



Research article

Study on sensible beginning divided-search enhanced Karnik-Mendel algorithms for centroid type-reduction of general type-2 fuzzy logic systems

Yang Chen* and Chenxi Li

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

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

Abstract: General type-2 fuzzy logic systems (GT2 FLSs) on the basis of alpha-plane representation of GT2 fuzzy sets (FSs) have attracted considerable attention in recent years. For the kernel type-reduction (TR) block of GT2 FLSs, the enhanced Karnik-Mendel (EKM) algorithm is the most popular approach. This paper proposes the sensible beginning divided-search EKM (SBDEKM) algorithms for completing the centroid TR of GT2 FLSs. Computer simulations are provided to show the performances of the SBDEKM algorithms. Compared with EKM algorithms and sensible beginning EKM (SBEKM) algorithms, the SBDEKM algorithms have almost the same accuracies and better computational efficiency.

Keywords: sensible beginning divided-search EKM algorithms; general type-2 fuzzy logic systems; centroid type-reduction; simulation; efficiency

Mathematics Subject Classification: 68WXX, 94D05

1. Introduction

Interval T2 (IT2) FLSs [1–3] have been applied to many fields with higher uncertainties and nonlinear characteristics. Despite this, since the alpha-plane (or z-slice) representation theory [4–7] of GT2 FSs was put forward by a few different research groups, much of the academic attention has shifted from IT2 fuzzy logic to GT2 fuzzy logic. As the computational complexity has been decreased significantly, they are increasingly being applied to in fields like border detection [8,9], fuzzy logic control [10,11], permanent magnetic drive [12–15], and so forth. As the secondary membership grades of GT2 FSs exist between zero and one, so that, they may be considered as more advanced order parameter models compared with IT2 FSs. Because the design degrees of freedom increase, GT2 FLSs have more of an ability to cope with issues arising from uncertainties in contrast to IT2 FLSs.

GT2 FLSs [7] are made up of five blocks: Fuzzifier, rules, inference, TR, and defuzzification. TR is the kernel block, which plays the role of transforming the GT2 FS to the type-1 (T1) FS. The block of defuzzification maps the T1 FS to the crisp output. The most popular enhanced Karnik-Mendel (EKM [16–23]) TR algorithms have the advantage of keeping the uncertainty flows between the lower membership functions (LMFs) and upper MFs (UMFs). Despite so, this kind of algorithm is still computationally intensive. Liu gave the initialization interpretations [18] for the EKM algorithms, then the sensible beginning (SB, or say reasonable initialization) EKM (SBEKM) algorithms [24] can be generated. Based on these studies, this paper further divides the search space [25] of SBEKM algorithms, and then proposes the sensible beginning divided-search EKM (SBDEKM) algorithms for performing the centroid TR for GT2 FLSs.

The remainder is organized as follows: Section 2 provides the background knowledge of GT2 FLSs. The SBDEKM algorithms are proposed to complete the centroid TR and defuzzification of GT2 FLSs in Section 3. Section 4 provides computer simulations to show the performance of the SBDEKM algorithms compared with the EKM algorithms and SBEKM algorithms. Finally, Section 5 gives the conclusions and expectations.

2. Background of GT2 FLSs

In general, we may categorize the GT2 FLSs into two types from the aspect of inference: Mamdani [7,26,27] and Takagi-Sugeno-Kang [12,29,30]. In this paper, without loss of generality, consider a GT2 FLS with n inputs $x_1 \in X_1, \dots, x_n \in X_n$ and one output \tilde{B} , among which, the sth is given as:

$$R^s : \text{if } x_1 \text{ is } \tilde{F}_1^s \text{ and } x_2 \text{ is } \tilde{F}_2^s \text{ and } \dots \text{ and } x_n \text{ is } \tilde{F}_n^s, \text{ then } y \text{ is } \tilde{G}^s, \quad (1)$$

in which $\tilde{F}_i^s (i=1, \dots, n; s=1, \dots, M)$ is the antecedent, and $\tilde{G}^s (s=1, 2, \dots, M)$ is the consequent.

For the sake of simplifying the expressions, here the singleton fuzzifier $x_i = x'_i$ is used, only one vertical slice (secondary membership) $\tilde{F}_i^s(x'_i)$ of antecedent \tilde{F}_i^s is activated, and the alpha-cut decomposition can be represented as:

$$\tilde{F}_i^s(x'_i) = \sup_{\forall \alpha \in [0,1]} \alpha / [c_{i,\alpha}^s(x'_i), d_{i,\alpha}^s(x'_i)]. \quad (2)$$

For each rule, the firing interval at the corresponding alpha-level is:

$$F_\alpha : \begin{cases} F_\alpha^s(x') \equiv [f_\alpha^s(x'), \bar{f}_\alpha^s(x')] \\ f_\alpha^s(x') \equiv T_{i=1}^p c_{i,\alpha}^s(x'_i) \\ \bar{f}_\alpha^s(x') \equiv T_{i=1}^p \bar{c}_{i,\alpha}^s(x'_i) \end{cases}, \quad (3)$$

where p denotes the number of antecedents, and T represents the minimum or product t-norm.

Let \tilde{G}_α^s be the alpha-plane of consequent \tilde{G}^s under the corresponding alpha-level, i.e.,

$$\tilde{G}_\alpha^s = \int_{\forall y \in Y} \tilde{G}_\alpha^s(y) / y = \int_{\forall y \in Y} [g_{L,\alpha}^s(y), g_{R,\alpha}^s(y)] / y. \quad (4)$$

In the following, we merge the firing interval of each rule with the related rule consequent

alpha-plane \tilde{G}_α^s to get the firing rule alpha-plane \tilde{B}_α^s , i.e.,

$$\tilde{B}_\alpha^s : \begin{cases} \text{FOU}(\tilde{B}_\alpha^s) = [f_{\tilde{B}_\alpha^s}(y|x'), \bar{f}_{\tilde{B}_\alpha^s}(y|x')] \\ f_{\tilde{B}_\alpha^s}(y|x') = f_{\tilde{B}_\alpha^s}^s(x') * g_{L,\alpha}^s(y) \\ \bar{f}_{\tilde{B}_\alpha^s}(y|x') = \bar{f}_{\tilde{B}_\alpha^s}^s(x') * g_{R,\alpha}^s(y) \end{cases} . \quad (5)$$

Aggregating all the \tilde{B}_α^s to get the output alpha-plane \tilde{B}_α , i.e.,

$$\tilde{B}_\alpha : \begin{cases} \text{FOU}(\tilde{B}_\alpha) = [f_{\tilde{B}_\alpha}(y|x'), \bar{f}_{\tilde{B}_\alpha}(y|x')] \\ f_{\tilde{B}_\alpha}(y|x') = f_{\tilde{B}_\alpha^1}(y|x') \vee \dots \vee f_{\tilde{B}_\alpha^M}(y|x') \\ \bar{f}_{\tilde{B}_\alpha}(y|x') = \bar{f}_{\tilde{B}_\alpha^1}(y|x') \vee \dots \vee \bar{f}_{\tilde{B}_\alpha^M}(y|x') \end{cases} . \quad (6)$$

Calculating the centroid of \tilde{B}_α at the corresponding alpha-level to obtain the type-reduced set $Y_{C,\alpha}(x')$, i.e.,

$$Y_{C,\alpha}(x') = C_{\tilde{B}_\alpha}(x') = \alpha / [l_{\tilde{B}_\alpha}(x'), r_{\tilde{B}_\alpha}(x')], \quad (7)$$

in which $r_{\tilde{B}_\alpha}(x')$ and $l_{\tilde{B}_\alpha}(x')$ can be computed by many different kinds of TR algorithms [16–21, 27,28] as:

$$l_{\tilde{B}_\alpha}(x') = \min_{f_{\tilde{B}_\alpha}(y_i) \in [f_{\tilde{B}_\alpha}(y_i), \bar{f}_{\tilde{B}_\alpha}(y_i)]} \frac{\sum_{i=1}^n y_i f_{R_{\tilde{B}_\alpha}}(y_i)}{\sum_{i=1}^n f_{R_{\tilde{B}_\alpha}}(y_i)}, \quad (8)$$

$$r_{\tilde{B}_\alpha}(x') = \max_{f_{\tilde{B}_\alpha}(y_i) \in [f_{\tilde{B}_\alpha}(y_i), \bar{f}_{\tilde{B}_\alpha}(y_i)]} \frac{\sum_{i=1}^n y_i f_{R_{\tilde{B}_\alpha}}(y_i)}{\sum_{i=1}^n f_{R_{\tilde{B}_\alpha}}(y_i)}, \quad (9)$$

where n is the number of sampling points, and the specific IT2 FS $R_{\tilde{B}_\alpha} = \alpha / \tilde{B}_\alpha$.

Aggregating all the $Y_{C,\alpha}$ to obtain the final T1 type-reduced set Y_C , i.e.,

$$Y_C = \sup_{\forall \alpha \in [0,1]} \alpha / Y_{C,\alpha}(x'). \quad (10)$$

Suppose that the number of alpha-planes is m , that is to say, alpha is decomposed into $\alpha_1, \alpha_2, \dots, \alpha_m$, and then the output should:

$$y(x') = \sum_{i=1}^m \alpha_i [(l_{\tilde{B}_{\alpha_i}}(x') + r_{\tilde{B}_{\alpha_i}}(x')) / 2] / \sum_{i=1}^m \alpha_i . \quad (11)$$

3. SBDEKM algorithms

First of all, we provide the sensible beginning (or say reasonable initialization) EKM algorithms [18, 24, 31,32] for finishing the centroid TR and defuzzification of GT2 FLSs. According to the inference, let the output of the GT2 FLSs be \tilde{B} , whose primary variable is discretized into $c = y_1 < \dots < y_n = d$, then $l_{\tilde{B}_\alpha}$ and $r_{\tilde{B}_\alpha}$ under the alpha-level can be computed by a continuous version of the KM algorithms:

$$l_{\tilde{B}_\alpha} = \min_{\eta \in [c,d]} F_{l_{\tilde{B}_\alpha}}(\eta) = \min_{\eta \in [c,d]} \frac{\int_c^\eta y \bar{f}_{R_{\tilde{B}_\alpha}}(y) dy + \int_\eta^d y \underline{f}_{R_{\tilde{B}_\alpha}}(y) dy}{\int_c^\eta \bar{f}_{R_{\tilde{B}_\alpha}}(y) dy + \int_\eta^d \underline{f}_{R_{\tilde{B}_\alpha}}(y) dy}, \quad (12)$$

$$r_{\tilde{B}_\alpha} = \max_{\eta \in [c,d]} F_{r_{\tilde{B}_\alpha}}(\eta) = \max_{\eta \in [c,d]} \frac{\int_c^\eta y \underline{f}_{R_{\tilde{B}_\alpha}}(y) dy + \int_\eta^d y \bar{f}_{R_{\tilde{B}_\alpha}}(y) dy}{\int_c^\eta \underline{f}_{R_{\tilde{B}_\alpha}}(y) dy + \int_\eta^d \bar{f}_{R_{\tilde{B}_\alpha}}(y) dy}. \quad (13)$$

For computing the two end points, Tables 1 and 2 give the computation steps for CKM and CEKM algorithms. According to the representations $F_{l_{\tilde{B}_\alpha}}$ in (12) and $F_{r_{\tilde{B}_\alpha}}$ in (13), i.e.,

$$\eta_{l_{\tilde{B}_\alpha}} = F_{l_{\tilde{B}_\alpha}}(\eta') = \eta', \quad (14)$$

$$\eta_{r_{\tilde{B}_\alpha}} = F_{r_{\tilde{B}_\alpha}}(\eta') = \eta'. \quad (15)$$

When $l_{\tilde{B}_\alpha} = \eta_{l_{\tilde{B}_\alpha}}$, and $r_{\tilde{B}_\alpha} = \eta_{r_{\tilde{B}_\alpha}}$, the iteration finishes, and we have that,

$$l_{\tilde{B}_\alpha} = F_{l_{\tilde{B}_\alpha}}(l_{\tilde{B}_\alpha}), \text{ and } r_{\tilde{B}_\alpha} = F_{r_{\tilde{B}_\alpha}}(r_{\tilde{B}_\alpha}). \quad (16)$$

For initializing the algorithms, as $y \in [c, d]$, let $\underline{f}_{R_{\tilde{B}_\alpha}}(y) = \bar{f}_{R_{\tilde{B}_\alpha}}(y) \equiv \theta(y)$, so that, $\theta(y) = [\bar{f}_{R_{\tilde{B}_\alpha}}(y) + \underline{f}_{R_{\tilde{B}_\alpha}}(y)]/2$. Then Eqs (12) and (13) turn out to be the same.

$$r_{\tilde{B}_\alpha} = l_{\tilde{B}_\alpha} = \frac{\int_c^d y \theta(y) dy}{\int_c^d \theta(y) dy} = \frac{\int_c^d y [\bar{f}_{R_{\tilde{B}_\alpha}}(y) + \underline{f}_{R_{\tilde{B}_\alpha}}(y)]/2 dy}{\int_c^d [\bar{f}_{R_{\tilde{B}_\alpha}}(y) + \underline{f}_{R_{\tilde{B}_\alpha}}(y)]/2 dy}. \quad (17)$$

The initialization of the CKM algorithms may be represented as $\eta^{(1)}$, that is to say,

$$\eta^{(1)} = \frac{\int_c^d y \theta(y) dy}{\int_c^d \theta(y) dy}. \quad (18)$$

Table 1. CKM algorithms [7,16–18,31,32] for computing the centroid.

| Step | For $l_{\tilde{B}_\alpha}, l_{\tilde{B}_\alpha} = \min_{\forall \theta(y) \in [f_{R_{\tilde{B}_\alpha}}(y), \bar{f}_{R_{\tilde{B}_\alpha}}(y)]} \frac{\int_c^d y\theta(y)dy}{\int_c^d \theta(y)dy}$ |
|------|--|
| 1 | Let $\theta(y) = [f_{R_{\tilde{B}_\alpha}}(y) + \bar{f}_{R_{\tilde{B}_\alpha}}(y)]/2$, and calculate $\eta' = \frac{\int_c^d y\theta(y)dy}{\int_c^d \theta(y)dy}$. Let $\theta(y) \equiv \bar{f}_{R_{\tilde{B}_\alpha}}(y)$ when $y \leq \eta'$, and $\theta(y) \equiv f_{R_{\tilde{B}_\alpha}}(y)$ when $y > \eta'$, and |
| 2 | calculate $\eta_{l_{\tilde{B}_\alpha}} = \frac{\int_c^d y\theta(y)dy}{\int_c^d \theta(y)dy}$. |
| 3 | Check whether $ \eta' - \eta_{l_{\tilde{B}_\alpha}} \leq \varepsilon$, if so, pause and let $l_{\tilde{B}_\alpha} = \eta_{l_{\tilde{B}_\alpha}}$, if it were not true, go to Step 4. |
| 4 | Let $\eta' = \eta_{l_{\tilde{B}_\alpha}}$ and go to the second step. |
| Step | For $r_{\tilde{B}_\alpha}, r_{\tilde{B}_\alpha} = \max_{\forall \theta(y) \in [f_{R_{\tilde{B}_\alpha}}(y), \bar{f}_{R_{\tilde{B}_\alpha}}(y)]} \frac{\int_c^d y\theta(y)dy}{\int_c^d \theta(y)dy}$ |
| 1 | Let $\theta(y) = [f_{R_{\tilde{B}_\alpha}}(y) + \bar{f}_{R_{\tilde{B}_\alpha}}(y)]/2$, and calculate $\eta' = \frac{\int_c^d y\theta(y)dy}{\int_c^d \theta(y)dy}$. Let $\theta(y) = f_{R_{\tilde{B}_\alpha}}(y)$ when $y \leq \eta'$, and $\theta(y) = \bar{f}_{R_{\tilde{B}_\alpha}}(y)$ when $y > \eta'$, and |
| 2 | calculate $\eta_{r_{\tilde{B}_\alpha}} = \frac{\int_c^d y\theta(y)dy}{\int_c^d \theta(y)dy}$. |
| 3 | Check whether $ \eta' - \eta_{r_{\tilde{B}_\alpha}} \leq \varepsilon$, if so, stop and let $r_{\tilde{B}_\alpha} = \eta_{r_{\tilde{B}_\alpha}}$, if it were not true, go to the next step. |
| 4 | Let $\eta' = \eta_{r_{\tilde{B}_\alpha}}$ and go to Step 2. |

Table 2. CEKM algorithms [7,16–18,31,32] for computing the centroid.

| Step | For $l_{\tilde{B}_\alpha}$ |
|------|---|
| 1 | Let $a = (d-c)/2.4 + c$, and calculate $\alpha = \int_c^a y \bar{f}_{R_{\tilde{B}_\alpha}}(y) dy + \int_a^d y \underline{f}_{R_{\tilde{B}_\alpha}}(y) dy$, $\beta = \int_c^a \bar{f}_{R_{\tilde{B}_\alpha}}(y) dy + \int_a^d \underline{f}_{R_{\tilde{B}_\alpha}}(y) dy$, $a' = \alpha / \beta$. |
| 2 | Check if $ a' - a \leq \varepsilon$, if so, stop and let $a' = l_{\tilde{B}_\alpha}$, if it were not true, return to the fourth step. |
| 3 | Calculate $s = \text{sign}(a' - a)$, $\alpha' = \alpha + s \int_{\min(a,a')}^{\max(a,a')} y [\bar{f}_{R_{\tilde{B}_\alpha}}(y) - \underline{f}_{R_{\tilde{B}_\alpha}}(y)] dy$, $\beta' = \beta + s \int_{\min(a,a')}^{\max(a,a')} [\bar{f}_{R_{\tilde{B}_\alpha}}(y) - \underline{f}_{R_{\tilde{B}_\alpha}}(y)] dy$, $a'' = \alpha' / \beta'$. |
| 4 | Let $a = a'$, $a' = a''$, $\alpha = \alpha'$, $\beta = \beta'$ and go to the Step 2. |
| Step | For $r_{\tilde{B}_\alpha}$ |
| 1 | Let $a = (d-c)/1.7 + c$, and compute $\alpha = \int_c^a y \underline{f}_{R_{\tilde{B}_\alpha}}(y) dy + \int_a^d y \bar{f}_{R_{\tilde{B}_\alpha}}(y) dy$, $\beta = \int_c^a \underline{f}_{R_{\tilde{B}_\alpha}}(y) dy + \int_a^d \bar{f}_{R_{\tilde{B}_\alpha}}(y) dy$, $a' = \alpha / \beta$. |
| 2 | Check if $ a' - a \leq \varepsilon$, if so, stop and let $a' = r_{\tilde{B}_\alpha}$, if it were not true, return to the fourth step. |
| 3 | Calculate $s = \text{sign}(a' - a)$, $\alpha' = \alpha - s \int_{\min(a,a')}^{\max(a,a')} y [\bar{f}_{R_{\tilde{B}_\alpha}}(y) - \underline{f}_{R_{\tilde{B}_\alpha}}(y)] dy$, $\beta' = \beta - s \int_{\min(a,a')}^{\max(a,a')} [\bar{f}_{R_{\tilde{B}_\alpha}}(y) - \underline{f}_{R_{\tilde{B}_\alpha}}(y)] dy$, $a'' = \alpha' / \beta'$. |
| 4 | Let $a = a'$, $a' = a''$, $\alpha = \alpha'$, and $\beta = \beta'$ and go to the Step 2. |

Tables 3 and 4 give the computation steps for the discrete KM, and discrete EKM algorithms, respectively. Here the discrete representation of $\eta^{(1)}$ can be written as:

$$k^{(1)} = \{k \mid y_k \leq \frac{\sum_{i=1}^n y_i [\bar{f}_{R_{\tilde{B}_\alpha}}(y_i) + \underline{f}_{R_{\tilde{B}_\alpha}}(y_i)]}{\sum_{i=1}^n [\bar{f}_{R_{\tilde{B}_\alpha}}(y_i) + \underline{f}_{R_{\tilde{B}_\alpha}}(y_i)]} < y_{k+1}, 1 \leq k \leq n-1\}. \quad (19)$$

Table 3. KM algorithms [7,16–18,31,32] for computing the centroid.

| Step | $l_{\bar{B}_\alpha} = \min_{\forall \theta_i \in [f_{\underline{R}_{\bar{B}_\alpha}}(y_i), \bar{f}_{\bar{R}_{\bar{B}_\alpha}}(y_i)]} \frac{\sum_{i=1}^n y_i \theta_i}{\sum_{i=1}^n \theta_i}$ |
|------|---|
| 1 | Let $\theta(i) = [f_{\underline{R}_{\bar{B}_\alpha}}(y_i) + \bar{f}_{\bar{R}_{\bar{B}_\alpha}}(y_i)] / 2$, and compute $a' = (\sum_{i=1}^n y_i \theta_i) / (\sum_{i=1}^n \theta_i)$. |
| 2 | Seek for $k' (1 \leq k' \leq n-1)$, satisfying $y_{k'} \leq a' < y_{k'+1}$. Let $\theta_i = \bar{f}_{\bar{R}_{\bar{B}_\alpha}}(y_i)$ as $i \leq k'$, and $\theta_i = f_{\underline{R}_{\bar{B}_\alpha}}(y_i)$ as $i > k'$, |
| 3 | and compute $l_{\bar{B}_\alpha}(k') = (\sum_{i=1}^n y_i \theta_i) / (\sum_{i=1}^n \theta_i)$. |
| 4 | Check if $l_{\bar{B}_\alpha}(k') = a'$, if so, pause and let $l_{\bar{B}_\alpha}(k') = l_{\bar{B}_\alpha}$, and $k' = L$, if it does not satisfy, return to the next step. |
| 5 | Let $a' = l_{\bar{B}_\alpha}(k')$ and go to the second step. |
| Step | $r_{\bar{B}_\alpha} = \max_{\forall \theta_i \in [f_{\underline{R}_{\bar{B}_\alpha}}(y_i), \bar{f}_{\bar{R}_{\bar{B}_\alpha}}(y_i)]} \frac{\sum_{i=1}^n y_i \theta_i}{\sum_{i=1}^n \theta_i}$ |
| 1 | Let $\theta(i) = [f_{\underline{R}_{\bar{B}_\alpha}}(y_i) + \bar{f}_{\bar{R}_{\bar{B}_\alpha}}(y_i)] / 2$, and compute $a' = (\sum_{i=1}^n y_i \theta_i) / (\sum_{i=1}^n \theta_i)$. |
| 2 | Seek for $k' (1 \leq k' \leq n-1)$, satisfying $y_{k'} \leq a' < y_{k'+1}$. Let $\theta_i = f_{\underline{R}_{\bar{B}_\alpha}}(y_i)$ as $i \leq k'$, and $\theta_i = \bar{f}_{\bar{R}_{\bar{B}_\alpha}}(y_i)$ as $i > k'$, |
| 3 | and compute $r_{\bar{B}_\alpha}(k') = (\sum_{i=1}^n y_i \theta_i) / (\sum_{i=1}^n \theta_i)$. |
| 4 | Check if $r_{\bar{B}_\alpha}(k') = a'$, if so, pause and let $r_{\bar{B}_\alpha}(k') = r_{\bar{B}_\alpha}$, and $k' = R$, if it is not true, return to the next step. |
| 5 | Let $a' = r_{\bar{B}_\alpha}(k')$ and go to the second step. |

Table 4. EKM algorithms [7,16–18,31,32] for computing the centroid.

| Step | For $l_{\tilde{B}_\alpha}$ |
|------|---|
| | Let $k = [n/2.4]$ and calculate |
| 1 | $\alpha = \sum_{i=1}^k y_i \bar{f}_{R_{\tilde{B}_\alpha}}(y_i) + \sum_{i=k+1}^n y_i \underline{f}_{R_{\tilde{B}_\alpha}}(y_i), \beta = \sum_{i=1}^k \bar{f}_{R_{\tilde{B}_\alpha}}(y_i) + \sum_{i=k+1}^n \underline{f}_{R_{\tilde{B}_\alpha}}(y_i), a' = \alpha / \beta.$ |
| 2 | Seek for $k' (1 \leq k' < n-1)$, satisfying $y_{k'} \leq a' < y_{k'+1}$. |
| 3 | Check if $k' = k$, if so, stop and let $a' = l_{\tilde{B}_\alpha}$, $k = L$, if it is not true, return to the next step. |
| 4 | Compute $s = \text{sign}(k' - k)$, $\alpha' = \alpha + s \sum_{i=\min(k,k')+1}^{\max(k,k')} y_i [\bar{f}_{R_{\tilde{B}_\alpha}}(y_i) - \underline{f}_{R_{\tilde{B}_\alpha}}(y_i)]$, $\beta' = \beta + s \sum_{i=\min(k,k')+1}^{\max(k,k')} [\bar{f}_{R_{\tilde{B}_\alpha}}(y_i) - \underline{f}_{R_{\tilde{B}_\alpha}}(y_i)], a'' = \alpha' / \beta'.$ |
| 5 | Let $a' = a'', \alpha = \alpha', \beta = \beta',$ and $k = k'$ and go to Step 2. |
| Step | For $r_{\tilde{B}_\alpha}$ |
| | Let $k = [n/1.7]$ and calculate |
| 1 | $\alpha = \sum_{i=1}^k y_i \underline{f}_{R_{\tilde{B}_\alpha}}(y_i) + \sum_{i=k+1}^n y_i \bar{f}_{R_{\tilde{B}_\alpha}}(y_i), \beta = \sum_{i=1}^k \underline{f}_{R_{\tilde{B}_\alpha}}(y_i) + \sum_{i=k+1}^n \bar{f}_{R_{\tilde{B}_\alpha}}(y_i), a' = \alpha / \beta.$ |
| 2 | Seek for $k' (1 \leq k' < n-1)$, satisfying $y_{k'} \leq a' < y_{k'+1}$. |
| 3 | Check if $k' = k$, if so, stop and let $a' = r_{\tilde{B}_\alpha}$, $k = R$, if it is not true, go to the next step. |
| 4 | Calculate $s = \text{sign}(k' - k)$, $\alpha' = \alpha - s \sum_{i=\min(k,k')+1}^{\max(k,k')} y_i [\bar{f}_{R_{\tilde{B}_\alpha}}(y_i) - \underline{f}_{R_{\tilde{B}_\alpha}}(y_i)]$, $\beta' = \beta - s \sum_{i=\min(k,k')+1}^{\max(k,k')} [\bar{f}_{R_{\tilde{B}_\alpha}}(y_i) - \underline{f}_{R_{\tilde{B}_\alpha}}(y_i)], a'' = \alpha' / \beta'.$ |
| 5 | Let $a' = a'', \alpha = \alpha', \beta = \beta',$ and $k = k'$ and go to Step 2. |

Suppose that,

$$\rho = \frac{\int_c^d \bar{f}_{R_{\tilde{B}_\alpha}}(y) dy}{\int_c^d \underline{f}_{R_{\tilde{B}_\alpha}}(y) dy}. \quad (20)$$

For $y \in [c, d]$, as $\bar{f}_{R_{\tilde{B}_\alpha}}(y) \geq \underline{f}_{R_{\tilde{B}_\alpha}}(y)$, therefore, $\rho \geq 1$. Let both $\bar{f}_{R_{\tilde{B}_\alpha}}(y)$ and $\underline{f}_{R_{\tilde{B}_\alpha}}(y)$ be

constants, that is to say, $\bar{f}_{R_{\bar{b}_\alpha}}(y) = \rho N > 0$, so that,

$$F_{l_{\bar{b}_\alpha}}(\eta) = \frac{\int_c^\eta y \bar{f}_{R_{\bar{b}_\alpha}}(y) dy + \int_\eta^d y \underline{f}_{R_{\bar{b}_\alpha}}(y) dy}{\int_c^\eta \bar{f}_{R_{\bar{b}_\alpha}}(y) dy + \int_\eta^d \underline{f}_{R_{\bar{b}_\alpha}}(y) dy} = \frac{\rho(\eta^2 - c^2) + (d^2 - \eta^2)}{2[\rho(\eta - c) + (d - \eta)]}. \quad (21)$$

Find the derivative of $F_{l_{\bar{b}_\alpha}}(\eta)$ to η , that is to say,

$$F'_{l_{\bar{b}_\alpha}}(\eta) = \frac{(\rho - 1)[\rho(\eta - c)^2 - (d - \eta)^2]}{2[\rho(\eta - c) + (d - \eta)]^2}. \quad (22)$$

Let $F'_{l_{\bar{b}_\alpha}}(\eta) = 0$, so that,

$$\frac{(d - \eta)^2}{(\eta - c)^2} = \rho \Rightarrow \frac{d - \eta}{\eta - c} = \sqrt{\rho}. \quad (23)$$

Therefore,

$$\eta_{l_{\bar{b}_\alpha}} = c + \frac{d - c}{1 + \sqrt{\rho}}. \quad (24)$$

As $\eta \in [a, \eta_{l_{\bar{b}_\alpha}})$, $F'_{l_{\bar{b}_\alpha}}(\eta) < 0$; and as $\eta \in (\eta_{l_{\bar{b}_\alpha}}, b]$, $F'_{l_{\bar{b}_\alpha}}(\eta) > 0$. So $\eta_{l_{\bar{b}_\alpha}}$ is the minimum value of $F_{l_{\bar{b}_\alpha}}(\eta)$.

In a similar manner, let us seek the maximum value of $F_{r_{\bar{b}_\alpha}}(\eta)$, i.e.,

$$F_{r_{\bar{b}_\alpha}}(\eta) = \frac{\int_c^\eta y \underline{f}_{R_{\bar{b}_\alpha}}(y) dy + \int_\eta^d y \bar{f}_{R_{\bar{b}_\alpha}}(y) dy}{\int_c^\eta \underline{f}_{R_{\bar{b}_\alpha}}(y) dy + \int_\eta^d \bar{f}_{R_{\bar{b}_\alpha}}(y) dy} = \frac{(\eta^2 - c^2) + \rho(d^2 - \eta^2)}{2[(\eta - c) + \rho(d - \eta)]}. \quad (25)$$

Find the derivative of $F_{r_{\bar{b}_\alpha}}(\eta)$ to η , that is to say,

$$F'_{r_{\bar{b}_\alpha}}(\eta) = -\frac{(\rho - 1)[(\eta - a)^2 - \rho(b - \eta)^2]}{2[(\eta - a) + \rho(b - \eta)]^2}. \quad (26)$$

Let $F'_{r_{\bar{b}_\alpha}}(\eta) = 0$, so that,

$$\frac{(d - \eta)^2}{(\eta - c)^2} = \frac{1}{\rho} \Rightarrow \frac{d - \eta}{\eta - c} = \sqrt{1/\rho}. \quad (27)$$

Therefore, the solution of above equation can be written as:

$$\eta_{r_{\bar{b}_\alpha}} = c + \frac{d - c}{1 + \sqrt{1/\rho}}. \quad (28)$$

As $\eta \in [c, \eta_{r_{\bar{b}_\alpha}})$, $F'_{r_{\bar{b}_\alpha}}(\eta) > 0$; and as $\eta \in (\eta_{r_{\bar{b}_\alpha}}, d]$, $F'_{r_{\bar{b}_\alpha}}(\eta) < 0$. Therefore, $\eta_{r_{\bar{b}_\alpha}}$ is the minimum value

of $F_{r_{\tilde{B}_\alpha}}(\eta)$.

Considering the Eqs (24) and (28) comprehensively, the sensible beginning method for η may be represented as $\eta^{(2)}$, i.e.,

$$\eta^{(2)} = \begin{cases} c + \frac{d-c}{1+\sqrt{\rho}} & \text{for } l_{\tilde{B}_\alpha}, \\ c + \frac{d-c}{1+\sqrt{1/\rho}} & \text{for } r_{\tilde{B}_\alpha}. \end{cases} \quad (29)$$

Comparing the initializations of discrete EKM and CEKM algorithms, we can get the discrete.

Table 5. SBEKM algorithms [16–18,31,32] for computing the centroid.

| Step | For $l_{\tilde{B}_\alpha}$ |
|------|---|
| | Let $k = \lceil n / (1 + \sqrt{\rho}) \rceil$ and compute |
| 1 | $\alpha = \sum_{i=1}^k y_i \bar{f}_{R_{\tilde{B}_\alpha}}(y_i) + \sum_{i=k+1}^n y_i \underline{f}_{R_{\tilde{B}_\alpha}}(y_i), \beta = \sum_{i=1}^k \bar{f}_{R_{\tilde{B}_\alpha}}(y_i) + \sum_{i=k+1}^n \underline{f}_{R_{\tilde{B}_\alpha}}(y_i), a' = \alpha / \beta.$ |
| 2 | Seek for $k' (1 \leq k' < n-1)$, satisfying $y_{k'} \leq a' < y_{k'+1}$. |
| 3 | Check if $k' = k$, if so, stop and let $a' = l_{\tilde{B}_\alpha}$, $k = L$, if not, go to the next step. |
| 4 | Compute $s = \text{sign}(k' - k)$, $\alpha' = \alpha + s \sum_{i=\min(k,k')+1}^{\max(k,k')} y_i [\bar{f}_{R_{\tilde{B}_\alpha}}(y_i) - \underline{f}_{R_{\tilde{B}_\alpha}}(y_i)]$, $\beta' = \beta + s \sum_{i=\min(k,k')+1}^{\max(k,k')} [\bar{f}_{R_{\tilde{B}_\alpha}}(y_i) - \underline{f}_{R_{\tilde{B}_\alpha}}(y_i)], a'' = \alpha' / \beta'.$ |
| 5 | Let $a' = a'', \alpha = \alpha', \beta = \beta'$, and $k = k'$ and go to Step 2. |
| Step | For $r_{\tilde{B}_\alpha}$ |
| | Let $k = \lceil n / (1 + \sqrt{1/\rho}) \rceil$ and compute |
| 1 | $\alpha = \sum_{i=1}^k y_i \underline{f}_{R_{\tilde{B}_\alpha}}(y_i) + \sum_{i=k+1}^n y_i \bar{f}_{R_{\tilde{B}_\alpha}}(y_i), \beta = \sum_{i=1}^k \underline{f}_{R_{\tilde{B}_\alpha}}(y_i) + \sum_{i=k+1}^n \bar{f}_{R_{\tilde{B}_\alpha}}(y_i), a' = \alpha / \beta.$ |
| 2 | Seek for $k' (1 \leq k' < n-1)$, satisfying $y_{k'} \leq a' < y_{k'+1}$. |
| 3 | Check if $k' = k$, if so, stop and let $a' = r_{\tilde{B}_\alpha}$, $k = R$, if not, go to the next step. |
| 4 | Calculate $s = \text{sign}(k' - k)$, $\alpha' = \alpha - s \sum_{i=\min(k,k')+1}^{\max(k,k')} y_i [\bar{f}_{R_{\tilde{B}_\alpha}}(y_i) - \underline{f}_{R_{\tilde{B}_\alpha}}(y_i)]$, $\beta' = \beta - s \sum_{i=\min(k,k')+1}^{\max(k,k')} [\bar{f}_{R_{\tilde{B}_\alpha}}(y_i) - \underline{f}_{R_{\tilde{B}_\alpha}}(y_i)], a'' = \alpha' / \beta'.$ |
| 5 | Let $a' = a'', \alpha = \alpha', \beta = \beta'$, and $k = k'$ and go to Step 2. |

Table 6. SBCEKM algorithms [16–18,31,32] for computing the centroid.

| Step | For $l_{\tilde{B}_\alpha}$ |
|------|---|
| | Let $\rho = (\int_c^d \bar{f}_{R_{\tilde{B}_\alpha}}(y)dy) / (\int_c^d \underline{f}_{R_{\tilde{B}_\alpha}}(y)dy)$, $a = c + (d - c) / (\sqrt{\rho} + 1)$, |
| 1 | and compute $\alpha = \int_c^a y \bar{f}_{R_{\tilde{B}_\alpha}}(y)dy + \int_a^d y \underline{f}_{R_{\tilde{B}_\alpha}}(y)dy$, $\beta = \int_c^a \bar{f}_{R_{\tilde{B}_\alpha}}(y)dy + \int_a^d \underline{f}_{R_{\tilde{B}_\alpha}}(y)dy$, $a' = \alpha / \beta$. |
| 2 | Check if $ a' - a \leq \varepsilon$, if so, stop and let $a' = l_{\tilde{B}_\alpha}$, if not, go to the fourth step. |
| 3 | Calculate $s = \text{sign}(a' - a)$, $\alpha' = \alpha + s \int_{\min(a,a')}^{\max(a,a')} y [\bar{f}_{R_{\tilde{B}_\alpha}}(y) - \underline{f}_{R_{\tilde{B}_\alpha}}(y)]dy$, $\beta' = \beta + s \int_{\min(a,a')}^{\max(a,a')} [\bar{f}_{R_{\tilde{B}_\alpha}}(y) - \underline{f}_{R_{\tilde{B}_\alpha}}(y)]dy$, $a'' = \alpha' / \beta'$. |
| 4 | Let $a = a'$, $a' = a''$, $\alpha = \alpha'$, $\beta = \beta'$ and go to the Step 2. |
| Step | For $r_{\tilde{B}_\alpha}$ |
| | Let $\rho = (\int_c^d \bar{f}_{R_{\tilde{B}_\alpha}}(y)dy) / (\int_c^d \underline{f}_{R_{\tilde{B}_\alpha}}(y)dy)$, $a = c + (d - c) / (\sqrt{1/\rho} + 1)$, |
| 1 | and compute $\alpha = \int_c^a y \underline{f}_{R_{\tilde{B}_\alpha}}(y)dy + \int_a^d y \bar{f}_{R_{\tilde{B}_\alpha}}(y)dy$, $\beta = \int_c^a \underline{f}_{R_{\tilde{B}_\alpha}}(y)dy + \int_a^d \bar{f}_{R_{\tilde{B}_\alpha}}(y)dy$, $a' = \alpha / \beta$. |
| 2 | Check if $ a' - a \leq \varepsilon$, if so, stop and let $a' = r_{\tilde{B}_\alpha}$, if not, go to the fourth step. |
| 3 | Compute $s = \text{sign}(a' - a)$, $\alpha' = \alpha - s \int_{\min(a,a')}^{\max(a,a')} y [\bar{f}_{R_{\tilde{B}_\alpha}}(y) - \underline{f}_{R_{\tilde{B}_\alpha}}(y)]dy$, $\beta' = \beta - s \int_{\min(a,a')}^{\max(a,a')} [\bar{f}_{R_{\tilde{B}_\alpha}}(y) - \underline{f}_{R_{\tilde{B}_\alpha}}(y)]dy$, $a'' = \alpha' / \beta'$. |
| 4 | Let $a = a'$, $a' = a''$, $\alpha = \alpha'$, $\beta = \beta'$ and go to the Step 2. |

form of Eq (29) as:

$$k^{(2)} = \begin{cases} [n / (1 + \sqrt{\rho})] & \text{for } l_{\tilde{B}_\alpha}, \\ [n / (1 + \sqrt{1/\rho})] & \text{for } r_{\tilde{B}_\alpha}. \end{cases} \quad (30)$$

in which $\rho = \frac{\sum_{i=1}^n \bar{f}_{R_{\tilde{B}_\alpha}}(y)}{\sum_{i=1}^n \underline{f}_{R_{\tilde{B}_\alpha}}(y)}$.

Interestingly, 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 the Eq (30) turns out to be the initialization of the EKM algorithms.

Meanwhile, for the SBEKM algorithms, ρ should be determined according to the corresponding experiments. Then the computation steps of the SBEKM and SBCEKM algorithms are given in Tables 5 and 6, respectively.

In order to give the proposed SBDEKM algorithms, we divide the searching space for SBEKM algorithms. Wang et al. proposed the divided-search EKM (DEKM) algorithms [20] for calculating the centroids of IT2 FSs. Here we first extend it for completing the centroid TR for GT2 FLSs. For computing $l_{\tilde{B}_\alpha}$, the initialization of the DEKM algorithms is:

$$k = \text{round}(n/2.4). \quad (31)$$

The initialization of the SBDEKM algorithms is:

$$k = \text{round}[n/(1 + \sqrt{\rho})]. \quad (32)$$

Next, we introduce how to divide the searching points for the SBDEKM (or DEKM) algorithms. Here, they are given as: $\text{round}(k/2)$, k (or $\text{round}[(k+n)/2]$). The analysis process is as follows:

At the corresponding alpha-level, for $l_{\tilde{B}_\alpha}$, initialize k and find k' . Comparing the values of the current $l_{\tilde{B}_\alpha} = \alpha/\beta$, y_k , and y_{k+1} . There may be three major situations:

Major situation 1: If $l_{\tilde{B}_\alpha} < y_k$, judge the relations between $l_{\tilde{B}_\alpha}$ and $y_{\text{round}(k/2)}$, for which there are three minor situations: i) $l_{\tilde{B}_\alpha} < y_{\text{round}(k/2)}$, ii) $y_{\text{round}(k/2)} \leq l_{\tilde{B}_\alpha} < y_{\text{round}(k/2)+1}$, and iii) $y_{\text{round}(k/2)+1} \leq l_{\tilde{B}_\alpha}$.

In i): Search down from $\text{round}(k/2) - 1$ to 1, seek the k' which satisfies $y_{k'} \leq \alpha/\beta \leq y_{k'+1}$.

In ii): $L = \text{round}(k/2)$.

In iii): Search up from $\text{round}(k/2) + 1$ to $k - 1$, find the k' which satisfies $y_{k'} \leq \alpha/\beta \leq y_{k'+1}$.

Major situation 2: If $y_k \leq l_{\tilde{B}_\alpha} < y_{k+1}$, $L = k$.

Major situation 3: If $y_{k+1} \leq l_{\tilde{B}_\alpha}$, then judge the relations between $l_{\tilde{B}_\alpha}$ and $y_{\text{round}[(k+n)/2]}$. There may be three minor situations: i) $l_{\tilde{B}_\alpha} < y_{\text{round}[(k+n)/2]}$, ii) $y_{\text{round}[(k+n)/2]} \leq l_{\tilde{B}_\alpha} < y_{\text{round}[(k+n)/2]+1}$, and iii) $y_{\text{round}[(k+n)/2]+1} \leq l_{\tilde{B}_\alpha}$.

In i): Search down from $\text{round}[(k+n)/2]$ to 1, seek the k' which satisfies $y_{k'} \leq \alpha/\beta \leq y_{k'+1}$.

In ii): $L = \text{round}[(k+n)/2]$.

In iii): Search up from $L = \text{round}[(k+n)/2] + 1$ to $n - 1$, find the k' which satisfies $y_{k'} \leq \alpha/\beta \leq y_{k'+1}$.

For $r_{\tilde{B}_\alpha}$, initialize k and find k' . Comparing the values of the current $r_{\tilde{B}_\alpha} = \alpha/\beta$, y_k , and y_{k+1} . There maybe three major situations:

Major situation 1: If $r_{\tilde{B}_\alpha} < y_k$, judge the relations between $r_{\tilde{B}_\alpha}$ and $y_{\text{round}(k/2)}$, for which there are three minor situations: i) $r_{\tilde{B}_\alpha} < y_{\text{round}(k/2)}$, ii) $y_{\text{round}(k/2)} \leq r_{\tilde{B}_\alpha} < y_{\text{round}(k/2)+1}$, and iii)

$$y_{\text{round}(k/2)+1} \leq r_{\tilde{B}_\alpha}.$$

In i): Search down from $\text{round}(k/2)-1$ to 1, seek for the k' which satisfies $y_{k'} \leq \alpha/\beta \leq y_{k'+1}$.

In ii): $R = \text{round}(k/2)$.

In iii): Search up from $\text{round}(k/2)+1$ to $k-1$, find the k' which satisfies $y_{k'} \leq \alpha/\beta \leq y_{k'+1}$.

Major situation 2: If $y_k \leq r_{\tilde{B}_\alpha} < y_{k+1}$, $R = k$.

Major situation 3: If $y_{k+1} \leq r_{\tilde{B}_\alpha}$, then judge the relations between $r_{\tilde{B}_\alpha}$ and $y_{\text{round}[(k+n)/2]}$. There may be three minor situations: i) $r_{\tilde{B}_\alpha} < y_{\text{round}[(k+n)/2]}$, ii) $y_{\text{round}[(k+n)/2]} \leq r_{\tilde{B}_\alpha} < y_{\text{round}[(k+n)/2]+1}$, and iii)

$$y_{\text{round}[(k+n)/2]+1} \leq r_{\tilde{B}_\alpha}.$$

In i): Search down from $\text{round}[(k+n)/2]$ to 1, seek for the k' which satisfies $y_{k'} \leq \alpha/\beta \leq y_{k'+1}$.

In ii): $R = \text{round}[(k+n)/2]$.

In iii): Search up from $R = \text{round}[(k+n)/2]+1$ to $n-1$, find the k' which satisfies $y_{k'} \leq \alpha/\beta \leq y_{k'+1}$.

Finally, we provide the computation steps for the SBDEKM algorithms.

For $l_{\tilde{B}_\alpha}$,

Step 1. Set $k = \lceil n / (1 + \sqrt{\rho}) \rceil$.

Step 2. Compute $\alpha = \sum_{i=1}^k y_i \bar{f}_{\tilde{B}_\alpha}(y_i) + \sum_{i=k+1}^n y_i \underline{f}_{\tilde{B}_\alpha}(y_i)$, and $\beta = \sum_{i=1}^k \bar{f}_{\tilde{B}_\alpha}(y_i) + \sum_{i=k+1}^n \underline{f}_{\tilde{B}_\alpha}(y_i)$.

Step 3. If $\alpha/\beta < y_k$ and $\alpha/\beta < y_{\text{round}(k/2)}$, then search down from $\text{round}(k/2)-1$ to 1, seek the k' which satisfies $y_{k'} \leq \alpha/\beta \leq y_{k'+1}$, and go to Step 10.

Step 4. If $y_{\text{round}(k/2)} \leq \alpha/\beta < y_{\text{round}(k/2)+1}$, then $l_{\tilde{B}_\alpha} = \alpha/\beta$, $L = k$, and go to Step 10.

Step 5. If $y_{\text{round}(k/2)} \leq \alpha/\beta < y_k$, then search down from $\text{round}(k/2)+1$ to $k-1$, find the k' which satisfies $y_{k'} \leq \alpha/\beta \leq y_{k'+1}$, and go to Step 10.

Step 6. If $y_k \leq \alpha/\beta < y_{k+1}$, then $l_{\tilde{B}_\alpha} = \alpha/\beta$, $L = k$, and go to Step 10.

Step 7. If $y_{k+1} \leq \alpha/\beta < y_{\text{round}[(k+n)/2]}$, then search down from $\text{round}[(k+n)/2]-1$ to $k+1$, find the k' which satisfies $y_{k'} \leq \alpha/\beta \leq y_{k'+1}$, and go to Step 10.

Step 8. If $y_{\text{round}[(k+n)/2]} \leq \alpha/\beta < y_{\text{round}[(k+n)/2]+1}$, let $l_{\tilde{B}_\alpha} = \alpha/\beta$, $L = \text{round}[(k+n)/2]$, and go to Step 10.

Step 9. If $y_{k+1} \leq \alpha/\beta$ and $y_{\text{round}[(k+n)/2]+1} \leq \alpha/\beta$, then search up from $\text{round}[(k+n)/2]+1$ to $n-1$, seek for the k' which meets $y_{k'} \leq \alpha/\beta \leq y_{k'+1}$, and go to Step 10.

Step 10. Calculate $s = \text{sign}(k' - k)$,

$$\alpha' = \alpha + s \sum_{i=\min(k,k')+1}^{\max(k,k')} y_i [\bar{f}_{\tilde{B}_\alpha}(y_i) - \underline{f}_{\tilde{B}_\alpha}(y_i)], \text{ and } \beta' = \beta + s \sum_{i=\min(k,k')+1}^{\max(k,k')} [\bar{f}_{\tilde{B}_\alpha}(y_i) - \underline{f}_{\tilde{B}_\alpha}(y_i)].$$

Step 11. Update $k = k'$.

Step 12. Search down from k to 1, seek the k' which satisfies $y_{k'} \leq \alpha/\beta \leq y_{k'+1}$.

Step 13. While $k' \neq k$.

Step 14. Update $\alpha' = \alpha - \sum_{i=k'+1}^k y_i [\bar{f}_{\tilde{B}_\alpha}(y_i) - \underline{f}_{\tilde{B}_\alpha}(y_i)]$, $\beta' = \beta - \sum_{i=k'+1}^k [\bar{f}_{\tilde{B}_\alpha}(y_i) - \underline{f}_{\tilde{B}_\alpha}(y_i)]$, and $k = k'$.

Step 15. Search down from k to 1, seek the k' which satisfies $y_{k'} \leq \alpha / \beta \leq y_{k'+1}$.

Step 16. End while.

Step 17. Let $l_{\tilde{B}} = \alpha / \beta$, $L = k$.

For $r_{\tilde{B}_\alpha}$:

Step 1. Set $k = \lceil n / (1 + \sqrt{1/\rho}) \rceil$.

Step 2. Compute $\alpha = \sum_{i=1}^k y_i \underline{f}_{\tilde{B}_\alpha}(y_i) + \sum_{i=k+1}^n y_i \bar{f}_{\tilde{B}_\alpha}(y_i)$, and $\beta = \sum_{i=1}^k \underline{f}_{\tilde{B}_\alpha}(y_i) + \sum_{i=k+1}^n \bar{f}_{\tilde{B}_\alpha}(y_i)$.

Step 3. If $\alpha / \beta < y_k$ and $\alpha / \beta < y_{\text{round}(k/2)}$, then search down from $\text{round}(k/2) - 1$ to 1, seek for the k' which satisfies $y_{k'} \leq \alpha / \beta \leq y_{k'+1}$, and go to Step 10.

Step 4. If $y_{\text{round}(k/2)+1} \leq \alpha / \beta < y_k$, then search up from $\text{round}(k/2) + 1$ to $k - 1$, find the k' which meets $y_{k'} \leq \alpha / \beta \leq y_{k'+1}$, and go to Step 10.

Step 5. If $y_{\text{round}(k/2)} \leq \alpha / \beta < y_{\text{round}(k/2)+1}$, let $r_{\tilde{B}_\alpha} = \alpha / \beta$, $R = k$, and go to Step 10.

Step 6. If $y_{k+1} \leq \alpha / \beta$ and $y_{\text{round}[(k+n)/2]+1} \leq \alpha / \beta$, then search up from $\text{round}[(k+n)/2] + 1$ to $n - 1$, seek for the k' which satisfies $y_{k'} \leq \alpha / \beta \leq y_{k'+1}$, and go to Step 10.

Step 7. If $y_{k+1} \leq \alpha / \beta < y_{\text{round}[(k+n)/2]}$, then search down from $\text{round}[(k+n)/2] - 1$ to $k + 1$, find the k' which satisfies $y_{k'} \leq \alpha / \beta \leq y_{k'+1}$, and go to Step 10.

Step 8. If $y_{\text{round}[(k+n)/2]} \leq \alpha / \beta < y_{\text{round}[(k+n)/2]+1}$, let $r_{\tilde{B}_\alpha} = \alpha / \beta$, $R = \text{round}[(k+n)/2]$, and go to Step 10.

Step 9. If $y_k \leq \alpha / \beta < y_{k+1}$, then $r_{\tilde{B}_\alpha} = \alpha / \beta$, $R = k$, and go to Step 10.

Step 10. Calculate $s = \text{sign}(k' - k)$,

$$\alpha' = \alpha - s \sum_{i=\min(k,k')+1}^{\max(k,k')} y_i [\bar{f}_{\tilde{B}_\alpha}(y_i) - \underline{f}_{\tilde{B}_\alpha}(y_i)], \text{ and } \beta' = \beta - s \sum_{i=\min(k,k')+1}^{\max(k,k')} [\bar{f}_{\tilde{B}_\alpha}(y_i) - \underline{f}_{\tilde{B}_\alpha}(y_i)].$$

Step 11. Update $k = k'$.

Step 12. Search up from k to $n - 1$, seek the k' which satisfies $y_{k'} \leq \alpha / \beta \leq y_{k'+1}$.

Step 13. While $k' \neq k$.

Step 14. Update $\alpha' = \alpha - \sum_{i=k'+1}^k y_i [\bar{f}_{\tilde{B}_\alpha}(y_i) - \underline{f}_{\tilde{B}_\alpha}(y_i)]$, $\beta' = \beta - \sum_{i=k'+1}^k [\bar{f}_{\tilde{B}_\alpha}(y_i) - \underline{f}_{\tilde{B}_\alpha}(y_i)]$, and $k = k'$.

Step 15. Search up from k to $n - 1$, seek the k' which satisfies $y_{k'} \leq \alpha / \beta \leq y_{k'+1}$.

Step 16. End while.

Step 17. Let $r_{\tilde{B}_\alpha} = \alpha / \beta$, $R = k$.

4. Simulation

Here we give four simulation examples. According to the fuzzy inference, by combining the firing fuzzy rule, let the output GT2 FS be known in advance.

In example one, the footprint of uncertainty (FOU) of primary membership function (MF) is composed of piecewise linear functions [17,18,33–35], while the secondary MF (or say vertical slice) is selected as the trapezoidal MF. In example two, for the primary MF, FOU is composed of

Gaussian functions and linear functions [36–41], while the vertical slice is selected as the trapezoidal MF. In example three, for the primary MF, whose FOU is completely composed of Gaussian functions [17,18,33–35], while the secondary MF is selected as the triangle MF. In instance four, for the primary MF, FOU is selected as the Gaussian T2 MF which has uncertain standard deviations [36–41], while the secondary MF is selected as the triangle MF.

In computer simulation experiments, the alpha is equally decomposed into: $0, 1/\Delta, \dots, 1$, where Δ represents the number of effective values. Here we let Δ be varied from one to a hundred with a step size of 1. First Table 7 and Figure 1 provide the defined FOUs for the four examples.

Let the primary variable be $y \in [0, 10]$ in Examples 1, 3 and 4. Suppose that the primary variable is $y \in [-5, 14]$ in instance two. Moreover, the Table 8 and Figure 2 provide the vertical slices of four instances.

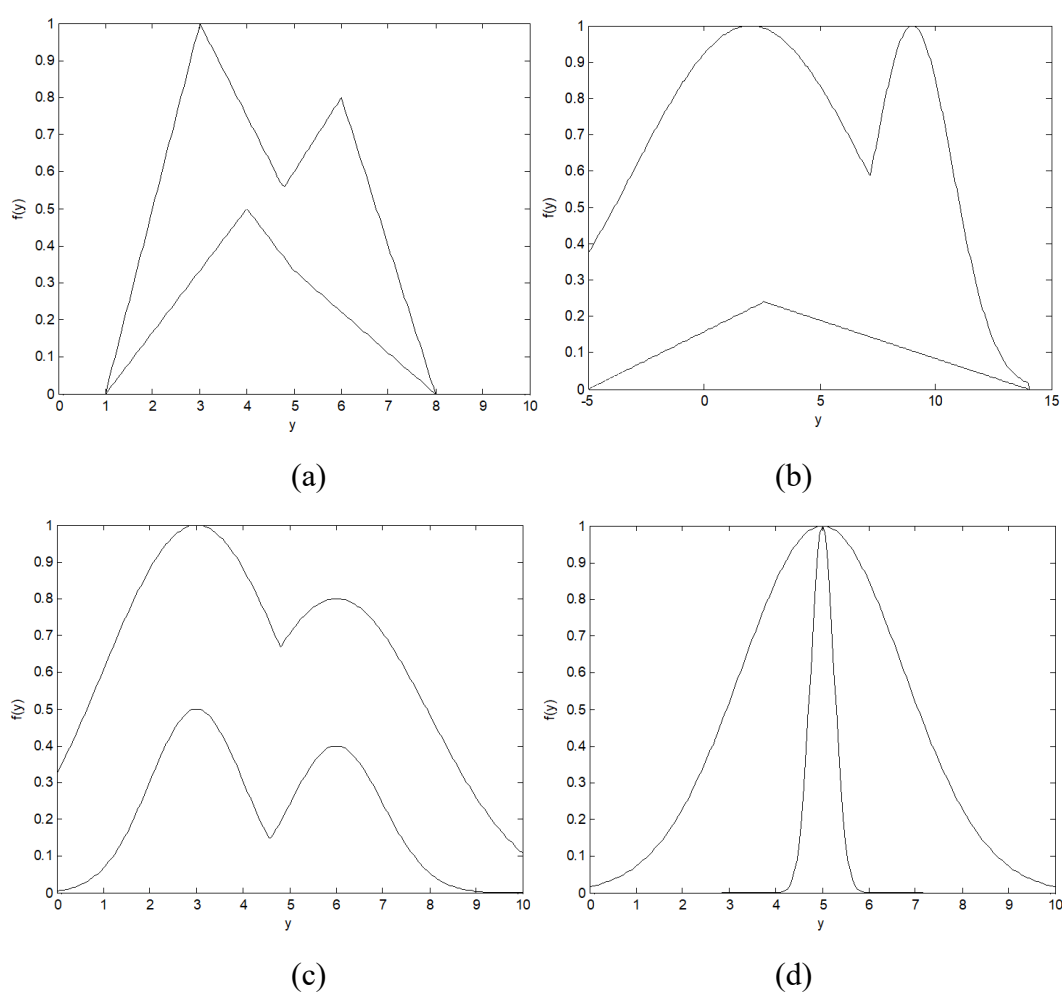


Figure 1. Graphs for FOUs; (a) instance one; (b) instance two; (c) instance three, and (d) instance four.

Table 7. Representation of FOUs [18,34–36].

| Num | Representation |
|-----|--|
| 1 | $\underline{f}_{\tilde{B}_1}(y) = \max \left\{ \begin{array}{l} \frac{y-1}{6}, \quad 1 \leq y \leq 3 \\ \frac{7-y}{6}, \quad y \in (4,7] \\ 0, \quad \text{else} \end{array} \right\}, \left[\begin{array}{l} \frac{y-3}{6}, \quad y \in [3,5] \\ \frac{8-y}{9}, \quad y \in (5,8] \\ 0, \quad \text{else} \end{array} \right\},$ |
| 2 | $\bar{f}_{\tilde{B}_1}(y) = \max \left\{ \begin{array}{l} \frac{y-1}{2}, \quad 1 \leq y \leq 3 \\ \frac{7-y}{4}, \quad y \in (3,7] \\ 0, \quad \text{else} \end{array} \right\}, \left[\begin{array}{l} \frac{y-2}{5}, \quad y \in [2,6] \\ \frac{16-2y}{5}, \quad 6 < y \leq 8 \\ 0, \quad \text{else} \end{array} \right\}.$ $\underline{f}_{\tilde{B}_2}(y) = \begin{cases} \frac{0.6(y+5)}{19}, & -5 \leq y \leq 2.6 \\ \frac{0.4(14-y)}{19}, & 2.6 < y \leq 14 \end{cases},$ $\bar{f}_{\tilde{B}_2}(y) = \begin{cases} \exp[-\frac{1}{2}(\frac{y-2}{5})^2], & y \in [-5, 7.185] \\ \exp[-\frac{1}{2}(\frac{y-9}{1.75})^2], & y \in (7.185, 14] \end{cases}.$ |
| 3 | $\underline{f}_{\tilde{B}_3}(y) = \max \left\{ 0.5 \exp[-\frac{(y-3)^2}{2}], 0.4 \exp[-\frac{(y-6)^2}{2}] \right\},$ $\bar{f}_{\tilde{B}_3}(y) = \max \left\{ \exp[-0.5 \frac{(y-3)^2}{4}], 0.8 \exp[-0.5 \frac{(y-6)^2}{4}] \right\}.$ |
| 4 | $\underline{f}_{\tilde{B}_4}(y) = \exp[-\frac{1}{2}(\frac{y-3}{0.25})^2],$ $\bar{f}_{\tilde{B}_4}(y) = \exp[-\frac{1}{2}(\frac{y-3}{1.75})^2].$ |

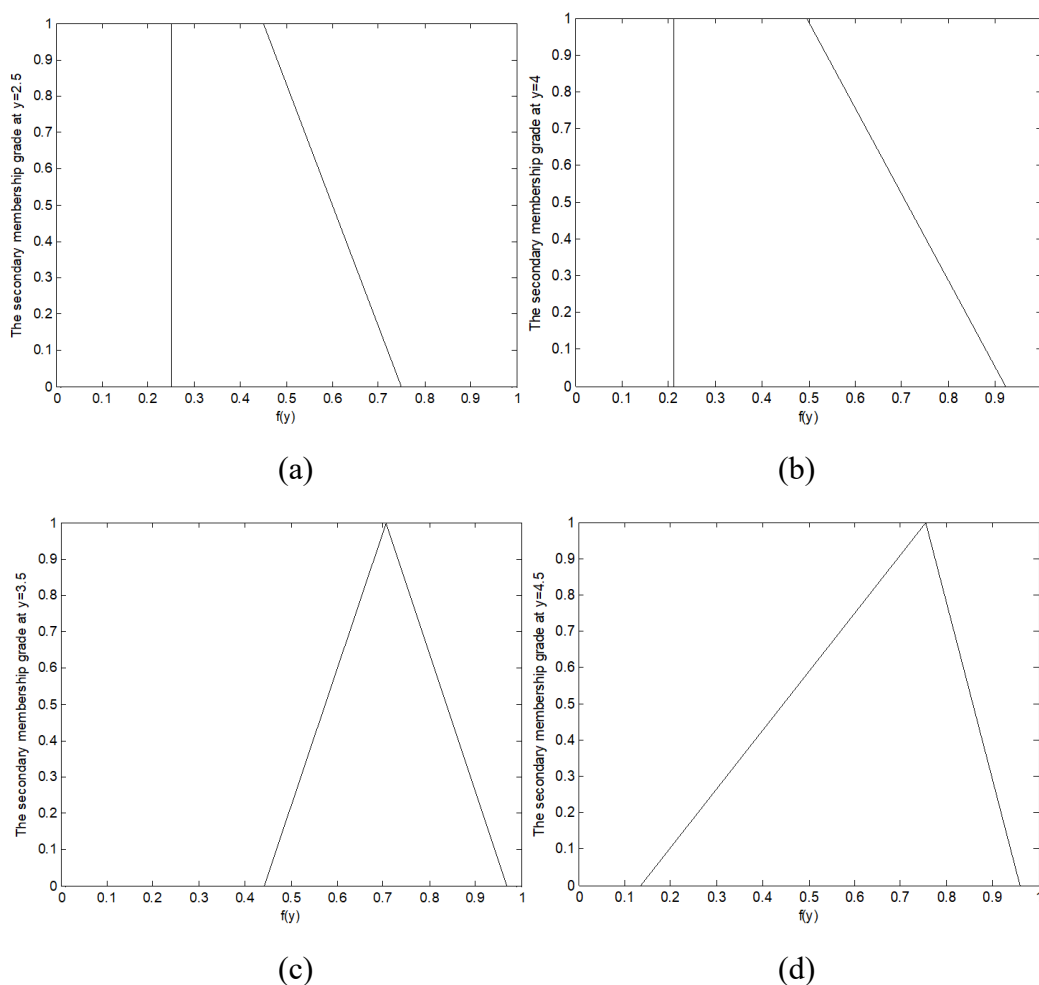


Figure 2. Graphs of secondaryMFs; (a) instance one; (b) instance two; (c) instance three, and (d) instance four.

Table 8. Representations of secondary membership functions for four examples [18,27–29].

| Num | Representation |
|-----|---|
| 1 | $L(y) = \underline{f}(y) + 0.6w[\overline{f}(y) - \underline{f}(y)],$ $R(y) = \overline{f}(y) - \frac{3}{5}(1-w)[\overline{f}(y) - \underline{f}(y)],$ <p style="text-align: center;">and w is 0</p> |
| 2 | Just as in example one |
| 3 | $Apex = \underline{f}(y) + w[\overline{f}(y) - \underline{f}(y)],$ and $w = 0.5$ |
| 4 | Just as in example three, and $w = 0.75$ |

Because the computations of $l_{\tilde{B}_\alpha}$ and $r_{\tilde{B}_\alpha}$ are quite similar, only the $l_{\tilde{B}_\alpha}$ is calculated in four simulations. Here the SBCEKM algorithms are considered as a benchmark. As the number of effective alpha-planes $\Delta = 100$, let the error accuracy $\varepsilon = 10^{-6}$, then the left centroid type-reduced

sets calculated by the benchmark are given in Figure 3.

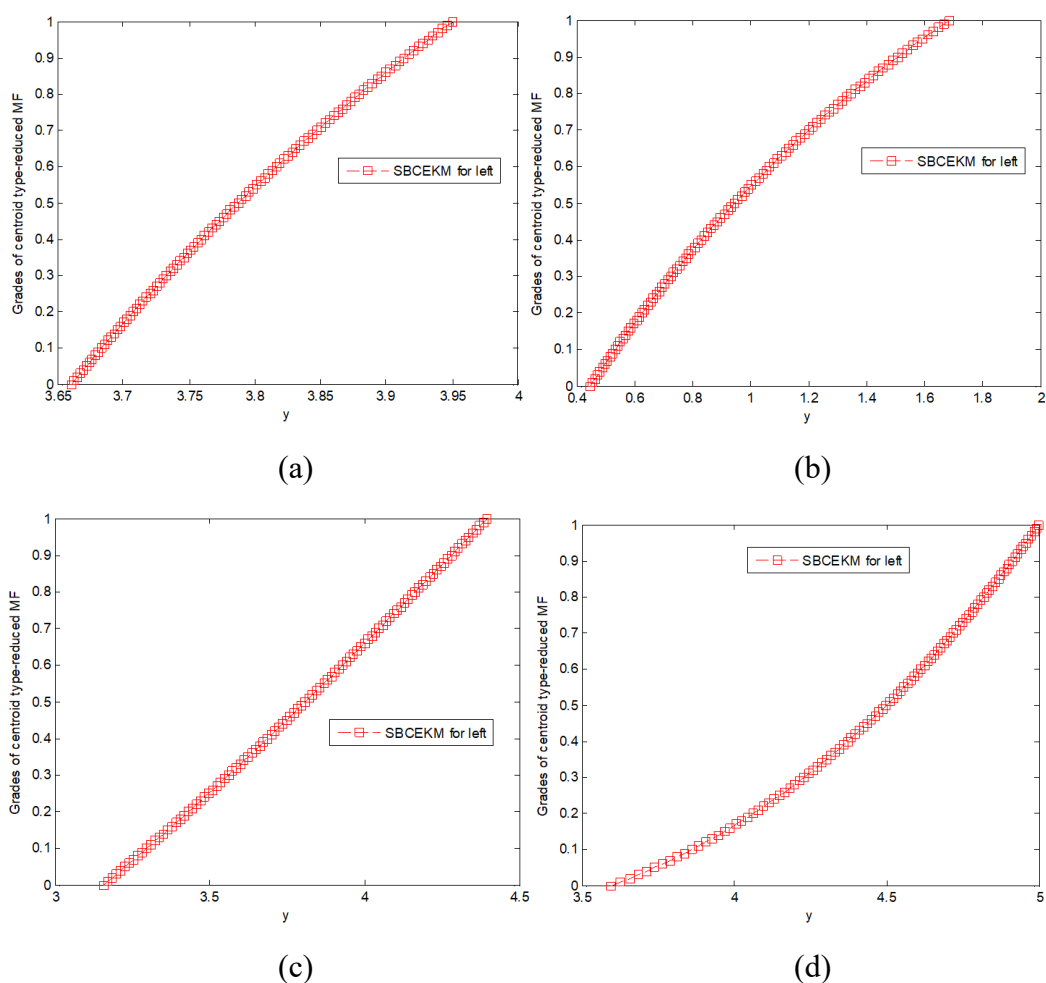


Figure 3. The left centroid type-reduced sets computed by the SBCEKM algorithms; (a) instance one; (b) instance two; (c) instance three, and (d) instance four.

Then the left centroid type-reduced sets computed by the EKM algorithms, SBEKM algorithms and proposed SBDEKM algorithms are provided in Figure 4.

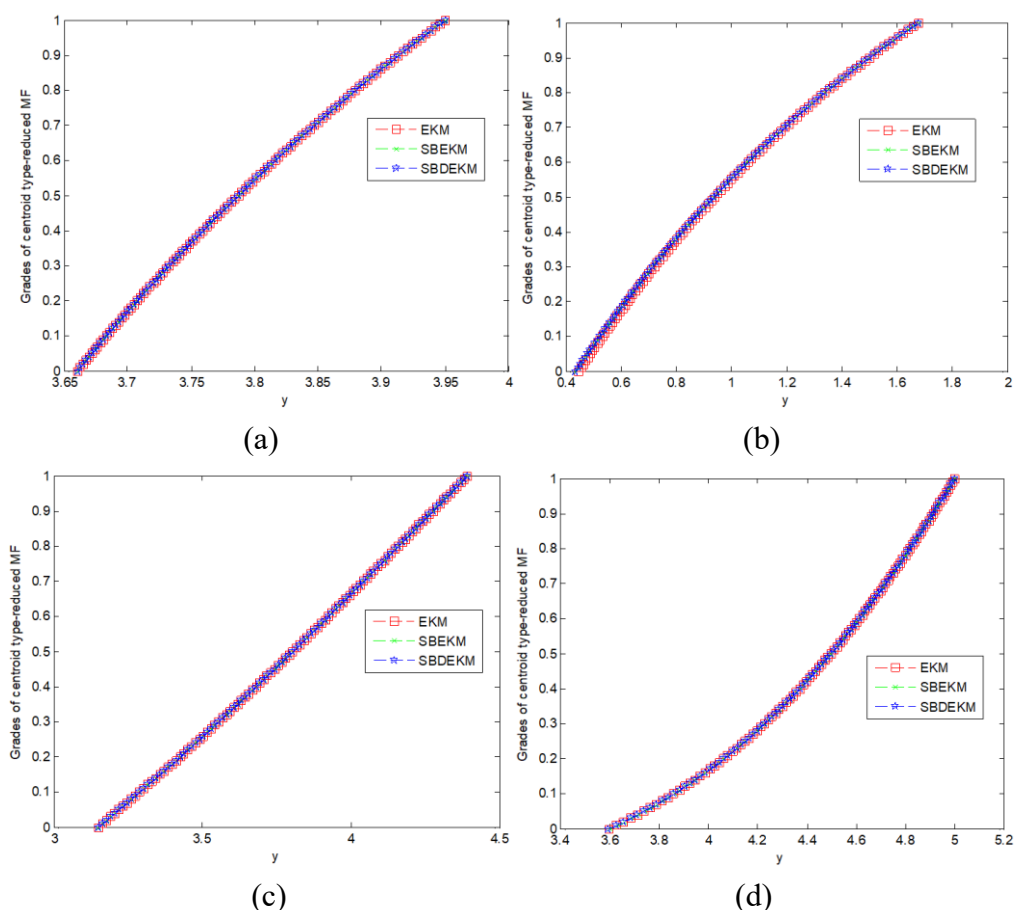


Figure 4. The left type-reduced sets computed by EKM, SBEKM, and SBDEKM algorithms; (a) instance one; (b) instance two; (c) instance three, and (d) instance four.

Furthermore, for comparing the performances of these three algorithms, the defined graphs for the absolute errors of type-reduced sets between the benchmark SBCEKM, EKM algorithms, SBEKM algorithms and the proposed SBDEKM algorithms are provided in Figure 5, in which the independent variable is the grade of type-reduced MF.

In order to measure the calculational accuracies of the EKM, SBEKM, and SBDEKM algorithms quantitatively, we define the means of relative absolute errors as: $|C_{EKM_i, SBEKM_i, SBDEKM_i} - C_{SBCEKM_i}| / |C_{SBCEKM_i}| (i=1, 2, \dots, 4)$, where C_{SBCEKM_i} is the benchmark results of SBCEKM algorithms. Table 9 provides the means of absolute errors of the EKM, SBDEKM, and SBDEKM algorithms, where the results in this table are divided by 10^{-4} .

Table 9. Means of relative absolute errors of the EKM, SBEKM, and SBDEKM algorithms ($/10^{-4}$).

| Algorithms | EKM | SBEKM | SBDEKM |
|---------------|---------|---------|---------|
| Example 1 | 0.00058 | 0.00054 | 0.00052 |
| Example 2 | 0.41340 | 0.68047 | 0.68162 |
| Example 3 | 0.11373 | 0.11440 | 0.11451 |
| Example 4 | 0.00612 | 0.01134 | 0.00685 |
| Total average | 0.13346 | 0.20169 | 0.20088 |

Observing Figures 4, 5 and Table 9, we can make the following conclusions:

For the absolute errors of type-reduced sets between the benchmark SBCEKM algorithms and the EKM, SBEKM, and SBDEKM algorithms, the results of all of them converge to certain values. In example one, the SBDEKM algorithms obtain the smallest value. In other three examples and for the total average, the EKM algorithms get the smallest value. However, as the results in Table 9 should be divided by 10^{-4} , the differences between these results are very small (see the Figure 3), and so these three kinds of discrete algorithms can get almost the same computational accuracies. If more attention was paid to computational accuracy, we should use the SBWEKM algorithms.

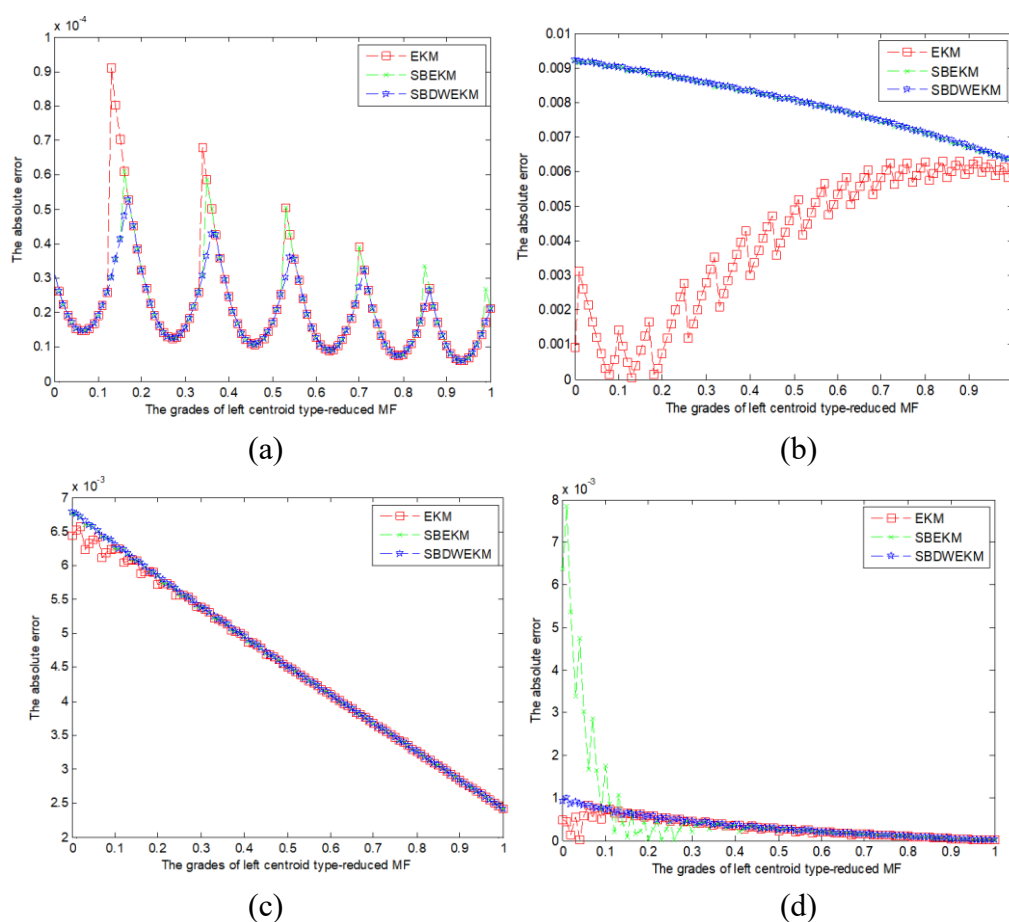


Figure 5. Absolute errors of left type-reduced sets between the SBCEKM, EKM algorithms, SBEKM algorithms and SBDEKM algorithms; (a) Example 1; (b) Example 2; (c) Example 3, and (d) Example 4.

Next, the more important computational times for these three types of algorithms are researched for potential applications. Here the unrepeatable computational times rely on the computer hardware and software environments. The simulation platform consists of a dual core CPU desktop and MATLAB 2013a. In the test, as the number of samples of primary variable is selected as $n = 50:5000$ with a stepsize of 50. The specific unrepeatable calculation times for four instances are given in Figure 6.

Generally speaking, the computational times of the three types of algorithms vary linearly in correspondence with the number of samples. Therefore, the linear regression model $t = g + hn$ can

be selected here for simulation, where t represents the calculation time, and g and h are two coefficients. Finally, Table 10 provides the regression coefficients. The calculational time variance rate is given as:

$$\frac{(\max_{i=1,\dots,4}\{t_i\} - \min_{i=1,\dots,4}\{t_i\})}{\max_{i=1,\dots,4}\{t_i\}}. \quad (33)$$

Observing Figure 6 and Table 10, it is clear that the computational times of proposed SBDEKM algorithms are much less than the EKM and SBEKM algorithms. The SBDEKM algorithms should be considered as the first, and the other two types of algorithms should be viewed as the second. As the start of the SBEKM algorithms is more complicated than the EKM algorithms, the computational times of EKM algorithms are better than SBEKM algorithms. In a word, the relation between the times of three kinds of algorithms is:

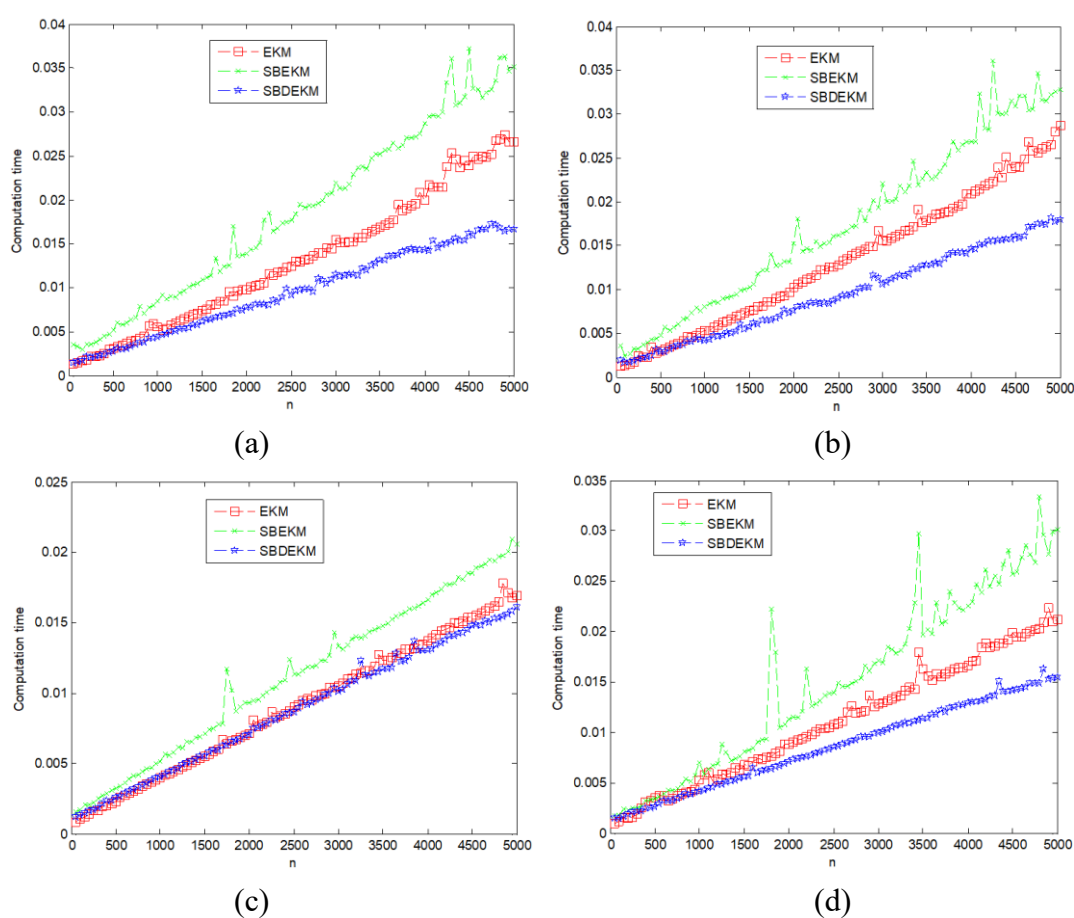


Figure 6. Computation times comparisons for EKM, SBEKM and SBDEKM algorithms; (a) instance one; (b) instance two; (c) instance three, and (d) instance four.

$SBDEKM < EKM < SBEKM$. Moreover, the computational time difference rate for these four instances is between 17.68%~71.16%.

Table 10. Regression coefficients of three types of algorithms.

| Coefficient / $10^{-4} (g, h)$ | EKM | | SBEKM | | SBDEKM | |
|--------------------------------|--------|---------|--------|---------|--------|---------|
| Example 1 | 0.0524 | -1.4608 | 0.0682 | 13.8054 | 0.0330 | 11.5248 |
| Example 2 | 0.0531 | -0.7299 | 0.0645 | 12.8895 | 0.0337 | 9.7175 |
| Example 3 | 0.0329 | 7.1359 | 0.0382 | 15.7954 | 0.0303 | 11.1144 |
| Example 4 | 0.0412 | 7.0540 | 0.0581 | 4.4558 | 0.0290 | 12.8824 |

On the basis of the above analysis, it is reasonable to see that the initialization and searching spaces have an effect on completing the centroid TR. Therefore, we should adopt the SBDEKM algorithms for studying the TR. If we only consider the computational accuracies, these three types of algorithms almost have the same performances. However, if we consider both the calculational accuracies and times comprehensively, it is suggested that one uses the SBDEKM algorithms for performing the centroid TR with piece-wise defined primary MF and trapezoidal secondary MF as in instance 1, Gaussian and linear primary MFs and trapezoidal vertical slice as in Example 2, Gaussian primary MF and triangular secondary MF as in Example 3, and uncertain derivation primary MF and triangular secondary MF as in Example 4.

5. Conclusions

This paper explains the sensible beginning and divides the searching space for EKM algorithms, and then the SBDEKM algorithms are put forward to complete the centroid TR. Compared with both the EKM and SBEKM algorithms, the SBDEKM algorithms can significantly improve the efficiencies of computations.

Next, we will study the SBWEKM algorithms for finishing the TR of IT2 and GT2 FLSs. Both the computational accuracy and efficiency should be enhanced. Furthermore, the center-of-sets (COS) TR [42,43] of IT2 FLSs and GT2 FLSs should continue to be investigated, and designing and applying T2 FLSs optimized with the hybrid application of different kinds of optimization algorithms [9,10, 40–49] will be further focused on.

Use of AI tools declaration

The authors declare that they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The paper is supported by the National Natural Science Foundation of China (61973146, 61773188), and the Education Department of Liaoning Province (LJKQZ2021143).

The author is thankful to the Professor Mendel, who has provided some important advices.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. J. M. Mendel, R. I. John, F. L. Liu, Interval type-2 fuzzy logic systems made simple, *IEEE T. Fuzzy Sys.*, **14** (2006), 808–821. <https://doi.org/10.1109/TFUZZ.2006.879986>
2. D. R. Wu, J. M. Mendel, Uncertainty measures for interval type-2 fuzzy sets, *Inform. Sci.*, **177** (2007), 5378–5393. <https://doi.org/10.1016/j.ins.2007.07.012>
3. A. Khosravi, S. Nahavandi, Load forecasting using interval type-2 fuzzy logic systems: Optimal type reduction, *IEEE T. Indust. Inform.*, **10** (2014), 1055–1063. <https://doi.org/10.1109/TII.2013.2285650>
4. J. M. Mendel, F. L. Liu, D. Y. Zhai, Alpha-plane representation for type-2 fuzzy sets: Theory and applications, *IEEE T. Fuzzy Syst.*, **17** (2009), 1189–1207. <https://doi.org/10.1109/TFUZZ.2009.2024411>
5. C. Wagner, H. Hagra, Toward general type-2 fuzzy logic systems based on zSlices, *IEEE T. Fuzzy Syst.*, **18** (2010), 637–660. <https://doi.org/10.1109/TFUZZ.2010.2045386>
6. F. L. Liu, An efficient centroid type-reduction strategy for general type-2 fuzzy logic system, *Inf. Sci.*, **178** (2008), 2224–2236. <https://doi.org/10.1016/j.ins.2007.11.014>
7. 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>
8. C. I. Gonzalez, P. Melin, J. R. Castro, O. Mendoza, O. Castillo, An improved sobel edge detection method based on generalized type-2 fuzzy logic, *Soft Comput.*, **20** (2016), 773–784. <https://doi.org/10.1007/s00500-014-1541-0>
9. P. Melin, C. I. Gonzalez, J. R. Castro, O. Mendoza, O. Castillo, Edge-detection method for image processing based on generalized type-2 fuzzy logic, *IEEE T. Fuzzy Syst.*, **22** (2014), 1515–1525. <https://doi.org/10.1109/TFUZZ.2013.2297159>
10. M. A. Sanchez, O. Castillo, 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 Syst. Appl.*, **42** (2015), 5904–5914. <https://doi.org/10.1016/j.eswa.2015.03.024>
11. 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>
12. Y. Chen, D. Z. Wang, W. Ning, Forecasting by TSK general type-2 fuzzy logic systems optimized with genetic algorithms, *Opt. Contr. Appl. Meth.*, **39** (2018), 393–409. <https://doi.org/10.1002/oca.2353>
13. Y. Chen, C. X. Li, J. X. Yang, Design and application of Nagar-Bardini structure-based interval type-2 fuzzy logic systems optimized with the combination of backpropagation algorithms and recursive least square algorithms, *Expert Syst. Appl.*, **211** (2023), 1–10. <https://doi.org/10.1016/J.ESWA.2022.118596>
14. Y. Chen, D. Z. Wang, Forecasting by general type-2 fuzzy logic systems optimized with QPSO algorithms, *Int. J. Contr. Auto. Syst.*, **15** (2017), 2950–2958. <https://doi.org/10.1007/s12555-017-0793-0>
15. L. X. Wang, A new look at type-2 fuzzy sets and type-2 fuzzy logic systems, *IEEE T. Fuzzy Syst.*, **25** (2017), 693–706. <https://doi.org/10.1109/TFUZZ.2016.2543746>

16. 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>
17. 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>
18. 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>
19. X. L. Liu, S. P. Wan, Combinatorial iterative algorithms for computing the centroid of an interval type-2 fuzzy set, *IEEE T. Fuzzy Syst.*, **28** (2020), 607–617. <https://doi.org/10.1109/TFUZZ.2019.2911918>
20. 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>
21. C. Chen, R. John, J. Twycross, J. M. Garibaldi, A direct approach for determining the switch points in the Karnik-Mendel algorithm, *IEEE T. Fuzzy Syst.*, **26** (2018), 1079–1085. <https://doi.org/10.1109/TFUZZ.2017.2699168>
22. Z. Zhang, X. Zhao, Y. Qin, H. Si, L. Zhou, Interval type-2 fuzzy TOPSIS approach with utility theory for subway station operational risk evaluation, *J. Ambient Intel. Human. comput.*, **13** (2022), 4849–4863. <https://doi.org/10.1007/s12652-021-03182-0>
23. X. L. Liu, Y. C. Lin, New efficient algorithms for the centroid of an interval type-2 fuzzy set, *Inform. Sci.*, **570** (2021), 1–19. <https://doi.org/10.1016/j.ins.2021.04.032>
24. 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>
25. J. H. Wang, W. Ji, X. K. Fang, S. S. Gu, Improvement of enhanced Karnik-Mendel algorithm for interval type-2 fuzzy sets, *Contr. Deci.*, **28** (2013), 1165–1172. <https://doi.org/10.13195/j.kzyjc.2013.08.002>
26. 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>
27. Y. Chen, D. Z. Wang, Forecasting by designing Mamdani general type-2 fuzzy logic systems optimized with quantum particle swarm optimization algorithms, *Trans. Inst. Meas. Contr.*, **41** (2019), 2886–2896. <https://doi.org/10.1177/0142331218816753>
28. 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>
29. Y. Chen, J. X. Yang, C. X. Li, Design of Takagi Sugeno Kang type interval type-2 fuzzy logic systems optimized with hybrid algorithms, *Int. J. Fuzzy Syst.*, **25** (2023), 868–879. <https://doi.org/10.1007/S40815-022-01410-Z>
30. Y. Chen, Study on non-iterative algorithms for center-of-sets type-reduction of Takagi-Sugeno-Kang type general type-2 fuzzy logic systems, *Compl. Intell. Syst.*, **9** (2023), 4015–4023. <https://doi.org/10.1007/s40747-022-00927-y>

31. Y. Chen, Study on centroid type-reduction of general type-2 fuzzy logic systems with sensible beginning weighted enhanced Karnik-Mendel algorithms, *Soft Comput.*, **27** (2023), 9261–9279. <https://doi.org/10.1007/S00500-023-08269-8>
32. Y. Chen, J. X. Yang, C. X. Li, Design of reasonable initialization weighted enhanced Karnik-Mendel algorithms for centroid type-reduction of interval type-2 fuzzy logic systems, *AIMS Math.*, **7** (2022), 9846–9870. <https://doi.org/10.3934/math.2022549>
33. 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>
34. 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>
35. Y. Chen, Study on centroid type-reduction of general type-2 fuzzy logic systems with sensible beginning weighted enhanced Karnik-Mendel algorithms, *Soft Comput.*, **27** (2023), 9261–9279. <https://doi.org/10.1007/S00500-023-08269-8>
36. 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>
37. 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>
38. 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>
39. S. Greenfield, F. Chiclana, S. Coupland, R. John, The collapsing method of defuzzification for discretised interval type-2 fuzzy sets, *Inform. Sci.*, **179** (2009), 2055–2069. <https://doi.org/10.1016/j.ins.2008.07.011>
40. 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>
41. 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>
42. 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>
43. Y. Chen, C. X. Li, J. X. Yang, Design of discrete noniterative algorithms for center-of-sets type reduction of general type-2 fuzzy logic systems, *Int. J. Fuzzy Syst.*, **24** (2022), 2024–2035. <https://doi.org/10.1007/S40815-022-01256-5>
44. 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>

45. 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>
46. F. Y. Wang, H. Mo, Some fundamental issues on type-2 fuzzy sets, *Acta Auto. Sin.*, **43** (2017), 1114–1141. <https://doi.org/10.16383/j.aas.2017.c160638>
47. 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>
48. N. Mansoureh, F. Z. M. Hossein, B. Susan, A multilayer general type-2 fuzzy community detection model in large-scale social networks, *IEEE T. Fuzzy Syst.*, **30** (2022), 4494–4503. <https://doi.org/10.1109/TFUZZ.2022.3153745>
49. E. Ontiveros, P. Melin, O. Castillo, Comparative study of interval type-2 and general type-2 fuzzy systems in medical diagnosis, *Inform. Sci.*, **525** (2020), 37–53. <https://doi.org/10.1016/j.ins.2020.03.059>



AIMS Press

© 2024 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)