

AIMS Mathematics, 5(6): 6149–6168. DOI: 10.3934/math.2020395 Received: 22 June 2020 Accepted: 20 July 2020 Published: 29 July 2020

http://www.aimspress.com/journal/Math

Research article

Study on reasonable initialization enhanced Karnik-Mendel algorithms for centroid type-reduction of interval type-2 fuzzy logic systems

Yang Chen*, Jinxia Wu and Jie Lan

College of Science, Liaoning University of Technology, Jinzhou, Liaoning, 121001, P. R. China

* Correspondence: Email: lxychenyang@lnut.edu.cn; Tel: +8613897856294; Fax: +864164199415.

Abstract: Type-reduction (TR) is a key block for interval type-2 fuzzy logic systems (IT2 FLSs). In general, Karnik-Mendel (KM) (or enhanced Karnik-Mendel (EKM)) algorithms are used to perform the TR. These two types of algorithms have the advantage of preserving the uncertainties of membership functions (MFs) flow in IT2 FLSs. This paper gives the initialization explanations of KM and EKM algorithms, and proposes reasonable initialization enhanced Karnik-Mendel (RIEKM) algorithms for centroid TR of IT2 FLSs. By considering the accurate continuous Nie-Tan (CNT) algorithms as the benchmark, four computer simulation examples are adopted to illustrate and analyze the performances of RIEKM algorithms, the proposed RIEKM algorithms have smaller absolute errors and faster convergence speeds, which afford the potential value for designing and applying IT2 FLSs.

Keywords: interval type-2 fuzzy logic systems; centroid; reasonable initialization enhanced Karnik-Mendel (RIEKM) algorithms; absolute errors; computer simulation **Mathematics Subject Classification:** 68XX, 68Uxx

1. Introduction

As we all know, the membership grades of type-1 fuzzy sets (T1 FSs) are crisp numbers as 0 or 1, therefore, the membership functions (MFs) of T1 FSs are not inherently uncertain and they can only measure uncertainties in a limited scope. The membership grades of interval type-2 fuzzy sets (IT2 FSs) are intervals, so the IT2 FSs are capable of better model uncertainties [1,2]. In the past decades, great progresses have been made transiting from type-1 fuzzy logic systems (T1 FLSs) to

interval type-2 fuzzy logic systems (IT2 FLSs). IT2 FLSs based on IT2 FSs can approximate real continuous functions defined on compact sets with arbitrary accuracy. Furthermore, IT2 FLSs have been successfully applied in many fields with uncertainty, nonlinearity and time-varying characteristics like autonomous mobile robots [3,4], intelligent controllers [5], financial systems [6], power systems [7,8], permanent magnetic drive [9–11], edge detection [12], medical systems [13] and hot strip mill [14,15] and so on.

Generally speaking, IT2 FLSs (see Figure 1) are composed of fuzzifier, rules, inference [16], type-reducer and defuzzifier. Among which, the block of type-reduction (TR) under the guidance of inference plays the central role, and its main function is to transform the IT2 FS to the T1 FS. Then the defuzzification changes the T1 FS to the crisp output. The operations in TR are the differences between T1 and IT2 FLSs, which makes the latter with more challenges.



Figure 1. An IT2 FLS [2,17].

Currently, the computationally intensive iterative Karnik-Mendel (KM) algorithms and enhanced Karnik-Mendel (EKM) algorithms [18–20] are the most popular approaches for performing the TR. These two types of algorithms have the advantages of preserving the uncertainties flow in the systems and converge in super-convergence speed. However, the initializations of KM and EKM algorithms are usually given by trial and error of extensive simulation experiments. This paper analyzes the initializations [21] of KM and EKM algorithms, and provides reasonable initialization EKM (RIEKM) algorithms for performing the centroid TR and defuzzification of IT2 FLSs. According the accurate benchmark CNT algorithms [22], the proposed RIEKM algorithms have smaller absolute errors and faster convergence speeds compared with the EKM algorithms.

The rest of this paper is organized as follows. Section 2 gives the background of IT2 FLSs. Section 3 provides the RIEKM algorithms, and how to adopt them to perform the centroid TR of IT2 FLSs. In Section 4, four computer simulation experiments are used to illustrate and analyze the performances of RIEKM algorithms. Finally the conclusions and expectations are given in Section 5.

2. IT2 FLSs

The rules uncertainties of IT2 FLSs generate from numerical or language uncertainties in knowledge, while these uncertainties can be solved by T2 FSs. In fact, the concept of T2 FSs can be viewed as the extension of concept of T1 FSs.

Definition 1. A T2 FS \tilde{A} can be characterized by its T2 MF $\mu_{\tilde{A}}(x,u)$, i.e.,

$$\tilde{A} = \{(x,u), \mu_{\tilde{A}}(x,u) \mid \forall x \in X, \forall u \in [0,1]\}$$

$$\tag{1}$$

in which the primary variable $x \in X$, and the secondary variable $u \in [0,1]$, here equation (1) is usually referred to as the point-value expression, and whose compact form is as:

$$\widetilde{A} = \int_{x \in X} \int_{u \in J_x \subseteq [0,1]} \mu_{\widetilde{A}}(x,u) / (x,u).$$
⁽²⁾

Definition 2. A vertical slice of $\mu_{\tilde{A}}(x,u)$ is the secondary MF, i.e.,

$$\mu_{\tilde{A}}(x=x',u) \equiv \mu_{\tilde{A}}(x') = \int_{\forall u \in J_{x'}} f_{x'}(u) / u.$$
(3)

Definition 3. The two dimensional support of $\mu_{\tilde{A}}(x,u)$ is called as the footprint of uncertainty (FOU), i.e.,

$$FOU(\tilde{A}) = \{(x, u) \in X \times [0, 1] | \mu_{\tilde{A}}(x, u) > 0\}$$
(4)

here the upper and lower bounds of $FOU(\widetilde{A})$ are referred to as the upper MF (UMF) and lower MF (LMF), respectively, i.e.,

$$\operatorname{UMF}(\widetilde{A}) = \overline{\mu}_{\widetilde{A}}(x) = \overline{\operatorname{FOU}(\widetilde{A})}, \quad \operatorname{LMF}(\widetilde{A}) = \underline{\mu}_{\widetilde{A}}(x) = \underline{\operatorname{FOU}(\widetilde{A})}.$$
 (5)

For any IT2 FSs, they can be completely characterized by their UMFs and LMFs. That's because the secondary membership grades of IT2 FSs must be equal to 1, i.e., $f_x(u) \equiv 1$.

From the aspect of inference structure, IT2 FLSs can be divided into Mamdani type [7,11,14] and Takagi-Sugeno-Kang (TSK) type [10,15]. Without loss of generality, consider a Mamdani IT2 FLS with *p* inputs $x_1 \in X_1, x_2 \in X_2, \Lambda$, $x_p \in X_p$ and one output $y \in Y$, there are a total of *M* fuzzy rules, where the *lth* rule is of the form:

If
$$x_1$$
 is \tilde{F}_1^l and x_2 is \tilde{F}_2^l and ... and x_p is \tilde{F}_p^l , then y is \tilde{G}^l $(l=1,\Lambda,M)$. (6)

In order to simplify the expressions, we model the input measurements as crisp sets, i.e., singleton fuzzifier is adopted. Then the process of fuzzy reasoning [7,16,23] is given as follows:

The fuzzy relation of each fuzzy rule is as:

$$\widetilde{R}^{l}:\widetilde{F}_{1}^{l}\times\widetilde{F}_{2}^{l}\Lambda\times\widetilde{F}_{p}^{l}\to\widetilde{G}^{l}=\widetilde{A}^{l}\to\widetilde{G}^{l}$$
(7)

whose MF is as:

$$\mu_{\tilde{R}^{l}}(x, y) = \mu_{\tilde{A}^{l} \to \tilde{G}^{l}}(x, y) = [I_{i=1}^{p} \mu_{\tilde{R}^{l}}(x_{i}')]I_{\tilde{G}^{l}}(y)$$
(8)

where the sign I is the product or minimum t-norm [24].

The T2 output of each fuzzy rule is $\tilde{B}^{l} = A_{r} \circ \tilde{R}^{l}$, and whose MF is as:

$$\mu_{\tilde{B}^{l}}(y) = Y_{x \in X}[\mu_{A_{x}}(x) I \ \mu_{\tilde{A}^{l} \to \tilde{G}^{l}}(x, y)] = \mu_{\tilde{G}^{l}}(y) I \ [I \ _{i=1}^{p} \mu_{\tilde{F}^{l}_{i}}(x'_{i})] = \mu_{\tilde{G}^{l}}(y) I \ F^{l}(x')$$
(9)

in which o is the composition operation, and Y is the maximum t-norm. $F^{l}(x')$ is the defined firing interval, i.e.,

$$F^{l}:\begin{cases} F^{l}(x') \equiv [\underline{f}^{l}(x'), \overline{f}^{l}(x')] \\ \underline{f}^{l}(x') \equiv I^{p}_{i=1} \underline{\mu}_{\tilde{F}_{i}^{l}}(x'_{i}) \\ \overline{f}^{l}(x') \equiv I^{p}_{i=1} \overline{\mu}_{\tilde{F}_{i}^{l}}(x'_{i}) \end{cases}$$
(10)

Here the most popular centroid TR approach [21] is selected, i.e., the output IT2 FS \tilde{B}^{l} of each rule (which generates by merging each fuzzy rule and its corresponding consequent IT2 FS) is described as:

$$\begin{cases} \text{FOU}(\widetilde{B}^{l}) = [\underline{\mu}_{\widetilde{B}^{l}}(y \mid x'), \overline{\mu}_{\widetilde{B}^{l}}(y \mid x')] \\ \underline{\mu}_{\widetilde{B}^{l}}(y \mid x') = \underline{f}^{l}(x') * \underline{\mu}_{\widetilde{G}^{l}}(y) \\ \overline{\mu}_{\widetilde{B}^{l}}(y \mid x') = \overline{f}^{l}(x') * \overline{\mu}_{\widetilde{G}^{l}}(y) \end{cases}$$
(11)

where * denotes the product or minimum t-norm.

Then aggregating all the firing rule IT2 FS \tilde{B}^l to obtain the final output \tilde{B} , i.e.,

$$\widetilde{B}:\begin{cases} \operatorname{FOU}(\widetilde{B}) = [\underline{\mu}_{\widetilde{B}}(y \mid x'), \overline{\mu}_{\widetilde{B}}(y \mid x')] \\ \underline{\mu}_{\widetilde{B}}(y \mid x') = \underline{\mu}_{\widetilde{B}^{1}}(y \mid x') \lor \Lambda \lor \underline{\mu}_{\widetilde{B}^{M}}(y \mid x') \\ \overline{\mu}_{\widetilde{B}}(y \mid x') = \overline{\mu}_{\widetilde{B}^{1}}(y \mid x') \lor \Lambda \lor \overline{\mu}_{\widetilde{B}^{M}}(y \mid x') \end{cases}$$
(12)

where \vee represents the maximum operation. Then the type-reduced set Y_c can be obtained by computing the centroid $C_{\tilde{B}}$ of \tilde{B} , i.e.,

$$Y_C = C_{\tilde{B}} = 1/[l_{\tilde{B}}, r_{\tilde{B}}]$$
⁽¹³⁾

where $l_{\tilde{B}}$ and $r_{\tilde{B}}$ can be computed by KM types of algorithms as:

$$l_{\tilde{B}} = \min_{k=1,\Lambda,N} l_{\tilde{B}}(k) = \min_{k=1,\Lambda,N} \frac{\sum_{i=1}^{k} y_{i} \mu_{\tilde{B}}(y_{i}) + \sum_{i=k+1}^{N} y_{i} \mu_{\tilde{B}}(y_{i})}{\sum_{i=1}^{k} \mu_{\tilde{B}}(y_{i}) + \sum_{i=k+1}^{N} \mu_{\tilde{B}}(y_{i})}$$
(14)
$$r_{\tilde{B}} = \max_{k=1,\Lambda,N} r_{\tilde{B}}(k) = \max_{k=1,\Lambda,N} \frac{\sum_{i=1}^{k} y_{i} \mu_{\tilde{B}}(y_{i}) + \sum_{i=k+1}^{N} y_{i} \mu_{\tilde{B}}(y_{i})}{\sum_{i=1}^{k} \mu_{\tilde{B}}(y_{i}) + \sum_{i=k+1}^{N} \mu_{\tilde{B}}(y_{i})}$$
(15)

where N is the number of sampling of primary variable y, and k is the switch point.

Finally the crisp output is computed by taking the arithmetic average of $l_{\tilde{B}}$ and $r_{\tilde{B}}$, i.e.,

$$y = (l_{\tilde{B}} + r_{\tilde{B}})/2. \tag{16}$$

3. **RIEKM algorithms**

First of all, we derive the theoretical interpretations of initialization of KM and EKM algorithms. Let $a = y_1 < y_2 < \Lambda < y_N = b$, then the continuous KM and EKM algorithms compute as (see Eqs (14) and (15)):

$$l_{\tilde{B}} = \min_{\xi \in [a,b]} F_l(\xi) = \min_{\xi \in [a,b]} \frac{\int_a^{\xi} y \overline{\mu}_{\tilde{B}}(y) dy + \int_{\xi}^{b} y \underline{\mu}_{\tilde{B}}(y) dy}{\int_a^{\xi} \overline{\mu}_{\tilde{B}}(y) dy + \int_{\xi}^{b} \underline{\mu}_{\tilde{B}}(y) dy}$$
(17)

$$r_{\tilde{B}} = \min_{\xi \in [a,b]} F_r(\xi) = \min_{\xi \in [a,b]} \frac{\int_a^{\xi} y \underline{\mu}_{\tilde{B}}(y) dy + \int_{\xi}^b y \overline{\mu}_{\tilde{B}}(y) dy}{\int_a^{\xi} \underline{\mu}_{\tilde{B}}(y) dy + \int_{\xi}^b \overline{\mu}_{\tilde{B}}(y) dy}.$$
(18)

In addition, the specific computation steps of CKM and CEKM algorithms are given in Tables 1 and 2. According to the notations F_l in (17) and F_r in (18), from Steps 2 and 4 in Table 1, we can find:

$$\xi_l = F_l(\xi'), \text{ and } \xi_l = \xi' \tag{19}$$

$$\xi_r = F_r(\xi'), \text{ and } \xi_r = \xi'.$$
(20)

When iterations terminate, $l_{\tilde{B}} = \xi_l$ and $r_{\tilde{B}} = \xi_r$, therefore

$$l_{\tilde{B}} = F_l(l_{\tilde{B}}), \quad r_{\tilde{B}} = F_r(r_{\tilde{B}})$$
(21)

here $l_{\tilde{B}}$ and $r_{\tilde{B}}$ are the fixed points of $F_l(\xi)$ and $F_r(\xi)$, respectively. In like manner, the relations of (21) still hold for the CEKM algorithms as in Table 2.

In order to set the initialization of the algorithms, let $\underline{\mu}_{\tilde{B}}(y) = \overline{\mu}_{\tilde{B}}(y) = \theta(y)$ for all $y \in [a,b]$, then $\theta(y) = [\underline{\mu}_{\tilde{B}}(y) + \overline{\mu}_{\tilde{B}}(y)]/2$. In this case, Eqs (17) and (18) become the same, so that,

$$l_{\tilde{B}} = r_{\tilde{B}} = \frac{\int_{a}^{b} y\theta(y)dy}{\int_{a}^{b} \theta(y)dy} = \frac{\int_{a}^{b} y[\underline{\mu}_{\tilde{B}}(y) + \overline{\mu}_{\tilde{B}}(y)]/2dy}{\int_{a}^{b} [\underline{\mu}_{\tilde{B}}(y) + \overline{\mu}_{\tilde{B}}(y)]/2dy}$$
(22)

Equation (22) is the initialization approach of CKM algorithms given in Table 1, denoted here as $\xi^{(1)}$, i.e.,

$$\xi^{(1)} = \frac{\int_{a}^{b} y\theta(y)dy}{\int_{a}^{b} \theta(y)dy}$$
(23)

in which $\theta(y) = [\underline{\mu}_{\tilde{B}}(y) + \overline{\mu}_{\tilde{B}}(y)]/2$.

					~
Table 1.	CKM algorithms	[18,21,25]	for computing	the centroid of	В.

Step	CKM algorithm for $l_{\tilde{B}}$,
	$l_{\tilde{B}} = \min_{\forall \theta(y) \in [\underline{\mu}_{\tilde{B}}(y), \overline{\mu}_{\tilde{B}}(y)]} \frac{\int_{a}^{b} y \theta(y) dy}{\int_{a}^{b} \theta(y) dy}$
1	Let $\theta(y) = [\underline{\mu}_{\tilde{B}}(y) + \overline{\mu}_{\tilde{B}}(y)]/2$, and compute $\xi' = \frac{\int_{a}^{b} y \theta(y) dy}{\int_{a}^{b} \theta(y) dy}$.
2	Set $\theta(y) = \overline{\mu}_{\tilde{B}}(y)$ when $y \le \xi'$, and $\theta(y) = \underline{\mu}_{\tilde{B}}(y)$ when $y > \xi'$,
	and compute $\xi_l = \frac{\int_a^b y \theta(y) dy}{\int_a^b \theta(y) dy}.$
3	Check if $ \xi' - \xi_l \leq \varepsilon$ (ε is a given error bound), if yes, stop and set
	$l_{\tilde{B}} = \xi_l$, if no, go
4	to Step 4. Set $\mathcal{E}' = \mathcal{E}$ and go to Step 2
Stop	<u>r</u>
Step	CKM algorithm for $r_{\tilde{B}}$, $r_{\tilde{B}} = \max_{\forall \theta(y) \in [\underline{\mu}_{\tilde{B}}(y), \overline{\mu}_{\tilde{B}}(y)]} \frac{\int_{a}^{y} y \theta(y) dy}{\int_{a}^{b} \theta(y) dy}$
1	Let $\theta(y) = [\mu_{z}(y) + \overline{\mu}_{z}(y)]/2$, and compute $\xi' = \frac{\int_{a}^{b} y \theta(y) dy}{dy}$.
	$\int_{a}^{b} \theta(y) dy$
2	Set $\theta(y) = \underline{\mu}_{\tilde{B}}(y)$ when $y \le \xi'$, and $\theta(y) = \overline{\mu}_{\tilde{B}}(y)$ when $y > \xi'$,
2	Set $\theta(y) = \underline{\mu}_{\tilde{B}}(y)$ when $y \le \xi'$, and $\theta(y) = \overline{\mu}_{\tilde{B}}(y)$ when $y > \xi'$, and compute $\xi_r = \frac{\int_a^b y \theta(y) dy}{\int_a^b \theta(y) dy}$.
2 3	Set $\theta(y) = \underline{\mu}_{\tilde{B}}(y) \mu_{\tilde{B}}(y) \text{ dy}$ and compute $\xi_r = \frac{\int_a^b y\theta(y)dy}{\int_a^b \theta(y)dy}$. Check if $ \xi' - \xi_r \le \varepsilon$ (ε is a given error bound), if yes, stop and set
2 3	Set $\theta(y) = \underline{\mu}_{\tilde{B}}(y) + \mu_{\tilde{B}}(y) \text{ dy}$ and compute $\xi_r = \frac{\int_a^b y\theta(y)dy}{\int_a^b \theta(y)dy}$. Check if $ \xi' - \xi_r \le \varepsilon \ (\varepsilon \text{ is a given error bound})$, if yes, stop and set $r_{\tilde{B}} = \xi_r$, if no, go
2 3	Set $\theta(y) = \underline{\mu}_{\tilde{B}}(y) \mu_{\tilde{B}}(y) \text{ when } y \leq \xi'$, and $\theta(y) = \overline{\mu}_{\tilde{B}}(y) \text{ when } y > \xi'$, and compute $\xi_r = \frac{\int_a^b y \theta(y) dy}{\int_a^b \theta(y) dy}$. Check if $ \xi' - \xi_r \leq \varepsilon$ (ε is a given error bound), if yes, stop and set $r_{\tilde{B}} = \xi_r$, if no, go to Step 4.

Step	CEKM algorithm for $l_{\tilde{B}}$		
1	Set $c = a + (b-a)/2.4$, and compute $\alpha = \int_a^c y \overline{\mu}_{\widetilde{B}}(y) dy + \int_c^b y \underline{\mu}_{\widetilde{B}}(y) dy$,		
	$\beta = \int_a^c \overline{\mu}_{\widetilde{B}}(y) dy + \int_c^b \underline{\mu}_{\widetilde{B}}(y) dy, c' = \alpha / \beta.$		
2	Check if $ c'-c \leq \varepsilon$, if yes, stop and set $c' = l_{\tilde{B}}$, if no, go to Step 4.		
3	Compute $s = sign(c'-c)$, $\alpha' = \alpha + s \int_{\min(c,c')}^{\max(c,c')} y[\overline{\mu}_{\widetilde{B}}(y) - \underline{\mu}_{\widetilde{B}}(y)] dy$,		
	$\beta' = \beta + s \int_{\min(c,c')}^{\max(c,c')} [\overline{\mu}_{\tilde{B}}(y) - \underline{\mu}_{\tilde{B}}(y)] dy, \ c'' = \alpha' / \beta'.$		
4	Set $c = c', c' = c'', \alpha = \alpha', \beta = \beta'$ and go to Step 2.		
Step	CEKM algorithm for $r_{\tilde{B}}$		
1	Set $c = a + (b-a)/1.7$, and compute $\alpha = \int_a^c y \underline{\mu}_{\tilde{B}}(y) dy + \int_c^b y \overline{\mu}_{\tilde{B}}(y) dy$,		
	$\beta = \int_{-\infty}^{\infty} \mu(y) dy + \int_{-\infty}^{\infty} \frac{1}{(y)} dy c' = \alpha / \beta$		
	$p \int_{a} \underline{\mu}_{\widetilde{B}}(y) dy + \int_{c} \mu_{B}(y) dy, c = d + p$		
2	Check if $ c'-c \leq \varepsilon$, if yes, stop and set $c' = r_{\tilde{B}}$, if no, go to Step 4.		
2 3	Check if $ c'-c \le \varepsilon$, if yes, stop and set $c' = r_{\tilde{B}}$, if no, go to Step 4. Compute $s = sign(c'-c)$, $\alpha' = \alpha - s \int_{\min(c,c')}^{\max(c,c')} y[\overline{\mu}_{\tilde{B}}(y) - \underline{\mu}_{\tilde{B}}(y)] dy$,		
2 3	$\beta' = \beta - s \int_{\min(c,c')}^{\max(c,c')} [\overline{\mu}_{\tilde{B}}(y) - \underline{\mu}_{\tilde{B}}(y)] dy, c'' = \alpha' / \beta'.$ Check if $ c' - c \le \varepsilon$, if yes, stop and set $c' = r_{\tilde{B}}$, if no, go to Step 4. Compute $s = sign(c' - c), \alpha' = \alpha - s \int_{\min(c,c')}^{\max(c,c')} y[\overline{\mu}_{\tilde{B}}(y) - \underline{\mu}_{\tilde{B}}(y)] dy,$ $\beta' = \beta - s \int_{\min(c,c')}^{\max(c,c')} [\overline{\mu}_{\tilde{B}}(y) - \underline{\mu}_{\tilde{B}}(y)] dy, c'' = \alpha' / \beta'.$		

Table 2. CEKM algorithms [18,21,25] for computing the centroid of \widetilde{B} .

The specific computation steps of discrete KM and EKM algorithms are provided in Tables 3 and 4. And the discrete form of $\xi^{(1)}$ is established in Steps 1 and 2 in Table 3, i.e.,

$$k^{(1)} = \{k \mid y_{k} \leq \frac{\sum_{i=1}^{N} y_{i}[\underline{\mu}_{\tilde{B}}(y_{i}) + \overline{\mu}_{\tilde{B}}(y_{i})]}{\sum_{i=1}^{N} [\underline{\mu}_{\tilde{B}}(y_{i}) + \overline{\mu}_{\tilde{B}}(y_{i})]} < y_{k+1}, 1 \leq k \leq N-1\}$$
(24)

this KM initialization approach can provide excellent results as $\underline{\mu}_{\tilde{B}}(y)$ and $\overline{\mu}_{\tilde{B}}(y)$ are very close to each other, that's because it becomes the exact optimal solution of (14) or (15) when $\mu_{\tilde{B}}(y) = \overline{\mu}_{\tilde{B}}(y)$.

For the EKM algorithms, the initialization approach is based on the difference between $\underline{\mu}_{\tilde{B}}(y)$ and $\overline{\mu}_{\tilde{B}}(y)$. Let's make an assumption that

$$\rho = \frac{\int_{a}^{b} \overline{\mu}_{\tilde{B}}(y) dy}{\int_{a}^{b} \underline{\mu}_{\tilde{B}}(y) dy}.$$
(25)

Attention that $\rho \ge 1$, because $\overline{\mu}_{\tilde{B}}(y) \ge \underline{\mu}_{\tilde{B}}(y)$ for all $y \in [a,b]$.

In order to initialize the EKM algorithms, suppose that both $\underline{\mu}_{\tilde{B}}(y)$ and $\overline{\mu}_{\tilde{B}}(y)$ be constants

for $y \in [a,b]$, i.e., $\underline{\mu}_{\tilde{B}}(y) = n > 0$, and $\overline{\mu}_{\tilde{B}}(y) = \rho n > 0$, according to Eq (14), so that

$$F_{l}(\xi) = \frac{\int_{a}^{\xi} y \overline{\mu}_{\tilde{B}}(y) dy + \int_{\xi}^{b} y \underline{\mu}_{\tilde{B}}(y) dy}{\int_{a}^{\xi} \overline{\mu}_{\tilde{B}}(y) dy + \int_{\xi}^{b} \underline{\mu}_{\tilde{B}}(y) dy} = \frac{\int_{a}^{\xi} \rho ny dy + \int_{\xi}^{b} ny dy}{\int_{a}^{\xi} \rho ndy + \int_{\xi}^{b} ndy}$$

$$= \frac{\int_{a}^{\xi} \rho y dy + \int_{\xi}^{b} y dy}{\int_{a}^{\xi} \rho dy + \int_{\xi}^{b} dy} = \frac{\rho(\xi^{2} - a^{2}) + (b^{2} - \xi^{2})}{2[\rho(\xi - a) + (b - \xi)]}$$
(26)

Then find the derivative of $F_l(\xi)$, so that

$$F'_{l}(\xi) = \frac{(\rho-1)[\rho(\xi-a)^{2} - (b-\xi)^{2}]}{2[\rho(\xi-a) + (b-\xi)]^{2}}.$$
(27)

Table 3. KM algorithms [18,21,25] for computing the centroid of \widetilde{B} .

Step	KM algorithm for $l_{\tilde{B}}$, $l_{\tilde{B}} = \min_{\forall \theta_i \in [\underline{\mu}_{\tilde{B}}(y_i), \overline{\mu}_{\tilde{B}}(y_i)]} (\sum_{i=1}^N y_i \theta_i) / (\sum_{i=1}^N \theta_i)$
1	Set $\theta(i) = [\underline{\mu}_{\tilde{B}}(y_i) + \overline{\mu}_{\tilde{B}}(y_i)]/2, i = 1, \Lambda, N$ and compute
	$c' = \left(\sum_{i=1}^{N} y_i \theta_i\right) / \left(\sum_{i=1}^{N} \theta_i\right).$
2	Find $k'(1 \le k' \le N - 1)$ such that $y_{k'} \le c' < y_{k'+1}$.
3	Set $\theta_i = \overline{\mu}_{\widetilde{B}}(y_i)$ when $i \le k'$, and $\theta_i = \underline{\mu}_{\widetilde{B}}(y_i)$ when $i > k'$,
	and compute $l_{\tilde{B}}(k') = (\sum_{i=1}^{N} y_i \theta_i) / (\sum_{i=1}^{N} \theta_i)$.
4	Check if $l_{\tilde{B}}(k') = c'$, if yes, stop and set $l_{\tilde{B}}(k') = l_{\tilde{B}}$ and $k' = L$, if no, go
	to Step 5.
5	Set $c' = l_{\tilde{B}}(k')$ and go to Step 2.
Step	KM algorithm for $r_{\tilde{B}}$, $r_{\tilde{B}} = \max_{\forall \theta_i \in [\underline{\mu}_{\tilde{B}}(y_i), \mu_{\tilde{B}}(y_i)]} (\sum_{i=1}^N y_i \theta_i) / (\sum_{i=1}^N \theta_i)$
Step 1	KM algorithm for $r_{\tilde{B}}$, $r_{\tilde{B}} = \max_{\forall \theta_i \in [\underline{\mu}_{\tilde{B}}(y_i), \overline{\mu}_{\tilde{B}}(y_i)]} (\sum_{i=1}^N y_i \theta_i) / (\sum_{i=1}^N \theta_i)$ Set $\theta(i) = [\underline{\mu}_{\tilde{B}}(y_i) + \overline{\mu}_{\tilde{B}}(y_i)] / 2, i = 1, \Lambda, N$ and compute
Step 1	KM algorithm for $r_{\tilde{B}}$, $r_{\tilde{B}} = \max_{\forall \theta_i \in [\underline{\mu}_{\tilde{B}}(y_i), \overline{\mu}_{\tilde{B}}(y_i)]} (\sum_{i=1}^N y_i \theta_i) / (\sum_{i=1}^N \theta_i)$ Set $\theta(i) = [\underline{\mu}_{\tilde{B}}(y_i) + \overline{\mu}_{\tilde{B}}(y_i)] / 2, i = 1, \Lambda, N$ and compute $c' = (\sum_{i=1}^N y_i \theta_i) / (\sum_{i=1}^N \theta_i).$
Step 1 2	KM algorithm for $r_{\tilde{B}}$, $r_{\tilde{B}} = \max_{\forall \theta_i \in [\underline{\mu}_{\tilde{B}}(y_i), \overline{\mu}_{\tilde{B}}(y_i)]} (\sum_{i=1}^N y_i \theta_i) / (\sum_{i=1}^N \theta_i)$ Set $\theta(i) = [\underline{\mu}_{\tilde{B}}(y_i) + \overline{\mu}_{\tilde{B}}(y_i)] / 2, i = 1, \Lambda, N$ and compute $c' = (\sum_{i=1}^N y_i \theta_i) / (\sum_{i=1}^N \theta_i)$. Find $k'(1 \le k' \le N - 1)$ such that $y_{k'} \le c' < y_{k'+1}$.
Step 1 2 3	$\begin{aligned} & \text{KM algorithm for } r_{\tilde{B}}, \ r_{\tilde{B}} = \max_{\forall \theta_i \in [\underline{\mu}_{\tilde{B}}(y_i), \overline{\mu}_{\tilde{B}}(y_i)]} (\sum_{i=1}^N y_i \theta_i) / (\sum_{i=1}^N \theta_i) \\ & \text{Set } \theta(i) = [\underline{\mu}_{\tilde{B}}(y_i) + \overline{\mu}_{\tilde{B}}(y_i)] / 2, i = 1, \Lambda, N \text{ and compute} \\ & c' = (\sum_{i=1}^N y_i \theta_i) / (\sum_{i=1}^N \theta_i) . \\ & \text{Find } k'(1 \le k' \le N - 1) \text{ such that } y_{k'} \le c' < y_{k'+1}. \\ & \text{Set } \theta_i = \underline{\mu}_{\tilde{B}}(y_i) \text{ when } i \le k', \text{ and } \theta_i = \overline{\mu}_{\tilde{B}}(y_i) \text{ when } i > k', \end{aligned}$
Step 1 2 3	$\begin{split} & \text{KM algorithm for } r_{\tilde{B}}, \ r_{\tilde{B}} = \max_{\forall \theta_i \in [\underline{\mu}_{\tilde{B}}(y_i), \mu_{\tilde{B}}(y_i)]} (\sum_{i=1}^N y_i \theta_i) / (\sum_{i=1}^N \theta_i) \\ & \text{Set } \theta(i) = [\underline{\mu}_{\tilde{B}}(y_i) + \overline{\mu}_{\tilde{B}}(y_i)] / 2, i = 1, \Lambda, N \text{ and compute} \\ & c' = (\sum_{i=1}^N y_i \theta_i) / (\sum_{i=1}^N \theta_i) . \\ & \text{Find } k'(1 \le k' \le N - 1) \text{ such that } y_{k'} \le c' < y_{k'+1}. \\ & \text{Set } \theta_i = \underline{\mu}_{\tilde{B}}(y_i) \text{ when } i \le k', \text{ and } \theta_i = \overline{\mu}_{\tilde{B}}(y_i) \text{ when } i > k', \\ & \text{ and compute } r_{\tilde{B}}(k') = (\sum_{i=1}^N y_i \theta_i) / (\sum_{i=1}^N \theta_i) . \end{split}$
Step 1 2 3 4	$\begin{array}{l} \text{KM algorithm for } r_{\tilde{B}}, \ r_{\tilde{B}} = \max_{\forall \theta_i \in [\underline{\mu}_{\tilde{B}}(y_i), \mu_{\tilde{B}}(y_i)]} (\sum_{i=1}^N y_i \theta_i) / (\sum_{i=1}^N \theta_i) \\ \text{Set } \theta(i) = [\underline{\mu}_{\tilde{B}}(y_i) + \overline{\mu}_{\tilde{B}}(y_i)] / 2, i = 1, \Lambda, N \text{ and compute} \\ c' = (\sum_{i=1}^N y_i \theta_i) / (\sum_{i=1}^N \theta_i) . \\ \text{Find } k'(1 \le k' \le N - 1) \text{ such that } y_{k'} \le c' < y_{k'+1}. \\ \text{Set } \theta_i = \underline{\mu}_{\tilde{B}}(y_i) \text{ when } i \le k', \text{ and } \theta_i = \overline{\mu}_{\tilde{B}}(y_i) \text{ when } i > k', \\ \text{and compute } r_{\tilde{B}}(k') = (\sum_{i=1}^N y_i \theta_i) / (\sum_{i=1}^N \theta_i) . \\ \text{Check if } r_{\tilde{B}}(k') = c', \text{ if yes, stop and set } r_{\tilde{B}}(k') = r_{\tilde{B}} \text{ and } k' = R, \text{ if no, go} \end{array}$
Step 1 2 3 4	KM algorithm for $r_{\tilde{B}}$, $r_{\tilde{B}} = \max_{\forall \theta_i \in [\underline{\mu}_{\tilde{B}}(y_i), \mu_{\tilde{B}}(y_i)]} (\sum_{i=1}^N y_i \theta_i) / (\sum_{i=1}^N \theta_i)$ Set $\theta(i) = [\underline{\mu}_{\tilde{B}}(y_i) + \overline{\mu}_{\tilde{B}}(y_i)] / 2, i = 1, \Lambda, N$ and compute $c' = (\sum_{i=1}^N y_i \theta_i) / (\sum_{i=1}^N \theta_i)$. Find $k'(1 \le k' \le N - 1)$ such that $y_{k'} \le c' < y_{k'+1}$. Set $\theta_i = \underline{\mu}_{\tilde{B}}(y_i)$ when $i \le k'$, and $\theta_i = \overline{\mu}_{\tilde{B}}(y_i)$ when $i > k'$, and compute $r_{\tilde{B}}(k') = (\sum_{i=1}^N y_i \theta_i) / (\sum_{i=1}^N \theta_i)$. Check if $r_{\tilde{B}}(k') = c'$, if yes, stop and set $r_{\tilde{B}}(k') = r_{\tilde{B}}$ and $k' = R$, if no, go to Step 5.

Step	EKM algorithm for $l_{\tilde{B}}$
1	Set $k = [N/2.4]$ (the nearest integer to $N/2.4$) and compute
	$\alpha = \sum_{i=1}^{k} y_i \overline{\mu}_{\widetilde{B}}(y_i) + \sum_{i=k+1}^{N} y_i \underline{\mu}_{\widetilde{B}}(y_i),$
	$\beta = \sum_{i=1}^{k} \overline{\mu}_{\widetilde{B}}(y_i) + \sum_{i=k+1}^{N} \underline{\mu}_{\widetilde{B}}(y_i), c' = \alpha / \beta.$
2	Find $k'(1 \le k' < N-1)$ such that $y_{k'} \le c' < y_{k'+1}$.
3	Check if $k' = k$, if yes, stop and set $c' = l_{\tilde{B}}$ and $k = L$, if no, go to Step 4.
4	Compute $s = sign(k'-k)$, $\alpha' = \alpha + s \sum_{i=\min(k,k')+1}^{\max(k,k')} [\overline{\mu}_{\widetilde{B}}(y_i) - \underline{\mu}_{\widetilde{B}}(y_i)],$
	$\beta' = \beta + s \sum_{i=\min(k,k')+1}^{\max(k,k')} [\overline{\mu}_{\widetilde{B}}(y_i) - \underline{\mu}_{\widetilde{B}}(y_i)], \ c'' = \alpha' / \beta'.$
5	Set $c' = c'', \alpha = \alpha', \beta = \beta', k = k'$ and go to Step 2.
5	,
Step	EKM algorithm for $r_{\tilde{B}}$
Step 1	EKM algorithm for $r_{\tilde{B}}$ Set $k = [N/1.7]$ and compute $\alpha = \sum_{i=1}^{k} y_i \underline{\mu}_{\tilde{B}}(y_i) + \sum_{i=k+1}^{N} y_i \overline{\mu}_{\tilde{B}}(y_i)$,
Step 1	EKM algorithm for $r_{\tilde{B}}$ Set $k = [N/1.7]$ and compute $\alpha = \sum_{i=1}^{k} y_i \underline{\mu}_{\tilde{B}}(y_i) + \sum_{i=k+1}^{N} y_i \overline{\mu}_{\tilde{B}}(y_i)$, $\beta = \sum_{i=1}^{k} \underline{\mu}_{\tilde{B}}(y_i) + \sum_{i=k+1}^{N} \overline{\mu}_{\tilde{B}}(y_i)$, $c' = \alpha / \beta$.
Step 1 2 2	EKM algorithm for $r_{\tilde{B}}$ Set $k = [N/1.7]$ and compute $\alpha = \sum_{i=1}^{k} y_i \underline{\mu}_{\tilde{B}}(y_i) + \sum_{i=k+1}^{N} y_i \overline{\mu}_{\tilde{B}}(y_i)$, $\beta = \sum_{i=1}^{k} \underline{\mu}_{\tilde{B}}(y_i) + \sum_{i=k+1}^{N} \overline{\mu}_{\tilde{B}}(y_i)$, $c' = \alpha / \beta$. Find $k'(1 \le k' < N - 1)$ such that $y_{k'} \le c' < y_{k'+1}$.
Step 1 2 3	EKM algorithm for $r_{\tilde{B}}$ Set $k = [N/1.7]$ and compute $\alpha = \sum_{i=1}^{k} y_i \underline{\mu}_{\tilde{B}}(y_i) + \sum_{i=k+1}^{N} y_i \overline{\mu}_{\tilde{B}}(y_i)$, $\beta = \sum_{i=1}^{k} \underline{\mu}_{\tilde{B}}(y_i) + \sum_{i=k+1}^{N} \overline{\mu}_{\tilde{B}}(y_i)$, $c' = \alpha / \beta$. Find $k'(1 \le k' < N - 1)$ such that $y_{k'} \le c' < y_{k'+1}$. Check if $k' = k$, if yes, stop and set $c' = r_{\tilde{B}}$ and $k = R$, if no, go to Step 4.
Step 1 2 3 4	EKM algorithm for $r_{\tilde{B}}$ Set $k = [N/1.7]$ and compute $\alpha = \sum_{i=1}^{k} y_i \underline{\mu}_{\tilde{B}}(y_i) + \sum_{i=k+1}^{N} y_i \overline{\mu}_{\tilde{B}}(y_i)$, $\beta = \sum_{i=1}^{k} \underline{\mu}_{\tilde{B}}(y_i) + \sum_{i=k+1}^{N} \overline{\mu}_{\tilde{B}}(y_i)$, $c' = \alpha / \beta$. Find $k'(1 \le k' < N - 1)$ such that $y_{k'} \le c' < y_{k'+1}$. Check if $k' = k$, if yes, stop and set $c' = r_{\tilde{B}}$ and $k = R$, if no, go to Step 4. Compute $s = sign(k' - k)$, $\alpha' = \alpha - s \sum_{i=\min(k,k')+1}^{\max(k,k')} [\overline{\mu}_{\tilde{B}}(y_i) - \underline{\mu}_{\tilde{B}}(y_i)]$,
Step 1 2 3 4 4	EKM algorithm for $r_{\tilde{B}}$ Set $k = [N/1.7]$ and compute $\alpha = \sum_{i=1}^{k} y_i \underline{\mu}_{\tilde{B}}(y_i) + \sum_{i=k+1}^{N} y_i \overline{\mu}_{\tilde{B}}(y_i)$, $\beta = \sum_{i=1}^{k} \underline{\mu}_{\tilde{B}}(y_i) + \sum_{i=k+1}^{N} \overline{\mu}_{\tilde{B}}(y_i)$, $c' = \alpha / \beta$. Find $k'(1 \le k' < N - 1)$ such that $y_{k'} \le c' < y_{k'+1}$. Check if $k' = k$, if yes, stop and set $c' = r_{\tilde{B}}$ and $k = R$, if no, go to Step 4. Compute $s = sign(k'-k)$, $\alpha' = \alpha - s \sum_{i=\min(k,k')+1}^{\max(k,k')} [\overline{\mu}_{\tilde{B}}(y_i) - \underline{\mu}_{\tilde{B}}(y_i)]$, $\beta' = \beta - s \sum_{i=\min(k,k')+1}^{\max(k,k')} [\overline{\mu}_{\tilde{B}}(y_i) - \underline{\mu}_{\tilde{B}}(y_i)]$, $c'' = \alpha' / \beta'$.
Step 1 2 3 4 5	EKM algorithm for $r_{\tilde{B}}$ Set $k = [N/1.7]$ and compute $\alpha = \sum_{i=1}^{k} y_i \underline{\mu}_{\tilde{B}}(y_i) + \sum_{i=k+1}^{N} y_i \overline{\mu}_{\tilde{B}}(y_i)$, $\beta = \sum_{i=1}^{k} \underline{\mu}_{\tilde{B}}(y_i) + \sum_{i=k+1}^{N} \overline{\mu}_{\tilde{B}}(y_i)$, $c' = \alpha/\beta$. Find $k'(1 \le k' < N - 1)$ such that $y_{k'} \le c' < y_{k'+1}$. Check if $k' = k$, if yes, stop and set $c' = r_{\tilde{B}}$ and $k = R$, if no, go to Step 4. Compute $s = sign(k'-k)$, $\alpha' = \alpha - s \sum_{i=\min(k,k')+1}^{\max(k,k')} [\overline{\mu}_{\tilde{B}}(y_i) - \underline{\mu}_{\tilde{B}}(y_i)]$, $\beta' = \beta - s \sum_{i=\min(k,k')+1}^{\max(k,k')} [\overline{\mu}_{\tilde{B}}(y_i) - \underline{\mu}_{\tilde{B}}(y_i)]$, $c'' = \alpha'/\beta'$. Set $c' = c'', \alpha = \alpha', \beta = \beta', k = k'$ and go to Step 2.

Table 4. EKM algorithms [18,19,21,25] for computing the centroid of \tilde{B} .

Setting $F'_l(\xi) = 0$, therefore

$$\frac{(b-\xi)^2}{(\xi-a)^2} = \rho \,, \ \frac{b-\xi}{\xi-a} = \sqrt{\rho} \,.$$
(28)

Solving Eq (28) for ξ , we can obtain

$$\xi_l = \frac{b + a\sqrt{\rho}}{1 + \sqrt{\rho}} = a + \frac{b - a}{1 + \sqrt{\rho}}.$$
(29)

For the Eq (27), ξ_l is the minimum value of $F_l(\xi)$. Because $F'_l(\xi) < 0$ for $\xi \in [a, \xi_l)$, and $F'_l(\xi) > 0$ for $\xi \in (\xi_l, b]$. Therefore, $l_{\tilde{B}} = F_l(\xi_l) = \xi_l$.

In like manners, it can be obtained:

AIMS Mathematics

$$F_r(\xi) = \frac{(\xi^2 - a^2) + \rho(b^2 - \xi^2)}{2[(\xi - a) + \rho(b - \xi)]}$$
(30)

Therefore,

$$F'_{r}(\xi) = -\frac{(\rho - 1)[(\xi - a)^{2} - \rho(b - \xi)^{2}]}{2[(\xi - a) + \rho(b - \xi)]^{2}}.$$
(31)

Setting $F'_r(\xi) = 0$, it follows that:

$$\frac{(b-\xi)^2}{(\xi-a)^2} = \frac{1}{\rho}, \quad \frac{b-\xi}{\xi-a} = \sqrt{1/\rho}.$$
(32)

Solving Eq (30) for ξ , we can obtain

$$\xi_r = \frac{b + a\sqrt{1/\rho}}{1 + \sqrt{1/\rho}} = a + \frac{b - a}{1 + \sqrt{1/\rho}}.$$
(33)

For the Eq (30), ξ_r is the maximum value of $F_r(\xi)$. Because $F'_r(\xi) > 0$ for $\xi \in [a, \xi_r)$, and $F'_r(\xi) < 0$ for $\xi \in (\xi_r, b]$. Therefore, $r_{\tilde{B}} = F_r(\xi_r) = \xi_r$. Considering the Eqs (29) and (33) together, the new initialization approach for ξ can be denoted as $\xi^{(2)}$, i.e.,

$$\xi^{(2)} = \begin{cases} a + \frac{b-a}{1+\sqrt{\rho}} & \text{for } l_{\tilde{B}}, \\ a + \frac{b-a}{1+\sqrt{1/\rho}} & \text{for } r_{\tilde{B}}. \end{cases}$$
(34)

Table 5. RIEKM algorithms for computing the centroid of \widetilde{B} .

1 Set $k = [N/(1+\sqrt{\rho})]$ $(\rho = [\sum_{i=1}^{N} \overline{\mu}_{\widetilde{B}}(y)]/[\sum_{i=1}^{N} \underline{\mu}_{\widetilde{B}}(y)]$, which is dep	end on
the specific examples) and compute	
$\alpha = \sum_{i=1}^{k} y_i \overline{\mu}_{\widetilde{B}}(y_i) + \sum_{i=k+1}^{N} y_i \underline{\mu}_{\widetilde{B}}(y_i), \beta = \sum_{i=1}^{k} \overline{\mu}_{\widetilde{B}}(y_i) + \sum_{i=k+1}^{N} \underline{\mu}_{\widetilde{B}}(y_i), c' = \sum_{i=1}^{k} y_i \overline{\mu}_{\widetilde{B}}(y_i) + \sum_{i=k+1}^{N} y_i \underline{\mu}_{\widetilde{B}}(y_i), \beta = \sum_{i=1}^{k} y_i \overline{\mu}_{\widetilde{B}}(y_i) + \sum_{i=k+1}^{N} y_i \underline{\mu}_{\widetilde{B}}(y_i), \beta = \sum_{i=1}^{k} y_i \overline{\mu}_{\widetilde{B}}(y_i) + \sum_{i=k+1}^{N} y_i \underline{\mu}_{\widetilde{B}}(y_i), \beta = \sum_{i=1}^{k} y_i \overline{\mu}_{\widetilde{B}}(y_i) + \sum_{i=k+1}^{N} y_i \underline{\mu}_{\widetilde{B}}(y_i), \beta = \sum_{i=1}^{k} y_i \overline{\mu}_{\widetilde{B}}(y_i) + \sum_{i=k+1}^{N} y_i \underline{\mu}_{\widetilde{B}}(y_i), \beta = \sum_{i=1}^{k} y_i \overline{\mu}_{\widetilde{B}}(y_i) + \sum_{i=k+1}^{N} y_i \underline{\mu}_{\widetilde{B}}(y_i), \beta = \sum_{i=1}^{k} y_i \overline{\mu}_{\widetilde{B}}(y_i) + \sum_{i=k+1}^{N} y_i \underline{\mu}_{\widetilde{B}}(y_i), \beta = \sum_{i=1}^{k} y_i \overline{\mu}_{\widetilde{B}}(y_i) + \sum_{i=k+1}^{N} y_i \underline{\mu}_{\widetilde{B}}(y_i), \beta = \sum_{i=1}^{k} y_i \overline{\mu}_{\widetilde{B}}(y_i) + \sum_{i=k+1}^{N} y_i \underline{\mu}_{\widetilde{B}}(y_i), \beta = \sum_{i=1}^{k} y_i \overline{\mu}_{\widetilde{B}}(y_i) + \sum_{i=k+1}^{N} y_i \underline{\mu}_{\widetilde{B}}(y_i), \beta = \sum_{i=1}^{k} y_i \overline{\mu}_{\widetilde{B}}(y_i) + \sum_{i=k+1}^{N} y_i \underline{\mu}_{\widetilde{B}}(y_i) + \sum_$	α/β .
2 Find $k'(1 \le k' < N-1)$ such that $y_{k'} \le c' < y_{k'+1}$.	
3 Check if $k' = k$, if yes, stop and set $c' = l_{\tilde{B}}$ and $k = L$, if no, go to S	tep 4.
4 Compute $s = sign(k'-k)$, $\alpha' = \alpha + s \sum_{i=\min(k,k')+1}^{\max(k,k')} [\overline{\mu}_{\widetilde{B}}(y_i) - \underline{\mu}_{\widetilde{B}}(y_i)],$	
$\beta' = \beta + s \sum_{i=\min(k,k')+1}^{\max(k,k')} [\overline{\mu}_{\widetilde{B}}(y_i) - \underline{\mu}_{\widetilde{B}}(y_i)], \ c'' = \alpha' / \beta'.$	
5 Set $c' = c'', \alpha = \alpha', \beta = \beta', k = k'$ and return to Step 2.	

Continued on next page

Step	RIEKM algorithm for $r_{\tilde{B}}$
1	Set $k = [N/(1 + \sqrt{1/\rho})]$ and compute $\alpha = \sum_{i=1}^{k} y_i \underline{\mu}_{\tilde{B}}(y_i) + \sum_{i=k+1}^{N} y_i \overline{\mu}_{\tilde{B}}(y_i)$,
	$\beta = \sum_{i=1}^{k} \underline{\mu}_{\tilde{B}}(y_i) + \sum_{i=k+1}^{N} \overline{\mu}_{\tilde{B}}(y_i), \ c' = \alpha / \beta$
2	Find $k'(1 \le k' < N-1)$ such that $y_{k'} \le c' < y_{k'+1}$.
3	Check if $k' = k$, if yes, stop and set $c' = r_{\tilde{B}}$ and $k = R$, if no, go to Step 4.
4	Compute $s = sign(k'-k)$, $\alpha' = \alpha - s \sum_{i=\min(k,k')+1}^{\max(k,k')} [\overline{\mu}_{\widetilde{B}}(y_i) - \underline{\mu}_{\widetilde{B}}(y_i)]$,
	$\beta' = \beta - s \sum_{i=\min(k,k')+1}^{\max(k,k')} [\overline{\mu}_{\widetilde{B}}(y_i) - \underline{\mu}_{\widetilde{B}}(y_i)], \ c'' = \alpha' / \beta'.$
5	Set $c' = c'', \alpha = \alpha', \beta = \beta', k = k'$ and return to Step 2.

Here $\rho \ge 1$, therefore, $\xi^{(2)} \le a + \frac{1}{2}(b-a)$ for $l_{\tilde{B}}$, and $\xi^{(2)} \ge a + \frac{1}{2}(b-a)$ for $r_{\tilde{B}}$.

Comparing the initializations of EKM algorithms in Table 4 and CEKM algorithms in Table 2, the discrete form of Eq (34) can be as:

$$k^{(2)} = \begin{cases} [N/(1+\sqrt{\rho})] & \text{for } l_{\tilde{B}}, \\ [N/(1+\sqrt{1/\rho})] & \text{for } r_{\tilde{B}}. \end{cases}$$
(35)

where *N* denotes the number of sampling of primary variable, and this $\rho = \left[\sum_{i=1}^{N} \overline{\mu}_{\tilde{B}}(y)\right] / \left[\sum_{i=1}^{N} \underline{\mu}_{\tilde{B}}(y)\right].$

Interestingly, when $\rho = 2$, it can be obtained that $1 + \sqrt{\rho} = 1 + \sqrt{2} \approx 2.4$, and $1 + \sqrt{1/\rho} = 1 + \sqrt{1/2} \approx 1.7$, therefore, the equation (35) becomes:

$$k^{(2)} = \begin{cases} [N/2.4] & \text{for } l_{\tilde{B}}, \\ [N/1.7] & \text{for } r_{\tilde{B}}. \end{cases}$$
(36)

Equation (36) is just the initialization approach of EKM algorithms, which generates from empirical extensive simulation experiments. For the proposed reasonable initialization EKM (RIEKM) algorithms, we should determine each specific ρ for the corresponding simulations. Finally the specific computation steps of discrete RIEKM algorithms are provided in Table 5.

4. Simulation experiments

Four computer simulation examples are provided in this section to illustrate the performances RIEKM algorithms. Before performing the centroid TR, suppose that the FOU of centroid output IT2 FS be known by weighting and aggregating all the fuzzy rules under the guidance of inference. Here the primary variable of centroid output IT2 FS is denoted by the letter x, x is uniformly sampled, and the number of sampling is chosen as N = 50:50:9000.

In the first example, the FOU is bounded by the piecewise linear functions [21-23,25-26]. In the second example, the FOU is bounded by both the Gaussian functions and piecewise linear functions [20-22,27-31]. In the third example, the FOU is bounded by the Gaussian functions [21-23,25-26]. In the last example, the FOU is a Gaussian IT2 MF with uncertainty standard deviation [20-22,27-31]. Then the Figure 2 and Table 6 provide the defined FOUs for four examples.



Figure 2. The graphs of FOUs, (a) example 1; (b) example 2; (c) example 3, and (d) example 4.

In examples 1, 2 and 4, let the primary variable $x \in [0, 10]$. In addition, let the primary variable $x \in [-5, 14]$ in example 2. The accurate continuous Nie-Tan (CNT) algorithms [22,25] compute the centroid output of IT2 FS \tilde{A} as:

$$y_{CNT} = \frac{\int_{a}^{b} x[\underline{\mu}_{\tilde{A}}(x) + \overline{\mu}_{\tilde{A}}(x)]dx}{\int_{a}^{b} [\underline{\mu}_{\tilde{A}}(x) + \overline{\mu}_{\tilde{A}}(x)]dx}.$$
(37)

Num	Expressions		
1	$\underline{\mu}_{\tilde{A}_{1}}(x) = \max\left\{ \begin{bmatrix} \frac{x-1}{6}, & 1 \le x \le 4 \\ \frac{7-x}{6}, & 4 < x \le 7 \\ 0, & \text{otherwise} \end{bmatrix}, \begin{bmatrix} \frac{x-3}{6}, & 3 \le x \le 5 \\ \frac{8-x}{9}, & 5 < x \le 8 \\ 0, & \text{otherwise} \end{bmatrix} \right\}$		
	$\overline{\mu}_{\widetilde{A}_{1}}(x) = \max\left\{ \begin{bmatrix} \frac{x-1}{2}, & 1 \le x \le 3\\ \frac{7-x}{4}, & 3 < x \le 7\\ 0, & \text{otherwise} \end{bmatrix}, \begin{bmatrix} \frac{x-2}{5}, & 2 \le x \le 6\\ \frac{16-2x}{5}, & 6 < x \le 8\\ 0, & \text{otherwise} \end{bmatrix} \right\}$		
2	$\underline{\mu}_{\tilde{A}_{2}}(x) = \begin{cases} \frac{0.6(x+5)}{19}, -5 \le x \le 2.6\\ \frac{0.4(14-x)}{19}, 2.6 < x \le 14 \end{cases} \overline{\mu}_{\tilde{A}_{2}}(x) = \begin{cases} \exp[-\frac{1}{2}(\frac{x-2}{5})^{2}], -5 \le x \le 7.185\\ \exp[-\frac{1}{2}(\frac{x-9}{1.75})^{2}], 7.185 < x \le 14 \end{cases}$		
3	$\underline{\mu}_{\tilde{A}_3}(x) = \max\{0.5 \exp[-\frac{(x-3)^2}{2}], 0.4 \exp[-\frac{(x-6)^2}{2}]\}$ $\overline{\mu}_{\tilde{A}_3}(x) = \max\{\exp[-0.5\frac{(x-3)^2}{4}], 0.8 \exp[-0.5\frac{(x-6)^2}{4}]\}$		
4	$\underline{\mu}_{\tilde{A}_4}(x) = \exp[-\frac{1}{2}(\frac{x-3}{0.25})^2], \ \overline{\mu}_{\tilde{A}_4}(x) = \exp[-\frac{1}{2}(\frac{x-3}{1.75})^2]$		

Table 6. MF expressions for FOUs.

Therefore, the CNT algorithms are firstly considered as the benchmark to compute the defuzzified values for four examples as: $y_1^* = 4.320794$, $y_2^* = 3.714087$, $y_3^* = 4.395260$, and $y_4^* = 4.999999$. Then graphs of defuzzified values computed by two types of discrete EKM algorithms are shown in Figure 3.

Furthermore, the absolute errors between the benchmark CNT algorithms and two types of discrete EKM algorithms are provided in Figure 4.



Figure 3. The graphs of defuzzifed values computed by the EKM and RIEKM algorithms, (a) example 1; (b) example 2; (c) example 3, and (d) example 4.



Figure 4. The functional graphs of absolute errors between the CNT algorithms and two types of discrete EKM algorithms, (a) example 1; (b) example 2; (c) example 3, and (d) example 4.

In order to further measure the performances of two types of algorithms, here we define the absolute errors sum $\sum_{i=1}^{180} |y_{CNT_i}^* - y_{EKM_i(RIEKM_i)}|$ of defuzzified values for four examples, and they are shown in Table 7, in which the last column denotes the total average of absolute errors sum for the EKM algorithms and RIEKM algorithms.

Table 7. The absolute errors sum of defuzzified values for the CNT algorithms and two types of discrete EKM algorithms.

Algorithms	EKM	RIEKM
Example 1	0.932160	0.889015
Example 2	15.595056	14.933244
Example 3	3.577131	3.546304
Example 4	0.027159	0.000251
Total average	5.032877	4.842204

Observing from the Figures 3,4 and the Table 7, the following conclusions can be made:

1) As the number of sampling increases, these two types of discrete EKM algorithms all converge to certain values. The absolute errors of RIEKM algorithms are always less than the EKM algorithms in these four examples (as the red error curves of RIEKM algorithms are under the corresponding blue error curves of EKM algorithms).

2) In example 1, the RIEKM algorithms can obtain the result that is closed to the benchmark value with the number of sampling of 150. In other three examples, the RIEKM algorithms can almost get the optimal value just at the number of sampling of 50 (starting number of sampling).

3) As N = 50:50:9000, the RIEKM algorithms can obtain smaller absolute errors sum than the EKM algorithms in all four examples. In addition, the total average absolute errors sum of RIEKM algorithms is less than the EKM algorithms.

4) According to the items one to three, it is evidence that the proposed RIEKM algorithms can obtain better calculation accuracy than the EKM algorithms.

For the sake of applying these algorithms, next we study the specific unrepeatable computation times that depend on the hardware and software environments. Here the computer simulation platform is a dual CPU desktop with the Microsoft Windows XP operating system, E5300@2.6GHz and 2.00GB memory. All the programs are performed by the Matlab 2013a. Then Figure 5 gives the computation times for four examples, and the unit of time is the second (s).



Figure 5. The computation times of EKM and RIEKM algorithms, (a) example 1; (b) example 2; (c) example 3, and (d) example 4.

As shown in Figure 5, except for some fluctuations, the computation times of two types of discrete EKM algorithms emerge linear variation. In most number of sampling, the computation times of EKM algorithms are slightly better than the RIEKM algorithms. In other words, the convergence speeds of RIEKM algorithms are faster than the EKM algorithms in general. When the number of sampling is fixed, the computation times of two types of EKM algorithms are as: RIEKM>EKM for all four examples. This may because the initialization of RIEKM algorithms is more complex than the simple EKM algorithms.

Here we should point out that the paper focuses on the theoretical performances of RIEKM algorithms compared with the EKM algorithms. Four computer simulation examples show the advantages of RIEKM algorithms on high computation accuracy requirement. However, if the requirement of calculation accuracy is not very high, the simple EKM algorithms will complete with slightly faster computation speeds.

5. Conclusions

This paper gives the fuzzy reasoning process of IT2 FLSs, and discusses the initializations of EKM algorithms. And the reasonable initialization EKM (RIEKM) algorithms are provided to perform the centroid TR and defuzzification of IT2 FLSs. For computing the defuzzified values of

IT2 FLSs, the proposed RIEKM algorithms can obtain better absolute errors and faster convergence speeds than the EKM algorithms.

Many interesting works still lie ahead, including extending and weighting the RIEKM algorithms to perform the centroid TR [17,27,28,31,32,40–44,49,50] of general type-2 fuzzy logic systems, and studying the center-of-sets TR of T2 FLSs [33], and the relations between discrete and continuous TR algorithms [18,21,28–30]. Future studies will also be focused on designing and applying IT2 or GT2 FLSs based on intelligent optimization algorithms [3,7–12,34–39,45–48,53] for forecasting, control [51,52] and identification.

Acknowledgments

The paper is sponsored by the National Natural Science Foundation of China (No. 61973146, No. 61773188, No. 61903167, No. 61803189), the Liaoning Province Natural Science Foundation Guidance Project (No. 20180550056), and Talent Fund Project of Liaoning University of Technology (No. xr2020002). The author is very grateful for professor Jerry Mendel, who has given the author some valuable suggestions.

Conflict of interest

The authors declare that they have no conflict of interest.

References

- 1. D. R. Wu, J. M. Mendel, Uncertainty measures for interval type-2 fuzzy sets, Inf. Sci., 177 (2007), 5378–5393.
- 2. J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*, Englewood Cliffs, NJ, USA: Prentice-Hall, 2001, 1–547.
- 3. P. Melin, L. Astudillo, O. Castillo, et al. *Optimal design of type-2 and type-1 fuzzy tracking controllers for autonomous mobile robots under perturbed torques using a new chemical optimization paradigm*, Expert Syst. Appl., **40** (2013), 3185–3195.
- 4. H. Hagras, *A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots*, IEEE Trans. Fuzzy Syst., **12** (2004), 524–539.
- 5. C. W. Tao, J. S. Taur, C. W. Chang, et al. *Simplified type-2 fuzzy sliding controller for wing rocket system*, Fuzzy Sets Syst., **207** (2012), 111–129.
- 6. D. Bernardo, H. Hagras, E. Tsang, A genetic type-2 fuzzy logic based system for the generation of summarized linguistic predictive models for financial applications, Soft Comput., **17** (2013), 2185–2201.
- Y. Chen, D. Z. Wang, S. C. Tong, Forecasting studies by designing Mamdani interval type-2 fuzzy logic systems: With combination of BP algorithms and KM algorithms, Neurocomputing, 174 (2016), 1133–1146.
- 8. A. Khosravi, S. Nahavandi, D. Creighton, et al., *Interval type-2 fuzzy logic systems for load forecasting: a comparative study*, IEEE Trans. Power Syst., **27** (2012), 1274–1282.
- 9. S. Barkat, A. Tlemcani, H. Nouri, *Noninteracting adaptive control of PMSM using interval type-2 fuzzy logic systems*, IEEE Trans. Fuzzy Syst., **19** (2011), 925–936.

- 10. D. Z. Wang, Y. Chen, Study on permanent magnetic drive forecasting by designing Takagi Sugeno Kang type interval type-2 fuzzy logic systems, Trans. Institute Meas. Control, **40** (2018), 2011–2023.
- Y. Chen, D. Z. Wang, Forecasting by designing Mamdani general type-2 fuzzy logic systems optimized with quantum particle swarm optimization algorithms, Trans. Institute Meas. Control, 41 (2019), 2886–2896.
- 12. P. Melin, O. Mendoza, O. Castillo, *An improved method for edge detection based on interval type-2 fuzzy logic*, Expert Syst. Appl., **37** (2010), 8527–8535.
- 13. C. S. Lee, M. H. Wang, H. Hagras, *Type-2 fuzzy ontology and its application to personal diabetic-diet recommendation*, IEEE Trans. Fuzzy Syst., **18** (2010), 316–328.
- 14. G. M. Méndez, M. D. L. A. Hernandez, *Hybrid learning for interval type-2 fuzzy logic systems based on orthogonal least-squares and back-propagation methods*, Inf. Sci., **179** (2009), 2146–2157.
- 15. G. M. Méndez, M. D. L. A. Hernandez, *Hybrid learning mechanism for interval A2-C1 type-2 non-singleton type-2 Takagi-Sugeno-Kang fuzzy logic systems*, Inf. Sci., **220** (2013), 149–169.
- 16. T. Wang, Y. Chen, S. C. Tong, *Fuzzy reasoning models and algorithms on type-2 fuzzy sets*, Int. J. Innovative Comput. Inf. Control, **4** (2008), 2451–2460.
- 17. J. M. Mendel, *General type-2 fuzzy logic systems made simple: A tutorial*, IEEE Trans. Fuzzy Sys., **22** (2014), 1162–1182.
- J. M. Mendel, On KM algorithms for solving type-2 fuzzy set problems, IEEE Trans. Fuzzy Syst., 21 (2013), 426–446.
- 19. D. R. Wu, J. M. Mendel, *Enhanced Karnik-Mendel algorithms*, IEEE Trans. Fuzzy Syst., **17** (2009), 923–934.
- J. M. Mendel, F. L. Liu, Super-exponential convergence of the Karnik-Mendel algorithms for computing the centroid of an interval type-2 fuzzy set, IEEE Trans. Fuzzy Syst., 15 (2007), 309–320.
- X. W. Liu, J. M. Mendel, D. R. Wu, Study on enhanced Karnik-Mendel algorithms: Initialization explanations and computation improvements, Inf. Sci., 184 (2012), 75–91.
- J. W. Li, R. John, S. Coupland, et al., On Nie-Tan operator and type-reduction of interval type-2 fuzzy sets, IEEE Trans. Fuzzy Syst., 26 (2018), 1036–1039.
- 23. Y. Chen, Study on weighted Nagar-Bardini algorithms for centroid type-reduction of interval type-2 fuzzy logic systems, J. Intell. Fuzzy Syst., **34** (2018), 2417–2428.
- 24. J. M. Mendel, R. I. John, F. L. Liu, *Interval type-2 fuzzy logic systems made simple*, IEEE Trans. Fuzzy Syst., **14** (2006), 808–821.
- 25. Y. Chen, D. Z. Wang, Study on centroid type-reduction of general type-2 fuzzy logic systems with weighted Nie-Tan algorithms, Soft Comput., **22** (2018), 7659–7678.
- F. L. Liu, An efficient centroid type-reduction strategy for general type-2 fuzzy logic system, Inf. Sci., 178 (2008), 2224–2236.
- J. M. Mendel, X. W. Liu, Simplified interval type-2 fuzzy logic systems, IEEE Trans. Fuzzy Syst., 21 (2013), 1056–1069.
- 28. S. Greenfield, F. Chiclana, Accuracy and complexity evaluation of defuzzification strategies for the discretised interval type-2 fuzzy set, Int. J. Approximate Reasoning, **54** (2013), 1013–1033.
- 29. Y. Chen, Study on centroid type-reduction of interval type-2 fuzzy logic systems based on noniterative algorithms, Complexity, **2019** (2019), 1–12.
- 30. T. Kumbasar, *Revisiting Karnik-Mendel algorithms in the framework of linear fractional programming*, Int. J. Approximate Reasoning, **82** (2017), 1–21.

- 31. S. Greenfield, F. Chiclana, S. Coupland, et al., *The collapsing method of defuzzification for discretised interval type-2 fuzzy sets*, Inf. Sci., **179** (2009), 2055–2069.
- 32. D. R. Wu, Approaches for reducing the computational cost of interval type-2 fuzzy logic systems: overview and comparisons, IEEE Trans. Fuzzy Syst., **21** (2013), 80–99.
- M. A. Khanesar, A. Jalalian, O. Kaynak, Improving the speed of center of set type-reduction in interval type-2 fuzzy systems by eliminating the need for sorting, IEEE Trans. Fuzzy Syst., 25 (2017), 1193–1206.
- 34. Y. Chen, D. Z. Wang, W. Ning, *Forecasting by TSK general type-2 fuzzy logic systems optimized with genetic algorithms*, Optimal Control Appl. Methods, **39** (2018), 393–409.
- 35. Y. Chen, D. Z. Wang, *Forecasting by general type-2 fuzzy logic systems optimized with QPSO algorithms*, Int. J. Control, Automation Syst., **15** (2017), 2950–2958.
- 36. F. Gaxiola, P. Melin, F. Valdez, et al. *Optimization of type-2 fuzzy weights in backpropagation learning for neural networks using GAs and PSO*, Appl. Soft Comput., **38** (2016), 860–871.
- 37. Q. F. Fan, T. Wang, Y. Chen, et al., *Design and application of interval type-2 TSK fuzzy logic system based on QPSO algorithm*, Int. J. Fuzzy Syst., **20** (2018), 835–846.
- 38. C. H. Hsu, C. F. Juang, *Evolutionary robot wall-following control using type- 2 fuzzy controller with species-de-activated continuous ACO*, IEEE Trans. Fuzzy Syst., **21** (2013), 100–112.
- D. R. Wu, J. M. Mendel, *Recommendations on designing practical interval type-2 fuzzy systems*, Eng. Appl. Artif. Intell., 85 (2019), 182–193.
- 40. X. L. Liu, S. P. Wan, *Combinatorial iterative algorithms for computing the centroid of an interval type-2 fuzzy set*, IEEE Trans. Fuzzy Syst., 2019, DOI: 10.1109/TFUZZ.2019.2911918.
- 41. H. Z. Hu, Y. Wang, Y. L. Cai, Advantages of the enhanced opposite direction searching algorithm for computing the centroid of an interval type-2 fuzzy set, Asian J. Control, **14** (2012), 1422–1430.
- 42. J. H. Hu, P. P. Chen, Y. Yang, *The fruit fly optimization algorithms for patient-centered care based on interval trapezoidal type-2 fuzzy numbers*, Int. J. Fuzzy Syst., **21** (2019), 1270–1287.
- 43. M. Javanmard, H. Mishmast Nehi, *A solving method for fuzzy linear programming problem with interval type-2 fuzzy numbers*, Int. J. Fuzzy Syst., **21** (2019), 882–891.
- 44. C. Chen, R. John, J. Twycross, et al. A direct approach for determining the switch points in the *Karnik-Mendel algorithm*, IEEE Trans. Fuzzy Syst., **26** (2018), 1079–1085.
- 45. O. Castillo, L. Amador-Angulo, J. R. Castro, et al. A comparative study of type-1 fuzzy logic systems, interval type-2 fuzzy logic systems and generalized type-2 fuzzy logic systems in control problems, Inf. Sci., **354** (2016), 257–274.
- 46. L. Cervantes, O. Castillo, *Type-2 fuzzy logic aggregation of multiple fuzzy controllers for airplane flight control*, Inf. Sci., **324** (2015), 247–256.
- 47. O. Castillo, P. Melin, E. Ontiveros, et al. *A high-speed interval type 2 fuzzy system approach for dynamic parameter adaptation in metaheuristics*, Eng. Appl. Artificial Intelligence, **85** (2019), 666–680.
- 48. E. Ontiveros-Robles, P. Melin, O. Castillo, *Comparative analysis of noise robustness of type 2 fuzzy logic controllers*, Kybernetika, **54** (2018), 175–201.
- 49. E. Ontiveros-Robles, P. Melin, O. Castillo, *New methodology to approximate type-reduction based on a continuous root-finding karnik mendel algorithm*, Algorithms, **10** (2017), 77–96.
- 50. Y. Chen, Study on sampling-based discrete noniterative algorithms for centroid type-reduction of interval type-2 fuzzy logic systems, Soft Comput., **24** (2020), 11819–11828.
- 51. S. C. Tong, Y. M. Li, Robust adaptive fuzzy backstepping output feedback tracking control for nonlinear system with dynamic uncertainties, Sci. China Inf. Sci., **53** (2010), 307–324.

- 52. S. C. Tong, Y. M. Li, Observer-based adaptive fuzzy backstepping control of uncertain pure-feedback systems, Sci. China Inf. Sci., 57 (2014), 1–14.
- 53. M. Deveci, I. Z. Akyurt, S. Yavuz, GIS-based interval type-2 fuzzy set for public bread factory site selection, J. Enterprise Inf. Manage., 31 (2018), 820–847.



© 2020 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0)