International Journal of Innovative
Computing, Information and Control
Volume **15**, Number **5**, October **2019**

 ISSN 1349-4198
pp. **1673–1683**

STUDY ON IMPROVED ENHANCED KARNIK-MENDEL ALGORITHMS FOR CENTROID TYPE-REDUCTION OF GENERAL TYPE-2 FUZZY LOGIC SYSTEMS

Yang Chen

College of Science
Liaoning University of Technology
No. 169, Shiying Street, Guta District, Jinzhou 121001, P. R. China
lxychenyang@lnut.edu.cn

Received January 2019; revised May 2019

Abstract. *In recent years, general type-2 fuzzy logic systems (GT2 FLSs) have become a hot topic in current academic research. The block of type-reduction (TR) under the guidance of inference plays the central role for T2 FLSs. In early times, the TR algorithms are developed resort to calculating the centroid of interval type-2 fuzzy sets (IT2 FSs). Generally speaking, the computational intensive enhanced Karnik-Mendel (EKM) algorithms are the standard way to perform the centroid TR of interval type-2 fuzzy logic systems (IT2 FLSs). Based on the α-planes representation theory of general type-2 fuzzy sets (GT2 FSs), this paper introduces the improved EKM (IEKM) algorithms to perform the centroid TR of GT2 FLSs. Two computer simulation examples are used to illustrate and analyze the performances of IEKM algorithms. Compared with the most commonly used EKM algorithms, the IEKM algorithms have faster computation speed without loosing calculation accuracy, which provides the potential value for designing and applying T2 FLSs.*

Keywords: General type-2 fuzzy logic systems, Alpha-planes, Enhanced Karnik-Mendel algorithms, Improved EKM algorithms, Computer simulation

1. **Introduction.** As we all know, the membership grades of T2 FSs are themselves T1 FSs. As the design degrees of freedom increases, T2 FSs have the potential to outperform their T1 counterparts on modeling and dealing with uncertainties. T2 FLSs can be considered as a type of expert systems based on fuzzy rules. IT2 FLSs have been successfully applied to many areas with high uncertainty and nonlinearity like power systems [1,2], financial systems [3,4], permanent magnetic drive [5,6], autonomous mobile robots [7], medical systems [8], pattern recognition systems [9], and database and information systems [10]. GT2 FSs can be considered as higher order uncertainty models than IT2 FSs. Therefore, it is difficult to design and apply the computational complexity GT2 FLSs. Until recent years, studying on GT2 FLSs [11,12] has become a hot topic as the development of α-planes or z-slices representation of GT2 FSs. GT2 FLSs [13-17] have been gradually applied to some areas.

In general, T2 FLSs are composed of five blocks as fuzzifier, rules, inference [18], type-reduction and defuzzification. Among them, the type-reduction (TR) under the guidance of inference is a central block, which has the function of transforming T2 FS to T1 FS. Naturally, GT2 FLSs use GT2 FSs. As the secondary membership grades of GT2 FSs lie between 0 and 1, GT2 FSs can measure the uncertainties of membership functions (MFs)

DOI: 10.24507/ijicic.15.05.1673

uniformly. Recent studies show that GT2 FLSs [13,15,17,19] outperform their T1 and IT2 counterparts on some fields.

The traditional and popular Karnik-Mendel (KM) algorithms [20,21] were developed based on calculating the centroids of IT2 FSs. Then the monotonicity and super convergence properties [22] of KM algorithms were proved by Mendel and Liu. Because the KM algorithms need two to six iterations to stop, Wu and Mendel developed the enhanced version of KM (EKM) algorithms [23] to reduce the computational cost. The simulation results show that the EKM algorithms can save about two iterations compared with the KM algorithms. However, for the EKM algorithms, there still exist defects for seeking the switching points. In order to improve the computation efficiency, Wang et al. proposed the improved EKM (IEKM) algorithms [24] for solving the centroid of IT2 FSs. These TR algorithms of IT2 FSs or IT2 FLSs have laid some foundations for studying the TR of GT2 FLSs.

Based on the α-planes representation of GT2 FSs, this paper extends the IEKM algorithms to perform the centroid TR of GT2 FLSs. Furthermore, the blocks of inference, TR and defuzzification are also discussed. Two simulation examples are provided to illustrate and analyze the performances of IEKM algorithms compared with the EKM algorithms. As for computing the centroid left endpoints of GT2 FLSs, the results show that the IEKM algorithms are computationally faster without loosing the calculation accuracy.

The rest of the paper is organized as follows. Section 2 briefly introduces the GT2 FLSs. Section 3 gives the IEKM algorithms for performing the centroid TR of GT2 FLSs. Section 4 provides two simulation examples, compares and analyzes the performances of IEKM algorithms and EKM algorithms. Finally, Section 5 gives the conclusions.

2. GT2 FLSs.

Generally speaking, GT2 FLSs can be divided into two types from the aspect of structure. And they are Mamdani type [4,11,25] and Takagi Sugeno Kang (TSK) type [2,6,13,15]. Here we only focus on the Mamdani type. Without loss of generality, suppose that a Mamdani GT2 FLS has n inputs $x_1 \in X_1, \ldots, x_n \in X_n$, and one output $y \in Y$, the system can be characterized by N fuzzy rules, where the sth rule is of the form:

$$\text{If } x_1 \text{ is } \widetilde{F}_1^s \text{ and } \ldots \text{ and } x_n \text{ is } \widetilde{F}_n^s, \text{ then } y \text{ is } \widetilde{G}^s \ (s = 1, \ldots, N) \tag{1}$$

The process of inference [18,26] is as follows.

In order to simplify the expressions, here we adopt the singleton fuzzifier [11], i.e., as $x_i = x_i'$, only the vertical slice [12,27,28] (or secondary MF) $\widetilde{F}_i^s(x_i')$ of GT2 FS $\widetilde{F}_i^s$ is activated, and its α-cut decomposition is as:

$$\widetilde{F}_i^s(x_i') = \sup_{\forall \alpha \in [0,1]} \alpha / \left[a_{i,\alpha}^s(x_i'), b_{i,\alpha}^s(x_i') \right] \tag{2}$$

For each fuzzy rule, we first calculate the firing interval at the α-level as:

$$F_\alpha : \begin{cases} F_\alpha^s(x') \equiv \left[\underline{f}_\alpha^s(x'), \overline{f}_\alpha^s(x') \right] \\ \underline{f}_\alpha^s(x') \equiv T_{i=1}^n a_{i,\alpha}^s(x_i') \\ \overline{f}_\alpha^s(x') \equiv T_{i=1}^n b_{i,\alpha}^s(x_i') \end{cases} \tag{3}$$

where T represents the product or minimum operation.

At the corresponding α-level, let the α-plane (or horizontal slice) of consequent GT2 FS $\widetilde{G}^s$ be $\widetilde{G}_\alpha^s$, i.e.,

$$\widetilde{G}_\alpha^s = \int_{\forall y \in Y} \widetilde{G}_\alpha^s(y)/y = \int_{\forall y \in Y} \left[g_{L,\alpha}^s(y), g_{R,\alpha}^s(y) \right]/y \tag{4}$$

Then merge the firing interval of each fuzzy rule with the related α-plane $\widetilde{G}_\alpha^s$ of consequent to obtain the firing rule α-plane $\widetilde{B}_\alpha^s$, i.e.,

$$\widetilde{B}_\alpha^s : \begin{cases} FOU\left(\widetilde{B}_\alpha^s\right) = \left[\underline{\mu}_{\widetilde{B}_\alpha^s}(y|x'), \overline{\mu}_{\widetilde{B}_\alpha^s}(y|x')\right] \\ \underline{\mu}_{\widetilde{B}_\alpha^s}(y|x') = \underline{f}_\alpha^s(x') * g_{L,\alpha}^s(y) \\ \overline{\mu}_{\widetilde{B}_\alpha^s}(y|x') = \overline{f}_\alpha^s(x') * g_{R,\alpha}^s(y) \end{cases} \tag{5}$$

Next we aggregate all the $\widetilde{B}_\alpha^s$ $(s = 1, \ldots, N)$ to get the output α-plane $\widetilde{B}_\alpha$, i.e.,

$$\widetilde{B}_\alpha : \begin{cases} FOU\left(\widetilde{B}_\alpha\right) = \left[\underline{\mu}_{\widetilde{B}_\alpha}(y|x'), \overline{\mu}_{\widetilde{B}_\alpha}(y|x')\right] \\ \underline{\mu}_{\widetilde{B}_\alpha}(y|x') = \underline{\mu}_{\widetilde{B}_\alpha^1}(y|x') \vee \cdots \vee \underline{\mu}_{\widetilde{B}_\alpha^N}(y|x') \\ \overline{\mu}_{\widetilde{B}_\alpha}(y|x') = \overline{\mu}_{\widetilde{B}_\alpha^1}(y|x') \vee \cdots \vee \overline{\mu}_{\widetilde{B}_\alpha^N}(y|x') \end{cases} \tag{6}$$

In order to obtain the type-reduced set $Y_{C,\alpha}(x')$ at the α-level, we compute the centroid [11,12,29] for $\widetilde{B}_\alpha$, i.e.,

$$Y_{C,\alpha}(x') = C_{\widetilde{B}_\alpha}(x') = \alpha / \left[l_{\widetilde{B}_\alpha}(x'), r_{\widetilde{B}_\alpha}(x')\right] \tag{7}$$

$$l_{\widetilde{B}_\alpha}(x') = \min_{\mu_{\widetilde{B}_\alpha}(y_i) \in \left[\underline{\mu}_{\widetilde{B}_\alpha}(y_i), \overline{\mu}_{\widetilde{B}_\alpha}(y_i)\right]} \frac{\sum_{i=1}^M y_i \mu_{R_{\widetilde{B}_\alpha}}(y_i)}{\sum_{i=1}^M \mu_{R_{\widetilde{B}_\alpha}}(y_i)} \tag{8}$$

$$r_{\widetilde{B}_\alpha}(x') = \max_{\mu_{\widetilde{B}_\alpha}(y_i) \in \left[\underline{\mu}_{\widetilde{B}_\alpha}(y_i), \overline{\mu}_{\widetilde{B}_\alpha}(y_i)\right]} \frac{\sum_{i=1}^M y_i \mu_{R_{\widetilde{B}_\alpha}}(y_i)}{\sum_{i=1}^M \mu_{R_{\widetilde{B}_\alpha}}(y_i)} \tag{9}$$

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

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

In the practical computations, suppose that the number α-planes be m, then the value of α can be uniformly decomposed to $\alpha_1, \alpha_2, \ldots, \alpha_m$, and the output of GT2 FLSs is as:

$$y(x') = \sum_{i=1}^m \alpha_i \left[\left(l_{\widetilde{B}_{\alpha_i}}(x') + r_{\widetilde{B}_{\alpha_i}}(x')\right) \Big/ 2\right] \Bigg/ \sum_{i=1}^m \alpha_i \tag{11}$$

Equation (11) was first proposed by Wagner and Hagras [12], which can be called as the average of endpoints defuzzification method. Then we extend the IEKM algorithms to perform the TR of GT2 FLSs in the next section.

3. **IEKM Algorithms.** In order to give the IEKM algorithms, we first introduce the EKM algorithms [23,29]. For the IT2 $R_{\widetilde{B}_\alpha}$ at the α-level, here $R_{\widetilde{B}_\alpha} = \alpha / \widetilde{B}_\alpha$, the two endpoints of centroid interval $l_{\widetilde{B}_\alpha}$ and $r_{\widetilde{B}_\alpha}$ can be computed as:

$$l_{\widetilde{B}_\alpha} = \frac{\sum_{i=1}^k y_i \overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) + \sum_{i=k+1}^M y_i \underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i)}{\sum_{i=1}^k \overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) + \sum_{i=k+1}^M \underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i)} \tag{12}$$

$$r_{\widetilde{B}_\alpha} = \frac{\sum_{i=1}^k y_i \underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) + \sum_{i=k+1}^M y_i \overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i)}{\sum_{i=1}^k \underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) + \sum_{i=k+1}^M \overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i)} \tag{13}$$

The EKM algorithms improve the KM algorithms in three ways: 1) a better initialization is adopted; 2) an unnecessary iteration is cancelled by altering the termination condition of the iterations; 3) according to a subtle calculation technique, the computational cost of each algorithm's iterations is reduced.

However, the EKM algorithms always search for the points k' start from 1 up from the bottom. This makes the computation times comparatively long. In addition, the initialization of EKM algorithms is still not perfect. For the initialization of EKM algorithms, [M/2.4] denotes the largest integer that is not greater than M/2.4, but this is not in accordance with the integer that is closest to M/2.4 as stated in EKM algorithms. Therefore, the IEKM algorithms improve both the initialization condition and the approach of seeking for the switch points. They make use of the properties of IT2 FSs [24], and realize the aim of searching both upward and downward to reduce the computation comparing times.

Here we expand the IEKM algorithms to perform the centroid TR of GT2 FLSs based on the α-planes representation of GT2 FS. The IEKM algorithms change the initialization of left and right endpoints of EKM algorithms in such ways:

$$k = Round(M/2.4), \quad k = Round(M/1.7) \tag{14}$$

where *Round* denotes the round up or down operation, and this will guarantee the initial value to be the only one.

Property 3.1. *For a positive integer k, if $y_k > l_{\widetilde{B}_\alpha}(k)$, then $k > L$.*

Property 3.2. *For a positive integer k, if $y_{k+1} \leq l_{\widetilde{B}_\alpha}(k)$, then $k < L$.*

Property 3.3. *For a positive integer k, if $y_k > r_{\widetilde{B}_\alpha}(k)$, then $k > R$.*

Property 3.4. *For a positive integer k, if $y_{k+1} \leq r_{\widetilde{B}_\alpha}(k)$, then $k < R$.*

All above four properties have been proved in [24].

Next we give the explanations for the IEKM algorithms.

1) For the left endpoint $l_{\widetilde{B}_\alpha}$, we first set the initial value k. In order to reduce the computation time, comparing the current $l_{\widetilde{B}_\alpha}$ with y_k and y_{k+1} while searching for the k'. Therefore, there exist three conditions as: ① $y_k > l_{\widetilde{B}_\alpha}(k)$; ② $y_{k+1} \leq l_{\widetilde{B}_\alpha}(k)$; ③ $y_k \leq l_{\widetilde{B}_\alpha}(k) < y_{k+1}$. Then we discuss as follows.

① When $y_k > l_{\widetilde{B}_\alpha}(k)$, according to Property 3.1, $k > L$. Then we judge the relations between the current value $l_{\widetilde{B}_\alpha}(k)$ and $y_{Round(k/2)}$ and $y_{Round(k/2)+1}$. There may be three conditions as: i) $y_{Round(k/2)} > l_{\widetilde{B}_\alpha}(k)$; ii) $y_{Round(k/2)+1} \leq l_{\widetilde{B}_\alpha}(k)$; and iii) $y_{Round(k/2)} \leq l_{\widetilde{B}_\alpha}(k) < y_{Round(k/2)+1}$.

i) When $y_{Round(k/2)} > l_{\widetilde{B}_\alpha}(k)$, the k' that satisfies $y_{k'} \leq l_{\widetilde{B}_\alpha}(k) < y_{k'+1}$ is searching downward from $Round(k/2) - 1$ to 1.

ii) When $y_{Round(k/2)+1} \leq l_{\widetilde{B}_\alpha}(k)$, the k' that satisfies $y_{k'} \leq l_{\widetilde{B}_\alpha}(k) < y_{k'+1}$ is searching upward from $Round(k/2) + 1$ to $k - 1$.

iii) When $y_{Round(k/2)} \leq l_{\widetilde{B}_\alpha}(k) < y_{Round(k/2)+1}$, $L = Round(k/2)$.

② When $y_{k+1} \leq l_{\widetilde{B}_\alpha}(k)$, according to Property 3.2, $k < L$. Then we judge the relations between the current value $l_{\widetilde{B}_\alpha}(k)$ and $y_{Round((k+M)/2)}$ and $x_{Round((k+M)/2)+1}$. There may be three conditions as: i) $y_{Round((k+M)/2)} > l_{\widetilde{B}_\alpha}(k)$; ii) $y_{Round((k+M)/2)+1} \leq l_{\widetilde{B}_\alpha}(k)$; and iii) $y_{Round((k+M)/2)} \leq l_{\widetilde{B}_\alpha}(k) < y_{Round((k+M)/2)+1}$.

i) When $y_{Round((k+M)/2)} > l_{\widetilde{B}_\alpha}(k)$, the k' that satisfies $y_{k'} \leq l_{\widetilde{B}_\alpha}(k) < y_{k'+1}$ is searching downward from $Round((k + M)/2) - 1$ to $k + 1$.

ii) When $y_{Round((k+M)/2)+1} \leq l_{\widetilde{B}_\alpha}(k)$, the k' that satisfies $y_{k'} \leq l_{\widetilde{B}_\alpha}(k) < y_{k'+1}$ is searching upward from $Round((k + M)/2) + 1$ to $M - 1$.

iii) When $y_{Round((k+M)/2)} \leq l_{\widetilde{B}_\alpha}(k) < y_{Round((k+M)/2)+1}$, $L = Round((k + M)/2)$.

③ When $y_k \leq l_{\widetilde{B}_\alpha}(k) < y_{k+1}$, $L = k$.

2) For the right endpoint $r_{\widetilde{B}_\alpha}$, we first set the initial value k. Compare the current value of $r_{\widetilde{B}_\alpha}(k)$ with y_k and y_{k+1} while searching for the k'. Therefore, there exist three conditions as: ① $y_k > r_{\widetilde{B}_\alpha}(k)$; ② $y_{k+1} \le r_{\widetilde{B}_\alpha}(k)$; and ③ $y_k \le r_{\widetilde{B}_\alpha}(k) < y_{k+1}$.

① When $y_k > r_{\widetilde{B}_\alpha}(k)$, according to Property 3.3, $k > R$. Then we judge the relations between the current value $r_{\widetilde{B}_\alpha}(k)$ and $y_{Round(k/2)}$ and $y_{Round(k/2)+1}$. There may be three conditions as: i) $y_{Round(k/2)} > r_{\widetilde{B}_\alpha}(k)$; ii) $y_{Round(k/2)+1} \le r_{\widetilde{B}_\alpha}(k)$; and iii) $y_{Round(k/2)} \le r_{\widetilde{B}_\alpha}(k) < y_{Round(k/2)+1}$.

i) When $y_{Round(k/2)} > r_{\widetilde{B}_\alpha}(k)$, the k' that satisfies $y_{k'} \le r_{\widetilde{B}_\alpha}(k) < y_{k'+1}$ is searching downward from $Round(k/2) - 1$ to 1.

ii) When $y_{Round(k/2)+1} \le r_{\widetilde{B}_\alpha}(k)$, the k' that satisfies $y_{k'} \le r_{\widetilde{B}_\alpha}(k) < y_{k'+1}$ is searching upward from $Round(k/2) + 1$ to $k - 1$.

iii) When $y_{Round(k/2)} \le r_{\widetilde{B}_\alpha}(k) < y_{Round(k/2)+1}$, $R = Round(k/2)$.

② When $y_{k+1} \le r_{\widetilde{B}_\alpha}(k)$, according to Property 3.4, $k < R$. Then we judge the relations between the current value $r_{\widetilde{B}_\alpha}(k)$ and $y_{Round((k+M)/2)}$ and $y_{Round((k+M)/2)+1}$. There may be three conditions as: i) $y_{Round((k+M)/2)} > r_{\widetilde{B}_\alpha}(k)$; ii) $y_{Round((k+M)/2)+1} \le r_{\widetilde{B}_\alpha}(k)$; and iii) $y_{Round((k+M)/2)} \le r_{\widetilde{B}_\alpha}(k) < y_{Round((k+M)/2)+1}$.

i) When $y_{Round((k+M)/2)} > r_{\widetilde{B}_\alpha}(k)$, the k' that satisfies $y_{k'} \le r_{\widetilde{B}_\alpha}(k) < y_{k'+1}$ is searching downward from $Round((k + M)/2) - 1$ to $k + 1$.

ii) When $y_{Round((k+M)/2)+1} \le r_{\widetilde{B}_\alpha}(k)$, the k' that satisfies $y_{k'} \le r_{\widetilde{B}_\alpha}(k) < y_{k'+1}$ is searching upward from $Round((k + M)/2) + 1$ to $M - 1$.

iii) When $y_{Round((k+M)/2)} \le r_{\widetilde{B}_\alpha}(k) < y_{Round((k+M)/2)+1}$, $R = Round((k + M)/2)$.

③ When $y_k \le r_{\widetilde{B}_\alpha}(k) < y_{k+1}$, $R = k$.

The specific steps for the IEKM algorithms to compute the centroid endpoints of an IT2 FS are as follows.

IEKM algorithms to compute $l_{\widetilde{B}_\alpha}$:

1) Initialize $k = Round(M/2.4)$, compute $a = \sum_{i=1}^{k} y_i \overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) + \sum_{i=k+1}^{M} y_i \underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i)$, $b = \sum_{i=1}^{k} \overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) + \sum_{i=k+1}^{M} \underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i)$, $c' = a/b$.

2) i) When $y_k > a/b$ and $y_{Round(k/2)} > a/b$, find $k' \in [1, Round(k/2) - 1]$ satisfies $y_{k'} \le c' < y_{k'+1}$; ii) when $y_k > a/b$ and $y_{Round(k/2)+1} \le a/b$, find $k' \in [Round(k/2) + 1, k - 1]$ satisfies $y_{k'} \le c' < y_{k'+1}$; iii) when $y_k > a/b$ and $y_{Round(k/2)} \le c' < y_{Round(k/2)+1}$, set $c' = l_{\widetilde{B}_\alpha}$, $L = Round(k/2)$; iv) when $y_{k+1} \le a/b$ and $y_{Round((k+M)/2)} > a/b$, find $k' \in [k + 1, Round((k + M)/2) - 1]$ satisfies $y_{k'} \le c' < y_{k'+1}$; v) when $y_{k+1} \le a/b$ and $y_{Round((k+M)/2)+1} \le a/b$, find $k' \in [Round((k + M)/2) + 1, M - 1]$ satisfies $y_{k'} \le c' < y_{k'+1}$; vi) when $y_{k+1} \le a/b$ and $y_{Round((k+M)/2)} \le c' < y_{Round((k+M)/2)+1}$, set $c' = l_{\widetilde{B}_\alpha}$, $L = Round((k + M)/2)$.

3) Check if $k' = k$, if yes, stop and set $c' = l_{\widetilde{B}_\alpha}$, $k = L$; if no, turn to the next.

4) Compute $s = sign(k' - k)$, $a' = a + \sum_{i=\min(k',k)+1}^{\max(k',k)} y_i \left[\overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) - \underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i)\right]$, $b' = b + \sum_{i=\min(k',k)+1}^{\max(k',k)} \left[\overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) - \underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i)\right]$, $c'' = a'/b'$.

5) Set $c' = c''$, $a = a'$, $b = b'$, and $k = k'$.

6) Find k' such that $y_{k'} \le c' < y_{k'+1}$ (it is searching start from k downward to 1), check if $k' = k$, if yes, stop and set $c' = l_{\widetilde{B}_\alpha}$, $k = L$; if no, turn to the next step.

7) Compute $s = sign(k' - k)$, $a' = a - \sum_{i=k'+1}^{k} y_i \left[\overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) - \underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i)\right]$, $b' = b - \sum_{i=k'+1}^{k} \left[\overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) - \underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i)\right]$, $c'' = a'/b'$, and return to step 5.

It is very similar to compute $r_{\widetilde{B}_\alpha}$. Except for: in 1), $k = Round(M/1.7)$, $a = \sum_{i=1}^{k} y_i \underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) + \sum_{i=k+1}^{M} y_i \overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i)$, $b = \sum_{i=1}^{k} \underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) + \sum_{i=k+1}^{M} \overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i)$; in 4), $a' = a - \sum_{i=\min(k',k)+1}^{\max(k',k)} y_i \left[\overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) - \underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i)\right]$, $b' = b - \sum_{i=\min(k',k)+1}^{\max(k',k)} \left[\overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) - \underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i)\right]$, and in 7) $a' = a - \sum_{i=k+1}^{k'} y_i \left[\overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) - \underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i)\right]$, $b' = b - \sum_{i=k+1}^{k'} \left[\overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) - \underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i)\right]$.

Finally, we summarize the five steps about performing the centroid TR and defuzzification of GT2 FLSs by the EKM and IEKM algorithms.

1) According to the fuzzy reasoning, by weighing or merging all fuzzy rules, find the centroid output GT2 FS $\widetilde{B}$.

2) Break the α into 0, $1/\Delta$, ..., $(\Delta - 1)/\Delta$, 1, totally Δ effective values, in addition, the $\widetilde{B}$ is also decomposed to the corresponding α-level α-plane $\widetilde{B}_\alpha$.

3) Adopt the EKM and IEKM algorithms to calculate each centroid of corresponding IT2 FS $\alpha/\left[l_{\widetilde{B}_\alpha}, r_{\widetilde{B}_\alpha}\right]$.

4) Aggregate the union of all centroids, and compute the defuzzified output (see Equations (10) and (11)).

5) Compare the performances of EKM and IEKM algorithms in the simulation experiments.

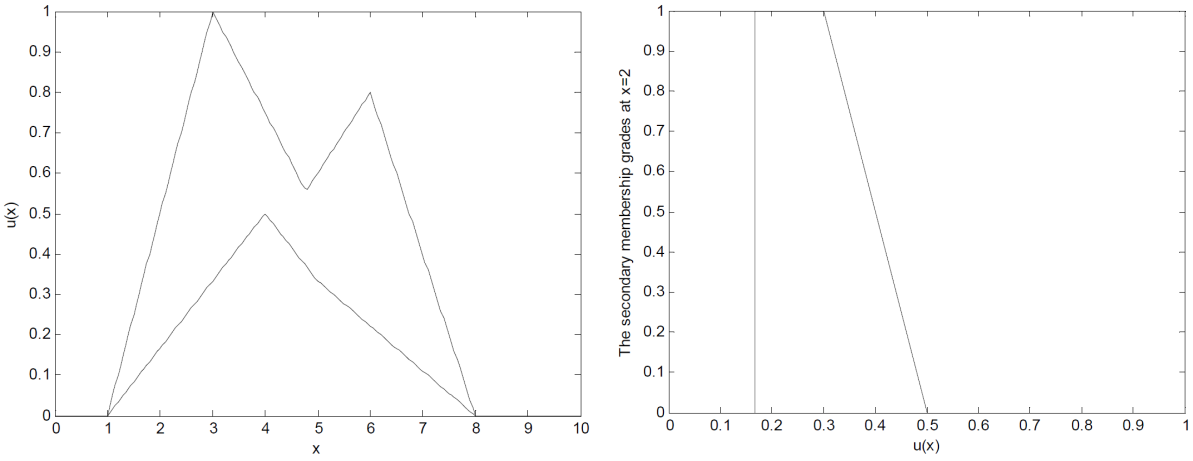
4. **Simulations.** Here we make the assumptions that, before performing the centroid TR and defuzzification, the footprint of uncertainty (FOU) and its corresponding secondary MF (vertical slice) of centroid output GT2 FSs have been known according to weigh or aggregate all fuzzy rules under the guidance of inference. The primary variable x of centroid output GT2 FSs is uniformly sampled and $x \in [0, 10]$. The total number of sampling points is 200, and $x_{i+1} - x_i = 0.05$. In addition, the number of effective α-planes is chosen as 100. Because the computations of $l_{\widetilde{B}_\alpha}$ and $r_{\widetilde{B}_\alpha}$ are very similar, here we only calculate $l_{\widetilde{B}_\alpha}$ for each α-level in the simulation experiments. In the first example, the FOU is composed of piecewise linear functions [27-31], and the corresponding secondary MFs (vertical slices) are trapezoidal MFs. In the second example, the FOU is composed of Gaussian functions [18,22,27-29,32], and the corresponding vertical slices are triangular MFs.

Example 4.1. *Piecewise linear functions with trapezoidal vertical slices.*

As shown in Figure 1, the upper bound of FOU is composed of two linear functions, i.e.,

$$u_1(x) = \begin{cases} \dfrac{x-1}{2}, & 1 \le x \le 3 \\ \dfrac{7-x}{4}, & 3 < x \le 7 \\ 0, & otherwise \end{cases} \tag{15}$$

$$u_2(x) = \begin{cases} \dfrac{x-2}{5}, & 2 \le x \le 6 \\ \dfrac{16-2x}{5}, & 6 < x \le 8 \\ 0, & otherwise. \end{cases} \tag{16}$$



(a) (b)

FIGURE 1. (a) FOU of example one, and (b) its corresponding vertical slice at $x = 2$

The lower bound of FOU is composed of another two linear functions, i.e.,

$$u_3(x) = \begin{cases} \dfrac{x-1}{6}, & 1 \le x \le 4 \\ \dfrac{7-x}{6}, & 4 < x \le 7 \\ 0, & otherwise \end{cases} \tag{17}$$

$$u_4(x) = \begin{cases} \dfrac{x-3}{6}, & 3 \le x \le 5 \\ \dfrac{8-x}{9}, & 5 < x \le 8 \\ 0, & otherwise \end{cases} \tag{18}$$

For any x, the corresponding vertical slice is the trapezoidal MF, and the top left and right end points can be defined as:

$$L(x) = \underline{u}(x) + 0.6w\left(\overline{u}(x) - \underline{u}(x)\right), \quad R(x) = \overline{u}(x) - 0.6(1-w)\left(\overline{u}(x) - \underline{u}(x)\right) \tag{19}$$

where $\underline{u}(x)$ and $\overline{u}(x)$ denote the lower and upper bounds of primary MF, respectively. Here we select $w = 0$ for the test.

As $\Delta = 100$, the left half centroid type-reduced sets computed by the EKM and IEKM algorithms are shown in Figure 2.

Example 4.2. *Piecewise Gaussian functions with triangular vertical slices.*

As shown in Figure 3, the upper bound of FOU is the maximum of two Gaussian functions, i.e.,

$$u_1(x) = \exp\left[-\frac{(x-3)^2}{8}\right] \tag{20}$$

$$u_2(x) = 0.8\exp\left[-\frac{(x-6)^2}{8}\right] \tag{21}$$

The lower bound of FOU is the maximum of two Gaussian functions, i.e.,

$$u_3(x) = 0.5\exp\left[-\frac{(x-3)^2}{2}\right] \tag{22}$$

$$u_4(x) = 0.4 \exp\left[-\frac{(x-6)^2}{2}\right] \tag{23}$$

For any x, the corresponding vertical slice is the triangular MF, and the Apex is defined as:

$$Apex = \underline{u}(x) + w\left[\overline{u}(x) - \underline{u}(x)\right] \tag{24}$$

where $\underline{u}(x)$ and $\overline{u}(x)$ denote the lower and upper bounds of primary MF, respectively. In this test, we choose $w = 0.5$.

As $\Delta = 100$, the left half centroid type-reduced sets computed by the EKM and IEKM algorithms are shown in Figure 4.

Next, the computation times of two types of algorithms are studied. The specific computation times depend on the hardware and software environments, in addition, the

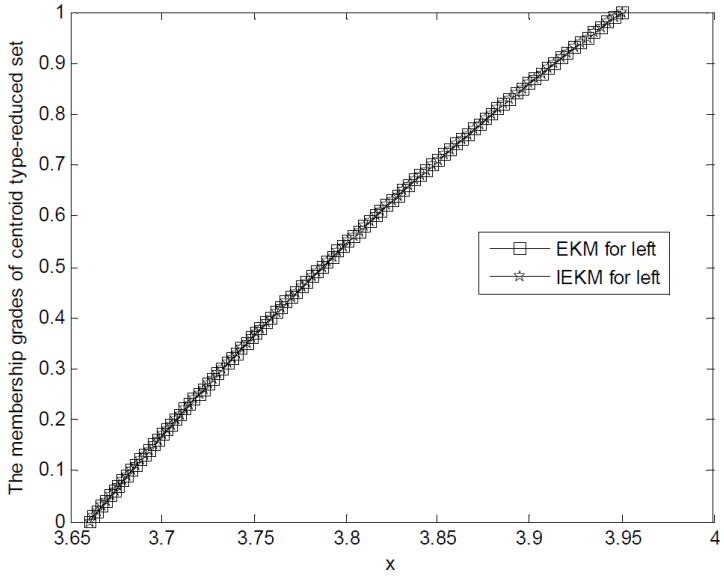
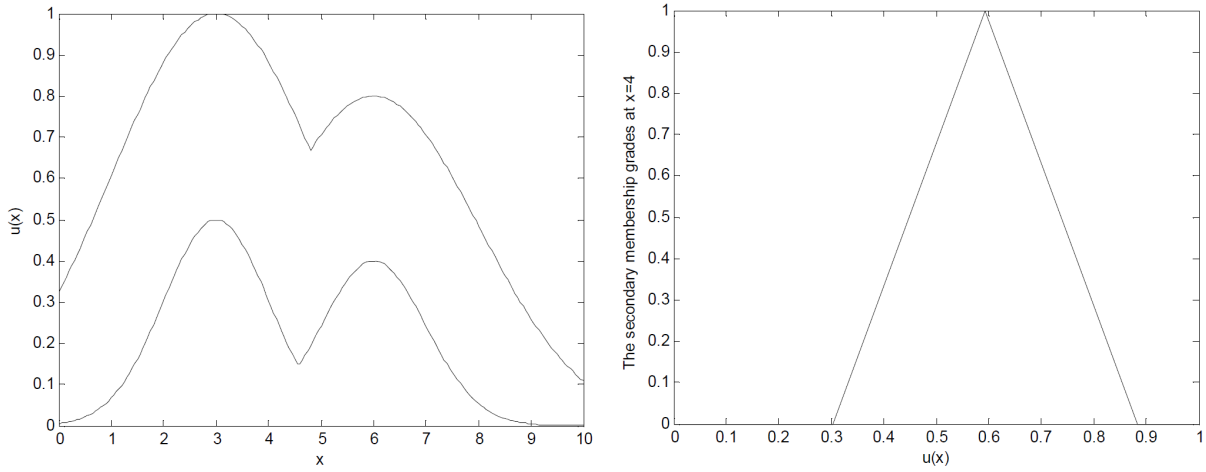


FIGURE 2. The left half centroid type-reduced sets computed by the EKM and IEKM algorithms in Example 4.1



(a) (b)

FIGURE 3. (a) FOU of example two, and (b) its corresponding vertical slice at $x = 4$

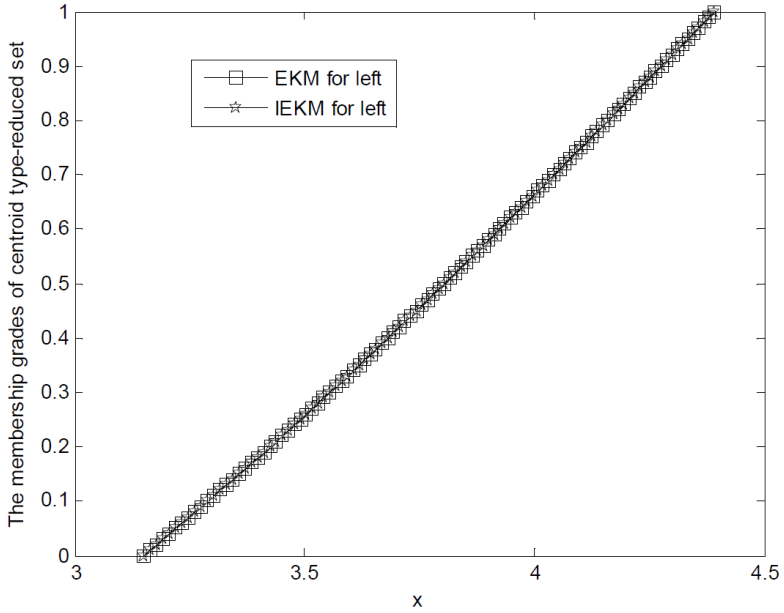


FIGURE 4. The left half centroid type-reduced sets computed by the EKM and IEKM algorithms in Example 4.2

TABLE 1. The total computation times of two types of algorithms for computing the left half centroid type-reduced MFs

Num	EKM	IEKM	TRR (%)
Example 4.1	0.00196	0.00106	45.92
Example 4.2	0.00189	0.00112	40.74
Average	0.00193	0.00109	43.33

computation results are unrepeatable. Here the simulation platform is selected as the Microsoft Windows XP Professional system, and the dual-core CPU dell desktop with E5300@2.6GHz and 2.00GB memory. We program all the algorithms by the Matlab 2013a. Table 1 gives the total time consuming of two types of algorithms for computing the left half centroid type-reduced MFs, in which the unit of time is second (s), the third line in Table 1 denotes the average of two examples, and the fourth column in Table 1 represents the relative time reducing rate (TRR) for the IEKM algorithms compared with the EKM algorithms, which is defined as:

$$TRRIEKM = (t_0 - t_n)/t_0 \times 100\% \tag{25}$$

where t_0 represents the computation time for the EKM algorithms, and t_n is the computation time for the IEKM algorithms.

Observing from Table 1 and Figures 2 and 4, we can get the conclusions for both the computation results and the computation times. (1) When computing the left half centroid type-reduced MFs in two examples, the EKM and IEKM algorithms can obtain almost the same result, for the curves of two types of algorithms coincide with each other. (2) For the computation times, the IEKM algorithms can obtain the largest time reducing rate as 45.92% and the average time reducing rate as 43.33% compared with the EKM algorithms.

The above simulation analyses verify that the IEKM algorithm can be an effective method for performing the centroid type-reduction of GT2 FLSs. Although the calculation results of two types of algorithms are the same, it is apparent that the computation times

of IEKM algorithms are less than the EKM algorithms. Therefore, the IEKM algorithms can reduce the computation cost of performing the TR of GT2 FLSs.

5. **Conclusions.** According to the fuzzy reasoning the α-planes representation of GT2 FSs, this paper extends the IEKM algorithms to perform the centroid TR of GT2 FLSs. Two simulation examples illustrate the performances of IEKM algorithms for computing the left half centroid type-reduced sets and the calculation times. The computation results of the IEKM and EKM algorithms are the same; however, the computation times of the former are less than the latter, which can provide the potential value for applying GT2 FLSs.

Next we will investigate the centroid and center-of-sets TR of IT2 and GT2 FLSs [24,27,29-36], and study the forecasting and control problems by designing T2 FLSs [11,13-15,17,19] optimized with intelligent algorithms. Future studies will focus on T2 FLSs design and applications based on [4-6,12,13,15,29,37] and this paper.

Acknowledgment. The paper is sponsored by the National Natural Science Foundation of China (No. 61773188, No. 61803189), and the Liaoning Province Natural Science Foundation Guidance Project (No. 20180550056). The author is very grateful for professor Jerry Mendel, who has provided some valuable suggestions.

REFERENCES

[1] A. Khosravi, S. Nahavandi, D. Creighton and D. Srinivasan, Interval type-2 fuzzy logic systems for load forecasting: A comparative study, *IEEE Trans. Power Systems*, vol.27, no.3, pp.1274-1282, 2012.

[2] A. Khosravi and S. Nahavandi, Load forecasting using interval type-2 fuzzy logic systems: Optimal type reduction, *IEEE Trans. Industrial Informatics*, vol.10, no.2, pp.1055-1063, 2014.

[3] D. Bernardo, H. Hagras and E. Tsang, A genetic type-2 fuzzy logic based system for the generation of summarized linguistic predictive models for financial applications, *Soft Computing*, vol.17, no.12, pp.2185-2201, 2013.

[4] Y. Chen, D. Wang and S. Tong, Forecasting studies by designing Mamdani interval type-2 fuzzy logic systems: With combination of BP algorithms and KM algorithms, *Neurocomputing*, vol.174, pp.1133-1146, 2016.

[5] S. Barkat, A. Tlemcani and H. Nouri, Noninteracting adaptive control of PMSM using interval type-2 fuzzy logic systems, *IEEE Trans. Fuzzy Systems*, vol.19, no.5, pp.925-936, 2011.

[6] Y. Chen and D. Wang, Study on permanent magnetic drive forecasting by designing Takagi Sugeno Kang type interval type-2 fuzzy logic systems, *Transactions of the Institute of Measurement and Control*, vol.40, no.6, pp.2011-2023, 2018.

[7] H. Hagras, A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots, *IEEE Trans. Fuzzy Systems*, vol.12, no.4, pp.524-539, 2004.

[8] C. S. Lee, M. H. Wang and H. Hagras, Type-2 fuzzy ontology and its application to personal diabetic-diet recommendation, *IEEE Trans. Fuzzy Systems*, vol.18, no.2, pp.316-328, 2010.

[9] O. Mendoza, P. Melin and O. Castillo, Interval type-2 fuzzy logic and modular networks for face recognition applications, *Applied Soft Computing*, vol.9, no.4, pp.1377-1387, 2009.

[10] A. Niewiadomski, On finity, countability, cardinalities, and cylindric extensions of type-2 fuzzy sets in linguistic summarization of databases, *IEEE Trans. Fuzzy Systems*, vol.18, no.3, pp.532-545, 2010.

[11] J. M. Mendel, General type-2 fuzzy logic systems made simple: A tutorial, *IEEE Trans. Fuzzy Systems*, vol.22, no.5, pp.1162-1182, 2014.

[12] C. Wagner and H. Hagras, Toward general type-2 fuzzy logic systems based on zSlices, *IEEE Trans. Fuzzy Systems*, vol.18, no.4, pp.637-660, 2010.

[13] Y. Chen, D. Wang and W. Ning, Forecasting by TSK general type-2 fuzzy logic systems optimized with genetic algorithms, *Optimal Control Applications & Methods*, vol.39, no.1, pp.393-409, 2018.

[14] P. Melin, C. I. Gonzalez, J. R. Castro, O. Mendoza and O. Castillo, Edge-detection method for image processing based on generalized type-2 fuzzy logic, *IEEE Trans. Fuzzy Systems*, vol.22, no.6, pp.1515-1525, 2014.

[15] Y. Chen and D. Wang, Forecasting by general type-2 fuzzy logic systems optimized with QPSO algorithms, *International Journal of Control, Automation and Systems*, vol.15, no.6, pp.2950-2958, 2017.

[16] C. Gonzalez, J. R. Castro, P. Melin and O. Castillo, An edge detection method based on generalized type-2 fuzzy logic, *Soft Computing*, vol.20, no.2, pp.773-784, 2016.

[17] O. Castillo, L. Amador-Angulo, J. R. Castro and 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, *Information Sciences*, vol.354, pp.257-274, 2016.

[18] T. Wang, Y. Chen and S. Tong, Fuzzy reasoning models and algorithms on type-2 fuzzy sets, *International Journal of Innovative Computing, Information and Control*, vol.4, no.10, pp.2451-2460, 2008.

[19] M. A. Sanchez, O. Castillo and J. R. Castro, Generalized type-2 fuzzy systems for controlling a mobile robot and a performance comparison with interval type-2 and type-1 fuzzy systems, *Expert Systems with Applications*, vol.42, no.14, pp.5904-5914, 2015.

[20] J. M. Mendel, On KM algorithms for solving type-2 fuzzy set problems, *IEEE Trans. Fuzzy Systems*, vol.21, no.3, pp.426-446, 2013.

[21] K. K. Karnik and J. M. Mendel, Centroid of a type-2 fuzzy set, *Information Sciences*, vol.132, no.1, pp.195-220, 2001.

[22] J. M. Mendel and F. Liu, Super-exponential convergence of the Karnik-Mendel algorithms for computing the centroid of an interval type-2 fuzzy set, *IEEE Trans. Fuzzy Systems*, vol.15, no.2, pp.309-320, 2007.

[23] D. Wu and J. M. Mendel, Enhanced Karnik-Mendel algorithms, *IEEE Trans. Fuzzy Systems*, vol.17, no.4, pp.923-934, 2009.

[24] J. Wang, W. Ji, X. Fang and S. Gu, Improvement of enhanced Karnik-Mendel algorithm for interval type-2 fuzzy sets, *Control and Decision*, vol.28, no.8, pp.1165-1172, 2013.

[25] Y. Chen and D. Wang, Forecasting by designing Mamdani general type-2 fuzzy logic systems optimized with quantum particle swarm optimization algorithms, *Transactions of the Institute of Measurement and Control*, DOI: 10.1177/0142331218816753, 2019.

[26] J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*, Prentice-Hall, Englewood Cliffs, NJ, USA, 2001.

[27] F. Liu, An efficient centroid type-reduction strategy for general type-2 fuzzy logic system, *Information Sciences*, vol.178, no.9, pp.2224-2236, 2008.

[28] J. M. Mendel, F. Liu and D. Zhai, Alpha-plane representation for type-2 fuzzy sets: Theory and applications, *IEEE Trans. Fuzzy Systems*, vol.17, no.5, pp.1189-1207, 2009.

[29] Y. Chen, D. Wang and W. Ning, Studies on centroid type-reduction algorithms for general type-2 fuzzy logic systems, *International Journal of Innovative Computing, Information and Control*, vol.11, no.6, pp.1987-2000, 2015.

[30] Y. Chen, Studies on centroid type-reduction of general type-2 fuzzy logic systems with enhanced opposite direction searching algorithms, *International Journal of Innovative Computing, Information and Control*, vol.15, 2019.

[31] Y. Chen and D. Wang, Study on centroid type-reduction of general type-2 fuzzy logic systems with weighted enhanced Karnik-Mendel algorithms, *Soft Computing*, vol.22, no.4, pp.1361-1380, 2018.

[32] J. M. Mendel and X. Liu, Simplified interval type-2 fuzzy logic systems, *IEEE Trans. Fuzzy Systems*, vol.21, no.6, pp.1056-1069, 2013.

[33] S. Greenfield and F. Chiclana, Accuracy and complexity evaluation of defuzzification strategies for the discretised interval type-2 fuzzy set, *International Journal of Approximate Reasoning*, vol.54, no.8, pp.1013-1033, 2013.

[34] Y. Chen, Study on weighted Nagar-Bardini algorithms for centroid type-reduction of interval type-2 fuzzy logic systems, *Journal of Intelligent & Fuzzy Systems*, vol.34, no.4, pp.2417-2428, 2018.

[35] J. W. Li, R. John, S. Coupland and G. Kendall, On Nie-Tan operator and type-reduction of interval type-2 fuzzy sets, *IEEE Trans. Fuzzy Systems*, vol.26, no.2, pp.1036-1039, 2018.

[36] M. A. Khanesar, A. Jalalian and O. Kaynak, Improving the speed of center of set type-reduction in interval type-2 fuzzy systems by eliminating the need for sorting, *IEEE Trans. Fuzzy Systems*, vol.25, no.5, pp.1193-1206, 2017.

[37] T. Kumbasar, Revisiting Karnik-Mendel algorithms in the framework of linear fractional programming, *International Journal of Approximate Reasoning*, vol.82, pp.1-21, 2017.