Mathematics

AIMS Mathematics, 7(6): 9846–9870
DOI: 10.3934/math.2022549
Received: 22 December 2021
Revised: 01 March 2022
Accepted: 04 March 2022
Published: 18 March 2022

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

Research article

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

Yang Chen*, Jiaxiu Yang and Chenxi Li

College of Science, Liaoning University of Technology, Jinzhou, Liaoning, 121001, China

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

Abstract: Interval type-2 fuzzy logic systems (IT2 FLSs) already become an emerging technology in recent years. As the most popular type-reduction (TR) algorithms, Karnik-Mendel (KM) algorithms own the advantage of maintaining the uncertainties flow in systems. This paper analyzes the initialization for KM types of algorithms. Furthermore, the weighting approaches of them are also given by means of the Newton-Cotes quadrature formulas. Importantly, the reasonable initialization weighted enhanced Karnik-Mendel (RIWEKM) algorithms are provided to complete the centroid type-reduction of IT2 FLSs. Three computer simulation experiments illustrate that, the proposed RIWEKM algorithms own both smaller absolute errors and faster convergence speeds in contrast to the EKM and RIEKM algorithms.

Keywords: initialization; interval type-2 fuzzy logic systems; Newton-Cotes quadrature formulas; reasonable initialization weighted EKM algorithms; simulations
Mathematics Subject Classification: 68XX, 68Uxx

1. Introduction

In the past few years, researching on IT2 FLSs [1–3] has attracted more attentions. The process of financial systems [4], power systems [5,6], permanent magnetic drive [8,9], intelligent controllers [7], plant monitoring and diagnostics [10], database and information systems [11] and so on are all with high uncertainty and nonlinearity [12]. The IT2 FSs have the potential to exceed T1 FSs as the former own the additional third mathematical dimension in the footprint of uncertainty (FOU). Therefore, IT2 FLSs based on IT2 FSs own the capability to reduce the influence of uncertainty better compared with their T1 counterparts. Recent study report shows that IT2 FLSs [13] even have superior approximation ability to the nonparametric network networks.

An IT2 FLS is made up of five modules as: Fuzzifier, inference, rules, type-reduction (TR) and defuzzification (see Figure 1). Among which, the most important module is the TR, which has the function of mapping the IT2 FS to the T1 FS. While the T1 FLSs only use T1 FSs. The TR can be viewed as the extended of T1 defuzzification. And the defuzzification changes the T1 FS into the output. In addition, the systems can be defined as IT2 FLSs if an antecedent or consequent in fuzzy rules were IT2 FS. The operations in IT2 FLSs are more complicated than in T1 FLSs as the former involves the TR. In this paper, we only focus on the TR.

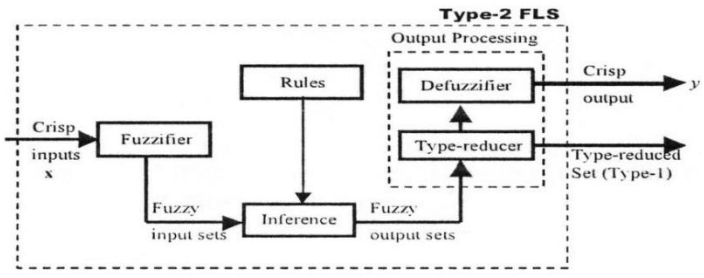


Figure 1. An IT2 FLS [1,5,14].

To date, the centroid TR [15,16] is still one of the most popular approaches in theory studies. Karnik and Mendel developed the KM algorithms to complete the centroid type-reduction of IT2 FLSs. For better understand the KM algorithms, Mendel and Wu put forward the continuous of KM (CKM) algorithms. Furthermore, the monotonic and super exponential convergence [17] of continuous KM algorithms were proved, which provided a major impulse for investigating IT2 FLSs. Wu and Mendel [12,18] applied the KM algorithms to calculate the uncertainties measure in IT2 FSs and merged the language weighted average of IT2 FSs into intuition calculations. For the sake of reducing the computation time, Wu and Mendel [19] put forward the enhanced type of KM (EKM) algorithms. The theoretical explanations for the initializing of EKM algorithms are provided by Liu et al. [20], in addition, the EKM algorithms are extended to weighted EKM (WEKM) algorithms to compute the centroids of IT2 fuzzy sets. All the above works have laid abundant theory for applying the TR algorithms.

On the basis of analyzing the sum operation in discrete EKM algorithms and the integration operation in CEKM algorithms, this paper first gives the explanations for the initialization of EKM algorithms. In terms of the Newton-Cotes quadrature formulas, three different forms of weighted EKM (WEKM) algorithms are also explained. Then the reasonable initialization weighted enhanced Karnik-Mendel (RIWEKM) algorithms are proposed to complete the centroid type-reduction of IT2 FLSs. While calculating the centroid type-reduced sets, the provided RIWEKM algorithms get smaller absolute errors and faster convergence speeds in contrast to both the EKM and RIEKM algorithms.

The rest of this paper is arranged as follows. Section 2 gives the background of IT2 FSs and IT2 FLSs. Section 3 shows the Newton-Cotes formulas, the EKM and CEKM algorithms, and how to calculate the centroid type-reduction of IT2 FLSs by means of RICEKM and RIWEKM algorithms. Three simulation instances are given in Section 4 to show the performances of the RIWEKM algorithms. Finally the last section gives the conclusions.

2. Background

2.1. IT2 FSs

Definition 1. An IT2 fuzzy set $\tilde{A}$ can be described by the corresponding type-2 membership function (MF) $\mu_{\tilde{A}}(x,u)$, that is to say, [1,3,21]

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

in which $x \in X$, and $u \in [0,1]$, Eq (1) is called as the point-value expression, while the compact form is as:

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

Definition 2. A secondary MF can be viewed as the vertical slice of $\mu_{\tilde{A}}(x,u)$, that is to say, [1]

$$\mu_{\tilde{A}}(x = x', u) \equiv \mu_{\tilde{A}}(x') = \int_{u \in [0,1]} f_{x'}(u)/u \tag{3}$$

where $\forall u \in [0,1]$, for an IT2 FS, the secondary MF grade $f_{x'}(u) \equiv 1$.

Definition 3. The two dimensional support of $\mu_{\tilde{A}}(x,u)$ is called as the footprint of uncertainty (FOU) of $\tilde{A}$, that is to say, [1]

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

in which $\mathrm{FOU}(\tilde{A})$ is limited between the upper MF (UMF) and the lower MF (LMF), i.e.,

$$\mathrm{UMF}(\tilde{A}) = \overline{\mu}_{\tilde{A}}(x) = \overline{\mathrm{FOU}(\tilde{A})}, \tag{5}$$

$$\mathrm{LMF}(\tilde{A}) = \underline{\mu}_{\tilde{A}}(x) = \underline{\mathrm{FOU}(\tilde{A})}\,. \tag{6}$$

Definition 4. An embedded T1 fuzzy set is decided by $\mu_{\tilde{A}}(x,u)$, that is to say, [1]

$$A_e = \{(x, u(x)) \mid \forall x \in X, u \in [0,1]\}\,. \tag{7}$$

As the secondary MF grades of IT2 fuzzy sets are equal to one, IT2 FSs can be entirely described by the LMF and UMF. Here an example of IT2 FS is provided in Figure 2.

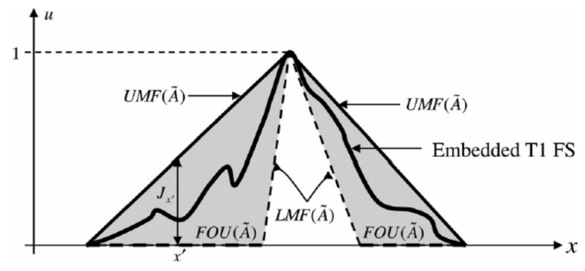


Figure 2. An IT2 FS and its corresponding quantities [24].

2.2. IT2 FLSs

From the aspect of structure, IT2 FLSs can be usually divided into Mamdani type [5,21,22] and Takagi-Sugeno-Kang type [8,23]. Here we concentrate on the Mamdani type. Without loss of generality, consider a Mamdani type IT2 FLS with n inputs $x_1 \in X_1, x_2 \in X_2, \cdots x_n \in X_n$ and one output $y \in Y$. The system can be characterized by M fuzzy rules, where the sth fuzzy rule is of the form:

$$R^s: \text{If } x_1 \text{ is } \tilde{F}_1^s \text{ and } x_2 \text{ is } \tilde{F}_2^s \text{ and} \cdots \text{and } x_n \text{ is } \tilde{F}_n^s, \text{then } y \text{ is } \tilde{G}^s (s = 1, \cdots, M) \tag{6}$$

in which $\tilde{F}_i^s (i = 1, \cdots, n; s = 1, \cdots, M)$ represents the antecedent IT2 fuzzy set, and $\tilde{G}^s (s = 1, \cdots, M)$ represents the consequent IT2 fuzzy set.

For simplicity, the singleton fuzzifier is adopted, that is to say, we model the input measurements as crisp sets. For each fuzzy rule, as $x = x'$, the firing interval can be calculated as:

$$F^s: \begin{cases} F^s(x') \equiv [\underline{f}^s(x'), \overline{f}^s(x')], \\ \underline{f}^s(x') \equiv T_{i=1}^n \underline{\mu}_{\tilde{F}_i^s}(x_i'), \\ \overline{f}^s(x') \equiv T_{i=1}^n \overline{\mu}_{\tilde{F}_i^s}(x_i') \end{cases} \tag{7}$$

in which T represents the product or minimum t-norm, $\underline{f}^s(x')$ and $\bar{f}^s(x')$ denote the left endpoint and right end point of firing interval, respectively.

For the centroid type-reduction, combine the firing interval with the corresponding consequent IT2 fuzzy set to get the fired-rule output fuzzy set $\tilde{B}^s$:

$$\tilde{B}^s: \begin{cases} \text{FOU}(\tilde{B}^s) = [\underline{\mu}_{\tilde{B}^s}(y \mid x'), \overline{\mu}_{\tilde{B}^s}(y \mid x')], \\ \underline{\mu}_{\tilde{B}^s}(y \mid x') = \underline{f}^s(x') * \underline{\mu}_{\tilde{G}^s}(y), \\ \overline{\mu}_{\tilde{B}^s}(y \mid x') = \overline{f}^s(x') * \overline{\mu}_{\tilde{G}^s}(y) \end{cases} \tag{8}$$

in which $*$ represents the product or minimum t-norm.

Then the aggregated IT2 fuzzy set $\tilde{B}$ can be got by unioning all the $\tilde{B}^s$ as:

$$\tilde{B}: \begin{cases} \text{FOU}(\tilde{B}) = [\underline{\mu}_{\tilde{B}}(y \mid x'), \overline{\mu}_{\tilde{B}}(y \mid x')], \\ \underline{\mu}_{\tilde{B}}(y \mid x') = \underline{\mu}_{\tilde{B}^1}(y \mid x') \vee \underline{\mu}_{\tilde{B}^2}(y \mid x') \vee \cdots \vee \underline{\mu}_{\tilde{B}^M}(y \mid x'), \\ \overline{\mu}_{\tilde{B}}(y \mid x') = \overline{\mu}_{\tilde{B}^1}(y \mid x') \vee \overline{\mu}_{\tilde{B}^2}(y \mid x') \vee \cdots \vee \overline{\mu}_{\tilde{B}^M}(y \mid x') \end{cases} \tag{9}$$

in which $\vee$ represents the maximum operation.

Finally the centroid of $\tilde{B}$ can be computed to get the type-reduced set $Y_C(x')$, i.e.,

$$Y_C(x') = 1/[l_{\tilde{B}}(x'), r_{\tilde{B}}(x')] \tag{10}$$

in which $l_{\tilde{B}}(x')$ and $r_{\tilde{B}}(x')$ can be computed by the TR algorithms, i.e.,

$$l_{\tilde{B}}(x') = \min_{\mu_{\tilde{B}}(y_i)\in[\underline{\mu}_{\tilde{B}}(y_i),\bar{\mu}_{\tilde{B}}(y_i)]} \frac{\sum_{i=1}^{N} y_i \mu_{\tilde{B}}(y_i)}{\sum_{i=1}^{N} \mu_{\tilde{B}}(y_i)} \tag{11}$$

and

$$r_{\tilde{B}}(x') = \max_{\mu_{\tilde{B}}(y_i)\in[\underline{\mu}_{\tilde{B}}(y_i),\bar{\mu}_{\tilde{B}}(y_i)]} \frac{\sum_{i=1}^{N} y_i \mu_{\tilde{B}}(y_i)}{\sum_{i=1}^{N} \mu_{\tilde{B}}(y_i)} \tag{12}$$

in which N represents the number of sampling of primary variable, and the defuzzified output is the average of two end points.

3. RIWEKM algorithms

Before giving the proposed RIWEKM algorithms, the theoretical interpretations of KM algorithms and EKM algorithms are provided. Then the Newton-Cotes quadrature formulas [25] are introduced.

3.1. The initialization explanations of KM and EKM algorithms

Let the primary variable of output IT2 fuzzy set $\tilde{B}$ be as: $a = y_1 < \cdots < y_N = b$, then the continuous KM and EKM (CKM and CEKM) algorithms for computing $l_{\tilde{B}}$ and $r_{\tilde{B}}$ are as (see Tables 1 and 2):

Table 1. CKM algorithms [16–19,25–27] for calculating the centroid of $\tilde{B}$.

Step	For $l_{\tilde{B}}$, $l_{\tilde{B}} = \min_{\forall\theta(y)\in[\underline{\mu}_{\tilde{B}}(y),\bar{\mu}_{\tilde{B}}(y)]} \frac{\int_a^b y\theta(y)dy}{\int_a^b \theta(y)dy}$
1	Let $\theta(y) = [\underline{\mu}_{\tilde{B}}(y) + \bar{\mu}_{\tilde{B}}(y)]/2$, and compute $\xi' = \frac{\int_a^b y\theta(y)dy}{\int_a^b \theta(y)dy}$.
2	Set $\theta(y) = \bar{\mu}_{\tilde{B}}(y)$ when $y \le \xi'$, and $\theta(y) = \underline{\mu}_{\tilde{B}}(y)$ when $y > \xi'$, and calculate $\xi_{l_{\tilde{B}}} = \frac{\int_a^b y\theta(y)dy}{\int_a^b \theta(y)dy}$.
3	Check if $\lvert \xi' - \xi_{l_{\tilde{B}}} \rvert \le \varepsilon$, if yes, stop and set $l_{\tilde{B}} = \xi_{l_{\tilde{B}}}$, if no, return to Step 4.
4	Set $\xi' = \xi_l$ and return to Step 2.

Continued on next page

Step	For $r_{\tilde{B}}$, $r_{\tilde{B}} = \max_{\forall \theta(y) \in [\underline{\mu}_{\tilde{B}}(y), \bar{\mu}_{\tilde{B}}(y)]} \frac{\int_a^b y\theta(y)dy}{\int_a^b \theta(y)dy}$
1	Let $\theta(y) = [\underline{\mu}_{\tilde{B}}(y) + \bar{\mu}_{\tilde{B}}(y)]/2$, and calculate $\xi' = \frac{\int_a^b y\theta(y)dy}{\int_a^b \theta(y)dy}$.
2	Set $\theta(y) = \underline{\mu}_{\tilde{B}}(y)$ when $y \le \xi'$, and $\theta(y) = \bar{\mu}_{\tilde{B}}(y)$ when $y > \xi'$, and compute $\xi_{r_{\tilde{B}}} = \frac{\int_a^b y\theta(y)dy}{\int_a^b \theta(y)dy}$.
3	Check if $\lvert \xi' - \xi_{r_{\tilde{B}}} \rvert \le \varepsilon$, if yes, stop and set $r_{\tilde{B}} = \xi_{r_{\tilde{B}}}$, if no, return to Step 4.
4	Set $\xi' = \xi_{r_{\tilde{B}}}$ and return to Step 2.

Table 2. CEKM algorithms [16–19,25–27] for calculating the centroid of $\tilde{B}$.

Step	For $l_{\tilde{B}}$
1	Set $c = a + (b-a)/2.4$, and compute $\alpha = \int_a^c y\bar{\mu}_{\tilde{B}}(y)dy + \int_c^b y\underline{\mu}_{\tilde{B}}(y)dy$, $\beta = \int_a^c \bar{\mu}_{\tilde{B}}(y)dy + \int_c^b \underline{\mu}_{\tilde{B}}(y)dy$, $c' = \alpha/\beta$.
2	Check if $\lvert c' - c \rvert \le \varepsilon$, if yes, stop and set $c' = l_{\tilde{B}}$, if no, return to Step 4.
3	Calculate $s = sign(c' - c)$, $\alpha' = \alpha + s\int_{\min(c,c')}^{\max(c,c')} y[\bar{\mu}_{\tilde{B}}(y) - \underline{\mu}_{\tilde{B}}(y)]dy$, $\beta' = \beta + s\int_{\min(c,c')}^{\max(c,c')} [\bar{\mu}_{\tilde{B}}(y) - \underline{\mu}_{\tilde{B}}(y)]dy$, $c'' = \alpha'/\beta'$.
4	Set $c = c', c' = c'', \alpha = \alpha', \beta = \beta'$ and return to Step 2.
Step	For $r_{\tilde{B}}$
1	Set $c = a + (b-a)/1.7$, and compute $\alpha = \int_a^c y\underline{\mu}_{\tilde{B}}(y)dy + \int_c^b y\bar{\mu}_{\tilde{B}}(y)dy$, $\beta = \int_a^c \underline{\mu}_{\tilde{B}}(y)dy + \int_c^b \bar{\mu}_{\tilde{B}}(y)dy$, $c' = \alpha/\beta$.
2	Check if $\lvert c' - c \rvert \le \varepsilon$, if yes, stop and set $c' = r_{\tilde{B}}$, if no, go to Step 4.
3	Compute $s = sign(c' - c)$, $\alpha' = \alpha - s\int_{\min(c,c')}^{\max(c,c')} y[\bar{\mu}_{\tilde{B}}(y) - \underline{\mu}_{\tilde{B}}(y)]dy$, $\beta' = \beta - s\int_{\min(c,c')}^{\max(c,c')} [\bar{\mu}_{\tilde{B}}(y) - \underline{\mu}_{\tilde{B}}(y)]dy$, $c'' = \alpha'/\beta'$.
4	Set $c = c', c' = c'', \alpha = \alpha', \beta = \beta'$ and return to Step 2.

$$l_{\tilde{B}} = \min_{\xi \in [a,b]} F_{l_{\tilde{B}}}(\xi) = \min_{\xi \in [a,b]} \frac{\int_a^{\xi} y\bar{\mu}_{\tilde{B}}(y)dy + \int_{\xi}^{b} y\underline{\mu}_{\tilde{B}}(y)dy}{\int_a^{\xi} \bar{\mu}_{\tilde{B}}(y)dy + \int_{\xi}^{b} \underline{\mu}_{\tilde{B}}(y)dy}, \tag{13}$$

$$r_{\tilde{B}} = \max_{\xi\in[a,b]} F_{r_{\tilde{B}}}(\xi) = \max_{\xi\in[a,b]} \frac{\int_a^{\xi} y\underline{\mu}_{\tilde{B}}(y)dy + \int_{\xi}^{b} y\overline{\mu}_{\tilde{B}}(y)dy}{\int_a^{\xi} \underline{\mu}_{\tilde{B}}(y)dy + \int_{\xi}^{b} \overline{\mu}_{\tilde{B}}(y)dy}. \tag{14}$$

As the continuous version of TR algorithms are always considered as the benchmark, here the calculational steps of CKM algorithms and CEKM algorithms are provided in Table1 and Table 2, respectively. In terms of the notations $F_{l_{\tilde{B}}}$ in (13) and $F_{r_{\tilde{B}}}$ in (14), and see the Steps 2 and 4 in Table 1, so that,

$$\xi_{l_{\tilde{B}}} = F_{l_{\tilde{B}}}(\xi'), \text{ and } \xi_{l_{\tilde{B}}} = \xi' \tag{15}$$

$$\xi_{r_{\tilde{B}}} = F_{r_{\tilde{B}}}(\xi'), \text{ and } \xi_{r_{\tilde{B}}} = \xi'. \tag{16}$$

When the iterations terminate, $l_{\tilde{B}} = \xi_{l_{\tilde{B}}}$ and $r_{\tilde{B}} = \xi_{r_{\tilde{B}}}$, so that,

$$l_{\tilde{B}} = F_{l_{\tilde{B}}}(l_{\tilde{B}}),\ \ r_{\tilde{B}} = F_{r_{\tilde{B}}}(r_{\tilde{B}}) \tag{17}$$

where $l_{\tilde{B}}$ and $r_{\tilde{B}}$ are the fixed points of $F_{l_{\tilde{B}}}(\xi)$ and $F_{r_{\tilde{B}}}(\xi)$, respectively. In a similar way, the relations of (17) are also true for the CEKM algorithms as in Table 2.

In the aim of initializing the algorithms, we set $\underline{\mu}_{\tilde{B}}(y) = \overline{\mu}_{\tilde{B}}(y) = \theta(y)$ for $y \in [a,b]$, so that, $\theta(y) = [\underline{\mu}_{\tilde{B}}(y) + \overline{\mu}_{\tilde{B}}(y)]/2$, and the Eqs (13) and (14) become the same, therefore,

$$l_{\tilde{B}} = r_{\tilde{B}} = \frac{\int_a^b y\theta(y)dy}{\int_a^b \theta(y)dy} = \frac{\int_a^b y[\underline{\mu}_{\tilde{B}}(y) + \overline{\mu}_{\tilde{B}}(y)]/2dy}{\int_a^b [\underline{\mu}_{\tilde{B}}(y) + \overline{\mu}_{\tilde{B}}(y)]/2dy} \tag{18}$$

here (18) is the initialization method of CKM algorithms as shown in the Table 1, represented as $\xi^{(1)}$, that is to say,

$$\xi^{(1)} = \frac{\int_a^b y\theta(y)dy}{\int_a^b \theta(y)dy} \tag{19}$$

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

The calculational steps of discrete KM algorithms and EKM algorithms are given in the following Tables 3 and 4. For the Eq (19), the discrete form of $\xi^{(1)}$ is as in Steps 1 and 2 in the Table 3, that is to say,

$$k^{(1)} = \{k \mid y_k \le \frac{\sum_{i=1}^{N} y_i[\underline{\mu}_B(y_i) + \overline{\mu}_B(y_i)]}{\sum_{i=1}^{N} [\underline{\mu}_B(y_i) + \overline{\mu}_B(y_i)]} < y_{k+1}, 1 \le k \le N-1\} \tag{20}$$

for the KM initialization method, the results can be good as $\underline{\mu}_{\tilde{B}}(y)$ and $\overline{\mu}_{\tilde{B}}(y)$ are very close due to the exact optimal solution of (11) or (12) as $\underline{\mu}_{\tilde{B}}(y) = \overline{\mu}_{\tilde{B}}(y)$.

Table 3. KM algorithms [16–19,25–27] for calculating the centroid of $\tilde{B}$.

Step	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 $\theta(i) = [\underline{\mu}_{\tilde{B}}(y_i) + \overline{\mu}_{\tilde{B}}(y_i)]/2, i = 1, \cdots, N$ and calculate $c' = (\sum_{i=1}^{N} y_i \theta_i)/(\sum_{i=1}^{N} \theta_i)$.
2	Find $k'(1 \le k' \le N-1)$ which satisfies $y_{k'} \le c' < y_{k'+1}$.
3	Set $\theta_i = \overline{\mu}_{\tilde{B}}(y_i)$ when $i \le k'$, and $\theta_i = \underline{\mu}_{\tilde{B}}(y_i)$ when $i > k'$, and calculate $l_{\tilde{B}}(k') = (\sum_{i=1}^{N} y_i \theta_i)/(\sum_{i=1}^{N} \theta_i)$.
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, return to Step 5.
5	Set $c' = l_{\tilde{B}}(k')$ and return to Step 2.
Step	For $r_{\tilde{B}}$, $r_{\tilde{B}_\alpha} = \max_{\forall \theta_i \in [\underline{\mu}_{\tilde{B}}(y_i), \overline{\mu}_{\tilde{B}}(y_i)]} (\sum_{i=1}^{N} y_i \theta_i)/(\sum_{i=1}^{N} \theta_i)$
1	Set $\theta(i) = [\underline{\mu}_{\tilde{B}}(y_i) + \overline{\mu}_{\tilde{B}}(y_i)]/2, i = 1, \cdots, N$ and calculate $c' = (\sum_{i=1}^{N} y_i \theta_i)/(\sum_{i=1}^{N} \theta_i)$.
2	Find $k'(1 \le k' \le N-1)$ which satisfies $y_{k'} \le c' < y_{k'+1}$.
3	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 calculate $r_{\tilde{B}}(k') = (\sum_{i=1}^{N} y_i \theta_i)/(\sum_{i=1}^{N} \theta_i)$.
4	Check if $r_{\tilde{B}}(k') = c'$, if yes, stop and set $r_{\tilde{B}}(k') = r_{\tilde{B}}$ and $k' = R$, if no, return to Step 5.
5	Set $c' = r_{\tilde{B}}(k')$ and return to Step 2.

Table 4. EKM algorithms [16–19,25–27] for calculating the centroid of $\tilde{B}$.

Step	For $l_{\tilde{B}}$
1	Set $k = [N/2.4]$ (the nearest integer to $N/2.4$) and calculate $\alpha = \sum_{i=1}^{k} y_i \overline{\mu}_{\tilde{B}}(y_i) + \sum_{i=k+1}^{N} y_i \underline{\mu}_{\tilde{B}}(y_i)$, $\beta = \sum_{i=1}^{k} \overline{\mu}_{\tilde{B}}(y_i) + \sum_{i=k+1}^{N} \underline{\mu}_{\tilde{B}}(y_i)$, $c' = \alpha/\beta$.
2	Find $k'(1 \le k' < N-1)$ which satisfies $y_{k'} \le c' < y_{k'+1}$.
3	Check if $k' = k$, if yes, stop and set $c' = l_{\tilde{B}}$ and $k = L$, if no, return to Step

Continued on next page

Step	For $l_{\tilde{B}}$
4	Compute $s = sign(k'-k)$, $\alpha' = \alpha + s\sum_{i=\min(k,k')+1}^{\max(k,k')} y_i\,[\overline{\mu}_{\tilde{B}}(y_i) - \underline{\mu}_{\tilde{B}}(y_i)]$, $\beta' = \beta + s\sum_{i=\min(k,k')+1}^{\max(k,k')} [\overline{\mu}_{\tilde{B}}(y_i) - \underline{\mu}_{\tilde{B}}(y_i)]$, $c'' = \alpha'/\beta'$.
5	Set $c' = c'', \alpha = \alpha', \beta = \beta', k = k'$ and go to Step 2.

Step	For $r_{\tilde{B}}$
1	Set $k = [N/1.7]$ and compute $\alpha = \sum_{i=1}^{k} y_i \underline{\mu}_{\tilde{B}}(y_i) + \sum_{i=k+1}^{N} y_i \overline{\mu}_{\tilde{B}}(y_i)$, $\beta = \sum_{i=1}^{k} \underline{\mu}_{\tilde{B}}(y_i) + \sum_{i=k+1}^{N} \overline{\mu}_{\tilde{B}}(y_i)$, $c' = \alpha/\beta$.
2	Find $k'(1 \le k' < N-1)$ which satisfies $y_{k'} \le c' < y_{k'+1}$.
3	Check if $k' = k$, if yes, stop and set $c' = r_{\tilde{B}}$ and $k = R$, if no, return to Step
4	Compute $s = sign(k'-k)$, $\alpha' = \alpha - s\sum_{i=\min(k,k')+1}^{\max(k,k')} y_i\,[\overline{\mu}_{\tilde{B}}(y_i) - \underline{\mu}_{\tilde{B}}(y_i)]$, $\beta' = \beta - s\sum_{i=\min(k,k')+1}^{\max(k,k')} [\overline{\mu}_{\tilde{B}}(y_i) - \underline{\mu}_{\tilde{B}}(y_i)]$, $c'' = \alpha'/\beta'$.
5	Set $c' = c'', \alpha = \alpha', \beta = \beta', k = k'$ and return to Step 2.

While for the initializations of EKM algorithms, the method is on account of the difference between $\underline{\mu}_{\tilde{B}}(y)$ and $\overline{\mu}_{\tilde{B}}(y)$. Assume that,

$$\rho = \frac{\int_a^b \overline{\mu}_{\tilde{B}}(y)dy}{\int_a^b \underline{\mu}_{\tilde{B}}(y)dy}. \tag{21}$$

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

For the sake of initializing the EKM algorithms, here we let $\underline{\mu}_{\tilde{B}}(y)$ and $\overline{\mu}_{\tilde{B}}(y)$ be constants for $y \in [a,b]$, that is to say, suppose that $\overline{\mu}_{\tilde{B}}(y) = \rho n > 0$, see the Eq (11), it can be obtained that,

$$F_{l_{\tilde{B}}}(\xi) = \frac{\int_a^{\xi} y\overline{\mu}_{\tilde{B}}(y)dy + \int_{\xi}^{b} y\underline{\mu}_{\tilde{B}}(y)dy}{\int_a^{\xi} \overline{\mu}_{\tilde{B}}(y)dy + \int_{\xi}^{b} \underline{\mu}_{\tilde{B}}(y)dy} = \frac{\int_a^{\xi} \rho nydy + \int_{\xi}^{b} nydy}{\int_a^{\xi} \rho ndy + \int_{\xi}^{b} ndy} = \frac{\rho(\xi^2 - a^2) + (b^2 - \xi^2)}{2[\rho(\xi - a) + (b - \xi)]} \tag{22}$$

Compute the derivative of $F_{l_{\tilde{B}}}(\xi)$ with respect to ξ to obtain that,

$$F'_{l_{\tilde{B}_\alpha}}(\xi) = \frac{(\rho - 1)[\rho(\xi - a)^2 - (b - \xi)^2]}{2[\rho(\xi - a) + (b - \xi)]^2}. \tag{23}$$

While letting $F'_{l_{\tilde{B}}}(\xi) = 0$, so that,

$$\frac{(b-\xi)^2}{(\xi-a)^2}=\rho\text{, and }\frac{b-\xi}{\xi-a}=\sqrt{\rho}\,. \tag{24}$$

Then solve the Eq (24) to get the ξ, i.e.,

$$\xi_{l_{\tilde{B}}}=a+\frac{b-a}{1+\sqrt{\rho}}. \tag{25}$$

Here $\xi_{l_{\tilde{B}}}$ denotes the minimum value of $F_{l_{\tilde{B}}}(\xi)$. As $F'_{l_{\tilde{B}}}(\xi)<0$ for $\xi\in[a,\xi_{l_{\tilde{B}}})$, and $F'_{l_{\tilde{B}}}(\xi)>0$ for $\xi\in(\xi_{l_{\tilde{B}}},b]$, so that, $l_{\tilde{B}}=F_{l_{\tilde{B}}}(\xi_{l_{\tilde{B}}})=\xi_{l_{\tilde{B}}}$.

Similar process can be done for finding the maximum value of $F_{r_{\tilde{B}}}(\xi)$ as follows:

$$F_{r_{\tilde{B}}}(\xi)=\frac{\int_a^{\xi} y\underline{\mu}_{\tilde{B}}(y)dy+\int_{\xi}^{b} y\overline{\mu}_{\tilde{B}}(y)dy}{\int_a^{\xi}\underline{\mu}_{\tilde{B}}(y)dy+\int_{\xi}^{b}\overline{\mu}_{\tilde{B}}(y)dy}=\frac{(\xi^2-a^2)+\rho(b^2-\xi^2)}{2[(\xi-a)+\rho(b-\xi)]} \tag{26}$$

Then calculate the derivative of $F_{r_{\tilde{B}}}(\xi)$ with respect to ξ to obtain that,

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

While letting $F'_{r_{\tilde{B}}}(\xi)=0$, so that,

$$\frac{(b-\xi)^2}{(\xi-a)^2}=\frac{1}{\rho}\text{, and }\frac{b-\xi}{\xi-a}=\sqrt{1/\rho}\,. \tag{28}$$

Solve the Eq (28) to get the ξ, i.e.,

$$\xi_{r_{\tilde{B}}}=a+\frac{b-a}{1+\sqrt{1/\rho}}. \tag{29}$$

Here $\xi_{r_{\tilde{B}}}$ represents the maximum value of $F_{r_{\tilde{B}}}(\xi)$. As $F'_{r_{\tilde{B}}}(\xi)>0$ for $\xi\in[a,\xi_{r_{\tilde{B}}})$, and $F'_{r_{\tilde{B}}}(\xi)>0$ for $\xi\in(\xi_{r_{\tilde{B}}},b]$, so that, $r_{\tilde{B}}=F_{r_{\tilde{B}}}(\xi_{r_{\tilde{B}}})=\xi_{r_{\tilde{B}}}$.

By considering the Eqs (25) and (29) simultaneously, the new initialization method for ξ can be represented as $\xi^{(2)}$, i.e.,

$$\xi^{(2)}=\begin{cases} a+\dfrac{b-a}{1+\sqrt{\rho}} & \text{for } l_{\tilde{B}}, \\ a+\dfrac{b-a}{1+\sqrt{1/\rho}} & \text{for } r_{\tilde{B}}. \end{cases} \tag{30}$$

where $\rho\geq 1$, so that, $\xi^{(2)}\leq a+\frac{1}{2}(b-a)$ for $l_{\tilde{B}}$, and $\xi^{(2)}\geq a+\frac{1}{2}(b-a)$ for $r_{\tilde{B}}$.

By comparing the initializations of discrete form of EKM algorithms as shown in Table 4 and CEKM algorithms as shown in Table 2, we can obtain the discrete form of Eq (30) as:

$$k^{(2)} = \begin{cases} [N/(1+\sqrt{\rho})] & \text{for } l_{\tilde{B}}, \\ [N/(1+\sqrt{1/\rho})] & \text{for } r_{\tilde{B}}. \end{cases} \tag{31}$$

in which N is the number of sampling of primary variable, and $\rho = \frac{\sum_{i=1}^{N} \overline{\mu}_{\tilde{B}}(y)}{\sum_{i=1}^{N} \underline{\mu}_{\tilde{B}}(y)}$.

As $\rho = 2$, $1+\sqrt{\rho} = 1+\sqrt{2} \approx 2.4$, and $1+\sqrt{1/\rho} = 1+\sqrt{1/2} \approx 1.7$, so that, the Eq (31) turns out to be:

$$k^{(2)} = \begin{cases} [N/2.4] & \text{for } l_{\tilde{B}}, \\ [N/1.7] & \text{for } r_{\tilde{B}}. \end{cases} \tag{32}$$

here the Eq (32) provides the initialization method of EKM algorithms, which comes from empirical large amount of simulations. For the proposed RIEKM algorithms, each specific $\rho = [\sum_{i=1}^{N} \overline{\mu}_{\tilde{B}}(y)]/[\sum_{i=1}^{N} \underline{\mu}_{\tilde{B}}(y)]$ should be defined for the corresponding simulation experiments. In this section, we provide the computation steps of discrete RIEKM algorithms and RICEKM algorithms as in Tables 5 and 6, respectively.

Table 5. RIEKM algorithms [16–19,25–27,32] for calculating the centroid of $\tilde{B}$.

Step	For $l_{\tilde{B}}$
1	Set $k = [N/(1+\sqrt{\rho})]$ and calculate $\alpha = \sum_{i=1}^{k} y_i \underline{\mu}_{\tilde{B}}(y_i) + \sum_{i=k+1}^{N} y_i \overline{\mu}_{\tilde{B}}(y_i)$, $\beta = \sum_{i=1}^{k} \underline{\mu}_{\tilde{B}}(y_i) + \sum_{i=k+1}^{N} \overline{\mu}_{\tilde{B}}(y_i)$, $c' = \alpha/\beta$
2	Find $k'(1 \le k' < N-1)$ which satisfies $y_{k'} \le c' < y_{k'+1}$.
3	Check if $k' = k$, if yes, stop and set $c' = l_{\tilde{B}}$ and $k = L$, if no, return to Step 4
4	Compute $s = sign(k'-k)$, $\alpha' = \alpha + s \sum_{i=\min(k,k')+1}^{\max(k,k')} y_i [\overline{\mu}_{\tilde{B}}(y_i) - \underline{\mu}_{\tilde{B}}(y_i)]$, $\beta' = \beta + s \sum_{i=\min(k,k')+1}^{\max(k,k')} [\overline{\mu}_{\tilde{B}}(y_i) - \underline{\mu}_{\tilde{B}}(y_i)]$, $c'' = \alpha'/\beta'$.
5	Set $c' = c'', \alpha = \alpha', \beta = \beta', k = k'$ and return to Step 2.
Step	**For $r_{\tilde{B}}$**
1	Set $k = [N/(1+\sqrt{1/\rho})]$ and calculate $\alpha = \sum_{i=1}^{k} y_i \underline{\mu}_{\tilde{B}}(y_i) + \sum_{i=k+1}^{N} y_i \overline{\mu}_{\tilde{B}}(y_i)$, $\beta = \sum_{i=1}^{k} \underline{\mu}_{\tilde{B}}(y_i) + \sum_{i=k+1}^{N} \overline{\mu}_{\tilde{B}}(y_i)$, $c' = \alpha/\beta$
2	Find $k'(1 \le k' < N-1)$ which satisfies $y_{k'} \le c' < y_{k'+1}$.

Continued on next page

Step	For $r_{\tilde{B}}$
3	Check if $k' = k$, if yes, stop and set $c' = r_{\tilde{B}}$ and $k = R$, if no, return to Step
4	Calculate $s = sign(k' - k)$, $\alpha' = \alpha - s\sum_{i=\min(k,k')+1}^{\max(k,k')} y_i\,[\overline{\mu}_{\tilde{B}}(y_i) - \underline{\mu}_{\tilde{B}}(y_i)]$, $\beta' = \beta - s\sum_{i=\min(k,k')+1}^{\max(k,k')}[\overline{\mu}_{\tilde{B}}(y_i) - \underline{\mu}_{\tilde{B}}(y_i)]$, $c'' = \alpha'/\beta'$.
5	Set $c' = c'', \alpha = \alpha', \beta = \beta', k = k'$ and go to Step 2.

Table 6. RICEKM algorithms [16–19,25–27] for calculating the centroid of $\tilde{B}$.

Step	For $l_{\tilde{B}}$
1	Set $\rho = (\int_a^b \overline{\mu}_{\tilde{B}}(y)dy)/(\int_a^b \underline{\mu}_{\tilde{B}}(y)dy)$, $c = a + (b-a)/(1+\sqrt{\rho})$, and compute $\alpha = \int_a^c y\overline{\mu}_{\tilde{B}}(y)dy + \int_c^b y\underline{\mu}_{\tilde{B}}(y)dy$, $\beta = \int_a^c \overline{\mu}_{\tilde{B}}(y)dy + \int_c^b \underline{\mu}_{\tilde{B}}(y)dy$, $c' = \alpha/\beta$.
2	Check if $\lvert c' - c\rvert \le \varepsilon$, if yes, stop and set $c' = l_{\tilde{B}}$, if no, return to Step 4.
3	Compute $s = sign(c' - c)$, $\alpha' = \alpha + s\int_{\min(c,c')}^{\max(c,c')} y[\overline{\mu}_{\tilde{B}}(y) - \underline{\mu}_{\tilde{B}}(y)]dy$, $\beta' = \beta + s\int_{\min(c,c')}^{\max(c,c')}[\overline{\mu}_{\tilde{B}}(y) - \underline{\mu}_{\tilde{B}}(y)]dy$, $c'' = \alpha'/\beta'$.
4	Set $c = c', c' = c'', \alpha = \alpha', \beta = \beta'$ and return to Step 2.
Step	**For $r_{\tilde{B}}$**
1	Set $\rho = (\int_a^b \overline{\mu}_{\tilde{B}}(y)dy)/(\int_a^b \underline{\mu}_{\tilde{B}}(y)dy)$, $c = a + (b-a)/(1+\sqrt{1/\rho})$, and compute $\alpha = \int_a^c y\underline{\mu}_{\tilde{B}}(y)dy + \int_c^b y\overline{\mu}_{\tilde{B}}(y)dy$, $\beta = \int_a^c \underline{\mu}_{\tilde{B}}(y)dy + \int_c^b \overline{\mu}_{\tilde{B}}(y)dy$, $c' = \alpha/\beta$.
2	Check if $\lvert c' - c\rvert \le \varepsilon$, if yes, stop and set $c' = r_{\tilde{B}}$, if no, return to Step 4.
3	Compute $s = sign(c' - c)$, $\alpha' = \alpha - s\int_{\min(c,c')}^{\max(c,c')} y[\overline{\mu}_{\tilde{B}}(y) - \underline{\mu}_{\tilde{B}}(y)]dy$, $\beta' = \beta - s\int_{\min(c,c')}^{\max(c,c')}[\overline{\mu}_{\tilde{B}}(y) - \underline{\mu}_{\tilde{B}}(y)]dy$, $c'' = \alpha'/\beta'$.
4	Set $c = c', c' = c'', \alpha = \alpha', \beta = \beta'$ and return to Step 2.

3.2. Newton-Cotes quadrature formulas

The numerical integration approximates the $\int_a^b f(x)dx$ as the linear combination of functional values $f(x_i)$ on discrete points. So that the computation of definite integral can be changed to calculate the functional values.

Definition 5 (quadrature formula). Let $a = x_0 < x_1 < \cdots < x_n = b$, the following equation

$$Q(f)=\sum_{i=0}^{N} w_i f(x_i)=w_0 f(x_0)+w_1 f(x_1)+\cdots+w_N f(x_N) \tag{33}$$

which has the property

$$\int_a^b f(x)dx=Q(f)+E(f) \tag{34}$$

here (33) is referred to as the quadrature formula, $\{w_i\}_{i=0}^N$ is referred to as the weight coefficient, $\{x_i\}_{i=0}^N$ is referred to as the integration node, and the $E(f)$ is called as the remainder.

Then the following composite trapezoidal rule, composite Simpson rule, and composite Simpson 3/8 rule are used to approximate the function $f(x)$ as the straight line, quadratic polynomial, and cubic polynomial, respectively. Note that this section is adapted from the references [14,20,25].

Theorem 1 (Composite trapezoidal rule). Let the function $f(x)$ be defined on the interval $[a,b]$. Decompose the $[a,b]$ into n subintervals $\{x_{i-1},x_i\}_{i=1}^n$ with the equal distance $h=\frac{b-a}{n}$, where the uniformly-spaced node is as: $x_i=x_0+ih(i=0,1,2,\cdots,n)$. Then the numerical approximation of $\int_a^b f(x)dx$ with the composite trapezoidal rule can be as:

$$\int_a^b f(x)dx=\frac{h}{2}[f(a)+f(b)+2\sum_{i=1}^{n-1} f(x_i)]+E_T(f,h) \tag{35}$$

if f were second-order continuous derivable on the$[a,b]$, then $E_T(f,h)=-\frac{(b-a)f''(\zeta)}{12}h^2$, in which $\zeta\in(a,b)$.

Theorem 2 (Composite Simpson rule). Let $f(x)$ be defined on the interval $[a,b]$. Decompose the $[a,b]$ into $2n$ subintervals $\{x_{i-1},x_i\}_{i=1}^{2n}$ with the equal distance $h=\frac{b-a}{2n}$, in which the uniformly-spaced node is as: $x_i=x_0+ih(i=0,1,2,\cdots,2n)$. Then the numerical approximation of $\int_a^b f(x)dx$ with the composite Simpson rule can be as:

$$\int_a^b f(x)dx=\frac{h}{3}[f(a)+f(b)+2\sum_{i=1}^{n-1} f(x_{2i})+2\sum_{i=0}^{n-1} f(x_{2i+1})+E_S(f,h) \tag{36}$$

if f were fourth-order continuous derivable on $[a,b]$, then $E_S(f,h)=-\frac{(b-a)f^{(4)}(\zeta)}{180}h^4$, in which $a<\zeta<b$.

Theorem 3 (Composite Simpson 3/8 rule). Let $f(x)$ be defined on the interval $[a,b]$. Decompose the $[a,b]$ into $3n$ subintervals $\{x_{i-1},x_i\}_{i=1}^{3n}$ with the equal distance $h=\frac{b-a}{3n}$, where the uniformly-spaced node is as: $x_i=x_0+ih(i=0,1,2,\cdots,3n)$. Then the numerical approximation of $\int_a^b f(x)dx$ with the composite Simpson rule can be as:

$$\int_a^b f(x)dx = \frac{3h}{8}[f(a)+f(b)+\sum_{i=1}^{n}2f(x_{3i})+\sum_{i=1}^{n}3f(x_{3i-2})+\sum_{i=1}^{n}3f(x_{3i-1})]+E_{SC}(f,h) \tag{37}$$

if f were fourth-order continuous derivable on $[a,b]$, then $E_{SC}(f,h) = -\frac{(b-a)f^{(4)}(\zeta)}{80}h^4$, in which $a<\zeta<b$.

3.3. RIWEKM algorithms

According to the former two subsections, this section proposes the RIWEKM algorithms to complete the centroid type-reduction of IT2 FLSs. Here the reasonable initialization continuous EKM (RICEKM) algorithms are considered as the benchmark, then the computation results of RIWEKM algorithms are used for comparing with both the EKM algorithms and RIEKM algorithms. The RIWEKM algorithms can be viewed as the numerical implementation of RICEKM algorithms. To compare the Tables 5 and 6, we can find that the operations in discrete RIEKM and RICEKM are similar, because the sum operations in discrete algorithms are changed to the integral operations in continuous algorithms, i.e., the sum operations of sampling points in RIEKM algorithms act as integration operations of related function in RICEKM algorithms. By means of the Newton-Cotes quadrature formulas, we can assign the corresponding coefficient w_i for each node x_i, then the more accurate computation results may be obtained. Table 7 provides the steps for calculating the centroid of an IT2 FS according to the RIWEKM algorithms. The RIEKM algorithms are just a particular case of the RIWEKM algorithms as the coefficients of the latter are selected as $w_i = 1(i=1,2,\cdots,N)$. Although many coefficients assignment methods can be used, here we only adopt the numerical integration approaches on the basis of the Newton-Cotes quadrature formulas in Section 3.2, which are referred to as the composite trapezoidal rule, composite Simpson rule, and composite Simpson 3/8 rule. Here the Table 8 provides the weight assignment approaches of RIWEKM algorithms, in which all the sampling points are equally distributed on $[a,b]$, that is to say, $x_i = a+\frac{i-1}{N-1}(b-a)$ $(i=1,\cdots,N)$. Here the letter of primary variable of output IT2 fuzzy set is represented by the x.

Table 7. RIWEKM algorithms for calculating the centroid of $\tilde{B}$.

Step	For $l_{\tilde{B}}$
1	Set $k=[N/(1+\sqrt{\rho})]$ and compute $\alpha=\sum_{i=1}^{k}w_i y_i \underline{\mu}_{\tilde{B}}(y_i)+\sum_{i=k+1}^{N}w_i y_i \overline{\mu}_{\tilde{B}}(y_i)$, $\beta=\sum_{i=1}^{k}w_i \underline{\mu}_{\tilde{B}}(y_i)+\sum_{i=k+1}^{N}w_i \overline{\mu}_{\tilde{B}}(y_i)$, $c'=\alpha/\beta$.
2	Find $k'(1\le k'<N-1)$ which satisfies $y_{k'}\le c'<y_{k'+1}$.
3	Check if $k'=k$, if yes, stop and set $c'=l_{\tilde{B}}$ and $k=L$, if no, return to Step 4.

Continued on next page

Step	For $l_{\tilde{B}}$
4	Calculate $s = sign(k'-k)$, $\alpha' = \alpha + s\sum_{i=\min(k,k')+1}^{\max(k,k')} w_i y_i[\overline{\mu}_{\tilde{B}}(y_i) - \underline{\mu}_{\tilde{B}}(y_i)]$, $\beta' = \beta + s\sum_{i=\min(k,k')+1}^{\max(k,k')} w_i[\overline{\mu}_{\tilde{B}}(y_i) - \underline{\mu}_{\tilde{B}}(y_i)]$, $c'' = \alpha'/\beta'$.
5	Set $c' = c'', \alpha = \alpha', \beta = \beta', k = k'$ and return to Step 2.
Step	For $r_{\tilde{B}}$
1	Set $k = [N/(1+\sqrt{1/\rho})]$ and calculate $\alpha = \sum_{i=1}^{k} w_i y_i \underline{\mu}_{\tilde{B}}(y_i) + \sum_{i=k+1}^{N} w_i y_i \overline{\mu}_{\tilde{B}}(y_i$ $\beta = \sum_{i=1}^{k} w_i \underline{\mu}_{\tilde{B}}(y_i) + \sum_{i=k+1}^{N} w_i \overline{\mu}_{\tilde{B}}(y_i)$, $c' = \alpha/\beta$
2	Find $k'(1 \le k' < N-1)$ which satisfies $y_{k'} \le c' < y_{k'+1}$.
3	Check if $k' = k$, if yes, stop and set $c' = r_{\tilde{B}}$ and $k = R$, if no, return to Step 4.
4	Calculate $s = sign(k'-k)$, $\alpha' = \alpha - s\sum_{i=\min(k,k')+1}^{\max(k,k')} w_i y_i[\overline{\mu}_{\tilde{B}}(y_i) - \underline{\mu}_{\tilde{B}}(y_i)]$, $\beta' = \beta - s\sum_{i=\min(k,k')+1}^{\max(k,k')} w_i[\overline{\mu}_{\tilde{B}}(y_i) - \underline{\mu}_{\tilde{B}}(y_i)]$, $c'' = \alpha'/\beta'$.
5	Set $c' = c'', \alpha = \alpha', \beta = \beta', k = k'$ and return to Step 2.

Table 8. Weight assignment approaches of RIWEKM algorithms.

Algorithms	Integration Rule	Weight coefficient
RIEKM	____________	$w_i = 1(i = 1,\cdots,N)$
RITWEKM	Composite Trapezoidal Rule	$w_i = \begin{cases} 1/2, & i = 1, N, \\ 1, & i \ne 1, N. \end{cases}$
RISWEKM	Composite Simpson Rule	$w_i = \begin{cases} 1/2, & i = 1, N, \\ 1, & i = 1\bmod(2), and\, i \ne 1, N, \\ 2, & i = 0\bmod(2), and\, i \ne N. \end{cases}$
RIS3/8WEKM	Composite Simpson3/8 Rule	$w_i = \begin{cases} 1/3, & i = 1, N, \\ 2/3, & i = 1\bmod(3), and\, i \ne 1, N, \\ 1, & i = 2\bmod(3), \text{and}\, i \ne N, \\ 1, & i = 0\bmod(2), and\, i \ne N. \end{cases}$

In order to perform the centroid type-reduction of IT2 FLSs, we give the detail steps as:

1) According to the fuzzy reasoning [28,29], merging the fired fuzzy rules of IT2 FLSs to obtain the centroid output IT2 FS $\tilde{B}$.

2) Adopt the RICEKM and RIWEKM algorithms to compute the centroid defuzzified values of IT2 FLSs.

3) Compare and analyze the calculation results of RIWEKM algorithms, EKM algorithms and RIEKM algorithms by viewing the CEKM algorithms as a baseline.

4) Compare and analyze the calculaltional times of RIWEKM algorithms, EKM algorithms and

RIEKM algorithms.

In Table 8, except for the RIEKM algorithms, the coefficients for other three types of RIWEKM algorithms are assigned by means of the following steps:

1) Substitute $x_0 = a$, $x_N = b$, $x_i (i = 0,1,\cdots,N)$ in Eq (35), $x_0 = a$, $x_{2N} = b$, $x_i (i = 0,1,2,\cdots,2N)$ in Eq (36), and $x_0 = a$, $x_{3N} = b$, $x_i (i = 0,\cdots,3N)$ in Eq (37) all by $x_1 = a$, $x_N = b$, $x_i (i = 1,2,\cdots,N)$.

2) Coefficients $h/2$, $h/3$ and $3h/8$ in Eqs (35)–(37) can all be balanced out by the quotient of two integrals as in Table 6.

3) Observing from the Table 8, the coefficients of RITWEKM algorithms and RISWEKM algorithms are assigned as a half of brackets in Eqs (35) and (36).

4) The number of sampling points of RISWEKM algorithms and RIS3/8WEKM algorithms are not only restrict to $N = 2n + 1$ and $N = 3n + 1$ (here n is an arbitrary integer), but as the requirements $N = 1 \bmod (2)$ and $N = 1 \bmod (3)$ in Eqs (36) and (37) (in which mod represents the modulus operator).

Finally we can make the following conclusions for the relations between RICEKM algorithms and RIWEKM algorithms as for completing the centroid type-reduction and defuzzification of IT2 FLSs:

1) The RIWEKM algorithms are based on the sum operations of sampling points $y_i (i = 1,\cdots,N)$ to complete centroid type-reduction of IT2 FLSs. When it stops, the optimal switching points are used to approximate the centroids. While the RICEKM algorithms adopt the integration operations to complete the centroid type-reduction and defuzzification, which may obtain the comparatively accurate type-reduced sets and defuzzified values. Theoretically, as the number of sampled points approaches infinity, the solutions of RIWEKM algorithms will approach to RICEKM algorithms.

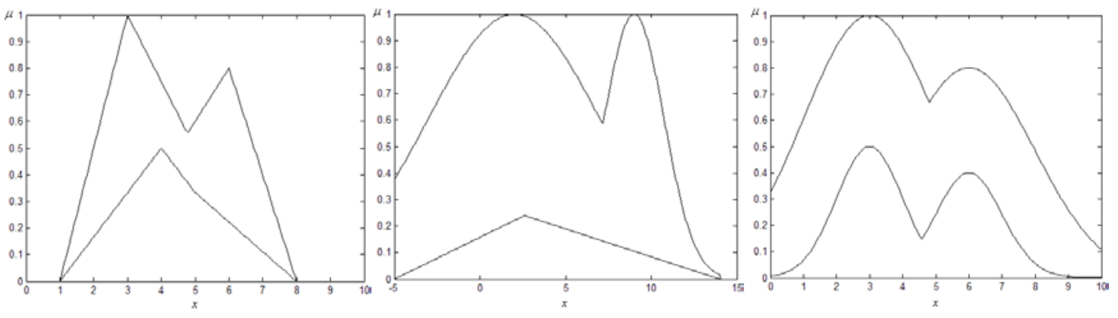
2) As for the RIWEKM algorithms, we can increase the number of sampled points to get the more accurate computation results. But for the RICEKM algorithms, the calculation accuracy can be improved by setting the boundary error ε to control the two adjacent iteration steps.

3) The RIWEKM algorithms complete the numerical computations in terms of the sum operations, which the RICEKM algorithms finish the computations according to the integration operations. In a word, the RIWEKM algorithms can be considered as the numerical realization of RICEKM algorithms by the numerical integration methods.

4. Simulation experiments

This section provides three numerical simulation examples. Here we make the assumption that, under the guidance the fuzzy inference (reasoning) block, the centroid output IT2 fuzzy set has been got by merging fuzzy rules before the type-reduction and defuzzification.

In the first instance, the FOU is composed of piecewise linear functions [19–21,27,29]. In the second instance, the FOU is made up of piecewise linear functions and Gaussian functions [14–17,30,31]. In the third instance, the FOU is composed of Gaussian functions [19–21,27,29]. The Figure 3 and Table 9 show the defined FOUs for three instances.



(a) (b) (c)

Figure 3. Graphs of FOUs for three instances; (a) instance 1, (b) instance 2, and (c) instance 3.

Table 9. FOUs for membership function expressions of three examples.

Num	FOUs for membership function expressions
1	$\underline{\mu}_{\tilde{A}_1}(x)=\max\{\begin{bmatrix}\frac{x-1}{6}, & 1\le x\le 4\\ \frac{7-x}{6}, & 4<x\le 7\\ 0, & \text{otherwise}\end{bmatrix},\begin{bmatrix}\frac{x-3}{6}, & 3\le x\le 5\\ \frac{8-x}{9}, & 5<x\le 8\\ 0, & \text{otherwise}\end{bmatrix}\}$, $\overline{\mu}_{\tilde{A}_1}(x)=\max\{\begin{bmatrix}\frac{x-1}{2}, & 1\le x\le 3\\ \frac{7-x}{4}, & 3<x\le 7\\ 0, & \text{otherwise}\end{bmatrix},\begin{bmatrix}\frac{x-2}{5}, & 2\le x\le 6\\ \frac{16-2x}{5}, & 6<x\le 8\\ 0, & \text{otherwise}\end{bmatrix}\}$
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	$\underline{\mu}_{\tilde{A}_3}(x)=\max\{0.5\exp[-\frac{(x-3)^2}{2}], 0.4\exp[-\frac{(x-6)^2}{2}]\}$, $\overline{\mu}_{\tilde{A}_3}(x)=\max\{\exp[-0.5\frac{(x-3)^2}{4}], 0.8\exp[-0.5\frac{(x-6)^2}{4}]\}$, $x\in[0,10]$

Here the error accuracy is chosen as $\varepsilon=10^{-6}$. Because the computations of $l_{\tilde{B}}$ and $r_{\tilde{B}}$ are very similar, we only calculate $l_{\tilde{B}}$ in the simulation experiments. Firstly, the RICEKM algorithms are viewed as the benchmark to calculate the left centroid end points for three instances, and they are as: $y^*_{RICEKM_1} = 3.660534$, $y^*_{RICEKM_2} = 0.445924$, and $y^*_{RICEKM_3} = 3.155741$, respectively. As for studying the proposed RIWEKM algorithms, the number of sampling points of primary variable of output IT2 FS is chosen as $N = 50:50:9000$. Then we compare the performances between the EKM algorithms, RIEKM algorithms and RIWEKM algorithms. The graphs of left centroid end points for the above mentioned algorithms are shown in Figure 4. In addition, the graphs of absolute errors between EKM algorithms, RIEKM algorithms, RIWEKM algorithms and RICEKM algorithms (benchmark) are given Figure 5.

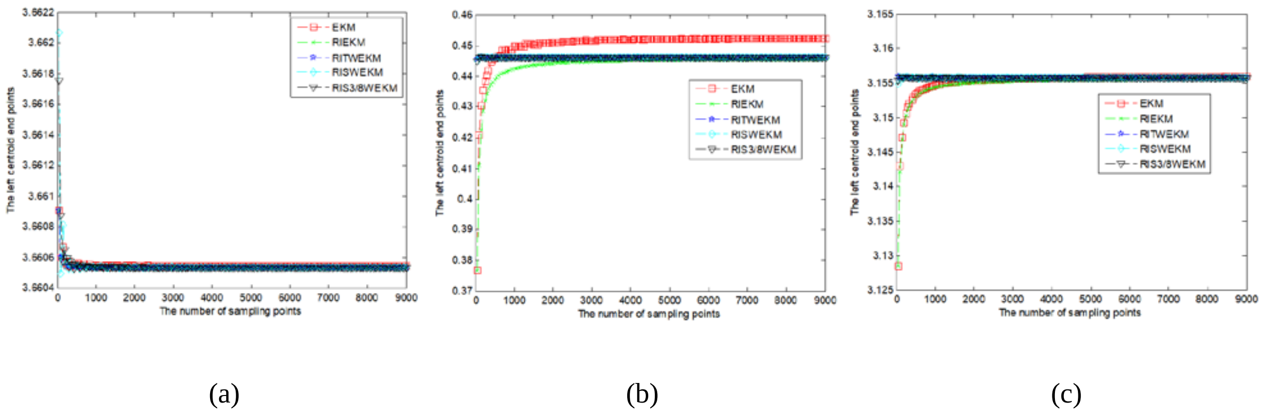


Figure 4. The left centroid end points calculated by EKM, RIEKM and RIWEKM algorithms; (a) instance 1, (b) instance 2, and (c) instance 3.

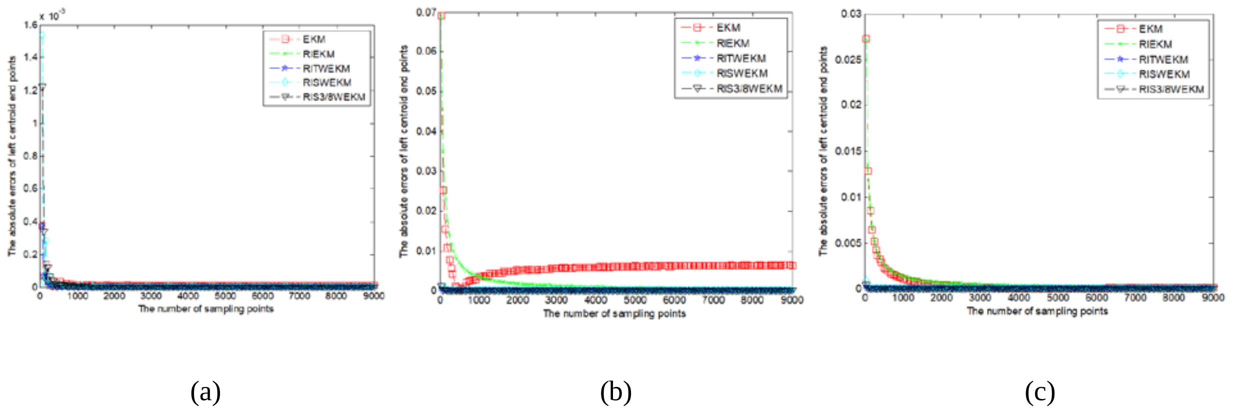


Figure 5. The absolute errors of left centroid end points between EKM algorithms, RIEKM algorithms, RIWEKM algorithms and RICEKM algorithms; (a) instance 1, (b) instance 2, and (c) instance 3.

For the sake of measuring the performances of RIWEKM algorithms qualitatively, here we define the relative errors as $| y_{EKM_i,RIEKM_i} - y^*_{RICEKM_i} | / | y^*_{RICEKM_i} | (i = 1,\cdots,3)$. Table 10 provides the mean relative errors of EKM algorithms, RIEKM algorithms and RIWEKM algorithms for the above three examples, where the last column represents the total average of mean relative errors.

For the above three examples, we can obtain the following conclusions by observing the Figures 4 and 5 and the Table 10:

1) The absolute errors of left centroid end points between the EKM algorithms, RIEKM

algorithms and RIWEKM algorithms almost all converge as the number of sampled points of primary variable of centroid output IT2 FSs increases. For the first example, the smallest absolute error (also the smallest error amplitude of variation) can be obtained by the RIEKM algorithms and RITWEKM algorithms, in which the EKM algorithms obtain the comparatively largest absolute error. For both the second and the third instances, the proposed RIWEKM algorithms can get the absolute errors (error amplitude of variations) that are obviously much less than both the EKM algorithms and RIEKM algorithm.

2) See from the Table 10, we can find the largest absolute errors of EKM algorithms and RIEKM algorithms are 1.359220% and 0.502985%, respectively, while the largest absolute error of RIWEKM algorithms is only 0.002404%, that is to say, the latter is less than two hundredths of two formers. In addition, the total average relative error of EKM algorithms and RIEKM algorithms are 0.460223%, and 0.177165%,respectively, whereas the smallest total average relative errors of RIWEKM algorithms is only 0.00065%, i.e., the latter is less than 0.04% of two formers.

3) By analyzing both the items 1) and 2) comprehensively, compared the EKM algorithms and RIEKM algorithms, we can get the conclusion that it is reasonable to adopt the appropriate RIWEKM algorithms for pursuing better error accuracy and stability.

Next, we investigate the computational time of these types of algorithms for applying them. The unrepeatable computation time of algorithms relies on the software and hardware environment, which is completely different from calculating the defuzzified values. Here the simulation experiments are completed by a dual-core CPU dell desktop with E5300@2.60GHz and 2.00GB memory. Then the comparisons of computation time of three examples are provided in Figures 6–8.

Table 10. Mean relative errors $|y_{EKM_i,RIEKM_i} - y^*_{RICEKM_i}|/|y^*_{RICEKM_i}|$ for three examples.

Algorithms	EKM	RIEKM	RITWEKM	RISWEKM	RIS3/8WEKM
Case 1	0.00000389	0.00000127	0.00000127	0.00000349	0.00000330
Case 2	0.01359220	0.00502985	0.00001328	0.00002162	0.00002404
Case 3	0.00021060	0.00027383	0.00000485	0.00000581	0.00000489
Total average	0.00460223	0.00177165	0.00000065	0.00001031	0.00001074

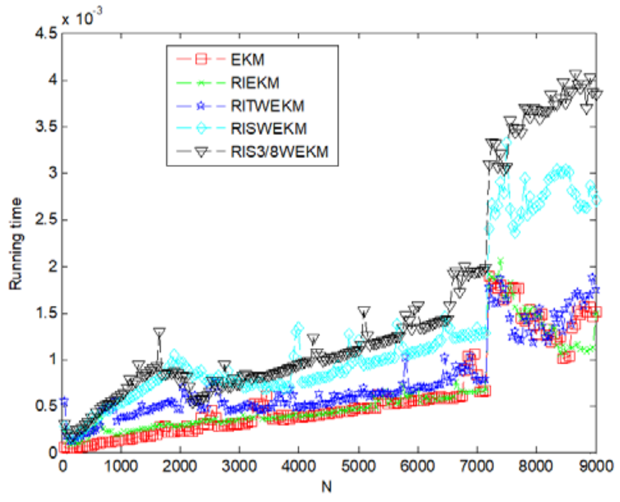


Figure 6. Comparsion of computation time for instance 1.

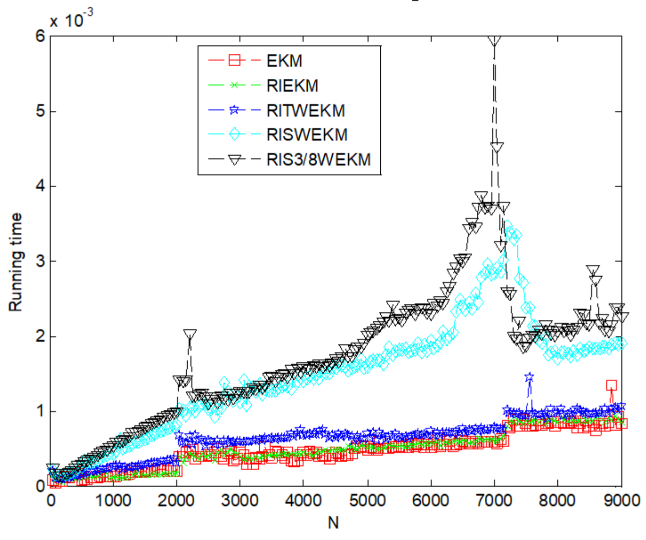


Figure 7. Comparsion of computation time for instance 2.

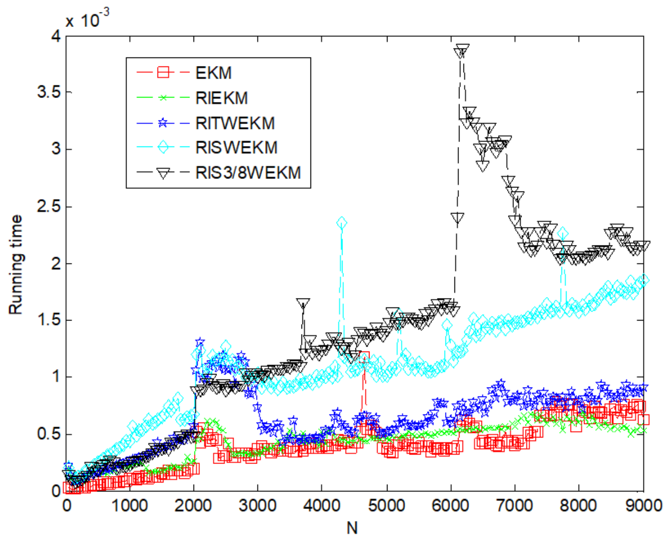


Figure 8. Comparsion of computation time for instance 3.

If the fluctuation influence of number of sampled points N were not considered, the above EKM algorithms, RIEKM algorithms and RIWEKM algorithms vary linearly with respect to N in general. Therefore, the least square regression model can be selected as $t = a + bN$ for all types of algorithms, in which t denotes the computational time, and the coefficients are provided in the following Table 11. Furthermore, the computational time difference rate is defined as:

$$(\max_{i=1,2,3}\{t_i\} - \min_{i=1,2,3}\{t_i\}) / \max_{i=1,2,3}\{t_i\}\,. \tag{38}$$

Table 11. Regression model coefficients of type-reduction algorithms.

Coefficient	EKM $a/10^{-3}$	EKM $b/10^{-3}$	RIEKM $a/10^{-3}$	RIEKM $b/10^{-3}$	RITWEKM $a/10^{-3}$	RITWEKM $b/10^{-3}$	RISWEKM $a/10^{-3}$	RISWEKM $b/10^{-3}$	RIS3/8WEKM $a/10^{-3}$	RIS3/8WEKM $b/10^{-3}$
Case 1	0.000163	-0.134676	0.000142	-0.027860	0.000138	0.125481	0.000269	0.024937	0.000385	-0.214929
Case 2	0.000087	0.085536	0.000093	0.070055	0.000093	0.215750	0.000244	0.377291	0.000286	0.435158
Case 3	0.000069	0.087296	0.000052	0.202952	0.000059	0.370410	0.000158	0.389515	0.000294	0.097664
Total mean	0.000106	0.012719	0.000096	0.081716	0.000096	0.237213	0.000223	0.263915	0.000321	0.105964

According to the Table 11 and the Figures 6–8, we can see that the computation times of EKM algorithms, RIEKM algorithms and RITWEKM algorithms are slightly better than that of RISWEKM algorithms and RIS3/8WEKM algorithms. Here we may ascribe the former three types of algorithms as the first type, and the former two types of algorithms as the second type. The computation times of the first type are better than the second type as the former has comparatively simpler weights than the latter. Generally speaking, the proposed RIWEKM algorithms have faster convergence speeds than that of EKM algorithms and RIEKM algorithms, i.e., for the RIWEKM algorithms, we can use the fewer number of sampled points to get the same computation time as the EKM algorithms and RIEKM algorithms. The overall specific computational time size relations are as: EKM<RIEKM<RITWEKM<RISEKM<RIS3/8WEKM. Furthermore, the defined computational time difference rate of three examples is between 26.47%–89.89%.

According to the above analysis, it is obviously to find that the initialization and weighting have influence for EKM algorithms to complete the centroid type-reduction of IT2 FLSs, i.e., the Newton-Cotes quadrature formulas based RIWEKM algorithms can be used for investigating the centroid TR of IT2 FLSs. If only the calculation accuracy were considered, the RITWEKM algorithms are the best option (see the Table 10). However, if the calculation accuracy and the computational time were taken in account comprehensively (see Table 10 and Figures 6–8), we suggest one use the RIEKM or RITWEKM algorithms to complete the centroid type-reduction of IT2 FLSs with piecewise linear functions as in instance 1, and adopt the RITWEKM algorithms to complete the centroid type-reduction of IT2 FLSs with both piecewise linear functions and Gaussian functions as in instance 2 and Gaussian functions as in instance 3.

Finally we should point out that the paper only focuses the performances of RIWEKM algorithms from the viewpoint theory. Four simulation examples show that, on a considerable amount of sampling points, the RIWEKM algorithms can enhance the calculation accuracy in contrast to both the EKM algorithms and RIEKM algorithms. Despite so, if the accuracy requirement is not high, the simpler EKM algorithms can get good effects, and then the RIWEKM algorithms will not show their advantages.

5. Conclusions

This paper first provides the reasonable initialization explanations for EKM algorithms, then the RIEKM algorithms are extended to different types of RIWEKM algorithms resort to the Newton-Cotes quadrature formulas. By considering the RICEKM algorithms as the baseline, five types of discrete TR algorithms are used to complete the centroid type-reduction of IT2 FLSs. Three simulation instances are given to analyze and verify that the proposed RIWEKM algorithms can get smaller absolute errors and faster convergence speeds in contrast to both the EKM algorithms and RIEKM algorithms.

There exist many important works lie ahead, including adopting the RIWEKM algorithms [32] to study the centroid type-reduction of general T2 FLSs, studying the non-iterative algorithms [21,27,29–30,33,54–57] for centroid and center-of-sets type-reduction [34,35,51] of IT2 FLSs and GT2 FLSs, the colony intelligence based algorithms [36–40] for optimizing T2 FLSs, and the forecasting, identification and control problems [22,41–50,52–56] based on T2 FLSs and so on.

Acknowledgments

The paper is supported by the National Natural Science Foundation of China (No. 61773188, No. 61973146), the Doctoral Start-up Foundation of Liaoning Province (No. 2021-BS-258), the Youth Fund of Education Department of Liaoning Province (LJKQZ2021143), and the Talent Fund Project of Liaoning University of Technology (No. xr2020002).

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. J. M. Mendel, *Uncertain rule-based fuzzy systems*, Springer International Publishing, 2017. https://doi.org/10.1007/978-3-319-51370-6
2. H. Hagras, C. Wagner, Towards the wide spread use of type-2 fuzzy logic systems in real world applications, *IEEE Comput. Intell. M.*, **7** (2012), 14–24. https://doi.org/10.1109/MCI.2012.2200621
3. J. M. Mendel, Type-2 fuzzy sets and systems: An overview, *IEEE Comput. Intell. M.*, **2** (2007), 20–29. https://doi.org/10.1109/MCI.2007.380672
4. M. H. F. Zarandi, B. Rezaee, I. B. Turksen, E. Neshat, A type-2 fuzzy rule-based expert system model for stock price analysis, *Expert Syst. Appl.*, **36** (2009), 139–154. https://doi.org/10.1016/j.eswa.2007.09.034
5. 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. https://doi.org/10.1016/j.neucom.2015.10.032
6. A. Khosravi, S. Nahavandi, Load forecasting using interval type-2 fuzzy logic systems: Optimal type reduction, *IEEE T. Ind. Electron.*, **10** (2014), 1055–1063. https://doi.org/10.1109/TII.2013.2285650
7. M. Biglarbegian, W. W. Melek, J. M. Mendel, Design of novel interval type-2 fuzzy controllers for modular and reconfigurable robots: theory and experiments, *IEEE T. Ind. Electron.*, **58** (2011), 1371–1384. https://doi.org/10.1109/TIE.2010.2049718

8. D. Z. Wang, Y. Chen, Study on permanent magnetic drive forecasting by designing Takagi Sugeno Kang type interval type-2 fuzzy logic systems, *T. I. Meas. Control*, **40** (2018), 2011–2023. https://doi.org/10.1177/0142331217694682
9. R. S. Rama, P. Latha, An effective torque ripple reduction for permanent magnet synchronous motor using ant colony optimization, *Int. J. Fuzzy Syst.*, **17** (2015), 577–584. https://doi.org/10.1007/s40815-015-0077-5
10. O. Linda, M. Manic, Interval type-2 voter design for fault tolerant systems, *Inform. Sci.*, **181** (2011), 2933–2950. https://doi.org/10.1016/j.ins.2011.03.008
11. A. Niewiadomski, On finity, countability, cardinalities, and cylindric extensions of type-2 fuzzy sets in linguistic summarization of databases, *IEEE T. Fuzzy Syst.*, **18** (2010), 532–545. https://doi.org/10.1109/TFUZZ.2010.2042719
12. D. R. Wu, J. M. Mendel, Uncertainty measures for interval type-2 fuzzy sets, *Inform. Sci.*, **177** (2007), 5378–2393. https://doi.org/10.1016/j.ins.2007.07.012
13. A. Khosravi, S. Nahavandi, D. Creighton, D. Srinivasan, Interval type-2 fuzzy logic systems for load forecasting: a comparative study, *IEEE T. Power Syst.*, **27** (2012), 1274–1282. https://doi.org/10.1109/TPWRS.2011.2181981
14. 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. https://doi.org/10.3233/JIFS-171669
15. J. M. Mendel, On KM algorithms for solving type-2 fuzzy sets problems, *IEEE T. Fuzzy Syst.*, **21** (2013), 426–446. https://doi.org/10.1109/TFUZZ.2012.2227488
16. T. Kumbasar, Revisiting Karnik-Mendel algorithms in the framework of linear fractional programming, *Int. J. Approx. Reason.*, **82** (2017), 1–21. https://doi.org/10.1016/j.ijar.2016.11.019
17. 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 T. Fuzzy Syst.*, **15** (2007), 309–320. https://doi.org/10.1109/TFUZZ.2006.882463
18. D. R. Wu, J. M. Mendel, Perceptual reasoning for perceptual computing: a similarity based approach, *IEEE T. Fuzzy Syst.*, **17** (2009), 1397–1411. https://doi.org/10.1109/TFUZZ.2009.2032652
19. D. R. Wu, J. M. Mendel, Enhanced Karnik-Mendel algorithms, *IEEE T. Fuzzy Syst.*, **17** (2009), 923–934. https://doi.org/10.1109/TFUZZ.2008.924329
20. X. W. Liu, J. M. Mendel, D. R. Wu, Study on enhanced Karnik-Mendel algorithms: initialization explanations and computation improvements, *Inform. Sci.*, **184** (2012), 75–91. https://doi.org/10.1016/j.ins.2011.07.042
21. 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. https://doi.org/10.1007/s00500-020-04998-2
22. O. Castillo, P. Melin, E. Ontiveros, C. Peraza, P. Ochoa, F. Valdez, et al., A high-speed interval type 2 fuzzy system approach for dynamic parameter adaptation in metaheuristics, *Eng. Appl. Artif. Intell.*, **85** (2019), 666–680. https://doi.org/10.1016/j.engappai.2019.07.020
23. 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, *Inform. Sci.*, **220** (2013), 149–169. https://doi.org/10.1016/j.ins.2012.01.024
24. J. M. Mendel, Type-2 fuzzy sets and systems: an overview, *IEEE Comput. Intell. Mag.*, **2** (2007), 20–29. https://doi.org/10.1109/MCI.2007.380672

25. Y. Chen, D. Z. Wang, Study on centroid type-reduction of general type-2 fuzzy logic systems with weighted enhanced Karnik-Mendel algorithms, *Soft Comput.*, **22** (2018), 1361–1380. https://doi.org/10.1007/s00500-017-2938-3
26. 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. https://doi.org/10.1007/s00500-018-3551-9
27. D. R. Wu, Approaches for reducing the computational cost of interval type-2 fuzzy logic systems: overview and comparisons, *IEEE T. Fuzzy Syst.*, **21** (2013), 80–99. https://doi.org/10.1109/TFUZZ.2012.2201728
28. T. Wang, Y. Chen, S. C. Tong, Fuzzy reasoning models and algorithms on type-2 fuzzy sets, *Int. J. Innov. Comput. Inform. Control*, **24** (2008), 2451–2460.
29. J. W. Li, R. John, S. Coupland, G. Kendall, On Nie-Tan operator and type-reduction of interval type-2 fuzzy sets, *IEEE T. Fuzzy Syst.*, **26** (2018), 1036–1039. https://doi.org/10.1109/TFUZZ.2017.2666842
30. S. Greenfield, F. Chiclana, Accuracy and complexity evaluation of defuzzification strategies for the discretised interval type-2 fuzzy set, *Int. J. Approx. Reason.*, **54** (2013), 1013–1033. https://doi.org/10.1016/j.ijar.2013.04.013
31. E. Ontiveros-Robles, P. Melin, O. Castillo, New methodology to approximate type-reduction based on a continuous root-finding karnik mendel algorithm, *Algorithms*, 1**0** (2017), 77–96. https://doi.org/10.3390/a10030077
32. Y. Chen, J. X. Wu, J. Lan, Study on reasonable initialization enhanced Karnik-Mendel algorithms for centroid type-reduction of interval type-2 fuzzy logic systems, AIMS Math., **5** (2020), 6149–6168. https://doi.org/10.3934/math.2020395
33. J. M. Mendel, X. W. Liu, Simplified interval type-2 fuzzy logic systems, *IEEE T. Fuzzy Syst.*, **21** (2013), 1056–1069. https://doi.org/10.1109/TFUZZ.2013.2241771
34. M. A. Khanesar, A. Jalalian, O. Kaynak, H. Gao, Improving the speed of center of sets type reduction in interval type-2 fuzzy systems by eliminating the need for sorting, *IEEE T. Fuzzy Syst.*, **25** (2017), 1193–1206. https://doi.org/10.1109/TFUZZ.2016.2602392
35. J. M. Mendel, General type-2 fuzzy logic systems made simple: A tutorial, *IEEE T. Fuzzy Syst.*, **22** (2014), 1162–1182. https://doi.org/10.1109/TFUZZ.2013.2286414
36. C. H. Hsu, C. F. Juang, Evolutionary robot wall-following control using type-2 fuzzy controller with species-de-activated continuous ACO, *IEEE T. Fuzzy Syst.*, **21** (2013), 100–112. https://doi.org/10.1109/TFUZZ.2012.2202665
37. Y. Chen, D. Z. Wang, W. Ning, Forecasting by TSK general type-2 fuzzy logic systems optimized with genetic algorithms, *Optim. Control Appl. Method.*, **39** (2018), 393–409. https://doi.org/10.1002/oca.2353
38. F. Gaxiola, P. Melin, F. Valdez, J. R. Castro, O. Castillo, Optimization of type-2 fuzzy weights in backpropagation learning for neural networks using GAs and PSO, *Appl. Soft Comput.*, **38** (2016), 860–871. https://doi.org/10.1016/j.asoc.2015.10.027
39. Y. Chen, D. Z. Wang, Forecasting by general type-2 fuzzy logic systems optimized with QPSO algorithms, *Int. J. Control Autom. Syst.*, **15** (2017), 2950–2958. https://doi.org/10.1007/s12555-017-0793-0
40. Y. Maldonado, O. Castillo, P. Melin, Particle swarm optimization of interval type-2 fuzzy systems for FPGA applications, *Appl. Soft Comput.*, 1**3** (2013), 496–508. https://doi.org/10.1016/j.asoc.2012.08.032

41. E. Ontiveros-Robles, P. Melin, O. Castillo, Comparative analysis of noise robustness of type 2 fuzzy logic controllers, *Kybernetika*, **54** (2018), 175–201. https://doi.org/10.14736/kyb-2018-1-0175
42. L. Cervantes, O. Castillo, Type-2 fuzzy logic aggregation of multiple fuzzy controllers for airplane flight control, *Inform. Sci.*, **324** (2015), 247–256. https://doi.org/10.1016/j.ins.2015.06.047
43. O. Castillo, L. Amador-Angulo, J. R. Castro, M. Garcia-Valdez, 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, *Inform. Sci.*, **354** (2016), 257–274. https://doi.org/10.1016/j.ins.2016.03.026
44. C. W. Tao, J. S. Taur, C. W. Chang, Y. H. Chang, Simplified type-2 fuzzy sliding controller for wing rocket system, *Fuzzy Set. Syst.*, **207** (2012), 111–129. https://doi.org/10.1016/j.fss.2012.02.015
45. D. R. Wu, J. M. Mendel, Recommendations on designing practical interval type-2 fuzzy systems, *Eng. Appl. Artif. Intell.*, **85** (2019), 182–193. https://doi.org/10.1016/j.engappai.2019.06.012
46. S. C. Tong, Y. M. Li, Observer-based adaptive fuzzy backstepping control of uncertain pure-feedback systems, *Sci. China Inform. Sci.*, **57** (2014), 1–14. https://doi.org/10.1007/s11432-013-5043-y
47. M. Deveci, I. Z. Akyurt, S. Yavuz, GIS-based interval type-2 fuzzy set for public bread factory site selection, *J. Enterp. Inform. Manag.*, **31** (2018), 820–847. https://doi.org/10.1108/JEIM-02-2018-0029
48. S. C. Tong, Y. M. Li, Robust adaptive fuzzy backstepping output feedback tracking control for nonlinear system with dynamic uncertainties, *Sci. China Inform. Sci.*, **53** (2010), 307–324. https://doi.org/10.1007/s11432-010-0031-y
49. F. Y. Wang, H. Mo, Some fundamental issues on type-2 fuzzy sets, *Acta Autom. Sin.*, **43** (2017), 1114–1141.
50. H. Mo, F. Y. Wang, M. Zhou, R. Li, Z. Xiao, Footprint of uncertainty for type-2 fuzzy sets, *Inform. Sci.*, **272** (2014), 96–110. https://doi.org/10.1016/j.ins.2014.02.092
51. Y. Chen, J. X. Yang, Study on center-of-sets type-reduction of interval type-2 fuzzy logic systems with noniterative algorithms, *J. Intell. Fuzzy Syst.*, **40** (2021), 11099–11106. https://doi.org/10.3233/JIFS-202264
52. X. Tao, J. Yi, Z. Pu, T. Xiong, Robust adaptive tracking control for hypersonic vehicle based on interval type-2 fuzzy logic system and small-gain approach, *IEEE T. Cybernetics*, **51** (2021), 2504–2517. https://doi.org/10.1109/TCYB.2019.2927309
53. Y. Chen, J. X. Yang, Design of back propagation optimized Nagar-Bardini structure-based interval type-2 fuzzy logic systems for fuzzy identification, *T. I. Meas. Control*, **43** (2021), 2780–2787. https://doi.org/10.1177/01423312211006635
54. L. Wu, F. Qian, L. Wang, X. Ma, An improved type-reduction algorithm for general type-2 fuzzy sets, *Inform. Sci.*, 2022.
55. Y. Chen, Study on weighted-based noniterative algorithms for computing the centroids of general type-2 fuzzy sets, *Int. J. Fuzzy Syst.*, **24** (2022), 587–606. https://doi.org/10.1007/s40815-021-01166-y
56. C. Chen, D. Wu, J. M. Garibaldi, R. I. John, J. Twycross, J. M. Mendel, A comprehensive study of the efficiency of type-reduction algorithms, *IEEE T. Fuzzy Syst.*, **29** (2020), 1556–1566. https://doi.org/10.1109/TFUZZ.2020.2981002