

Constrained Interval Type-2 Fuzzy Sets

Pasquale D'Alterio, *Student Member, IEEE*, Jonathan M. Garibaldi, *Senior Member, IEEE*,
Robert I. John, *Senior Member, IEEE*, and Amir Pourabdollah, *Member, IEEE*

Abstract—In many contexts, type-2 fuzzy sets are obtained from a type-1 fuzzy set to which we wish to add uncertainty. However, in the current type-2 representation there is no restriction on the shape of the footprint of uncertainty and the embedded sets that can be considered acceptable. This leads, usually, to the loss of the semantic relationship between the type-2 fuzzy set and the concept it models. As a consequence, the interpretability of some of the embedded sets and the explainability of the uncertainty measures obtained from them can decrease. To overcome these issues, constrained type-2 fuzzy sets have been proposed. However, no formal definitions for some of their key components (e.g. acceptable embedded sets) and constrained operations have been given. The goal of this paper is to provide some theoretical underpinning for the definition of constrained type-2 sets, their inferencing and defuzzification method. To conclude, the constrained inference framework is presented, applied to two real world cases and briefly compared to the standard interval type-2 inference and defuzzification method.

Index Terms—Type-2 fuzzy logic, embedded sets, constrained type-2 fuzzy sets, xai

I. INTRODUCTION

Type-2 fuzzy sets (T2 FS) were introduced by Zadeh [1] in 1975 as an extension of type-1 (T1) FSs so that it would be possible to model the uncertainty of membership functions. However, their use remained rare due to the significant increase in complexity of algorithms, until Mendel introduced many advances that made their practical use possible. Type-2, and particularly interval type-2 (IT2) sets, have been successfully applied in many areas such as control [2], [3], classification and regression [4], and many other contexts.

Type-2 fuzzy systems have required the creation of additional representations, definitions and algorithms, including to allow the creation of complete rule-based inferencing systems. One of these is the concept of the Footprint of Uncertainty (FOU), introduced by Mendel [5], which represents the existence of non-zero secondary membership values as a two-dimensional shaded area. Other novel methodologies have been suggested for type-reduction and defuzzification, largely based on algorithms for centroid defuzzification of type-2 sets initially by Karnik and Mendel [6], and subsequently enhanced by Wu and Mendel [7], and others. The original KM, modern EKM, and other similar recent algorithms, are based on the concept of the embedded set (ES). Intuitively, an ES is a path along the surface of a T2 set and it has been proven [8] that any T2 FS can be represented as the union of all its ESs (representation theorem). The process of finding the centroid of a type-2 set then depends on finding the ES with the leftmost centroid, and that with the rightmost centroid.

Whilst the current T2 and IT2 framework has shown to have many advantages over T1 approaches, particularly in their

ability to exhibit greater performance in most situations, we believe there are drawbacks. Two properties which we believe decrease the overall interpretability of T2 systems are: (i) there is currently no agreed mechanism to derive the FOU, particularly in the situation in which a concept being modelled by a T1 set has uncertainty added to form a T2 set representing the same concept; and (ii) ESs may have any shape, including ones which bear no relationship to the concept being modelled.

To overcome these issues, Constrained Type-2 (CT2) fuzzy sets have been proposed [9], [10]. The idea behind them is to address the two limitations above by: (i) providing an explicit method for generating the boundaries of the footprint of uncertainty that keeps a *shape coherency* [9] throughout the generation of the type-2 set, based on an underlying concept modelled by a type-1 set; and (ii) restricting the acceptable embedded sets that may be used to only a subset of all the ESs, in order to process only shapes that may be considered meaningful in that specific context. Even though the concept of CT2 FS has already been formulated [9], [10], some key components are currently lacking formal definitions such as the acceptable embedded sets, constrained inference and centroid defuzzification.

In this paper, we will provide some theoretical underpinning for this new constrained representation, focusing specifically on constrained interval type-2 (CIT2) fuzzy sets. In addition to formal definitions, a full inferencing and defuzzification framework is then proposed for the creation of CIT2 Mamdani-style fuzzy inference systems. Next, we compare and contrast the CIT2 approach with the recent framework introduced by Wu et al [11], [12] for creating ‘well-shaped’ type-2 sets. Finally, a practical application will be shown and compared with the conventional IT2 representation in terms of interpretability and explainability of the outputs, performances and running times. Specifically, a genetic architecture will be described for the automatic generation of CIT2 fuzzy systems which will be tested on two real world data-sets. Whilst interpretability is itself a difficult and complex concept to define, and is somewhat subjective in nature, nevertheless we use worked examples and the practical applications to illustrate ways in which interpretability is enhanced. Throughout, we stress that the proposed CIT2 approach, which may be used in contexts in which explainability and interpretability are considered important, is an *alternative* to other approaches including the conventional type-2 approach.

II. PRELIMINARY DEFINITIONS

In this section we will provide some formal definition of fuzzy concepts that will be used throughout the paper (definitions taken or rephrased from [8], [9], [13], [14]).

Definition 1. A type-1 fuzzy set (T1 FS), denoted A , is characterized by a T1 membership function (MF) $\mu_A : X \mapsto [0, 1]$, i.e.,

$$A = \{(x, \mu_A(x)) \mid x \in X\} \quad (1)$$

with X being the universe of discourse (UOD).

Definition 2. A type-2 fuzzy set (T2 FS), denoted \tilde{A} , is characterized by a T2 MF $\mu_{\tilde{A}} : X \times [0, 1] \mapsto [0, 1]$, i.e.,

$$\tilde{A} = \{((x, u), \mu_{\tilde{A}}(x, u)) \mid x \in X, u \in [0, 1]\} \quad (2)$$

in which X is the UOD.

Definition 3. An interval type-2 fuzzy set (IT2 FS), denoted \tilde{A} , is characterized by an IT2 MF $\mu_{\tilde{A}} : X \times [0, 1] \mapsto \{0, 1\}$ (i.e. $\mu_{\tilde{A}}(x, u)$ is either 0 or 1):

$$\tilde{A} = \{((x, u), \mu_{\tilde{A}}(x, u)) \mid x \in X, u \in [0, 1]\} \quad (3)$$

in which X is the UOD.

Definition 4. Given a T2 FS \tilde{A} , its footprint of uncertainty (FOU) is the set of points (x, u) for which $\mu_{\tilde{A}}(x, u) > 0$:

$$FOU(\tilde{A}) = \{(x, u) \mid (x, u) \in X \times [0, 1], \mu_{\tilde{A}}(x, u) > 0\} \quad (4)$$

Definition 5. Given a T2 FS \tilde{A} and a value $x \in X$, we define the set of pairs J_x as:

$$J_x = \{(x, u) \mid u \in [0, 1], \mu_{\tilde{A}}(x, u) > 0\} \quad (5)$$

Definition 6. Given a T2 FS \tilde{A} and a value $x \in X$, a secondary MF is a function $\mu_{\tilde{A}(x)}$ such that:

$$\mu_{\tilde{A}(x)} : [0, 1] \mapsto [0, 1]; \mu_{\tilde{A}(x)}(u) = \mu_{\tilde{A}}(x, u), \forall x \in X \quad (6)$$

The domain of a secondary MF is called the primary membership of x .

Definition 7. A type-2 embedded set (T2 ES), denoted \tilde{A}_E , is a path along the T2 set it belongs to. It contains only one primary degree u_x for each x , with its associated secondary grade v_x :

$$\mu_{\tilde{A}_E}(x, u_x) = v_x \quad x \in X, (x, u_x) \in J_x \quad (7)$$

Definition 8. A type-1 embedded set (T1 ES), denoted A_E represents a projection of a T2 ES, i.e. its secondary degree has been dropped. Therefore it contains one primary degree u_x for each x :

$$\mu_{A_E}(x) = u_x \quad x \in X, (x, u_x) \in J_x \quad (8)$$

III. MOTIVATION

In the literature, there are three main approaches to determine the upper and lower bounds of the FOU of T2 FSs when starting from already existing T1 MFs modeling the same concept. The first one identifies the two boundary MFs by taking the parameters of the existing T1 MFs and adding some uncertainty to them [15]–[20]. For example, in the case of a T1 Gaussian with mean m and variance v , the upper and lower bounds of the FOU could be the Gaussians with mean m and variances $v - k$ and $v + k$ respectively, with k being a positive real number.

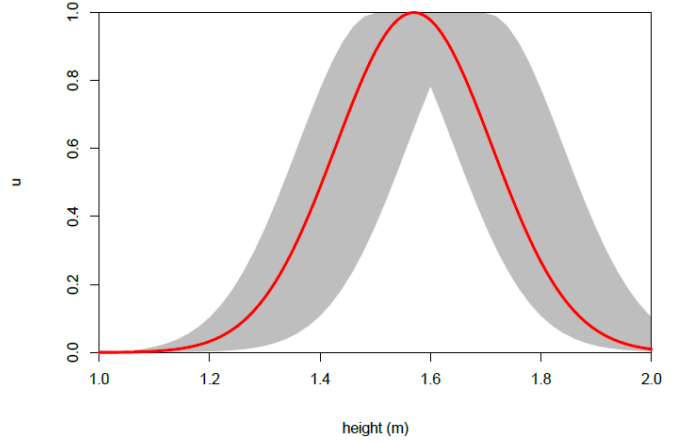


Fig. 1. In red, one of the embedded sets of the interval type-2 fuzzy set in grey (picture from [9])

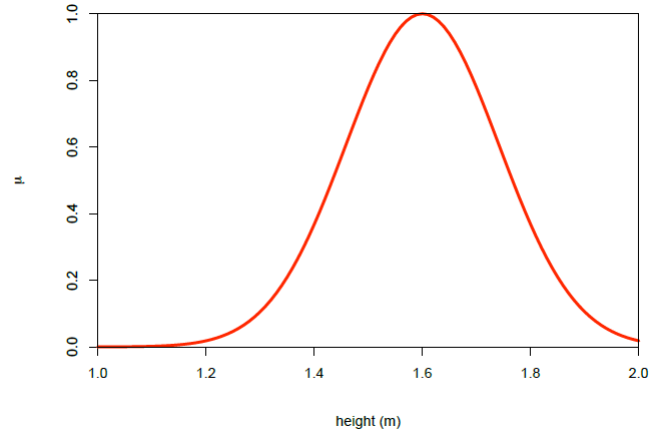


Fig. 2. T1 Gaussian MF (picture from [9])

A different method defines the FOU as the area covered by the translation along the x-axis of the starting T1 MF by a factor c and $-c$, $c \in \mathbb{R}$ [21]–[23]. The result is a symmetrical blurring around the starting T1 MF. An example of an FOU obtained with this approach with a T1 Gaussian can be seen in Fig. 1.

Another approach has also been proposed. It models the FOU so that it embeds all the T1 MFs obtainable from observations [24] or from the modeling of the same concept under different circumstances [25].

All those methods have in common the fact that they identify some T1 shapes as “meaningful” in their context and then use them to build the FOU. However, when some fuzzy operators such as the Karnik-Mendel (KM) type-reduction algorithm [6] are used, all the ESs are processed, regardless of their shape. As a consequence of that, ESs that could hardly represent the concept they are modelling, will likely determine the end-point of the defuzzified centroid. Since those ESs have a low interpretability due to their shape, the explainability of the output and, consequently, of the fuzzy system or set that generates it, decreases. However, in the recent years building explainable intelligent systems has become increasingly important [26], [27]. We will now use

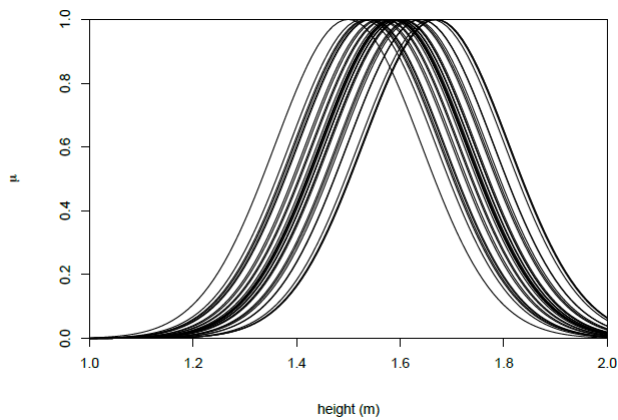


Fig. 3. Possible result of the thought experiment described above (picture from [9])

the following examples to support our claims. Suppose that we decide to model the concept of *medium height* using a T1 Gaussian MF, as shown in Fig. 2. We will call this set the T1 generator set (GS). If we want to build an IT2 FS from that, one of the possible approaches would be to ask different people to place the mean of the Gaussian on the x-axis, after its variance value had been previously determined (similar approaches can be found in [28], [29]).

It is likely that we would obtain something similar to what is shown in Fig. 3, since the concept of *medium height* would vary slightly from person to person. Now we can use this collection of T1 MFs to determine the FOU of our IT2 FS.

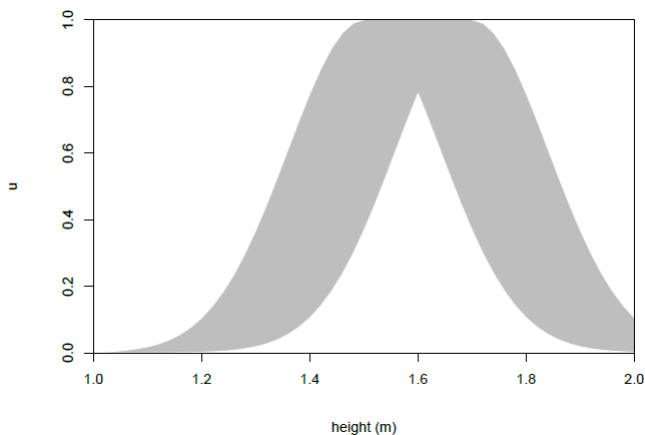


Fig. 4. FOU of a possible IT2 FS modelling medium height (picture from [9])

As in [25], we will embed those sets in our FOU. To do so, we will use the translation method mentioned above, i.e. we will define our FOU as the area covered by the shifting of the GS from the leftmost to the rightmost Gaussian to embed. The result of this operation is shown in Fig. 4.

If we use the standard IT2 representation, the ESs within the FOU will have arbitrary shapes. That makes even the ES shown in Fig. 5 acceptable. In this particular context, it is clear that a T1 ES like that has very little relation with the concept of *medium height*. In fact, no observation of the

participants' opinion during the experiment led to such shape. Furthermore, this representation affects the centroid value and its explainability. The set shown in Fig. 6 has been obtained with the process described in our thought experiment above.

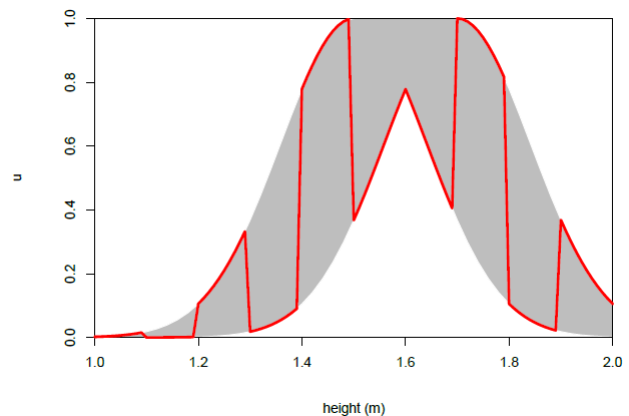


Fig. 5. One of the embedded set of the FOU shown (picture from [9])

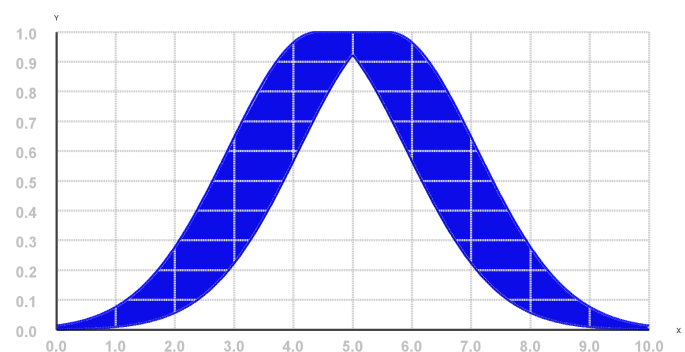


Fig. 6. Possible FOU generated from a Gaussian T1 MF

If one uses the KM procedure [6] to type-reduce it, the algorithm will find the two ESs that give us the left and right endpoints of the centroid. For the IT2 FS in Fig. 6, the results are shown in Fig. 7.

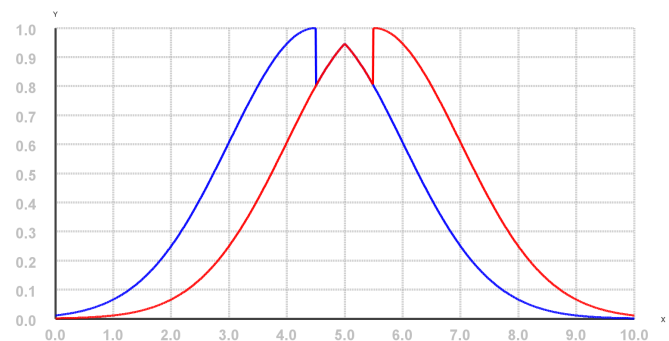


Fig. 7. ESs used by the KM procedure to obtain the centroid of the IT2 FS in Fig. 6

These sets do not seem to fit our case very well. That is because, to obtain the type-reduced value, the algorithm chose two ESs that did not represent any of the observations made during the experiment; additionally, those shapes could

hardly represent the concept of *medium height* that is being considered.

System output defuzzification represents another useful example to see how the standard IT2 representation affects the interpretability and explainability of fuzzy systems. Consider, for example, the fuzzy output set shown in Fig. 8, and its associated left and right endpoints shown in Figs. 9 and 10, respectively. In Fig. 9, the embedded sets of the left endpoint derived using the constrained centroid (Fig. 9(a)), and the KM procedure (Fig. 9(b)) are compared. Similarly, Fig. 10 compares those of the right endpoint. The ES used for the constrained centroid preserve the same level of interpretability of T1 system outputs in that the shapes of the generator sets are clearly identifiable and so are the firing strengths that generated them. As a consequence of this, it is possible to get an intuitive idea of the sets that lead to the end-points. In addition to that, knowing which rules (and therefore which inputs and antecedents) generated the ES from which the endpoints are obtained, gives an explanation to how and why the final output of the system has been obtained. In the KM case, on the other hand, the shape coherency with the original shape is partly lost and the firing strengths are not as clear as in the CIT2 case.

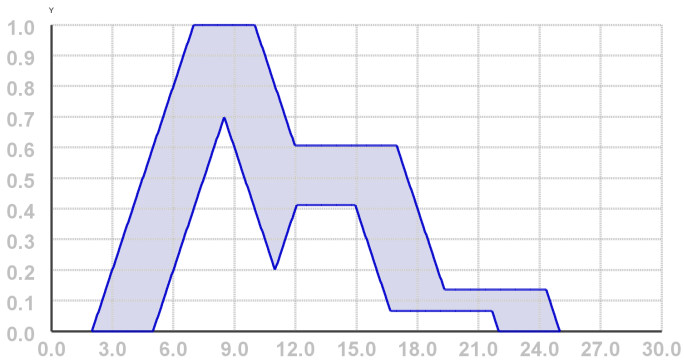


Fig. 8. Fuzzy output of a CIT2 fuzzy system

Intuitively, the standard T2 definition gives too much “mathematical freedom” in some contexts, posing no restrictions on the shape of the FOU and of the ESs, especially when modeling T2 MFs from an underlying concept represented as a T1 FS with uncertainty. For these reasons, CIT2 FSs were proposed, in which both the FOU and the ESs considered as acceptable have a shape that is “meaningful” for the context in which they are used.

The specific sense of “meaningfulness” can vary. The intuitive idea is that the shape of the MFs should be reasonable for the semantic meaning they carry. For example, in the case of the concept of medium height, only a MF that monotonically increases up to a plateau and then monotonically decreases would be “meaningful”. That is simply because any MF without these properties would result in a counter-intuitive set for the representation of the medium height concept.

In other contexts, meaningful shapes can be obtained as a result of experimental observations, data analysis or experts’ knowledge. The topic has been discussed in detail in [30], in

which a possible mathematical definition for the concept of meaningfulness in the context of fuzzy sets has been given.

IV. CONSTRAINED INTERVAL TYPE-2 FUZZY SETS

Although we assert the main concepts of CT2 FSs can be extended to all T2 FSs, the rest of the paper will only focus on *interval* type-2 fuzzy sets and their constrained representation (CIT2). The motivations behind this decision will be discussed later in the paper. Also, we assume that the universe of discourse (UOD) we are working with is a connected subset of \mathbb{R} .

The idea behind CIT2 FSs is to generate a T2 FS starting from a T1 FS modeling the same semantic concept. This T1 FS is called type-1 generator set (T1 GS) (e.g. the T1 FS in Fig. 2 is the T1 GS for our thought experiment in Sec. III). To obtain the CIT2 FS, we add uncertainty on the location of the T1 GS on the x-axis. We do that by using a set of offsets, that intuitively represent all the possible valid locations of our T1 GS. We call this set of offsets the *displacement set*:

Definition 9. A displacement set (DS), denoted D , is a closed set of real numbers such that:

$$D \subseteq \mathbb{R}, 0 \in D \quad (9)$$

When the DS is a continuous interval, it can be expressed as $D=[a,b]$, where $a, b \in \mathbb{R}, a \leq 0, b \geq 0$

With a DS plus a T1 GS, we can define the T1 FSs that will represent the acceptable embedded sets (AES) of the CIT2 FS we are modeling.

Definition 10. A collection of T1 acceptable embedded sets (CAES), is a set of T1 FSs obtained from the shifting of a T1 GS G . Formally, each of the acceptable embedded sets (AES) S in a CAES can be expressed as:

$$S = \{(x, \mu_S(x)) \mid x \in X\} \quad (10)$$

where

$$\mu_S : X \mapsto [0, 1], \exists c \in D : \mu_S(x) = \mu_G(x-c), \forall x \in X \quad (11)$$

given a UOD X , a DS D , a T1 GS G .

Given a CAES, we can generate a CIT2 FS:

Definition 11. A constrained interval type-2 fuzzy set (CIT2 FS) \check{A} , is defined as follows:

$$\check{A} = \{((x, u), 1) \mid x \in X, u \in \bigcup_{S \in \text{CAES}_{\check{A}}} \mu_S(x)\} \quad (12)$$

with $\text{CAES}_{\check{A}}$ being the CAES from which we obtain \check{A} . In this case, J_x can be rewritten as follows:

$$J_x = \bigcup_{S \in \text{CAES}_{\check{A}}} (x, \mu_S(x)), \quad \mu_S(x) > 0 \quad (13)$$

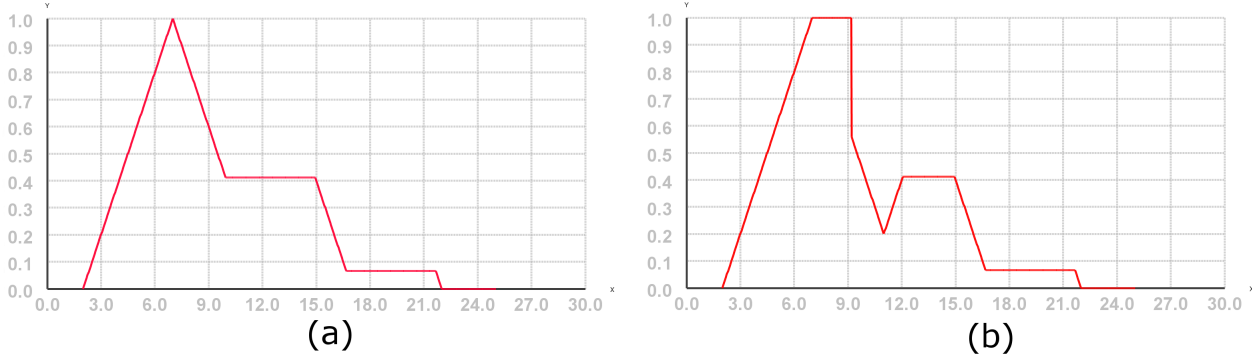


Fig. 9. ESs that determine the left value of the CIT2 (a) and KM (b) centroid of the set in Fig. 8.

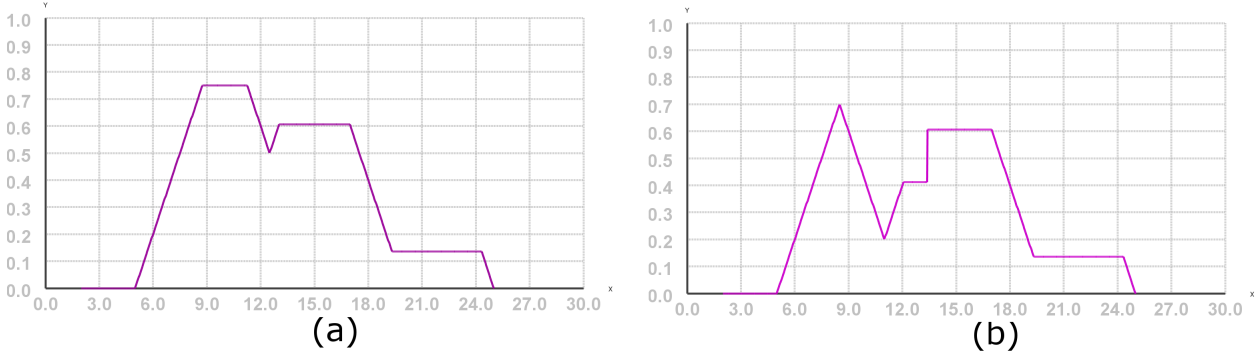


Fig. 10. ESs that determine the right value of the CIT2 (a) and KM (b) centroid of the set in Fig. 8.

\check{A} can also be written as:

$$\begin{aligned} \check{A} &= \int_{x \in X} \int_{u: (x,u) \in J_x} 1 / (x, u) \\ &= \int_{x \in X} \int_{u \in \bigcup_{S \in \widetilde{\text{CAES}}_{\check{A}}} \mu_S(x)} 1 / (x, u) \end{aligned} \quad (14)$$

It is important to note that CIT2 FSs represent a subset of IT2 FSs since they impose additional constraints on their mathematical definition, just like IT2 FSs represent a subset of the more general T2 FSs.

In order to prove an important property, we need to build a three-dimensional version of the sets in our CAES. Since they are T1 FSs, building their three-dimensional representation is straightforward. Given a T1 set A , its three-dimensional representation \tilde{A} (i.e. its representation as a T2 FS) is defined as follows:

$$\tilde{A} = \{(x, \mu_A(x), 1) \mid x \in X\} \quad (15)$$

By applying (15) to all the sets in a given CAES, we obtain a collection of IT2 acceptable embedded sets.

Definition 12. A collection of acceptable IT2 embedded sets ($\widetilde{\text{CAES}}$) of a CIT2 set \check{A} , denoted $\widetilde{\text{CAES}}_{\check{A}}$, is a set of CIT2 embedded sets described as follows:

$$\widetilde{\text{CAES}}_{\check{A}} = \{\tilde{S} \mid S \in \text{CAES}_{\check{A}}\} \quad (16)$$

with

$$\tilde{S} = \{(x, \mu_S(x), 1) \mid x \in X\} \quad (17)$$

Each of the sets \tilde{S} , can also be described as:

$$\tilde{S} = \int_{x \in X} \int_{\mu_S(x)} 1 / x = \int_{x \in X} (\mu_S(x), 1) / x \quad (18)$$

The sets in the $\widetilde{\text{CAES}}_{\check{A}}$ are actual T2 ESs of \check{A} , since they satisfy Definition 7.

While all the definitions up to this point could be easily extended to the general CT2 case, the conversion of T1 MFs to AESs of a general T2 FS would not be so trivial. That is because the membership degree of each of the pairs $((x, \mu_S(x)))$ could not be easily determined since it could be any value between 0 and 1. The conversion to AES of IT2 FS, instead, is straightforward and shown in Def. 12. A possible solution to this has been proposed in [9], in which a similarity function is used on each AES S and the GS to determine $\mu_{\tilde{S}}(x, \mu_S(x)), \forall x$. However, the use of this and other possible approaches, together with the interpretability of three-dimensional embedded sets will be analyzed in future work. Definition 12 is very important since it allows us to introduce the CIT2 representation theorem:

Theorem 1. Given a CIT2 set \check{A} and its $\widetilde{\text{CAES}}_{\check{A}}$, \check{A} can be expressed as the crisp set union of all the IT2 sets \tilde{S} in $\widetilde{\text{CAES}}_{\check{A}}$:

Proof. To do that, we simply show that it is possible to write the union of all the $\tilde{S} \in \widetilde{\text{CAES}}_{\check{A}}$ as (14), by rewriting \tilde{S} as in

(18):

$$\begin{aligned} \int_{\widetilde{S} \in \widetilde{\text{CAES}}_{\check{A}}} \check{S} &= \int_{\widetilde{S} \in \widetilde{\text{CAES}}_{\check{A}}} \left(\int_{x \in X} \int_{u = \mu_S(x)} 1 / (x, u) \right) \\ &= \int_{x \in X} \int_{u \in \bigcup_{S \in \text{CAES}_{\check{A}}} \mu_S(x)} 1 / (x, u) \end{aligned} \quad (19)$$

□

Theorem 1 allows us to define CIT2 operations by only working with AESs. For example, the union of two sets \check{A} and \check{B} is defined as follows.

Corollary 1. *Given two CIT2 sets \check{A} and \check{B} , their union is the union of the T2 embedded sets \check{S} in $\widetilde{\text{CAES}}_{\check{A}}$ and $\text{CAES}_{\check{B}}^1$:*

$$\begin{aligned} \check{A} \cup \check{B} &= \int_{\check{A}' \in \widetilde{\text{CAES}}_{\check{A}}} \check{A}' \cup \int_{\check{B}' \in \widetilde{\text{CAES}}_{\check{B}}} \check{B}' \\ \check{A} \cup \check{B} &= \int_{\check{A}' \in \widetilde{\text{CAES}}_{\check{A}}} \int_{\check{B}' \in \widetilde{\text{CAES}}_{\check{B}}} \check{A}' \cup \check{B}' \end{aligned} \quad (20)$$

Intuitively, we are considering all the combinations of all the AES of the two CIT2 sets involved in the operation. The unions between the AESs of \check{A} and \check{B} generate the AESs of the FS generated from the union of \check{B} and \check{B} .

Analogously, we can derive the CIT2 intersection and complement:

$$\check{A} \cap \check{B} = \int_{\check{A}' \in \widetilde{\text{CAES}}_{\check{A}}} \int_{\check{B}' \in \widetilde{\text{CAES}}_{\check{B}}} \check{A}' \cap \check{B}' \quad (21)$$

$$\check{\bar{A}} = \int_{\check{A}' \in \widetilde{\text{CAES}}_{\check{A}}} \check{\bar{A}}' \quad (22)$$

Also the upper and lower MFs of the FOU of a CT2 FS can be expressed in terms of the AES:

Definition 13. *Given an CIT2 FS \check{A} , we define its upper MF $\bar{\mu}_{\check{A}}$ and lower MF $\underline{\mu}_{\check{A}}$ as follows:*

$$\bar{\mu}_{\check{A}}(x) = \sup_{S \in \text{CAES}_{\check{A}}} \mu_S(x) \quad (23)$$

$$\underline{\mu}_{\check{A}}(x) = \inf_{S \in \text{CAES}_{\check{A}}} \mu_S(x) \quad (24)$$

Even though IT2 and CIT2 operations may seem similar, they are conceptually different. In the IT2 case, the only goal of operations such as the union and intersection is to generate the new upper and lower-bound MFs and therefore the FOU. In the CIT2 case that is not enough. In fact, they key point of CIT2 operators is the generation of a new CAES, that determines which ES are considered acceptable and therefore which ES will be considered by other CIT2 fuzzy operators (such as the centroid). This property is necessary to maintain the concept of interpretability (as semantic relation) described so far in the paper.

¹(20) involves integral and union signs, where the integral sign is shorthand for lots of union signs. The union sign indicates the union between members of a set, whereas the integral sign represents the union of the sets themselves.

Since every $\widetilde{\text{CIT2}}$ set can be expressed as the union of the AES in its CAES, we can use this property to define the constrained centroid, denoted as $C(\check{A})$:

$$C(\check{A}) = \int_{\check{A}' \in \widetilde{\text{CAES}}_{\check{A}}} C(\check{A}') \quad (25)$$

That is, the union of all the centroids of the sets in $\widetilde{\text{CAES}}_{\check{A}}$. The constrained centroid is analogous to the IT2 one, in which the centroid is the union of the centroids of all its embedded sets [13]. The difference is that in the CIT2 case we only take into account the set of the AESs. They represent a subset of all the ESs examined in the standard IT2 approach. In addition, since the CAES is a subset of all the ES embedded in a given FOU, the constrained centroid will always be contained (or will be equal to) the standard IT2 centroid.

When a CIT2 FS is not the result of a CIT2 fuzzy operator but is generated from a T1 GS with a continuous DS, the CIT2 centroid has an interesting mathematical property. In fact, in that case, the centroid can be rewritten as the following interval:

$$C(\check{A}) = [C(\check{A}_L), C(\check{A}_R)], \quad \check{A}_L, \check{A}_R \in \widetilde{\text{CAES}}_{\check{A}} \quad (26)$$

with \check{A}_L, \check{A}_R being the left-most and right-most AES of \check{A} . The proof for that equation is straightforward: since all the AES of a CIT2 generated from a GS share the same shape, the AES obtained from the leftmost shift will trivially have the lowest centroid value and will therefore determine the left endpoint of the centroid; analogously, the right endpoint is generated by the rightmost AES.

However, (26) may not hold anymore after the application of a set theory operation. Intuitively, that is because (26) can be used when all the sets in $\widetilde{\text{CAES}}$ have the same shape. An example of a case in which (26) can not be used is given by the CIT2 FS in Fig. 12. Its AES (e.g. Fig. 9 (a), 10 (a)) are obtained as the aggregation of three triangular MFs “truncated” (i.e. inferred) at different heights. In that case, determining which “truncation values” generate the AES with the lowest and highest centroid value is non-trivial, as will be also discussed in Sec. V-A.

Lastly, the FOU (see (4)) of a CIT2 FS \check{A} can be rewritten using only the AESs:

Definition 14. *The FOU of a CIT2 FS \check{A} can be defined as:*

$$\text{FOU}(\check{A}) = \{(x, u) \mid ((x, u), 1) \in \int_{\check{S} \in \widetilde{\text{CAES}}_{\check{A}}} \check{S}\} \quad (27)$$

V. INFERENCE WITH CIT2 SETS

Now that we have a formal definition of CIT2 FSs and all their components, we can use them to build fuzzy rules and fuzzy systems. For CIT2 fuzzy systems to be usable, however, we need to define the procedure to carry out the Mamdani inference with singleton fuzzification.

Consider the following constrained interval-type-2 fuzzy rule (CIT2 fuzzy rule), i.e. a fuzzy rule in which all the sets involved are CIT2 FSs:

$$\text{IF } x_1 \text{ IS } \check{A} \text{ AND } x_2 \text{ IS } \check{B} \text{ THEN } y \text{ IS } \check{C} \quad (28)$$

Using Theorem 1, we can rewrite this as:

$$\begin{aligned} &\text{IF } x_1 \text{ IS } \int_{\widetilde{A'} \in \widetilde{\text{CAES}}_{\tilde{A}}} \tilde{A}' \text{ AND } x_2 \text{ IS } \int_{\widetilde{B'} \in \widetilde{\text{CAES}}_{\tilde{B}}} \tilde{B}' \text{ THEN} \\ &y \text{ IS } \int_{\widetilde{C'} \in \widetilde{\text{CAES}}_{\tilde{C}}} \tilde{C}' \end{aligned} \quad (29)$$

Since all the sets in the $\widetilde{\text{CAES}}$ are a three-dimensional representation of T1 sets (see (15)), we can use T1 mathematics to operate with them.

After the singleton fuzzification of the input, the antecedent operation is straightforward. For example, for the fuzzified input x_1 in the rule mentioned above, we obtain:

$$\int_{A' \in \text{CAES}_{\tilde{A}}} \mu_{A'}(x'_1) \quad (30)$$

where x'_1 is a specific value of x_1 .

The antecedent composition is therefore given by the following formula:

$$\int_{A' \in \text{CAES}_{\tilde{A}}} \mu_{A'}(x'_1) \star \int_{B' \in \text{CAES}_{\tilde{B}}} \mu_{B'}(x'_2) = \quad (31)$$

$$\int_{A' \in \text{CAES}_{\tilde{A}}} \int_{B' \in \text{CAES}_{\tilde{B}}} \mu_{A'}(x'_1) \star \mu_{B'}(x'_2)$$

with \star being a T-norm. The antecedent composition as described so far, returns a set of real numbers. Each of these values can be then used to apply the implication method (i.e. any T-norm) to each of the AES $C' \in \text{CAES}_{\tilde{C}}$, producing the $\text{CAES}_{\tilde{C}^*}$ of the fuzzy CIT2 output \tilde{C}^* . In the rest of the paper, we assume that the minimum operator is used for the implication method and informally refer to this operation as *truncation*. To defuzzify \tilde{C}^* , we implemented a procedure that is based on the result shown in (25). Our CIT2 centroid is a pair (l, u) , where:

$$l = \inf(C(\tilde{C}^*)) \quad (32)$$

$$u = \sup(C(\tilde{C}^*)) \quad (33)$$

remembering from (25) that:

$$C(\tilde{C}^*) = \int_{\widetilde{C'} \in \widetilde{\text{CAES}}_{\tilde{C}^*}} C(\tilde{C}') \quad (34)$$

Since each of the IT2 set in the $\widetilde{\text{CAES}}_{\tilde{C}^*}$ is just a three-dimensional representation of a T1 set, we can defuzzify the equivalent T1 sets in $\text{CAES}_{\tilde{C}^*}$ instead, by using the standard T1 centroid defuzzification method. Therefore, the pair (l, u) provides us a lower (l) and an upper (u) bound for the set of centroids in (25). This approach is conceptually similar to the Karnik-Mendel (KM) [6] procedure, in the sense that both return a pair composed of the upper and the lower bound of a set of centroids (that in the case of the KM approach, is the set of the centroids of all the ES of the IT2 FS).

The whole inference process where the CIT2 FSs involved have a finite number of AES, is described in pseudo-code in Algorithm 1.

A. Result of CIT2 operators

It is interesting to see how the result of CIT2 operators on CIT2 FSs, may result in a FS in which it may not be possible to identify a T1 GS G in the CAES from which we can obtain the remaining AESs by shifting G . That is because there is no guarantee that all the sets obtained as the result the implication operator, for example, will have the same shape.

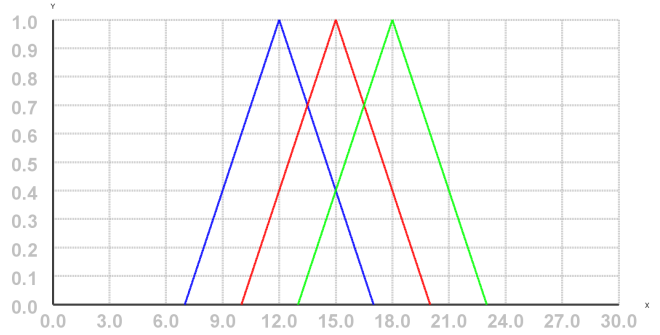


Fig. 11. Consequent CIT2 in the rule generating the output set shown in Fig. 12

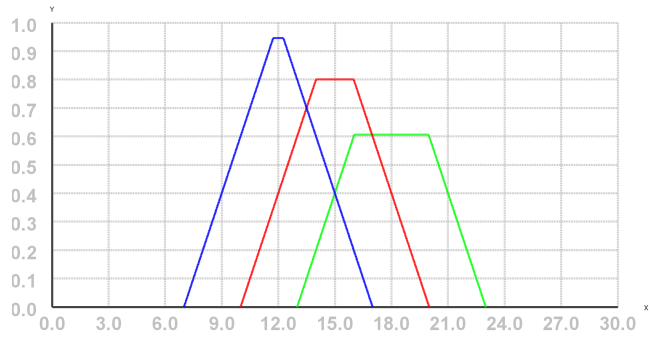


Fig. 12. Some AES of the CIT2 output from the inference of a CIT2 rule in which all the sets involved are CIT2 sets

However, the shape of the T1 GS is not totally lost after the application of CIT2 fuzzy operators. Fig. 12 shows some of the AES of the inference output of a CIT2 fuzzy rule of the form IF x_1 IS \tilde{A} THEN y IS \tilde{C} where all the CIT2 FSs involved have a discrete DS (i.e. a finite number of AES). It is possible to see that even though the sets forming the CAES of the output do not share exactly the same shape, they all come from the same generator set (i.e. a triangular T1 FS) truncated at different heights during the inference process (the consequent CIT2 FS \tilde{C} before the inference can be found in picture Fig. 11).

Intuitively, these AESs are meaningful even if they have different shapes because they represent actual T1 inference results that are obtainable from T1 inference by picking one of the AES from each of the antecedent and consequent CIT2 FSs in our fuzzy rule. The fact that each of the AESs is obtained from a shifted GS truncated at a given height is extremely important to build interpretable and explainable CIT2 systems. In fact, when one of those acceptable embedded sets is selected, its interpretability is guaranteed by the semantic connection with the concept it is modeling, since it has the

Algorithm 1 Inference and Type-Reduction Algorithm

```

1: procedure CIT2 MAMDANI INFERENCE AND TYPE-REDUCTION (CIT2-FUZZY SYSTEM S, INPUT  $[x_1, \dots, x_n]$ )
2:   for each rule  $R_i \in S$  do
3:     for each permutation P of the AES of the CIT2 antecedents in  $R_i$  do
4:       firing_strengths.add(P.evaluateAntecedents()); ▷ the FSs in P are T1 AES
5:     end for
6:     for each consequent  $C \in R_i$  do
7:       for each AES  $E \in C$  do
8:         for each  $c \in$  firing_strengths do
9:           CIT2_result_ $R_i$ .add_AES(implicate(E, c)); ▷ add a new AES to the rule  $i$  output
10:        end for
11:       end for
12:     end for
13:   end for
14:   CIT2_output= $\{\emptyset\}$  ▷ CIT2 FS representing the output of the system
15:   for each rule  $R_i \in$  fuzzy system S do
16:     CIT2_output=CIT2_union(CIT2_output, CIT2_result_ $R_i$ ); ▷ union of rule outputs
17:   end for
18:   left_value= $\inf_{E \in \text{CIT2\_rb\_output}} (\text{centroid}(E))$ ; ▷ lowest AES centroid value
19:   right_value= $\sup_{E \in \text{CIT2\_rb\_output}} (\text{centroid}(E))$ ; ▷ highest AES centroid value
20:   return (left_value, right_value);
21: end procedure

```

same shape as the GS, while its truncation height is directly related to the firing strength of the rule(s) that generated it. Therefore, it is possible to give an explanation for how this AES has been generated by showing the rules (and therefore, the inputs) that contributed to its creation.

The theoretical issue of having AES with different shapes was already pointed out in [10] and has now been addressed in [30], where the CIT2 fuzzy output has been formally defined as a CIT2 FS thanks to a different definition of the CAES (Def. 10) based on the concept of mathematical constraint satisfaction.

The analysis of this new definition, however, goes beyond the scope of this paper since, as already stated in [30], it does not affect any of the CIT2 operations but, in this context, just fills a theory gap.

B. On the interpretability and explainability of CIT2 sets and systems

As shown In Sec. IV, the CIT2 FOU is a set of points, exactly like the FOU of a standard IT2 FS. If one considers the shape of a CIT2 FS alone, it is clear that its interpretability depends only on the shape of its FOU (and its boundaries) and not on the specific set of ES that are embedded into it. However, some T2 uncertainty measures do make use of these embedded sets and it is in these cases that CIT2 are able to provide a clear advantage over IT2 FS, allowing for the creation of explainable CIT2 FS and systems. Specifically, each of the AES that that can be selected by the above mentioned fuzzy operators in the CIT2 case, has been created so that it is able to carry meaningful information. This is done both by keeping a semantic relation with the concept it is modeling (i.e. by keeping the same shape as the generator set) and by conveying, in the case of rule-base systems,

information on the rule that generated it and its firing strength. In other words, it is possible to build CIT2 fuzzy systems that not only are able to solve, for example, classification problems, but that are also able to explain, in terms of the input space, how each endpoint of the interval centroid has been obtained. With a standard IT2 system this property is lost simply because in the defuzzification process, the ES that produce the endpoints don not carry any meaningful information on which rules played a role in their generation and why. Therefore, in IT2 systems an explanation in terms of the input space can not be provided for the centroid but only for the boundaries of the FOU of the fuzzy output of the system. The ability of CIT2 fuzzy systems to explain also the endpoints of the centroid, on the other hand, clearly represents a novelty and a progress for T2 FSs in the increasingly popular explainable artificial intelligence (XAI, [26]) field.

C. Efficiency

The main goal of Algorithm 1, is to provide a procedure to compute the inferencing and defuzzification processes described in this section. For now, the optimization of computational complexity has not been our focus. It is clear that the proposed algorithm is slower than the current IT2 inferencing and defuzzification methods. That is because after the evaluation of the whole rule-base, the output is a set of AESs (line 16, in Algorithm 1) that can be quite big in size: each rule can produce (line 9) a number of implication sets that, in the worst case, is equal to the size of the permutations of the AESs of the antecedents, multiplied by the cardinality of the DS of the consequent. Additionally, at line 16 we generate the unions of all the possible permutations of the AESs of the CIT2 resulting from the single rules. This union, generates a number of AES that grows as a double exponential, being

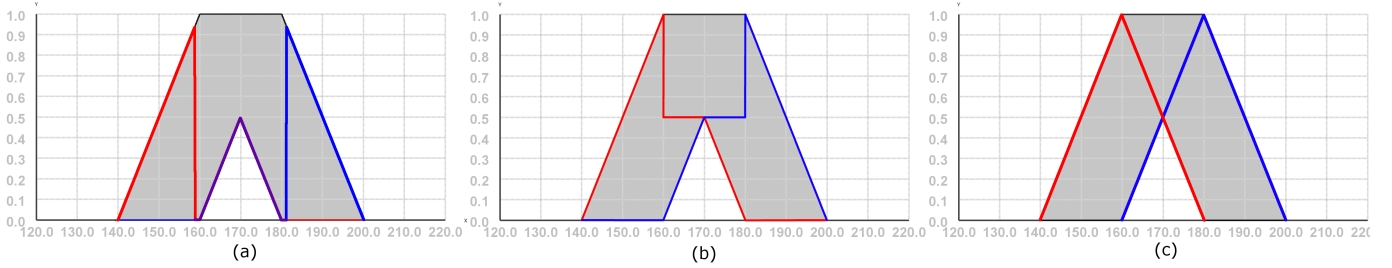


Fig. 13. ESs that determine the end-point values of the KM (a), W-CIT2 (b) and CIT2 centroid (c). In (a), the area where the 2 ESs overlap is coloured in purple.

$O(k^{n+1})^m$ where m is the number of rules, n the number of antecedents per rule and k the number of AES of each of the CIT2 involved.

Since this approach enumerates all the AESs to find the final defuzzified output, it is the analogous of the exhaustive defuzzification method rather than the KM one. In fact, the strength of the KM procedure is that it quickly identifies the ESs to be used for the left and right centroid values. On the contrary, in Algorithm 1 the AESs that give the left and right centroid value are found using a brute force approach, first building all the AESs of the total rule-base evaluation (line 16) and then finding among them the two that will give us the left and right centroid values (lines 18 and 19). For use in real-world problems, this approach is impractical because of its prohibitive computational complexity. For this reason, the alternative, much faster and practical defuzzification Algorithm 2 is proposed in Sec. VII. This algorithm is then used within the genetic framework described in Sec. VIII, in which it is applied on two well known real-world datasets and compared to the KM procedure.

VI. COMPARISON WITH A DIFFERENT CONSTRAINED APPROACH

In this section, the constrained representation presented in this paper will be compared to a different approach (that here will be called W-CIT2) proposed by Wu et al. in [11], [12]. They start from the observation that ESs have been used to obtain theoretical results such as the definition of uncertainty measures and are processed regardless of their shape.

However, the authors point out that in many fuzzy logic applications the MFs that are used are convex and normal. Consequently, they propose a constrained representation theorem that allows the definition of the FOU of well-shaped (see [12] for details) IT2 FSs by using only convex and normal ESs. They claim that this definition is more general than the one that only considers ES with the same shape and doesn't require any expert knowledge or data analysis to determine which shapes are meaningful in a given context. Using this new theorem, many constrained uncertainty measures (such as centroid, entropy and cardinality) are defined mathematically. In addition to that, the authors show how the convexity and normality constraints can be simply added to the KM algorithm to find the constrained centroid value of a well-shaped IT2 FS. Finally, the authors also state that this approach

can't be used in Mamdani systems since their outputs can be non-well-shaped.

The main difference between the representation theorem proposed in this paper and the W-CIT2 one is in the definition of the ESs that are considered acceptable. Even though it is true that the W-CIT2 theorem allows the presence of multiple shapes among the ESs, normality and convexity can be not sufficient and not necessary to obtain shapes that are meaningful, as also discussed in [30]. Those two properties alone, still do not guarantee there will be a meaningful connection between an ES and the concept it models. To support this claim, a comparison is provided between the ESs that determine the end-points of the W-CIT2, CIT2 and IT2 centroid with the KM procedure (Fig. 13). The set to defuzzify has been obtained starting from a triangular T1 MF as a generator set, using the approach described in this paper to build the FOU around it. The comparison shows how the ES used by the KM approach (Fig. 13 (a)) are both non-normal and non-convex. In addition to that, they could hardly represent any word or label. As a result, the meaningfulness and interpretability of the centroid value returned as an output decreases. On the other hand though, the KM algorithm can be applied to any IT2 FS, regardless of the approach used to obtain its FOU. The ESs used by the W-CIT2 approach, instead, are both normal and convex. However, also in this case the relation between the original T1 triangular shape (i.e. the one that has been used as a generator set) and the ESs is lost. Again, these sets would hardly model the same concept (e.g. medium height) from which we obtained the generator set. The ESs used by the CIT2 approach, instead, keep the same level of the interpretability as the generator set as they share its shape. The only difference between them is their location on the x-axis. From this experiment, we can conclude that normality and convexity alone may not be sufficient to guarantee the meaningfulness of a FS.

In addition to that, the fact that W-CIT2 FSs are not usable in Mamdani systems represents a significant limitation that can be overcome by the CIT2 definition provided in this paper, as shown in Sec. V.

VII. SAMPLING APPROACH FOR THE CIT2 CENTROID

As already discussed in Sec. V, the evaluation of the CIT2 centroid as described in Algorithm 1 is prohibitive due to the astronomical number of AESs that are examined to determine the defuzzified value. Therefore, even though the algorithm

Algorithm 2 CIT2 Sampling Algorithm

```

1: procedure CIT2 SAMPLING ALGORITHM (CIT2-FUZZY SYSTEM  $S$ , INT TOTAL_SAMPLES, INPUT  $[x_1, \dots, x_n]$ )
2:   Set centroids=new Set(); ▷ This set will contain the centroids of the sampled AES
3:   HashMap<CIT2, T1MF> cit2_to_aes; ▷ Mapping each CIT2 into one of its AES
4:   for int index=0; index<total_samples; index++ do ▷ Each iteration generates 1 sample
5:     T1_Rulebase t1_rulebase=new T1_Rulebase();
6:     cit2_to_aes=new HashMap<>(); ▷ For each sample, new AESs are chosen
7:     for each CIT2_Rule  $R_i \in S$  do
8:       T1_Antecedents antecedents=new T1_Antecedents();
9:       T1_Consequents consequents=new T1_Consequents();
10:      for each CIT2_Antecedent curr_ant  $\in R_i$  do
11:        T1MF current_random_aes;
12:        ▷ If a random AES for this set has never been generated before in this iteration...
13:        if cit2_to_aes.get(curr_ant)==null then
14:          ▷ ...generate it and add it to the Map
15:          cit2_to_aes.put(curr_ant, curr_ant.randomAES());
16:          ▷ By doing this, if a set appears multiple times as an antecedent in different rules, the same AES will
17:          be used to replace it in the current iteration
18:        end if
19:        antecedents.add(cit2_to_aes.get(curr_ant)); ▷ Add the AES to the rule
20:      end for
21:      for each CIT2_Consequent curr_cons  $\in R_i$  do
22:        T1MF current_random_aes;
23:        ▷ If a random AES for this set has never been generated before in this iteration...
24:        if cit2_to_aes.get(curr_cons)==null then
25:          ▷ ...generate it and add it to the Map
26:          cit2_to_aes.put(curr_cons, curr_cons.randomAES());
27:        end if
28:        consequents.add(curr_cons.randomAES());
29:      end for
30:      t1_rulebase.addRule(new T1_Rule(T1_Antecedents, T1_Consequents));
31:    end for
32:    for each Input input $_i \in$  t1_rulebase do
33:      input $_i$ .setValue(x $_i$ );
34:    end for
35:    ▷The rulebase produces the sampled AES and we need its centroid value
36:    centroids.add(t1_rulebase.run().getCentroid());
37:  end for
38:  return (centroids.minValue(), centroids.maxValue());
39: end procedure

```

proposed earlier in this paper is theoretically correct for the computation of the CIT2 centroid, it is not usable in practice for real world problems. Conceptually, the problem is very similar to the one that is faced when exhaustive defuzzification is applied to T2 FSs. In that context, many approximation algorithms have been proposed to overcome the computational complexity of the exhaustive defuzzification. One of them is the sampling method [31]. The intuitive idea is that each of the ESs in a T2 FS only has a minimal contribution to the final result, therefore generating a random sample of the ESs is a good and efficient way to obtain an approximation of the actual centroid value, as showed in [32]. In this case, we apply the same concept to sample a fixed number of AESs to determine the constrained centroid. A sample, is obtained by replacing each CIT2 FS in the rulebase with one of its AES

chosen at random (rather than replacing each set with all its AES, as in Algorithm 1).

The fuzzy output of the T1 system obtained by carrying out all the substitutions will produce a single sampled AES. As a consequence of that, only a subset of all the AES and is generated, making this approach an approximation algorithm. Once the number of desired samples has been obtained, the AES are defuzzified and the lowest and highest centroid values among them will determine respectively the left and right endpoint of the constrained centroid.

Conceptually, the following steps are used to produce a single sampled AES of the CIT2 fuzzy output:

- For each set \check{A} involved in the FLS:
 - Generate a random number k within its DS
 - Use k to shift the generator set of \check{A} along the x-axis,

obtaining E , an AES of \check{A} ; remembering that given a function $f(x)$ its translated version by a factor k along the x-axis can be written as $f'(x) = f(x) - k$, this step can be done in constant time without the need to store all the AES to choose one randomly

– Loop through all the rules and replace \check{A} with E

- Once all the CIT2 FS have been replaced with a random AES, a T1 rulebase is generated
- The fuzzy inferred result of the rule-base represents a sampled AES

The output interpretability offered by CIT2 FLS is given by the process used to produce the AES. In fact, each of them represents a T1 fuzzy output and as such keeps all the interpretability properties that belong to the outputs of T1 FLSs: the shapes of the consequent set involved in the rules are clearly identifiable together with the firing strengths used for the inference operator (e.g. see Fig. 9(a), 10(a), 13(a)). These properties, also make possible a direct connection between the endpoints of the interval centroid and the rules that were used in its generation.

The pseudo-code (mainly written following OOP conventions) of the sampling method is described in Algorithm 2.

Other than the reduction in the computational cost, the other main advantage of this approach is its applicability to systems in which the CIT2 FSs involved have a continuous DS, i.e. the number of AESs per CIT2 FS is infinite. In fact, Algorithm 1 only works with a discrete number of AESs and may therefore require an additional discretization step. With the sampling approach each CIT2 FS involved in the rule can be easily substituted with one of its AES by shifting its generator set by a random value in the DS during the conversion step (mentioned above) of the CIT2 rule into a T1 one.

VIII. PRACTICAL APPLICATION

In this section, a framework for the automatic learning of CIT2 fuzzy systems will be described and applied to two real-world classification problems. The aim is not to compare this learning method to other approaches proposed in literature in terms of performances, but rather to present a possible way of generating CIT2 fuzzy systems and show a practical application of these new fuzzy sets and their inference framework described so far.

Classification problems have been chosen because they represent one of the contexts in which interpretability and especially explainability play a crucial role. In many applications, in fact, knowing both the output (the interval centroid) and how it has been obtained (i.e. which rules and which inputs determined the ES that produced the endpoints) is of great value and it is the main reason for the emergence of the new XAI field.

A. Genetic CIT2 fuzzy systems

Genetic algorithms have been widely used for the automatic generation and optimization of fuzzy systems [33] since they allow for the creation of both the rule-base and the MFs without the need of any expert knowledge. Even though these systems are obtained through machine learning techniques,

they can maintain the typical interpretability of fuzzy logic systems as long as they contain a reasonably small number of rules and it is possible to give a linguistic label to the MFs involved [34]. The genetic approach proposed for the generation of CIT2 fuzzy systems, is based on the architecture described in [35]. Each of the input variables of the system is partitioned in 3 triangular MFs. The center of each triangular generator set for the antecedent CIT2 FSs is determined using the well known fuzzy C-Means clustering algorithm (FCM) [36] on each input variable. The end-points of the triangles are the center of the previous and next clusters, if they exist, or the closest end-point of the UOD increased by 10% of the UOD size, so that every point in the UOD belongs to at least one of the MFs with a membership value greater than 0. The continuous DS is an interval $[-c, c]$, $c > 0$ with $2c = 5\%$ of the distance between the starting and end point of each triangular generator set. The output variable is partitioned with a number of CIT2 FS equal of the number of classes in the problem. Each of them is given an integer index from 0 to the number of classes involved. The index represents the peak of their triangular generator set while the start and end point of the triangles are obtained respectively subtracting and adding 1 to their peak points. The DS for all the CIT2 MF partitioning the output is an interval $[-c, c]$, $c > 0$ with $2c = 10\%$ of the UOD. Once the MFs are determined, there is a first evolutionary stage to generate the rule-base of the system. During this process, the MFs are not changed. The number of rules is fixed (as shown in [35], redundant rules can be eliminated with an additional stage) and each chromosome codes an entire rule-base. With n input variables, each rule is coded with a set of $n + 1$ integers. Each gene p_i represents the index of the MF to use for the $i - th$ antecedent or for the consequent, if $i = n + 1$. A value of -1 for $p_i, i \leq n$, indicates that the $i - th$ input must not be included in the rule p_i belongs to. A sequence of encoded rules represents a rule-base.

TABLE I
PARAMETERS USED FOR THE LEARNING ARCHITECTURE

Parameters	Values
Population size	100
Iteration limit	100 per stage
Crossover	Single-Point
Crossover rate	0.7
Elitism	5%
Fuzziness in FCM	2.0
Mutation rate	1/chromosome_size
Fitness function	Accuracy value
Memberships per variable	3
Fuzzy Rules per chromosome	Fixed, 10
Number of samples in CIT2 centroid	50
Random distribution for the random sampling	Uniform
Discretization points for AES defuzzification	100

At the end of the first stage, the fittest chromosome is returned. The rule-base encoded by this chromosome is passed to the second stage of the learning process, with the goal of optimizing the MFs involved in the system. Each triangular CIT2 MF is encoded with 4 real numbers: 3 modelling the generator set (starting point, center and ending point of

the triangle) and one representing the size of the DS as a percentage of the UOD. Thanks to the way CIT2 MFs are built starting from a T1 generator set, the encoding of CIT2 MFs only requires 1 additional parameter with respect to their T1 counterpart. That is because the upper and lower MFs bounding the FOU of the set, are determined from the T1 generator shape and the DS. Standard IT2 representations, instead, may require up to twice the number of parameters of their T1 counterpart to fully represent the FOU and its bounds. The optimized rule-base obtained at the end of the second stage is then returned as the final output of the learning process. The architecture is summarised in Fig. 14. For more information on the tuning and learning process, please refer to [35].

TABLE II
RESULTS OF THE GENETIC CIT2 FUZZY SYSTEM WITH TWO DIFFERENT
DEFUZZIFICATION APPROACHES

	CIT2	IT2(KM)	CIT2 Time	IT2 Time
Iris	96.0%	94.667%	4h5m	1h11m
New-Thyroid	91.167%	91.667%	6h19m	1h31m

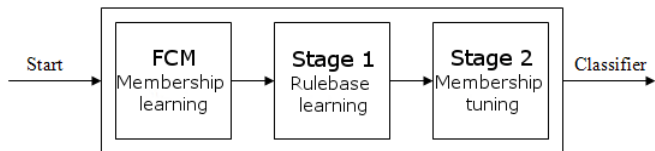


Fig. 14. The learning architecture used in this paper. Adapted from [35]

B. Application on real data-sets

The genetic architecture described above has been tested on two real world classification problems using two well known data-sets: iris [37] and new-thyroid [38]. The 10-fold cross validation method has been used to evaluate the CIT2 fuzzy systems; both data-sets, including the train and test partitions of each cross validation iteration, are publicly available on the KEEL website [39]. In both stages a single-point crossover has been used and the fitness function has been defined as the accuracy value of the rule-base encoded in the chromosome. A more detailed list of the parameters used in the optimization can be found in Table I. The optimization has been carried out twice, once using the CIT2 sampling method with 50 samples to defuzzify the output and once using the implementation of the KM iterative procedure implemented in the Java library Juzzy [40]. The architecture has been implemented in Java using multi-thread computation on an i7-7600U CPU. The average results of both approaches and their running times for the 10 runs are reported in Table II. It can be seen that the execution time of the CIT2 systems, featuring Algorithm 2, are higher than the IT2 systems. However, these execution times represent approximately 10^7 individual defuzzification operations throughout the optimization process — i.e. each individual CIT2 defuzzification using Algorithm 2 takes around 1.5 milliseconds using multi-threading to generate the samples. Whilst not as efficient as current IT2 defuzzification algorithms, this is

clearly usable in real world applications, particularly decision problems.

As it is possible to see, the two approaches give similar results and perform well on both the data-sets analyzed. Therefore, to determine if and under what conditions one of the two defuzzification methods gives superior results more experiments are required, with a bigger number of data-sets and a statistical evaluation of their performances. To demonstrate the superior interpretability and explainability of the CIT2 approach, in Fig. 15 are shown the ES used to determine the right end-point of the constrained (a) and “standard IT2” (b) centroid generated by the KM procedure. Those ES have been obtained as the result of the defuzzification of the output of a CIT2 FLS generated through the learning framework described in this section. As discussed in Sec. III, the AES selected by the CIT2 approach, provides a clearer understanding of the final system output, giving an intuitive idea of how the centroid value is obtained since, just like any T1 fuzzy output, it is still possible to identify the shapes of the consequent MFs and see the respective firing levels. Additionally, the firing strengths can be traced back to the rules and the inputs that generated the endpoints. The ability to produce explanations for each of the system outputs, *together* with the interpretable rule-based structure (characteristic of any FLS) make CIT2 FLS a valid alternative to IT2 for the development of FLS in the XAI field.

Currently running times seem to be the main drawback of this approach. In fact, in both the tests the IT2 approach with the KM procedure has proven to be roughly 3.5-4 times faster than the CIT2 one. In future works, we plan on developing new and faster defuzzification methods to address this issue.

IX. CONCLUSION

In this paper, we have fully formalized constrained interval type-2 (CIT2), showing how they can be obtained starting from a T1 FS with uncertainty on its exact location on the x-axis. The main idea behind CIT2 FSs is to produce a representation that considers only the ESs that have meaningful shape for a given concept; these embedded sets, called *acceptable* (AES), can then be used to define the FOU of our CIT2 FS and CIT2 fuzzy operators. The use of AESs rather than their unconstrained version, guarantees that CIT2 operators will only process embedded sets with a meaningful shape, increasing the interpretability of their output (as discussed in Sec. III, VIII.B).

Formal definitions of CIT2 FSs and AESs have been provided, together with the formulation of a new constrained representation theorem (Theorem 1). This allowed us to define all the main CIT2 operators, including the centroid defuzzification, by working only with “meaningful” ESs. Finally, a full inference framework has been presented for a CIT2 fuzzy system together with a defuzzification procedure. As a test case, a genetic architecture for the generation of CIT2 fuzzy systems has been described and applied to two real world datasets. The preliminary results, presented here, show how the performances of the CIT2 approach are comparable to the ones obtained from the IT2 one, with the CIT2 system outputs presenting a higher level of interpretability. On the other hand,

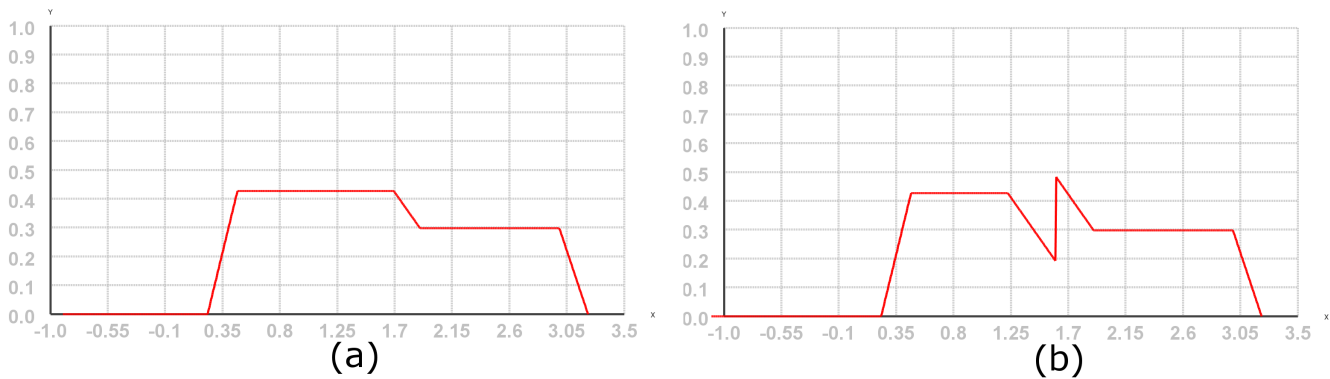


Fig. 15. ESs that determine the right end-point value of the CIT2 (a) and KM (b) centroid in a CIT2 system obtained through the genetic architecture described in this paper.

CIT2 have been shown to be slower, requiring approximately 4 times more time than their IT2 counterpart to complete the learning process.

X. FUTURE WORK

Now that formal definitions have been given for CIT2 inferencing and defuzzification procedures, a considerable amount of research work needs to be done to fully understand their potential for practical applications. We speculate that the properties of CIT2 FSs, such as the fact that all the AES share the same shape, can be exploited for deploying a faster inference and defuzzification process. Also understanding how the number of AES used in Algorithm 1 and Algorithm 2 affect the centroid value and its convergence is of significant importance and will be examined in future works. Additionally, it would be useful to have a more thorough comparison between IT2 and CIT2, using more real world cases from different scenarios. Understanding in which cases and why the results of IT2 and CIT2 systems differ, would help designers choose between the two approaches in their context. That is one of the main things we plan on working on in the near future. We also expect that the differences in the interpretability of the ES of the CIT2 and IT2 representation require a more formal comparison as this is a subject that is present in many research works but has never been deeply studied and formalized. While this paper is only focused on CIT2 rule-based systems, it is necessary to evaluate the meaningfulness and interpretability of these new sets more thoroughly, in different applications. For example, we believe that the use of CIT2 FSs would be very suitable for the fuzzy linguistic and the computing with words approaches [41]. However, to be applied in those contexts, similarity measures for CIT2 FSs need to be defined. Finally, the CIT2 definitions can be extended to include general T2 FSs, in order to have general constrained T2 FSs, overcoming the problems discussed in Sec. IV.

REFERENCES

- [1] L. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning," *Information Sciences*, vol. 8, no. 3, pp. 199 – 249, 1975.
- [2] 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.
- [3] L. Cervantes and O. Castillo, "Type-2 fuzzy logic aggregation of multiple fuzzy controllers for airplane flight control," *Inf. Sci.*, vol. 324, no. C, pp. 247–256, Dec. 2015.
- [4] A. H. M. Pimenta and H. d. A. Camargo, "Genetic interval type-2 fuzzy classifier generation: A comparative approach," in *2010 Eleventh Brazilian Symposium on Neural Networks*, Oct 2010, pp. 194–199.
- [5] J. M. Mendel and R. John, "Footprint of uncertainty and its importance to type-2 fuzzy sets," in *Proceedings 6th IASTED Int'l. Conf. on Artificial Intelligence and Soft Computing (ASC 2002)*, July 2002, pp. 587 – 592.
- [6] N. N. Karnik and J. M. Mendel, "Centroid of a type-2 fuzzy set," *Information Sciences*, vol. 132, no. 1-4, pp. 195–220, 2001.
- [7] D. Wu and J. M. Mendel, "Enhanced karnik–mendel algorithms," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 4, pp. 923–934, 2009.
- [8] 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, Apr 2002.
- [9] J. M. Garibaldi and S. Guadarrama, "Constrained type-2 fuzzy sets," in *Advances in Type-2 Fuzzy Logic Systems (T2FUZZ), 2011 IEEE Symposium on*. IEEE, 2011, pp. 66–73.
- [10] P. D'Alterio, J. M. Garibaldi, and A. Pourabdollah, "Exploring constrained type-2 fuzzy sets," in *2018 IEEE World Congress on Computational Intelligence (WCCI 2018)*, July 2018.
- [11] D. Wu, "A constrained representation theorem for interval type-2 fuzzy sets using convex and normal embedded type-1 fuzzy sets, and its application to centroid computation," *Proceedings of World Conference on Soft Computing*, 2011.
- [12] D. Wu, H. Zhang, and J. Huang, "A constrained representation theorem for well-shaped interval type-2 fuzzy sets, and the corresponding constrained uncertainty measures," *IEEE Transactions on Fuzzy Systems*, pp. 1–1, 2018.
- [13] 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, Dec 2006.
- [14] J. M. Mendel, M. R. Rajati, and P. Sussner, "On clarifying some definitions and notations used for type-2 fuzzy sets as well as some recommended changes," *Information Sciences*, vol. 340-341, pp. 337 – 345, 2016.
- [15] N. N. Karnik, J. M. Mendel, and Q. Liang, "Type-2 fuzzy logic systems," *IEEE transactions on Fuzzy Systems*, vol. 7, no. 6, pp. 643–658, 1999.
- [16] 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.
- [17] R. Sepúlveda, O. Castillo, P. Melin, A. Rodríguez-Díaz, and O. Montiel, "Experimental study of intelligent controllers under uncertainty using type-1 and type-2 fuzzy logic," *Information Sciences*, vol. 177, no. 10, pp. 2023 – 2048, 2007, including Special Issue on Hybrid Intelligent Systems.
- [18] P. A. S. Birkin and J. M. Garibaldi, "A comparison of type-1 and type-2 fuzzy controllers in a micro-robot context," in *2009 IEEE International Conference on Fuzzy Systems*, Aug 2009, pp. 1857–1862.

- [19] S. Coupland, M. A. Gongora, R. John, and K. Wills, "A comparative study of fuzzy logic controllers for autonomous robots." in *IPMU 2006 Conference*, 2006.
- [20] C. Lynch, H. Hagra, and V. Callaghan, "Embedded type-2 flc for real-time speed control of marine and traction diesel engines," in *The 14th IEEE International Conference on Fuzzy Systems, 2005. FUZZ '05.*, May 2005, pp. 347–352.
- [21] D. Wu and W. W. Tan, "A type-2 fuzzy logic controller for the liquid-level process," in *2004 IEEE International Conference on Fuzzy Systems (IEEE Cat. No.04CH37542)*, vol. 2, July 2004, pp. 953–958 vol.2.
- [22] L. Amador-Angulo, O. Castillo, and M. Pulido, "Comparison of fuzzy controllers for the water tank with type-1 and type-2 fuzzy logic," in *2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS)*, June 2013, pp. 1062–1067.
- [23] 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.
- [24] J. Aisbett, J. T. Rickard, and D. Morgenthaler, "Multivariate modeling and type-2 fuzzy sets," *Fuzzy Sets and Systems*, vol. 163, no. 1, pp. 78 – 95, 2011, theme: Classification and Modelling.
- [25] H. Hagra, "Developing a type-2 flc through embedded type-1 flcs," in *Fuzzy Systems, 2008. FUZZ-IEEE 2008.(IEEE World Congress on Computational Intelligence). IEEE International Conference on.* IEEE, 2008, pp. 148–155.
- [26] D. Gunning, "Explainable artificial intelligence (xai)," *Defense Advanced Research Projects Agency (DARPA)*, nd Web, 2017.
- [27] D. Castelvecchi, "Can we open the black box of ai?" *Nature News*, vol. 538, no. 7623, p. 20, 2016.
- [28] J. M. Mendel, "Computing with words and its relationships with fuzzistics," *Information Sciences*, vol. 177, no. 4, pp. 988 – 1006, 2007.
- [29] A. Norwich and I. Turksen, "A model for the measurement of membership and the consequences of its empirical implementation," *Fuzzy Sets and Systems*, vol. 12, no. 1, pp. 1 – 25, 1984.
- [30] P. D'Alterio, J. M. Garibaldi, and R. John, "On the concept of meaningfulness in constrained type-2 fuzzy sets," in *International Conference on Fuzzy Systems (FUZZ-IEEE 2019)*, 2019.
- [31] S. Greenfield, R. John, and S. Coupland, "A novel sampling method for type-2 defuzzification," in *Proc. UKCI 2005*, 09 2005, pp. 120–127.
- [32] S. Greenfield, F. Chiclana, R. John, and S. Coupland, "The sampling method of defuzzification for type-2 fuzzy sets: Experimental evaluation," *Information Sciences*, vol. 189, pp. 77 – 92, 2012.
- [33] O. Cordon, F. Gomide, F. Herrera, F. Hoffmann, and L. Magdalena, "Ten years of genetic fuzzy systems: current framework and new trends," *Fuzzy Sets and Systems*, vol. 141, no. 1, pp. 5 – 31, 2004.
- [34] A. Fernandez, F. Herrera, O. Cordon, M. J. del Jesus, and F. Marcelloni, "Evolutionary fuzzy systems for explainable artificial intelligence: Why, when, what for, and where to?" *IEEE Computational Intelligence Magazine*, vol. 14, no. 1, pp. 69–81, 2019.
- [35] A. H. M. Pimenta and H. A. Camargo, "Interval type-2 fuzzy classifier design using genetic algorithms," in *International Conference on Fuzzy Systems*, July 2010, pp. 1–7.
- [36] J. C. Bezdek, R. Ehrlich, and W. Full, "Fcm: The fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol. 10, no. 2, pp. 191 – 203, 1984.
- [37] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.
- [38] D. Coomans, I. Broeckaert, M. Jonckheer, and D. L. Massart, "Comparison of multivariate discrimination techniques for clinical data—application to the thyroid functional state," *Methods of information in medicine*, vol. 22, pp. 93–101, 05 1983.
- [39] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework," *Journal of Multiple-Valued Logic & Soft Computing*, vol. 17, 2011.
- [40] C. Wagner, "Juzzy - a java based toolkit for type-2 fuzzy logic," in *2013 IEEE Symposium on Advances in Type-2 Fuzzy Logic Systems (T2FUZZ)*, April 2013, pp. 45–52.
- [41] L. A. Zadeh, "Fuzzy logic = computing with words," *IEEE Transactions on Fuzzy Systems*, vol. 4, no. 2, pp. 103–111, May 1996.