

Research Article

Study on Centroid Type-Reduction of Interval Type-2 Fuzzy Logic Systems Based on Noniterative Algorithms

Yang Chen 

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

Correspondence should be addressed to Yang Chen; lxychenyang@lnut.edu.cn

Received 31 October 2018; Revised 25 February 2019; Accepted 18 March 2019; Published 11 April 2019

Academic Editor: Baris Yuce

Copyright © 2019 Yang Chen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Interval type-2 fuzzy logic systems have favorable abilities to cope with uncertainties in many applications. While the block type-reduction under the guidance of inference plays the central role in the systems, Karnik-Mendel (KM) iterative algorithms are standard algorithms to perform the type-reduction; however, the high computational cost of type-reduction process may hinder them from real applications. The comparison between the KM algorithms and other alternative algorithms is still an open problem. This paper introduces the related theory of interval type-2 fuzzy sets and discusses the blocks of fuzzy reasoning, type-reduction, and defuzzification of interval type-2 fuzzy logic systems by combining the Nagar-Bardini (NB) and Nie-Tan (NT) noniterative algorithms for solving the centroids of output interval type-2 fuzzy sets. Moreover, the continuous version of NT (CNT) algorithms is proved to be accurate algorithms for performing the type-reduction. Four computer simulation examples are provided to illustrate and analyze the performances of two kinds of noniterative algorithms. The NB and NT algorithms are superior to the KM algorithms on both calculation accuracy and time, which afford the potential application value for designer and adopters of type-2 fuzzy logic systems.

1. Introduction

As we all know, the membership grades of type-1 fuzzy sets (T1 FSs) are crisp numbers. While the membership grades of type-2 fuzzy sets (T2 FSs) are themselves T1 FSs, type-2 fuzzy sets (T2 FSs) and type-2 fuzzy logic systems (T2 FLSs) are hot topics in current academic area. As the membership grades of interval type-2 fuzzy sets (IT2 FSs) are all uniformly equal to 1, they can be completely characterized by the corresponding footprint of uncertainty (FOU) composed of the upper membership function (UMF) and lower membership function (LMF). Currently, the computational relative simple IT2 FLSs have been successfully applied in many areas with high uncertainty, nonlinearity and time-varying behavior [1] like financial systems [2], computing with words [3], power systems [4–6], intelligent controllers [7–9], permanent magnetic drive [10, 11], pattern recognition [12, 13], medical systems [14], recommendation systems [15], and so on. Moreover, the performances of IT2 FLSs are superior to their T1 counterparts.

In general, an IT2 FLS is composed of five blocks as fuzzifier, rules, inference, type-reducer, and defuzzifier (see Figure 1). In fact, the principle of IT2 FLSs is similar to T1 FLSs, while the major difference between them is that there must be at least one IT2 FS in the rules of IT2 FLSs. A T1 FLSs does not contain the block of type-reduction. In addition, the type-reduction under the guidance of inference is a central block in IT2 FLSs, which mainly plays the role of transforming the output IT2 FS obtained from fuzzy reasoning [16] to the T1 FS. Next, the crisp output is acquired according to the block of defuzzification. The process of type-reduction is usually performed by the most popular computationally intensive KM iterative algorithms [17–19]; however, the complex computations make the applications of IT2 FLSs more challenge. When all the uncertainties of IT2 FSs disappear, the IT2 FLSs degenerate to the T1 FLSs.

In early times, Mendel et al. developed the KM algorithms to compute the centroids IT2 FSs or perform the centroid type-reduction (TR) of IT2 FLSs. In order to reduce the computational cost, Wu and Mendel proposed the enhanced

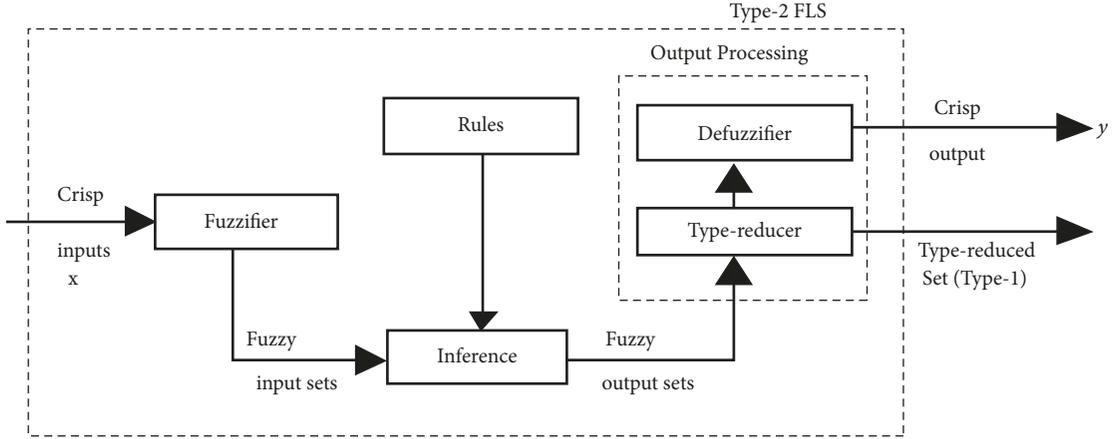


FIGURE 1: An IT2 FLS [41].

version of KM (EKM [20]) algorithms. As the continuous versions of KM or EKM algorithms (CKM or CEKM [21, 22]) were proposed, which gave a huge push for the theoretical studies of IT2 FLSs. Mendel and Liu analyzed and proved the monotonicity and superexponential convergence [23] of CKM algorithms. In recent years, Nagar and Bardini verified that IT2 FLSs based on the NB algorithms [24] have superior performances to respond to the effect of uncertainties in the systems' parameters to the IT2 FLSs based on other type-reduction (TR) algorithms like EKM, Wu-Mendel Uncertainty Bound (WM-UB [25]), Begian Melek Mendel (BMM [26, 27]), and Greenfield-Chiclana Collapsing Defuzzifier (GCCD[28]). Furthermore, Nie-Tan (NT) algorithms [29] were proved to be excellent approach to simplify IT2 FLSs. While the continuous versions of NT (CNT) algorithms [30] were verified to be accurate algorithms for solving the centroids of IT2 FSs, in a word, all these works laid enrich theoretical foundations for studying and applying the TR algorithms of IT2 FLSs.

Inspired by [16, 19, 22–24, 29–37], this paper proposes the continuous of NB and NT (CNB and CNT) algorithms and brings connections between NB, NT, and BMM algorithms. The blocks of fuzzy reasoning, centroid type-reduction, and defuzzification of IT2 FLSs are performed by combining the NB and NT algorithms. Moreover, the CNT algorithms are proved to be accurate algorithms for performing the centroid TR of IT2 FLSs. Numerical simulation examples analyze and illustrate the performances of two kinds of non-iterative algorithms for computing the centroid defuzzified values compared with the KM algorithms. The simulation studies show that the two kinds of noniterative algorithms can effectively improve both the calculation accuracy and the efficiency for obtaining the outputs of IT2 FLSs.

The rest of this paper is organized as follows. Section 2 introduces the background knowledge of IT2 FSs and IT2 FLSs. Section 3 provides the NB and NT algorithms and how to use them to obtain the outputs of IT2 FLSs. Section 4 compares the performances of two kinds of noniterative algorithms with the KM algorithms according to the numerical examples. Finally the conclusions are given in Section 5.

2. Backgrounds

2.1. IT2 FSs

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

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \frac{\mu_{\tilde{A}}(x, u)}{(x, u)} \quad (1)$$

where $x \in X$, $u \in J_x \subseteq [0, 1]$, and X denotes the universe of the primary variable x of \tilde{A} . Here \iint represents all the admissible x and u . The point-value representation of \tilde{A} is

$$\tilde{A} = \{(x, u), \mu_{\tilde{A}}(x, u) \mid \forall x \in X, \forall u \in [0, 1]\}. \quad (2)$$

Definition 2. The secondary MF of \tilde{A} is also called a vertical slice of $\mu_{\tilde{A}}(x, u)$, i.e., [38]

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

where $0 \leq f_{x'}(u) \leq 1$, and $\mu_{\tilde{A}}(x')$ denotes the secondary MF of \tilde{A} . The secondary membership grades of IT2 FSs all equal 1, that is to say, for any $x = x'$, $f_{x'}(u) \equiv 1$.

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

$$\text{FOU}(\tilde{A}) = \bigcup_{x \in X} J_x = \{(x, u) \mid u \in J_x \subseteq [0, 1]\} \quad (4)$$

where J_x represents the primary membership of \tilde{A} ; here the lower MF (LMF) $\underline{\mu}_{\tilde{A}}(x)$ and upper MF (UMF) $\overline{\mu}_{\tilde{A}}(x)$ comprise the FOU, where [39]

$$\underline{\mu}_{\tilde{A}}(x) = \text{LMF}(\tilde{A}) \quad (5)$$

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

$$\overline{\mu}_{\tilde{A}}(x) = \text{UMF}(\tilde{A}) \quad (6)$$

$$= \sup \{u \mid u \in [0, 1], \mu_{\tilde{A}}(x, u) > 0\}.$$

2.2. *IT2 FLSs*. Generally speaking, there are two major architectures for IT2 FLSs, Mamdani type [5, 31] and Takagi-Sugeno-Kang (TSK) type [10, 40]. The paper pays attention to the Mamdani type. Without loss of generality, consider a Mamdani IT2 FLS with p inputs $x_1 \in X_1, \dots, x_p \in X_p$ and one output $y \in Y$. The system can be completely characterized by M fuzzy rules, where the l th fuzzy rule is of the form:

$$\begin{aligned} \tilde{R}^l: \text{If } x_1 \text{ is } \tilde{F}_1^l \text{ and } \dots \text{ and } x_p \text{ is } \tilde{F}_p^l, \\ \text{then } y \text{ is } \tilde{G}^l \quad (l = 1, \dots, M). \end{aligned} \quad (7)$$

The process of inference [5, 16, 31, 41] is as follows.

In order to simplify the expressions, here we adopt the singleton fuzzifier [41]; i.e., the input measurements are modeled as type-0 FSs (singletons). The fuzzy implication relation between each fuzzy rule is as

$$\tilde{R}^l : \tilde{F}_1^l \times \dots \times \tilde{F}_p^l \longrightarrow \tilde{G}^l = \tilde{A}^l \longrightarrow \tilde{G}^l \quad (8)$$

The MF of (8) is as

$$\begin{aligned} \mu_{\tilde{R}^l}(x, y) &= \mu_{\tilde{A}^l \longrightarrow \tilde{G}^l}(x, y) \\ &= \mu_{\tilde{F}_1^l}(x_1) \cap \dots \cap \mu_{\tilde{F}_p^l}(x_p) \cap \mu_{\tilde{G}^l}(y) \\ &= \left[\bigcap_{i=1}^p \mu_{\tilde{F}_i^l}(x_i) \right] \cap \mu_{\tilde{G}^l}(y) \end{aligned} \quad (9)$$

where \cap represents the minimum or product t-norm operation [42].

Then the output T2 FS of each fuzzy rule is $\tilde{B}^l = \tilde{A}_x \circ \tilde{R}^l$, whose MF

$$\begin{aligned} \mu_{\tilde{B}^l}(y) &= \mu_{A_x \circ \tilde{R}^l}(y) = \bigcup_{x \in X} [\mu_{A_x}(x) \cap \mu_{\tilde{A}^l \longrightarrow \tilde{G}^l}(x, y)] \\ &= \mu_{\tilde{G}^l}(y) \cap \left[\bigcap_{i=1}^p \mu_{\tilde{F}_i^l}(x_i) \right] = \mu_{\tilde{G}^l}(y) \cap F^l(x') \end{aligned} \quad (10)$$

where \circ denotes the composition operation and \cup represents the maximum t-conorm operation [41]. Here $F^l(x')$ is defined as the firing interval of each fuzzy rule, when $x = x'$ and compute F^l :

$$\begin{aligned} F^l(x') &\equiv \left[\underline{f}^l(x'), \overline{f}^l(x') \right] \\ \underline{f}^l(x') &\equiv \bigcap_{i=1}^p \mu_{\tilde{F}_i^l}(x_i) \\ \overline{f}^l(x') &\equiv \bigcap_{i=1}^p \overline{\mu}_{\tilde{F}_i^l}(x_i) \end{aligned} \quad (11)$$

While adopting the popular centroid TR [32], the firing output set \tilde{B}^l is generated from by each fuzzy rule and the corresponding consequent IT2 FS, i.e.,

$$\tilde{B}^l : \begin{cases} FOU(\tilde{B}^l) = \left[\underline{\mu}_{\tilde{B}^l}(y | x'), \overline{\mu}_{\tilde{B}^l}(y | x') \right] \\ \underline{\mu}_{\tilde{B}^l}(y | x') = \underline{f}^l(x') * \underline{\mu}_{\tilde{G}^l}(y) \\ \overline{\mu}_{\tilde{B}^l}(y | x') = \overline{f}^l(x') * \overline{\mu}_{\tilde{G}^l}(y) \end{cases} \quad (12)$$

where $*$ also denotes the minimum or product t-norm operation.

Next, the final output \tilde{B} can be obtained by merging all the rule firing output sets \tilde{B}^l , i.e.,

$$\tilde{B} : \begin{cases} FOU(\tilde{B}) = \left[\underline{\mu}_{\tilde{B}}(y | x'), \overline{\mu}_{\tilde{B}}(y | x') \right] \\ \underline{\mu}_{\tilde{B}}(y | x') = \underline{\mu}_{\tilde{B}^1}(y | x') \vee \dots \vee \underline{\mu}_{\tilde{B}^M}(y | x') \\ \overline{\mu}_{\tilde{B}}(y | x') = \overline{\mu}_{\tilde{B}^1}(y | x') \vee \dots \vee \overline{\mu}_{\tilde{B}^M}(y | x') \end{cases} \quad (13)$$

where \vee denotes the maximum operation. Then the type-reduced set $Y_C(x')$ can be obtained by computing the centroid $C_{\tilde{B}}$ of \tilde{B} , i.e.,

$$Y_C(x') = C_{\tilde{B}}(x') = \frac{1}{[\underline{l}_{\tilde{B}}(x'), \overline{r}_{\tilde{B}}(x')]} \quad (14)$$

where the two points $\underline{l}_{\tilde{B}}(x')$ and $\overline{r}_{\tilde{B}}(x')$ can be calculated by popular type-reduction algorithms like KM [19, 41], EKM [20], and weighted EKM (WEKM [18, 22]).

3. NB and NT Algorithms

In this section, we provide two kinds of noniterative for performing the centroid TR of IT2 FLSs; they are the NB and NT algorithms [24, 29, 30, 43].

3.1. *NB Algorithms*. The NB algorithms [24] offered a closed form of TR. After the process of fuzzy reasoning [16], let \tilde{B} be the obtained output IT2 FS and the primary variable y be equally discretized into N points that satisfies $y_1 < y_2 < \dots < y_N$; then the two endpoints of centroid interval can be computed as

$$c_l = \frac{\sum_{i=1}^N y_i \underline{\mu}_{\tilde{B}}(y_i)}{\sum_{i=1}^N \underline{\mu}_{\tilde{B}}(y_i)}, \quad (15)$$

$$\text{and } c_r = \frac{\sum_{i=1}^N y_i \overline{\mu}_{\tilde{B}}(y_i)}{\sum_{i=1}^N \overline{\mu}_{\tilde{B}}(y_i)}.$$

The defuzzified output is computed as

$$y_{NB} = \frac{(c_l + c_r)}{2}. \quad (16)$$

Similar to the CKM (or CEKM) algorithms [21–23, 34, 44], the continuous form of NB (CNB) algorithms can be

adopted for studying the theoretical property of TR and defuzzification of IT2 FLSs.

Suppose that a and b are the left and right endpoints of primary variable y , and $y_1 < y_2 < \dots < y_N$; then the continuous form of (15) is as

$$c_l = \frac{\int_a^b y \underline{\mu}_{\tilde{B}}(y) dy}{\int_a^b \underline{\mu}_{\tilde{B}}(y) dy}, \quad (17)$$

$$\text{and } c_r = \frac{\int_a^b y \overline{\mu}_{\tilde{B}}(y) dy}{\int_a^b \overline{\mu}_{\tilde{B}}(y) dy}.$$

Unlike the traditional computational intensive KM algorithms [17–21], the NB algorithms may be suitable for real time applications. For the centroid interval, the two end points can be computed without iterations. Furthermore, the output is simply composed of the linear combination of two T1 FLSs: one constructed from the LMFs and the other constructed from the UMFs.

3.2. NT Algorithms. The NT algorithms also provide a closed form of TR. Supposing that the primary variable y is equally discretized into N points which satisfies $y_1 < y_2 < \dots < y_N$, then the output centroid interval can be calculated as

$$y_{NT} = \frac{\sum_{i=1}^N y_i [\underline{\mu}_{\tilde{B}}(y_i) + \overline{\mu}_{\tilde{B}}(y_i)]}{\sum_{i=1}^N [\underline{\mu}_{\tilde{B}}(y_i) + \overline{\mu}_{\tilde{B}}(y_i)]}. \quad (18)$$

Then the continuous form of (18) can be as

$$y_{CNT} = \frac{\int_a^b y [\underline{\mu}_{\tilde{B}}(y) + \overline{\mu}_{\tilde{B}}(y)] dy}{\int_a^b [\underline{\mu}_{\tilde{B}}(y) + \overline{\mu}_{\tilde{B}}(y)] dy}. \quad (19)$$

The most recent studies on calculating the centroid of IT2 FSs illustrate that the CNT algorithm [30] is an accurate approach. Next we prove that the CNT algorithm is accurate method for performing the centroid TR and defuzzification of IT2 FLSs according to simple statements and explanations as follows.

As we all know, the random sampling is an approximate method for performing the TR and defuzzification of IT2 FLSs. As the number of sampling points approaches infinity, it is reasonable to believe that the random sampling becomes an accurate TR and defuzzification.

Theorem 4. *As the number of sampling approaches infinity, a random sampling algorithm can perform the accurate centroid TR and defuzzification of IT2 FLSs.*

Proof. For the discrete output IT2 FS \tilde{B} defined in the universe of discourse of Y , let m be the number of horizontal slices along the μ -axis and n be the number of vertical slices along the y -axis. Because each T1 FS contains n vertical slices, the total number of embedded T1 FSs in \tilde{B} is m^n .

Suppose that $\overline{\mu}(y)$ and $\underline{\mu}(y)$ is the UMF and LMF of FOU, respectively. In addition, we select k embedded T1 FSs from \tilde{B} randomly, and let $\mu_i(y)$ be the MF of the i th embedded set.

Then we aggregate k embedded T1 FS, i.e.,

$$\sum_{i=1}^k \mu_i(y) = \sum_{j=1}^n \sum_{i=1}^k \mu_i(y_j) \quad (20)$$

therefore

$$\lim_{k \rightarrow \infty} \frac{1}{k} \sum_{i=1}^k \mu_i(y) = \lim_{k \rightarrow \infty} \frac{1}{k} \sum_{i=1}^k \sum_{j=1}^n \mu_i(y_j). \quad (21)$$

Here $\mu_i(y_j)$ is a random value uniformly distributed in the interval $[\underline{\mu}_i(y_j), \overline{\mu}_i(y_j)]$, so the right side of (21) can be turned to

$$\lim_{k \rightarrow \infty} \frac{1}{k} \sum_{i=1}^k \sum_{j=1}^n \mu_i(y_j) = \frac{1}{m} \sum_{k=1}^m \sum_{j=1}^n \mu_k(y_j) \quad (22)$$

For (22), as the right side of which is a random sampling with $k \rightarrow \infty$, the right side becomes exactly the aggregation of MFs of all embedded FSs of \tilde{B} . Because m/k is just a constant, it has no effect for calculating the centroid of \tilde{B} (see (19)).

When \tilde{B} is a continuous IT2 FS, we simply substitute $\sum_{j=1}^n$ and $\sum_{k=1}^m$ by $\int_{y_{\min}}^{y_{\max}}$ and $\int_{\underline{\mu}(y)}^{\overline{\mu}(y)}$, respectively. Then we get

$$\lim_{k \rightarrow \infty} \frac{m}{k} \sum_{i=1}^k \mu_i(y) = \int_{y_{\min}}^{y_{\max}} \int_{\underline{\mu}(y)}^{\overline{\mu}(y)} \mu(y) d\mu dy. \quad (23)$$

□

Theorem 5. *A representative T1 embedded FS of \tilde{B} has the MF*

$$\mu^*(y) = \frac{1}{2} [\underline{\mu}(y) + \overline{\mu}(y)] \quad (24)$$

and the centroid of \tilde{B} can be computed as

$$C(\tilde{B}) = \frac{\int_{y_{\min}}^{y_{\max}} y \mu^*(y) dy}{\int_{y_{\min}}^{y_{\max}} \mu^*(y) dy} \quad (25)$$

Proof. For the discrete output IT2 FS \tilde{B} , we choose k embedded T1 FSs from \tilde{B} randomly. Suppose that $\mu_i(y)$ is the MF of i th embedded FS and y_j is an arbitrary vertical slice. In addition, $\mu_i(y_j)$ can be denoted as $\mu_i(y_j) = \underline{\mu}(y_j) + \omega_i[\overline{\mu}(y_j) - \underline{\mu}(y_j)]$, among which let ω_i be a random number uniformly distributed within $[0, 1]$. Thus, we get

$$\begin{aligned} \sum_{i=1}^k \mu_i(y_j) &= \sum_{i=1}^k [\underline{\mu}(y_j) + \omega_i(\overline{\mu}(y_j) - \underline{\mu}(y_j))] \\ &= k\underline{\mu}(y_j) + [\overline{\mu}(y_j) - \underline{\mu}(y_j)] \sum_{i=1}^k \omega_i \end{aligned} \quad (26)$$

As ω_i is uniformly distributed with $[0, 1]$, hence

$$\lim_{k \rightarrow \infty} \frac{1}{k} \sum_{i=1}^k \omega_i = \frac{k}{2}. \quad (27)$$

Then we have

$$\lim_{k \rightarrow \infty} \frac{1}{k} \sum_{i=1}^k \mu_i(y_j) = \frac{1}{2} [\underline{\mu}(y_j) + \overline{\mu}(y_j)] \quad (28)$$

Due to the fact that y_j is an arbitrary vertical slice, (28) holds for all y . Thus, we get

$$\lim_{k \rightarrow \infty} \frac{1}{k} \sum_{i=1}^k \mu_i(y) = \frac{1}{2} [\underline{\mu}(y) + \overline{\mu}(y)] \quad (29)$$

By mean of the above analysis (see (29) and Theorem 4), we know that $\mu^*(y) = (1/2)[\underline{\mu}(y) + \overline{\mu}(y)]$ is the MF of a representative T1 FS and it can perform the accurate centroid TR and defuzzification of IT2 FLSs.

Here we adapt Theorems 4 and 5 from [30]. It is simply to substitute the sum operation $\sum_{i=1}^n$ by the integral operation $\int_{y_{\min}}^{y_{\max}}$ to confirm the proofs to a continuous IT2 FS. The CNT algorithm performs the accurate centroid TR and defuzzification as it is equivalent to an exhaustive TR.

In fact, both the NB and the NT algorithms are special cases of BMM algorithms. And the BMM algorithms have been extensively studied for their stability and robustness [26, 27]. Next, we give an explanation as follows.

The closed form BMM TR algorithms calculate the output centroid interval as

$$y_{BMM} = m \frac{\sum_{i=1}^N y_i \underline{\mu}_{\overline{B}}(y_i)}{\sum_{i=1}^N \underline{\mu}_{\overline{B}}(y_i)} + n \frac{\sum_{i=1}^N y_i \overline{\mu}_{\overline{B}}(y_i)}{\sum_{i=1}^N \overline{\mu}_{\overline{B}}(y_i)} \quad (30)$$

Here m and n are adjustable coefficients.

Comparing (30), (15), and (16), we find that the NB and BMM algorithms are the same while $m = m_{NB} = 1/2$ and $n = n_{NB} = 1/2$. Next, let us consider the NT algorithms. Equation (18) can be transformed to

$$\begin{aligned} y_{NT} &= \frac{\sum_{i=1}^N \underline{\mu}_{\overline{B}}(y_i)}{\sum_{i=1}^N [\underline{\mu}_{\overline{B}}(y_i) + \overline{\mu}_{\overline{B}}(y_i)]} \times \frac{\sum_{i=1}^N y_i \underline{\mu}_{\overline{B}}(y_i)}{\sum_{i=1}^N \underline{\mu}_{\overline{B}}(y_i)} \\ &+ \frac{\sum_{i=1}^N \overline{\mu}_{\overline{B}}(y_i)}{\sum_{i=1}^N [\underline{\mu}_{\overline{B}}(y_i) + \overline{\mu}_{\overline{B}}(y_i)]} \times \frac{\sum_{i=1}^N y_i \overline{\mu}_{\overline{B}}(y_i)}{\sum_{i=1}^N \overline{\mu}_{\overline{B}}(y_i)} \\ &= m_{NT} \frac{\sum_{i=1}^N y_i \underline{\mu}_{\overline{B}}(y_i)}{\sum_{i=1}^N \underline{\mu}_{\overline{B}}(y_i)} + n_{NT} \frac{\sum_{i=1}^N y_i \overline{\mu}_{\overline{B}}(y_i)}{\sum_{i=1}^N \overline{\mu}_{\overline{B}}(y_i)} \end{aligned} \quad (31)$$

where

$$m_{NT} = \frac{\sum_{i=1}^N \underline{\mu}_{\overline{B}}(y_i)}{\sum_{i=1}^N [\underline{\mu}_{\overline{B}}(y_i) + \overline{\mu}_{\overline{B}}(y_i)]} \quad (32)$$

$$n_{NT} = \frac{\sum_{i=1}^N \overline{\mu}_{\overline{B}}(y_i)}{\sum_{i=1}^N [\underline{\mu}_{\overline{B}}(y_i) + \overline{\mu}_{\overline{B}}(y_i)]}. \quad (33)$$

Comparing (31) and (16), we find that the NT and BMM algorithms are the same when $m = m_{NT}$ and $n = n_{NT}$. The above analysis provides connections between BMM algorithm and NB and NT algorithms. And both NB and NT algorithms are special cases of BMM algorithms. \square

4. Simulation

In this section, four computer simulation examples are provided to illustrate the performances of two kinds of noniterative algorithms compared with the KM algorithms. Before performing the TR, we suppose that the FOU of output IT2 FS is known according to merging or weighting all the fuzzy rules under the guidance of inference [16]. For the first example, the FOU is bounded by the piecewise linear functions [22, 29, 30, 36]. For the second example, the FOU is composed of piecewise linear functions and Gaussian functions [22, 23, 27, 30]. For the third example, the FOU is bounded by the Gaussian functions [22, 29, 30, 36]. For the last example, the whole FOU is the symmetric Gaussian MF with uncertainty derivation [22, 23, 27, 30].

Here we perform the centroid TR for the IT2 FLSs. Suppose that the primary variable of the output IT2 FS is the letter x . Furthermore, select the number of sampling points of primary variable as $N = 50 : 50 : 4000$. Figure 2 gives the FOU graphs for four examples. In addition, Table 1 provides the FOU MF expressions for four examples. Considering both Figure 2 and Table 1, we can clearly find that the nonlinearity of these FOUs gets greater from Example one to Example four. Then let us compare the performances of NB, NT, and KM algorithms on the benchmark of CNT algorithms.

In the first example, the CNT algorithms are first adopted to calculate the benchmark defuzzified value $y_1^* = 4.320794$. Then the graph of defuzzified values for the NB, NT, and KM algorithms is shown in Figure 3(a); moreover, the functional graph of absolute errors between CNT and NB, NT, and KM algorithms is provided in Figure 3(b).

For the second example, we adopt the CNT algorithms to calculate the benchmark defuzzified value $y_2^* = 3.714087$. Then we provide the graph of defuzzified values for the NB, NT, and KM algorithms in Figure 3(a) and the functional graph of absolute errors between CNT and NB, NT, and KM algorithms in Figure 3(b).

In the third example, the CNT algorithms are first adopted to calculate the benchmark defuzzified value $y_3^* = 4.395260$. Then the graph of defuzzified values for the NB, NT, and KM algorithms is shown in Figure 5(a); moreover, the functional graph of absolute errors between CNT and NB, NT, and KM algorithms is provided in Figure 5(b).

For the fourth example, we adopt the CNT algorithms to calculate the benchmark defuzzified value $y_4^* = 4.999999$. Then we provide the graph of defuzzified values for the NB, NT, and KM algorithms in Figure 6(a) and the functional graph of absolute errors between the CNT and NB, NT, and KM algorithms in Figure 6(b).

Then we measure the performances of NB, NT, and KM algorithms for the above examples. As the number of sampling points is chosen as $N = 50 : 50 : 4000$, we define the

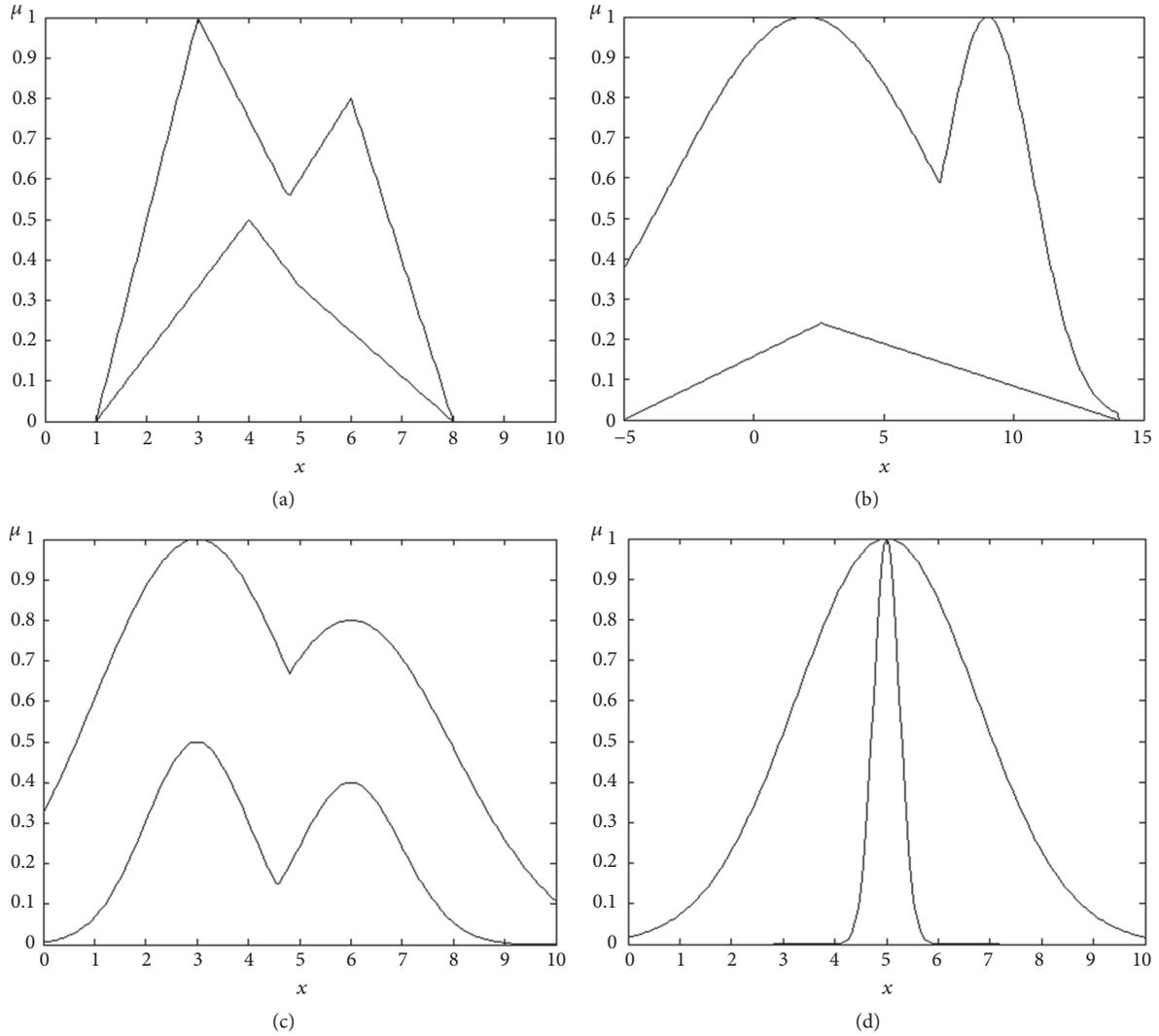


FIGURE 2: FOU graphs of four examples; (a) example 1, (b) example 2, (c) example 3, and (d) example 4.

relative error as $|y_i - y_i^*|/|y_i^*|$ ($i = 1, 2, 3$). Table 2 gives the mean relative errors of NB, NT, and KM algorithms, where the last column is the total average mean relative errors of them.

From Figures 3–6 and Table 2, the following conclusions can be obtained:

(1) The absolute errors of NB, NT, and KM algorithms all converge to some extent as the number of sampling points increases. For both examples one and three, the NT algorithms can obtain the smallest absolute error, whereas the NB algorithms obtain the largest absolute error. For example three, the NT algorithms also obtain the smallest absolute error, whereas the KM algorithms obtain the largest absolute error. For example four, these three types of algorithms almost obtain the same absolute error (see Figure 6(b), the unit of vertical axis is 10^{-6}). Moreover, the amplitude of variation of absolute errors for them three is really not big.

(2) See Table 2; the largest mean relative error of NT algorithms is 0.013729%, while the largest mean relative errors

for the NB and KM algorithms are 1.664610% and 2.211606%, respectively. Furthermore, the total average mean relative error of NT algorithms is 0.020351%, while the total average mean relative errors for the NB and NT algorithms are 0.652444% and 0.694472%, respectively.

(3) Considering the above items (1) and (2) comprehensively, we believe that the NT algorithms can obtain the best accuracy, while the NB algorithms take second place.

Next we investigate the computation time for the above algorithms. The unit of time is chosen as the second (s). As for calculating the defuzzified values, the computation time results rely on the specific hardware and software environments. In addition, it is unrepeatable for the results. In the paper, the simulations are performed by a dual-core CPU dell desktop with E5300@2.60GHz and 2.00 GB memory, Windows XP operation system, and the programming software of Matlab 2013a. Then we give the computation time comparisons for the four examples in the Figures 7–10.

TABLE 1: FOU MF expressions of four examples.

Num	FOU MF expressions
1	$\underline{\mu}_{\bar{A}_1}(x) = \max \left\{ \begin{array}{l} \left[\frac{x-1}{6}, 1 \leq x \leq 4 \right] \\ \left[\frac{7-x}{6}, 4 < x \leq 7 \right] \\ 0, \text{ otherwise} \end{array} \right\}, \left\{ \begin{array}{l} \left[\frac{x-3}{6}, 3 \leq x \leq 5 \right] \\ \left[\frac{8-x}{9}, 5 < x \leq 8 \right] \\ 0, \text{ otherwise} \end{array} \right\}$ $\bar{\mu}_{\bar{A}_1}(x) = \max \left\{ \begin{array}{l} \left[\frac{x-1}{2}, 1 \leq x \leq 3 \right] \\ \left[\frac{7-x}{4}, 3 < x \leq 7 \right] \\ 0, \text{ otherwise} \end{array} \right\}, \left\{ \begin{array}{l} \left[\frac{x-2}{5}, 2 \leq x \leq 6 \right] \\ \left[\frac{16-2x}{5}, 6 < x \leq 8 \right] \\ 0, \text{ otherwise} \end{array} \right\}$
2	$\underline{\mu}_{\bar{A}_2}(x) = \begin{cases} \frac{0.6(x+5)}{19}, & -5 \leq x \leq 2.6 \\ \frac{0.4(14-x)}{19}, & 2.6 < x \leq 14 \end{cases}$ $\bar{\mu}_{\bar{A}_2}(x) = \begin{cases} \exp \left[-\frac{1}{2} \left(\frac{x-2}{5} \right)^2 \right], & -5 \leq x \leq 7.185 \\ \exp \left[-\frac{1}{2} \left(\frac{x-9}{1.75} \right)^2 \right], & 7.185 < x \leq 14 \end{cases}$
3	$\underline{\mu}_{\bar{A}_3}(x) = \max \left\{ 0.5 \exp \left[-\frac{(x-3)^2}{2} \right], 0.4 \exp \left[-\frac{(x-6)^2}{2} \right] \right\}$ $\bar{\mu}_{\bar{A}_3}(x) = \max \left\{ \exp \left[-0.5 \frac{(x-3)^2}{4} \right], 0.8 \exp \left[-0.5 \frac{(x-6)^2}{4} \right] \right\}$ <p style="text-align: center;">$x \in [0, 10].$</p>
4	$\underline{\mu}_{\bar{A}_4}(x) = \exp \left[-\frac{1}{2} \left(\frac{x-3}{0.25} \right)^2 \right]$ $\bar{\mu}_{\bar{A}_4}(x) = \exp \left[-\frac{1}{2} \left(\frac{x-3}{1.75} \right)^2 \right]$ <p style="text-align: center;">$x \in [0, 10].$</p>

TABLE 2: Mean relative errors $|y_i - y_i^*|/|y_i^*|$ ($i = 1, 2, 3$) for the number of sampling points $N = 50 : 50 : 4000$.

Algorithms	NB	NT	KM
Example one	0.00366072	0.00000043	0.00122808
Example two	0.01664610	0.00067605	0.02211606
Example three	0.00579088	0.00013729	0.00443448
Example four	0.00000028	0.00000028	0.00000028
Total average	0.00652449	0.00020351	0.00694472

Supposing that the influence of fluctuation of number of sampling points N with respect to the computation time is not considered, then the above three types of algorithms vary linearly in terms of N . Here we select the least square regression model $t = \alpha + \beta N$ for them, in which t is the computation time and Table 3 provides the regression coefficients. Moreover, the computation time difference rate is defined as

$$\frac{(\max_{i=1,\dots,4} \{t_i\} - \min_{i=1,\dots,4} \{t_i\})}{\max_{i=1,\dots,4} \{t_i\}} \quad (34)$$

where t_i ($i = 1, 2, 3, 4$) is the computation time for NB, NT and KM algorithms.

From Figures 7–10 and Table 3, we can draw the conclusion that the computation time of NB and NT algorithms are much better than the KM algorithms. In the above four examples, the average computation time of both NB and NT algorithms is at least twenty times better than the KM algorithms. Here the computation time of NB and NT algorithms is very similar, while the latter is only slightly better than the former. Furthermore, the computation time difference rate for three types of algorithms is between 61.65%~99.20%.

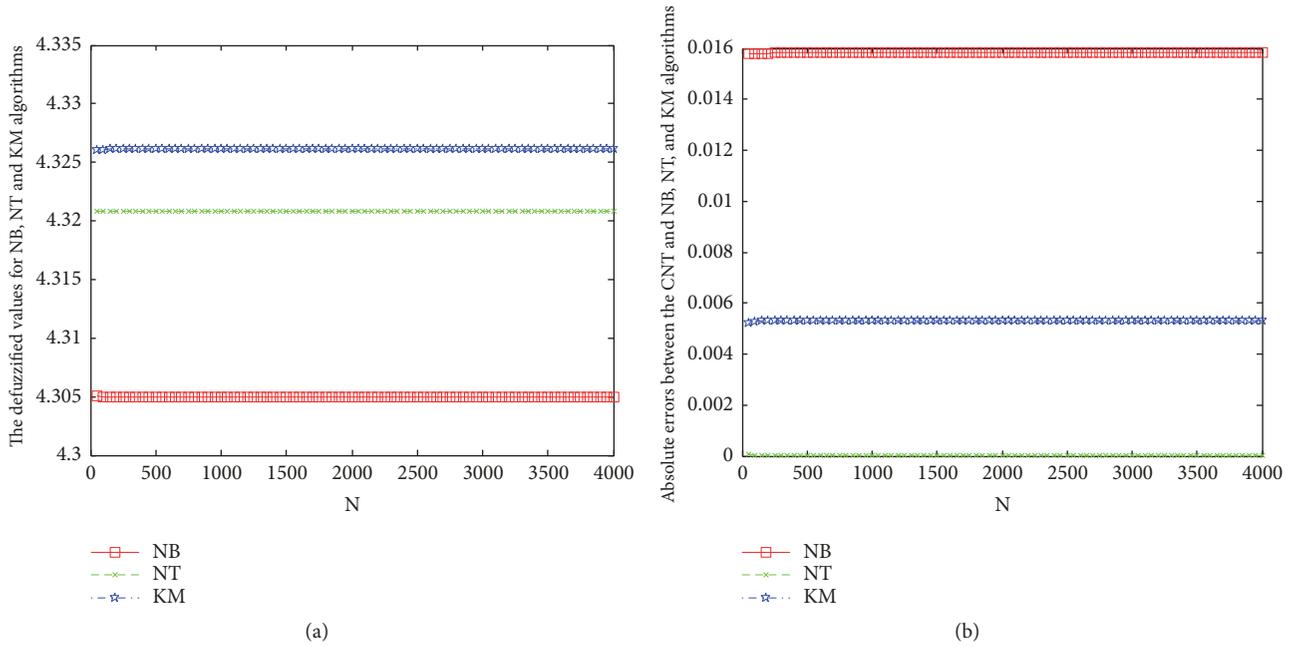


FIGURE 3: Calculation results of example one.

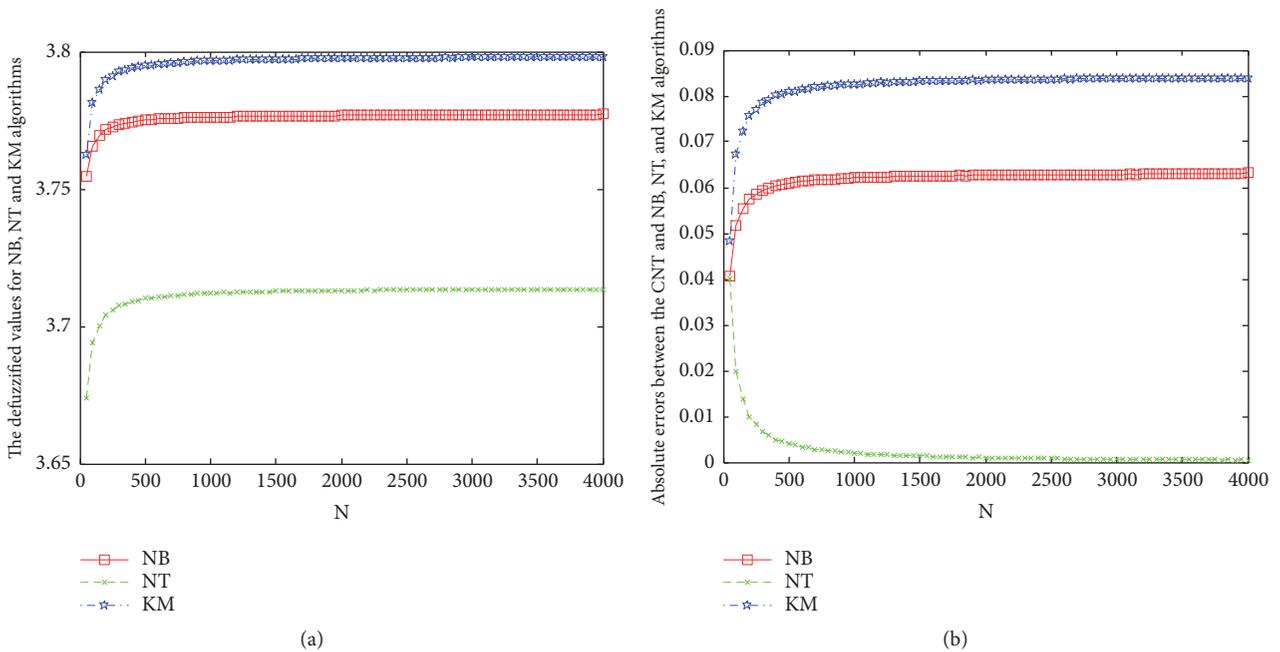


FIGURE 4: Calculation results of example two.

The noniterative algorithms can be adopted for studying the centroid TR of IT2 FLSs. If only the computation accuracy were considered, according to Table 2, the best choice should have been the NT algorithms. Observing both Figures 7–10 and Table 2, we suggest one shares the NT algorithms to perform the centroid TR of IT2 FLSs with the piecewise linear functions as in example one, the hybrid functions as in example two, and the Gaussian functions as in example three

and adopt the NB or NT algorithms to compute the centroid TR of IT2 FLSs with the Gaussian T2 MF with uncertain derivation as in example four.

At present, because there is no comprehensive comparison on the performance of KM TR algorithms and other alternative TR algorithms (which is still an interesting open problem), the paper pays attention to the theoretical studies of NB, NT, and KM algorithms according to the comparisons.

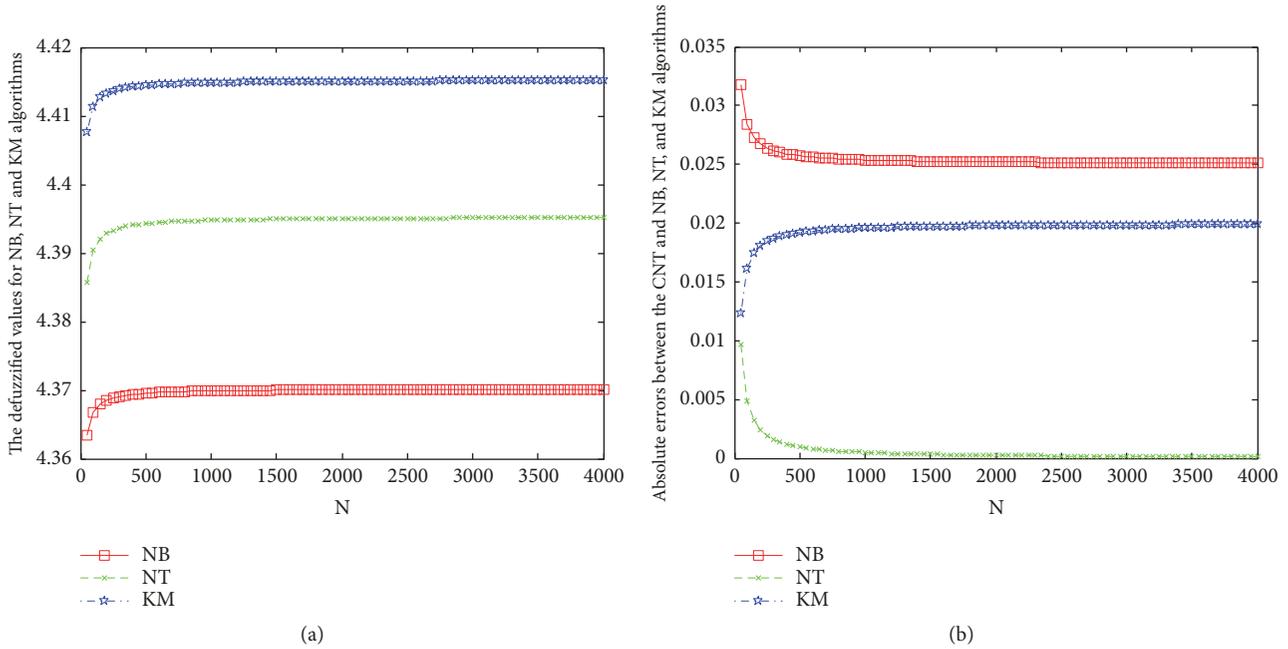


FIGURE 5: Calculation results of example three.

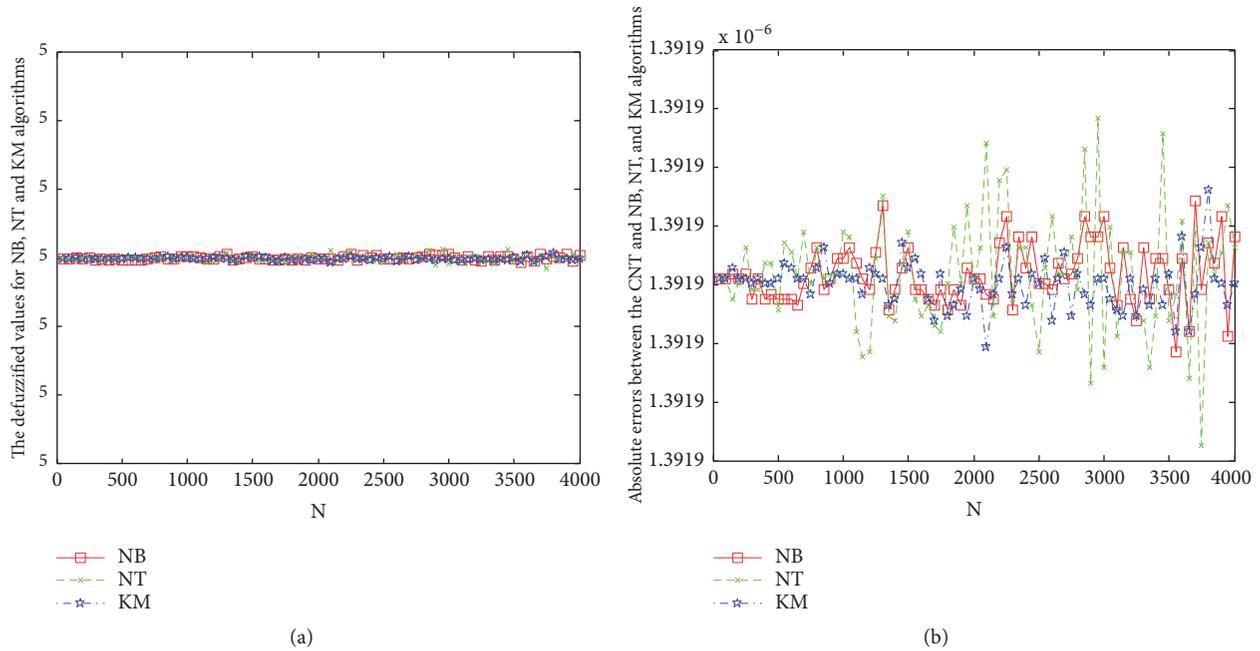


FIGURE 6: Calculation results of example four.

On the basis of four numerical examples, as a certain number of sampling points is considered, both NB and NT algorithms improve the computation accuracy compared with the KM algorithms. After choosing the accurate CNT TR algorithms as the benchmark, the NT algorithms can obtain the best computation accuracy.

5. Conclusions

This paper proposes the CNB algorithms, proves the CNT algorithms to be accurate approach for performing the centroid TR of IT2 FLSS, and provides connections between the BMM algorithms and the NB, and NT algorithms.

TABLE 3: Computations of the regression model coefficients by the least square for the NB, NT, and KM algorithms.

Coefficients	NB	NT	KM
	$\alpha/10^{-3}$ $\beta/10^{-3}$	$\alpha/10^{-3}$ $\beta/10^{-3}$	$\alpha/10^{-3}$ $\beta/10^{-3}$
Example one	0.0001	0.0002	0.0050
	0.1572	0.0827	0.0502
Example two	0.0001	0.0001	0.0062
	0.1392	0.0915	-0.2596
Example three	0.0001	0.0001	0.0051
	0.1217	0.0872	-0.0017
Example four	0.0001	0.0001	0.0069
	0.0564	0.0803	0.1580
Average	0.0001	0.0001	0.0058
	0.1186	0.0854	-0.0133

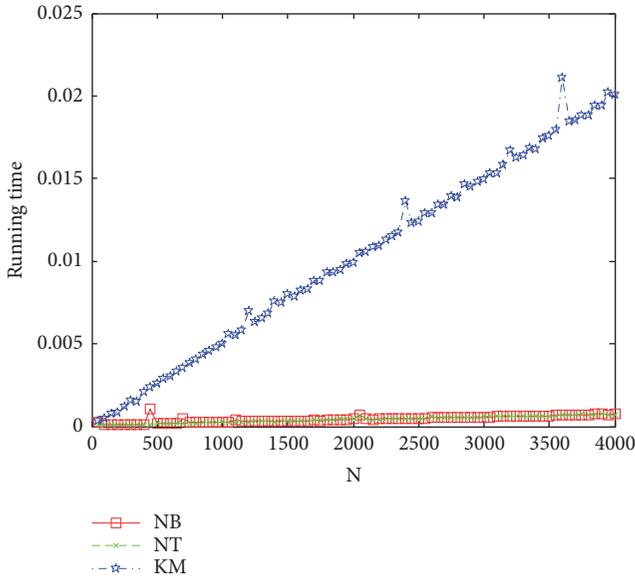


FIGURE 7: Computation time comparisons of example one.

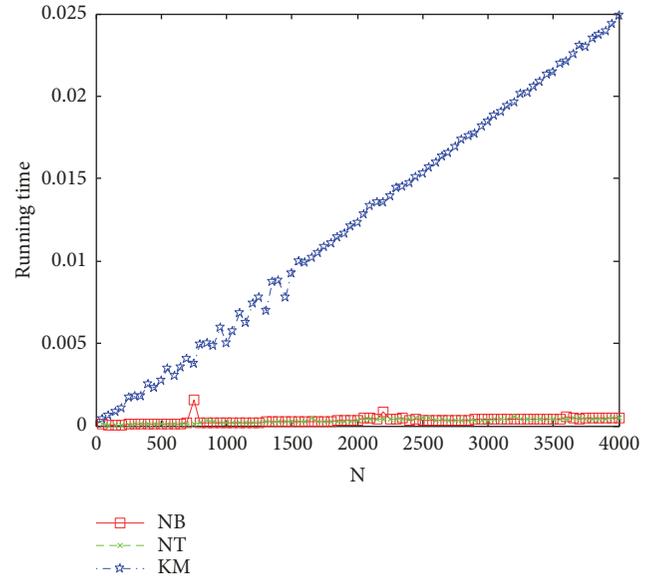


FIGURE 8: Computation time comparisons of example two.

Then we choose the CNT algorithms as the benchmark to compute the defuzzified output of IT2 FLSs. According to the computation accuracy and time, four computer simulation examples illustrate the performances of NT, NT, and KM algorithms. Compared with the NB and KM algorithms, the NT algorithms can obtain smaller absolute errors and less computation time.

There are still many interesting works that lie ahead, including studying the center of sets TR of IT2 and general T2 FLSs [30, 31, 34–37, 45] and investigating forecasting and control problems based on IT2 and GT2 FLSs optimized with swarm intelligence algorithms [46–51]. Future studies will give attention to the T2 FLSs design and applications.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declares that they have no conflicts of interest.

Acknowledgments

This paper is supported by the Natural Science Foundation of China (Nos. 61773188, 61803189) and Liaoning Province Natural Science Foundation Guidance Project (No. 20180550056). The author is very thankful to Professor J. M. Mendel, who has provided some valuable suggestions.

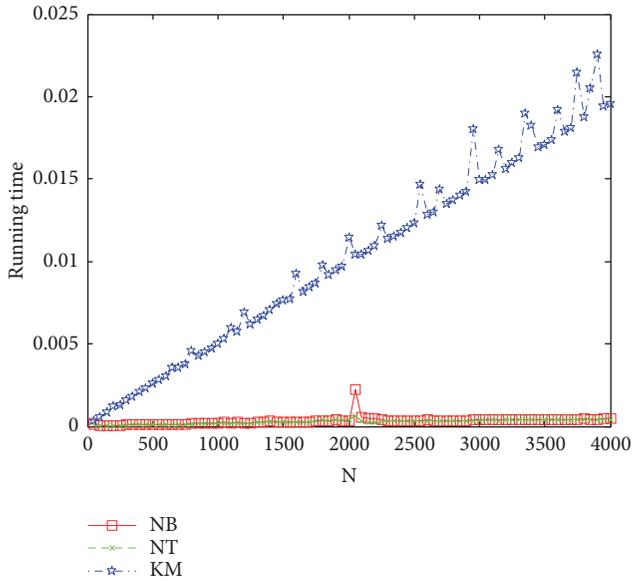


FIGURE 9: Computation time comparisons of example three.

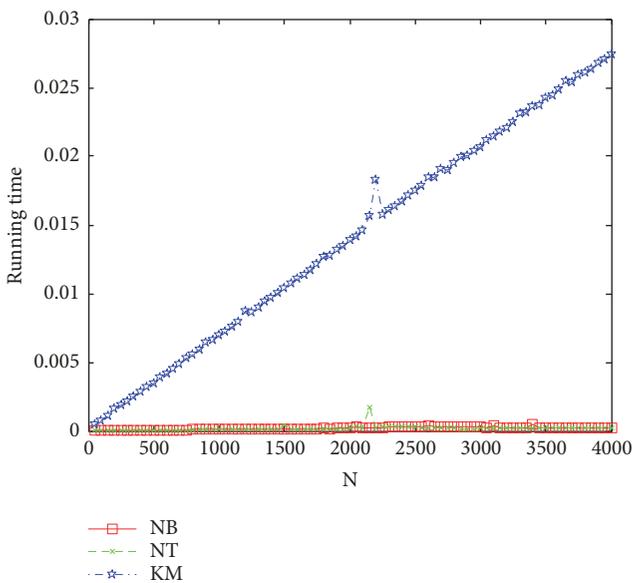


FIGURE 10: Computation time comparisons of example four.

References

- [1] D. Wu and J. M. Mendel, "Uncertainty measures for interval type-2 fuzzy sets," *Information Sciences*, vol. 177, no. 23, pp. 5378–5393, 2007.
- [2] D. Bernardo, H. Hagsras, and E. Tsang, "A genetic type-2 fuzzy logic based system for the generation of summarised linguistic predictive models for financial applications," *Soft Computing*, vol. 17, no. 12, pp. 2185–2201, 2013.
- [3] D. Wu and J. M. Mendel, "Computing with words for hierarchical decision making applied to evaluating a weapon system," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 3, pp. 441–460, 2010.
- [4] A. Khosravi, S. Nahavandi, D. Creighton, and D. Srinivasan, "Interval type-2 fuzzy logic systems for load forecasting: a comparative study," *IEEE Transactions on Power Systems*, vol. 27, no. 3, pp. 1274–1282, 2012.
- [5] Y. Chen, D. Wang, and S. Tong, "Forecasting studies by designing Mamdani interval type-2 fuzzy logic systems: with the combination of BP algorithms and KM algorithms," *Neurocomputing*, vol. 174, (part b), pp. 1133–1146, 2016.
- [6] K. Sabahi, S. Ghaemi, and S. Pezeshki, "Gain scheduling technique using MIMO type-2 fuzzy logic system for LFC in restructure power system," *International Journal of Fuzzy Systems*, vol. 19, no. 5, pp. 1464–1478, 2017.
- [7] H. Hagsras and C. Wagner, "Towards the wide spread use of type-2 fuzzy logic systems in real world applications," *IEEE Computational Intelligence Magazine*, vol. 7, no. 3, pp. 14–24, 2012.
- [8] C.-H. Hsu and C.-F. Juang, "Evolutionary robot wall-following control using type-2 fuzzy controller with species-DE-activated continuous ACO," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 1, pp. 100–112, 2013.
- [9] L. Wang, Z. Liu, Y. Zhang, C. L. Chen, and X. Chen, "Type-2 fuzzy logic controller using SRUKF-based state estimations for biped walking robots," *International Journal of Fuzzy Systems*, vol. 15, no. 4, pp. 423–434, 2013.
- [10] D. Wang and Y. Chen, "Study on permanent magnetic drive forecasting by designing Takagi Sugeno Kang type interval type-2 fuzzy logic systems," *Transactions of the Institute of Measurement and Control*, vol. 40, no. 6, pp. 2011–2023, 2018.
- [11] S. Barkat, A. Tlemçani, and H. Nouri, "Noninteracting adaptive control of PMSM using interval type-2 fuzzy logic systems," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 5, pp. 925–936, 2011.
- [12] O. Mendoza, P. Melin, and O. Castillo, "Interval type-2 fuzzy logic and modular neural networks for face recognition applications," *Applied Soft Computing*, vol. 9, no. 4, pp. 1377–1387, 2009.
- [13] P. Melin, O. Mendoza, and O. Castillo, "Face recognition with an improved interval type-2 fuzzy logic sugeno integral and modular neural networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 41, no. 5, pp. 1001–1012, 2011.
- [14] C.-S. Lee, M.-H. Wang, and H. Hagsras, "A type-2 fuzzy ontology and its application to personal diabetic-diet recommendation," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 2, pp. 316–328, 2010.
- [15] L. Gu and Y.-Q. Zhang, "Web shopping expert using new interval type-2 fuzzy reasoning," *Soft Computing*, vol. 11, no. 8, pp. 741–751, 2007.
- [16] T. Wang, Y. Chen, and S. Tong, "Fuzzy reasoning models and algorithms on Type-2 fuzzy sets," *International Journal of Innovative Computing, Information and Control*, vol. 4, no. 10, pp. 2451–2460, 2008.
- [17] N. N. Karnik and J. M. Mendel, "Centroid of a type-2 fuzzy set," *Information Sciences*, vol. 132, no. 1–4, pp. 195–220, 2001.
- [18] Y. Chen and D.-Z. Wang, "Type-reduction of interval type-2 fuzzy logic systems with weighted Karnik-Mendel algorithms," *Control Theory and Applications*, vol. 33, no. 10, pp. 1327–1336, 2016.
- [19] J. M. Mendel, "On KM algorithms for solving type-2 fuzzy set problems," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 3, pp. 426–446, 2013.
- [20] D. Wu and J. M. Mendel, "Enhanced Karnik-Mendel algorithms," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 4, pp. 923–934, 2009.

- [21] J. M. Mendel and H. Wu, "Type-2 fuzzistics for symmetric interval type-2 fuzzy sets—part 1, forward problems," *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 6, pp. 781–792, 2006.
- [22] X. Liu, J. M. Mendel, and D. Wu, "Study on enhanced Karnik-Mendel algorithms: Initialization explanations and computation improvements," *Information Sciences*, vol. 184, no. 1, pp. 75–91, 2012.
- [23] J. M. Mendel and F. Liu, "Super-exponential convergence of the Karnik-Mendel algorithms for computing the centroid of an interval type-2 fuzzy set," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 2, pp. 309–320, 2007.
- [24] A. M. El-Nagar and M. El-Bardini, "Simplified interval type-2 fuzzy logic system based on new type-reduction," *Journal of Intelligent & Fuzzy Systems*, vol. 27, no. 4, pp. 1999–2010, 2014.
- [25] H. Wu and J. M. Mendel, "Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 5, pp. 622–639, 2002.
- [26] M. Biglarbegian, W. W. Melek, and J. M. Mendel, "On the stability of interval type-2 TSK fuzzy logic systems," *IEEE Transactions on Systems, Man and Cybernetics-Part B, Cybernetics*, vol. 40, no. 3, pp. 798–818, 2010.
- [27] M. Biglarbegian, W. Melek, and J. Mendel, "On the robustness of type-1 and interval type-2 fuzzy logic systems in modeling," *Information Sciences*, vol. 181, no. 7, pp. 1325–1347, 2011.
- [28] S. Greenfield, F. Chiclana, S. Coupland, and R. John, "The collapsing method of defuzzification for discretised interval type-2 fuzzy sets," *Information Sciences*, vol. 179, no. 13, pp. 2055–2069, 2009.
- [29] J. M. Mendel and X. Liu, "Simplified interval type-2 fuzzy logic systems," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 6, pp. 1056–1069, 2013.
- [30] J. Li, R. John, S. Coupland, and G. Kendall, "On Nie-Tan operator and type-reduction of interval type-2 fuzzy sets," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 2, pp. 1036–1039, 2018.
- [31] J. M. Mendel, "General type-2 fuzzy logic systems made simple: a tutorial," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 5, pp. 1162–1182, 2014.
- [32] F. Liu, "An efficient centroid type-reduction strategy for general type-2 fuzzy logic system," *Information Sciences*, vol. 178, no. 9, pp. 2224–2236, 2008.
- [33] Y. Chen, D. Wang, and W. Ning, "Studies on centroid type-reduction algorithms for general type-2 fuzzy logic systems," *International Journal of Innovative Computing, Information and Control*, vol. 11, no. 6, pp. 1987–2000, 2015.
- [34] Y. Chen and D. Wang, "Study on centroid type-reduction of general type-2 fuzzy logic systems with weighted enhanced Karnik–Mendel algorithms," *Soft Computing*, vol. 22, no. 4, pp. 1361–1380, 2018.
- [35] T. Kumbasar, "Revisiting Karnik-Mendel algorithms in the framework of linear fractional programming," *International Journal of Approximate Reasoning*, vol. 82, pp. 1–21, 2017.
- [36] S. Greenfield and F. Chiclana, "Accuracy and complexity evaluation of defuzzification strategies for the discretised interval type-2 fuzzy set," *International Journal of Approximate Reasoning*, vol. 54, no. 8, pp. 1013–1033, 2013.
- [37] S. Greenfield and F. Chiclana, "Defuzzification of the discretised generalised type-2 fuzzy set: Experimental evaluation," *Information Sciences*, vol. 244, pp. 1–25, 2013.
- [38] J. M. Mendel and R. I. B. John, "Type-2 fuzzy sets made simple," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 117–127, 2002.
- [39] J. Aisbett, J. T. Rickard, and D. G. Morgenthaler, "Type-2 fuzzy sets as functions on spaces," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 4, pp. 841–844, 2010.
- [40] A. Khosravi and S. Nahavandi, "Load forecasting using interval type-2 fuzzy logic systems: Optimal type reduction," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1055–1063, 2014.
- [41] J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*, Prentice-Hall, Englewood Cliffs, NJ, USA, 2001.
- [42] J. M. Mendel, R. I. John, and F. Liu, "Interval type-2 fuzzy logic systems made simple," *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 6, pp. 808–821, 2006.
- [43] M. Nie and W. W. Tan, "Towards an efficient type-reduction method for interval type-2 fuzzy logic systems," in *Proceedings of the 2008 IEEE International Conference on Fuzzy Systems, FUZZ 2008*, pp. 1425–1432, June 2008.
- [44] X. Liu and J. M. Mendel, "Connect Karnik-Mendel algorithms to root-finding for computing the centroid of an interval type-2 fuzzy set," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 4, pp. 652–665, 2011.
- [45] M. A. Khanesar, A. J. Khakshour, O. Kaynak, and H. Gao, "Improving the speed of center of sets type reduction in interval Type-2 fuzzy systems by eliminating the need for sorting," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 5, pp. 1193–1206, 2017.
- [46] Y. Chen, D. Wang, and W. Ning, "Forecasting by TSK general type-2 fuzzy logic systems optimized with genetic algorithms," *Optimal Control Applications and Methods*, vol. 39, no. 1, pp. 393–409, 2018.
- [47] Y. Chen and D. Wang, "Forecasting by general type-2 fuzzy logic systems optimized with QPSO algorithms," *International Journal of Control, Automation, and Systems*, vol. 15, no. 6, pp. 2950–2958, 2017.
- [48] M. A. Sanchez, O. Castillo, and J. R. Castro, "Generalized Type-2 Fuzzy Systems for controlling a mobile robot and a performance comparison with Interval Type-2 and Type-1 Fuzzy Systems," *Expert Systems with Applications*, vol. 42, no. 14, pp. 5904–5914, 2015.
- [49] Y. Chen and D. Wang, "Study on centroid type-reduction of general type-2 fuzzy logic systems with weighted Nie–Tan algorithms," *Soft Computing*, vol. 22, no. 22, pp. 7659–7678, 2018.
- [50] C. I. Gonzalez, J. R. Castro, P. Melin, and O. Castillo, "An edge detection method based on generalized type-2 fuzzy logic," *Soft Computing*, vol. 20, no. 2, pp. 773–784, 2016.
- [51] Y. Chen, "Study on weighted Nagar-Bardini algorithms for centroid type-reduction of interval type-2 fuzzy logic systems," *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, vol. 34, no. 4, pp. 2417–2428, 2018.

