International Journal of Innovative
Computing, Information and Control
Volume **15**, Number **4**, August **2019**

 ISSN 1349-4198
pp. 1425–1439

STUDY ON CENTROID TYPE-REDUCTION OF GENERAL TYPE-2 FUZZY LOGIC SYSTEMS WITH ENHANCED OPPOSITE DIRECTION SEARCHING ALGORITHMS

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 September 2018; revised January 2019

Abstract. *As an emerging technology, type-2 fuzzy logic systems have excellent abilities to cope with uncertainties. However, the high computational cost of type-reduction (TR) block under the guidance of inference may hinder them from real world applications. In early times, type-reduction algorithms are developed based on the calculation of centroids of interval type-2 fuzzy sets (IT2 FSs). Generally speaking, the Karnik-Mendel (KM) algorithms are used to perform the TR. Recently, general type-2 fuzzy logic systems (GT2 FLSs) based on the α-planes representation theory of general type-2 fuzzy sets (GT2 FSs) have become a hot topic in academic field. This paper introduces the efficient enhanced opposite direction searching (EODS) algorithms for performing the centroid TR of general type-2 fuzzy logic systems (GT2 FLSs). In addition, the blocks of fuzzy reasoning and defuzzification are also discussed. Two computer simulation examples are used to illustrate and analyze the performances of EODS algorithms when computing the outputs of GT2 FLSs. Compared with the most commonly used KM algorithms, simulation results show that the EODS algorithms are computationally faster without loosing the calculation accuracy, which provide the potential value for designers and adopters of type-2 fuzzy logic systems.*

Keywords: General type-2 fuzzy logic systems, α-planes, KM algorithms, EODS algorithms, Computer simulation

1. **Introduction.** As we all know, the membership grades of T1 FSs are crisp numbers. While the membership grades of T2 FSs are themselves T1 FSs. T2 FSs can be considered as higher order uncertain parameter models compared with T1 FSs. The design degree of freedom increases as it transits from the T1 FSs to the T2 FSs, which makes the latter can model and cope with uncertainties better. As the most widely used fuzzy systems, IT2 FLSs have been successfully used in many areas like power systems [1,2], financial systems [3,4], medical systems [5], hot strip mills [6], permanent magnetic drive [7,8], autonomous mobile robots [9,10], and pattern recognition systems [11,12]. The processes of these areas are characterized high uncertainty, nonlinearity, and time varying behavior [13]. Recent load forecasting studies also show that the IT2 FLSs [14] have advantages over T1 FLSs for dealing with uncertainties. The design and computations of GT2 FLSs are more complicated than IT2 FLSs. However, the recent GT2 FLSs based on the α-planes (or zSlices) representation of GT2 FSs [15-17,41] greatly reduce the computations, and this makes the GT2 FLSs gradually applied in some areas.

DOI: 10.24507/ijicic.15.04.1425

In general, a type-2 fuzzy logic system is composed of five blocks (as shown in Figure 1), which are fuzzifier, inference, rules, type-reducer, and defuzzifier. Among them, the type-reduction under the guidance of inference plays the central role, which transforms the T2 FS into the T1 FS. T2 FLSs are much more complicated as they contain the block of type-reduction. Naturally, IT2 FLSs use IT2 FSs. While GT2 FLSs adopt GT2 FSs. As we all know, the secondary membership grades of IT2 FSs are all equal to 1. The secondary membership grades of GT2 FSs lie on the interval $[0, 1]$, which make them measure the uncertainties of membership functions (MFs) in non-uniformly manner. Therefore, GT2 FSs can be viewed as higher order uncertainty model than IT2 FSs. As the number of design degrees of freedom increases, GT2 FLSs have the potential to outperform T1 and IT2 FLSs on coping with uncertainties.

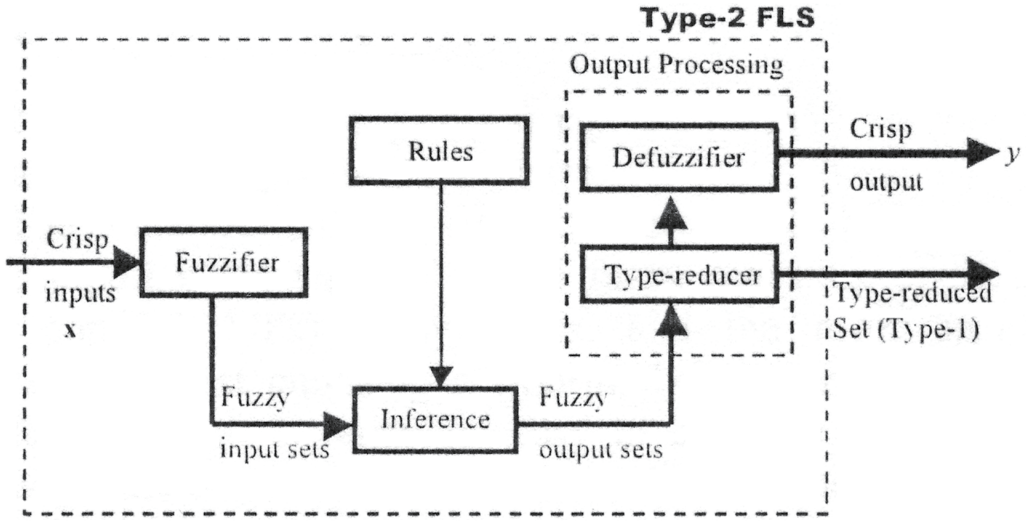


FIGURE 1. A T2 FLS [18]

The most popular KM algorithms [19,20] were developed to compute the centroids of IT2 FSs or perform the TR of IT2 FLSs. This type of computational intensive iterative algorithms converges monotonically and super-exponentially fast [29]. Moreover, the KM algorithms can preserve the MFs uncertainties flow in the systems. However, it usually needs two to six iterations for each KM iteration to converge. In order to reduce the computation cost, Wu and Mendel proposed improvements to the KM algorithms in [21], which were called as EKM algorithms. Extensive simulations have shown that on average the EKM algorithms can save about two iterations, which correspond to a more than 39% reduction in computation time. Another type of non-KM algorithms was developed by Hu et al., which were referred to as the EODS algorithms [22,23]. Simulations have proved that the EODS algorithms are computationally faster than the KM algorithms on solving the centroids of IT2 FSs.

Inspired by [15,17,18,23-27], this paper performs the centroid TR of GT2 FLSs by combining the α-planes representation theory of GT2 FSs and the EODS algorithms. In addition, the blocks of fuzzy reasoning, type-reduction and defuzzification are also discussed. As for a certain number of α-planes and the number of α-planes in a certain range, two simulation examples are provided to illustrate and analyze the performances of the EODS algorithms for computing the centroid type-reduced sets and defuzzified outputs. Compared with the KM algorithms, the EODS algorithms are computationally faster without loosing the calculation accuracy.

The rest of the paper is organized as follows. Section 2 briefly introduces the background of GT2 FLSs. Section 3 provides the EODS algorithms for performing the centroid TR of GT2 FLSs. Section 4 gives the performances of the EODS algorithms compared with the KM algorithms according to simulation examples. Finally, Section 5 gives the conclusions.

2. GT2 FLSs.

Definition 2.1. *A GT2 FS $\widetilde{A}$ can be characterized by its T2 MF $\mu_{\widetilde{A}}(x,u)$, i.e.,*

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

where the primary variable $u \in J_x \subseteq [0,1]$, Equation (1) is called as the point-value expression of $\widetilde{A}$, while its compact form is as

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

Definition 2.2. *The two dimensional support of $\mu_{\widetilde{A}}(x,u)$ is referred to as the footprint of uncertainty (FOU) of $\widetilde{A}$, i.e.,*

$$FOU\left(\widetilde{A}\right) = \bigcup\nolimits_{x \in X} J_x = \left\{(x,u) \in X \times [0,1] | \mu_{\widetilde{A}}(x,u) > 0\right\} \tag{3}$$

where $FOU\left(\widetilde{A}\right)$ is bounded by its upper membership function (UMF) and lower membership function (LMF), and they are denoted as $\overline{\mu}_{\widetilde{A}}(x)$ and $\underline{\mu}_{\widetilde{A}}(x)$, i.e.,

$$UMF\left(\widetilde{A}\right) = \overline{\mu}_{\widetilde{A}}(x) = \overline{FOU\left(\widetilde{A}\right)}, \quad LMF\left(\widetilde{A}\right) = \underline{\mu}_{\widetilde{A}}(x) = \underline{FOU\left(\widetilde{A}\right)} \tag{4}$$

Definition 2.3. *A vertical slice of $\mu_{\widetilde{A}}(x,u)$ is a secondary MF, i.e.,*

$$\mu_{\widetilde{A}}(x = x', u) \equiv \mu_{\widetilde{A}}(x') = \int_{u \in [0,1]} f_{x'}(u)/u \tag{5}$$

here we represent the secondary MF $\mu_{\widetilde{A}}(x')$ as $\widetilde{A}(x)$ for simplicity. Therefore, Equation (2) can be reexpressed as

$$\widetilde{A} = \int_{\forall x \in X} \widetilde{A}(x)/x \tag{6}$$

Let $\widetilde{A}_\alpha(x)$ denote the α-cut of $\widetilde{A}(x)$, $\alpha \in [0,1]$, i.e.,

$$\widetilde{A}_\alpha(x) = \{u | f_x(u) \geq \alpha\} = [a_\alpha(x), b_\alpha(x)] \tag{7}$$

Then $\widetilde{A}(x)$ can be represented by the following α-cuts decomposition, i.e.,

$$\widetilde{A}(x) = \bigcup_{\forall \alpha \in [0,1]} \left[\alpha / \widetilde{A}_\alpha(x)\right] = \sup_{\forall \alpha \in [0,1]} \{\alpha / [a_\alpha(x), b_\alpha(x)]\} \tag{8}$$

where sup denotes the supremum, and $\bigcup$ denotes the union operation.

Next, we substitute (8) into (6) to obtain the vertical slices representation of $\widetilde{A}$, i.e.,

$$\widetilde{A} = \int_{\forall x \in X} \left\{ \bigcup_{\forall \alpha \in [0,1]} \left[\alpha / \widetilde{A}_\alpha(x)\right] \right\} \Big/ x \tag{9}$$

Then the horizontal slices [15,17,18,27] representation of $\widetilde{A}$ can be obtained as

$$\widetilde{A} = \bigcup_{\forall \alpha \in [0,1]} \left\{ \int_{\forall x \in X} \left[\alpha / \widetilde{A}_\alpha(x)\right] / x \right\} = \bigcup_{\forall \alpha \in [0,1]} \left\{ \alpha \Big/ \left[\int_{\forall x \in X} \widetilde{A}_\alpha(x)/x \right] \right\} \tag{10}$$

where the α-plane $\widetilde{A}_\alpha$ is the union of primary membership functions of $\widetilde{A}$ whose secondary membership grades must be greater than or equal to α, i.e.,

$$\widetilde{A}_\alpha = \int_{\forall x \in X} \int_{\forall u \in [0,1]} \left\{ (x,u) | \mu_{\widetilde{A}(x)}(u) \geq \alpha \right\} \tag{11}$$

Definition 2.4. *An α-plane which is raised to the α-level is usually represented by $R_{\widetilde{A}_\alpha}$, i.e.,*

$$R_{\widetilde{A}_\alpha} = \alpha / \widetilde{A}_\alpha \tag{12}$$

Generally speaking, GT2 FLSs can be divided into Mamdani type and TSK type. Here we only focus on the Mamdani type. Suppose that a Mamdani GT2 FLS has n inputs $x_1 \in X_1, \ldots, x_n \in X_n$ and one output $y \in Y$, without loss of generality, which can be characterized by N fuzzy rules, where the sth rule is the form:

$$\text{If } x_1 \text{ is } \widetilde{F}_1^s \text{ and } \ldots \text{ and } x_n \text{ is } \widetilde{F}_n^s, \text{ then } y \text{ is } \widetilde{G}^s \ (s = 1, \ldots, N) \tag{13}$$

Then the inference process [26] is as follows.

Here we adopt the singleton fuzzifier [19] for simplicity, i.e., when $x_i = x_i'$, only the vertical slice (secondary MF) $\widetilde{F}_i^s(x_i')$ of GT2 FS $\widetilde{F}_i^s$ is activated, and which has the α-cuts decomposition as

$$\widetilde{F}_i^s(x_i') = \sup_{\forall \alpha \in [0,1]} \alpha / \left[a_{i,\alpha}^s(x_i'), b_{i,\alpha}^s(x_i') \right] \tag{14}$$

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

$$F_\alpha : \begin{cases} F_\alpha^s(x') \equiv \left[\underline{f}_\alpha^s(x'), \overline{f}_\alpha^s(x') \right] \\ \underline{f}_\alpha^s(x') \equiv T_{i=1}^n a_{i,\alpha}^s(x_i') \\ \overline{f}_\alpha^s(x') \equiv T_{i=1}^n b_{i,\alpha}^s(x_i') \end{cases} \tag{15}$$

where T denotes the product or minimum t-norm.

Suppose that the consequent GT2 FS $\widetilde{G}^s$ has the α-plane (or zSlices) $\widetilde{G}_\alpha^s$ at the corresponding α-level, i.e.,

$$\widetilde{G}_\alpha^s = \int_{\forall y \in Y} \widetilde{G}_\alpha^s(y)/y = \int_{\forall y \in Y} [g_{L,\alpha}^s(y), g_{R,\alpha}^s(y)]/y \tag{16}$$

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

$$\widetilde{B}_\alpha^s : \begin{cases} FOU\left(\widetilde{B}_\alpha^s\right) = \left[\underline{\mu}_{\widetilde{B}_\alpha^s}(y|x'), \overline{\mu}_{\widetilde{B}_\alpha^s}(y|x') \right] \\ \underline{\mu}_{\widetilde{B}_\alpha^s}(y|x') = \underline{f}_\alpha^s(x') * g_{L,\alpha}^s(y) \\ \overline{\mu}_{\widetilde{B}_\alpha^s}(y|x') = \overline{f}_\alpha^s(x') * g_{R,\alpha}^s(y) \end{cases} \tag{17}$$

where $*$ denotes the product or minimum operation.

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

$$\widetilde{B}_\alpha : \begin{cases} FOU\left(\widetilde{B}_\alpha\right) = \left[\underline{\mu}_{\widetilde{B}_\alpha}(y|x'), \overline{\mu}_{\widetilde{B}_\alpha}(y|x') \right] \\ \underline{\mu}_{\widetilde{B}_\alpha}(y|x') = \underline{\mu}_{\widetilde{B}_\alpha^1}(y|x') \vee \cdots \vee \underline{\mu}_{\widetilde{B}_\alpha^N}(y|x') \\ \overline{\mu}_{\widetilde{B}_\alpha}(y|x') = \overline{\mu}_{\widetilde{B}_\alpha^1}(y|x') \vee \cdots \vee \overline{\mu}_{\widetilde{B}_\alpha^N}(y|x') \end{cases} \tag{18}$$

For the sake of obtaining the type-reduced set $Y_{C,\alpha}(x')$ at the α-level, compute the centroid [29] of $\widetilde{B}_\alpha$, i.e.,

$$Y_{C,\alpha}(x') = C_{\widetilde{B}_\alpha}(x') = \alpha / \left[l_{\widetilde{B}_\alpha}(x'), r_{\widetilde{B}_\alpha}(x') \right] \tag{19}$$

where the two end points $l_{\widetilde{B}_\alpha}(x')$ and $r_{\widetilde{B}_\alpha}(x')$ can be calculated by the type-reduction algorithms as

$$l_{\widetilde{B}_\alpha}(x') = \min_{\mu_{\widetilde{B}_\alpha}(y_i)\in\left[\underline{\mu}_{\widetilde{B}_\alpha}(y_i),\overline{\mu}_{\widetilde{B}_\alpha}(y_i)\right]} \frac{\sum_{i=1}^{M} y_i \mu_{R_{\widetilde{B}_\alpha}}(y_i)}{\sum_{i=1}^{M} \mu_{R_{\widetilde{B}_\alpha}}(y_i)} \tag{20}$$

$$r_{\widetilde{B}_\alpha}(x') = \max_{\mu_{\widetilde{B}_\alpha}(y_i)\in\left[\underline{\mu}_{\widetilde{B}_\alpha}(y_i),\overline{\mu}_{\widetilde{B}_\alpha}(y_i)\right]} \frac{\sum_{i=1}^{M} y_i \mu_{R_{\widetilde{B}_\alpha}}(y_i)}{\sum_{i=1}^{M} \mu_{R_{\widetilde{B}_\alpha}}(y_i)} \tag{21}$$

Finally we aggregate all the α-planes $Y_{C,\alpha}$ to constitute the type-1 FS Y_C, i.e.,

$$Y_C = \sup_{\forall\alpha\in[0,1]} \alpha / Y_{C,\alpha}(x') \tag{22}$$

In the practical computations, let the number of effective α-planes be m, we uniformly decompose the value of α into m alpha-planes at $\alpha_1, \alpha_2, \ldots, \alpha_n$, and then the output of GT2 FLSs should be

$$y(x') = \sum_{i=1}^{m} \alpha_i \left[\left(l_{\widetilde{B}_{\alpha_i}}(x') + r_{\widetilde{B}_{\alpha_i}}(x')\right) \Big/ 2\right] \Big/ \sum_{i=1}^{m} \alpha_i \tag{23}$$

Equation (23) is called as the average of end points defuzzification method, which is proposed by Wagner and Hagras [17]. So far, most type-reduction algorithms will obtain different output values. However, it is not clear which algorithm can obtain better results than others.

3. **EODS Algorithms.** In order to give the EODS algorithms, here we first introduce the ODS algorithms. Mendel and Liu had proved in [29] that the left end point of an IT2 FS $l_{\widetilde{B}_\alpha}$ must lie between the left switching point and its right neighbor, i.e.,

$$y_L \le l_{\widetilde{B}_\alpha} = l_{\widetilde{B}_\alpha}(L) < y_{L+1} \tag{24}$$

According to Equations (19)-(21), the two end points $l_{\widetilde{B}_\alpha}(x')$ and $r_{\widetilde{B}_\alpha}(x')$ can be calculated as

$$l_{\widetilde{B}_\alpha}(x') = \frac{\sum_{i=1}^{L} y_i \overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) + \sum_{i=L+1}^{N} y_i \underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i)}{\sum_{i=1}^{L} \overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) + \sum_{i=L+1}^{N} \underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i)} \tag{25}$$

$$r_{\widetilde{B}_\alpha}(x') = \frac{\sum_{i=1}^{R} y_i \underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) + \sum_{i=R+1}^{N} y_i \overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i)}{\sum_{i=1}^{R} \underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) + \sum_{i=R+1}^{N} \overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i)} \tag{26}$$

where L and R are the switching points.

In the light of Equation (25), Equation (24) can be extended to Equations (27) and (28), i.e.,

$$\sum_{i=1}^{L-1} (y_L - y_i) \overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) \le \sum_{i=L+1}^{N} (y_i - y_L) \underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) \tag{27}$$

$$\sum_{i=1}^{L} (y_{L+1} - y_i) \overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) > \sum_{i=L+2}^{N} (y_i - y_{L+1}) \underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) \tag{28}$$

Choose $\overline{S_L(k)}$ and $\underline{S_R(k)}$ as

$$\overline{S_L(k)} = \sum_{i=1}^{k-1} (y_k - y_i) \overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) \tag{29}$$

$$\underline{S_R(k)} = \sum_{i=k+1}^{N} (y_i - y_k)\underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) \tag{30}$$

Equations (27) and (28) can be united as Equation (31), i.e.,

$$\left(\overline{S_L(L)} \leq \underline{S_R(L)}\right) \& \left(\overline{S_L(L+1)} > \underline{S_R(L+1)}\right) \tag{31}$$

where & denotes the logic and.

Then $\overline{S_L(k)}$ and $\underline{S_R(k)}$ can be computed in the iterative manner as

$$\begin{aligned} \overline{S_L(k)} - \overline{S_L(k-1)} &= \sum_{i=1}^{k-1} (y_k - y_i)\overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) - \sum_{i=1}^{k-2} (y_{k-1} - y_i)\overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) \\ &= (y_k - y_{k-1}) \sum_{i=1}^{k-1} \overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) \end{aligned} \tag{32}$$

$$\begin{aligned} \underline{S_R(k)} - \underline{S_R(k+1)} &= \sum_{i=k+1}^{N} (y_i - y_k)\underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) - \sum_{i=k+2}^{N} (y_i - y_{k+1})\underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) \\ &= (y_{k+1} - y_k) \sum_{i=k+1}^{N} \underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) \end{aligned} \tag{33}$$

Equations (32) and (33) illustrate that $\overline{S_L(k)}$ iterates from $\overline{S_L(k-1)}$ while $\underline{S_R(k)}$ iterates from $\underline{S_R(k+1)}$. The calculation of $\overline{S_L(k)}$ is a positive searching process with the increase of k, while the calculation of $\underline{S_R(k)}$ is a negative searching process with the decrease of k. Interestingly, these two opposite direction searching processes meet just at the switching point $k = L$.

In like manner, the left end point of an IT2 FS $r_{\widetilde{B}_\alpha}$ must lie between the right switching point and its right neighbor, i.e.,

$$y_R \leq r_{\widetilde{B}} = r_{\widetilde{B}}(R) < y_{R+1} \tag{34}$$

Then the right switching point R satisfies

$$\left(\underline{S_L(R)} \leq \overline{S_R(R)}\right) \& \left(\underline{S_L(R+1)} > \overline{S_R(R+1)}\right) \tag{35}$$

where $\underline{S_L(k)}$ and $\overline{S_R(k)}$ are as

$$\underline{S_L(k)} = \sum_{i=1}^{k-1} (y_k - y_i)\underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) \tag{36}$$

$$\overline{S_R(k)} = \sum_{i=k+1}^{N} (y_i - y_k)\overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) \tag{37}$$

Then we still use the iterative manner to calculate $\underline{S_L(k)}$ and $\overline{S_R(k)}$ as

$$\begin{aligned} \underline{S_L(k)} - \underline{S_L(k-1)} &= \sum_{i=1}^{k-1} (y_k - y_i)\underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) - \sum_{i=1}^{k-2} (y_{k-1} - y_i)\underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) \\ &= (y_k - y_{k-1}) \sum_{i=1}^{k-1} \underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) \end{aligned} \tag{38}$$

$$\begin{aligned}\overline{S_R(k)} - \overline{S_R(k+1)} &= \sum_{i=k+1}^{N} (y_i - y_k)\overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) - \sum_{i=k+2}^{N} (y_i - y_{k+1})\overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) \\ &= (y_{k+1} - y_k) \sum_{i=k+1}^{N} \overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i)\end{aligned} \tag{39}$$

When the two switching points L and R are found, both the left and right centroid end points can be simply computed by Equations (25) and (26).

Hu et al. [22] proposed the first initial enhanced version of ODS algorithms, which were called as the EODS algorithms. From the ODS algorithms, we can draw that

$$\begin{aligned}&\underline{S_R(L)} - \overline{S_L(L)} \\ &= \sum_{i=L+1}^{N} (y_i - y_L)\underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) - \sum_{i=1}^{L-1} (y_L - y_i)\overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) \\ &= \sum_{i=1}^{L} y_i\overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) + \sum_{i=L+1}^{N} y_i\underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) y_L \left[\sum_{i=1}^{L} \overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) + \sum_{i=L+1}^{N} \underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i)\right]\end{aligned} \tag{40}$$

Then the quick formula of computing centroid end points can be obtained as

$$l_{\widetilde{B}_\alpha} = y_L + \frac{\underline{S_R(L)} - \overline{S_L(L)}}{\sum_{i=1}^{L} \overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) + \sum_{i=L+1}^{N} \underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i)} \tag{41}$$

Similarly, the quick formula of computing centroid right end point can be obtained as

$$r_{\widetilde{B}_\alpha} = y_R + \frac{\overline{S_R(R)} - \underline{S_L(R)}}{\sum_{i=1}^{R} \underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i) + \sum_{i=R+1}^{N} \overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_i)} \tag{42}$$

We can observe that S_L and S_R are memorized in the form of arrays, which may reduce the computation speed to some extent. Then Wu introduced a new type of EODS algorithms in [23], which were based on the idea of initial EODS algorithms; however, the implementation was slightly different from [22] because two improvements were made.

1) Arrays S_L and S_R in [22] are changed to scalars S_l and S_r to reduce the memory requirement and accelerate the computation speed.
2) Substitute y_L and y_R in [22] by y_m to simplify the computations of $l_{\widetilde{B}_\alpha}$ and $r_{\widetilde{B}_\alpha}$.

For each of $l_{\widetilde{B}_\alpha}$ and $r_{\widetilde{B}_\alpha}$, the new EODS algorithms iteratively calculate a positive search process (see S_l in Table 1) and a negative search process (see S_r in Table 1).

In order to perform the EODS algorithms to compute the centroid TR of GT2 FLSs, the following specific steps can be defined.

Step 1: Unite all the fired rules in the GT2 FLSs to find the output GT2 FS $\widetilde{B}$.

Step 2: Resolve the α to Δ effective values as $0, 1/\Delta, \ldots, (\Delta-1)/\Delta, 1$. Moreover, decompose the GT2 FS $\widetilde{B}$ into multiple α-planes $\widetilde{B}_\alpha$ with above α values.

Step 3: Use the KM and EODS algorithms to calculate the centroid $\alpha/\left[l_{\widetilde{B}_\alpha}(x'), r_{\widetilde{B}_\alpha}(x')\right]$ for each of the related IT2 FS $R_{\widetilde{B}_\alpha}$.

Step 4: Calculate the union of all centroids in Step 3.

Step 5: Compare the performances of the EODS algorithms with the KM algorithms according to computing the centroid type-reduced MFs and defuzzified values. Table 1 gives the EODs algorithms to compute the centroid end points of an IT2 FS.

TABLE 1. EODS algorithms to compute the centroid end points of an IT2 FS [23]

Step	EODS algorithms to compute $l_{\widetilde{B}_\alpha}$
1	Initialize $m=2$, $n=N-1$ and compute $S_l=(y_m-y_1)\overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_1)$, $S_r=(y_N-y_n)\underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_N)$, $F_l=\underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_N)$, $F_r=\overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_1)$
2	If $m=n$, go to step 4
3	If $S_l>S_r$, then $F_l=F_l+\underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_n)$, $n=n-1$, $S_r=S_r+F_l(y_{n+1}-y_n)$; otherwise $F_r=F_r+\overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_m)$, $m=m+1$, $S_l=S_l+F_r(y_m-y_{m-1})$, return to step 2 to check
4	If $S_l\le S_r$, then $L=m$, $F_r=F_r+\overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_m)$; otherwise $L=m-1$, $F_l=F_l+\underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_m)$
5	$l_{\widetilde{B}_\alpha}=y_m+\dfrac{S_r-S_l}{F_r+F_l}$
Step	EODS algorithms to compute $r_{\widetilde{B}_\alpha}$
1	Initialize $m=2$, $n=N-1$ and compute $S_l=(y_m-y_1)\underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_1)$, $S_r=(y_N-y_n)\overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_N)$, $F_l=\underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_1)$, $F_r=\overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_N)$
2	If $m=n$, go to step 4
3	If $S_l>S_r$, then $F_r=F_r+\overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_n)$, $n=n-1$, $S_r=S_r+F_r(y_{n+1}-y_n)$; otherwise $F_l=F_l+\underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_m)$, $m=m+1$, $S_l=S_l+F_l(y_m-y_{m-1})$, return to step 2 to check
4	If $S_l\le S_r$, then $R=m$, $F_l=F_l+\underline{\mu}_{R_{\widetilde{B}_\alpha}}(y_m)$; otherwise $R=m-1$, $F_r=F_r+\overline{\mu}_{R_{\widetilde{B}_\alpha}}(y_m)$
5	$r_{\widetilde{B}_\alpha}=y_m+\dfrac{S_r-S_l}{F_r+F_l}$

4. **Simulations.** In this section, two computer simulation examples are used to illustrate the performances of EODS algorithms. Before performing the TR, the FOUs of centroid output of GT2 FLSs and their corresponding secondary MFs are supposed to be known according to merging or weighing the fuzzy rules in terms of the inference engine [26]. The primary variable of the centroid output set is uniformly sampled, where the difference between adjacent points is $x_{i+1}-x_i=0.05$. In the simulations, the α is decomposed into Δ effective values as $\alpha=0,1/\Delta,\ldots,(\Delta-1)/\Delta,1$. In addition, we choose the value of Δ to range from 1 to 100. For the first example, the whole FOU is composed of piecewise linear functions [15,24,25,27], and the corresponding secondary MFs (vertical slices) are trapezoidal MFs. For the second example, the whole FOU is composed of piecewise linear functions and Gaussian MFs [27,29-31], and the corresponding vertical slices are triangle MFs.

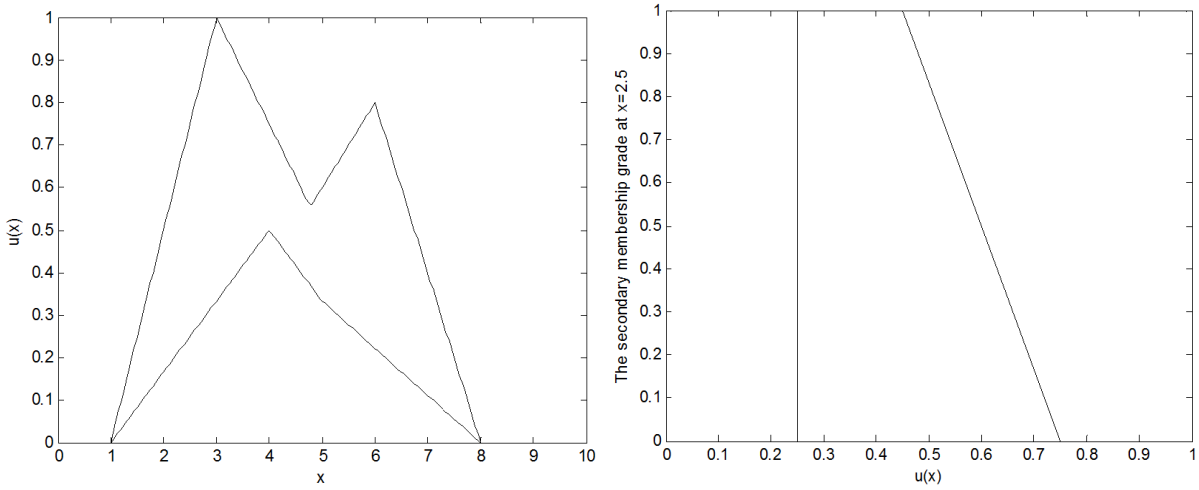
Example 4.1. *Piecewise linear functions with trapezoidal vertical slices.*

As shown in Figure 2, the upper bound of the FOU is composed of the maximum of two piecewise linear functions, i.e.,

$$u_1(x)=\begin{cases}\dfrac{x-1}{2}, & 1\le x\le 3\\ \dfrac{7-x}{4}, & 3<x\le 7\\ 0, & \textit{otherwise}\end{cases} \tag{43}$$

and

$$u_2(x) = \begin{cases} \dfrac{x-2}{5}, & 2 \le x \le 6 \\ \dfrac{16-2x}{5}, & 6 < x \le 8 \\ 0, & \textit{otherwise} \end{cases} \tag{44}$$



(a) (b)

FIGURE 2. (a) FOU of example one, and (b) its corresponding vertical slice at $x = 2.5$

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

$$u_3(x) = \begin{cases} \dfrac{x-1}{6}, & 1 \le x \le 4 \\ \dfrac{7-x}{6}, & 4 < x \le 7 \\ 0, & \textit{otherwise} \end{cases} \tag{45}$$

and

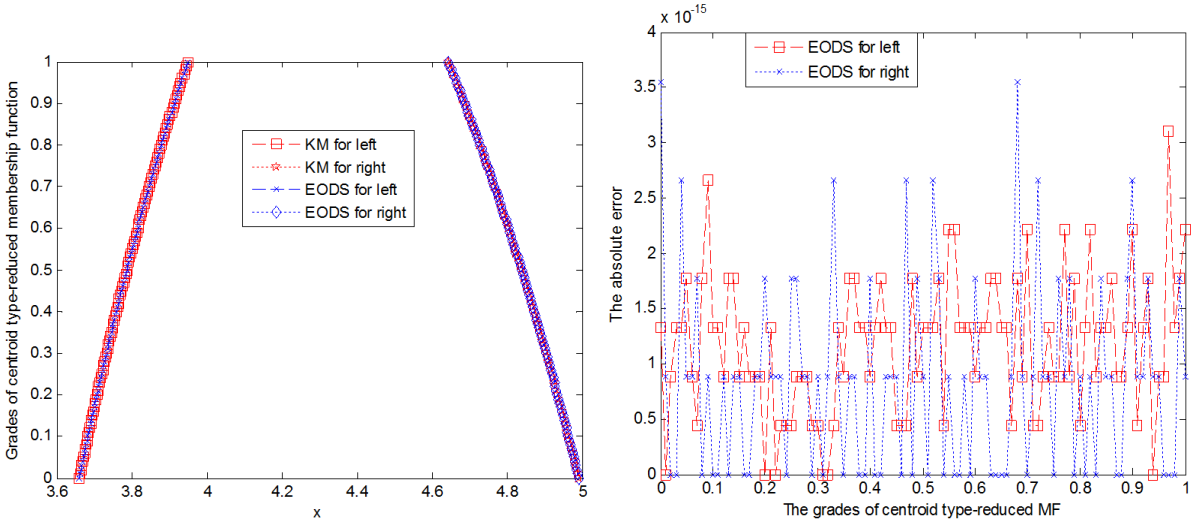
$$u_4(x) = \begin{cases} \dfrac{x-3}{6}, & 3 \le x \le 5 \\ \dfrac{8-x}{9}, & 5 < x \le 8 \\ 0, & \textit{otherwise} \end{cases} \tag{46}$$

For any x, the corresponding vertical slice is trapezoidal MF, whose top left and right end points are defined as

$$L(x) = \underline{u}(x) + 0.6w\left(\overline{u}(x) - \underline{u}(x)\right), \quad R(x) = \overline{u}(x) - 0.6(1-w)\left(\overline{u}(x) - \underline{u}(x)\right) \tag{47}$$

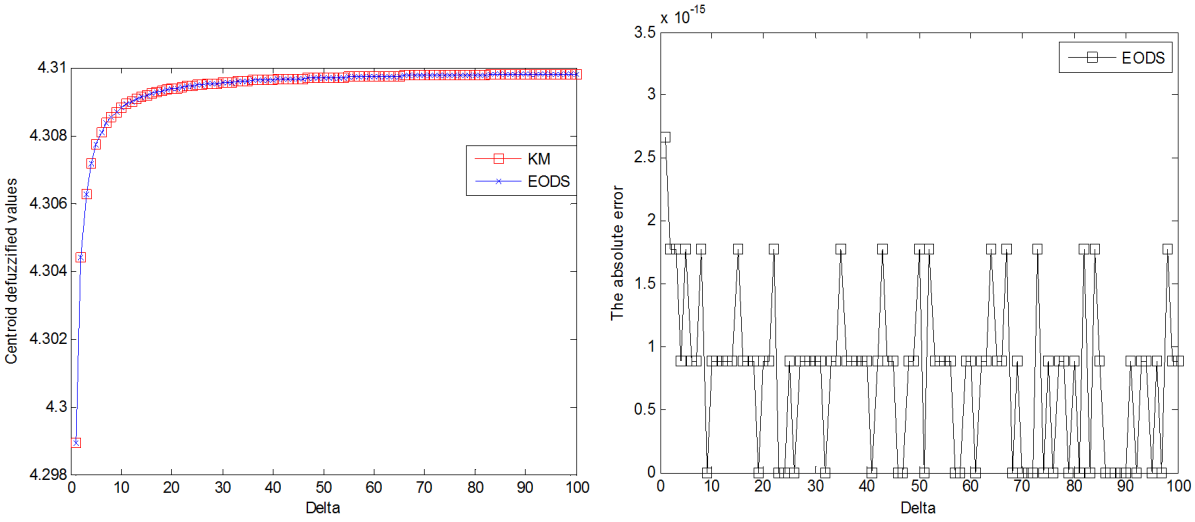
where $\overline{u}(x)$ and $\underline{u}(x)$ are the upper and lower bounds of the primary MF, respectively. Here we choose $w = 0$ for the test.

Then we study the performances of the EODS algorithms. For $\Delta = 100$, the type-reduced T1 FSs for KM and EODS algorithms are give in Figure 3(a), and the absolute error of centroid type-reduced T1 FS between the KM algorithms and the EODS algorithms is shown in Figure 3(b), where the grades of centroid type-reduced MF are chosen as the independent variable, and the absolute error $|C_{KM} - C_{EODS}|$ is the dependent variable.



(a) (b)

FIGURE 3. (a) The centroid type-reduced T1 FSs for KM and EODS algorithms, and (b) the absolute error of centroid type-reduced T1 FS between the KM and EODS algorithms

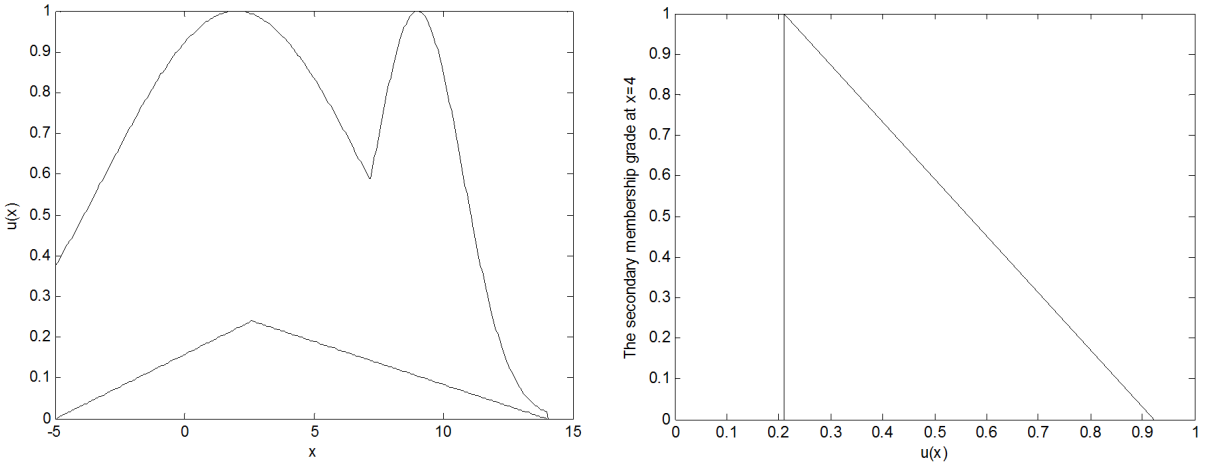


(a) (b)

FIGURE 4. (a) The centroid defuzzified values for KM and EODS algorithms, and (b) the absolute error of centroid defuzzified values between the KM and EODS algorithms

Moreover, the centroid defuzzified values for KM and EODS algorithms are shown in Figure 4(a), and the absolute error of centroid defuzzified values between the KM algorithms and EODS algorithms is shown in Figure 4(b), where we choose the effective number of α-planes (Δ) as the independent variable, and the absolute error $|y_{KM} - y_{EODS}|$ as the dependent variable.

Example 4.2. *Hybrid functions with triangle vertical slices.*



(a) (b)

FIGURE 5. (a) FOU of Example 4.2, and (b) its corresponding vertical slice at $x = 4$

See Figure 5, the upper bound of the FOU is the piecewise Gaussian function, i.e.,

$$u_1(x) = \begin{cases} \exp\left[-\frac{1}{2}\left(\frac{x-2}{5}\right)^2\right], & -5 \le x \le 7.185 \\ \exp\left[-\frac{1}{2}\left(\frac{x-9}{1.75}\right)^2\right], & 7.185 < x \le 14 \end{cases} \tag{48}$$

The lower bound of the FOU is the piecewise linear function, i.e.,

$$u_2(x) = \begin{cases} \frac{0.6(x+5)}{19}, & -5 \le x \le 2.6 \\ \frac{0.4(14-x)}{19}, & 2.6 < x \le 14 \end{cases} \tag{49}$$

For any x, the corresponding vertical slice is triangle MF, whose apex is defined as

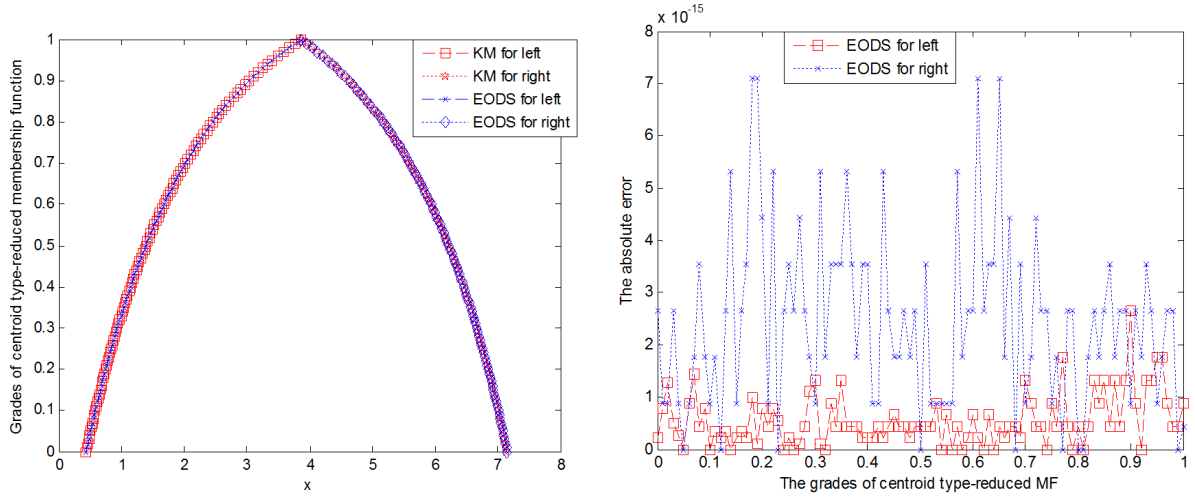
$$Apex = \underline{u}(x) + w\left(\overline{u}(x) - \underline{u}(x)\right) \tag{50}$$

where $\overline{u}(x)$ and $\underline{u}(x)$ are the upper and lower bounds of the primary MF, respectively. In this test, we choose $w = 0$.

For $\Delta = 100$, the type-reduced T1 FSs for KM and EODS algorithms are give in Figure 6(a), and the absolute error of centroid type-reduced T1 FS between the KM and EODS algorithms is shown in Figure 6(b).

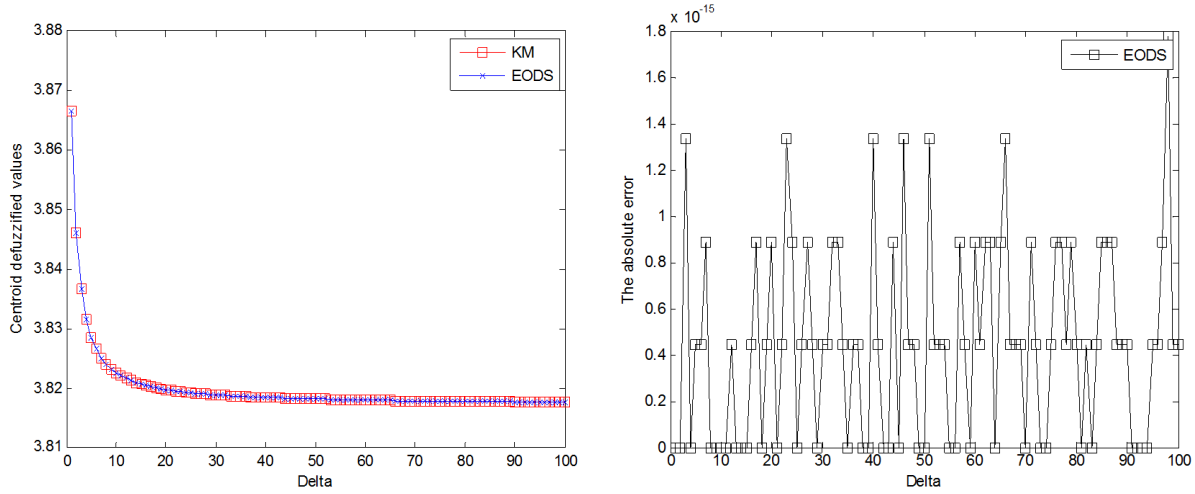
Moreover, the centroid defuzzified values for KM and EODS algorithms are shown in Figure 7(a), and the absolute error of centroid defuzzified values between the KM and EODS algorithms is shown in Figure 7(b).

Observing from the above two examples, we can obtain that: (1) for the KM and EODS type-reduction algorithms, the shape of the type-reduced T1 FS depends on the shape of the secondary MF, e.g., the first test with trapezoid secondary MF derived a trapezoid-looking T1 MF, and the second test with triangular secondary MF derived a triangular-looking T1 MF; (2) as the number of effective α-planes increases, the centroid defuzzified values all converge to certain real values, and the real values may be viewed as the defuzzified values of continuous T2 FSs; (3) from Figures 3 and 4 and 6 and 7, we can find that both the centroid type-reduced MFs and the defuzzified values computed



(a) (b)

FIGURE 6. (a) The centroid type-reduced T1 FSs for KM and EODS algorithms, and (b) the absolute error of centroid type-reduced T1 FS between the KM and EODS algorithms



(a) (b)

FIGURE 7. (a) The centroid defuzzified values for KM and EODS algorithms, and (b) the absolute error of centroid defuzzified values between the KM and EODS algorithms

by two types of algorithms are almost the same, for unit of absolute errors between them is 10^{-15}; that is to say, the EODS algorithms have the same calculation accuracy as the KM algorithms for performing the centroid TR and defuzzification of GT2 FLSs.

Next, we study the specific computation times for two types of algorithms. Here we choose the simulation platform as the Microsoft Windows XP Professional system, and the dual-core CPU dell desktop with E5300@2.6GHz and 2.00GB memory. All the algorithms are programmed by the Matlab 2013a. Table 2 and Table 3 provide the total time consuming of two types of algorithms for computing the centroid type-reduced MFs and the defuzzified values, where the unit of time is second (s), the third line in tables represents the average of two examples, and the fourth column in tables denotes the relative

TABLE 2. The total computation times of two types of algorithms for computing the centroid type-reduced MFs

Num	KM	EODS	TRR (%)
Example 4.1	0.086835	0.001783	97.45
Example 4.2	0.204717	0.002577	98.74
Average	0.145776	0.00218	98.10

TABLE 3. The total computation times of two types of algorithms for computing the centroid defuzzified values

Num	KM	EODS	TRR (%)
Example 4.1	4.347709	0.105459	97.57
Example 4.2	10.484264	0.157093	98.50
Average	7.415987	0.131276	98.04

time reducing rate (TRR) for the EODS algorithms compared with the KM algorithms, and it is defined as:

$$TRREODS = (t_0 - t_n)/t_0 \times 100\% \tag{51}$$

where t_0 is the computation time for the KM algorithms, and t_n is the computation time for the EODS algorithms.

Observing from the above Tables 2 and 3, we can get the conclusions: (1) when computing the centroid type-reduced MFs in two examples, compared with the KM algorithms, the EODS algorithms can obtain the largest time reducing rate as 98.74% and the average time reducing rate as 98.10%; (2) when computing the centroid defuzzified values, the EODS algorithms can obtain the largest time reducing rate as 98.50% and the average time reducing rate as 98.04% compared with the KM algorithms.

The above simulation analyses about computing the centroid type-reduced MFs and the centroid defuzzified values of GT2 FLSs show that the EODS algorithms can be an effective approach for performing the centroid type-reduction and defuzzification of GT2 FLSs. Although the computation results of KM and EODS algorithms are almost the same, the computation times for the EODS algorithms are much less than the KM algorithms; therefore, the former can greatly reduce the computation cost of GT2 FLSs.

5. **Conclusions.** Based on the α-planes representation of GT2 FSs, this paper extends the EODS algorithms to perform the centroid type-reduction of GT2 FLSs by combining the process of fuzzy reasoning. Two numerical simulation examples are provided to illustrate the performances of EODSs algorithms for computing the centroid type-reduced MFs and the centroid defuzzified values and their corresponding computation times. No matter we choose a certain number of effective α-planes or the number of effective α-planes varies in a range, the computation results of centroid type-reduced MFs and centroid defuzzified values for two types of algorithms are very similar; however, the computation times of EODS algorithms are significantly less than the KM algorithms, which provide the potential value for designers and adopters of GT2 FLSs.

In the future work, we will study the center of sets type-reduction of GT2 FLSs [18,27,32,33,39,40], and GT2 FLSs design and applications based on intelligent optimization algorithms [16,34-38] and so on. Future studies will focus on T2 FLSs design and applications based on [16-19,27,31,38,42] and this paper.

Acknowledgment. The paper is supported by the National Natural Science Foundation of China (No. 61773188, No. 61803189), Liaoning Province Natural Science Foundation Guidance Project (No. 20180550056) and Fundamental Research Funds for Liaoning's Universities (No. JL201615410). The author would like to thank professor Jerry Mendel, who has given some valuable suggestions.

REFERENCES

[1] 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.

[2] Y. Chen, D. Wang and S. Tong, Forecasting studies by designing Mamdani interval type-2 fuzzy logic systems, *Neurocomputing*, vol.174, Part B, pp.1133-1146, 2016.

[3] M. H. F. Zarandi, B. Rezaee, I. B. Turksen and E. Neshat, A type-2 fuzzy rule-based expert system model for stock price analysis, *Expert Systems with Applications*, vol.36, no.1, pp.139-154, 2009.

[4] 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.

[5] 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.

[6] G. M. Méndez and M. D. L. A. Hernández, Hybrid learning mechanism for interval A2-C1 type-2 non-singleton type-2 Takagi-Sugeno-Kang fuzzy logic systems, *Information Sciences*, vol.220, no.1, pp.149-169, 2013.

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

[8] 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.

[9] R. Martinez, O. Castillo and L. T. Aguilar, Optimization of interval type-2 fuzzy logic controller for a perturbed autonomous wheeled mobile robots using genetic algorithm, *Information Sciences*, vol.179, no.3, pp.2158-2174, 2009.

[10] 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.

[11] 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.

[12] P. Melin, O. Mendoza and O. Castillo, An improved method for edge detection based on interval type-2 fuzzy logic, *Expert Systems with Applications*, vol.37, no.12, pp.8527-8535, 2010.

[13] D. Wu and J. M. Mendel, Uncertainty measures for interval type-2 fuzzy sets, *Information Sciences*, vol.177, no.23, pp.5378-5393, 2007.

[14] 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.

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

[16] 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.

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

[18] 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.

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

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

[22] H. Z. Hu, Y. Wang and Y. L. Cai, Advantages of the enhanced opposite direction searching algorithm for computing the centroid of an interval type-2 fuzzy set, *Asian Journal of Control*, vol.14, no.5, pp.1422-1430, 2012.
[23] D. Wu, Approaches for reducing the computational cost of interval type-2 fuzzy logic systems: Overview and comparisons, *IEEE Trans. Fuzzy Systems*, vol.21, no.1, pp.80-99, 2013.
[24] 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.
[25] 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.
[26] 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.
[27] 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.
[28] N. N. Karnik and J. M. Mendel, Centroid of a type-2 fuzzy set, *Information Sciences*, vol.132, no.1, pp.195-220, 2001.
[29] 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.
[30] 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.
[31] X. Liu, J. M. Mendel and D. Wu, Study on enhanced Karnik-Mendel algorithms: Initialization explanations and computation improvements, *Information Sciences*, vol.184, no.1, pp.75-91, 2012.
[32] 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.
[33] M. A. Khanesar, A. Jalalian, O. Kaynak and H. Gao, Improving the speed of center of sets type reduction in interval type-2 fuzzy systems by eliminating the need for sorting, *IEEE Trans. Fuzzy Systems*, vol.25, no.5, pp.1193-1026, 2017.
[34] 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.
[35] 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.
[36] Y. Chen, D. Z. 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.
[37] 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.
[38] J. M. Mendel, H. Hagras, W. W. Tan, W. W. Melek and H. Ying, *Introduction to Type-2 Fuzzy Logic Control: Theory and Applications*, Wiley, Hoboken, NJ, USA, 2014.
[39] 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.
[40] T. Kumbasar, Revisiting Karnik-Mendel algorithms in the framework of linear fractional programming, *International Journal of Approximate Reasoning*, vol.82, pp.1-21, 2017.
[41] 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, Part C, pp.257-274, 2016.
[42] H. Uesu, Type-2 fuzzy contingency table and similarity indices, *ICIC Express Letters*, vol.12, no.8, pp.791-798, 2018.