Design of sampling-based noniterative algorithms for centroid type-reduction of general type-2 fuzzy logic systems

Yang Chen

Received: 29 January 2022 / Accepted: 18 May 2022 / Published online: 20 June 2022
© The Author(s) 2022

Abstract
General type-2 (GT2) fuzzy logic systems (FLSs) become a popular research topic for the past few years. Usually the Karnik–Mendel algorithms are the most prevalent approach to complete the type-reduction. Nonetheless, the iterative quality of these types of computational intensive algorithms might impede applying them. For the improved types of algorithms, some noniterative algorithms can enhance the calculation efficiencies greatly, while it is still an open problem for comparing the relation between the discrete TR algorithms and corresponding continuous TR algorithms. First, the sum and integral operations in discrete and continuous noniterative algorithms are compared. Then, three kinds of noniterative algorithms originate from the type-reduction of interval type-2 FLSs are extended to complete the centroid type-reduction of general T2 FLSs. Four computer simulations prove that while changing the number of samples suitably, the calculational results of discrete types of algorithms may accurately gain on the related continuous types of algorithms, and the calculational times of discrete types of algorithms are obviously less than the continuous types of algorithms, this may offer the possible meaning for designing and applying T2 FLSs.

Keywords
Type-reduction (TR) · General type-2 fuzzy logic systems · Noniterative algorithms · Sampling · Computer simulation

Introduction
As is known to all, the computational complexities of general T2 FLSs [1] are much higher than IT2 FLSs. Recently, as the alpha-planes representations of GT2 fuzzy sets (FSs [2–6]) were put forward by few study groups, the calculated amount of GT2 FLSs can be significantly decreased. As the secondary membership grades of interval T2 FSs are equal to one, they can only measure the uncertainty of membership function (MF) in a unified manner. For GT2 FSs, whose secondary membership grades lie between zero and one, therefore, they can measure the uncertainty of MF in a non-uniform way. In other words, GT2 FSs may be considered as more advanced uncertainty models than IT2 FSs. As the design degrees of freedom increase, general T2 FLSs [7–11] have the latent capacity to overmatch IT2 FLSs on dealing with more complex uncertainty environments.

In general, T2 FLSs are made up of five blocks as: fuzzifier, fuzzy rules, fuzzy inference, TR and defuzzification. Among which, the type-reduction acts as translating the type-2 FS to the type-1 FS. Then, the module of defuzzification transforms the T1 FS to a output. The centroid type-reduction [12–14] is the most prevalent study method. In the early days, the iterative Karnik–Mendel (KM) algorithms [15] were opened up to perform the TR of interval T2 FLSs. Then, the continuous type of Karnik–Mendel (CKM) algorithms [16, 17] were put forward; moreover, the monotone property and super-convergence nature of them were also proved. Usually, it requires two to six iteration steps for each KM algorithm. For the sake of improving the computational cost, many other iterative algorithms are put forward gradually, they are enhanced Karnik–Mendel (EKM) algorithms [18], weighted based EKM algorithms [5, 19], enhanced opposite direction searching (EODS) algorithms [20–22] and so on. The other kinds of noniterative algorithms which achieve the output of systems directly were also put forward, they are Uncertainty Boundary (UB) algorithms [23], Coupland and John algorithms [24], Nagar and Bardini algorithms [13, 25], Nie and Tan algorithms [26, 27], Begian and Melek

Yang Chen
lxychenyang@lnut.edu.cn

1 College of Science, Liaoning University of Technology, Jinzhou 121001, China
and Mendel algorithms [28, 29] and so forth. Interval T2 FLSs on account of NB algorithms have excellent shows over the impact of uncertainty [30–32]. Most recent theoretical researches about calculating the centroids of IT2 FSs show that the continuous type of NT algorithms are actually an accurate TR method. Furthermore, interval T2 FLSs on the basis of Begian–Melek–Mendel algorithms have advantages on both stability and robustness compared with the T1 FLSs. All these jobs have established foundations for investigating the noniterative algorithms for completing the centroid TR for general T2 FLSs.

The paper analyzes the sum and integral operations in the corresponding discrete and continuous algorithms. In addition, the Nagar and Bardini (NB) and Nie and Tan (NT) algorithms are demonstrated to be two specific circumstances of Begian–Melek–Mendel algorithms. For the GT2 FLSs on the basis of the alpha-planes expression of GT2 FSs, we provide the inference, type-reduction and defuzzification according to these three types of noniterative algorithms. Then, four simulation experiments are adopted to illustrate that while increasing the sampled number of primary variable, the computational results of discrete type of noniterative algorithms may accurately gain on the corresponding continuous type of noniterative algorithms.

The rest of the paper is arranged as the following. The next section provides the background knowledge of GT2 FLSs. Then, the third section gives the three kinds of noniterative NB, NT and BMM algorithms, and how to expand them three to complete the type-reduction of GT2 FLSs. According to the simulation instances, the fourth section illustrates and analyzes the computation results. Finally, the last section is the conclusion.

**GT2 FLSs**

From the viewpoint of inference [1, 33, 34], general T2 FLSs can be categorized into both the Mamdani type [9, 12, 35] and Takagi and Sugeno and Kang type [8, 36–38]. Take into account a Mamdani general T2 FLS which has m inputs $x_1 \in X_1$, $x_2 \in X_2$, $\ldots$ $x_m \in X_m$, and a single output $y \in Y$, and the system is described by $M$ rules, in which the $i$th rule can be as

If $x_1$ is $\tilde{E}_i^1$ and $x_2$ is $\tilde{E}_i^2$ $\ldots$ and $x_m$ is $\tilde{E}_i^m$,

then $y$ is $\tilde{H}_i^s$ ($s = 1, 2, \ldots, M$) \hspace{1cm} (1)

in which $\tilde{E}_i^s (i = 1, \ldots, n; \ s = 1, 2, \ldots, M)$ represents the antecedent GT2 fuzzy set, and $\tilde{H}_i^s (s = 1, 2, \ldots, M)$ denotes the consequent GT2 fuzzy set.

For the sake of simplifying the expressions, singleton fuzzifier is used, i.e., when $x_i = x_i'$, let the vertical slice (secondary MF) $\tilde{E}_i^s (x_i')$ be activated, then the $\alpha$-cut decomposition should be

$$\tilde{E}_i^s (x_i') = \sup_{\alpha \in [0, 1]} \alpha/\{a_{i, \alpha}^s (x_i'), b_{i, \alpha}^s (x_i')\}.$$ \hspace{1cm} (2)

For each rule, the firing interval for the relevant $\alpha$-level should be computed as

$$E_\alpha : \begin{align*}
E_\alpha^s (x_i') & \equiv [\ell_\alpha^s (x_i'), u_\alpha^s (x_i')] \\
\ell_\alpha^s (x_i') & \equiv T_{i=1}^p a_{i, \alpha}^s (x_i') \\
u_\alpha^s (x_i') & \equiv T_{i=1}^p b_{i, \alpha}^s (x_i')
\end{align*}$$ \hspace{1cm} (3)

in which $T$ is the minimum or product t-norm.

Let the $\alpha$-plane of consequent $\tilde{H}_i^s$ at the corresponding $\alpha$-level be $\tilde{H}_i^s$, then

$$\tilde{H}_i^s = \int_{y \in Y} [h_{L, \alpha}^s (y), h_{R, \alpha}^s (y)] / y.$$ \hspace{1cm} (4)

To get the firing rule $\alpha$-plane $\tilde{A}_\alpha$, for each rule, combining the fired interval with its relevant consequent $\alpha$-plane, that is to say,

$$\tilde{A}_\alpha : \begin{align*}
\text{FOU}(\tilde{A}_\alpha^s) & \equiv [\ell_{\tilde{A}_\alpha}^s (y|x'), u_{\tilde{A}_\alpha}^s (y|x')] \\
\ell_{\tilde{A}_\alpha}^s (y|x') & \equiv \ell_{\tilde{A}_\alpha}^s (y|x') * h_{L, \alpha}^s (y) \\
u_{\tilde{A}_\alpha}^s (y|x') & \equiv \nu_{\tilde{A}_\alpha}^s (y|x') * h_{R, \alpha}^s (y)
\end{align*}$$ \hspace{1cm} (5)

Aggregating all the $\tilde{A}_\alpha^s (s = 1, 2, \ldots, M)$ to get the $\alpha$-plane $\tilde{A}_\alpha$, that is to say,

$$\tilde{A}_\alpha : \begin{align*}
\text{FOU}(\tilde{A}_\alpha) & \equiv [\ell_{\tilde{A}_\alpha} (y|x'), u_{\tilde{A}_\alpha} (y|x')] \\
\ell_{\tilde{A}_\alpha} (y|x') & \equiv \ell_{\tilde{A}_\alpha} (y|x') \lor \ell_{\tilde{A}_\alpha} (y|x') \lor \cdot \cdot \cdot \lor \ell_{\tilde{A}_\alpha} (y|x') \\
u_{\tilde{A}_\alpha} (y|x') & \equiv \nu_{\tilde{A}_\alpha} (y|x') \lor \nu_{\tilde{A}_\alpha} (y|x') \lor \cdot \cdot \cdot \lor \nu_{\tilde{A}_\alpha} (y|x')
\end{align*}$$ \hspace{1cm} (6)

in which $\lor$ represents the maximum operation.

Then, the $Y_{C, \alpha} (x' \in Y)$ at the corresponding $\alpha$-level should be acquired by obtaining the centroid of $\tilde{A}_\alpha$, that is to say,

$$Y_{C, \alpha} (x') = \alpha / \{l_{\tilde{A}_\alpha} (x'), r_{\tilde{A}_\alpha} (x')\}.$$ \hspace{1cm} (7)

in which the $l_{\tilde{A}_\alpha} (x')$ and $r_{\tilde{A}_\alpha} (x')$ may be computed by different types of TR algorithms [5, 6, 12–27] as

$$l_{\tilde{A}_\alpha} (x') = \min_{u_{\tilde{A}_\alpha} (y) \in [\ell_{\tilde{A}_\alpha} (y|x'), u_{\tilde{A}_\alpha} (y)|x')] \left[ \sum_{i=1}^N y_i R_{i, \alpha} (y) \right] / \sum_{i=1}^N R_{i, \alpha} (y)$$ \hspace{1cm} (8)

and
in which the special IT2 FS $R_{\tilde{A}_u} = \alpha / \tilde{A}_u$, and $N$ denotes the number of sampling.

Finally, aggregate every $\alpha$-planes $Y_{C, \alpha}$ to construct the T1 FS $Y_C$, i.e.,

$$Y_C = \sup_{\forall \alpha \in [0, 1]} \alpha / Y_{C, \alpha}(x').$$

In practical computations, let the number of effective $\alpha$-planes be $p$, that is to say, the value of $\alpha$ is usually equally decomposed into $\alpha = \alpha_1, \alpha_2, \ldots, \alpha_p$, thus, the crisp output of GT2 FLSs [39–42] should be

$$y = \frac{\sum_{i=1}^{p} \alpha_i \left( \pi (l_{\tilde{A}_u} (x')) + r_{\tilde{A}_u} (x') \right)}{\sum_{i=1}^{p} \alpha_i},$$

(11)

Three types of noniterative algorithms

Three types of discrete noniterative algorithms are provided here for investigating the centroid type-reduction for GT2 FLSs.

Nagar–Bardini noniterative algorithms

The closed form of NB algorithms [25] based IT2 FLSs were shown to own distinguish advantages to cope with far-ranging of uncertainty. According to the inference, let the output be the GT2 fuzzy set $\tilde{A}$. Let the $\tilde{A}$ be evenly discretized to $n$ points, then the $l_{\tilde{A}_u}$ and $r_{\tilde{A}_u}$ can be calculated as

$$l_{\tilde{A}_u} = \sum_{i=1}^{n} \frac{y_i u_{R_{\tilde{A}_u}} (y_i)}{\sum_{i=1}^{n} u_{R_{\tilde{A}_u}} (y_i)},$$

(12)

$$r_{\tilde{A}_u} = \sum_{i=1}^{n} \frac{y_i \overline{u}_{R_{\tilde{A}_u}} (y_i)}{\sum_{i=1}^{n} \overline{u}_{R_{\tilde{A}_u}} (y_i)}.$$

(13)

According to the continuous type of KM algorithms [14–17, 19], the continuous type of NB (CNB) algorithms should be provided as

$$l_{\tilde{A}_u} = \frac{\int_{a}^{b} y u_{R_{\tilde{A}_u}} (y)dy}{\int_{a}^{b} u_{R_{\tilde{A}_u}} (y)dy},$$

(14)

$$r_{\tilde{A}_u} = \frac{\int_{a}^{b} y \overline{u}_{R_{\tilde{A}_u}} (y)dy}{\int_{a}^{b} u_{R_{\tilde{A}_u}} (y)dy}.$$

Then, the type-reduced sets and defuzzified outputs can be obtained in terms of Eqs. (10) and (11). Here, both the discrete NB algorithms and continuous NB algorithms calculate the output as the linear combination of 2 special type-1 FLSs: one relies the upper MFs, while the other depends on the lower MFs.

Nie–Tan noniterative algorithms

The discrete NT algorithms compute the defuzzified output of GT2 FS at the corresponding $\alpha$-level as

$$y_{NT, \alpha} = \frac{\sum_{i=1}^{n} y_i [u_{R_{\tilde{A}_u}} (y_i) + \overline{u}_{R_{\tilde{A}_u}} (y_i)]}{\sum_{i=1}^{n} [u_{R_{\tilde{A}_u}} (y_i) + \overline{u}_{R_{\tilde{A}_u}} (y_i)]}.$$  (15)

Then, the continuous NT (CNT) algorithms solve it as

$$y_{CNT, \alpha} = \frac{\int_{a}^{b} y [u_{R_{\tilde{A}_u}} (y) + \overline{u}_{R_{\tilde{A}_u}} (y)]dy}{\int_{a}^{b} [u_{R_{\tilde{A}_u}} (y) + \overline{u}_{R_{\tilde{A}_u}} (y)]dy}. $$

(17)

Most recent studies show that the CNT algorithms [26] are in reality an accurate TR approach. Here, the statements and explanations are provided as follows.

**Theorem 1** [17]. The random sampling methods (algorithms) should obtain the precise centroid type-reduction of GT2 FLSs as the sampled number is infinity.

**Proof.** According to the inference [43], suppose that $R_{\tilde{A}_u}$ be the output interval type-2 FS (under $\alpha$-level). Furthermore, along the $y$-axis, let the vertical slices number be $m$, and along the $u$-axis, $l$ denotes the horizontal slices number, therefore, the embedded T1 fuzzy sets number is $l^m$.

Let $u_{R_{\tilde{A}_u}} (y)$, and $\overline{u}_{R_{\tilde{A}_u}} (y)$ be the upper bound and lower bound of FOU for $R_{\tilde{A}_u}$. Here, select $M$ embedded type-1 fuzzy sets randomly for $R_{\tilde{A}_u}$, and suppose that $\mu_i (y)$ ($i = 1, 2, \ldots, M$) be the $i$th embedded FS. Aggregating all $M$ embedded fuzzy sets, therefore,

$$\sum_{i=1}^{M} x_i (y) = \sum_{j=1}^{m} \sum_{i=1}^{M} u_i (y_j).$$  (18)

So that,

$$\lim_{M \to \infty} \frac{1}{M} \sum_{i=1}^{M} u_i (y) = \lim_{M \to \infty} \frac{1}{M} \sum_{i=1}^{M} \sum_{j=1}^{m} u_i (y_j).$$  (19)
As \( u_i(y_j) \) is a random number on \( [u_i(y_j), \bar{u}_i(y_j)] \), therefore, the right side of Eq. (19) can be rewritten as

\[
\lim_{M \to \infty} \frac{1}{M} \sum_{i=1}^{M} \frac{1}{l} \sum_{k=1}^{l} \sum_{j=1}^{m} u_k(y_j). \quad (20)
\]

So that,

\[
\lim_{M \to \infty} \frac{1}{M} \sum_{i=1}^{M} u_i(y) = \sum_{k=1}^{l} \sum_{j=1}^{m} u_k(y_j). \quad (21)
\]

For Eq. (20), it is sampled randomly as \( M \to \infty \) for the left, and the right is the aggregation. Attention that \( l/M \) is only a constant, therefore, it has nothing relations to the calculations of the centroid.

As for the continuous type interval type-2 FS \( \tilde{R}_A^\alpha \), just replace the \( \sum_{j=1}^{m} \) and \( \sum_{k=1}^{l} \) with the \( \int_{y_{min}}^{y_{max}} \) and \( \int_{y_{min}}^{y(y)} \) will work. Therefore,

\[
\lim_{M \to \infty} \frac{1}{M} \sum_{i=1}^{M} u_i(y) = \int_{y_{min}}^{y_{max}} \pi(y) \int_{y(y)} u(y) d\mu(y). \quad (22)
\]

**Theorem 2** [17]. For the \( R_{A^\alpha} \), let whose membership function of representative embedded FS be the form

\[
u^*(y) = [\nu(y) + \bar{\nu}(y)] / 2. \quad (23)
\]
Table 1 Membership function expressions for FOUs

| Num | Expressions |
|-----|-------------|
| 1   | \[ \mu_{A_1}(x) = \max \left\{ \begin{array}{ll} \frac{x - 1}{6}, & x \in [1, 4] \\ \frac{7 - x}{6}, & x \in (4, 7] \\ 0, & \text{otherwise} \end{array} \right\} \] |
| 2   | \[ \mu_{A_2}(x) = \begin{cases} 0.6(x + 5)/19, & x \in [-5, 2.6] \\ 0.4(14 - x)/19, & x \in (2.6, 14] \end{cases} \] |
| 3   | \[ \mu_{A_3}(x) = \max \left\{ \frac{3 - x}{2}, \frac{7 - 2x}{3}, \frac{8 - x}{9}, 0 \right\} \] |

Actually, \[ \lim_{M \to \infty} \frac{1}{M} \sum_{i=1}^{M} \varphi_i = \frac{M}{2}, \] so that,

\[ \lim_{M \to \infty} \frac{1}{M} \sum_{i=1}^{M} u_i(y_j) = \frac{1}{2} [\mu(y_j) + \overline{\mu}(y_j)]. \]  

Computing the centroid a prevalent approach to defuzzify type-1 fuzzy set, i.e.,

\[ C(R_{\tilde{A}_u}) = \int_{\mu_{\min}}^{\mu_{\max}} \mu(y) \, dy. \]  

As \( y_j \) is just a random vertical slice of \( y \), on the basis of Eq. (26), it can be obtained that

\[ \lim_{M \to \infty} \frac{1}{M} \sum_{i=1}^{M} u_i(y) = \frac{1}{2} [\mu(y) + \overline{\mu}(y)]. \]

According to Eq. (27) and theorem 1, \( u^*(y) = [\mu(y) + \overline{\mu}(y)]/2 \) will be the membership function of representative embedded FS. Furthermore, this will be used to compute an accurate centroid of \( R_{\tilde{A}_u} \). When \( R_{\tilde{A}_u} \) is the continuous type, simply substitute the \( \sum_{i=1}^{M} \) by the \( \int_{\mu_{\min}}^{\mu_{\max}} \) for the proofs. Therefore, the continuous NT algorithms are actually equal to the exhaustive type-reduction algorithms, and this can be extended to perform the accurate type-reduction.

**Begian–Melek–Mendel noniterative algorithms**

The discrete BMM algorithms compute the output straightly, and the output of GT2 FS can be obtained as

\[ y_{BMM, a} = \sum_{i=1}^{n} \frac{y_i}{\sum_{i=1}^{n}} \frac{\mu_{R_{\tilde{A}_u}}(y_i)}{\sum_{i=1}^{n}} + b \sum_{i=1}^{n} \frac{\mu_{R_{\tilde{A}_u}}(y_i)}{\sum_{i=1}^{n}} \]  

in which the two adjustable coefficients are \( a \) and \( b \), respectively; here, the output is also a linear combination of 2 specific type-1 FLSs.

Then, the continuous BMM (CBMM) algorithms solve it as

\[ y_{CBMM, a} = \int_{a}^{b} \frac{y \mu_{R_{\tilde{A}_u}}(y)}{\int_{a}^{b} \mu_{R_{\tilde{A}_u}}(y)} \, dy + b \int_{a}^{b} \frac{y \mu_{R_{\tilde{A}_u}}(y)}{\int_{a}^{b} \mu_{R_{\tilde{A}_u}}(y)} \, dy. \]

In fact, the discrete BMM algorithms [14, 28, 29] can be considered as the more general form of discrete NB algorithms and NT algorithms. Then, the explanations are given as follows. Observe Eqs. (12), (13) and (29), it should be found that the NB algorithms and BMM algorithms are completely the same when \( a = a_{NB} = \frac{1}{2} \), and \( b = b_{NB} = \frac{1}{2} \). As for the NT algorithms, transforming Eq. (16) to
Fig. 2 Shape graphs for secondary membership functions

Table 2 Second membership function expressions for four cases

| Num | Second membership function expressions |
|-----|----------------------------------------|
| 1   | $L(x) = \mu(x) + \frac{3}{4}w(\overline{\mu}(x) - \mu(x))$ |
|     | $R(x) = \overline{\mu}(x) - \frac{3}{2}(1 - w)(\overline{\mu}(x) - \mu(x))$ |
|     | and $w = 0$ |
| 2   | Like in Case one |
| 3   | Apex $= \mu(x) + w[\overline{\mu}(x) - \mu(x)]$, and $w = \frac{1}{4}$ |
| 4   | Like in Case three, and $w = 0.75$ |

Observing Eqs. (29) and (31), as $a = a_{NT}$ and $b = b_{NT}$, we obtain that the NT algorithms and BMM algorithms will be the same.

Finally, the inner connections between the discrete and continuous algorithms for completing the centroid TR of GT2 FLSs should be as:

\[
a_{NT} = \frac{\sum_{i=1}^{n} \mu_{R_{\alpha_{i}}}(y_i)}{\sum_{i=1}^{n} [\mu_{R_{\alpha_{i}}}(y_i) + \overline{\mu}_{R_{\alpha_{i}}}(y_i)]}
\]
\[
b_{NT} = \frac{\sum_{i=1}^{n} \overline{\mu}_{R_{\alpha_{i}}}(y_i)}{\sum_{i=1}^{n} [\mu_{R_{\alpha_{i}}}(y_i) + \overline{\mu}_{R_{\alpha_{i}}}(y_i)]}.
\]
The discrete noniterative algorithms compute the centroids by means of the sum operation for the sampled points $y_i (i = 1, \ldots, n)$. In addition, the continuous noniterative algorithms compute in terms of the integral. In theory, the computational results of discrete noniterative algorithms can gain on the continuous noniterative algorithms while the number of samples approaches infinity.

For the discrete noniterative algorithms, the more precise calculation effects can be acquired by adding the number of samples.

Three discrete types of noniterative algorithms complete the numerical computations by means of the sum operation, while the continuous types of algorithms complete the calculations based on the integral.

Simulations

Four cases are given here. Suppose that the footprint of uncertainties (FOUs) and the related vertical slices (or secondary membership functions) of output GT2 FS be known according to the inference engine of GT2 FLSs. Here, four commonly used GT2 FSs are selected for studying. In case 1, the FOU is made up of the piecewise defined linear functions $[2, 3, 5, 6, 19]$, while the related secondary MF is selected as the trapezoidal type membership function. In the second case, the FOU is made up of the linear functions and Gaussian functions $[13–16, 26–29, 33]$, and the corresponding secondary membership functions is still selected as the trapezoidal type membership function. In the last case, the FOU is only a Gaussian primary membership function with $[2, 3, 5, 6, 19]$.
Fig. 4 Defuzzified values calculated by the continuous noniterative algorithms

uncertain standard deviations [13–16, 26–29, 33], and the corresponding secondary membership functions is the triangular type membership function. In experiments, the \( \alpha \) is equally divided into values as: \( \alpha = 0, 1/\Delta, \ldots, 1 \). Here, we let \( \Delta \) vary from one to a hundred with a step size of one. Figure 1 and Table 1 give the FOUs of all cases. In addition, Fig. 2 and Table 2 provide the related secondary membership functions.

Consider the continuous noniterative algorithms as benchmark to calculate both the type-reduced sets for \( \Delta = 100 \), and the defuzzified outputs for \( \Delta(\text{Delta}) \) ranging from one to a hundred with the step size of 1, and they are shown by Figs. 3 and 4, respectively. For the CBMM algorithms, here we choose the average of results of 20 random experiments for the adjustable coefficient \( a \) of CBMM algorithms, and another coefficient \( b = 1 - a \).

Then, the computational accuracies between continuous noniterative algorithms and sampling-based noniterative algorithms in relation with the number of samples are studied. Here, the number of samples are chosen as 20, 50, 100, 200 and 2000 (only in last three examples).

Here, the absolute errors between three kinds of continuous noniterative algorithms and sampling-based discrete noniterative algorithms for calculating type-reduced sets and defuzzified outputs are provided in Figs. 5, 6, 7, 8, 9, 10. Next, the quantitative studies for the mean of absolute errors are performed. Then, Tables 3, 4, 5 provide the means of absolute errors for type-reduced sets between the continuous noniterative algorithms and sampling-based discrete noniterative algorithms. Furthermore, the means of absolute errors for centroid defuzzified values are given in Tables 6, 7, 8.
Next, we investigate the unrepeatable calculation times for the continuous noniterative algorithms and their corresponding sampling-based discrete noniterative algorithms. Computer programs are completed with the software of Matlab 2013a. Here, we use the hardware platform as a double core CPU dell desktop.

Then, Tables 9, 10, 11, 12, 13, 14 provide the total computation times of noniterative algorithms for calculating the type-reduced sets and defuzzied values, respectively, where the time unit is the second(s). See from Figs. 5, 6, 7, 8, 9, 10 and Tables 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, the conclusions can be made as:

(1) For calculating both the centroid type-reduced sets and defuzzified values, as the number of samples increases, the results of sampling-based noniterative algorithms will all be more and more close to their corresponding continuous type of noniterative algorithms.

(2) In both examples 1 and 4, as the number of sampling is 100, the computational results of discrete type of noniterative algorithms will be the same as their related continuous type of noniterative algorithms (see Figs. 5a–10a, and Figs. 5d–10d); while for the middle two cases, the number of samples must be added to 19,000 to let the computational results of discrete type of noniterative algorithms be the same as their related continuous type of noniterative algorithms (see Figs. 5b–10b, and Figs. 5c–10c).

(3) While calculating average of absolute errors for type-reduced sets and defuzzified values, the discrete type of

![Fig. 5](https://example.com/fig5.png)
noniterative algorithms could better gain on their related continuous type of noniterative algorithms as the samples is chosen as 19,000 (see Tables 3, 4, 5, 6, 7, 8).

(4) For calculating the type-reduced sets and defuzzified values, the specific computational times of discrete noniterative algorithms are less than their corresponding continuous counterparts. Among them, the discrete type of noniterative algorithms which have the maximum number of samples take the longest computation times. Despite so, for the discrete type of noniterative algorithms which have the maximum number of samples, they only have about 0.6244%, 0.5208%, 0.7708%, and 0.6617%, 0.5635%, 0.6875% of their continuous counterparts, respectively (see Tables 9, 10, 11, 12, 13, 14).

(5) According to the items (1)–(4), it is obviously that the proposed sampling-based noniterative algorithms could be outstanding approximation approaches for their related continuous counterparts. Furthermore, the efficiencies of formers are distinctly higher than the continuous counterparts.

Conclusions and expectations

The paper compares the sum operation and integral operations in discrete and continuous algorithms on the foundation of type-2 fuzzy logic theory. As for four types of general
The absolute errors of type-reduced sets between continuous NT algorithms and sampling-based discrete NT algorithms for T2 fuzzy sets with different kinds of FOUs and secondary MFs, the continuous noniterative algorithms are selected as the criterion to calculate the type-reduced sets and centroid defuzzified outputs. Simulation instances are given to show the shows of proposed sampling-based discrete noniterative type of algorithms. As the number of samples is selected suitably, the proposed sampling-based noniterative type of algorithms can approach to the corresponding continuous counterparts exactly. In addition, the efficiencies of formers are significantly higher.

In the next, the author will investigate the initialization, searching space and stopping conditions of iterative algorithms for completing the center-of-sets (COS) type-reduction \cite{2, 6, 13–15, 19–29, 33, 44–46} of T2 FLSs. Furthermore, design and application of IT2 FLSs and GT2 FLSs on the basis of swarm intelligence algorithms \cite{8, 9, 11, 35, 37, 38, 47–52} will be investigated. Future studies will be focused on the theory \cite{39–42, 53–56} of type-2 FLSs and their related algorithms.

Fig. 7 The absolute errors of type-reduced sets between continuous NT algorithms and sampling-based discrete NT algorithms
Fig. 8 The absolute errors of defuzzified outputs between continuous NT algorithms and sampling-based discrete NT algorithms.
Fig. 9 The absolute errors of type-reduced sets between continuous BMM algorithms and sampling-based discrete BMM algorithms (a) Case 1 (b) Case 2 (c) Case 3 (d) Case 4
Fig. 10 The absolute errors of defuzzified outputs between continuous BMM algorithms and sampling-based discrete BMM algorithms

| Num | Approach | NB20 | NB50 | NB100 | NB200 | NB2000 | NB8000 | NB19000 |
|-----|----------|------|------|-------|-------|--------|--------|---------|
| 1   |          | 0.0009 | 0.0001 | 0 | 0 | – | – | – |
| 2   |          | 0.0501 | 0.0204 | 0.0101 | 0.0050 | 0.0005 | 0.0001 | 0 |
| 3   |          | 0.0173 | 0.0073 | 0.0037 | 0.0018 | 0.0002 | 0 | – |
| 4   |          | 0 | 0 | 0 | 0 | 0 | – | – |

Bold represents the smallest value of time

| Num | Approach | NT20 | NT50 | NT100 | NT200 | NT2000 | NT8000 | NT19000 |
|-----|----------|------|------|-------|-------|--------|--------|---------|
| 1   |          | 0.0008 | 0.0001 | 0 | 0 | – | – | – |
| 2   |          | 0.0819 | 0.0334 | 0.0165 | 0.0082 | 0.0008 | 0.0002 | 0 |
| 3   |          | 0.0173 | 0.0084 | 0.0042 | 0.0021 | 0.0002 | 0.0001 | 0 |
| 4   |          | 0 | 0 | 0 | 0 | 0 | – | – |

Bold represents the smallest value of time
| Num | Method               | BMM20 | BMM50 | BMM100 | BMM200 | BMM2000 | BMM8000 | BMM19,000 |
|-----|----------------------|-------|-------|--------|--------|---------|---------|------------|
| 1   | 0.0009               | 0.0001| 0.0001| 0.0001 | 0.0001 | 0.0001  | 0.0001  | 0.0001     |
| 2   | 0.0469               | 0.0191| 0.0095| 0.0047 | 0.0005 | 0.0001  | 0.0001  | 0.0001     |
| 3   | 0.0172               | 0.0073| 0.0036| 0.0018 | 0.0002 | 0.0001  | 0.0001  | 0.0001     |
| 4   | 0.0000               | 0.0000| 0.0000| 0.0000 | 0.0000 | 0.0000  | 0.0000  | 0.0000     |

**Table 5** Average of absolute errors of type-reduced sets between continuous BMM algorithms and sampling-based BMM algorithms

| Num | Approach           | NB20 | NB50 | NB100 | NB200 | NB2000 | NB8000 | NB19000 |
|-----|--------------------|------|------|-------|-------|--------|--------|---------|
| 1   | 0.0009             | 0.0001| 0.0001| 0.0001| 0.0001| 0.0001 | 0.0001 | 0.0001 |
| 2   | 0.0462             | 0.0196| 0.0097| 0.0048| 0.0005| 0.0001 | 0.0001 | 0.0001 |
| 3   | 0.0173             | 0.0073| 0.0037| 0.0018| 0.0002| 0.0001 | 0.0001 | 0.0001 |
| 4   | 0.0000             | 0.0000| 0.0000| 0.0000| 0.0000| 0.0000 | 0.0000 | 0.0000 |

**Table 6** Average of absolute errors of defuzzified outputs between continuous NB algorithms and sampling-based NB algorithms

| Num | Approach           | NT20 | NT50 | NT100 | NT200 | NT2000 | NT8000 | NT19000 |
|-----|--------------------|------|------|-------|-------|--------|--------|---------|
| 1   | 0.0008             | 0.0001| 0.0001| 0.0001| 0.0001| 0.0001 | 0.0001 | 0.0001 |
| 2   | 0.0768             | 0.0313| 0.0155| 0.0077| 0.0008| 0.0002 | 0.0001 | 0.0001 |
| 3   | 0.0186             | 0.0078| 0.0039| 0.0020| 0.0002| 0.0001 | 0.0001 | 0.0001 |
| 4   | 0.0000             | 0.0000| 0.0000| 0.0000| 0.0000| 0.0000 | 0.0000 | 0.0000 |

**Table 7** Average of absolute errors for defuzzified outputs between continuous NT algorithms and sampling-based NT algorithms

| Num | Method               | BMM20 | BMM50 | BMM100 | BMM200 | BMM2000 | BMM8000 | BMM19,000 |
|-----|----------------------|-------|-------|--------|--------|---------|---------|------------|
| 1   | 0.0010               | 0.0001| 0.0001| 0.0001| 0.0001| 0.0001  | 0.0001  | 0.0001     |
| 2   | 0.0451               | 0.0184| 0.0091| 0.0045| 0.0005| 0.0001  | 0.0001  | 0.0001     |
| 3   | 0.0172               | 0.0073| 0.0036| 0.0018| 0.0002| 0.0001  | 0.0001  | 0.0001     |
| 4   | 0.0000               | 0.0000| 0.0000| 0.0000| 0.0000| 0.0000  | 0.0000  | 0.0000     |

**Table 8** Average of absolute errors of defuzzified outputs between continuous BMM algorithms and sampling-based BMM algorithms

| Num | Approach           | CNB | NB20 | NB50 | NB100 | NB200 | NB2000 | NB8000 | NB19000 |
|-----|--------------------|-----|------|------|-------|-------|--------|--------|---------|
| 1   | 2.1082             | 0.0020| 0.0021| 0.0022| 0.0022| 0.0022| 0.0022| 0.0022| 0.0022 |
| 2   | 4.2118             | 0.0020| 0.0020| 0.0020| 0.0020| 0.0020| 0.0020| 0.0020| 0.0020 |
| 3   | 6.4697             | 0.0020| 0.0020| 0.0020| 0.0020| 0.0020| 0.0020| 0.0020| 0.0020 |
| 4   | 6.1196             | 0.0021| 0.0022| 0.0021| 0.0021| 0.0021| 0.0021| 0.0021| 0.0021 |

**Table 9** Calculational times comparisons of the continuous NB and sampling-based NB algorithms for calculating type-reduced sets

**Bold represents the smallest value of time**
Table 10 Calculational times comparisons of the continuous NT and sampling-based NT algorithms for computing type-reduced sets

| Num | Approach  | CNT   | NT20  | NT50  | NT100 | NT200 | NT2000 | NT8000 | NT19000 |
|-----|-----------|-------|-------|-------|-------|-------|--------|--------|---------|
| 1   |           | 2.1566| 0.0014| 0.0014| 0.0014| 0.0016| –      | –      | –       |
| 2   |           | 4.2617| 0.0016| 0.0013| 0.0013| 0.0016| 0.0036 | 0.0101 | 0.0225  |
| 3   |           | 6.5545| 0.0013| 0.0013| 0.0014| 0.0016| 0.0037 | 0.0111 | –       |
| 4   |           | 6.1673| 0.0013| 0.0013| 0.0014| 0.0015| 0.0041 | –      | –       |

Bold represents the smallest value of time

Table 11 Calculational times of the continuous BMM and sampling-based BMM algorithms for centroid type-reduced sets

| Num | Approach  | CBMM  | BMM20 | BMM50 | BMM100 | BMM200 | BMM2000 | BMM8000 | BMM19,000 |
|-----|-----------|-------|-------|-------|--------|--------|---------|---------|-----------|
| 1   |           | 2.1007| 0.0020| 0.0019| 0.0020 | 0.0021 | –       | –       | –         |
| 2   |           | 4.2813| 0.0019| 0.0019| 0.0020 | 0.0021 | 0.0051  | 0.0124  | 0.0330    |
| 3   |           | 6.4177| 0.0019| 0.0019| 0.0021 | 0.0021 | 0.0048  | 0.0120  | –         |
| 4   |           | 6.1271| 0.0020| 0.0019| 0.0020 | 0.0021 | 0.0056  | –       | –         |

Bold represents the smallest value of time

Table 12 Computational times comparisons of the CNB and sampling-based NB algorithms for computing defuzzified values

| Num | Approach  | CNB   | NB20  | NB50  | NB100 | NB200 | NB2000 | NB8000 | NB19000 |
|-----|-----------|-------|-------|-------|-------|-------|--------|--------|---------|
| 1   |           | 103.5734| 0.1035| 0.1054| 0.1102| 0.1178| –      | –      | –       |
| 2   |           | 210.5994| 0.1015| 0.1069| 0.1136| 0.1174| 0.2606 | 0.6674 | 1.3936   |
| 3   |           | 320.0638| 0.1003| 0.1074| 0.1111| 0.1168| 0.2382 | 0.6500 | –       |
| 4   |           | 304.9396| 0.1014| 0.1029| 0.1091| 0.1158| 0.2531 | –      | –       |

Bold represents the smallest value of time

Table 13 Computational times comparisons of the continuous NT and sampling-based NT algorithms for calculating defuzzified values

| Num | Approach  | CNT   | NT20  | NT50  | NT100 | NT200 | NT2000 | NT8000 | NT19000 |
|-----|-----------|-------|-------|-------|-------|-------|--------|--------|---------|
| 1   |           | 102.6065| 0.0688| 0.0715| 0.0743| 0.0785| –      | –      | –       |
| 2   |           | 209.0786| 0.0692| 0.0706| 0.0746| 0.0786| 0.2171 | 0.5367 | 1.1781   |
| 3   |           | 323.5862| 0.0691| 0.0711| 0.0748| 0.0799| 0.2013 | 0.5367 | 1.1813   |
| 4   |           | 313.8482| 0.0688| 0.0726| 0.0752| 0.0781| 0.2212 | –      | –       |

Bold represents the smallest value of time

Table 14 Computational times comparisons of the continuous BMM and sampling-based BMM algorithms for calculating defuzzified values

| Num | Approach  | CBMM  | BMM20 | BMM50 | BMM100 | BMM200 | BMM2000 | BMM8000 | BMM19,000 |
|-----|-----------|-------|-------|-------|--------|--------|---------|---------|-----------|
| 1   |           | 103.9631| 0.1019| 0.1041| 0.1072| 0.1141| –       | –       | –         |
| 2   |           | 214.4556| 0.1002| 0.1038| 0.1076| 0.1121| 0.2764  | 0.6408  | 1.4744    |
| 3   |           | 364.6653| 0.1020| 0.1043| 0.1072| 0.1144| 0.2490  | 0.7329  | –         |
| 4   |           | 310.9498| 0.1024| 0.1035| 0.1088| 0.1185| 0.2621  | –       | –         |

Bold represents the smallest value of time
Acknowledgements The paper is sponsored by the National Natural Science Foundation of China (No. 61973146, and No. 61773188), the Youth Fund of Education Department of Liaoning Province (No. LJKQZ2021143), the Doctoral Start-up Foundation of Liaoning Province (No. 2021-B5-258), and the Talent Fund Project of Liaoning University of Technology (No. xz2020002). The author is very grateful for the editors of this Journal.

Declarations

Conflict of interest No conflict exists. The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Mendel JM (2014) General type-2 fuzzy logic systems made simple: a tutorial. IEEE Trans Fuzzy Syst 22(5):1162–1182
2. Liu FL (2008) An efficient centroid type-reduction strategy for general type-2 fuzzy logic system. Inf Sci 178(1):2224–2236
3. Mendel JM, Liu FL, Zhai DY (2009) Alpha-plane representation for type-2 fuzzy sets: theory and applications: theory and applications. IEEE Trans Fuzzy Syst 17(5):1189–1207
4. Wagner C, Hagras H (2010) Toward general type-2 fuzzy logic systems based on sSlices. IEEE Trans Fuzzy Syst 18(4):637–660
5. Chen Y, Wang DZ (2018) Study on centroid type-reduction of general type-2 fuzzy logic systems with weighted enhanced Karnik-Mendel algorithms. Soft Comput 22(4):1361–1380
6. Greenfield S, Chiclana F (2013) Accuracy and complexity evaluation of defuzzification strategies for the discretised interval type-2 fuzzy set. Int J Approx Reason 54(8):1013–1033
7. Castillo O, Amador-Angulo L, Castro JR et al (2016) A comparative study of type-1 fuzzy logic systems, interval type-2 fuzzy logic systems, and generalized type-2 fuzzy logic systems in control problems. Inf Sci 354:257–274
8. Chen Y, Wang DZ, Ning W (2018) Forecasting by TSK general type-2 fuzzy logic systems optimized with genetic algorithms. Optim Control Appl Methods 39(1):393–409
9. Chen Y, Wang DZ (2019) Forecasting by designing Mamdani general type-2 fuzzy logic systems optimized with quantum particle swarm optimization algorithms. Trans Inst Meas Control 41(10):2886–2906
10. Sanchez MA, Castillo O, Castro JR (2015) Generalized type-2 fuzzy systems for controlling a mobile robot and a performance comparison with interval type-2 and type-1 fuzzy systems. Expert Syst Appl 42(14):5904–5914
11. Melin P, Gonzalez CI, Castro JR et al (2014) Edge-detection method for image processing based on generalized type-2 fuzzy logic. IEEE Trans Fuzzy Syst 22(6):1515–1525
12. Mendel JM (2001) Uncertain rule-based fuzzy logic systems: introduction and new directions. Prentice-Hall, Englewood Cliffs, NJ, USA
13. Chen Y (2018) Study on weighted Nagar-Bardini algorithms for centroid type-reduction of interval type-2 fuzzy logic systems. J Intell Fuzzy Syst 34(4):2417–2428
14. Chen Y (2019) Study on centroid type-reduction of interval type-2 fuzzy logic systems based on noniterative algorithms. Complexity 2019:1–12
15. Mendel JM (2013) On KM algorithms for solving type-2 fuzzy set problems. IEEE Trans Fuzzy Syst 21(3):426–446
16. Mendel JM, Wu HW (2006) Type-2 fuzzistics for symmetric interval type-2 fuzzy sets: part 1, forward problems. IEEE Trans Fuzzy Syst 14(6):781–792
17. Mendel JM, Liu FL (2007) Super-exponential convergence of the Karnik-Mendel algorithms for computing the centroid of an interval type-2 fuzzy set. IEEE Trans Fuzzy Syst 15(2):309–320
18. Wu DR, Mendel JM (2009) Enhanced Karnik-Mendel algorithms. IEEE Trans Fuzzy Syst 17(4):923–934
19. Liu XW, Mendel JM, Wu DR (2012) Study on enhanced Karnik-Mendel algorithms: initialization explanations and computation improvements. Inf Sci 184(1):75–91
20. Wu DR, Nie M (2011) Comparison and practical implementations of type reduction algorithms for type-2 fuzzy sets and systems. In: IEEE international conference on fuzzy systems, pp 3131–3138
21. Wu HW, Mendel JM (2002) Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems. IEEE Trans Fuzzy Syst 10(5):622–639
22. Coupland S, John R (2007) Geometric type-1 and type-2 fuzzy logic systems. IEEE Trans Fuzzy Syst 15(1):3–15
23. EI-Nagar AM, EI-Bardini M (2014) Simplified interval type-2 fuzzy logic system based on new type-reduction. J Intell Fuzzy Syst 27(4):1499–2010
24. Li JW, John R, Coupland S et al (2018) On Nie-Tan operator and type-reduction of interval type-2 fuzzy sets. IEEE Trans Fuzzy Syst 26(2):1036–1039
25. Mendel JM (2001) Uncertain rule-based fuzzy logic systems: introduction and new directions. Prentice-Hall, Englewood Cliffs, NJ, USA
26. Li JW, John R, Coupland S et al (2018) On Nie-Tan operator and type-reduction of interval type-2 fuzzy sets. IEEE Trans Fuzzy Syst 26(2):1036–1039
27. Chen Y, Wang DZ (2018) Study on centroid type-reduction of general type-2 fuzzy logic systems with weighted Nie-Tan algorithms. Soft Comput 22(22):7659–7678
28. Biglarbegian M, Melek WW, Mendel JM (2011) On the robustness of type-1 and interval type-2 fuzzy logic systems in modeling. Inf Sci 181(7):1325–1347
29. Biglarbegian M, Melek WW, Mendel JM (2010) On the stability of interval type-2 TSK fuzzy logic systems. IEEE Trans Syst Man Cybern B Cybern 40(3):798–818
30. Cheng P, Wang H, Stojanovic V et al (2021) Asynchronous fault detection observer for 2-D Markov jump systems. IEEE Trans Cybern (accepted)
31. Zhang X, Wang H, Stojanovic V et al (2021) Asynchronous fault detection for interval type-2 fuzzy nonhomogeneous higher-level Markov jump systems with uncertain transition probabilities. IEEE Trans Fuzzy Syst (accepted)
32. Cheng P, Chen M, Stojanovic V et al (2021) Asynchronous fault detection filtering for piecewise homogeneous Markov jump linear systems via a dual hidden Markov model. Mech Syst Signal Process 151(8):107353
33. Wu DR (2013) Approaches for reducing the computational cost of interval type-2 fuzzy logic systems: overview and comparisons. IEEE Trans Fuzzy Syst 21(1):80–99
34. Wu DR, Mendel JM (2019) Recommendations on designing practical interval type-2 fuzzy systems. Eng Appl Artif Intell 85:182–193
35. Chen Y, Wang DZ, Tong SC (2016) Forecasting studies by designing Mamdani interval type-2 fuzzy logic systems: with combination of BP algorithms and KM algorithms. Neurocomputing 174:1133–1146
36. Khosravi A, Nahavandi S, Creighton D et al (2012) Interval type-2 fuzzy logic systems for load forecasting: a comparative study. IEEE Trans Power Syst 27(3):1274–1282
37. Wang DZ, Chen Y (2018) Study on permanent magnetic drive forecasting by designing Takagi Sugeno Kang type interval type-2 fuzzy logic systems. Trans Inst Meas Control 40(6):2011–2023
38. Méndez GM, Hernandez MDLA (2013) Hybrid learning mechanism for interval A2–C1 type-2 non-singleton type-2 Takagi-Sugeno-Kang fuzzy logic systems. Inf Sci 220:149–169
39. Mendel JM, John R (2002) Type-2 fuzzy sets made simple. IEEE Trans Fuzzy Syst 10(2):117–127
40. Mendel JM, John R, Liu FL (2007) Interval type-2 fuzzy logic systems made simple. IEEE Trans Fuzzy Syst 14(6):808–821
41. Mo H, Wang FY, Zhou M et al (2014) Footprint of uncertainty for type-2 fuzzy sets. Inf Sci 272:96–110
42. Wang FY, Mo H (2017) Some fundamental issues on type-2 fuzzy sets. Acta Automat Sin 43(7):1114–1141
43. Wang T, Chen Y, Tong SC (2008) Fuzzy reasoning models and algorithms on type-2 fuzzy sets. Int J Innov Comput Inf Control 4(10):2451–2460
44. Chen Y (2019) Study on sampling based discrete Nie-Tan algorithms for computing the centroids of general type-2 fuzzy sets. IEEE Access 7(1):156984–156992
45. Khanesar MA, Jalalian A, Kaynak O (2017) Improving the speed of center of set type-reduction in interval type-2 fuzzy systems by eliminating the need for sorting. IEEE Trans Fuzzy Syst 25(5):1193–1206
46. Chen Y (2020) Study on sampling-based discrete noniterative algorithms for centroid type-reduction of interval type-2 fuzzy logic systems. Soft Comput 24(15):11819–11828
47. Gaxiola F, Melin P, Valdez F et al (2016) Optimization of type-2 fuzzy weights in backpropagation learning for neural networks using GAs and PSO. Appl Soft Comput 38:860–871
48. Hsu CH, Juang CF (2013) Evolutionary robot wall-following control using type-2 fuzzy controller with species-de-activated continuous ACO. IEEE Trans Fuzzy Syst 21(1):100–112
49. Tao CW, Taur JS, Chang CW et al (2012) Simplified type-2 fuzzy sliding controller for wing rocket system. Fuzzy Sets Syst 207(16):111–129
50. Ontiveros-Robles E, Melin P, Castillo O (2018) Comparative analysis of noise robustness of type 2 fuzzy logic controllers. Kybernetika 54(1):175–201
51. Cervantes L, Castillo O (2015) Type-2 fuzzy logic aggregation of multiple fuzzy controllers for airplane flight control. Inf Sci 324:247–256
52. Castillo O, Melin P, Ontiveros E et al (2019) A high-speed interval type 2 fuzzy system approach for dynamic parameter adaptation in metaheuristics. Eng Appl Artif Intell 85:666–680
53. Tong SC, Li YM (2010) Robust adaptive fuzzy backstepping output feedback tracking control for nonlinear system with dynamic uncertainties. Sci China Inf Sci 53(2):307–324
54. Tong SC, Li YM (2014) Observer-based adaptive fuzzy backstepping control of uncertain pure-feedback systems. Sci China Inf Sci 57(1):1–14
55. Lathamaheswari M, Nagarajan D, Kavikumar J et al (2021) Interval type-2 fuzzy aggregation operator in decision making and its application. Complex Intell Syst 7(3):1695–1708
56. Paik B, Mondal SK (2021) Representation and application of Fuzzy soft sets in type-2 environment. Complex Intell Syst 7(3):1597–1617

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.