

# A Fast Inference and Type-Reduction Process for Constrained Interval Type-2 Fuzzy Systems

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

**Abstract**—Constrained interval type-2 (CIT2) fuzzy sets have been introduced to preserve interpretability when moving from type-1 (T1) to interval type-2 (IT2) membership functions. Although they can be used to produce type-2 fuzzy systems with enhanced explainability, so far, the latter comes at the expense of high computational cost. Specifically, the exhaustive type-reduction method for CIT2 Mamdani systems has been shown to be too slow to be used in practical applications and even the current approximation procedure is much slower than modern type-reduction algorithms used for IT2 fuzzy sets. In this paper, a novel type-reduction procedure for CIT2 sets is presented, based on the concept of *switch indices*. The algorithm is applied on a real-world classification problem and compared to other type-reduction approaches used in IT2 and CIT2 systems. In the case studies presented, the new algorithm is significantly faster than the exhaustive and sampling CIT2 approaches while keeping the high level of interpretability of the type-reduction operation that characterizes CIT2 fuzzy sets.

**Index Terms**—Explainable AI, explainable fuzzy systems, explainable type-2 fuzzy systems, constrained type-2 fuzzy sets, switch indices.

## I. INTRODUCTION

In recent years, there has been a growing trend for more explainable and transparent AI systems that has led to the creation of the new research field of *explainable artificial intelligence* (XAI) [1], [2]. In contrast to other machine learning approaches (e.g. neural networks) that work like black boxes, fuzzy logic (FL) can be used to build intelligent systems in which the reasoning behind the predictions made by the model can be explained [3] through their underlying rules. The rule-based structure together with the use of linguistic labels [4], allow for the creation of fuzzy logic systems (FLS) that not only give reliable predictions in AI tasks but also have a high level of understandability for both an expert and non-expert audience. For this reason, FL represents a valuable tool in XAI which has already been successfully applied in some real-world problems [5], [6].

While the interpretability of type-1 (T1) FLS has already been examined in some research (e.g. [3]), the same cannot be said for interval type-2 (IT2) [7] and general type-2 (GT2) [8] fuzzy logic systems. Although the rule-base structure remains the same, the addition of a third dimension in the membership functions and the significant difference in the way some important measures are computed (e.g. the centroid [9]), may pose a serious restriction on the level of explainability obtainable from these systems when compared to their type-1 counterparts. Moreover, keeping the semantic meaning between a GT2 or IT2 fuzzy set and the linguistic

label it represents may be challenging, as the footprint of uncertainty (FOU, [10]) contains embedded sets (ES) that in some contexts may represent implausible relations between the data [11], [12]. These issues have led to the creation of a restricted version of type-2 fuzzy sets called constrained type-2 (CT2) fuzzy sets [13] and constrained interval type-2 (CIT2) [14], [15] fuzzy sets. These are special cases of GT2 and IT2 FSs obtained from a type-1 generator set (GS) representing a given linguistic label. Specifically, CIT2 FS impose restrictions on the shape of the FOU and the embedded sets that lead to more interpretable FLS compared to their IT2 counterparts [14].

A recent paper by D'Alterio et al [14] has shown how the higher level of interpretability comes at the cost of higher computational complexity in the type-reduction processes. In fact, while type-reducing a CIT2 set is trivial, the same operation becomes very computationally expensive for the inference. Specifically, the exhaustive procedure to type-reduce the output of a CIT2 Mamdani inference system has been shown to be impractical for real world applications due to its prohibitive computational cost. Even the approximation procedure (termed the CIT2 sampling method [14]), introduced for faster computation, has been shown to be significantly slower than the well-known Karnik-Mendel (KM) [9] algorithm for IT2 FSs.

The contribution of this paper is a refined inferencing mechanism with an associated novel type-reduction approach which enables the much faster computation of CIT2 Mamdani inference systems. The novel procedure efficiently and deterministically selects a small number of appropriate embedded sets from which it produces the final type-reduced set. This reduction in the search space makes the approach presented in this paper significantly faster than the exhaustive and sampling CIT2 type-reduction algorithms [14] while maintaining comparable outputs and keeping the high level of interpretability that characterizes CIT2 fuzzy sets.

The rest of the paper is organized as follows: after a brief introduction to CIT2 fuzzy sets (Section II), the novel inference and type-reduction technique is described and then formalized (Section III). Multiple experiments are carried out to compare this new algorithm with KM, its enhanced version (EKM, [16]) and the CIT2 sampling method to show the significant run time improvements (Section IV). Finally, the approach is applied to a real world classification problem, in which the explainability and accuracy of its classifications is discussed with respect to the KM approach (Section IV-C).

## II. CONSTRAINED INTERVAL TYPE-2 FUZZY SETS

Constrained type-2 fuzzy sets were first introduced by Garibaldi and Guadarrama in 2011 [13] and more recently have been detailed in the specific instance of *constrained interval type-2* sets [14]. They represent a special case of IT2 FSs since they impose additional mathematical restrictions on the shape of the footprint of uncertainty (FOU) and the shape of the embedded sets contained within them.

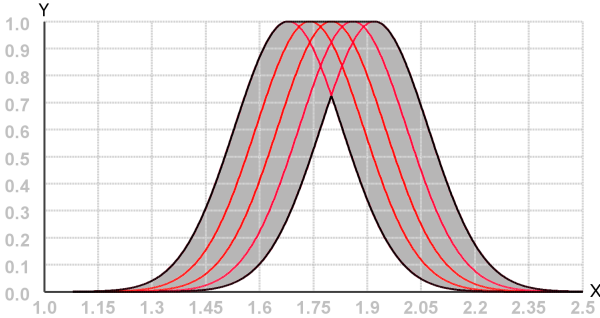


Fig. 1. Different Gaussian membership functions modeling medium height (in red) and the FOU in which they are embedded (in gray).

Intuitively, a CIT2 fuzzy set models a T1 fuzzy set (named a *generator set*) with uncertainty in its exact location on the x-axis. This situation, for example, may arise when different people are asked to place a membership function on the x-axis to model a given concept, such as *medium height*, as shown by the Gaussian memberships in red in Fig. 1.

TABLE I  
TABLE OF NOTATIONS

| Abbreviation     | Meaning  |
|------------------|--|
| FS               | Fuzzy Set  |
| FLS              | Fuzzy Logic System                                 |
| T1               | Type-1   |
| T2               | Type-2 [8]   |
| IT2              | Interval Type-2 [7]                                |
| $\tilde{A}$      | T2 fuzzy set, named A [8]                          |
| FOU              | Footprint Of Uncertainty [8]                       |
| ES               | Embedded Set [8]                                   |
| CT2              | Constrained Type-2 [13]                            |
| CIT2             | Constrained Interval Type-2 [14]                   |
| $\check{A}$      | CIT2 fuzzy set, named A                            |
| GS               | Generator Set [13]                                 |
| DS               | Displacement Set [14]                              |
| AES              | Acceptable Embedded Set [14]                       |
| CAES             | Collection of type-1 Acceptable Embedded Sets [14] |
| $\widehat{CAES}$ | Collection of type-2 Acceptable Embedded Sets [14] |

In this example, the CIT2 approach builds the FOU by ‘blurring’ the area between the left-most and right-most Gaussian membership and considers as *acceptable* only the embedded sets that have the same Gaussian shape as the generator used in the experiment. By doing so, sets that due to their shape could not model the concept of *medium height* are not included in the FOU and do not play a role in the output of fuzzy operations, such as type-reduction. In this example, all the Gaussian MFs in Fig. 1 are acceptable embedded sets.

The collection containing all the acceptable embedded sets is named *collection of acceptable embedded sets*.

Table I contains the type-2 and constrained interval type-2 notations used throughout the paper. For additional details, the reader can refer to the cited works.

## III. A NOVEL CONSTRAINED CENTROID DEFUZZIFICATION METHOD FOR MAMDANI CIT2 FUZZY SYSTEMS

Type-reducing a CIT2 fuzzy set is a trivial task: since all the AES share the same shape, the left-most and right-most ones will produce the two end-points of the type-reduced set.

However, the same operation for the inference in Mamdani systems is non-trivial and computationally expensive [14]. In fact, the AESs of the fired output of the system do not necessarily share the same shape anymore, as a consequence of the inference. This phenomenon is shown in Fig. 2 where the AES (which before the implication had a triangular shape) have been ‘truncated’ at different heights as a result of the implication (min) operator. In this situation, determining the endpoint of the type-reduced set is no longer trivial.

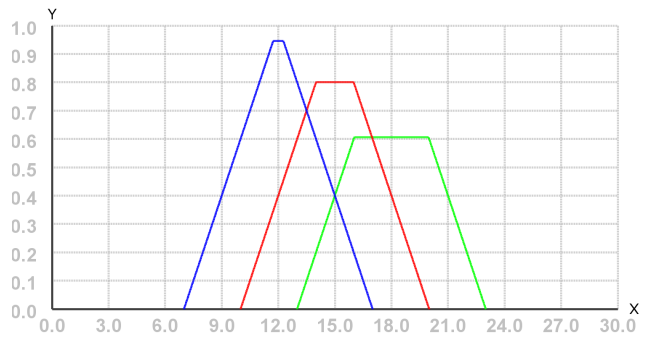


Fig. 2. Some of the AES generated from a CIT2 rule where the consequent has a triangular GS

The exhaustive approach [14] and its approximation procedure named the sampling method [14], have been shown to be significantly slower than current type-reduction algorithms for IT2 Mamdani systems, making the use of CIT2 FLS in real world scenarios impractical.

In this paper, a novel algorithm is proposed that selects a subset of AESs with specific criteria to compute the endpoints of the type-reduced set. In the experiments carried out and illustrated in Sec. IV, this new approach has been shown to be at least 7.5 times faster than both the exhaustive and sampling algorithms while providing comparable results in terms of the endpoints produced.

### A. Informal description

To type-reduce an IT2 FS, the KM algorithm finds the ESs with respectively the lowest and highest centroid. These two values, determine the endpoints of the interval that represents the type-reduced set. The MFs of these two ESs can be written using the lower and upper-bound MFs of the FOU they are embedded into:

$$\mu_L(x) = \begin{cases} \bar{\mu}_{\tilde{A}}(x), & x \leq S_L \\ \underline{\mu}_{\tilde{A}}(x), & x > S_L \end{cases}$$

$$\mu_R(x) = \begin{cases} \mu_{\check{A}}(x), & x \leq S_R \\ \check{\mu}_{\check{A}}(x), & x > S_R \end{cases}$$

where  $\mu_L$  and  $\mu_R$  are the MF of the ESs determining the endpoints,  $\check{A}$  is the IT2 FS they belong to,  $S_L$  and  $S_R$  are two values in the universe of discourse (UOD), called respectively left and right *switch point*. Informally, these two ES, ‘coincide’ with one of the two boundaries up to the switch point and then *switch* to the other boundary of the FOU. In the general case, if this approach is used to defuzzify a CIT2 FS, the ESs found by the KM algorithm would not be one of the AES (i.e. they would not have a *meaningful* shape).

As a result of the Mamdani CIT2 inferencing process (for more details, see [14]), the implication operator (min) is repeatedly applied to all the T1 AES of each consequent using all the values in the firing interval of the rule. An example of this operation is shown in Fig. 2, where the consequent before the implication had a triangular GS (i.e. the AES were triangles before they were “truncated”).

The idea in this novel approach is to use a binary choice for the implication operator for each of the AES. Rather than “truncating” them at all the possible values in the firing interval, each of the AES can be truncated either at the minimum or at the maximum value of the firing interval. For example, for a rule with a firing strength  $[a, b]$ , each AES of the consequent set can be truncated only at the values  $a$  or  $b$  rather than any of the values in  $[a, b]$ . As a consequence of that, the red set in Fig. 2, for example, would not be considered. Additionally, in the determination of the left endpoint of the type-reduced set, only the leftmost AES of each MF is considered while only the rightmost AES is used when computing the right endpoint.

Each of the consequent MF is given an ordinal index based on its position. The aim is to choose an integer value  $i$  such that for all the consequent MF with an index value smaller than  $i$ , a specific endpoint of the firing interval is used during the inference (e.g. the lower value); for all the MFs with an index value greater or equal to  $i$ , instead, the firing value used *switches*, so the other endpoint of the firing interval is used (e.g. if the lower value was used for the indices smaller than  $i$ , now the upper value is used). Hence,  $i$  is called *switch index*.

The goal of the new algorithm is to find the *switch indices* that produce the two sets with the maximum and minimum centroid value. Just like the switch points, the two switch indices can differ, respectively for the generation of the right and left endpoints of the type-reduced set.

### B. Speeding up CIT2 Mamdani inference

As described above, for the exhaustive or sampling type-reduction methods to be used, each CIT2 rule has to produce not just a firing interval but rather a set of firing values that are then used to carry out the implication on the AESs of a consequent set of each rule. For example, in Fig. 2, three distinct firing values (i.e. the three different heights at which the sets have been ‘truncated’) are used. With the novel approach introduced in this paper, however, only the endpoints of the firing strength of each CIT2 rule are needed.

This subsection introduces a theorem that allows the firing strengths to be quickly determined in a way that is analogous to that used for IT2 rules. Specifically, to compute the endpoints of the firing strength of a CIT2 rule, it is sufficient to work with the boundary functions of all the CIT2 sets involved in the rule.

**Theorem 1.** *Given a CIT2 rule (i.e. a fuzzy rule in which all of the fuzzy sets involved are CIT2 FSs):*

$$\text{IF } x_1 \text{ IS } \check{A}_1 \text{ AND... AND } x_i \text{ IS } \check{A}_i \text{ THEN } y \text{ IS } \check{A}_{i+1} \quad (1)$$

*the firing interval of the rule can be compute using only the upper-bound and lower-bound MFs  $\check{\mu}_{\check{A}}, \check{\underline{\mu}}_{\check{A}}$  of the CIT2 FS  $\check{A}_1, \dots, \check{A}_{i+1}$ .*

The proof of this theorem can be found in the Supplemental Material. The boundary functions  $\check{\mu}_{\check{A}}, \check{\underline{\mu}}_{\check{A}}$  of a CIT2 fuzzy set  $\check{A}$  are defined in the same way as the boundary functions of an IT2 fuzzy set [14], i.e. they represent the boundaries of the FOU. Theorem 1 leads to the same results that are obtained when one uses IT2 fuzzy sets [7]. The reason why Theorem 1 has to be proven again is in the different representation between CIT2 and IT2 fuzzy sets. In the IT2 case, the *representation theorem* holds [8], i.e. each IT2 fuzzy set can be represented as the union of its type-2 embedded sets; for CIT2 fuzzy sets, instead, the *constrained* representation theorem holds, i.e. a CIT2 fuzzy set can be represented as the union of its *acceptable* embedded sets. Since the collection of acceptable embedded sets is a subset of all the embedded sets, all the theorems for IT2 sets that make use of the embedded sets need to be proven again for CIT2 fuzzy sets showing that the same results hold when only acceptable embedded sets are considered.

Although Theorem 1 is *one* of the reasons behind the improved run-times of the novel algorithm, this way of computing the firing interval of CIT2 rules cannot be used by the exhaustive or sampling method. In fact, as discussed in the first paragraph of this Subsection, these algorithms require a discrete set of firing values (and not just the endpoints of the firing interval) to determine the type-reduced set. Additionally, the analysis of the computational complexity carried out in Sec. III-E does not include the computation of the firing of the rules in order to make a fair comparison between the three CIT2 type-reduction approaches.

### C. The algorithm

In this Subsection, a formal description of the algorithm is provided (Algorithm 1). As already mentioned, the idea is to find the *switch indices* that produce the AESs with the highest and lowest T1 centroid values. The algorithm described here, works with a single output variable at a time. In other words, it must be executed once for each output generated by the system. For simplicity, the analysis carried out in this paper assumes that the CIT2 FLS only produces one output (i.e. it has only one output variable).

In the *for-loop* starting at line 11, different AESs are generated, testing all the possible switch index values. At the end of the procedure, the highest and lowest T1 centroid

---

**Algorithm 1** Switch Index Type-Reduction Algorithm
 

---

```

1: Sort the CIT2 sets partitioning the output variable (i.e. all the sets used as consequents in the rulebase) in ascending order
   based on the min value of their support set
2: Give each sorted set  $\check{C}$  an ordinal index, obtaining the list  $(\check{C}_1, \dots, \check{C}_n)$ 
3:  $S = \emptyset$   $\triangleright$  This set, will contain the centroid values of all the AES generated by the switch index approach
4: for each  $\check{C}_i \in (\check{C}_1, \dots, \check{C}_n)$  do
5:    $F_i^L = 0, F_i^U = 0$   $\triangleright$  They will store the maximum lower and upper firing strength for  $C_i$ 
6:   for each rule  $R$  in which  $\check{C}_i$  appears as a consequent do
7:     Compute  $R_F.lower$  and  $R_F.upper$ , respectively the lower and upper bounds of the firing interval of  $R$ 
8:      $F_i^L = \max(F_i^L, R_F.lower), F_i^U = \max(F_i^U, R_F.upper)$ 
9:   end for
10: end for
11: for  $index=1$  to  $n$  do
12:   for  $\check{C}_i \in (\check{C}_1, \dots, \check{C}_n)$  do
13:     if  $i < index$  then
14:        $FS-1 = F_i^L$ 
15:        $FS-2 = F_i^U$ 
16:     else
17:        $FS-1 = F_i^U$ 
18:        $FS-2 = F_i^L$ 
19:     end if
20:     Apply the implication operator on the rightmost AES of  $\check{C}_i$  using  $FS-1$ , obtaining  $\bar{C}_i$ 
21:     Apply the implication operator on the leftmost AES of  $\check{C}_i$  using  $FS-2$ , obtaining  $\underline{C}_i$ 
22:   end for
23:    $\bar{C} = \bigcup_{1 \leq i \leq n} \bar{C}_i$   $\triangleright$  Do the union of the sets  $\bar{C}_1, \dots, \bar{C}_n$ 
24:    $\underline{C} = \bigcup_{1 \leq i \leq n} \underline{C}_i$ 
25:    $S = S \cup \{\text{centroid}(\bar{C})\} \cup \{\text{centroid}(\underline{C})\}$ 
26: end for
27: return the minimum and maximum centroid values  $x_L, x_U \in S$ 

```

---

among all the AESs that have been generated, are used as the endpoint of the type-reduced set returned as an output. The identification of the switch indices uses a brute force approach. This method has been chosen for its simplicity and as a first strategy to compute the novel concept of switch indices introduced in this paper. In future work, the mathematical properties of the AESs and the switch indices themselves will be analyzed to establish a criterion or a mathematical formula that could directly determine the right switch indices, similarly to what happens in the KM algorithm with the switch points.

Conceptually, the algorithm can be summarized in the following steps:

- 1) Give each CIT2 consequent MF an ordinal index by sorting them in ascending order of the minimum value of their support set, obtaining the list  $(\check{C}_1, \dots, \check{C}_n)$ .
- 2) For each consequent  $\check{C}_i$ , compute its lower firing value  $F_i^L$  as the maximum lower firing strength of all the rules where it appears as a consequent; analogously, compute its upper firing value  $F_i^U$  as the maximum upper firing strength of all the rules where it appears as a consequent.
- 3) If computing the right endpoint of the type-reduced set (i.e. to generate the AES with the maximum centroid value), replace each consequent MF with its rightmost AES; if computing the left endpoint, take the leftmost AES instead.

- 4) Test all the possible switch index values, between 0 and the maximum index given to the consequent MFs:
  - i. If computing the left endpoint, apply the inference operator on each replaced consequent MF  $C_i$  using  $F_i^U$  if the MFs has an index smaller than the switch index, use  $F_i^L$  otherwise; for the right endpoint instead, use  $F_i^L$  before the switch index and the  $F_i^U$  after it.
  - ii. Do the union of the AES resulting from the inference and defuzzify the set obtained, computing its centroid.
- 5) Return, as the final type-reduced set, the lowest and highest centroid values obtained from the defuzzification at the previous step.

A representation of the intermediate results of these steps can be found in Sec. IV-B1 and in Fig. 1 in the Supplemental Material.

#### D. Mathematical description

The exhaustive approach evaluates every combination of every *embedded set* at every firing strength that arises from each individual rule in combination in the output. Empirically, it has been observed that the combination of sets that produced the AES with the lowest (left-most) centroid, follow the AES obtained by carrying out the implication with the upper value

in the firing interval on the leftmost AES of the consequent sets for some left-hand portion of the universe, before switching at some point to following the left AES with the lower value in the firing interval for the remainder of the universe (and vice versa for the highest centroid). This observed behaviour has inspired the current algorithm to determine this switch-point and use the acceptable embedded sets with these properties for the type-reduction. An example of this phenomenon is shown in Fig. 3, in which the fired FOU is obtained as described in Fig. 1 in the Supplemental Material. In this case, for the magenta section, the leftmost embedded set obtained with the higher firing value is used; the green section is where the switch happens and the left AES with the *lower* firing value is used instead.

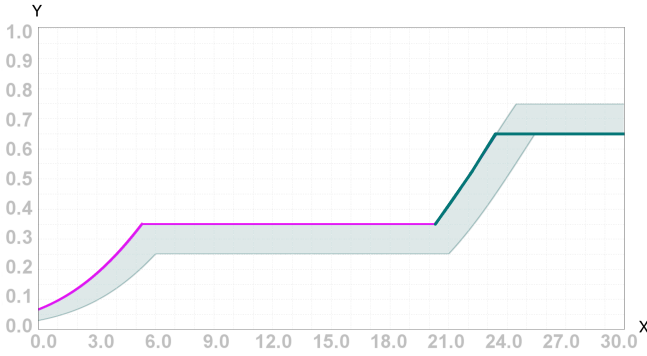


Fig. 3. The fired FOU of a CIT2 FLS (shaded) and the AES with the lowest centroid value. The magenta section of the AES is obtained following the leftmost AES after the implication with the upper firing value in the firing interval, while the section in green is obtained following the leftmost AES after the implication with the lower firing value.

Formally, the problem solved by the algorithm to compute the left endpoint of the constrained type-reduced set can be modelled mathematically as follows (the right endpoint can be expressed analogously):

$$\text{Left\_endpoint} = \min_{0 \leq SI \leq n} \left( \text{Centroid} \left( \bigcup_{1 \leq i \leq n} C_i^{L'} \right) \right) \quad (2)$$

$$\mu_{C_i^{L'}}(x) = \begin{cases} \min(\mu_{C_i^L}(x), F_i^U), & i < SI \\ \min(\mu_{C_i^L}(x), F_i^L), & i \geq SI \end{cases} \quad (3)$$

where  $SI$  is the switch index,  $C_i^L$  is the leftmost AES of the  $i$ -th consequent set,  $C_i^{L'}$  is the set obtained after the implication on  $C_i^L$ ,  $F_i^U$  and  $F_i^L$  are the maximum and minimum firing strength among all the rules in which  $\check{C}_i$  appears as a consequent (computed as in Algorithm 1 at line 8).

Determining whether Algorithm 1 computes the same type-reduced set as the exhaustive approach is not straightforward. In fact, in the exhaustive version, *all* the AES of each consequent  $\check{C}_i$  are considered and the possible firing strength  $F_i$  in the min operator in (3) could be *any* value in  $[F_i^U, F_i^L]$ .

Additionally, the union of the AES before the centroid computation in (2) may produce a non-convex and non-normal set (such as that in Fig. 31) while the overlapping of the MFs of each AES also plays a role in the final result and

makes the problem challenging to solve from a mathematical point of view. For these reasons and for space restrictions, we believe that the formal relationship between Algorithm 1 and the exhaustive method needs to be studied in a separate paper.

For now, the usefulness of the novel algorithm has been shown in the extensive tests reported in Sec. IV. Indeed, in all experiments undertaken so far, Algorithm 1 produces the same as the exhaustive method.

#### E. Analysis and computational complexity

The analysis carried out here does not include the computation needed to determine the firing strength of the rules (lines 4-9). In all the case studies examined in Sec. IV, the firing intervals of the rules are computed in the same way they are computed in IT2 inference, using Theorem 1.

Before the algorithm can build the AESs, it is necessary to sort the  $n$  consequent sets used in the CIT2 FLS in ascending order of the minimum value in their support set, which requires  $\mathcal{O}(n \log n)$  operations. Once the consequents are sorted, for each of the  $n$  iteration of the *for-loop* at line 11, two AES are generated, with each generation requiring  $\mathcal{O}(n)$  operations (because of the union at lines 23, 24). The defuzzification at line 25 requires  $\mathcal{O}(kn)$  operations with  $k$  being the discretization level used and assuming that for each discretized point  $x$  its membership degree with respect to  $\check{C}$  is computed as:

$$\mu_{\check{C}}(x) = \max_{1 \leq i \leq n} \mu_{\check{C}_i}(x) \quad (4)$$

and the membership degree of  $x$  with respect to  $\check{C}$  is calculated in the same way. Therefore, the final computational complexity of the algorithm is  $\mathcal{O}(2kn^2)$ , where  $n$  is the number of MFs that partition the output variable. This represents a significant improvement when compared to the original exhaustive algorithm that had a computational complexity of  $\mathcal{O}(k^{n+1})^m$  where  $m$  is the number of rules,  $n$  the number of antecedents per rule and  $k$  the number discrete number of AES that had to be selected for each of the CIT2 FSs in the CIT2 FLS [14]. A comparison of the run times of the novel procedure, the sampling method, and the exhaustive algorithm is presented in Section IV.

## IV. PRACTICAL APPLICATIONS

This section is focused on the application of Algorithm 1 in three case studies for the comparison of this novel approach with other type-reduction methods. The first subsection shows a run time comparison between KM, EKM, CIT2 sampling, CIT2 exhaustive and Algorithm 1 in the type-reduction of a large number of CIT2 FSs. The second part of the section, instead, compares the different constrained approaches in terms of endpoints of the produced type-reduced set to analyze their differences. Lastly, the third subsection presents a qualitative comparison between Algorithm 1, the sampling method and EKM in a real-world case study. Specifically, the problem of the recommendation of post-operative breast cancer treatment is analyzed. The accuracy values of the different approaches are compared, together with the interpretability of the classifications that they produce.

### A. Run time comparison<sup>1</sup>

The experiments reported here, consist in the type-reduction of a number of FLSs produced as the output of a CIT2 FLS. Since the computational complexity of Algorithm 1 is  $\mathcal{O}(kn^2)$  with  $n$  being the number of MFs that partition a given output variable and  $k$  being the discretization level used to defuzzify the AES, the experiments involve output variables partitioned with a different number of MFs. By doing this, it is possible to see how the algorithm performs as the cardinality of the partitioning increases.

The experimental setup is the following: 4 FLS have been produced with the output variable partitioned respectively with 2, 3, 5 and 7 MFs. Each of these MFs is used as the consequent of a different fuzzy rule with a single antecedent MF and one input variable. Therefore, a FLS with a partitioning size of  $n$  has  $n$  rules. The generator sets used in this experiments are triangular MFs with parameters  $(x - 1, x, x + 1)$ ,  $x \in \mathbb{N}$ ,  $1 \leq x \leq 7$ . The displacement set used to generate the FOU is the interval  $[-0.5, 0.5]$  and the resulting sets can be seen in Fig. 2 in the Supplemental Material. The minimum operator has been used to carry out the implication. Each system has been run  $5 \times 10^6$  times and its outputs type-reduced using different algorithms. The input values have been set randomly, whilst maintaining that each rule always fires with a minimum firing strength of 0.1.

The methods tested are KM [9], EKM [16], the sampling CIT2 method [14] with 50 samples with uniform random distribution (CIT2-S50) [14], the exhaustive method (CIT2-Exh.) [14] and the novel procedure introduced in this paper (Algorithm 1) (CIT2-SI). Additionally, the generator sets of the CIT2 FLSs have been used to create a T1 version of the FLSs described above to compare the run times of these T2 FLSs with their T1 counterparts. For the exhaustive CIT2 approach, 5 AES have been considered for each CIT2 FS (the generator set plus 2 AES at its left and 2 at its right, uniformly distributed). The experiments have been run in Java on a Windows machine with an i7-7600U CPU. For the KM, EKM and T1 FLSs implementations, the Juzzy library [17] has been used. To defuzzify the T1 AES, T1 ES and the output of the T1 FLS, they are uniformly discretized in 1000 points and their centroid is computed.

TABLE II  
RUNNING TIMES (IN SECONDS) OF THE DIFFERENT APPROACHES.

| MFs | T1     | KM      | EKM            | CIT2-Exh.      | CIT2-S50 | CIT2-SI        |
|-----|--------|---------|----------------|----------------|----------|----------------|
| 2   | 12.29s | 196.57s | 72.35s         | 635s           | 1077.21s | <b>19.09s</b>  |
| 3   | 26.00s | 285.60s | 107.58s        | 2655s          | 1424.17s | <b>62.94s</b>  |
| 5   | 41.15s | 479.54s | 198.56s        | 35600s (est.)  | 2323.88s | <b>191.76s</b> |
| 7   | 53.77s | 780.25s | <b>247.32s</b> | 714170s (est.) | 2979.54s | 398.88s        |

The run times of the different approaches are reported in Table II. The minimum value in each row **among the T2 approaches** is highlighted in bold. As it is possible to see, Algorithm 1 (CIT2-SI in the table) is at least 7.5 times faster in all the cases when compared to the sampling type-reduction technique. In addition to that, CIT2-SI performs overall better

than all the other approaches, being slower than EKM (but still faster than KM) only when the output is partitioned with more than 5 MFs. For the exhaustive approach in the last two FLS (CIT2-Exh. with 5 and 7 MFs), only 1000 type-reductions have been performed and then their run time multiplied by 5000 to obtain an estimate of the total time it would be required to perform  $5 \times 10^6$  type-reductions using that algorithm due to its impractical computational time.

Although it has been shown that run times are heavily affected by the specific programming language used to implement the type-reduction algorithm [18], the significant difference of at least one order of magnitude between the presented approach (CIT2-SI) and the other CIT2 algorithms (CIT2-Exh. and CIT2-S50) can hardly depend on implementation details. The relationship between CIT2-SI, KM and EKM, however, may be different in other programming languages, since the specific timings of each depend on both the algorithm and the programming language used.

### B. Comparison between the constrained approaches<sup>1</sup>

To compare the type-reduction set produced by the three different approaches (exhaustive, sampling and switch index) a FLS for a simplified version of the iris problem [19] is analyzed. In the original version, 4 input variables are used (sepal and petal length and width) to identify the type of iris plant. In this version, only 2 of them are used: petal length and width. This choice has been made because the computational time for the exhaustive approach grows very quickly with the number of antecedents and rules of the FLS. Therefore, in order to be able to use it for this comparison, a compact rule-base and a small number of input variables are necessary. Each variable is partitioned with 3 labels (*low*, *medium* and *high*) used to create the following 5 rules:

- 1) If petal length is low and petal width is low then species is *setosa*.
- 2) If petal length is medium and petal width is medium then species is *versicolor*.
- 3) If petal length is high and petal width is high then species is *virginica*.
- 4) If petal length is medium and petal width is high then species is *virginica*.
- 5) If petal length is high and petal width is medium then species is *virginica*.

To run the exhaustive algorithm, each CIT2 FS involved in the system has been discretized in 5 AES: the generator set plus 2 AES at its left and 2 at its right, evenly distributed. Additional details on the MFs used in this experiment can be found in the Supplemental Material, Table II. For the sampling method, the results have been obtained as the average of 50 executions of the sampling method computed with 50 samples each time. The standard deviation for this approach is also reported. The T1 AES selected by the different approaches are discretized in 1000 points to be defuzzified. In Table III, the interval representing the type-reduced set for the 3 approaches is reported for 3

<sup>1</sup>The code used in this experiment will be available on Code Ocean



different input values, one for each of the possible species.

TABLE III  
COMPARISON OF THE DIFFERENT CONSTRAINED TYPE-REDUCTION  
METHODS

| Inputs     | Exhaustive   | Sampling               | Switch Index |
|------------|--------------|------------------------|--------------|
| (0.2, 1.4) | [0.83, 1.26] | [0.85±0.01, 1.22±0.01] | [0.83, 1.26] |
| (1.4, 4.7) | [2.08, 2.69] | [2.19±0.04, 2.57±0.03] | [2.08, 2.69] |
| (2.1, 6.6) | [2.77, 3.18] | [2.80±0.01, 3.16±0.01] | [2.77, 3.18] |

In all the cases both the switch index and the exhaustive approach produce the same result while the sampling gives a slightly different value. Table IV, shows the average absolute difference (for both the endpoints of the type-reduced set) between the sampling and switch index procedures with respect to the exhaustive method over the 150 entries of the iris dataset. In other words, each entry is a pair  $[x, y]$  representing the average absolute difference between two approaches for the left ( $x$ ) and right ( $y$ ) endpoint of the interval representing the type-reduced set.

Also the standard deviation is reported. As can be seen, in the FLS analyzed here there is no difference between the switch index approach and the exhaustive one. At the moment, we are not able to prove whether they always produce the same results or this only happens in a subset of situations, perhaps caused by the specific MFs, discretization or partitioning used. The relation between Algorithm 1 and the exhaustive approach will be further studied in future work with a formal analysis and additional case studies.

TABLE IV  
AVERAGE ABSOLUTE DIFFERENCE BETWEEN THE APPROACHES

|            | Sampling               | Switch Index |
|------------|------------------------|--------------|
| Exhaustive | [0.06±0.04, 0.06±0.04] | [0.0, 0.0]   |

1) *Step-by-step application of the algorithm:* The iris CIT2 FLS presented above, will be used to illustrate each step of Algorithm 1, in order to clarify how the procedure works. In this example the input value for the petal length is 1 while its width is 3. The three MFs modeling respectively the setosa, versicolor and virginica species are represented (shaded) in Fig. 4. Algorithm 1 sorts them using the leftmost value of their support set in order to give each one of them an ordinal index (line 2). In this case, the index of setosa (in blue) is 0, since it is the leftmost CIT2 FS partitioning the output variable, while the indices of versicolor and virginica (in red and green) are respectively 1 and 2.

Then, the firing interval for each rule is computed (for-loop at line 4). The firing strengths of the rules in the system are the following:

- 1: IF Length IS Low AND Width IS Low THEN Species IS  
Setosa: [0.18, 0.32]
- 2: IF Length IS Medium AND Width IS Medium THEN Species IS  
Versicolor: [0.51, 0.82]
- 3: IF Length IS High AND Width IS High THEN Species IS  
Virginica: [0.03, 0.06]
- 4: IF Length IS Medium AND Width IS High THEN Species IS  
Virginica: [0.07, 0.14]

- 5: IF Length IS High AND Width IS Medium THEN Species IS  
Virginica: [0.03, 0.06]

For each of the three classes, the firing interval is computed as the maximum lower and maximum upper values of the firing strength of the rules in which they appear as consequent. In this case, the firing interval of each class are:

- Setosa: [0.18, 0.32]  
Versicolor: [0.51, 0.82]  
Virginica: [0.07, 0.14]

At line 20 of Algorithm 1, the implication operator (minimum) is then applied to the rightmost AESs of the three classes. The rightmost AES for each of the classes is represented with a solid line in Fig. 4. Line 21 carries out the implication on the leftmost AES. Since the two operations are very similar, only line 20 will be analyzed.

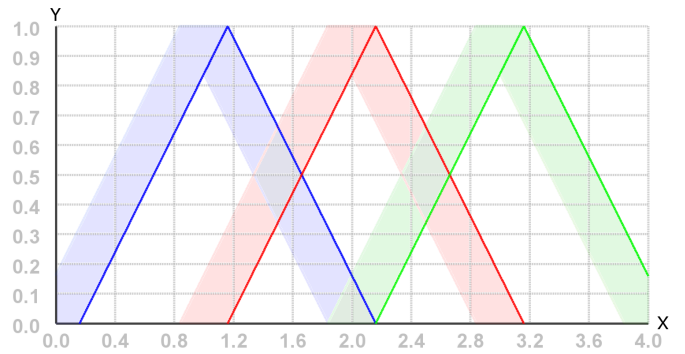


Fig. 4. CIT2 fuzzy sets modeling the three iris classes (shaded) and their rightmost AES

Before doing the implication, the procedure selects a current switch-index value to try for the current iteration of the for-loop at line 11. The result of line 20 for all the possible switch-index values tested by the for-loop is shown in picture Fig. 5.

If the switch index value is smaller than the index of the class, then the lower firing value is used for the implication, otherwise the upper firing value is used. For each of the switch index values, a single set is produced by doing the union of the three classes after the implication. The set obtained at this stage is an AES of the fuzzy output of the FLS. These three AESs obtained from the union are then defuzzified and their centroid values stored in a list  $S$  (line 25). After also line 21 is computed and the centroid values produced by it are added to  $S$ , the interval  $[\min(S), \max(S)]$  is returned as the value of the type-reduced set.

### C. Real-world application

In this subsection, the novel algorithm is qualitatively compared to the EKM procedure and sampling method on a real-world classification task.

The problem analyzed in this paper is the recommendation of post-operative therapy for breast cancer. In this case both the interpretability and the explainability of the system play a crucial role. An interpretable system is made of MFs with a clear semantic meaning (i.e. a linguistic label) and a rule-base composed of a limited number of rules [3]. This allows a non-expert audience, i.e. the physicians in this case, to get an

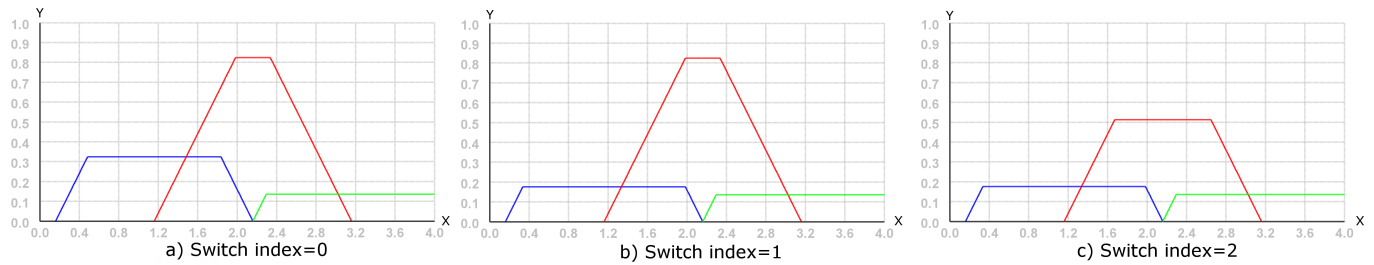


Fig. 5. Result of the implication with different switch-index values

intuitive understanding of the rules followed by the system to produce the final classification. Explainability, instead, is defined as the ability to “explain the user the process it followed to make the output decision” [3]. In other words, the system must provide an explanation for each of the classifications produced. Therefore, in FLS for XAI it is important to use defuzzification algorithms with a type-reduction process that can produce explanations for the outputs of the FLS.

The goal of the system proposed here, is to determine whether a chemotherapy treatment may or may not be beneficial as a post-operative treatment. This decision problem was first described by Garibaldi et al. [20].

To provide a final recommendation to the patient, a multi-disciplinary group of physicians decide on the most effective therapy to recommend. In this case, the goal of the system is to replicate the decision of the group of doctors with respect to the recommendation of chemotherapy only.

To make the fuzzy system interpretable, it has been built starting from the clinical protocol used by the Nottingham University Hospitals NHS Trust (Fig. 3 in the Supplemental Material), generating the rule-base shown in the Supplemental Material in Fig. 4.

The system has the following five inputs:

- NPI: Nottingham Prognostic Index, an index that indicates the prognosis after the surgery. It is calculated from three criteria: size of the lesion, number of involved lymph nodes and tumor grade. For this variable, 4 linguistic label (and therefore, 4 FLSs) were identifies from the recommendation protocol: *low*, *medium-low*, *medium-high* and *high*. The cut-off points between the labels are respectively 3.0, 3.4 and 4.4. The universe of discourse (UOD) is the interval  $[0,10]$ .
- ER: Estrogen Receptor test result, it shows whether estrogen fuels the tumor. This can be used to decide if hormone-suppression treatment would be beneficial. The linguistic labels in this case are *negative*, *weak* and *positive*, with the cut-off points being 20 and 100. The UOD is the interval  $[0, 300]$ .
- Age: the age of the patient. The labels are *young*, *middle age* and *old*, with their respective cut-off points being 40 and 60 while the UOD is  $[0, 90]$ .
- VI: Vascular Invasion, represents the presence of unequivocal tumor in vascular spaces. It has three labels, *yes*, *maybe*, *no* with the cut-off points being 1.5 and 2.5. The UOD is  $[1, 3]$ .
- LN: positive Lymph Node ratio, it's the ratio of lymph

nodes that are positive to cancer change on the total sample of tested lymph nodes. The labels in this case are *negative* and *positive* with the cut-off point being 0.03.

The description of these input variables is based on material previously presented in the original paper [20]. The output variable, instead, is the *chemotherapy recommendation* that is partitioned in three labels, *yes*, *no* and *maybe*. The *yes* and *no* cases, represent respectively a recommendation in favour and against the chemotherapy. The *maybe* case, instead, represents a situation in which an agreement among the physician could not be reached and therefore a clear recommendation can not be provided; as a consequence of that, the administration of the chemotherapy is further discussed with the patient.

To build interpretable MFs that keep their semantic meaning and cut-off points but also obtain a FLS with good performances, the following optimization process has been implemented. The T1 MFs used for the input variables of the VI-F FLS in [20] are used as a starting point by a genetic algorithm. To carry out the optimization in a way that keeps the cut-off points intact, the intersection points of the MFs remain unchanged and only the slopes of the intersecting segments of the MFs are tuned. For example, consider the T1 MFs for the *age* variable, as shown in Fig. 5 [20], in the Supplemental Material.

The goal of the genetic algorithm is to find the optimal slopes for the intersecting oblique lines of the *young*, *middle age* and *old* MFs. By doing that, their intersection points and therefore the cut-off points between them remain unchanged. The same optimization process is used for all the MFs partitioning the input variables to ensure a high level of interpretability of the systems and the adherence to the protocol described in Fig. 3. The parameters of the genetic optimization are reported in Table I in the Supplemental Material.

For the output variable instead, there are no indications in the protocol that can help build the three MFs (*yes*, *maybe*, *no*). For this reason, they have been designed as follows: the *maybe* MF is modeled as a isosceles triangles centered in 50 (the midpoint of the UOD) while its width is determined by the genetic algorithm. The *yes* and *no* MFs, instead, are shoulder MFs respectively ending and starting in the midpoint of the UOD. The cut-off points are the ones with a membership value of 0.5 in the *maybe* MF. An example of the partitioning generated by the genetic algorithm for the output variable *chemotherapy recommendation*, is shown in Fig. 6 in the Supplemental Material.



The process described so far, generates the T1 MFs that can be used as GSs of the CIT2 MFs. To obtain CIT2 MFs, however, also the displacement set (DS), i.e. the shifting values to generate the FOU, needs to be determined. The choice of the width of the DS for each CIT2 MF is made by the genetic algorithm. The FLS returned at the end of the optimization is the one with the highest accuracy value on the training set.

The real-world dataset used for the optimization of the system is the same one presented in the original paper [20]. However, due to its imbalanced nature, only some of its entries have been selected. Specifically, all the 191 *yes*, all the 52 *maybe* and 191 *no* cases have been chosen, for a total of 434 instances.

The optimization has been run four times to generate a T1 FLS, an IT2 FLS and 2 CIT2 FLS using respectively the sampling method and Algorithm 1 for the type-reduction step. The process to obtain the T1 FLS is the same one used to determine the GS of the IT2 and CIT2 FLS. The genetic optimization to obtain the FOU of IT2 and CIT2 FLS is the same. To run the systems, the Mamdani inference is used, with the *min* function implementing the AND and implication operators while the EKM type-reduction procedure is used for the IT2 FLSs. The final output of the system is calculated as the mid-point (centroid) of the type-reduced set. This value is then converted into a class using the cut-off points between the chemo MFs *no*, *maybe* and *yes*. Although the endpoints of the type-reduced set are not *directly* used at this step of the classification in this example, they are very useful in the development of explainable systems. In fact, producing an interval as an output rather than a crisp value and being able to explain how the interval has been generated would provide additional information to the end user regarding the effects of the uncertainty on the final classification (i.e. the width of the interval), thereby clearly showing the decision process followed by the FLS.

The accuracy values of each of these systems have been computed as the average of a 5-fold cross validation approach repeated 5 times for a total of 25 executions per system. The results are reported in Table V. All the FLSs have been designed in Java; the T1 and IT2 FLSs have been implemented with Juzzy [17] while Juzzy Constrained [21] has been used for the CIT2 FLSs.

TABLE V  
RESULTS OF THE DIFFERENT GENETIC FLS

| FLS                         | Accuracy |
|-----------------------------|----------|
| T1                          | 70.762%  |
| IT2 (EKM)                   | 71.826%  |
| CIT2 (Switch Index)         | 72.568%  |
| CIT2 (Sampling, 50 samples) | 72.845%  |

The data shows that the IT2 and the 2 CIT2 FLSs perform better than the T1 one; both the CIT2 also show a higher accuracy than the IT2 FLS, with the CIT2 FLS with the sampling method having the best performance (0.277% better than the switch index algorithm). Being this comparison only based on a single case study with a specific tuning algorithm, it is not sufficient to make any claims on which modeling

approach, i.e. IT2 or CIT2, performs better and under which circumstances. The main goal of this case study is to provide a worked example of the novel algorithm proposed in this paper, and show its potential in terms of its use in XAI applications, as discussed in the next subsection. However, a more formal comparison, using multiple datasets and a statistical analysis will be carried out in future work to get a better understanding of which approach is better in which situations.

1) *Interpretability*: With an IT2 fuzzy system, regardless of the type-reduction method used, it is possible to provide an explanation for the outputs of the system by analysing the rules that fired with a given set of inputs. Following a novel approach proposed by Mendel [22], any input can be linked to its IT2 first-order rule partition from which it is possible to determine the firing rules. These can then be shown to the end-user as an explanation for the output produced.

As a further enhancement to this capability, CIT2 fuzzy systems have the ability to also explain the type-reduced set. When a designer wants to explicitly model the effects of the uncertainty on the decision process, the interval obtained from type-reduction can be provided as the system output. An application of this concept is shown in Sec. IV-B1, where the firing of each class is reported as an interval; the same strategy can also be applied to the chemotherapy recommendation scenario, in which the system output is represented by an interval, e.g. [75, 90], showing how much the FLS is in favour of the chemotherapy treatment and how certain or uncertain its decision is. With a CIT2 FLS, the specific rules and inputs that determine *each of the endpoints of the interval*, i.e. 75 and 90 in this example, can be identified by analyzing the AESs that lead to those values during the type-reduction, as illustrated by the following analysis.

Fig. 6 shows one of the ES selected by the EKM procedure and the AES chosen by Algorithm 1 to type-reduce an output of the system. In other words, these are the ESs chosen by the procedures to obtain the right endpoints of the type-reduced set. In the CIT2 case, by looking at the way those AESs are generated, it is possible to see the contribution of each of the consequent MFs to the final result as well as the firing strengths obtained from the input values.

The AES in Fig. 6.b has been obtained as the union of the sets shown in Fig. 7. The latter sets, represent all the  $\bar{C}_i$  at line 23 of Algorithm 1, before the union. Through this analysis, it is possible to see that the *no* MF (the one in blue) was fired with a strength of 0.45, the *maybe* one (in the middle) with a value of 1 and *yes* with a value of 0.39. Additionally, it is possible to identify which rules generated the firing strengths (line 8 of Algorithm 1), making possible the generation of a textual explanation for each of the endpoints of the type-reduced set, similar to what can already be done for the outputs of T1 FLS (e.g. [5], [6]).

Linking each ESs identified by the KM procedure to rules or inputs of the systems, on the other hand, can be challenging. In fact, for the resolution of the well-defined mathematical problem carried out by the KM procedure, it makes no difference if the IT2 fuzzy set to type-reduce has been obtained as the output of an IT2 FLS or not. The procedure is, in fact, unaware of the existence of the rulebase.

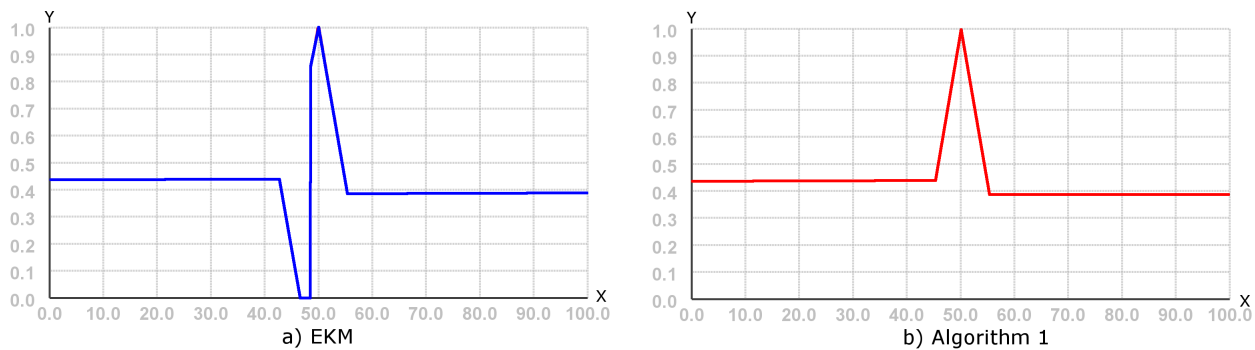


Fig. 6. ESs that determine the right value of the EKM (a) and CIT2 (b) centroid.

More recently, the ability to use the algorithm proposed in this paper in order to produce explanations has been further explored, showing how it can be used to generate natural-language explanations in classification problems [23].

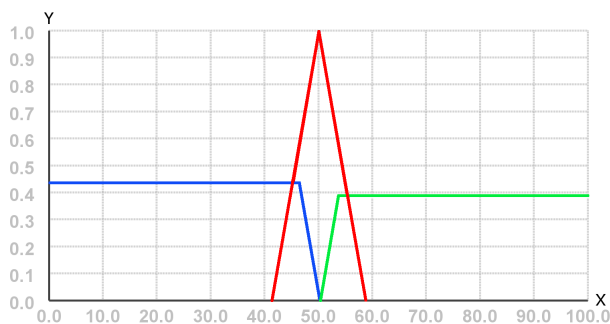


Fig. 7. The unions of these sets generates the AES shown in Fig. 6.b

## V. CONCLUSIONS AND FUTURE WORK

CT2 FSs have been proposed as a way to increase the interpretability and explainability of T2 FSs [13], being a specific way of generating T2 FSs when starting from a T1 MF modeling the same concept. Particularly, CIT2 FS have been previously described and analysed, showing how they can be used to produce CIT2 FLS with a high level of explainability [14], [15]. However, the two original type-reduction procedures originally presented, had the drawback of being significantly slower than the widely used KM [9] procedure.

In this paper, a novel inference and type-reduction algorithm for CIT2 FSs has been presented, based on the idea of *switch indices* rather than the *switch points* used in the KM procedure.

The running times of the novel algorithm presented in this paper have been compared to different T2 type-reduction procedures (KM, EKM, CIT2-S50), showing better performances in three of the four tests carried out.

Finally, a real-world classification application has been used as a case study to have a qualitative comparison in terms of accuracy and interpretability between the algorithm produced in this paper and the widely adopted EKM procedure. It has been shown that the CIT2 FLS with the novel algorithm keeps the same level of accuracy as its IT2 counterpart while

producing outputs with a higher level of interpretability (for each of the AES it is possible to determine which rules and input values generated them).

In future work, we plan on further decreasing the run time of Algorithm 1. In fact, the identification of the switch indices, for now, has been carried out using a brute force approach. Determining a different stopping criterion or a direct way to identify the switch indices (similarly to what happens with the switch points in the KM procedure) would further improve the computational complexity of the novel procedure presented here. Finally, the possible advantages and differences in the use of the constrained modeling approaches in systems like Takagi-Sugeno [24] will be studied.

## REFERENCES

- [1] D. Gunning, "Explainable artificial intelligence (xai)," *Defense Advanced Research Projects Agency (DARPA)*, nd Web, 2017.
- [2] D. Castelvechi, "Can we open the black box of ai?" *Nature News*, vol. 538, no. 7623, p. 20, 2016.
- [3] A. Fernandez, F. Herrera, O. Cordon, M. Jose 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, Feb 2019.
- [4] L. A. Zadeh, "Fuzzy logic= computing with words," in *Computing with Words in Information/Intelligent Systems 1*. Springer, 1999, pp. 3–23.
- [5] N. Potie, S. Giannoukagos, M. Hackenberg, and A. Fernandez, "On the need of interpretability for biomedical applications: Using fuzzy models for lung cancer prediction with liquid biopsy," in *International Conference on Fuzzy Systems (FUZZ-IEEE 2019)*, 2019.
- [6] I. Baaj and J.-P. Poli, "Natural language generation of explanations of fuzzy inference decisions," in *International Conference on Fuzzy Systems (FUZZ-IEEE 2019)*, 2019.
- [7] 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.
- [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] 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.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] 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.

- [14] P. D'Alterio, J. M. Garibaldi, R. John, and A. Pourabdollah, "Constrained interval type-2 fuzzy sets," *IEEE Transactions on Fuzzy Systems*, 2020.
- [15] 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.
- [16] D. Wu and J. M. Mendel, "Enhanced karnik–mendel algorithms," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 4, pp. 923–934, 2009.
- [17] 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.
- [18] C. Chen, D. Wu, J. M. Garibaldi, R. I. John, J. Twycross, and J. M. Mendel, "A comprehensive study of the efficiency of type-reduction algorithms," *IEEE Transactions on Fuzzy Systems*, pp. 1–1, 2020.
- [19] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.
- [20] J. M. Garibaldi, S.-M. Zhou, X.-Y. Wang, R. I. John, and I. O. Ellis, "Incorporation of expert variability into breast cancer treatment recommendation in designing clinical protocol guided fuzzy rule system models," *Journal of Biomedical Informatics*, vol. 45, no. 3, pp. 447 – 459, 2012.
- [21] P. D'Alterio, J. M. Garibaldi, R. I. John, and C. Wagner, "Juzzy constrained: Software for constrained interval type-2 fuzzy sets and systems in Java," in *2020 IEEE World Congress on Computational Intelligence (WCCI 2020)*, July 2020.
- [22] J. M. Mendel, "Explaining the performance potential of rule-based fuzzy systems as a greater sculpting of the state space," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 4, pp. 2362–2373, 2018.
- [23] P. D'Alterio, J. M. Garibaldi, and R. I. John, "Constrained interval type-2 fuzzy classification systems for explainable AI (XAI)," in *2020 IEEE World Congress on Computational Intelligence (WCCI 2020)*, July 2020.
- [24] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, no. 1, pp. 116–132, 1985.