suppose that $\boldsymbol{x}^* = (x_1^*, \ldots, x_n^*)$ is an optimal solution to Program 1. Then, according to Assumption 7.1, for any $j=1,\ldots,n$, and any $u \in [0, x_j]$, $\boldsymbol{x}' = (x_1^*, \ldots, u, \ldots, x_n^*)$ is also a feasible solution to Program 1. Since $\boldsymbol{x}^* = (x_1^*, \ldots, x_n^*)$ is optimal and $f_j(u) \leqslant f_j(x_j^*)$, we have that

$$\sum_{i=1}^{j-1} f_i(x_i^*) + f_j(u) + \sum_{i=j+1}^{n} f_i(x_i^*) \leqslant \sum_{i=1}^{n} f_i(x_i^*). \tag{7.5}$$

This means that $x_j^* \in \Delta^{f_j}$ $(j=1,\ldots,n)$, so Eq. (7.3) holds. Thus $\boldsymbol{x}^* = (x_1^*, \ldots, x_n^*)$ is a feasible solution to Program 6. □

The meaning of Theorem 7.1 can be illustrated in the following examples:

Example 7.1. Suppose that Program 1 satisfies Assumption 7.1 with $f_j(x)$ as given in Fig. 8. Then for any j, if $f_j(x)$ takes its maximum at $x_j = e_j$ we can omit the interval $[e_j, b_j]$.

Example 7.2. Suppose that Program 1 satisfies Assumption 7.1 with $f_j(x)$ as given in Fig. 9. Then for any j, if $f_j(x)$ takes its local maximum at $x_j = e_j$ we can omit the interval $[e_j, e_j + \delta]$, whenever $f_j(e_j + h) \leqslant f_j(e_j)$ $(h \leqslant \delta)$.

Example 7.3. Suppose that Program 1 satisfies Assumption 7.1 and that for some j, $f_j(x_j)$ is a polynomial as shown in Fig. 10. Then the jth component of the feasible solution cannot be found in the

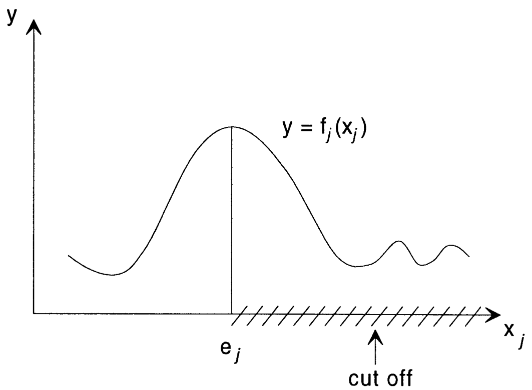

Fig. 8. A hypothetical nonlinear function (Example 7.1).

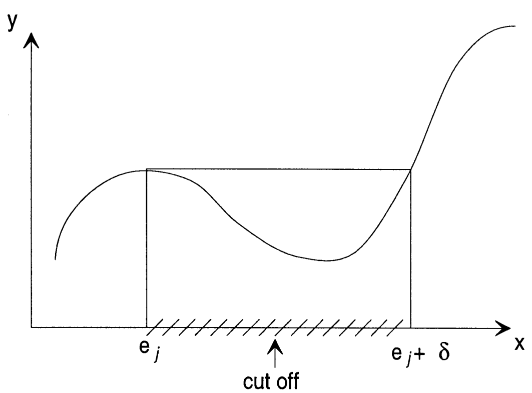

Fig. 9. A hypothetical nonlinear function (Example 7.2).

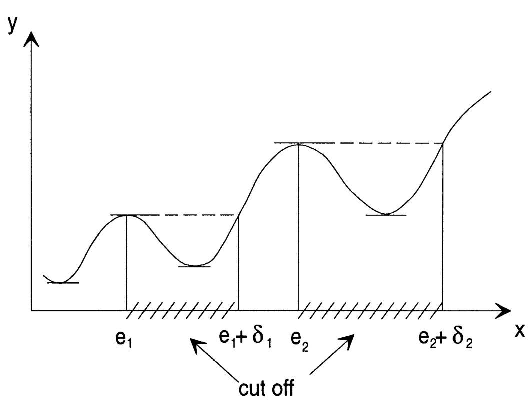

Fig. 10. A hypothetical nonlinear function (Example 7.3).

following sub-intervals of $[0, b_j]$:

$$\{[e_i, e_i + \delta_i] \mid f_i'(e_i) = 0;\ f_i(e_i + h) \leqslant f_i(e_i);\ (h \leqslant \delta_i);\ i \in \{1, K\}\}. \tag{7.6}$$

8. Conclusion

In this paper we have introduced the concept of fuzzy bases and fuzzy interpolation and presented a set of theoretical results that demonstrate their usefulness in solving nonlinear programming problems by means of fuzzy set theory. We have shown that fuzzy interpolation, a linear combination of the dimensions in a fuzzy basis, can be used to approximate a nonlinear function to any degree of accuracy. We have also shown that a mixed integer nonlinear separable programming problem with linear constraints can be relaxed to a bilinear program by means of fuzzy interpolation. If the nonlinear objective function is convex, we can use fuzzy interpolation to transform the NLP into an ordinary LP. If the objective function is nonconvex, we can use the concept of K-stage fuzzy linear bases to transform the NLP to a bilinear programming problem.

The results suggest several interesting paths for future research. Firstly, fuzzy interpolation may be used to recursively linearize complicated NLPs around local solutions. In particular, if the optimization problem is convex, our results indicate that fuzzy bases may provide a competitive alternative or, at least a powerful supplement to classical MINLP-algorithms. Secondly, further research aimed at applying cutting methods is worthwhile, since proper cuts can lead to drastic reduction in problem size and complexity in certain problems. Thirdly, research in the theory of fuzzy nonlinear bases is relevant. We can simplify complicated nonlinear functions using linear or nonlinear fuzzy bases, depending on the problem type. Nonlinear fuzzy bases may be more parsimonious than linear bases and still simpler in structure than the original function.

References

[1] E.E. Amman, A.E.-H. Kassem, On stability analysis of multivariate LP problems with fuzzy parameters, Fuzzy Sets and Systems 83 (1996) 331–334.

[2] R. Bellman, L.A. Zadeh, Decision-making in a fuzzy environment, Management Sci. 17B (1970) 141–164.
[3] J.J. Buckley, Joint solution to fuzzy programming problems, Fuzzy Sets and Systems 72 (1995) 215–220.
[4] H. Narazaki, T. Watanabe, A case-based approach for modelling nonlinear systems, Fuzzy Sets and Systems 77 (1996) 77–86.
[5] R. Östermark, Temporal interdependence in fuzzy MCDM problems, Fuzzy Sets and Systems 88 (1) (1997) 69–79.
[6] O.M. Saad, Stability on multiobjective linear programming problems with fuzzy parameters, Fuzzy Sets and Systems 74 (1995) 207–215.
[7] P.Z. Wang, The essential of fuzzy computing and its applications on nonlinear programming, J. Chinese Fuzzy System Assoc. 1 (1995) 29–38.
[8] P.Z. Wang, S.H. Tan, F. Song, P. Liang, Constructive theory for fuzzy systems, Fuzzy Sets and Systems 88 (1997) 195–203.
[9] P.Z. Wang, S.K. Tan, S.H. Tan, Toward a framework for fuzzy dynamic systems, IEEE Expert Intelligent Systems Appl. 9 (4) (1994) 39–40.
[10] P.Z. Wang, D.Z. Zhang, E. Sanchez, E.S. Lee, Latticized linear programming and fuzzy relation inequalities, J. Math. Anal. Appl. 159 (1) (1991) 72–87.
[11] H.-J. Zimmermann, Fuzzy programming and linear programming with several objective functions, Fuzzy Sets and Systems 1 (1978) 45–55.