Information Sciences 244 (2013) 1–25

Contents lists available at SciVerse ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Defuzzification of the discretised generalised type-2 fuzzy set: Experimental evaluation

Sarah Greenfield [a,*], Francisco Chiclana [a,b]

[a] Centre for Computational Intelligence, Department of Informatics, De Montfort University, Leicester LE1 9BH, UK

[b] DMU Interdisciplinary Group in Intelligent Transport Systems, Department of Informatics, De Montfort University, Leicester LE1 9BH, UK

ARTICLE INFO

Article history:
Received 10 August 2012
Received in revised form 29 January 2013
Accepted 29 April 2013
Available online 14 May 2013

Keywords:
Type-2 fuzzy set
Defuzzification
Exhaustive method
Sampling method
α-Planes method
VSCTR

ABSTRACT

The work reported in this paper addresses the challenge of the efficient and accurate defuzzification of discretised generalised type-2 fuzzy sets as created by the inference stage of a Mamdani Fuzzy Inferencing System. The exhaustive method of defuzzification for type-2 fuzzy sets is extremely slow, owing to its enormous computational complexity. Several approximate methods have been devised in response to this defuzzification bottleneck. In this paper we begin by surveying the main alternative strategies for defuzzifying a *generalised* type-2 fuzzy set: (1) Vertical Slice Centroid Type-Reduction; (2) the sampling method; (3) the elite sampling method; and (4) the α-planes method. We then evaluate the different methods experimentally for accuracy and efficiency. For accuracy the exhaustive method is used as the standard. The test results are analysed statistically by means of the Wilcoxon Nonparametric Test and the elite sampling method shown to be the most accurate. In regards to efficiency, Vertical Slice Centroid Type-Reduction is demonstrated to be the fastest technique.

1. Introduction

In this paper responses to the challenge of the efficient and accurate defuzzification of discretised generalised type-2 fuzzy sets are evaluated. Defuzzification is the crucial final stage of the five-stage Fuzzy Inferencing System (FIS) as illustrated in Fig. 1. Type-2 defuzzification consists of two parts—*type-reduction* and defuzzification proper. Type-reduction is the procedure by which a type-2 fuzzy set is converted to a type-1 fuzzy set. This set is then defuzzified to give a crisp number. Owing to its enormous computational complexity, the additional stage of type-reduction of a *type-2* FIS has come to be regarded as a bottleneck [24]. The progress of generalised type-2 applications has been impeded as developers have opted [3, pp. 7, 8, 16] for the computationally simpler interval type-2 FISs [39,38] for which an increasing number of applications are being developed in areas such as control, simulation and optimisation [20,22,42,1,52,27,31,41,6,36,29,32,2,5,21,4]. In contrast, there are relatively few, though varied, generalised type-2 fuzzy applications [24,39,45,33].

The main strength of type-2 fuzzy logic is its ability to deal with the second-order uncertainties that arise from several sources [26], among them the fact that the meanings of words are often vague [39, p. 117]. The Karnik–Mendel Iterative Procedure (KMIP) [25,48] is the established technique for defuzzification of interval sets. The capability of the generalised type-2 paradigm to handle uncertainty is explored in [14]. Regrettably interval type-2 fuzzy sets are not able to model uncertainty as fully as their generalised counterparts, as they lack the crucial variability of the third dimension [39]. Our research, therefore, sees developing *generalised* type-2 systems as a challenge for the research community.

* Corresponding author. Tel.: +44 112551551x6709.
E-mail addresses: s.greenfield@dmu.ac.uk (S. Greenfield), chiclana@dmu.ac.uk (F. Chiclana).

0020-0255/$ - see front matter
http://dx.doi.org/10.1016/j.ins.2013.04.032

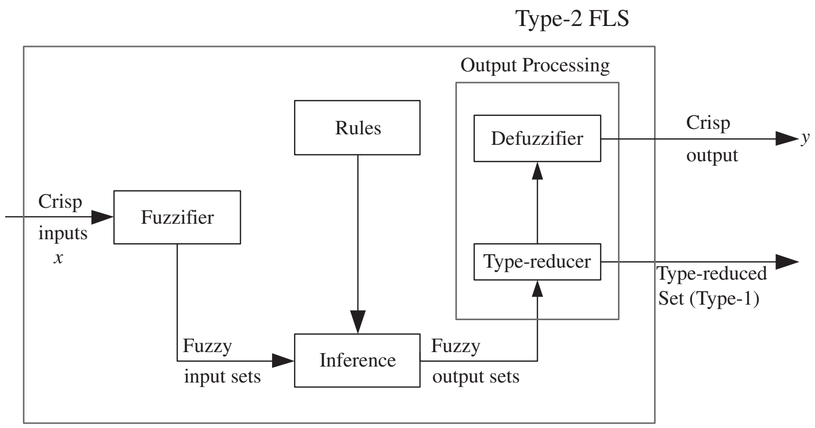


Fig. 1. Type-2 FIS (from Mendel [37]).

A triangular type-2 system with a defuzzification algorithm based on the KMIP has been developed by Starczewski [46]; this goes some way towards achieving our goal. Coupland and John [8,9] have exploited geometry to improve the speed of inferencing in generalised type-2 fuzzy sets. In 2008 Liu [34,40] proposed the α-planes method which involves decomposing a generalised type-2 set into a set of α-planes, which are horizontal slices akin to interval type-2 sets. This method is used in conjunction with an interval method such as the KMIP, the Greenfield-Chiclana Collapsing Defuzzifier [15], or the Nie-Tan Method [44]. The α-planes/KMIP method has been modified by Zhai and Mendel [53] to increase its efficiency. Experiments have shown that the α-planes decomposition introduces slight inaccuracies [18]; this is touched on in Section 7. Further inaccuracies are introduced by the associated interval method, as all the alternatives (apart from the interval exhaustive method) are approximations [15]. Independently, Wagner and Hagras have introduced the notion of zSlices [47], a concept similar to α-planes.

Table 1
Chronology of publication of defuzzification methods. The methods shown in bold are included in the evaluation reported below.

Date	Authors	Method	Reference	Publisher/publication
2001	Jerry M. Mendel	**Exhaustive**	[37]	Prentice-Hall PTR
February 2001	Nilesh N. Karnik Jerry M. Mendel	KMIP	[25]	Information Sciences
October 2002	Hongwei Wu Jerry M. Mendel	Wu-Mendel Approximation	[50]	IEEE Transactions on Fuzzy Systems
July 2007	Luı́s Alberto Lucas Tania M. Centeno Myriam R. Delgado	**VSCTR**	[35]	Proc. FUZZ-IEEE 2007
June 2008	Maowen Nie Woei Wan Tan	Nie-Tan	[44]	Proc. FUZZ-IEEE 2008
June 2008	Sarah Greenfield Robert I. John	Stratified TRS	[13]	Proc. IPMU 2008
May 2008	Feilong Liu	**α-Planes Representation**	[34]	Information Sciences
June 2009	Sarah Greenfield Francisco Chiclana Simon Coupland Robert I. John	Collapsing	[15]	Information Sciences
July 2009	Sarah Greenfield Francisco Chiclana Robert I. John	CORL	[17]	Proc. IFSA-EUSFLAT 2009
June 2011	Dongrui Wu Maowen Nie	EIASC	[49]	Proc. FUZZ-IEEE 2011
July 2011	Francisco Chiclana Shang-Ming Zhou	Type-1 OWA	[7]	Proc. EUSFLAT-LFA 2011
April 2012	Sarah Greenfield Francisco Chiclana Robert I. John Simon Coupland	**Sampling**	[19]	Information Sciences

Table 1 shows the development of the field of type-2 defuzzification over the past decade, as reflected in the major publications. A number of researchers have been working simultaneously and independently in this field, and the solutions developed are diverse and original. The application developer now has a choice of several methods; the stage has been reached where an experimental evaluation of the methods is desirable so as to establish the best performing method in the generalised case. Such an evaluation is the motivation behind this paper.

In this paper we shall be focussing on the predominant *discretised* type-2 FIS as created by the inference stage of a Mamdani FIS.[1] The exhaustive method of defuzzification for type-2 fuzzy sets is extremely slow, owing to its enormous computational complexity. Several approximate methods have been devised in response to this defuzzification bottleneck. We begin by surveying the main alternative strategies for defuzzifying a *generalised* type-2 fuzzy set: (1) Vertical Slice Centroid Type-Reduction; (2) the sampling method; (3) the elite sampling method; and (4) the α-planes method. It is timely that these techniques are evaluated experimentally; in this paper we report on how we have done this in relation to accuracy and efficiency. For accuracy the exhaustive method is used as the standard. The test results for accuracy are analysed statistically by means of the Wilcoxon Nonparametric Test. In regards to efficiency, timings are compared to establish the fastest technique.

For the research reported in this paper it is assumed (1) the type-2 fuzzy set is contained within a unit cube, (2) the type-2 fuzzy set may be viewed as a surface represented by (x,u,z) co-ordinates,[2] and (3) for type-1 sets the centroid method of defuzzification [28, p. 336] is employed.

1.1. Mathematical definition of the type-2 fuzzy set

Let X be a universe of discourse. A fuzzy set A in X is characterised by a membership function μ_A:$X \to [0,1]$. A fuzzy set A in X can be expressed as follows:

$$A = \{(x, \mu_A(x)); \mu_A(x) \in [0,1] \forall x \in X\}. \tag{1}$$

Note that the membership grades of A are crisp numbers.

Let $\widetilde{P}(X)$ be the set of fuzzy sets in X. A type-2 fuzzy set $\widetilde{A}$ in X is a fuzzy set whose membership grades are themselves fuzzy. This implies that $\mu_{\widetilde{A}}(x)$ is a fuzzy set in $[0,1]$ for all x, i.e.

$$\widetilde{A} = \{(x, \mu_{\widetilde{A}}(x)); \mu_{\widetilde{A}}(x) \in \widetilde{P}([0,1]) \forall x \in X\}. \tag{2}$$

This implies that $\forall x \in X \exists J_x \subseteq [0,1]$ such that $\mu_{\widetilde{A}}(x) : J_x \to [0,1]$. Applying (1), we have:

$$\mu_A(x) = \{(u, \mu_{\widetilde{A}}(x)(u)); \mu_{\widetilde{A}}(x)(u) \in [0,1] \forall u \in J_x \subseteq [0,1]\}. \tag{3}$$

J_x is called the primary membership of x while $\mu_{\widetilde{A}}(x)$ is called the secondary membership of x.

Putting (2) and (3) together we have

$$\widetilde{A} = \{(x, (u, \mu_{\widetilde{A}}(x)(u))) | \mu_{\widetilde{A}}(x)(u) \in [0,1],\ \forall x \in X \wedge \forall u \in J_x \subseteq [0,1]\}. \tag{4}$$

This 'vertical representation' of a type-2 fuzzy set is used to define the concept of an *embedded set* of a type-2 fuzzy set, which is fundamental to the definition of the *centroid* of a type-2 fuzzy set.

2. Discretisation

Conventionally, discretisation is the first step in creating a computer representation of a fuzzy set (of any type). It is the process by which a continuous set is converted into a discrete set through a process of slicing. The rationale for discretisation is that a computer can process a finite number of slices, whilst it is unable to process the continuous fuzzy sets from which the slices are taken.

Definition 1 (*Slice*). A *slice* of a type-2 fuzzy set is a plane either

1. through the x-axis, parallel to the u–z plane, or
2. through the u-axis, parallel to the x–z plane.

Definition 2 (*Vertical Slice [39]*). A *vertical slice* of a type-2 fuzzy set is a plane through the x-axis, parallel to the u–z plane.

Definition 3 (*Degree of Discretisation*). The *degree of discretisation* is the separation of the slices.

[1] Most FISs rely on discretisation, though a non-discretised FIS has been realised: Coupland and John [8,9] have exploited geometry to improve the speed of inferencing in generalised type-2 fuzzy sets.

[2] This paper is concerned solely with fuzzy sets for which the (primary) domain is numeric in nature.

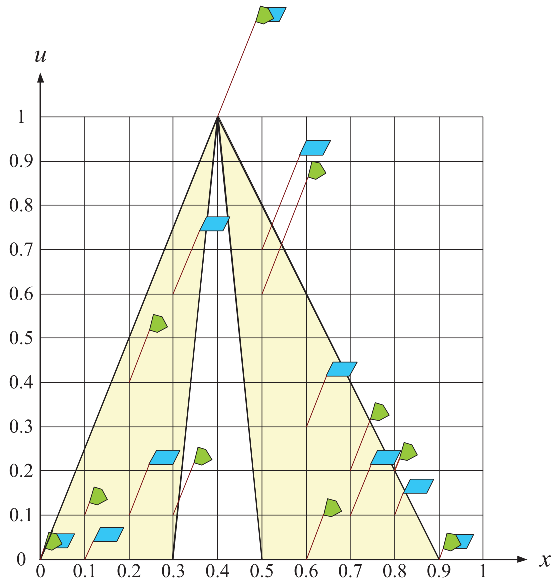


Fig. 2. Two embedded sets, indicated by different flag styles. The flag height reflects the secondary membership grade. Degree of discretisation of primary and secondary domains is 0.1. The shaded region is the FOU.

For a type-2 fuzzy set, both the primary and secondary domains are discretised, the former into vertical slices. The primary and secondary domains, which are both the unit interval $U = [0, 1]$, may have different degrees of discretisation. Furthermore the secondary domain's degree of discretisation is not necessarily constant. For type-2 fuzzy sets there is more than one discretisation strategy [11]. In the experimental evaluation reported below, we employ the grid method of discretisation [11].

3. Defuzzification of generalised type-2 fuzzy sets

For type-1 fuzzy sets defuzzification is a straightforward matter. There are several defuzzification techniques available, including the centroid, centre of maxima and mean of maxima [30]. Type-2 defuzzification of discretised type-2 fuzzy sets is a process that consists of two stages [37]:

1. type-reduction, which converts a type-2 fuzzy set to a type-1 fuzzy set, and
2. defuzzification of the type-1 fuzzy set.

Mathematically, the type-reduction algorithm depends upon the *Extension Principle* [51], which generalises operations defined for crisp numbers to type-1 fuzzy sets. Type-2 defuzzification techniques therefore derive from and incorporate type-1 defuzzification methods.[3]

3.1. The wavy-slice representation theorem

The concept of an *embedded type-2 fuzzy set (embedded set)* or *wavy-slice* [39] is crucial to type-reduction. An embedded set is a special kind of type-2 fuzzy set. It relates to the type-2 fuzzy set in which it is embedded in this way: For every primary domain value, x, there is a unique secondary domain value, u, plus the associated secondary membership grade that is determined by the primary and secondary domain values, $\mu_{\widetilde{A}}(x)(u)$.

Example 1. In Fig. 2 we have identified two embedded sets of a type-2 fuzzy set with primary and secondary domain degree of discretisation of 0.1. The embedded set $\widetilde{P}$ is represented by pentagonal, pointed flags, and embedded set $\widetilde{Q}$ is symbolised by quadrilateral shaped flags.

We can represent these embedded sets as sets of points, thus:

[3] Geometric defuzzification [9] is exceptional among type-2 defuzzification methods in not involving type-reduction and therefore not requiring type-1 defuzzification.

$$\widetilde{P} = \{[0.1/0]/0 + [0.1/0.1]/0.1 + [0.5/0.4]/0.2 + [0.5/0.1]/0.3 + [1/1]/0.4 + [0.9/0.6]/0.5 + [0.4/0]/0.6 \\ + [0.4/0.2]/0.7 + [0.2/0.2]/0.8 + [0.1/0]/0.9\}.$$

$$\widetilde{Q} = \{[0.1/0]/0 + [0.2/0]/0.1 + [0.5/0.1]/0.2 + [0.5/0.6]/0.3 + [1/1]/0.4 + [0.8/0.7]/0.5 + [0.5/0.3]/0.6 \\ + [0.5/0.1]/0.7 + [0.3/0.1]/0.8 + [0.1/0]/0.9\}.$$

Definition 4 (*Embedded Set*). Let $\widetilde{A}$ be a type-2 fuzzy set in X. For discrete universes of discourse X and U, *an embedded type-2 set* $\widetilde{A}_e$ of $\widetilde{A}$ is defined as the following type-2 fuzzy set

$$\widetilde{A}_e = \{(x_i, (u_i, \mu_{\widetilde{A}}(x_i)(u_i)))|\forall i \in \{1, \ldots, N\} : x_i \in X u_i \in J_{x_i} \subseteq U\}. \tag{5}$$

$\widetilde{A}_e$ contains exactly one element from $J_{x_1}, J_{x_2}, \ldots, J_{x_N}$, namely $u_1, u_2, \ldots, u_N$, each with its associated secondary grade, namely $\mu_{\widetilde{A}}(x_1)(u_1), \mu_{\widetilde{A}}(x_2)(u_2), \ldots, \mu_{\widetilde{A}}(x_N)(u_N)$.

Mendel and John have shown that a type-2 fuzzy set can be represented as the union of its type-2 embedded sets [39, p. 121]. This powerful result is known as the type-2 fuzzy set *Representation Theorem* or *Wavy-Slice Representation Theorem*; in [39] it was derived without reference to the Extension Principle. Bringing a conceptual simplicity to the manipulation of type-2 fuzzy sets, it is applied to give simpler derivations of results previously obtained through the Extension Principle [39].

The Representation Theorem is formally stated thus [39, p. 121]:

Let $\widetilde{A}_e^j$ denote the jth type-2 embedded set for type-2 fuzzy set $\widetilde{A}$, i.e.,

$$\widetilde{A}_e^j \equiv \left\{ \left(u_i^j, \mu_{\widetilde{A}}(x_i)(u_i^j) \right),\ i = 1, \ldots, N \right\},$$

where $\left\{ u_i^j, \ldots, u_N^j \right\} \in J_{x_i}$. Then $\widetilde{A}$ can be represented as the union of its type-2 embedded sets, i.e.,

$$\widetilde{A} = \sum_{j=1}^{n} \widetilde{A}_e^j,$$

where

$$n \equiv \prod_{i=1}^{N} M_i.$$

We regard the exhaustive defuzzification algorithm as the standard by which other algorithms must be evaluated.

The first stage of type-2 defuzzification is to create the Type-Reduced Set (TRS). Assuming that the primary domain X has been discretised, the TRS of a type-2 fuzzy set may be defined through the application of Zadeh's Extension Principle [51]. Alternatively the TRS may be defined via the Representation Theorem [39, p. 121].

Definition 5. The TRS associated with a type-2 fuzzy set $\widetilde{A}$ with primary domain X discretised into N points is

$$C_{\widetilde{A}} = \left\{ \left(\frac{\sum_{i=1}^{N} x_i \cdot u_i}{\sum_{i=1}^{N} u_i}, \mu_{\widetilde{A}}(x_1)(u_1) * \cdots * \mu_{\widetilde{A}}(x_N)(u_N) \right) \middle| \forall i \in \{1, \ldots, N\} : x_i \in X\ u_i \in J_{x_i} \subseteq U \right\}. \tag{6}$$

The type reduction stage requires the application of a t-norm ($*$) to the secondary membership grades. Because the product t-norm does not produce meaningful results for type-2 fuzzy sets with general secondary membership functions[4] it is to be avoided. For the work presented in this paper, the minimum t-norm is used.

In order for this definition of the TRS to be meaningful, the domain X must be numeric in nature. The TRS is a type-1 fuzzy set in U and its computation in practice requires the secondary domain U to be discretised as well. Algorithm 1 (adapted from Mendel [37]) is used to compute the TRS of a type-2 fuzzy set.

3.2. Exhaustive type-reduction

Mendel and John's Representation Theorem (Section 3.1) provides a precise, straightforward method for type-2 defuzzification. Though Definition 5 does not explicitly mention embedded sets, they appear implicitly in Eq. (6). When this equation is presented in algorithmic form (Algorithm 1), explicit mention is made of embedded sets. As *every* embedded set is processed, this strategy has become known as the *exhaustive method* [16]. Discretisation inevitably brings with it an element of approximation. However the exhaustive method does not introduce further inaccuracies subsequent to discretisation.

Exhaustive type-reduction processes every embedded set in turn. Each embedded set is defuzzified as a type-1 fuzzy set. The defuzzified value is paired with the minimum secondary membership grade of the embedded set. The set of ordered pairs constitutes the TRS.

[4] Under the product t-norm, $\lim_{N\to\infty}\left[\mu_{\widetilde{A}}(x_1)(u_1) * \ldots * \mu_{\widetilde{A}}(x_N)(u_N)\right] = 0$ [25, p. 201].

Algorithm 1. Type-reduction of a discretised type-2 fuzzy set to a type-1 fuzzy set, adapted from Mendel [37].

Input: a discretised generalised type-2 fuzzy set
Output: a discrete type-1 fuzzy set (the TRS)
1 **forall the** *embedded sets* **do**
2 find the minimum secondary membership grade (z);
3 calculate the primary domain value (x) of the type-1 centroid of the type-2 embedded set;
4 pair the secondary grade (z) with the primary domain value (x) to give set of ordered pairs (x,z) {some values of x may correspond to more than one value of z};
5 **end**
6 **forall the** *primary domain* (x) *values* **do**
7 select the maximum secondary grade {make each x correspond to a unique secondary domain value};
8 **end**

---

4. Efficient generalised type-reduction strategies

4.1. The sampling method

In response to the computational bottleneck engendered by exhaustive defuzzification, the sampling method, also known as the *sampling defuzzifier* [19], was devised as a cut-down version of the exhaustive method. Instead of all the embedded sets participating in type-reduction, a sample is randomly selected in order to derive an approximation for the defuzzified value. Associated with continuous type-2 fuzzy sets are an infinite number of embedded sets, and therefore the centroid values obtained via Algorithm 1 are in fact estimates of the real centroid values. Consequently discretisation in itself may be seen as a form of sampling of the continuous type-2 fuzzy set.

4.1.1. Random selection of an embedded set

Because the enumeration of all the possible embedded sets is not practical, a process of *random construction* is employed to sample them. For each primary domain value, a certain number of secondary domain (u) values lie within the FOU. For the grid method of discretisation, these are located at the grid intersections within the FOU. The construction of an embedded set requires the selection of a secondary domain value for each primary domain value. For each primary domain value, secondary domain values are selected using a random function, and therefore have the same probability of being chosen. This selection method ensures that the subsets of n embedded sets as described above constitute a random sample, but the embedded sets are not guaranteed to be unique.

4.1.2. User selected parameters

The *sample size*, i.e. the number of embedded sets, is a parameter selected by the user. A higher number of embedded sets will result in a better accuracy of defuzzification results. The *primary and secondary degrees of discretisation* are also user selected parameters. They are normally pre-selected prior to the invocation of the FIS.

4.1.3. The sampling algorithm

The user having selected the necessary parameters, the embedded sets are randomly selected and processed (Algorithm 2). The sampling method, despite having the extra stages indicated in the algorithm, is radically simpler computationally than the exhaustive method.

Algorithm 2. TRS obtained through sampling (in conjunction with the grid method of discretisation).

Input: a discretised generalised type-2 fuzzy set
Output: a discrete type-1 fuzzy set
1 select the primary domain degree of discretisation {normally pre-selected};
2 select the secondary domain degree of discretisation {normally pre-selected};
3 select the sample size;
4 **repeat**
5 randomly select (i.e. construct) an embedded set;
6 process the embedded set according to steps 2 to 4 of Algorithm 1;
7 **until** *the sample size is reached*;

4.2. The elite sampling method

The sampling algorithm (Algorithm 2) allows a given domain value to be associated with more than one secondary grade. However in *elite sampling* (Algorithm 3), each domain value is associated with only one membership grade, that being the maximum secondary grade available to the domain value (as with exhaustive type-reduction). Elite sampling, though more computationally complex than basic sampling, is designed to be more accurate in situations where there are a significant number of redundant embedded sets in the sample.

Algorithm 3. TRS obtained through elite sampling (in conjunction with the grid method of discretisation).

Input: a discretised generalised type-2 fuzzy set
Output: a discrete type-1 fuzzy set
select the primary domain degree of discretisation {normally pre-selected};
select the secondary domain degree of discretisation {normally pre-selected};
select the sample size;
repeat
  randomly select (i.e. construct) an embedded set;
  process the embedded set according to steps 2 to 4 of Algorithm 1;
until *the sample size is reached*;
forall the *primary domain* (x) *values* **do**
  select the maximum secondary grade {make each x correspond to a unique secondary domain value};
end

4.3. Vertical Slice Centroid Type-Reduction

Vertical Slice Centroid Type-Reduction (VSCTR) is a highly intuitive[5] method employed by John [23]; the paper of Lucas et al. [35] renewed interest in this strategy. In this approach the type-2 fuzzy set is cut into vertical slices, each of which is defuzzified as a type-1 fuzzy set (Algorithm 4). By pairing the domain value with the defuzzified value of the vertical slice, a type-1 fuzzy set is formed, which is easily defuzzified to give the defuzzified value of the type-2 fuzzy set. Though chronologically preceding it, this method is a generalisation of the Nie-Tan method for interval type-2 fuzzy sets [44].

Algorithm 4. VSCTR of a discretised type-2 fuzzy set to a type-1 fuzzy set.

Input: a discretised generalised type-2 fuzzy set
Output: a discrete type-1 fuzzy set (the TRS)
forall the *vertical slices* **do**
  find the defuzzified value using the centroid method;
  pair the domain value of the vertical slice with the defuzzified value to give set of ordered pairs (i.e. a type-1 fuzzy set);
end

4.4. The α-plane representation

In 2008 Liu [34,40] proposed the α-planes representation. By this technique a generalised type-2 fuzzy set is decomposed into a set of α-planes, which are horizontal slices akin to interval type-2 fuzzy sets. By repeated application of an interval defuzzification method, Liu [34] has shown that a generalised type-2 fuzzy set may be type-reduced. This method of type-reduction (Algorithm 5) is depicted in Fig. 3. By defuzzifying the resultant type-1 fuzzy set, the defuzzified value for the generalised type-2 fuzzy set is obtained.

Though the α-plane representation was envisaged as being used with the Karnik–Mendel Iterative Procedure (KMIP) [34], any interval method may be used. Any variation on the KMIP, such as the Enhanced Iterative Algorithm with Stop Condition [49] will locate the endpoints of the TRS interval. Other interval methods, such as the Greenfield-Chiclana Collapsing Defuzzifier [15], or the Nie-Tan Method [44], will defuzzify the α-plane; their defuzzified values (which will be located approximately in the centre of the interval) may then be formed into the type-1 TRS. In [16] the most accurate interval method

[5] No mathematical justification has been provided to show that VSCTR leads to the same defuzzified value as the exhaustive method.

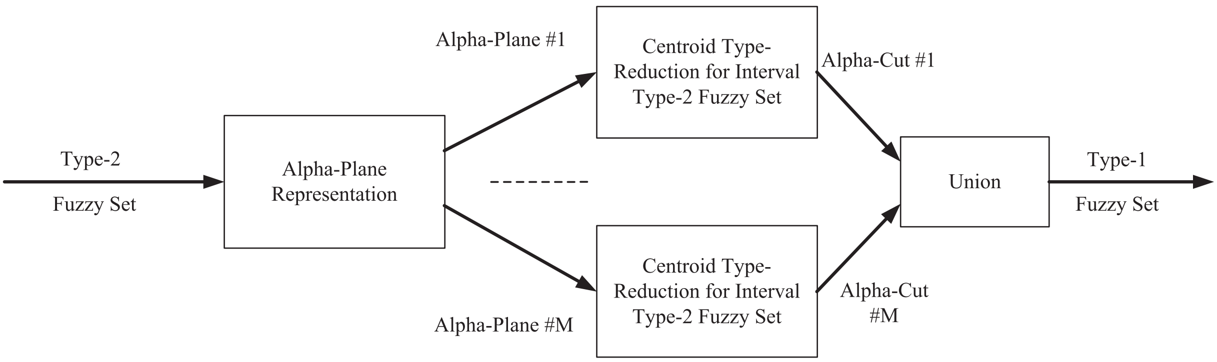


Fig. 3. Defuzzification using the α-planes representation (from Liu [34]).

was shown to be *Collapsing Outward Right–Left (CORL)*. CORL is therefore the interval technique chosen to be associated with the α-planes method for the experimental evaluation reported in Section 5.

Algorithm 5. Type-reduction of a type-2 fuzzy set to a type-1 fuzzy set using the α-plane/KMIP method.

Input: a discretised generalised type-2 fuzzy set
Output: a discrete type-1 fuzzy set
1 decompose the type-2 fuzzy set into α-planes;
2 **forall the** α-*planes* **do**
3 find the left and right endpoints using the KMIP;
4 pair each endpoint with the α-plane height to give set of ordered pairs (i.e. a type-1 fuzzy set){each α-plane is paired with two endpoints};
5 **end**

Independently to Liu, and at about the same time, Wagner and Hagras introduced the notion of zSlices [47], a concept very similar to α-planes. The α-planes/KMIP method has been modified by Zhai and Mendel [53] to increase its efficiency.

5. Experimental comparison

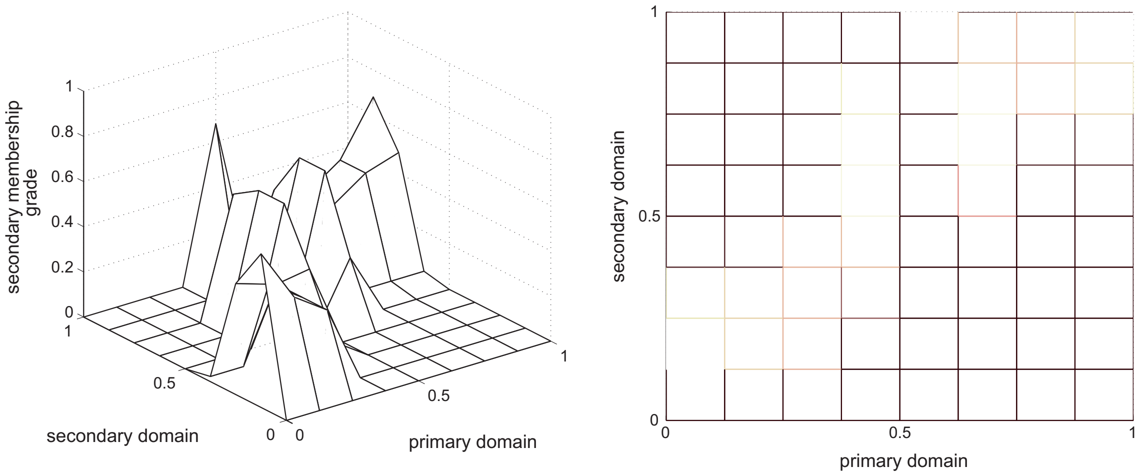
5.1. Test sets

Six FIS generated generalised type-2 fuzzy test sets were created,[6] depicted in Figs. 4–9. These are aggregated sets produced by the inferencing stage of *Fuzzer*, a prototype type-2 FIS [10]. For each inference the degree of discretisation adopted was sufficiently coarse to allow exhaustive defuzzification; without the benchmark defuzzified values obtained through exhaustive defuzzification, the methods could not have been compared for accuracy. Three rule sets were used. For each rule set the FIS was run with two distinct sets of parameters.[7] The FIS generated test sets were chosen because of the complexity and lack of symmetry evident in their graphs; their benchmark defuzzified values were found by exhaustive defuzzification. The three rule sets are shown in Tables 2–4. Table 5 contains a summary of the features of the test sets.

Heater FIS
This FIS is designed to calculate the desirable setting for a heater. It has five rules and two inputs which are tabulated in Table 2.

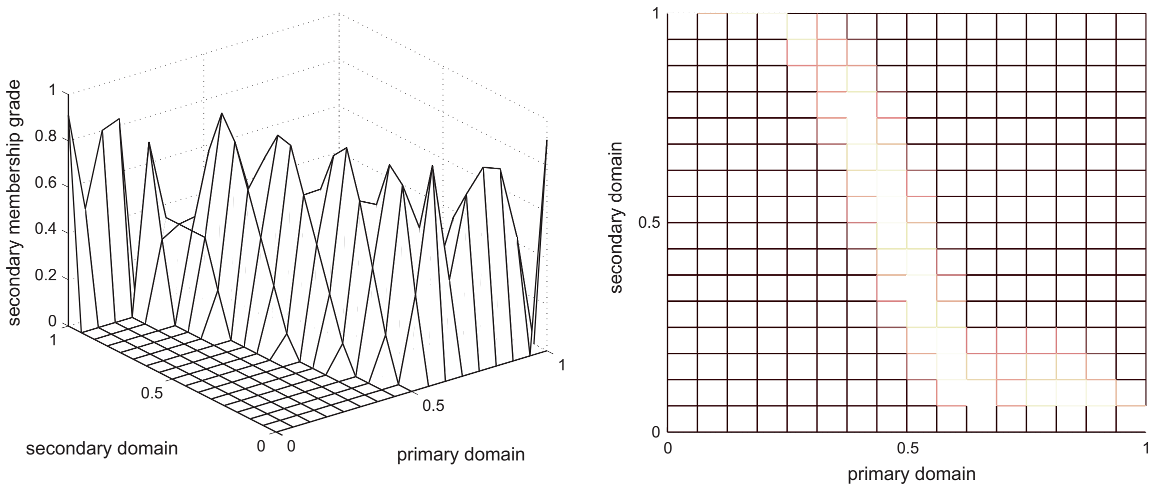
Washing Powder FIS
The purpose of this FIS is to determine the amount of washing powder required by a washing machine for a given wash load. It has four rules and three inputs which are summarised in Table 3.

[6] The initial intention was to include Liu's two generalised type-2 fuzzy test sets [34, pp. 2230–2233]. However this was not feasible, since (1) for Case A the secondary membership functions are derived by a random procedure and therefore cannot be recreated, and (2) in Case B the secondary membership functions are too similar to interval membership functions for this set to be of value as a generalised test set.
[7] For example Heater0p0625 is not a finer version of Heater0p125; it uses different parameters for the input rules. That these two test sets are completely different can be clearly seen from their 3D representations (Fig. 4 and Fig. 5).



(a) 3-D representation. (b) FOU.

Fig. 4. HeaterFIS0.125—Heater FIS generated generalised test set, domain degree of discretisation 0.125.



(a) 3-D representation. (b) FOU.

Fig. 5. HeaterFIS0.0625—Heater FIS generated generalised test set, domain degree of discretisation 0.0625.

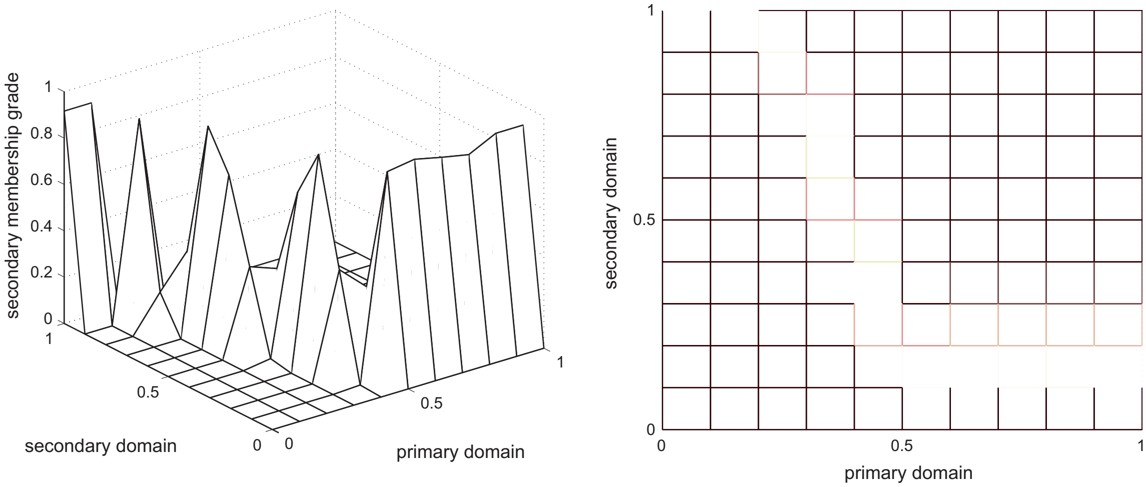
Shopping FIS

This FIS is designed to answer the dilemma of whether to go shopping by car, or walk, depending on weather conditions, amount of shopping, etc. The defuzzified value is therefore rounded to one of two possible answers. The FIS has four rules and three inputs as tabulated in Table 4.

5.2. Methodology for generalised methods comparison

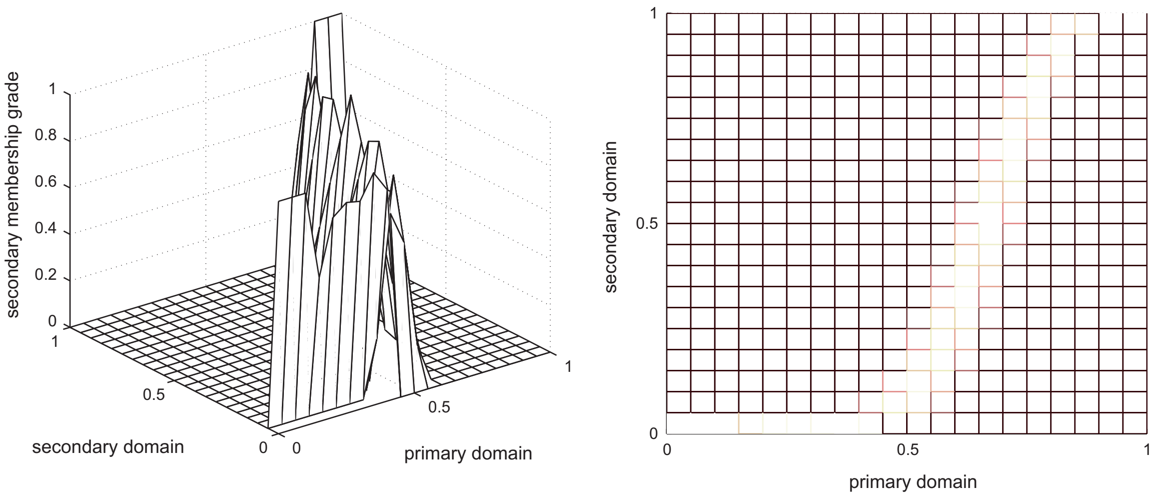
The six test sets were defuzzified using the following techniques:

1. The exhaustive method (as a benchmark for accuracy),
2. VSCTR,
3. the sampling method using sample sizes of 50, 100, 250, 500, 750, 1000, 5000, 10,000, 50,000 and 100,000,
4. the elite sampling method using sample sizes of 50, 100, 250, 500, 750, 1000, 5000, 10,000, 50,000 and 100,000,
5. the α-planes/CORL method using 3, 5, 9, 11, 21, 51, 101, 1001, 10,001 and 100,001 α-planes, and



(a) 3-D representation. (b) FOU.

Fig. 6. PowderFIS0.1—Powder FIS generated generalised test set, domain degree of discretisation 0.1.



(a) 3-D representation. (b) FOU.

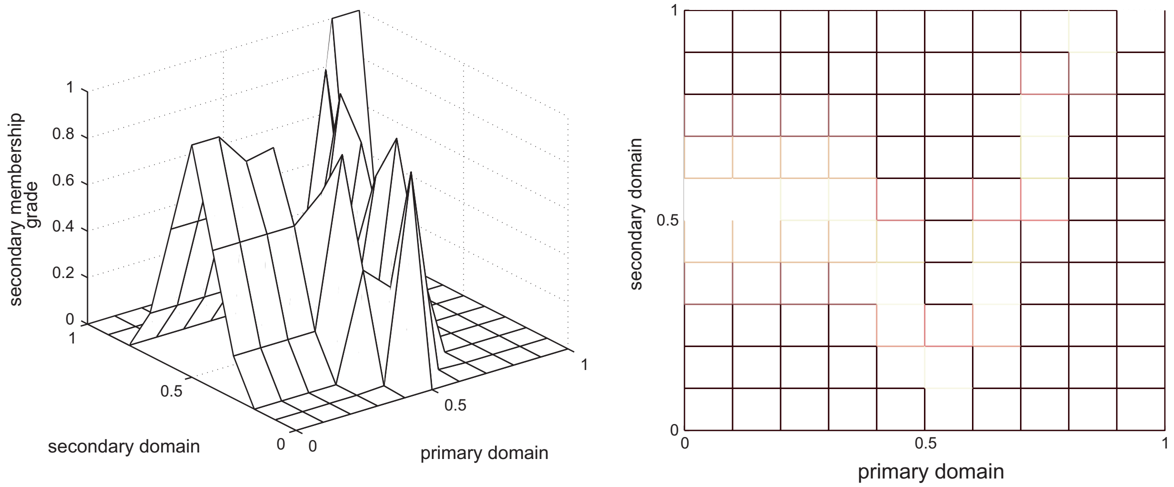
Fig. 7. PowderFIS0.05—Powder FIS generated generalised test set, domain degree of discretisation 0.05.

6. the α-planes/Interval Exhaustive method using 3, 5, 9, 11, 21, 51, 101, 1001, 10,001 and 100,001 α-planes (as an evaluation of the accuracy of the α-planes representation itself).[8]

For each test run the defuzzified value and the defuzzification time were recorded. For the timings, in most instances, multiple runs were performed and the times averaged to give results of greater accuracy than those that would have been obtained from a single run.

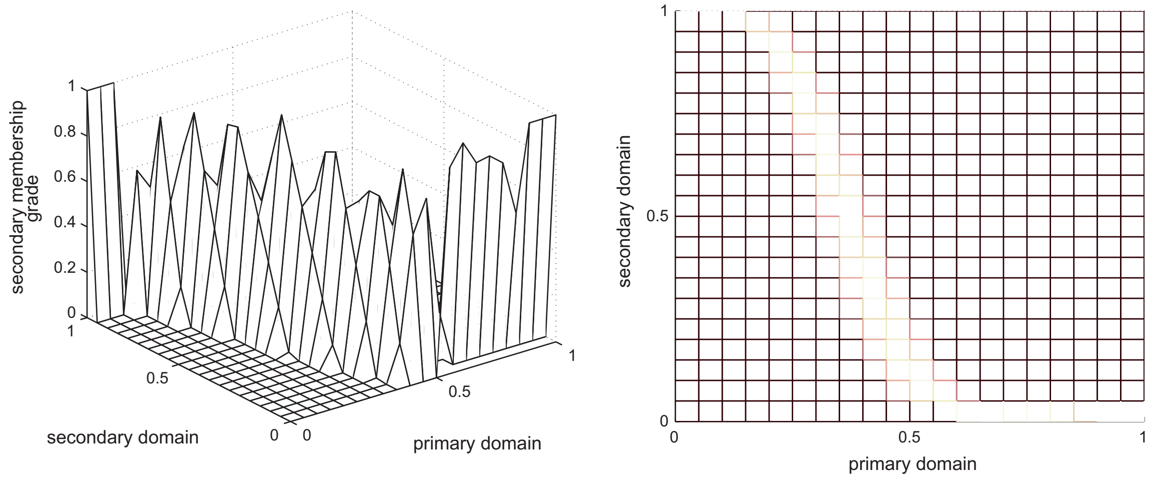
The defuzzification methods were coded in Matlab™ and tested on a laptop with an AMD Turion II Neo K645 CPU, a clock speed of 1.6 GHz, and a 4096 MB 1333 MHz Dual Channel DDR3 SDRAM, running the MS Windows® 7 SP1 Home Premium 64 bit operating system. For timings, the defuzzification software was run as a process with priority higher than that of the operating system, so as to eliminate, as far as possible, timing errors caused by other operating system processes.

[8] The extremely long processing times prevented defuzzification using 10,001 and 100,001 α-planes with test sets HeaterFIS0.0625, PowderFIS0.05 and ShoppingFIS0.05.



(a) 3-D representation. (b) FOU.

Fig. 8. ShoppingFIS0.1—Shopping FIS generated generalised test set, domain degree of discretisation 0.1.



(a) 3-D representation. (b) FOU.

Fig. 9. ShoppingFIS0.05—Shopping FIS generated generalised test set, domain degree of discretisation 0.05.

Table 2
Heater FIS rules.

Inputs		Outputs
Temperature	Date	Heating
Cold	–	High
–	Winter	High
Hot	Not winter	Low
–	Spring	Medium
–	Autumn	Medium

Table 3
Washing Powder FIS rules.

Inputs			Outputs
Washing	Water	Pre-soak	Powder
Very dirty	–	–	A lot
–	Hard	–	A lot
Slightly dirty	Soft	–	A bit
–	–	Lengthy	a Bit

Table 4
Shopping FIS rules.

Inputs			Outputs
Distance	Shopping	Weather	Travel method
Short	Light	–	Walk
Long	–	–	Go by car
–	Heavy	–	Go by car
–	–	Raining	Go by car

Table 5
Features of the generalised test sets.

Test set	Normal FOU	Normal sec. MF	Narrow FOU	No. of emb. sets
HeaterFIS0.125	Yes	No	No	14,580
HeaterFIS0.0625	Yes	No	Yes	13,778,100
PowderFIS0.1	Yes	No	Yes	24,300
PowderFIS0.05	Yes	Yes	Yes	3,840,000
ShoppingFIS0.1	Yes	Yes	No	312,500
ShoppingFIS0.05	Yes	Yes	Yes	3,840,000

6. Discussion of the test results

The test results are tabulated in Appendices A–F (Tables A.6–F.29), which record the defuzzified values, errors, and timings. The elite sampling tables (Tables A.8, B.12, C.16, D.20, E.24 and F.28) show data relating to non-redundant embedded sets.[9] The timings indicate that VSCTR is the most efficient method of those tested.

6.1. Statistical comparison of the methods

We now present a rigourous statistical analysis of the test results for accuracy. The hypothesis that we are testing in this subsection can be stated as follows:

The sampling, elite sampling, VSCTR and α-planes/CORL methods do not produce significantly different defuzzified values.

To compare each pair of methods we have to analyse two related samples, the defuzzified values obtained by each method's application to the same six test sets referred to above. The usual parametric test to use in these cases is the *t*-test applied to the difference scores. This test requires for its application the assumption of normality and independent distribution of the difference scores in the population from which the six test sets are drawn.[10] However, on the one hand, we consider these assumptions to be unjustifiable in our context since there is no evidence to support them, i.e. we have no information about the nature of the population from which the six test sets are drawn nor do we have any knowledge about any of its parameters. Also, by not requiring these stringent assumptions we can, on the other hand, achieve greater generality in our conclusions. Therefore, we conclude that nonparametric tests are most appropriate in our experimental study; we will use the Wilcoxon Matched-Pairs Signed-Ranks Test [43] to be described in the next subsection.

6.1.1. Wilcoxon Matched-pairs signed-ranks statistical test

Let $X_1, X_2, \ldots, X_n$ be a random sample of size n from some unknown continuous distribution function F. Let p be a positive real number, $0 < p < 1$, and let $\xi_p(F)$ denote the quantile of order p for the distribution function F, that is, $\xi_p(F)$ is a solution of $F(x) = p$. For $p = 0.5$, $\xi_{0.5}(F)$ is known as the median of F.

[9] A Non-Redundant Embedded Set (NRES) is an embedded set that is not eliminated during elite sampling.
[10] Although we did not apply any specific random sampling method, we consider the set of six test sets to constitute a sample representative of the whole set of generalised type-2 fuzzy sets.

A problem of location is set up by testing the null hypothesis $H_0: \xi_p(F) = \xi_0$ against one of the alternatives $\xi_p(F) > \xi_0$, $\xi_p(F) < \xi_0$ or $\xi_p(F) \neq \xi_0$. The Wilcoxon Signed-Ranks Test provides a statistical hypothesis test which takes into account the magnitude of the difference between the observations and the hypothesised quantile in order to solve the issue of location.

Let $H_0: \xi_{0.5}(F) = \xi_0$ be the null hypothesis. Consider the differences $D_i = X_i - \xi_0$, $i = 1, 2, \ldots, n$. Under H_0, the expected number of negative differences will be $n/2$ and negative and positive differences of equal absolute magnitude should occur with equal probability. Consider the absolute values $|D_1|, |D_2|, \ldots, |D_n|$ and rank them from 1 to n. Let T_+ be the sum of ranks assigned to those $D_i's$ that are positive and T_- be the sum of ranks assigned to those $D_i's$ that are negative. It follows that

$$T_+ + T_- = \sum_{k=1}^{n} k = \frac{n(n+1)}{2}$$

so T_+ and T_- are linearly related and offer equivalent criteria. A large value of T_+ indicates that most of the larger ranks are assigned to positive $D_i's$. It follows that large values of T_+ support $H_1: \xi_{0.5}(F) > \xi_0$. A similar analysis applies to the other two alternatives. So, the test rejects $H_0: \xi_{0.5}(F) = \xi_0$ to accept $H_1: \xi_{0.5}(F) > \xi_0$ if $T_+ > c_1$, it rejects H_0 to accept $H_1: \xi_{0.5}(F) < \xi_0$ if $T_- > c_2$ and it rejects H_0 to accept $H_1: \xi_{0.5}(F) \neq \xi_0$ if $T_+ > c_3$ or $T_- > c_4$ where c_i are the critical region values.

Under H_0, the common distribution of T_+ and T_- is symmetric with mean $E[T_+] = n(n+1)/4$ and variance $\mathrm{var}[T_+] = n(n+1)(2n+1)/24$. For large n, the standardised T_+ has approximately a standard normal distribution.

In the case of matched-paired data $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$ obtained from the application of two treatments (in our case – two generalised defuzzification methods) to the same set of subjects (in our case – the set of six test sets), in order to test $H_0 : \xi_{0.5}(F_{X_i - Y_i}) = \xi_0$ against one-sided or two-sided alternatives, the Wilcoxon Test is performed exactly as above by taking $D_i = X_i - Y_i - \xi_0$. In our study we want to test whether the application of the different generalised defuzzification methods produces significantly different defuzzified values, i.e. we are testing a null hypothesis with a value $\xi_0 = 0$, $H_0 : \xi_{0.5}(F_{X_i - Y_i}) = 0$. We are testing against the alternative hypothesis of method X being more accurate than method Y, so we will use one-tailed testing $H_1 : \xi_{0.5}(F_{X_i - Y_i}) < 0$.

We assume that two measures with test p-value under the null hypothesis lower than or equal to 0.05 (α) will be considered as significantly different; we refer to it as the test being significant and therefore we conclude that the null hypothesis tested is to be rejected. Otherwise, we will fail to reject the null hypothesis.

6.1.2. Results of the Wilcoxon Signed Rank Test

We wanted to test whether there is a significant difference in accuracy between the four generalised methods. The methods may be paired in six ways. For the α-planes/CORL method we analysed the results obtained by using the highest number of α-planes (100,001), so that discretisation effects on the u-axis would be eliminated as far as possible. For the sampling and elite sampling methods, the Wilcoxon Tests were applied at the sample sizes used in the test runs.

The Wilcoxon Signed Rank Test results, presented in Tables G.30–G.39, reveal a more complex and interesting picture than that revealed by simply ranking the test results. Taking each pair of comparisons in turn,

1. For every sample size the sampling and VSCTR methods do not produce significantly different defuzzified values.
2. For sample sizes up to and including 5000 the elite sampling and VSCTR methods do not produce significantly different defuzzified values. For sample sizes of 10,000 and above there is evidence to support the elite sampling method being more accurate than VSCTR.
3. For sample sizes up to and including 1000 the sampling and elite sampling methods do not produce significantly different defuzzified values. For sample sizes over 5000 there is evidence to support the elite sampling method being more accurate than sampling method.
4. For every sample size there is evidence to support VSCTR being more accurate than the α-planes/CORL method.

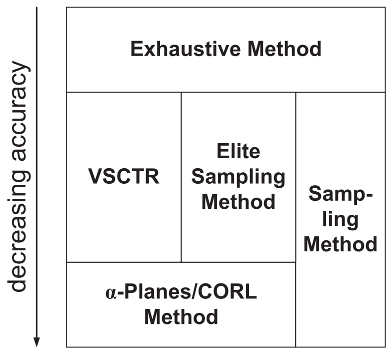


Fig. 10. Hierarchy of type-2 defuzzification methods' performance in relation to accuracy, for sample sizes 50 and 100. The exhaustive method is used as a benchmark.

5. For sample sizes of 50 and 100 the sampling and α-planes/CORL methods do not produce significantly different defuzzified values. For sample sizes of 250 and above there is evidence to support the sampling method being more accurate than the α-planes/CORL method.
6. For every sample size there is evidence to support the elite sampling method being more accurate than the α-planes/CORL method.

Figs. 10–13 display these relative accuracies graphically.

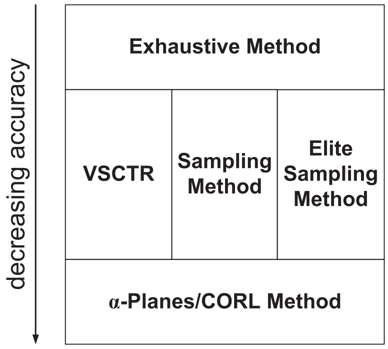


Fig. 11. Hierarchy of type-2 defuzzification methods' performance in relation to accuracy, for sample sizes 250, 500, 750 and 1000. The exhaustive method is used as a benchmark.

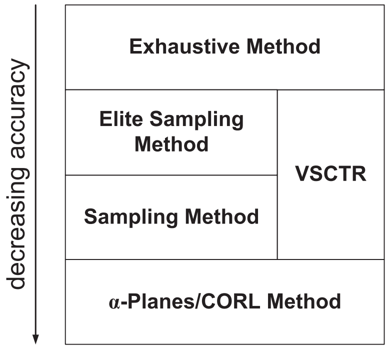


Fig. 12. Hierarchy of type-2 defuzzification methods' performance in relation to accuracy, for sample size 5000. The exhaustive method is used as a benchmark.

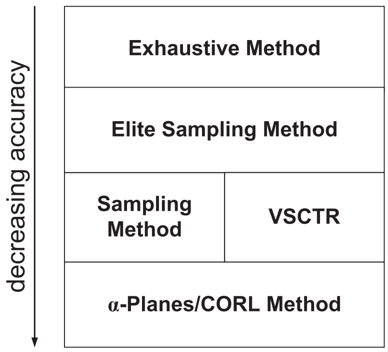


Fig. 13. Hierarchy of type-2 defuzzification methods' performance in relation to accuracy, for sample sizes 10,000–100,000. The exhaustive method is used as a benchmark.

7. Conclusions

Several conclusions may be drawn from this investigation:

Most accurate method: The experimental evaluation reveals a complex picture as regards the relative accuracies of the methods:

- The α-planes/CORL method is the least accurate of the techniques assessed apart from when compared with the sampling method at low sample sizes (50 and 100). In these instances, there is no evidence to support the sampling method being more accurate than the α-planes/CORL method.
- For sample sizes of 5000 and above elite sampling is more accurate than sampling.[11]
- For high sample sizes (10,000, 50,000 and 100,000) the elite sampling method is the most accurate of the techniques compared. However, with such high sample sizes, it may be argued that elite sampling is barely distinguishable from the exhaustive method, especially when the set to be defuzzified has a low number of embedded sets.
- For samples of moderate size (250, 500, 750 and 1000), VSCTR, sampling, and elite sampling are of equivalent accuracy.

Fastest method: VSCTR is undoubtedly the fastest method for defuzzification of type-2 fuzzy sets; none of the other methods challenge VSCTR for speed, no matter how low the sample size in the case of the sampling method, or the number of α-planes employed by the α-planes method.

Convergence of the α-planes/CORL results: As the number of α-planes increases, the α-planes/CORL results do not converge to the value obtained by generalised exhaustive defuzzification. Furthermore even the α-planes/interval exhaustive results (Tables A.9, C.17 and E.25) fail to converge to this value. The defuzzified values for both the α-planes/CORL and α-planes/interval exhaustive methods are similar (to a precision of about four decimal places) and appear to converge to the same number, which is *not* the value obtained from generalised exhaustive defuzzification. This discrepancy is indicative of an issue with the α-planes method itself, and has been previously reported in [18,12].

In summary, the results reported in this paper will motivate the development of generalised type-2 fuzzy applications. They will also help researchers in selecting the most appropriate defuzzification method for the application of generalised type-2 fuzzy logic in areas such as perceptual computing [24], fuzzy logic control [39], diagnostic medicine [45] and clustering [33].

8. Further work

Out of the research presented in this paper, certain issues have emerged that would benefit from further work:

Standard method of discretisation Investigate the accuracy and efficiency of the generalised type-2 defuzzification methods when implemented using the standard method of discretisation. We would expect there to be far fewer, if any, redundant embedded sets, and that consequently elite sampling would outperform sampling for accuracy, even at low sample sizes.

α-Planes method Investigate why the defuzzified value obtained through the α-planes method does not converge to the exhaustive defuzzified value as the number of α-planes is increased.

Appendix A. Generalised test set Heater0.125

Tables A.6, A.7, A.8, A.9.

Appendix B. Generalised test set Heater0.0625

Tables B.10, B.11, B.12, B.13.

Appendix C. Generalised test set Powder0.1

Tables C.14, C.15, C.16, C.17.

[11] Tables A.8, B.12, C.16, D.20, E.24 and F.28 show that there are numerous redundant embedded sets, which when eliminated from the calculation during elite sampling, leave few non-redundant embedded sets, making the effective sample size much smaller. For low sample sizes, elite sampling is not an improvement on sampling, but for higher sample sizes, elite sampling outperforms sampling, as even after the redundant embedded sets have been discarded, there are still sufficient to give a good approximation to the exhaustive defuzzified value.

Table A.6
Exhaustive and VSCTR results for the HeaterFIS0.125 test set.

Exhaustive defuzzified value	No. of emb. sets	No. of NRESs	Exhaustive timing	VSCTR defuzzified value	VSCTR error	VSCTR timing
0.6313618377	14,580	486	1.37 s	0.6327431582	0.0013813205	0.000268 s

Table A.7
Sampling results for the HeaterFIS0.125 test set. Number of embedded sets = 14,580. Percentage of embedded sets sampled $= \frac{\text{sample size}}{\text{number of embedded sets}} \times 100$. Exhaustive defuzzified value = 0.6313618377. Errors marked '◁' are lower than the corresponding errors for the elite sampling method.

Sample size	Percent. of emb. sets sampled (%)	Sampling defuzzified value	Sampling method error	Sampling method timing
50	0.34	0.6235041770	0.0078576607 ◁	0.0188 s
100	0.69	0.6255373882	0.0058244495 ◁	0.0377 s
250	1.71	0.6299440373	0.0014178004 ◁	0.0937 s
500	3.43	0.6262109521	0.0051508856	0.188 s
750	5.14	0.6263047645	0.0050570732	0.282 s
1000	6.86	0.6246724480	0.0066893897	0.377 s
5000	34.29	0.6251282506	0.0062335871	1.93 s
10,000	68.59	0.6256899730	0.0056718647	4.03 s
50,000	342.94	0.6252882201	0.0060736176	39.1 s
100,000	685.87	0.6254891164	0.0058727213	2.34 min

Table A.8
Elite sampling results for the HeaterFIS0.125 test set. Number of embedded sets = 14,580. Percentage of embedded sets sampled $= \frac{\text{sample size}}{\text{number of embedded sets}} \times 100$. Exhaustive defuzzified value = 0.6313618377. Errors shown in bold are smaller than the VSCTR error. Errors marked '◁' are lower than the corresponding errors for the sampling method.

Sample size	Percentage of ESs sampled (%)	No. of NRESs in sample	NRESs as percentage of sample size (%)	NRESs as percentage of ESs (%)	Elite sampling defuzzified value	Elite sampling method error	Elite sampling method timing (s)
50	0.34	43	86.00	0.29	0.6167008518	0.0146609859	0.0216
100	0.69	80	80.00	0.55	0.6228289822	0.0085328555	0.0432
250	1.71	147	58.80	1.01	0.6295092983	0.0018525394	0.108
500	3.43	221	44.20	1.52	0.6279777881	0.0033840496 ◁	0.215
750	5.14	252	33.60	1.73	0.6295741240	0.0017877137 ◁	0.328
1000	6.86	278	27.80	1.91	0.6293312286	0.0020306091 ◁	0.432
5000	34.29	408	8.16	2.80	0.6306797744	**0.0006820633** ◁	2.16
10,000	68.59	438	4.38	3.00	0.6310563647	**0.0003054730** ◁	5.01
50,000	342.94	486	0.97	3.33	0.6311913249	**0.0001705128** ◁	22.8
100,000	685.87	486	0.49	3.33	0.6313618377	**0.0000000000** ◁	42.9

Table A.9
α-Planes/CORL and α-planes/interval exhaustive results for the HeaterFIS0.125 test set. Exhaustive defuzzified value = 0.6313618377. Error = α-planes value − exhaustive value.

No. of α-planes	α-Planes/CORL defuzzified value	α-Planes/CORL error	α-Planes/CORL timing (s)	α-Planes/interval exhaustive defuzzified value	α-Planes/interval exhaustive error
3	0.5974411770	−0.0339206607	0.000566	0.5974395543	−0.0339222834
5	0.6014928819	−0.0298689558	0.000804	0.6014844463	−0.0298773914
9	0.6220020252	−0.0093598125	0.00154	0.6219954766	−0.0093663611
11	0.6202108548	−0.0111509829	0.00178	0.6202019617	−0.0111598760
21	0.6176529687	−0.0137088690	0.00346	0.6176441546	−0.0137176831
51	0.6149638697	−0.0163979680	0.00851	0.6149552604	−0.0164065773
101	0.6146818722	−0.0166799655	0.0169	0.6146732228	−0.0166886149
1001	0.6149166283	−0.0164452094	0.166	0.6149079069	−0.0164539308
10,001	0.6149818425	−0.0163799952	1.77	0.6149731309	−0.0163887068
100,001	0.6149818643	−0.0163799734	59.6	0.6149731532	−0.0163886845

Table B.10
Exhaustive and VSCTR results for the HeaterFIS0.0625 test set.

Exhaustive defuzzified value	No. of emb. sets	No. of NRESs	Exhaustive timing	VSCTR defuzzified value	VSCTR error	VSCTR timing
0.2621587894	13,778,100	2774	25.1 min	0.2592117473	0.0029470421	0.000453 s

Table B.11
Sampling results for the HeaterFIS0.0625 test set. Number of embedded sets = 13,778,100. Percentage of embedded sets sampled $= \frac{\text{sample size}}{\text{number of embedded sets}} \times 100$. Exhaustive defuzzified value = 0.2621587894. Errors shown in bold are smaller than the VSCTR error. Errors marked '⊲' are lower than the corresponding errors for the elite sampling method.

Sample size	Percent. of emb. sets sampled (%)	Sampling defuzzified value	Sampling method error	Sampling method timing
50	0.0004	0.2634998330	**0.0013410436** ⊲	0.0306 s
100	0.0007	0.2643678735	**0.0022090841** ⊲	0.0609 s
250	0.0018	0.2639954015	**0.0018366121** ⊲	0.152 s
500	0.0036	0.2644544522	**0.0022956628** ⊲	0.305 s
750	0.0054	0.2641746630	**0.0020158736** ⊲	0.458 s
1000	0.0073	0.2646109558	**0.0024521664** ⊲	0.609 s
5000	0.0363	0.2645765948	**0.0024178054**	3.11 s
10,000	0.0726	0.2645380675	**0.0023792781**	6.38 s
50,000	0.3629	0.2644304187	**0.0022716293**	51.9 s
100,000	0.7258	0.2645136689	**0.0023548795**	2.74 min

Table B.12
Elite sampling results for the HeaterFIS0.0625 test set. Number of embedded sets = 13,778,100. Percentage of embedded sets sampled $= \frac{\text{sample size}}{\text{number of embedded sets}} \times 100$. Exhaustive defuzzified value = 0.2621587894. Errors shown in bold are smaller than the VSCTR error. Errors marked '⊲' are lower than the corresponding errors for the sampling method.

Sample size	Percentage of ESs sampled (%)	No. of NRESs in sample	NRESs as percentage of sample size (%)	NRESs as percentage of ESs (%)	Elite sampling defuzzified value	Elite sampling method error	Elite sampling method timing
50	0.0004	48	96.00	0.0003	0.2663946510	0.0042358616	0.0330 s
100	0.0007	94	94.00	0.0007	0.2650536879	**0.0028948985**	0.0660 s
250	0.0018	217	86.8o	0.0016	0.2654055130	0.0032467236	0.165 s
500	0.0036	358	71.60	0.0026	0.2645029187	**0.0023441293**	0.330 s
750	0.0054	491	65.47	0.0036	0.2642948052	**0.0021360158**	0.495 s
1000	0.0073	606	60.60	0.0044	0.2646455386	**0.0024867492**	0.661 s
5000	0.0363	1140	22.80	0.0083	0.2637295835	**0.0015707941** ⊲	3.32 s
10,000	0.0726	1355	13.55	0.0098	0.2635483884	**0.0013895990** ⊲	6.67 s
50,000	0.3629	1809	3.62	0.0131	0.2631277384	**0.0009689490** ⊲	33.5 s
100,000	0.7258	1958	1.96	0.0142	0.2629459523	**0.0007871629** ⊲	1.12 min

Table B.13
α-Planes/CORL and α-planes/interval exhaustive results for the HeaterFIS0.0625 test set. Exhaustive defuzzified value = 0.2621587894. Error = α-planes value − exhaustive value.

No. of α-planes	α-Planes/CORL defuzzified value	α-Planes/CORL error	α-Planes/CORL timing (s)	α-Planes/interval exhaustive defuzzified value	α-Planes/interval exhaustive error
3	0.2911992286	0.0290404392	0.000900 s	0.2912056106	0.0290468212
5	0.2843138916	0.0221551022	0.00170 s	0.2843202930	0.0221615036
9	0.2781833083	0.0160245189	0.00329 s	0.2781887468	0.0160299574
11	0.2791783831	0.0170195937	0.00408 s	0.2791839651	0.0170251757
21	0.2839726877	0.0218138983	0.00769 s	0.2839784863	0.0218196969
51	0.2845058809	0.0223470915	0.0185 s	0.2845118383	0.0223530489
101	0.2857499961	0.0235912067	0.0365 s	0.2857559640	0.0235971746
1001	0.2836509843	0.0214921949	0.367 s	0.2836568708	0.0214980814
10,001	0.2835417182	0.0213829288	3.88 s	–	–
100,001	0.2835490870	0.0213902976	1.94 min	–	–

Appendix D. Generalised test set Powder0.05

Tables D.18, D.19, D.20, D.21.

Table C.14
Exhaustive and VSCTR results for the PowderFIS0.1 test set.

Exhaustive defuzzified value	No. of emb. sets	No. of NRESs	Exhaustive timing	VSCTR defuzzified value	VSCTR error	VSCTR timing
0.2806983775	24,300	1701	2.55 s	0.2646964681	0.0160019094	0.000310 s

Table C.15
Sampling results for the PowderFIS0.1 test set. Number of embedded sets = 24,300. Exhaustive defuzzified value = 0.2806983775. Percentage of embedded sets sampled= $\frac{\text{sample size}}{\text{number of embedded sets}} \times 100$. Errors shown in bold are smaller than the VSCTR error. Errors marked '◁' are lower than the corresponding errors for the elite sampling method.

Sample size	Percent. of emb. sets sampled (%)	Sampling defuzzified value	Sampling method error	Sampling method timing
50	0.21	0.2959967354	**0.0152983579**	0.0227 s
100	0.41	0.2983068036	0.0176084261	0.0453 s
250	1.03	0.2879898240	**0.0072914465** ◁	0.113 s
500	2.06	0.2883575902	**0.0076592127** ◁	0.225 s
750	3.09	0.2904003138	**0.0097019363**	0.340 s
1000	4.12	0.2885932629	**0.0078948854**	0.454 s
5000	20.58	0.2893665435	**0.0086681660**	2.32 s
10,000	41.15	0.2894760075	**0.0087776300**	4.84 s
50,000	205.76	0.2893699018	**0.0086715243**	43.9 s
100,000	411.52	0.2896395345	**0.0089411570**	2.46 min

Table C.16
Elite sampling results for the PowderFIS0.1 test set. Number of embedded sets = 24,300. Exhaustive defuzzified value = 0.2806983775. Percentage of embedded sets sampled = $\frac{\text{sample size}}{\text{number of embedded sets}} \times 100$. Errors shown in bold are smaller than the VSCTR error. Underlined errors are lower than the errors for the α-planes method, for all numbers of α-planes. Errors marked '◁' are lower than the corresponding errors for the sampling method.

Sample size	Percentage of ESs sampled (%)	No. of NRESs in sample	NRESs as percentage of sample size (%)	NRESs as percentage of ESs (%)	Elite sampling defuzzified value	Elite sampling method error	Elite sampling method timing (s)
50	0.21	48	96.00	0.20	0.2924967750	**0.0117983975** ◁	0.0254
100	0.41	94	94.00	0.39	0.2888488173	**0.0081504398** ◁	0.0510
250	1.03	199	79.60	0.82	0.2886109164	**0.0079125389**	0.127
500	2.06	360	72.00	1.48	0.2889039270	**0.0082055495**	0.254
750	3.09	477	63.60	1.96	0.2879727936	**0.0072744161** ◁	0.381
1000	4.12	567	56.70	2.33	0.2884496935	**0.0077513160** ◁	0.512
5000	20.58	1120	22.40	4.61	0.2838675377	<u>**0.0031691602**</u> ◁	2.56
10,000	41.15	1366	13.66	5.62	0.2829860213	<u>**0.0022876438**</u> ◁	5.13
50,000	205.76	1661	3.32	6.84	0.2807352287	<u>**0.0000368512**</u> ◁	25.8
100,000	411.52	1698	1.70	6.99	0.2807555453	<u>**0.0000571678**</u> ◁	51.7

Table C.17
α-Planes/CORL and α-planes/interval exhaustive results for the PowderFIS0.1 test set. Exhaustive defuzzified value = 0.2806983775. Error = α-planes value − exhaustive value. Errors shown in bold are smaller than the VSCTR error. The error marked '*' is lower than every error for the sampling method.

No. of α-planes	α-Planes/CORL defuzzified value	α-Planes/CORL error	α-Planes/CORL timing (s)	α-Planes/interval exhaustive defuzzified value	α-Planes/interval exhaustive error
3	0.3100683482	0.0293699707	0.000653 s	0.3100714646	0.0293730871
5	0.2990422650	0.0183438875	0.00123 s	0.2990446820	0.0183463045
9	0.2949801671	**0.0142817896**	0.00238 s	0.2949820128	0.0142836353
11	0.2860659413	**0.0053675638***	0.00296 s	0.2860677799	0.0053694024
21	0.2903153362	**0.0096169587**	0.00557 s	0.2903173044	0.0096189269
51	0.2928824383	**0.0121840608**	0.0133 s	0.2928844669	0.0121860894
101	0.2909066603	**0.0102082828**	0.0267 s	0.2909086286	0.0102102511
1001	0.2907821474	**0.0100837699**	0.267 s	0.2907840999	0.0100857224
10,001	0.2907215619	**0.0100231844**	2.88 s	0.2907235112	0.0100251337
100,001	0.2907192214	**0.0100208439**	1.89 min	0.2907211701	0.0100227926

Appendix E. Generalised test set Shopping0.1

Tables E.22, E.23, E.24, E.25.

Table D.18
Exhaustive and VSCTR results for the PowderFIS0.05 test set.

Exhaustive defuzzified value	No. of emb. sets	No. of NRESs	Exhaustive timing	VSCTR defuzzified value	VSCTR error	VSCTR timing
0.8180632180	3,840,000	5093	8.22 min	0.8185912163	0.0005279983	0.000555 s

Table D.19
Sampling results for the PowderFIS0.05 test set. Number of embedded sets = 3,840,000. Exhaustive defuzzified value = 0.8180632180. Percentage of embedded sets sampled $= \frac{\text{sample size}}{\text{number of embedded sets}} \times 100$. Error shown in bold is smaller than the VSCTR error. Errors marked '◁' are lower than the corresponding errors for the elite sampling method.

Sample size	Percent. of emb. sets sampled (%)	Sampling defuzzified value	Sampling method error	Sampling method timing
50	0.001	0.8165757956	0.0014874224 ◁	0.0326 s
100	0.003	0.8173514791	0.0007117389 ◁	0.0648 s
250	0.007	0.8176368830	**0.0004263350** ◁	0.162 s
500	0.013	0.8166109316	0.0014522864	0.323 s
750	0.020	0.8166335918	0.0014296262	0.485 s
1000	0.026	0.8165791599	0.0014840581	0.647 s
5000	0.130	0.8171269807	0.0009362373	3.30 s
10,000	0.260	0.8169971802	0.0010660378	6.73 s
50,000	1.302	0.8168484040	0.0012148140	54.4 s
100,000	2.604	0.8168981632	0.0011650548	2.82 min

Table D.20
Elite sampling results for the PowderFIS0.05 test set. Number of embedded sets = 3,840,000. Percentage of embedded sets sampled $= \frac{\text{sample size}}{\text{number of embedded sets}} \times 100$. Exhaustive defuzzified value = 0.8180632180. Errors shown in bold are smaller than the VSCTR error. Errors marked '◁' are lower than the corresponding errors for the sampling method.

Sample size	Percentage of ESs sampled (%)	No. of NRESs in sample	NRESs as percentage of sample size (%)	NRESs as percentage of ESs (%)	Elite sampling defuzzified value	Elite sampling method error	Elite sampling method timing
50	0.001	50	100.00	0.0013	0.8163161308	0.0017470872	0.0355 s
100	0.003	94	94.00	0.0024	0.8164985098	0.0015647082	0.0709 s
250	0.007	216	86.40	0.0056	0.8168936251	0.0011695929	0.178 s
500	0.013	406	81.20	0.0106	0.8169408395	0.0011223785 ◁	0.355 s
750	0.020	550	73.33	0.0143	0.8168162196	0.0012469984 ◁	0.533 s
1000	0.026	673	67.30	0.0175	0.8170654905	0.0009977275 ◁	0.711 s
5000	0.130	1595	31.90	0.0415	0.8171645726	0.0008986454 ◁	3.59 s
10,000	0.260	2029	20.29	0.0528	0.8173314906	0.0007317274 ◁	7.21 s
50,000	1.302	3026	6.05	0.0788	0.8175959636	**0.0004672544** ◁	36.3 s
100,000	2.604	3439	3.44	0.0896	0.8177743683	**0.0002888497** ◁	1.22 min

Table D.21
α-Planes/CORL and α-planes/interval exhaustive results for the PowderFIS0.05 test set. Exhaustive defuzzified value = 0.8180632180. Error = α-planes value − exhaustive value.

No. of α-planes	α-Planes/CORL defuzzified value	α-Planes/CORL error	α-Planes/CORL timing (s)	α-Planes/interval exhaustive defuzzified value	α-Planes/interval exhaustive error
3	0.8371816462	0.0191184282	0.00148 s	0.8371808680	0.0191176500
5	0.8132243650	−0.0048388530	0.00244 s	0.8132227384	−0.0048404796
9	0.8003904509	−0.0176727671	0.00433 s	0.8003883556	−0.0176748624
11	0.8028981616	−0.0151650564	0.00529 s	0.8028960507	−0.0151671673
21	0.8000431818	−0.0180200362	0.0101 s	0.8000408574	−0.0180223606
51	0.7987563133	−0.0193069047	0.0243 s	0.7987538800	−0.0193093380
101	0.7983826038	−0.0196806142	0.0483 s	0.7983801575	−0.0196830605
1001	0.7974846584	−0.0205785596	0.479 s	0.7974821984	−0.0205810196
10,001	0.7974345629	−0.0206286551	50.5 s	–	–
100,001	0.7974291278	−0.0206340902	2.46 min	–	–

Appendix F. Generalised test set Shopping0.05

Tables F.26, F.27, F.28, F.29.

Table E.22
Exhaustive and VSCTR results for the ShoppingFIS0.1 test set.

Exhaustive defuzzified value	No. of emb. sets	No. of NRESs	Exhaustive timing	VSCTR defuzzified value	VSCTR error	VSCTR timing
0.5954109472	312,500	2495	32.9 s	0.5939161160	0.0014948312	0.000315 s

Table E.23
Sampling results for the ShoppingFIS0.1 test set. Number of embedded sets = 312,500. Percentage of embedded sets sampled $= \frac{\text{sample size}}{\text{number of embedded sets}} \times 100$. Exhaustive defuzzified value = 0.5954109472. The error marked '◁' is lower than the corresponding error for the elite sampling method.

Sample size	Percent. of emb. sets sampled (%)	Sampling defuzzified value	Sampling method error	Sampling method timing
50	0.02	0.5893874958	0.0060234514	0.0218 s
100	0.03	0.5905449544	0.0048659928	0.0434 s
250	0.08	0.5926005506	0.0028103966 ◁	0.108 s
500	0.16	0.5926817464	0.0027292008	0.218 s
750	0.24	0.5923095537	0.0031013935	0.325 s
1000	0.32	0.5934992219	0.0019117253	0.435 s
5000	1.60	0.5931185649	0.0022923823	2.23 s
10,000	3.20	0.5929055726	0.0025053746	4.60 s
50,000	16.00	0.5933037587	0.0021071885	42.4 s
100,000	32.00	0.5933184632	0.0020924840	2.43 min

Table E.24
Elite sampling results for the ShoppingFIS0.1 test set. Number of embedded sets = 312,500. Exhaustive defuzzified value = 0.5954109472. Percentage of embedded sets sampled $= \frac{\text{sample size}}{\text{number of embedded sets}} \times 100$. Errors shown in bold are smaller than the VSCTR error. Underlined errors are lower than the errors for the α-planes method, for all numbers of α-planes. Errors marked '◁' are lower than the corresponding errors for the sampling method.

Sample size	Percentage of ESs sampled (%)	No. of NRESs in sample	NRESs as percentage of sample size (%)	NRESs as percentage of ESs (%)	Elite sampling defuzzified value	Elite sampling method error	Elite sampling method timing (s)
50	0.02	50	100.00	0.016	0.5954284597	**<u>0.0000175125</u>** ◁	0.0243
100	0.03	94	94.00	0.030	0.5935585538	0.0018523934 ◁	0.0485
250	0.08	210	84.00	0.067	0.5911745884	0.0042363588	0.121
500	0.16	368	73.60	0.118	0.5954734998	**<u>0.0000625526</u>** ◁	0.243
750	0.24	481	64.13	0.154	0.5935433933	0.0018675539 ◁	0.366
1000	0.32	570	57.00	0.182	0.5935158606	0.0018950866 ◁	0.486
5000	1.60	1134	22.68	0.363	0.5937003262	0.0017106210 ◁	2.45
10,000	3.20	1401	14.01	0.448	0.5948734695	**<u>0.0005374777</u>** ◁	4.92
50,000	16.00	1943	3.89	0.622	0.5949611026	**<u>0.0004498446</u>** ◁	24.8
100,000	32.00	2146	2.15	0.687	0.5952072004	**<u>0.0002037468</u>** ◁	49.9

Table E.25
α-Planes/CORL and α-planes/interval exhaustive results for the ShoppingFIS0.1 test set. Exhaustive defuzzified value = 0.5954109472. Error = α-planes value − exhaustive value. Error shown in bold is smaller than the VSCTR error.

No. of α-planes	α-Planes/CORL defuzzified value	α-Planes/CORL error	α-Planes/CORL timing (s)	α-Planes/interval exhaustive defuzzified value	α-Planes/interval exhaustive error
3	0.6151869952	0.0197760480	0.000911 s	0.6151852147	0.0197742675
5	0.6018755341	0.0064645869	0.00148 s	0.6018735720	0.0064626248
9	0.5932602572	−0.0021506900	0.00261 s	0.5932572151	−0.0021537321
11	0.5946487587	−**0.0007621885**	0.00322 s	0.5946460014	−0.0007649458
21	0.5929872008	−0.0024237464	0.00608 s	0.5929838018	−0.0024271454
51	0.5920148105	−0.0033961367	0.0146 s	0.5920110566	−0.0033998906
101	0.5919492352	−0.0034617120	0.0289 s	0.5919454769	−0.0034654703
1001	0.5914403564	−0.0039705908	0.288 s	0.5914366015	−0.0039743457
10,001	0.5914134660	−0.0039974812	3.12 s	0.5914097097	−0.0040012375
100,001	0.5914058776	−0.0040050696	2.13 min	0.5914021206	−0.0040088266

Table F.26
Exhaustive and VSCTR results for the ShoppingFIS0.05 test set.

Exhaustive defuzzified value	No. of emb. sets	No. of NRESs	Exhaustive timing	VSCTR defuzzified value	VSCTR error	VSCTR timing
0.1821425020	3,840,000	12,347	11.8 min	0.1814087837	0.0007337183	0.000552 s

Appendix G. Wilcoxon Signed Rank Test results

Tables G.30, G.31, G.32, G.33, G.34, G.35, G.36, G.37, G.38, G.39.

Table F.27
Sampling results for the ShoppingFIS0.05 test set. Number of embedded sets = 3,840,000. Percentage of embedded sets sampled $= \frac{\text{sample size}}{\text{number of embedded sets}} \times 100$. Exhaustive defuzzified value = 0.1821425020. Errors shown in bold are smaller than the VSCTR error. Errors marked '◃' are lower than the corresponding errors for the elite sampling method.

Sample size	Percent. of emb. sets sampled (%)	Sampling defuzzified value	Sampling method error	Sampling method timing
50	0.001	0.1826430434	**0.0005005414** ◃	0.0330 s
100	0.003	0.1820101587	**0.0001323433** ◃	0.0656 s
250	0.007	0.1838659287	0.0017234267	0.164 s
500	0.013	0.1831044751	0.0009619731 ◃	0.329 s
750	0.020	0.1829725179	0.0008300159 ◃	0.492 s
1000	0.026	0.1827985154	**0.0006560134** ◃	0.655 s
5000	0.130	0.1830080344	0.0008655324	3.33 s
10,000	0.260	0.1831606564	0.0010181544	6.79 s
50,000	1.302	0.1830777694	0.0009352674	54.6 s
100,000	2.604	0.1830956217	0.0009531197	2.85 min

Table F.28
Elite sampling results for the ShoppingFIS0.05 test set. Number of embedded sets = 3,840,000. Percentage of embedded sets sampled $= \frac{\text{sample size}}{\text{number of embedded sets}} \times 100$. Exhaustive defuzzified value = 0.1821425020. Errors shown in bold are smaller than the VSCTR error. Errors marked '◃' are lower than the corresponding errors for the sampling method.

Sample size	Percentage of ESs sampled (%)	No. of NRESs in sample	NRESs as percentage of sample size (%)	NRESs as percentage of ESs (%)	Elite sampling defuzzified value	Elite sampling method error	Elite sampling method timing
50	0.001	50	100.00	0.0013	0.1810634451	0.0010790569	0.0360 s
100	0.003	97	97.00	0.0025	0.1835334781	0.0013909761	0.0720 s
250	0.007	241	96.40	0.0063	0.1820180536	**0.0001244484** ◃	0.180 s
500	0.013	455	91.00	0.0118	0.1834734689	0.0013309669	0.360 s
750	0.020	668	89.07	0.0174	0.1830600422	0.0009175402	0.542 s
1000	0.026	819	81.90	0.0213	0.1828033788	**0.0006608768**	0.723 s
5000	0.130	2538	50.76	0.0661	0.1828180409	**0.0006755389** ◃	3.68 s
10,000	0.260	3520	35.20	0.0917	0.1828462673	**0.0007037653** ◃	7.44 s
50,000	1.302	6072	12.14	0.1581	0.1826417325	**0.0004992305** ◃	38.2 s
100,000	2.604	7209	7.21	0.1877	0.1825469839	**0.0004044819** ◃	1.29 min

Table F.29
α-Planes/CORL and α-planes/interval exhaustive results for the ShoppingFIS0.05 test set. Exhaustive defuzzified value = 0.1821425020. Error = α-planes value − exhaustive value.

No. of α-planes	α-Planes/CORL defuzzified value	α-Planes/CORL error	α-Planes/CORL timing (s)	α-Planes/interval exhaustive defuzzified value	α-Planes/interval exhaustive error
3	0.1628183538	−0.0193241482	0.00153 s	0.1628191321	−0.0193233699
5	0.1867756350	0.0046331330	0.00248 s	0.1867772616	0.0046347596
9	0.1996095491	0.0174670471	0.00442 s	0.1996116444	0.0174691424
11	0.1971018384	0.0149593364	0.00542 s	0.1971039493	0.0149614473
21	0.1999568182	0.0178143162	0.0103 s	0.1999591426	0.0178166406
51	0.2012436867	0.0191011847	0.0248 s	0.2012461200	0.0191036180
101	0.2016173962	0.0194748942	0.0496 s	0.2016198425	0.0194773405
1001	0.2025153416	0.0203728396	0.488 s	0.2025178016	0.0203752996
10,001	0.2025654371	0.0204229351	5.13 s	–	–
100,001	0.2025708722	0.0204283702	2.44 min	–	–

Table G.30
Comparing the errors from the four generalised defuzzification methods using the One-Sided Wilcoxon Signed Rank Test with $\alpha = 0.05$ and $n = 6$. The critical value is 2. For the sampling and elite sampling methods, the sample size is 50.

First method	Second method	T	Conclusion
VSCTR	Sampling	7	Cannot reject H_0
VSCTR	Elite sampling	9	Cannot reject H_0
VSCTR	α-Planes/CORL	2	Reject H_0; VSCTR more accurate than AP/CORL
Sampling	Elite sampling	9	Cannot reject H_0
Sampling	α-Planes/CORL	3	Cannot reject H_0
Elite sampling	α-Planes/CORL	2	Reject H_0; ES more accurate than AP/CORL

Table G.31
Comparing the errors from the four generalised defuzzification methods using the One-Sided Wilcoxon Signed Rank Test with $\alpha = 0.05$ and $n = 6$. The critical value is 2. For the sampling and elite sampling methods, the sample size is 100.

First method	Second method	T	Conclusion
VSCTR	Sampling	5	Cannot reject H_0
VSCTR	Elite sampling	7	Cannot reject H_0
VSCTR	α-Planes/CORL	2	Reject H_0; VSCTR more accurate than AP/CORL
Sampling	Elite sampling	10	Cannot reject H_0
Sampling	α-Planes/CORL	3	Cannot reject H_0
Elite sampling	α-Planes/CORL	0	Reject H_0; ES more accurate than AP/CORL

Table G.32
Comparing the errors from the four generalised defuzzification methods using the One-Sided Wilcoxon Signed Rank Test with $\alpha = 0.05$ and $n = 6$. The critical value is 2. For the sampling and elite sampling methods, the sample size is 250.

First method	Second method	T	Conclusion
VSCTR	Sampling	9	Cannot reject H_0
VSCTR	Elite sampling	9	Cannot reject H_0
VSCTR	α-Planes/CORL	2	Reject H_0; VSCTR more accurate than AP/CORL
Sampling	Elite sampling	4	Cannot reject H_0
Sampling	α-Planes/CORL	0	Reject H_0; S more accurate than AP/CORL
Elite sampling	α-Planes/CORL	1	Reject H_0; ES more accurate than AP/CORL

Table G.33
Comparing the errors from the four generalised defuzzification methods using the One-Sided Wilcoxon Signed Rank Test with $\alpha = 0.05$ and $n = 6$. The critical value is 2. For the sampling and elite sampling methods, the sample size is 500.

First method	Second method	T	Conclusion
VSCTR	Sampling	8	Cannot reject H_0
VSCTR	Elite sampling	8	Cannot reject H_0
VSCTR	α-Planes/CORL	2	Reject H_0; VSCTR more accurate than AP/CORL
Sampling	Elite sampling	8	Cannot reject H_0
Sampling	α-Planes/CORL	0	Reject H_0; S more accurate than AP/CORL
Elite sampling	α-Planes/CORL	0	Reject H_0; ES more accurate than AP/CORL

Table G.34
Comparing the errors from the four generalised defuzzification methods using the One-Sided Wilcoxon Signed Rank Test with $\alpha = 0.05$ and $n = 6$. The critical value is 2. For the sampling and elite sampling methods, the sample size is 750.

First method	Second method	T	Conclusion
VSCTR	Sampling	9	Cannot reject H_0
VSCTR	Elite sampling	10	Cannot reject H_0
VSCTR	α-Planes/CORL	2	Reject H_0; VSCTR more accurate than AP/CORL
Sampling	Elite sampling	3	Cannot reject H_0
Sampling	α-Planes/CORL	0	Reject H_0; S more accurate than AP/CORL
Elite sampling	α-Planes/CORL	0	Reject H_0; ES more accurate than AP/CORL

Table G.35
Comparing the errors from the four generalised defuzzification methods using the One-Sided Wilcoxon Signed Rank Test with $\alpha = 0.05$ and $n = 6$. The critical value is 2. For the sampling and elite sampling methods, the sample size is 1000.

First method	Second method	T	Conclusion
VSCTR	Sampling	10	Cannot reject H_0
VSCTR	Elite sampling	10	Cannot reject H_0
VSCTR	α-Planes/CORL	2	Reject H_0; VSCTR more accurate than AP/CORL
Sampling	Elite sampling	4	Cannot reject H_0
Sampling	α-Planes/CORL	0	Reject H_0; S more accurate than AP/CORL
Elite sampling	α-Planes/CORL	0	Reject H_0; ES more accurate than AP/CORL

Table G.36
Comparing the errors from the four generalised defuzzification methods using the One-Sided Wilcoxon Signed Rank Test with $\alpha = 0.05$ and $n = 6$. The critical value is 2. For the sampling and elite sampling methods, the sample size is 5000.

First method	Second method	T	Conclusion
VSCTR	Sampling	9	Cannot reject H_0
VSCTR	Elite sampling	5	Cannot reject H_0
VSCTR	α-Planes/CORL	2	Reject H_0; VSCTR more accurate than AP/CORL
Sampling	Elite sampling	0	Reject H_0; ES more accurate than S
Sampling	α-Planes/CORL	0	Reject H_0; S more accurate than AP/CORL
Elite sampling	α-Planes/CORL	0	Reject H_0; ES more accurate than AP/CORL

Table G.37
Comparing the errors from the four generalised defuzzification methods using the One-Sided Wilcoxon Signed Rank Test with $\alpha = 0.05$ and $n = 6$. The critical value is 2. For the sampling and elite sampling methods, the sample size is 10,000.

First method	Second method	T	Conclusion
VSCTR	Sampling	9	Cannot reject H_0
VSCTR	Elite sampling	2	Reject H_0; ES more accurate than VSCTR
VSCTR	α-Planes/CORL	2	Reject H_0; VSCTR more accurate than AP/CORL
Sampling	Elite sampling	0	Reject H_0; ES more accurate than S
Sampling	α-Planes/CORL	0	Reject H_0; S more accurate than AP/CORL
Elite sampling	α-Planes/CORL	0	Reject H_0; ES more accurate than AP/CORL

Table G.38
Comparing the errors from the four generalised defuzzification methods using the One-Sided Wilcoxon Signed Rank Test with $\alpha = 0.05$ and $n = 6$. The critical value is 2. For the sampling and elite sampling methods, the sample size is 50,000.

First method	Second method	T	Conclusion
VSCTR	Sampling	9	Cannot reject H_0
VSCTR	Elite sampling	0	Reject H_0; ES more accurate than VSCTR
VSCTR	α-Planes/CORL	2	Reject H_0; VSCTR more accurate than AP/CORL
Sampling	Elite sampling	0	Reject H_0; ES more accurate than S
Sampling	α-Planes/CORL	0	Reject H_0; S more accurate than AP/CORL
Elite sampling	α-Planes/CORL	0	Reject H_0; ES more accurate than AP/CORL

Table G.39
Comparing the errors from the four generalised defuzzification methods using the One-Sided Wilcoxon Signed Rank Test with $\alpha = 0.05$ and $n = 6$. The critical value is 2. For the sampling and elite sampling methods, the sample size is 100,000.

First method	Second method	T	Conclusion
VSCTR	Sampling	8	Cannot reject H_0
VSCTR	Elite sampling	0	Reject H_0; ES more accurate than VSCTR
VSCTR	α-Planes/CORL	2	reject H_0; VSCTR more accurate than AP/CORL
Sampling	Elite sampling	0	Reject H_0; ES more accurate than S
Sampling	α-Planes/CORL	0	Reject H_0; S more accurate than AP/CORL
Elite sampling	α-Planes/CORL	0	reject H_0; ES more accurate than AP/CORL

References

[1] Philip A. Birkin, Jonathan M. Garibaldi, A novel dual-surface type-2 controller for micro robots, in: Proceedings of FUZZ-IEEE 2010, Barcelona, Spain, 2010, pp. 359–366.
[2] Oscar Castillo, Patricia Melin, A review on the design and optimization of interval type-2 fuzzy controllers, Applied Soft Computing 12 (2012) 1267–1278.
[3] Oscar Castillo, Patricia Melin, Optimization of type-2 fuzzy systems based on bio-inspired methods: a concise review, Information Sciences 205 (2012) 1–19.
[4] Oscar Castillo, Patricia Melin, Witold Pedrycz, Design of interval type-2 fuzzy models through optimal granularity allocation, Applied Soft Computing 11 (2011) 5590–5601.
[5] Oscar Castillo, Ricardo Martínez-Marroquín, Patricia Melin, Fevrier Valdez, José Soria, Comparative study of bio-inspired algorithms applied to the optimization of type-1 and type-2 fuzzy controllers for an autonomous mobile robot, Information Sciences 192 (2012) 19–38.
[6] Juan R. Castro, Oscar Castillo, Patricia Melin, Antonio Rodríguez-Díaz, A hybrid learning algorithm for a class of interval type-2 fuzzy neural networks, Information Sciences 179 (13) (2009) 2175–2193.
[7] Francisco Chiclana, Shang-Ming Zhou, The type-1 OWA operator and the centroid of type-2 fuzzy sets, in: Proceedings of EUSFLAT-LFA 2011, Aix-les-Bains, France, July 2011, pp. 15–20.
[8] Simon Coupland, Robert I. John, Fuzzy logic and computational geometry, in: Proceedings of RASC 2004, Nottingham, England, December 2004, pp. 3–8.
[9] Simon Coupland, Robert I. John, Geometric type-1 and type-2 fuzzy logic systems, IEEE Transactions on Fuzzy Systems 15 (1) (2007) 3–15.
[10] Sarah Greenfield, Uncertainty, Imprecision and Vagueness, Master's Thesis, De Montfort University, 2005.
[11] Sarah Greenfield, The grid method of discretisation for type-2 fuzzy sets, in: Proc. UKCI 2012, Edinburgh, September 2012.
[12] Sarah Greenfield, Francisco Chiclana, Combining the α-plane representation with an interval defuzzification method, in: Proceedings of EUSFLAT-LFA 2011, Aix-les-Bains, France, July 2011, pp. 920–927.
[13] Sarah Greenfield, Robert I. John, Stratification in the type-reduced set and the generalised Karnik–Mendel iterative procedure, in: Proc. IPMU 2008, Malaga, June 2008, pp. 1282–1289.
[14] Sarah Greenfield, Robert I. John, The uncertainty associated with a type-2 fuzzy set, in: Rudolf Seising (Ed.), Views on Fuzzy Sets and Systems from Different Perspectives, in: Janusz Kacprzyk (Ser. Ed.), Studies in Fuzziness and Soft Computing, vol. 243, Springer-Verlag, 2009, pp. 471–483. http://dx.doi.org/10.1007/978-3-540-93802-6_23.
[15] Sarah Greenfield, Francisco Chiclana, Simon Coupland, Robert I. John, The collapsing method of defuzzification for discretised interval type-2 fuzzy sets, Information Sciences 179 (13) (2009) 2055–2069.
[16] Sarah Greenfield, Francisco Chiclana, Robert I. John, Type-reduction of the discretised interval type-2 fuzzy set, in: Proceedings of FUZZ-IEEE 2009, Jeju Island, Korea, August 2009, pp. 738–743.
[17] Sarah Greenfield, Francisco Chiclana, Robert I. John, The collapsing method: does the direction of collapse affect accuracy? in: Proceedings of IFSA-EUSFLAT 2009, Lisbon, Portugal, July 2009, pp. 980–985.
[18] Sarah Greenfield, Francisco Chiclana, Simon Coupland, Robert I. John, Type-2 defuzzification: two contrasting approaches, in: Proceedings of FUZZ-IEEE 2010, Barcelona, July 2010, pp. 1–7. http://dx.doi.org/10.1109/FUZZY.2010.5584007.
[19] Sarah Greenfield, Francisco Chiclana, Robert I. John, Simon Coupland, The sampling method of defuzzification for type-2 fuzzy sets: experimental evaluation, Information Sciences 189 (2012) 77–92, http://dx.doi.org/10.1016/j.ins.2011.11.042.
[20] Hani Hagras, Christian Wagner, Introduction to interval type-2 fuzzy logic controllers—towards better uncertainty handling in real world applications, IEEE Systems Man, and Cybernetics eNewsletter (27) (2009).
[21] Denisse Hidalgo, Patricia Melin, Oscar Castillo, An optimization method for designing type-2 fuzzy inference systems based on the footprint of uncertainty using genetic algorithms, Expert Systems with Applications 39 (2012) 4590–4598.
[22] Emmanuel A. Jammeh, Martin Fleury, Christian Wagner, Hani Hagras, Mohammed Ghanbari, Interval type-2 fuzzy logic congestion control for video streaming across IP networks, IEEE Transactions on Fuzzy Systems 17 (5) (2009) 1123–1142.
[23] Robert I. John, Perception Modelling Using Type-2 Fuzzy Sets, PhD Thesis, De Montfort University, 2000.
[24] Robert I. John, Simon Coupland, Type-2 fuzzy logic: a historical view, IEEE Computational Intelligence Magazine 2 (1) (2007) 57–62, http://dx.doi.org/10.1109/MCI.2007.357194.
[25] Nilesh N. Karnik, Jerry M. Mendel, Centroid of a type-2 fuzzy set, Information Sciences 132 (2001) 195–220.
[26] Nilesh N. Karnik, Jerry M. Mendel, Qilian Liang, Type-2 fuzzy logic systems, IEEE Transactions on Fuzzy Systems 7 (6) (1999) 643–658.
[27] Erdal Kayacan, Ozkan Cigdem, Okyay Kaynak, On novel training method based on variable structure systems approach for interval type-2 fuzzy neural networks, in: Proceedings of the IEEE Symposium on Advances in Type-2 Fuzzy Logic Systems 2011, Paris, April 2011.
[28] George J. Klir, Bo Yuan, Fuzzy Sets and Fuzzy Logic, Prentice-Hall PTR, 1995.
[29] Cecilia Leal-Ramírez, Oscar Castillo, Patricia Melin, Antonio Rodríguez-Díaz, Simulation of the bird age-structured population growth based on an interval type-2 fuzzy cellular structure, Information Sciences 181 (3) (2011) 519–535.
[30] Werner Van Leekwijck, Etienne E. Kerre, Defuzzification: criteria and classification, Fuzzy Sets and Systems 108 (1999) 159–178, http://dx.doi.org/10.1016/j.fss.2008.06.018.
[31] Leonardo Leottau, Miguel Melgarejo, Implementing an interval type-2 fuzzy processor onto a DSC 56F8013, in: Proceedings of FUZZ-IEEE 2010, Barcelona, Spain, 2010, pp. 1939–1942.
[32] Ondrej Linda, Milos Manic, Interval type-2 fuzzy voter design for fault tolerant systems, Information Sciences 181 (14) (2011) 2933–2950.
[33] Ondrej Linda, Milos Manic, General type-2 fuzzy c-means algorithm for uncertain fuzzy clustering, IEEE Transactions on Fuzzy Systems 20 (5) (2012) 883–897.
[34] Feilong Liu, An efficient centroid type-reduction strategy for general type-2 fuzzy logic system, Information Sciences 178 (9) (2008) 2224–2236.
[35] Luís Alberto Lucas, Tania Mezzadri Centeno, Myriam Regattieri Delgado, General type-2 fuzzy inference systems: analysis, design and computational aspects, in: Proceedings of FUZZ-IEEE 2007, London, 2007, pp. 1743–1747.
[36] Ricardo Martínez, Oscar Castillo, Luis T. Aguilar, Optimization of interval type-2 fuzzy logic controllers for a perturbed autonomous wheeled mobile robot using genetic algorithms, Information Sciences 179 (13) (2009) 2158–2174.
[37] Jerry M. Mendel, Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions, Prentice-Hall PTR, 2001.
[38] Jerry M. Mendel, On answering the question "Where do I start in order to solve a new problem involving interval type-2 fuzzy sets?", Information Sciences 179 (19) (2009) 3418–3431.
[39] Jerry M. Mendel, Robert I. John, Type-2 fuzzy sets made simple, IEEE Transactions on Fuzzy Systems 10 (2) (2002) 117–127.
[40] Jerry M. Mendel, Feilong Liu, Daoyuan Zhai, α-Plane representation for type-2 fuzzy sets: theory and applications, IEEE Transactions on Fuzzy Systems 17 (5) (2009) 1189–1207.
[41] Gerardo M. Méndez, M. de los Angeles Hernandez, Hybrid learning for interval type-2 fuzzy logic systems based on orthogonal least-squares and back-propagation methods, Information Sciences 179 (13) (2009) 2146–2157.
[42] Simon M. Miller, Viara Popova, Robert John, Mario Gongora, An interval type-2 fuzzy distribution network, in: Proc. 2009 IFSA World Congress/EUSFLAT Conference, Lisbon, July 2009, pp. 697–702.
[43] Douglas C. Montgomery, George C. Runger, Applied Statistics and Probability for Engineers, Wiley, 2007.
[44] Maowen Nie, Woei Wan Tan, Towards an efficient type-reduction method for interval type-2 fuzzy logic systems, in: Proceedings of FUZZ-IEEE 2008, Hong Kong, June 2008, pp. 1425–1432.

[45] Turhan Ozen, Jonathan M. Garibaldi, Investigating adaptation in type-2 fuzzy logic systems applied to umbilical acid-base assessment, in: Proceedings of 2003 European Symposium on Intelligent Technologies (EUNITE 2003), Oulu, Finland, July 2003, pp. 289–294.
[46] Janusz T. Starczewski, Efficient triangular type-2 fuzzy logic systems, International Journal of Approximate Reasoning 50 (5) (2009) 799–811.
[47] Christian Wagner, Hani Hagras, Toward general type-2 fuzzy logic systems based on zSlices, IEEE Transactions on Fuzzy Systems 18 (4) (2010) 637–660.
[48] Dongrui Wu, Jerry M. Mendel, Enhanced Karnik–Mendel algorithms, IEEE Transactions on Fuzzy Systems 17 (4) (2009) 923–934.
[49] Dongrui Wu, Maowen Nie, Comparison and practical implementation of type-reduction algorithms for type-2 fuzzy sets and systems, in: Proceedings of FUZZ-IEEE 2011, Taiwan, 2011, pp. 2131–2138.
[50] Hongwei Wu, Jerry M. Mendel, Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems, IEEE Transactions on Fuzzy Systems 10 (5) (2002) 622–639.
[51] Lotfi A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning, Information Sciences 8 (1975) 199–249.
[52] Mina Zaher, Hani Hagras, Amr Khairy, Mohamed Ibrahim, A type-2 fuzzy logic based model for renewable wind energy generation, in: Proceedings of FUZZ-IEEE 2010, Barcelona, Spain, 2010, pp. 511–518.
[53] Daoyuan Zhai, Jerry M. Mendel, Computing the centroid of a general type-2 fuzzy set by means of the centroid-flow algorithm, IEEE Transactions on Fuzzy Systems 19 (3) (2011) 401–422, http://dx.doi.org/10.1109/TFUZZ.2010.2103076.