Proceedings of the Federated Conference on
Computer Science and Information Systems pp. 49–54
ISBN 978-83-60810-51-4

An approximation method for Type Reduction of an Interval Type-2 fuzzy set based on α-cuts

Juan Carlos Figueroa García

Mathematical Modeling Applied to Industry (MMAI) and
Laboratory for Automation, Microelectronics and Computational Intelligence (LAMIC) members.
Universidad Distrital Francisco José de Caldas, Bogotá - Colombia.

***Abstract*—This paper shows a proposal for Type-reduction of an Interval Type-2 fuzzy set composed from α-cuts done over its primary membership functions. The definition of available Type-reduction methods for Interval Type-2 fuzzy sets are based on an homogeneous subdivision of the universe of discourse, so we propose an approximation algorithm for Type-reduction of an Interval type-2 fuzzy set through its primary α-cuts.**

Some definitions about the α-cut of a Type-2 fuzzy set are provided and used for computing the centroid of an Interval Type-2 fuzzy set through a mapping of its membership function, instead of its universe of discourse.

I. Introduction and Motivation

TYPE Reduction of a Type-2 fuzzy set (T2FS) is a key aspect of fuzzy inference and its computation is an important step in practical applications. The computation of the centroid of an Interval Type-2 fuzzy set (IT2FS) has well known results (See [1], [2], [3], [4], [5], [6], [7], [8], [9], and Melgarejo [10], [11], [12] and [13]) which are based on a mapping of the universe of discourse of x, $x \in X$ as the common way for computing a Fuzzy Logic System (FLS).

Some applications of Type-1 Fuzzy Relational Equations (T1FRE) are based on a mapping of the membership space through α-cuts (See Chanas and Kamburoswk [14], Shih-Pin Chen [15], and Chanas, Dubois and Zieliński [16]), so its extension to IT2FS is an interesting approach. Some definitions about the concept of the α-cut of an IT2FS are referred and discussed to define the representation of an IT2FS through its α-cuts, and finally an approximation method for Type-reduction of an IT2FS through this representation is presented.

This paper focuses on the use of Interval Type-2 L-R fuzzy sets applied to FRE's for computing its centroid. This does not imply that the proposed method cannot be applied to other kind of IT2FS, but in this paper only L-R IT2FS are addressed. In addition, an optimization example which uses L-R IT2FS and α-cuts is solved by the proposed method and its results are discussed.

This paper is divided into seven sections. Section 1 introduces the problem. In Section 2 some basic definitions about IT2FS are given; in Section 3, the Enhanced Karnik-Mendel (EKM) algorithm for Type-reduction is presented, some improvements are referred, and a discussion about the way how the EKM algorithm operates through a mapping of X, is presented. Section 4 presents the decomposition of an IT2FS into α-cuts, and some key aspects for computing its centroid. In Section 5, a proposal for computing the approximated centroid of an IT2FS based on its α-cuts is presented. In Section 6 an application example is presented and the Section 7 presents the concluding remarks of the study.

Juan Carlos Figueroa is Assistant professor of the Engineering Dept. of the Universidad Francisco José de Caldas, Bogotá - Colombia, e-mail: jcfigueroag@udistrital.edu.co

II. Basic definitions of IT2FS

A Type-2 fuzzy set is a collection of infinite Type-1 fuzzy sets into a single fuzzy set. It is defined by two membership functions: The first one defines the degree of membership of the universe of discourse X and the second one weights each one of the first Type-1 fuzzy sets.

According to Jerry Mendel [1], [2], [3], [4], [5], [6], [7], [8], [9], and Melgarejo [10], [11], [12] some basic definitions of Interval Type-2 fuzzy sets are given next:

Definition 2.1 (Type-2 fuzzy set): A Type-2 fuzzy set, $\tilde{A}$, is described as the following ordered pairs:

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) \,|\, x \in X\} \tag{1}$$

where $\mu_{\tilde{A}}(x))$ is composed by an infinite amount of Type-1 fuzzy sets in two ways: *Primary* fuzzy sets J_x weighted by *Secondary* fuzzy sets $f_x(u)$. In other words

$$\tilde{A} = \{((x, u), J_x, f_x(u)) \mid x \in X; u \in [0, 1]\} \tag{2}$$

Therefore, the FOU evolves all the embedded J_x weighted by the secondary membership function $f_x(u)/u$. These Type-2 fuzzy sets are known as *Generalized* Type-2 fuzzy sets, *(T2FS)*, since $f_x(u)/u$ is a Type-1 membership function. Now, an *Interval Type-2* fuzzy set, *(IT2FS)*, is a simplification of T2FS in the sense that the secondary membership function is assumed to be 1, as follows

Definition 2.2 (Interval Type-2 fuzzy set): An Interval Type-2 fuzzy set, $\tilde{A}$, is described as:

$$\tilde{A} = \int_{x\in X}\int_{u\in J_x} 1/(x,u) = \int_{x\in X}\left[\int_{u\in J_x} 1/u\right]\Big/ x, \tag{3}$$

where x is the *primary variable*, J_x is the *primary membership function* associated to x, u is the *secondary variable* and $\int_{u\in J_x} 1/u$ is the *secondary membership function*. Uncertainty about $\tilde{A}$ is conveyed by the union of all of J_x into the *Footprint Of Uncertainty* of $\tilde{A}$ [FOU($\tilde{A}$)], i.e.

$$\mathrm{FOU}(\tilde{A}) = \bigcup_{x \in X} J_x \tag{4}$$

A FOU is bounded by two membership functions: An *Upper* membership function *(UMF)* $\bar{\mu}_{\tilde{A}}(x)$ and a *Lower* membership function *(LMF)* $\underline{\mu}_{\tilde{A}}(x)$.

For discrete universes of discourse $X = \{x_1, x_2, \cdots, x_N\}$ and discrete J_x, an *Embedded T1 FS*, A_e has N elements, one each from $J_{x_1}, J_{x_2}, \cdots, J_{x_N}$, namely $u_1, u_2, \cdots, u_N$, e.g.

$$A_e = \sum_{i=1}^{N} u_i/x_i \quad u_i \in J_{x_i} \subseteq [0,1] \tag{5}$$

Its graphical representation is shown in the Figure 1

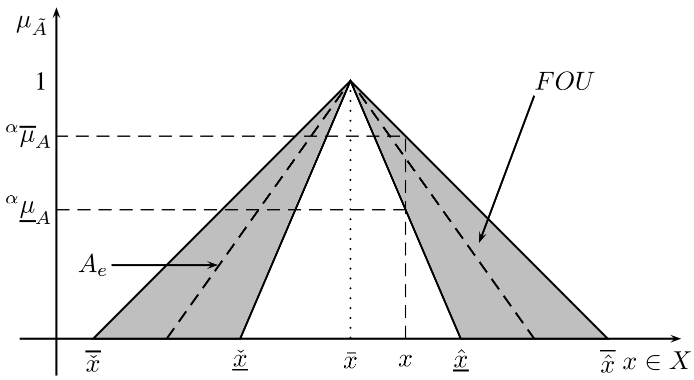

Fig. 1. Interval Type-2 Fuzzy set $\tilde{a}$ with Uncertain $\triangleleft = \triangleright$

Here, $\tilde{a}$ is a Type-2 fuzzy set, the universe of discourse is the set $x \in X$, the *support* of $\tilde{A}$, *supp*($\breve{A}$) is the interval $x \in [\bar{\tilde{x}}, \bar{\hat{x}}]$ and $\mu_{\tilde{A}}$ is a linear function with parameters $\bar{\tilde{x}}, \bar{\hat{x}}, \underline{\check{x}}, \underline{\hat{x}}$ and $\bar{x}$. ${}^{\alpha}\,\bar{\mu}_A(x)$ is the degree of membership an specific value x has regarding the upper fuzzy set $\bar{A}$ and ${}^{\alpha}\underline{\mu}_A(x)$ is the degree of membership an specific value x has regarding the lower fuzzy set $\underline{A}$. FOU is the *Footprint of Uncertainty* of the Type-2 fuzzy set and A_e is a Type-1 fuzzy set embedded in its FOU.

III. The Enhanced Karnik-Mendel (EKM) algorithm

Wu and Mendel in [17] and [18] defined the Enhanced Karnik-Mendel algorithms for Type-reduction of an IT2FS. The EKM algorithm for computing y_l is as follows

1) Sort $\underline{x}_i$ $(i = 1, 2, ..., N)$ in increasing order, and call the sorted $\underline{x}_i$ by the same name, but now, $\underline{x}_1 \leqslant \underline{x}_2 \leqslant \cdots \leqslant \underline{x}_N$. Match the weights w_i with their respective $\underline{x}_i$ and renumber them so that their index corresponds to the renumbered $\underline{x}_i$.
2) Set $k = [N/2.4]$ (the nearest integer to $N/2.4$), and compute

$$a = \sum_{i=1}^{k} \underline{x}_i \overline{w}_i + \sum_{i=k+1}^{N} \underline{x}_i \underline{w}_i \tag{6}$$

$$b = \sum_{i=1}^{k} \overline{w}_i + \sum_{i=k+1}^{N} \underline{w}_i \tag{7}$$

$$y = \frac{a}{b} \tag{8}$$

3) Find switch point k' $\quad (1 \leqslant k \leqslant N-1)$ such that

$$\underline{x}_{k'} \leqslant y \leqslant \underline{x}_{k'+1}$$

4) Check if $k' = k$. If yes, stop, set $y_l = y$, and call k L. If no, continue.
5) Compute $s = sign(k' - k)$, and

$$a' = a + s \sum_{i=\min(k',k)+1}^{\max(k',k)} \underline{x}_i(\overline{w}_i - \underline{w}_i) \tag{9}$$

$$b' = b + s \sum_{i=\min(k',k)+1}^{\max(k',k)} (\overline{w}_i - \underline{w}_i) \tag{10}$$

$$y' = \frac{a'}{b'} \tag{11}$$

6) Set $y = y'$, $a = a'$, $b = b'$, and $k = k'$. Go to step 3).

The EKM algorithm for computing y_r is as follows

1) Sort $\overline{x}_i$ $(i = 1, 2, ..., N)$ in increasing order, and call the sorted $\overline{x}_i$ by the same name, but now, $\overline{x}_1 \leqslant \overline{x}_2 \leqslant \cdots \leqslant \overline{x}_N$. Match the weights w_i with their respective $\overline{x}_i$ and renumber them so that their index corresponds to the renumbered $\overline{x}_i$.
2) Set $k = [N/1.7]$ (the nearest integer to $N/2.4$), and compute

$$a = \sum_{i=1}^{k} \overline{x}_i \underline{w}_i + \sum_{i=k+1}^{N} \overline{x}_i \overline{w}_i \tag{12}$$

$$b = \sum_{i=1}^{k} \underline{w}_i + \sum_{i=k+1}^{N} \overline{w}_i \tag{13}$$

$$y = \frac{a}{b} \tag{14}$$

3) Find switch point k' $\quad (1 \leqslant k \leqslant N-1)$ such that

$$\overline{x}_{k'} \leqslant y \leqslant \overline{x}_{k'+1}$$

4) Check if $k' = k$. If yes, stop, set $y_r = y$, and call k R. If no, continue.
5) Compute $s = sign(k' - k)$, and

$$a' = a - s \sum_{i=\min(k',k)+1}^{\max(k',k)} \overline{x}_i(\overline{w}_i - \underline{w}_i) \tag{15}$$

$$b' = b - s \sum_{i=\min(k',k)+1}^{\max(k',k)} (\overline{w}_i - \underline{w}_i) \tag{16}$$

$$y' = \frac{a'}{b'} \tag{17}$$

6) Set $y = y'$, $a = a'$, $b = b'$, and $k = k'$. Go to step 3).

A. Mapping Y

Some improvements of this widely used algorithm was proposed by Miguel Melgarejo in [11], [13]. These Type-reducers are based on a mapping of Y, this means that $\overline{w}_i$ and $\underline{w}_i$ are obtained from the bounds of $\tilde{Y}$ as its *lower* and *upper* membership degrees $\overline{\mu}(x_i)$ and $\underline{\mu}(x_i)$, respectively.

Some common facts about the computation of $\tilde{Y}$ are:

1) $\tilde{Y}$ is obtained from Fuzzy Logic Systems (FLS), usually composed by a rule base.
2) The computation of $\tilde{Y}$ comes from a discretization of the universe of discourse of the antecedents of the FLS.
3) The upper and membership functions of $\tilde{Y}$ are obtained from fuzzy relations among the antecedents, so its membership degrees depends on the mapping done in the antecedents.
4) The α-cut[1] approach to compute $\tilde{Y}$ is not widely used.

Some applications of IT2FS are based on α-cuts instead of a mapping of each antecedent e.g. fuzzy relational equations, decision making, fuzzy optimization, among others. Its extension to IT2FS is an upcoming field to be treated, since decision making under uncertainty is an open problem.

The decomposition of IT2FS into its α-cuts for composing a response surface and the computation of the resultant centroid is an emergent problem for decision making, so the use of the α-cuts appears as a new way to map IT2FS. In the following Section, some useful definitions are presented to illustrate how the α-cuts change the behavior of the EKM algorithm.

Remark 3.1: Although the EKM algorithm is defined for non-α convex IT2FS, this proposal is intended for α convex IT2FS e.g. L-R fuzzy sets. Its use in non-α convex IT2FS is an extension which is not considered in this work.

In following sections, some basic definitions about α-cuts are presented in order to illustrate the problem.

IV. Computing α-cuts of an IT2FS

The α-cut of an IT2FS is defined by the bounds where all embedded T1FS into the FOU fulfill the condition of a Type-1 *α-cut* on its primary membership function. Mendel in [19], [20] and Liu in [21] and [22] defined a secondary *α-cut* called *α-plane* which is specially useful to compute the centroid of a Type- fuzzy set. According to Figueroa [23], an α-cut of an IT2FS is

Definition 4.1 (Primary α-cut of an IT2FS): The *Primary α-cut* of an Interval Type-2 fuzzy set ${}^{\alpha}\tilde{A}$ is the union of all Type-1 fuzzy sets which fulfill the condition ${}^{\alpha}J_x = \{x|\mu_A(x) \geqslant \alpha\}$ on its primary membership function.

Here, ${}^{\alpha}J_x = \{x|\mu_A(x) \geqslant \alpha\}$ is the Type-1 α-cut on the primary membership function of $\tilde{A}$, defined as follows

$$ {}^{\alpha}\tilde{A} = \int_{x\in X} \left[\int_{u\in J_x \geqslant \alpha} f_x(u)/u \right] \Big/ x \; ; \; \alpha \subseteq [0,1] \tag{18} $$

$$ {}^{\alpha}\tilde{A} = \int_{x\in X} \int_{u\in f_x} \{(x,u)|J_x \geqslant \alpha\} \; ; \; \alpha, f_x \subseteq [0,1] \tag{19} $$

[1]An α-cut over $\tilde{Y}$ is the α-cut made over $\overline{\mu}_{\tilde{Y}}$ and $\underline{\mu}_{\tilde{Y}}$.

An alternative representation is

$$ {}^{\alpha}\tilde{A} = \bigcup_{x\in J_x} \{\mu_{\tilde{A}}(x,u)|J_x \geqslant \alpha\} \tag{20} $$

$$ {}^{\alpha}\mathit{FOU}(\tilde{A}) = \bigcup_{x\in X} \{J_x \geqslant \alpha\} \tag{21} $$

In this way, the crisp bounds of the primary α-*cut* of a Type-2 fuzzy set are defined as the α^I-cut, $f_x, \alpha \subseteq [0,1]$ (See Figueroa in [24]):

$$ {}^{\alpha^I}\tilde{A} = \int_{x\in X} \int_{u\in f_x} \{(x,u)|J_x = \alpha\}\} \tag{22} $$

$$ {}^{\alpha^I}\tilde{A} = \bigcup_{x\in X} \{\mu_{\tilde{b}}(x,u)|J_x = \alpha\} \tag{23} $$

This primary α-cut of a Type-2 fuzzy set can be seen as a *cut* of the FOU of the set because it encloses all values of $f_x(u)$ which are contained on the interval $[\,\overline{\mu}_{\tilde{A}}(x) = \alpha, \underline{\mu}_{\tilde{A}}(x) = \alpha\,]$, so all these values are cuts of all primary memberships of $\breve{A}$.

A graphical representation of ${}^{\alpha}\tilde{A}$ and ${}^{\alpha^I}\tilde{A}$ defined as IT2FS is presented in Figure 2.

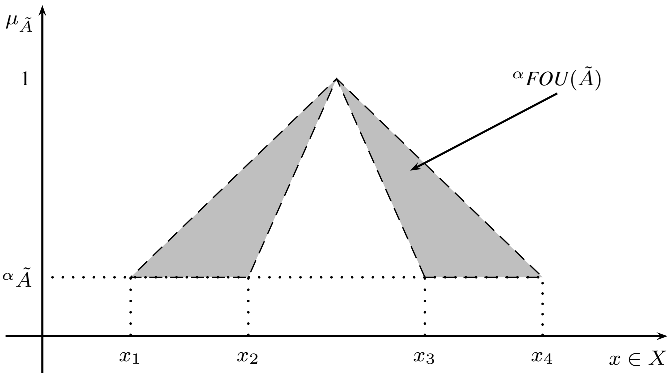

Fig. 2. ${}^{\alpha}\tilde{A}$ of an Interval Type-2 Fuzzy set $\tilde{A}$.

In Figure 2, the dashed line encloses the ${}^{\alpha}\tilde{A}$, the shaded region is the ${}^{\alpha}\tilde{A}$ of the FOU and the pointed line is the ${}^{\alpha^I}\tilde{A}$. In other words, the bounds where $[\,\overline{\mu}_{\tilde{A}}(x) = \alpha, \underline{\mu}_{\tilde{A}}(x) = \alpha\,]$. In other words, ${}^{\alpha^I}\mu_{\tilde{A}} \in [\inf_x\{{}^{\alpha}\mu_{\tilde{A}}(x,u)\}, \sup_x\{{}^{\alpha}\mu_{\tilde{A}}(x,u)\}]$, which is equivalent to say ${}^{\alpha^I}\mu_{\tilde{A}} \in [[x_1,x_2],[x_3,x_4]]$

If we map $\tilde{A}$ through its α-cuts, there is a high probability of not getting an uniform mapping of X. This means that an α-cut made over $\overline{\mu}_{\tilde{A}}$ has no an image over $\underline{\mu}_{\tilde{A}}$, and viceversa.

To illustrate this problem, Figure 3 shows the difference of mapping $\tilde{A}$ through α-cuts instead of X.

Figure 3 is a discretization of Figure 1. It is composed by α-cuts, so the accuracy level depends on the amount of α -cuts done over $\tilde{A}$. A key aspect about this representation is that there is no any guarantee that an α-cut made over $\overline{\mu}_{\tilde{A}}$ has an image over $\underline{\mu}_{\tilde{A}}$ and viceversa. This leads to the following remark.

Remark 4.1: In Figure 3, note that x_1 has only image over $\underline{\mu}_{\tilde{A}}$, but it has no image over $\underline{\mu}_{\tilde{A}}$. This means that ${}^{0.2}\underline{\mu}_{\tilde{A}}$ gets x_1, but there is no any ${}^{\alpha}\overline{\mu}_{\tilde{A}}$. For x_2, the analysis is similar.

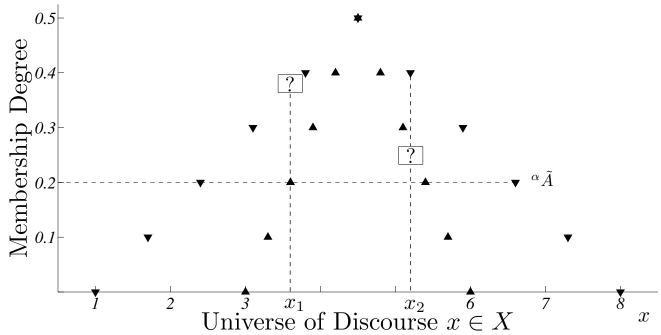

Fig. 3. Some α-cuts made on $\tilde{A}$

This is an important problem for the EKM and IASCO Type-reduction algorithms since they need a complete mapping of $\tilde{A}$ through X in order to start $\underline{x}_i$ and $\overline{x}_i$. So a solution for computing the absent values of $\underline{w}_i$ and $\overline{w}_i$ is needed.

This representation is specially useful in some punctual applications as decision making, fuzzy optimization, fuzzy regression, etc. As usual, its usage goes along with other classical techniques as relational equations, linear programming, etc, so each application has different computational costs.

V. Approximation method for computing the centroid of an IT2FS based on α-cuts

The EKM and IASCO Type-reduction algorithms are based on a mapping of X and then compute $\underline{w}_i$ and $\overline{w}_i$ for each of the two stages or the algorithm: y_l and y_r. When $\tilde{Y}$ is composed through α-cuts as shown in Figure 3, there is a high probability to have an incomplete map of $\underline{w}_i$ and $\overline{w}_i$, so we propose a simple strategy for computing the absent values of $\underline{w}_i$ and $\overline{w}_i$ through linear interpolation, as presented in Algorithm 1.

Therefore, from step 2 and forward, the IASCO and EKM algorithms can be applied in its original way.

Remark 5.1: If having absent values of x_{m_3} and/or x_{m_4} for a particular value of x_{n_1} and/or x_{n_2}, then the Algorithm 1 cannot compute $\underline{w}_{n_1}$ and/or $\underline{w}_{n_2}$. In this case, we recommend to assign zero to $\underline{w}_{n_1}$ and/or $\underline{w}_{n_2}$ since it implies that $\underline{\mu}_{\tilde{Y}} \approx 0$.

VI. Application example

In this section we present an application of the proposed method to a small optimization example, where all its technological coefficients are defined as IT2FS, so its optimal values can be computed from solving the problem for each α-cut. The obtained solution is also an IT2FS, but its type-reduced cannot be computed using the classical IASCO and EKM algorithms.

The following is the addressed example:

$$\max\ z = 3.5x_1 + 2.5x_2$$

$$s.t.$$

$$\tilde{A}_{11}x_1 + \tilde{A}_{12}x_2 \precsim 10$$

$$\tilde{A}_{21}x_1 + \tilde{A}_{22}x_2 \precsim 12$$

Procedure 1 Linear Interpolation

Require: $m, n \in [1, \cdots, N], \alpha_n \in [0,1]$
  Compute ${}^{\alpha_n}\overline{\mu}_{\tilde{Y}}$ for each n
  Compute ${}^{\alpha_n}\underline{\mu}_{\tilde{Y}}$ for each n
  for $n = 1 \to N$ **do**
    return Set $x_{n_1} = \inf_x\{{}^{\alpha_n}\overline{\mu}_{\tilde{Y}}\}$ and $\overline{w}_{n_1} = \alpha_n$
    return Set $x_{n_2} = \sup_x\{{}^{\alpha_n}\overline{\mu}_{\tilde{Y}}\}$ and $\overline{w}_{n_2} = \alpha_n$
    return Set $x_{n_3} = \inf_x\{{}^{\alpha_n}\underline{\mu}_{\tilde{Y}}\}$ and $\underline{w}_{n_3} = \alpha_n$
    return Set $x_{n_4} = \sup_x\{{}^{\alpha_n}\underline{\mu}_{\tilde{Y}}\}$ and $\underline{w}_{n_4} = \alpha_n$
  end for
  for $n = 1 \to N, m \in [1, \cdots, N]$ **do**
    For x_{n_1} find $x_{m_3} < x_{n_1} < x_{m+1_3}$
      set $\underline{w}_{n_1} = \frac{x_{n_1} - x_{m_3}}{x_{m+1_3} - x_{m_3}}(\underline{w}_{m+1_3} - \underline{w}_{m_3}) + \underline{w}_{m_3}$
    For x_{n_2} find $x_{m_4} < x_{n_2} < x_{m+1_4}$
      set $\underline{w}_{n_2} = \frac{x_{m+1_4} - x_{n_2}}{x_{m+1_4} - x_{m_4}}(\underline{w}_{m+1_4} - \underline{w}_{m_4}) + \underline{w}_{m+1_4}$
    For x_{n_3} find $x_{m_1} < x_{n_3} < x_{m+1_1}$
      set $\overline{w}_{n_3} = \frac{x_{n_3} - x_{m_1}}{x_{m+1_1} - x_{m_1}}(\overline{w}_{m+1_1} - \overline{w}_{m_1}) + \overline{w}_{m_1}$
    For x_{n_4} find $x_{m_2} < x_{n_4} < x_{m+1_2}$
      set $\overline{w}_{n_4} = \frac{x_{m+1_2} - x_{n_4}}{x_{m+1_2} - x_{m_2}}(\overline{w}_{m+1_2} - \overline{w}_{m_2}) + \overline{w}_{m+1_2}$
  end for
  Sort $x_{n_1}, x_{n_2}, x_{n_3}, x_{n_4}$ $(n = 1, 2, \cdots, N)$ in increasing order, create a vector of its corresponding weights $\overline{w}_{n_1}, \underline{w}_{n_1}, \overline{w}_{n_2}, \underline{w}_{n_2}, \overline{w}_{n_3}, \underline{w}_{n_3}, \overline{w}_{n_4}, \underline{w}_{n_4}$ and call the sorted as $x_i, \overline{w}_i, \underline{w}_i$, but now $x_1 \leqslant x_2 \leqslant \cdots \leqslant x_{4N}$
  return $x_i, \overline{w}_i, \underline{w}_i$ for each $i \in (1, \cdots, 4N)$.

where $x_j \in \mathbb{R}$, each $\tilde{A}_{ij}$ is an IT2FS and $\precsim$ is an IT2 fuzzy partial order. The shapes of each $\tilde{A}_{ij}$ are shown in Table I.

TABLE I
Membership functions of $\tilde{A}_{ij}$

i, j	$\overline{\mu}_{\tilde{A}_{ij}}$ $j = 1$		$\underline{\mu}_{\tilde{A}_{ij}}$ $j = 2$	
$i = 1$	$T(2,6,8)$	$T(1,4,7)$	$T(3,6,7)$	$T(3,4,6)$
$i = 2$	$G(2,9)$	$G(2,8)$	$G(1.5,9)$	$G(1.5,8)$

Table I shows the parameters of two kind of membership functions: Triangular and gaussian, denoted by *T(a,b,c)* and *G*(μ, δ), respectively. We compute six α-cuts over each $\tilde{A}_{ij}$, $\alpha \in [0.05, 0.2, 0.4, 0.6, 0.8, 1]$ according to equation (22). Each ${}^{\alpha}\tilde{A}_{ij}$ reaches a different value of z associated to α, so a set $\tilde{Z}$ of 22 optimal solutions[2] is obtained, which is shown in Figure 4.

Note that all values of z are a function of α, so $\tilde{Z}$ is mapped by α-cuts instead of $z \in \mathbb{R}$. As we pointed out in Figure 3, we have no all values of $\overline{w}_i$ and $\underline{w}_i$, so the Algorithm 1 is applied to complete the information we need to compute z_l and z_r. The obtained results are shown in Table II.

[2]Each set of $\tilde{A}_{ij}$ leads to solve a linear programming model, so we solved 22 linear models.

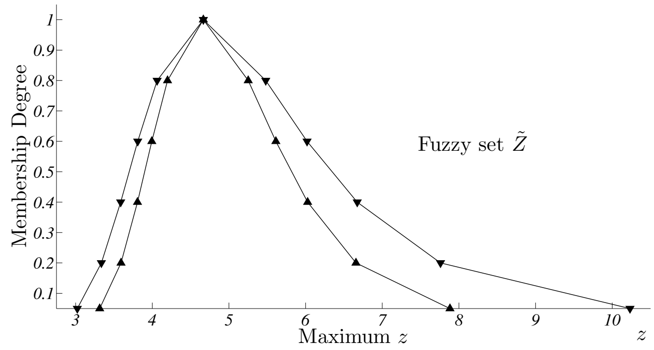

Fig. 4. Fuzzy set $\tilde{Z}$ composed by α-cuts

TABLE II
OBTAINED RESULTS

α_n	${}^{\alpha_n}\overline{\mu}_{\tilde{z}}$	${}^{\alpha_n}\underline{\mu}_{\tilde{z}}$	x_i	$\overline{w}_i$	$\underline{w}_i$	$\overline{w}_i$	$\underline{w}_i$
0.05	3.0227	—	3.0227	0.05	0	—	—
0.05	—	3.3144	3.3144	—	—	0.1894	0.05
0.2	3.3365	—	3.3365	0.2	0.0619	—	—
0.4	3.5876	—	3.5876	0.4	0.1974	—	—
0.2	—	3.5925	3.5925	—	—	0.4044	0.2
0.4	—	3.8075	3.8075	—	—	0.5972	0.4
0.6	3.8106	—	3.8106	0.6	0.4033	—	—
0.6	—	3.9939	3.9939	—	—	0.7450	0.6
0.8	4.0635	—	4.0635	0.8	0.6678	—	—
0.8	—	4.1992	4.1992	—	—	0.8450	0.8
1	4.6667	4.6667	9.3334	1	1	1	1
0.8	—	5.2513	5.2513	—	—	0.8563	0.8
0.8	5.4802	—	5.4802	0.8	0.6731	—	—
0.6		5.612	5.612	—	—	0.7511	0.6
0.6	6.0189	—	6.0189	0.6	0.4038	—	—
0.4	—	6.0267	6.0267	—	—	0.5976	0.4
0.2	—	6.6572	6.6572	—	—	0.4052	0.2
0.4	6.6741	—	6.6741	0.4	0.1979	—	—
0.2	7.76	—	7.76	0.2	0.0650	—	—
0.05	—	7.8829	7.8829	—	—	0.1925	0.05
0.05	10.231	—	10.231	0.05	0	—	—

In this table, $\underline{w}_n$ and $\overline{w}_n$ are the resultant values of their respective ${}^{\alpha_n}\overline{\mu}_{\tilde{z}}$ and ${}^{\alpha_n}\underline{\mu}_{\tilde{z}}$ which are obtained by α_n and the parameters of the Table I. For instance, if we refer to $x_i = 3.5876$ which is obtained through $\alpha = 0.4$ (See first column), then we get $\overline{w}_i = 0.4, \underline{w}_i = 0.1974$, and for $x_i = 4.1992$ which is obtained through $\alpha = 0.8$ (See first column), we get $\overline{w}_i = 0.845, \underline{w}_i = 0.8$, and so on. Also note that x_i (fourth column) is composed by the ordered values of ${}^{\alpha_n}\overline{\mu}_{\tilde{z}}$ and ${}^{\alpha_n}\underline{\mu}_{\tilde{z}}$, and both first and last values of $\underline{w}_i$ are zero as we pointed out in Remark 5.1.

Note in Figure 4 that $\overline{\mu}_{\tilde{z}}$ is composed by the values of the second column of Table II, and $\underline{\mu}_{\tilde{z}}$ is composed by the values of the third column of Table II, which are obtained by optimizing the example for ${}^{\alpha_n}\overline{\mu}_{\tilde{A}_{ij}}$ and ${}^{\alpha_n}\underline{\mu}_{\tilde{A}_{ij}}$, respectively.

Now that we have all values of $x_i, \overline{w}_i$ and $\underline{w}_i$, we apply the IASCO algorithm (Melgarejo [13]) for obtaining the centroid of $\tilde{Z}$ as a deffuzzification value of the behavior of the optimal values of the problem, namely z_r, z_l and z_c. The obtained results are as follows:

$$z_r = 4.661$$
$$z_l = 5.117$$
$$z_c = \frac{z_r + z_l}{2} = 4.889$$

It is clear that by using the results of $\tilde{Z}$ composed only by its α-cuts, we cannot compute z_r, z_l and z_c, so the Algorithm 1 is an useful tool for completing $\overline{w}_i$ and $\underline{w}_i$, which are needed by either the EKM or the IASCO algorithms.

VII. CONCLUDING REMARKS.

Some conclusions and recommendations can be suggested:

1) Although there are different methods for computing the centroid of an IT2FS, this paper focuses in the computation of all memberships $\overline{w}_i$ and $\underline{w}_i$ for all the available values of the universe of discourse $x_i \in X$.
2) The presented approximation method can be applied to a family of Type-2 fuzzy problems which are solved through primary α-cuts, in a simple way.
3) The concept of primary α-cut is applied to an optimization problem, using the presented algorithm for computing its deffuzzified centroid, with successful and consistent results.
4) Finally, the presented approximation algorithm can be applied to any representation of an IT2FS with incomplete $\overline{w}_i$ and $\underline{w}_i$, so its application to optimization and decision making problems has a wide potential.

A. Further Topics.

The Generalized Type-2 Fuzzy Sets (GT2 FS) approach arises as the next step on Type-reduction. This approach uses the secondary membership function $f_x(u)/u$ of n T2FS, inducing researchers to new directions.

REFERENCES

[1] J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*, N. Upper Saddle River, Ed. Prentice Hall, 2001.

[2] ——, "Type-2 Fuzzy Sets: Some Questions and Answers." *IEEE coNNectionS. A publication of the IEEE Neural Networks Society.*, no. 8, pp. 10–13, 2003.

[3] ——, "Fuzzy sets for words: a new beginning." in *The IEEE International Conference on Fuzzy Systems.*, 2003, pp. 37–42.

[4] J. M. Mendel, R. I. John, and F. Liu, "Interval type-2 fuzzy logic systems made simple," *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 6, pp. 808–821, 2006.

[5] J. M. Mendel and R. I. John, "Type-2 fuzzy sets made simple," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 117–127, 2002.

[6] Q. Liang and J. M. Mendel, "Interval type-2 fuzzy logic systems: Theory and design," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 5, pp. 535–550, 2000.

[7] N. N. Karnik and J. M. Mendel, "Operations on type-2 fuzzy sets," *Fuzzy Sets and Systems*, vol. 122, pp. 327–348, 2001.

[8] N. N. Karnik, J. M. Mendel, and Q. Liang, "Type-2 fuzzy logic systems," *Fuzzy Sets and Systems*, vol. 17, no. 10, pp. 643–658, 1999.

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

[10] M. A. Melgarejo, "Implementing Interval Type-2 Fuzzy processors," *IEEE Computational Intelligence Magazine*, vol. 2, no. 1, pp. 63–71, 2007.

[11] ——, "A Fast Recursive Method to compute the Generalized Centroid of an Interval Type-2 Fuzzy Set," in *Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS)*. IEEE, 2007, pp. 190–194.
[12] M. A. Melgarejo, C. A. Peña, and E. Sanchez, "A genetic-fuzzy system approach to control a model of the hiv infection dynamics," in *2006 International Conference on Fuzzy Systems*, IEEE, Ed. IEEE, 2006, pp. 2323–2330.
[13] K. Duran, H. Bernal, and M. Melgarejo, "Improved iterative algorithm for computing the generalized centroid of an interval type-2 fuzzy set," in *2008 Annual Meeting of the IEEE North American Fuzzy Information Processing Society (NAFIPS)*, 2008.
[14] S. Chanas and J. Kamburoswk, "The use of fuzzy variables in PERT," *Fuzzy Sets and Systems*, vol. 5, pp. 1–9, 1981.
[15] S.-P. Chen, "Analysis of critical paths in a project network with fuzzy activity times," *European Journal of Operational Research*, vol. 183, pp. 442–459, 2007.
[16] S. Chanas, D. Dubois, and P. Zieliński, "On the sure criticality of tasks in activity networks with imprecise durations," *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 32, pp. 393–407, 2002.
[17] D. Wu and J. M. Mendel, "Enhanced Karnik-Mendel algorithms for Interval Type-2 fuzzy sets and systems," in *Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS)*, vol. 26. IEEE, 2007, pp. 184–189.
[18] ——, "Enhanced Karnik–Mendel Algorithms," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 4, pp. 923–934, 2009.
[19] J. Mendel, F. Liu, and D. Zhai, "α-plane representation for type-2 fuzzy sets: Theory and applications," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 5, pp. 1189–1207, 2009.
[20] ——, "Comments on α-plane representation for type-2 fuzzy sets: Theory and applications," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 1, pp. 229–230, 2010.
[21] F. Liu, "An efficient centroid type reduction strategy for general type-2 fuzzy logic system," IEEE Comput. Intell. Soc.,Walter J. Karplus Summer Res. Grant Rep., Tech. Rep., 2006.
[22] ——, "An efficient centroid type reduction strategy for general type-2 fuzzy logic system," *Information Sciences*, vol. 178, no. 5, pp. 2224–2236, 2008.
[23] J. C. Figueroa, "Interval type-2 fuzzy linear programming: Uncertain constraints," in *IEEE Symposium Series on Computational Intelligence*. IEEE, 2011, pp. 1–6.
[24] ——, "Linear programming with interval type-2 fuzzy right hand side parameters." in *2008 Annual Meeting of the IEEE North American Fuzzy Information Processing Society (NAFIPS)*, 2008.