Type-2 Fuzzy Logic: Circumventing the Defuzzification Bottleneck

Sarah Greenfield, BA, MSc.

Submitted in partial fulfilment of the
requirements for the degree of Doctor of Philosophy
at De Montfort University

May, 2012

Acknowledgements

"*Take delight in the LORD, and he will give you the desires of your heart.*" Psalm 37, verse 4.

I would like to thank my supervisors Dr. Francisco Chiclana, Prof. Robert John and Dr. Simon Coupland for their support and guidance. I would also like to thank my husband Steve and our children Jack, Louise and Gregory for their encouragement and interest.

More specifically I would like to thank

- my elder son Jack Greenfield, for his drawing of the *Defuzzification Bottleneck*,
- Prof. Jerry Mendel and Dr. Peter Innocent for their helpful insights,
- and Dr. Ben Passow for his advice about Matlab^{TM} diagrams.

Sarah Greenfield: Publications Relating to PhD Thesis

International Journals

[1] Sarah Greenfield, Francisco Chiclana, Simon Coupland and Robert I. John, “The Collapsing Method of Defuzzification for Discretised Interval Type-2 Fuzzy Sets” *Information Sciences Special Issue, “High Order Fuzzy Sets: Theory and Applications”*, volume 179, issue 13, pages 2055–2069, June 2009. DOI: 10.1016/j.ins.2008.07.011; ISSN: 0020–0255.

[2] Sarah Greenfield, Francisco Chiclana, Robert I. John and Simon Coupland, “The Sampling Method of Defuzzification for Type-2 Fuzzy Sets: Experimental Evaluation” *Information Sciences*, volume 189, pages 77–92, April 2012. DOI: 10.1016/j.ins.2011.11.042.

[3] Sarah Greenfield and Francisco Chiclana, “Type-Reduction of the Discretised Interval Type-2 Fuzzy Set: Approaching the Continuous Case through Progressively Finer Discretisation”. Accepted in November 2011 for the *Journal of Artificial Intelligence and Soft Computing Research*.

Conferences

[4] Sarah Greenfield, Robert I. John and Simon Coupland, “A Novel Sampling Method for Type-2 Defuzzification”, in *Proc. UKCI 2005*, London, September 2005, pages 120–127.

[5] Sarah Greenfield and Robert I. John, “Optimised Generalised Type-2 Join and Meet Operations”, in *Proc. FUZZ-IEEE 2007*, London, July 2007, pages 141–146.

[6] Sarah Greenfield, Francisco Chiclana, Robert I. John and Simon Coupland, “The Collapsing Method of Defuzzification for Discretised Interval Type-2 Fuzzy Sets”, in *Proc. UKCI 2007*, London, July 2007.

[7] Sarah Greenfield and Robert I. John, “Stratification in the Type-Reduced Set and the Generalised Karnik-Mendel Iterative Procedure”, in *Proc. Intelligent Processing and the Management of Uncertainty 2008 - IPMU 2008*, Malaga, June 2008, pages 1282–1289.

[8] Sarah Greenfield, Francisco Chiclana and Robert I. John, “The Collapsing Method: Does the Direction of Collapse Affect Accuracy?”, in *Proc. IFSA-EUSFLAT 2009*, Lisbon, July 2009, pages 980–985.

[9] Sarah Greenfield, Francisco Chiclana and Robert I. John, “Type-Reduction of the Discretised Interval Type-2 Fuzzy Set”, in *Proc. FUZZ-IEEE 2009*, Jeju Island, Korea, August 2009, pages 738–743.

[10] Sarah Greenfield, Francisco Chiclana, Simon Coupland and Robert I. John, “Type-2 Defuzzification: Two Contrasting Approaches”, in *Proc. FUZZ-IEEE 2010*, Barcelona, July 2010, pages 1–7.

[11] Sarah Greenfield and Francisco Chiclana, “Type-Reduction of the Discretised Interval Type-2 Fuzzy Set: What Happens as Discretisation Becomes Finer?”, in *Proc. IEEE Symposium on Advances in Type-2 Fuzzy Logic Systems*, Paris, April 2011, pages 102–109.

[12] Sarah Greenfield and Francisco Chiclana, “Combining the α-Plane Representation with an Interval Defuzzification Method”, in *Proc. EUSFLAT-LFA*, Aix-Les-Bains, France, July 2011, pages 920-927.

Book Chapter

[13] Sarah Greenfield and Robert I. John (2009), “The Uncertainty Associated with a Type-2 Fuzzy Set”, chapter in *Views on Fuzzy Sets and Systems from Different Perspectives*, edited Rudolf Seising, in ‘Studies in Fuzziness and Soft Computing”, series editor Janusz Kacprzyk, Springer Verlag, pages 471–483. DOI: 10.1007/978-3-540-93802-6_23; ISSN: 1434–9922; ISBN: 978-3-540-93801-9.

Abstract

Type-2 fuzzy inferencing for generalised, discretised type-2 fuzzy sets has been impeded by the computational complexity of the defuzzification stage of the fuzzy inferencing system. Indeed this stage is so complex computationally that it has come to be known as the *defuzzification bottleneck*. The computational complexity derives from the enormous number of embedded sets that have to be individually processed in order to effect defuzzification.

Two new approaches to type-2 defuzzification are presented, the *sampling method* and the *Greenfield-Chiclana Collapsing Defuzzifier*. The sampling method and its variant, elite sampling, are techniques for the defuzzification of generalised type-2 fuzzy sets. In these methods a relatively small sample of the totality of embedded sets is randomly selected and processed. The small sample size drastically reduces the computational complexity of the defuzzification process, so that it may be speedily accomplished.

The Greenfield-Chiclana Collapsing Defuzzifier relies upon the concept of the *representative embedded set*, which is an embedded set having the same defuzzified value as the type-2 fuzzy set that is to be defuzzified. By a process termed *collapsing* the type-2 fuzzy set is converted into a type-1 fuzzy set which, as an approximation to the representative embedded set, is known as the *representative embedded set approximation*. This type-1 fuzzy set is easily defuzzified to give the defuzzified value of the original type-2 fuzzy set. By this method the computational complexity of type-2 defuzzification is reduced enormously, since the representative embedded set approximation replaces the entire collection of embedded sets. The strategy was conceived as a generalised method, but so far only the interval version has been derived mathematically.

The *grid method of discretisation* for type-2 fuzzy sets is also introduced in this thesis.

Work on the defuzzification of type-2 fuzzy sets began around the turn of the millennium. Since that time a number of investigators have contributed methods in this area. These different approaches are surveyed, and the major methods implemented in code prior to their experimental evaluation. In these comparative experiments the grid method of defuzzification is employed. The experimental results show beyond doubt that the collapsing method performs the best of the interval alternatives. However, though the sampling method performs well experimentally, the results do not demonstrate it to be the best performing generalised technique.

Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

CORL Collapsing Outward Right-Left

EIASC Enhanced Iterative Algorithm with Stop Condition

FIS Fuzzy Inferencing System

FOU Footprint Of Uncertainty

GCCD Greenfield-Chiclana Collapsing Defuzzifier

NTS Nie-Tan Set

KMIP Karnik-Mendel Iterative Procedure

NRES Non-Redundant Embedded Set

OWA Ordered Weighted Averaging

RES Representative Embedded Set

RESA Representative Embedded Set Approximation

RS Representative Set

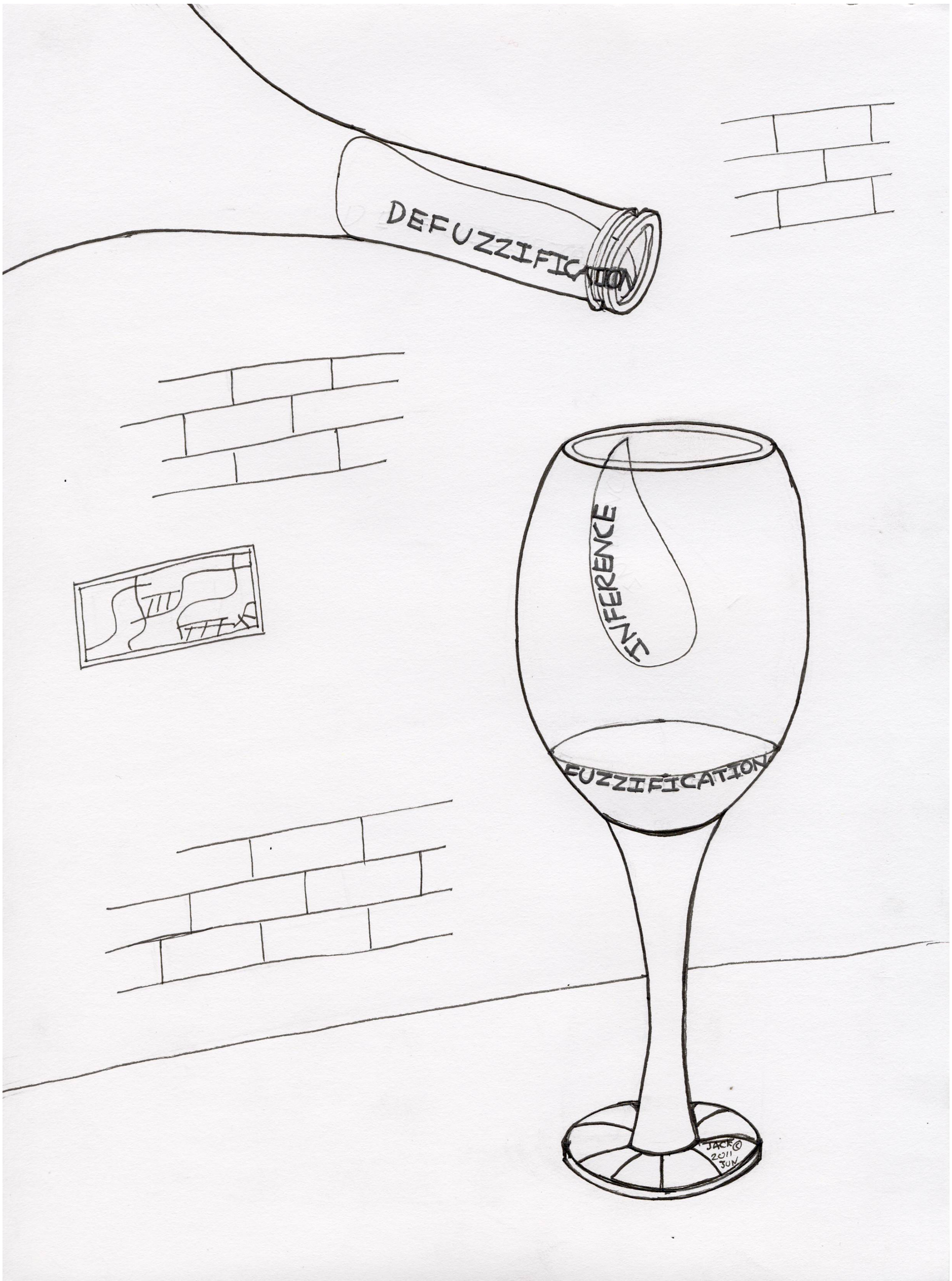
SCSL Solitary Collapsed Slice Lemma

TRS Type-Reduced Set

VSCTR Vertical Slice Centroid Type-Reduction

Part I

INTRODUCTION AND BACKGROUND


DEFUZZIFICATION
INFERENCE
FUZZIFICATION
JACK© 2011 JUN

Chapter 1

Introduction

The work reported in this thesis addresses the challenge of the **efficient and accurate defuzzification of discretised type-2 fuzzy sets**. Defuzzification is the crucial final stage of a *Fuzzy Inferencing System (FIS)*. Owing to its enormous computational complexity, the defuzzification stage of a *type-2* FIS has come to be regarded as a bottleneck [31]. The progress of generalised type-2 applications has been impeded as developers have opted for the computationally simpler interval type-2 FISs [44] for which an increasing number of applications are being developed [2, 25, 29, 34, 39, 46, 56]. This thesis presents novel strategies for circumventing the defuzzification bottleneck.

This chapter introduces fuzzy set theory (Section 1.1), focussing in particular on the type-2 fuzzy set. The relationship between fuzzy set theory and fuzzy logic is briefly discussed, leading on to a description of the FIS (Subsection 1.2.1). There follows (Section 1.3) a discussion about discretisation. The research hypothesis is then set out (Section 1.4), and the chapter closes (Section 1.5) with an outline of the structure of the remainder of the thesis.

1.1 Fuzzy Set Theory

Fuzzy set theory was originated by Lotfi Zadeh [52] in the 1960s. As far back as 1937, however, Max Black had proposed a concept similar to fuzzy sets, which he termed *vague sets* [3]. An extension of classical set theory, in which an object either satisfies or fails to satisfy a specific description, fuzzy set theory is concerned with *the extent to which* an object satisfies a description. A fuzzy set is a set that does not have sharp boundaries. Unlike classical set theory, fuzzy set theory can be seen to reflect real-life in allowing for *degree of truth*. Truth-values form a continuum on a scale from 0 to 1, with 0 representing *false*, and 1 representing *true*. Every fuzzy set is associated with a *membership function*; it is through its membership function that a fuzzy set is defined. The membership function maps each element of the domain onto its *degree of membership*, i.e. its truth-value. Figure 1.1 shows a possible graphical representation of the fuzzy set *tall*. Beyond the height of 6′ the S-curve membership function flattens out, reflecting the common perception that a person of height 6′ or over is definitely tall.

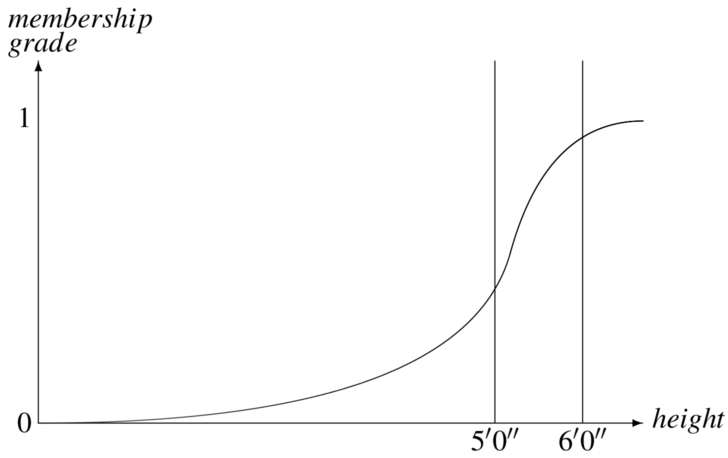


Fig. 1.1. Membership function for the fuzzy set *tall*.

1.1.1 The Type-2 Fuzzy Set

The ‘ordinary’ fuzzy sets discussed above are known as type-1 fuzzy sets. Type-1 membership functions are subject to uncertainty arising from various sources [44]. Their accuracy is therefore questionable; it seems counterintuitive to use real numbers, possibly expressed to several decimal places, to represent degrees of membership. Klir and Folger comment:

> “... it may seem problematical, if not paradoxical, that a representation of fuzziness is made using membership grades that are themselves precise real numbers. Although this does not pose a serious problem for many applications, it is nevertheless possible to extend the concept of the fuzzy set to allow the distinction between grades of membership to become blurred. Sets described in this way are known as *type 2 fuzzy sets*.” [35, page 12]

Here Klir and Folger describe blurring a type-1 fuzzy set to form an *interval type-2 fuzzy set* (Subsection 1.1.2). Mendel and John take this idea a stage further [44, page 118], describing the transition from a type-1 fuzzy set to a *generalised type-2 fuzzy set* (Subsection 1.1.2), again by blurring the type-1 membership function:

> “Imagine blurring the type-1 membership function [...] by shifting the points [...] either to the left or the right, and not necessarily by the same amounts, [...]. Then, at a specific value of x, say x', there no longer is a single value for the membership function (u'); instead the membership function takes on values wherever the vertical

> line $[x = x']$ intersects the blur. These values need not all be weighted the same; hence, we can assign an amplitude distribution to all of these points. Doing this for all $x \in X$, we create a three-dimensional membership function — a type-2 membership function — that characterizes a type-2 fuzzy set."

The difference between interval and general type-2 fuzzy sets is in the secondary membership grades: In the interval case they are 1 throughout, whereas in the generalised case they may take any value from 0 to 1. Thus the interval type-2 fuzzy set is a special case of the generalised type-2 fuzzy set. Type-2 fuzzy sets have elements whose membership grades are themselves fuzzy sets (of type-1). It follows that the graph of a type-2 fuzzy set is 3-dimensional (Figure 1.2(a)).

1.1.2 Type-2 Fuzzy Set: Definitions

Let X be a universe of discourse. A type-1 fuzzy set A on X is characterised by a membership function $\mu_A : X \to [0,1]$ and can be expressed as follows [52]:

$$A = \{(x, \mu_A(x)) |\ \mu_A(x) \in [0,1]\ \forall x \in X\}. \tag{1.1}$$

Note that the membership grades of A are crisp numbers. In the following we will use the notation $U = [0,1]$.

Let $\tilde{P}(U)$ be the set of fuzzy sets in U. A type-2 fuzzy set $\tilde{A}$ in X is a fuzzy set whose membership grades are themselves fuzzy [53–55]. This implies that $\mu_{\tilde{A}}(x)$ is a fuzzy set in U for all x, i.e. $\mu_{\tilde{A}} : X \to \tilde{P}(U)$ and

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) |\ \mu_{\tilde{A}}(x) \in \tilde{P}(U)\ \forall x \in X\}. \tag{1.2}$$

It follows that $\forall x \in X\ \exists J_x \subseteq U$ such that $\mu_{\tilde{A}}(x) : J_x \to U$. Applying (1.1), we have:

$$\mu_{\tilde{A}}(x) = \{(u, \mu_{\tilde{A}}(x)(u)) |\ \mu_{\tilde{A}}(x)(u) \in U\ \forall u \in J_x \subseteq U\}. \tag{1.3}$$

X is called the primary domain and J_x the primary membership of x while U is known as the secondary domain and $\mu_{\tilde{A}}(x)$ the secondary membership of x.

Putting (1.2) and (1.3) together we obtain

$$\tilde{A} = \{(x, (u, \mu_{\tilde{A}}(x)(u))) |\ \mu_{\tilde{A}}(x)(u) \in U,\ \forall x \in X \wedge \forall u \in J_x \subseteq U\}. \tag{1.4}$$

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 (Definition 2.1), which is fundamental to the definition of the *centroid* of a type-2 fuzzy set (Definition 2.2). Alternative notations may be found in [1].

Definition 1.1 (Interval Type-2 Fuzzy Set)**.** *An* interval type-2 fuzzy set *is a type-2 fuzzy set whose secondary membership grades are all* 1.

In the interval case, Equation 1.4 reduces to:

$$\tilde{A} = \{(x,(u,1))|\, \mu_{\tilde{A}}(x)(u) \in U, \forall x \in X \wedge \forall u \in J_x \subseteq U\}. \tag{1.5}$$

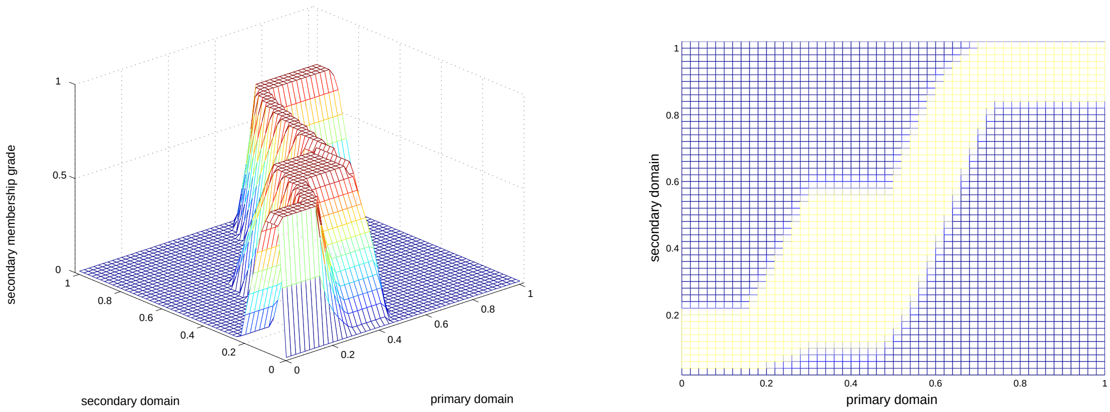
Definition 1.2 (Footprint Of Uncertainty)**.** *The* Footprint Of Uncertainty (FOU) *is the projection of the type-2 fuzzy set onto the $x-u$ plane.*

Definition 1.3 (Lower Membership Function)**.** *The* lower membership function *of a type-2 fuzzy set is the type-1 membership function associated with the lower bound of the FOU.*

Definition 1.4 (Upper Membership Function)**.** *The* upper membership function *of a type-2 fuzzy set is the type-1 membership function associated with the upper bound of the FOU.*

Definition 1.5 (Vertical Slice)**.** *A* vertical slice *is a plane which intersects the x-axis (primary domain) and is parallel to the u-axis (secondary domain).*

Figure 1.2 shows a type-2 fuzzy set (from a Matlab^{TM} application), together with its FOU. Figure 1.3 depicts an FOU from another type-2 fuzzy set, showing two vertical slices at x_1 and x_2.



(a) 3-D representation (b) FOU

Fig. 1.2. Aggregated type-2 fuzzy set created during the inference stage of a Fuzzy Inferencing System.

Figures 1.4 and 1.5 show triangular secondary membership functions at the vertical slices x_1 and x_2; Figures 1.6 and 1.7 show rectangular secondary membership functions[1] at the vertical slices x_1 and x_2.

[1]The rectangular secondary membership function (as used in interval type-2 fuzzy sets) is a special case of the trapezoidal secondary membership function.

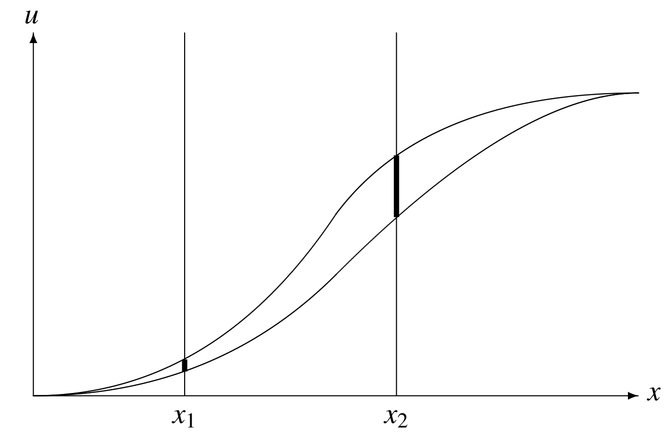


Fig. 1.3. Footprint of uncertainty with two vertical slices at x_1 and x_2. J_{x_1} and J_{x_2} are represented by the bold sections of the corresponding vertical slices.

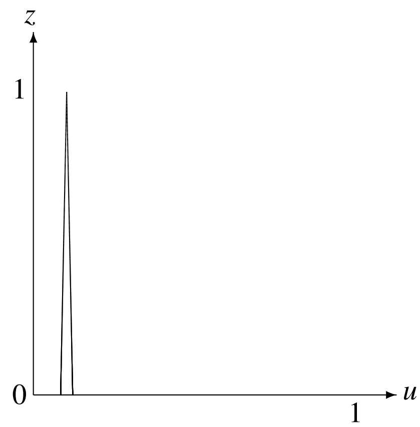


Fig. 1.4. Triangular secondary membership function at the vertical slice x_1 (Figure 1.3).

1.1.3 Operations on Fuzzy Sets

Operations on fuzzy sets are developed out of the corresponding operations on crisp sets (Appendix A), via the *t-norm* (triangular norm) and *t-conorm* (triangular conorm) operators [36]. T-norms are used for the intersection (or conjunction) operation, and t-conorms for the union (or disjunction) operation. These operators have arguments and values ranging between 0 and 1 inclusive. There are several alternatives for both t-norms and t-conorms.

T-norms, symbolized here by T, satisfy the following conditions:

1. $T(a,b) = T(b,a)$,
2. $T(T(a,b),c) = T(a,T(b,c))$,
3. if $a \leq b$ and $c \leq d$ then $T(a,c) \leq T(b,d)$, and

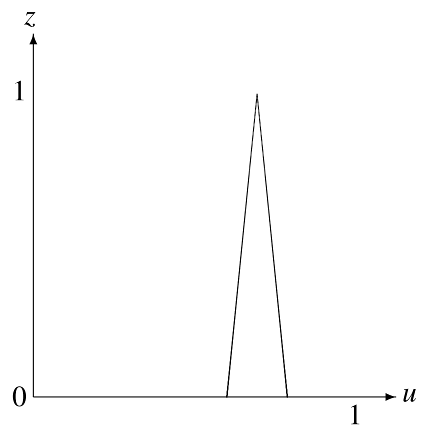


Fig. 1.5. Triangular secondary membership function of the vertical slice x_2 (Figure 1.3).

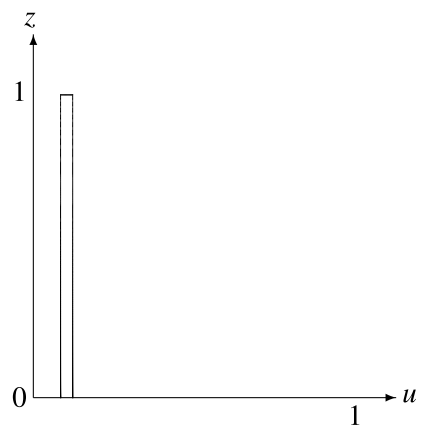


Fig. 1.6. Rectangular secondary membership function of the vertical slice x_1 (Figure 1.3). This is a membership function typical of an interval type-2 fuzzy set.

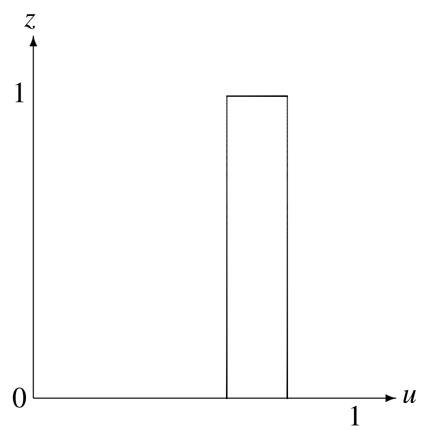


Fig. 1.7. Rectangular secondary membership function of the vertical slice x_2 (Figure 1.3). This is a membership function typical of an interval type-2 fuzzy set.

4. $T(a,1) = a$.

where a, b, c and d are real numbers in the interval [0, 1].
T-conorms, represented by S, also satisfy four conditions.

1. $S(a,b) = S(b,a)$,
2. $S(S(a,b),c) = S(a,S(b,c))$,
3. if $a \leq b$ and $c \leq d$ then $S(a,c) \leq S(b,d)$, and
4. $S(a,0) = a$.

where a, b, c and d are real numbers in the interval [0, 1].
The first three conditions are shared by both t-norms and t-conorms, but in relation to their fourth condition they differ [26].

For type-2 fuzzy sets the terms *join* and *meet* are used for operations corresponding to union and intersection respectively. These operations are defined by applying the Extension Principle (Section 2.1) to the t-norm and t-conorm operators. In this thesis the maximum t-conorm is used for union, and the minimum t-norm for intersection of type-1 fuzzy sets; this reflects common practice as evidenced by the literature.

1.2 Fuzzy Logic

Logic is concerned with propositions. The relationship between conventional set theory and conventional logic is such that the set-theoretic statement "Leicester $\in \{\text{cities}\}$" may be translated into the proposition "Leicester is a city.". There is a similar relationship between fuzzy sets and fuzzy propositions. Fuzzy logic is the calculus of fuzzy propositions. Under the usual formulation the set of real numbers between 0 and 1 inclusive represents the degrees of truth of fuzzy propositions, with 1 denoting absolute truth, and 0 absolute falsity. Thus fuzzy logic is a form of many-valued logic. The connectives used have to operate classically on the extremes of the interval. The fuzzy truth t of complex statements can be evaluated by employing a standard trio of rules[2]:

$$t(A \wedge B) = min(t(A), t(B));$$

$$t(A \vee B) = max(t(A), t(B));$$

$$t(\neg A) = 1 - t(A).$$

[2]Alternative versions of the first two rules are possible, corresponding to the various t-norms and t-conorms [11, page 216].

Hence fuzzy logic is a truth-functional system. An example of fuzzy inferencing might be: Suppose $t(Tall(Peter)) = 0.85$, and also $t(Old(Peter)) = 0.30$. From this it follows that

$$t(Tall(Peter) \wedge Old(Peter)) = 0.30.$$

This statement seems intuitively correct, but fuzzy logic can give rise to some unexpected results. To extend the example, it is quite simple to derive:

$$t(Tall(Peter) \wedge \neg Tall(Peter)) = 0.15.$$

This result reflects the fact that fuzzy set theory does not obey the law of excluded middle (Appendix A) [37].

1.2.1 Fuzzy Inferencing Systems

Practically, fuzzy logic is reliant on the computer and is implemented through an FIS which works by applying fuzzy logic operators to common-sense linguistic rules. An FIS may be of any type; its type is determined by the highest type of the fuzzy sets it employs[3]. Type-2 FISs fall into two categories: 1. The Mamdani style in which the output membership function is a type-2 fuzzy set (requiring defuzzification), and 2. The Takagi-Sugeno-Kang style for which the output membership functions are either linear or constant; defuzzification is superfluous as the outputs are aggregated via a simple weighted sum. This thesis is solely concerned with the Mamdani style type-2 FIS.

Starting with a crisp number, a Mamdani FIS (of any type) passes through three stages: fuzzification, inferencing, and finally defuzzification:

Fuzzification is the process by which the degree of membership of a fuzzy set is determined, based on the crisp input value and the membership function of the fuzzy set.

Inferencing is the main stage of the FIS and may be broken down into three further stages:

1. antecedent computation,
2. implication, and
3. aggregation.

The output of inferencing is a fuzzy set known as the *aggregated set*.

Defuzzification During this stage this fuzzy set is converted into another crisp number, the final result of the processing of the FIS.

Figure 1.8 provides a representation of a Mamdani-style type-2 FIS, showing the defuzzification stage as consisting of two parts, *type-reduction* and defuzzification proper. Type-reduction is

[3] It follows that a type-2 FIS may make use of type-1 fuzzy sets.

the procedure by which a type-2 fuzzy set is converted to a type-1 fuzzy set known as the *Type-Reduced Set (TRS)*. This set is then defuzzified to give a crisp number. The additional stage of type-reduction distinguishes the type-2 FIS from its type-1 counterpart and has been a processing bottleneck in type-2 fuzzy inferencing [7, 13, 17, 31] because it relies on finding the centroids of an extraordinarily large number of type-1 fuzzy sets (embedded sets) into which the type-2 fuzzy set is decomposed. The research hypothesis of this thesis (Section 1.4) addresses the problem of the defuzzification bottleneck.

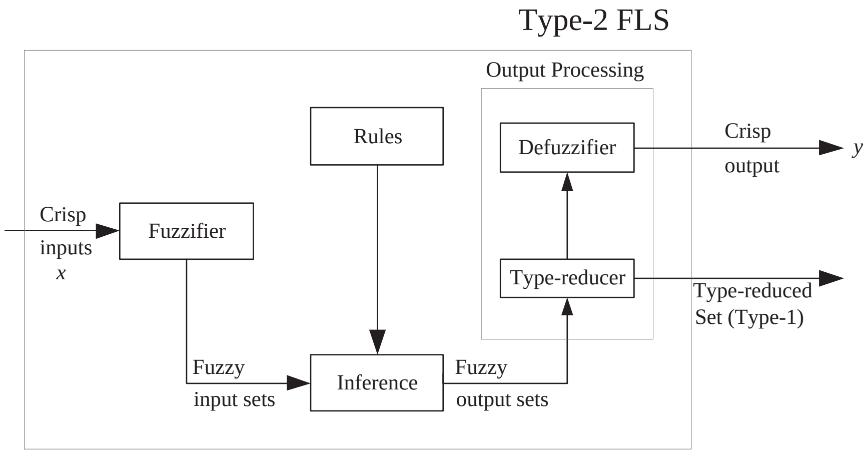


Fig. 1.8. Type-2 FIS (from Mendel [43]).

1.3 Discretisation

With no loss of generality it is assumed that the type-2 fuzzy set is contained within a unit cube and may be viewed as a surface represented by (x, u, z) co-ordinates[4]. 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 1.6 (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.*

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

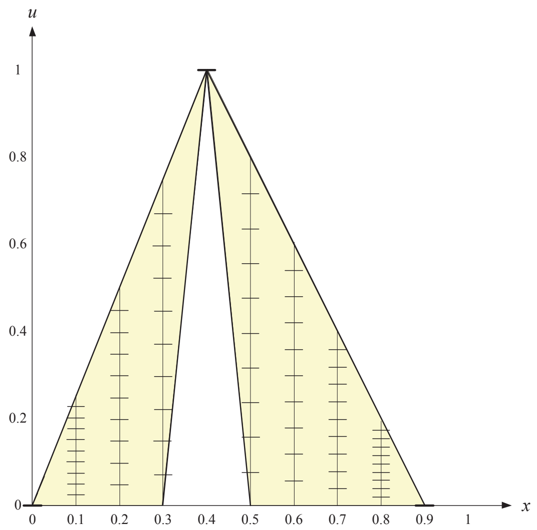


Fig. 1.9. $x-u$ plane under the standard method of discretisation. Primary domain degree of discretisation $= 0.1$.

Definition 1.7 (Vertical Slice [44])**.** *A* vertical slice *of a type-2 fuzzy set is a plane through the x-axis, parallel to the $u-z$ plane.*

Definition 1.8 (Degree of Discretisation)**.** *The* degree of discretisation *is the separation of the slices.*

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:

1.3.1 Standard Method of Discretisation

In this discretisation technique (Figure 1.9) the primary domain of the type-2 fuzzy set is sliced vertically at even intervals. Each of the slices generated intersects the FOU; each line of intersection (within the FOU) is itself sliced at even intervals parallel to the $x-z$ plane. This results in different secondary domain degrees of discretisation according to the vertical slice [32]. The primary degree of discretisation and the number of horizontal slices are arbitrary, context dependent parameters, chosen by the developer after considering factors such as the power of the computer.

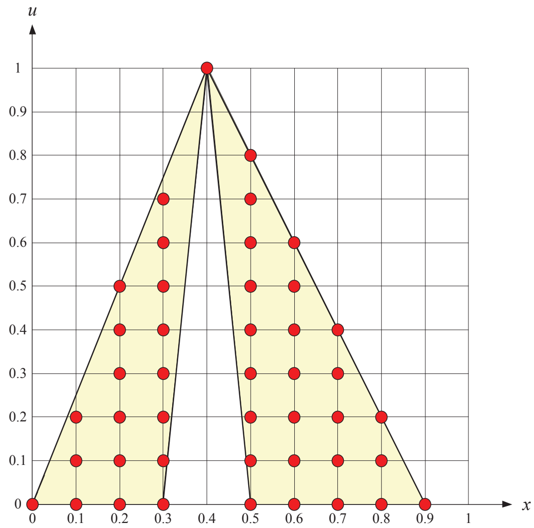


Fig. 1.10. $x-u$ plane under the grid method of discretisation. Primary domain degree of discretisation $=0.1$; secondary domain degree of discretisation $=0.1$.

1.3.2 Grid Method of Discretisation

An original, alternative method valid for all type-2 fuzzy sets is the *grid method of discretisation* (Figure 1.10). In this approach the $x-u$ plane, $[0,1]^2$, is evenly divided into a rectangular grid, as determined by the degrees of discretisation of the x and u-axes. The fuzzy set surface, consisting of the secondary membership grades corresponding to each grid point (x,u) in the FOU, may be represented by a matrix of the secondary grades, in which the x and u co-ordinates are implied by the secondary grade's position within the matrix [16].

1.3.3 Critique of the Standard and Grid Methods

The grid method has certain advantages over the standard method:

1. Conceptually, the grid approach is very straightforward and easy to understand.

2. The grid approach confers a data structure on the type-2 fuzzy set. The set is represented by a rectangular matrix, which encapsulates the surface of the set. Therefore the set does not need to be constructed from its membership functions as with the standard method.

3. The grid method is more general than the standard method in that it is applied to the whole theoretical domains X and U[5], i.e. it is not necessary to identify the actual domains of the

[5] X is normally equal to U.

membership function $\mu_{\tilde{A}}$ and the secondary membership functions $\mu_{\tilde{A}}(x)$ prior to discretisation.

4. Processing is simpler using the grid method as opposed to the standard method.

5. By employing the grid method, join and meet operations may be optimised [22].

However, a drawback of the grid method is that if the FOU has a narrow section, the discretisation has to be made finer in order to represent the type-2 fuzzy set adequately.

Owing to the first four advantages, the grid method was adhered to in the type-2 fuzzy sets prepared for the testing regime described in Chapter 5.

1.4 Research Hypothesis

The research hypothesis addressed in this thesis can be stated as:

> **The development of discretised, generalised type-2 fuzzy inferencing systems has been impeded by computational complexity, particularly in relation to defuzzification. The development of alternative defuzzification algorithms will resolve this defuzzification bottleneck.**

We regard the technique of *exhaustive defuzzification* (Chapter 2) as the ultimate standard of accuracy. A faithful implementation of the exhaustive type-reduction algorithm (Algorithm 2.1) requires that every embedded set be processed. The number of embedded sets within a type-2 fuzzy set is

$$\prod_{i=1}^{N} M_i,$$

where N is the number of vertical slices into which the type-2 fuzzy set has been discretised, and M_i is the number of elements on the i^{th} slice. On any method of discretisation, the finer the discretisation, the better the representation of a given type-2 fuzzy set, but the greater the number of embedded sets generated. A reasonably fine discretisation can give rise to astronomical numbers of embedded sets. For instance, in one example in which a type-2 FIS was employed, discretised under the grid method with primary and secondary degrees of discretisation of 0.02, the number of embedded sets generated was of the order of 2.9×10^{63}.

1.5 Structure of the Thesis

This thesis comprises three parts.

Part I is introductory and apart from this chapter contains Chapter 2, which is a survey of the available approaches to type-2 defuzzification.

Part II consists of two chapters setting out the original contributions to the topic of type-2 defuzzification: Chapter 3 concerns the sampling method and Chapter 4 the collapsing method.

Part III contains two chapters: Chapter 5 evaluates the various defuzzification techniques and includes a practical comparison of the methods with respect to speed and accuracy. Chapter 6 concludes the thesis.

The thesis is supplemented with appendices containing definitions relating to type-1 fuzzy sets, graphs of type-2 fuzzy test sets, and results of the experiments described in Chapter 5 of Part III.

Chapter 2

A Survey of Defuzzification Techniques for Type-2 Fuzzy Sets

2.1 Introduction

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 [38]. Type-2 defuzzification is a process that usually consists of two stages [43]:

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* [53], 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[1]. The research presented in this thesis makes use solely of the centroid method of type-1 defuzzification (Appendix B).

This chapter is a survey of the published approaches to type-2 defuzzification.

2.2 The Wavy-Slice Representation Theorem

The concept of an *embedded type-2 fuzzy set (embedded set)* or *wavy-slice* [44] 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, $\mu_{\tilde{A}}(x)(u)$.

[1]Geometric defuzzification [8] is exceptional among type-2 defuzzification methods in not involving type-reduction and therefore not requiring type-1 defuzzification.

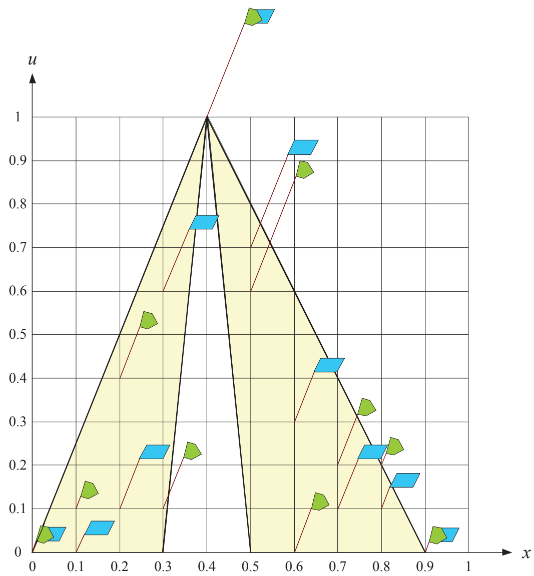


Fig. 2.1. Two embedded sets, indicated by different flag styles. The flag height reflects the secondary membership grade. Degree of discretisation of primary and secondary domains is 0.1. The shaded region is the FOU.

Example 1. *In Figure 2.1 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 $\tilde{P}$ is represented by pentagonal, pointed flags, and embedded set $\tilde{Q}$ is symbolised by quadrilateral shaped flags.*

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

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

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

Definition 2.1 (Embedded Set)**.** *Let $\tilde{A}$ be a type-2 fuzzy set in X. For discrete universes of discourse X and U,* an embedded type-2 set *$\tilde{A}_e$ of $\tilde{A}$ is defined as the following type-2 fuzzy set*

$$\tilde{A}_e = \{(x_i, (u_i, \mu_{\tilde{A}}(x_i)(u_i))) | \, \forall i \in \{1, \ldots, N\} : \, x_i \in X \;\, u_i \in J_{x_i} \subseteq U\}. \tag{2.1}$$

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

Mendel and John have shown that a type-2 fuzzy set can be represented as the union of its type-2 embedded sets [44, page 121]. This powerful result is known as the type-2 fuzzy set *Representation Theorem* or *Wavy-Slice Representation Theorem*; in [44] 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 [44].

The Representation Theorem is formally stated thus [44, page 121]:

> Let $\tilde{A}_e^j$ denote the jth type-2 embedded set for type-2 fuzzy set $\tilde{A}$, i.e.,
>
> $$\tilde{A}_e^j \equiv \left\{ \left(u_i^j, \mu_{\tilde{A}}(x_i)(u_i^j) \right), i = 1, \ldots, N \right\}$$
>
> where $\{u_i^j, \ldots, u_N^j\} \in J_{x_i}$. Then $\tilde{A}$ can be represented as the union of its type-2 embedded sets, i.e.,
>
> $$\tilde{A} = \sum_{j=1}^{n} \tilde{A}_e^j$$
>
> where
>
> $$n \equiv \prod_{i=1}^{N} M_i.$$

2.3 Generalised Type-2 Fuzzy Sets

This section is concerned with the defuzzification of generalised type-2 fuzzy sets. The first stage of type-2 defuzzification is to create the 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 [53] (Section 2.1). Alternatively the TRS may be defined via the Representation Theorem [44, page 121].

Definition 2.2. *The TRS associated with a type-2 fuzzy set $\tilde{A}$ with primary domain X discretised into N points is*

$$C_{\tilde{A}} = \left\{ \left(\frac{\sum_{i=1}^{N} x_i \cdot u_i}{\sum_{i=1}^{N} u_i}, \mu_{\tilde{A}}(x_1)(u_1) * \ldots * \mu_{\tilde{A}}(x_N)(u_N) \right) \middle| \; \forall i \in \{1, \ldots, N\} : \; x_i \in X \;\; u_i \in J_{x_i} \subseteq U \right\}. \quad (2.2)$$

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[2] it is to be avoided. For the work presented in this thesis,

[2] Under the product t-norm, $\lim_{N\to\infty} \left[\mu_{\tilde{A}}(x_1)(u_1) * \ldots * \mu_{\tilde{A}}(x_N)(u_N)\right] = 0$ [33, page 201].

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 2.1 (adapted from Mendel [43]) is used to compute the TRS of a type-2 fuzzy set.

2.3.1 Exhaustive Type-Reduction

Mendel and John's Representation Theorem (Subsection 2.2) provides a precise, straightforward method for type-2 defuzzification. Though Definition 2.2 does not explicitly mention embedded sets, they appear implicitly in Equation 2.2. When this equation is presented in algorithmic form (Algorithm 2.1), explicit mention is made of embedded sets. As *every* embedded set is processed, this stratagem has become known as the *exhaustive method* [19]. 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.

```
   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
```

Algorithm 2.1: Type-reduction of a discretised type-2 fuzzy set to a type-1 fuzzy set, adapted from Mendel [43].

Stage 3 of Algorithm 2.1 requires the calculation of the embedded set's centroid. Example 2 relates to the embedded sets introduced in Example 1.

Example 2. *Embedded set $\tilde{P}$ has minimum secondary grade $z_{\tilde{P}} = 0.1$ and primary domain value* *of its type-1 centroid* $x_{\tilde{P}} = 0.4308$*:*

$$x_{\tilde{P}} = \frac{\sum_{i=1}^{N} x_i \cdot u_i}{\sum_{i=1}^{N} u_i} = \frac{1.12}{2.6} = 0.4308.$$

Similarly embedded set $\tilde{Q}$ *has minimum secondary grade* $z_{\tilde{Q}} = 0.1$ *and primary domain value of its type-1 centroid* $x_{\tilde{Q}} = 0.4414$*:*

$$x_{\tilde{Q}} = \frac{\sum_{i=1}^{N} x_i \cdot u_i}{\sum_{i=1}^{N} u_i} = \frac{1.28}{2.9} = 0.4414.$$

In Section 1.4 we saw the major shortcoming of this method — its computational complexity.

2.3.2 The Sampling Defuzzifier

The sampling method of defuzzification [24] is an efficient, cut-down alternative to exhaustive defuzzification. By processing only a relatively small sample of embedded sets, the computational complexity of type-reduction is drastically reduced. A full exposition of this technique is to be found in Chapter 3.

2.3.3 Vertical Slice Centroid Type-Reduction

Vertical Slice Centroid Type-Reduction (VSCTR) is a highly intuitive method employed by John [30]; the paper of Lucas et al. [42] 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. 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 (Subsection 2.4.4).

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

Algorithm 2.2: VSCTR of a discretised type-2 fuzzy set to a type-1 fuzzy set.

2.3.4 The α-Plane Representation

In 2008 Liu [41, 45] 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 [41] has shown that a generalised type-2 fuzzy set may be type-reduced. This method of type-reduction (Algorithm 2.3) is depicted in Figure 2.2. 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) [41], any interval method may be used. Any variation on the KMIP, such as Enhanced Iterative Algorithm with Stop Condition (EIASC) (Subsection 2.4.1) will locate the endpoints of the TRS interval. Other interval methods, such as the Greenfield-Chiclana Collapsing Defuzzifier (Chapter 4), or the Nie-Tan Method (Subsection 2.4.4), 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.

Input: a discretised generalised type-2 fuzzy set
Output: a discrete type-1 fuzzy set

decompose the type-2 fuzzy set into α-planes ;
forall the α-*planes* **do**
    find the left and right endpoints using the KMIP ;
    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 } ;
end

Algorithm 2.3: Type-reduction of a type-2 fuzzy set to a type-1 fuzzy set using the α-plane method.

Independently to Liu, and at about the same time, Wagner and Hagras introduced the notion of zSlices [48], a concept very similar to α-planes. The α-planes/KMIP method has been modified by Zhai and Mendel [57, 58] to increase its efficiency.

2.3.5 The Stratified Type-Reduced Set

Arising out of research into the sampling defuzzifier (Chapter 3), Greenfield and John observed the stratified structure of the TRS [23]. Figure 2.3 shows a typical TRS derived from a sample of 500 randomly generated embedded sets from a generalised type-2 fuzzy set. The stratification pattern is readily apparent in this figure. This stratified structure is exploited in an extension of the KMIP (Subsection 2.4.1) to generalised type-2 fuzzy sets [23]. The technique employed is to defuzzify each stratum individually, then combine the strata's defuzzified values appropriately

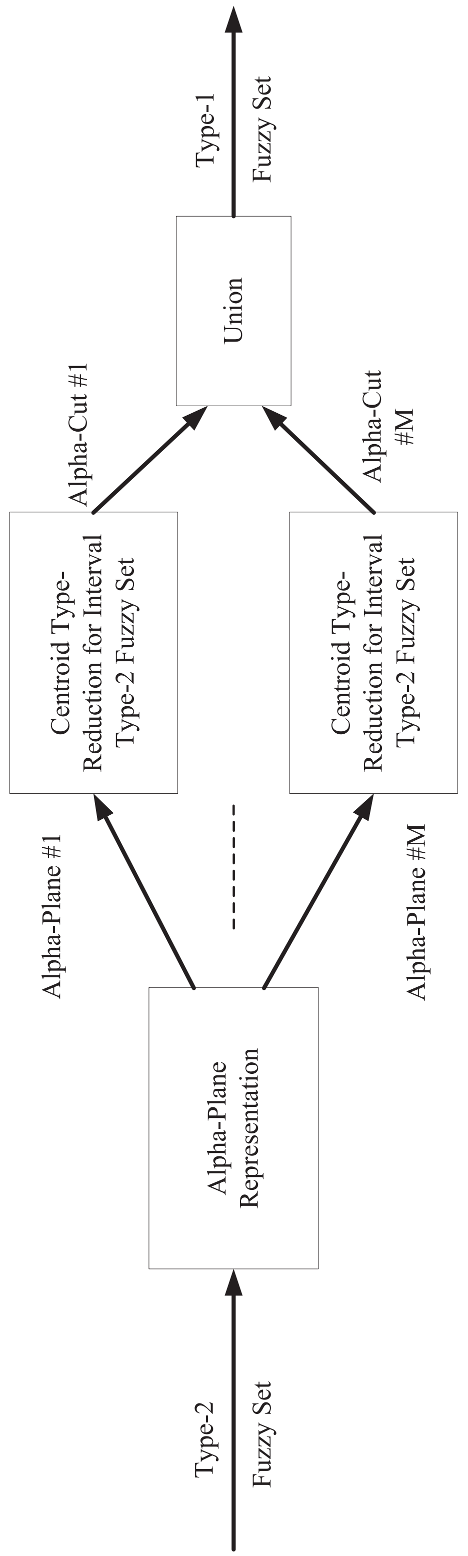


Fig. 2.2. Defuzzification using the α-Planes Representation (from Liu [41]).

to give the defuzzified value of the type-2 fuzzy set. This defuzzification strategy has not been implemented in software.

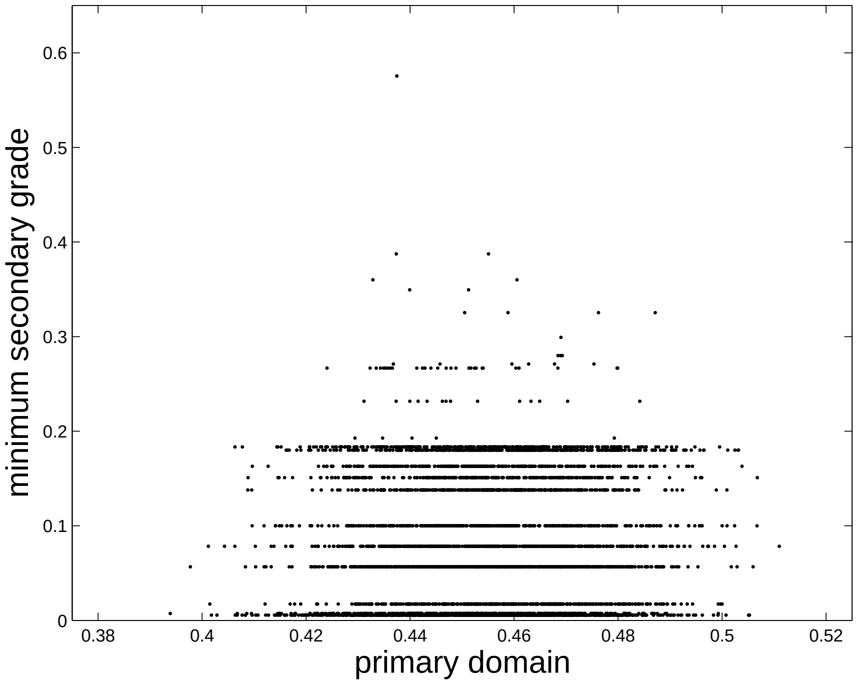


Fig. 2.3. The TRS strata. Each dot represents a TRS tuple. Sample size = 5000.

2.3.6 The Type-1 OWA Based Approach

Chiclana and Zhou have shown that the type-1 fuzzy set derived from a generalised type-2 fuzzy set through the application of a type-1 *Ordered Weighted Averaging (OWA)* operator [59] coincides with the TRS of the type-2 fuzzy set [6] — both are essentially aggregation problems. This identification offers another promising approach to type-reduction. This technique has yet to be implemented in software.

2.4 Interval Type-2 Fuzzy Sets

As the interval type-2 fuzzy set is a special case of the generalised type-2 fuzzy set, the generalised methods of defuzzification described above are all applicable to interval type-2 fuzzy sets. However, this section concerns techniques specifically developed as interval methods.

For the TRS of an interval type-2 fuzzy set, Definition 2.2 reduces to:

Definition 2.3 (TRS of an Interval Type-2 Fuzzy Set)**.** *The TRS associated with an interval type-2 fuzzy set $\tilde{A}$ with primary domain X discretised into N points is*

$$C_{\tilde{A}} = \left\{ \left(\frac{\sum_{i=1}^{N} x_i \cdot u_i}{\sum_{i=1}^{N} u_i}, 1 \right) \middle| \, \forall i \in \{1,\ldots,N\} : \, x_i \in X \;\; u_i \in J_{x_i} \subseteq U \right\}. \tag{2.3}$$

2.4.1 The Karnik-Mendel Iterative Procedure

The most widely adopted method for type-reducing an interval type-2 fuzzy set is the *KMIP* [33]. The result of type-reduction of an interval type-2 fuzzy set is an interval set (which is a particular case of a type-1 fuzzy set)[3], with the defuzzified value of the type-2 fuzzy set located at the midpoint. The iterative procedure is an efficient method for finding the endpoints of the interval. There is an element of approximation in the defuzzified value, as in general the TRS tuples are not symmetrically distributed over the interval[4].

Since the publication of the KMIP, various enhanced versions have been proposed [49]. They differ somewhat in their search strategy. Wu and Nie [50] present five variations, and go on to compare them experimentally in relation to efficiency. They found the optimum algorithm to be the EIASC [50, Section III] (Algorithm 2.4). Wu and Nie's Matlab^{TM} code is to be found in Appendix A of [50].

2.4.2 The Wu-Mendel Approximation

In [51] Wu and Mendel provide a closed form formula for the centroid of a type-2 interval fuzzy set by calculating approximations[5] to the endpoints (or uncertainty bounds) of the type-reduced interval. The algorithm [51, Appendix III, page 635] is set out below (Algorithm 2.5). The parameters of the Wu-Mendel Approximation, as used in the algorithm, are shown diagrammatically in Figure 2.4.

2.4.3 The Greenfield-Chiclana Collapsing Defuzzifier

By means of the iterative *collapsing formula*, an interval type-2 fuzzy set is type-reduced to a type-1 fuzzy set, the *Representative Embedded Set Approximation (RESA)*. This original method is explored fully in Chapter 4.

[3]The endpoints of the interval are termed *uncertainty bounds* as the length of the TRS is regarded as a measure of the uncertainty pertaining to the aggregated set [51, page 622].

[4]As discretisation is made finer the gaps between the tuples decrease, and in the limiting case (degree of discretisation = 0) the tuples form a continuous line. In this case the defuzzified value is located exactly at the midpoint of the interval. However, since the KMIP is a search algorithm, it is not possible to apply it in the continuous case, and therefore it is not guaranteed that the exact centroid will be obtained.

[5]This contrasts with the KMIP, which, in the discretised case, is intended to find the endpoints accurately.

Input: a discretised interval type-2 fuzzy set
Output: the endpoints of the TRS

1 set x_i $i-1,2\ldots,N$ to be the domain values of the vertical slices ;
2 set L_i to be the lower membership grade of J_i ;
3 set U_i to be the upper membership grade of J_i ;
4 {to compute the left endpoint} ;
5 initialise $a = \sum_{i=1}^{N} x_i L_i$;
6 initialise $b = \sum_{i=1}^{N} L_i$;
7 initialise $y_l = x_N$ {left endpoint} ;
8 initialise $l = 0$;
9 calculate $l = l+1$;
10 calculate $a = a + x_l(U_l - L_l)$;
11 calculate $b = b + U_l - L_l$;
12 calculate $c = \frac{a}{b}$;
13 **if** $c > y_l$ **then**
14   set $l = l-1$;
15   stop ;
16 **end**
17 **otherwise**
18   set $y_l = c$;
19   go to Step 9 ;
20 **endsw**
21 {to compute the right endpoint} ;
22 initialise $a = \sum_{i=1}^{N} x_i L_i$;
23 initialise $b = \sum_{i=1}^{N} L_i$;
24 initialise $y_r = x_1$ {right endpoint} ;
25 initialise $l = 0$;
26 calculate $a = a + x_r(U_r - L_r)$;
27 calculate $b = b + U_r - L_r$;
28 calculate $c = \frac{a}{b}$;
29 calculate $r = r-1$;
30 **if** $c < y_r$ **then**
31   set $r = r+1$;
32   stop ;
33 **end**
34 **otherwise**
35   set $y_r = c$;
36   go to Step 26 ;
37 **endsw**

Algorithm 2.4: EIASC.

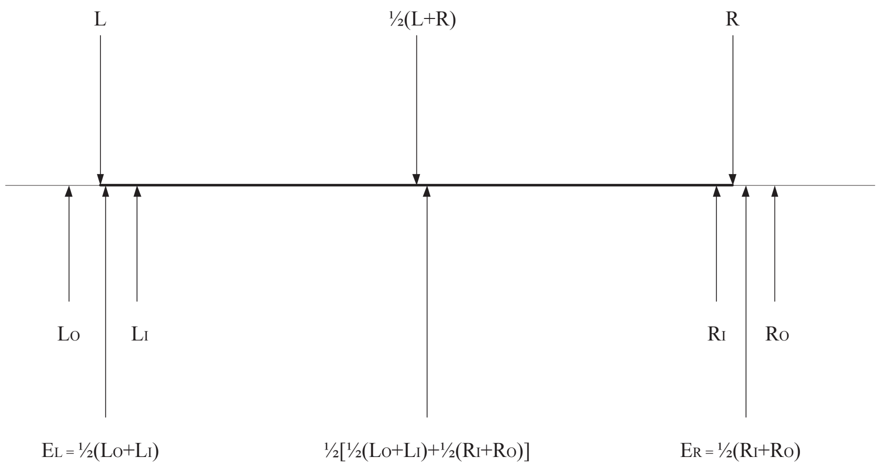


Fig. 2.4. The Wu-Mendel Approximation (adapted from [51]). The KMIP finds the left uncertainty bound L and the right uncertainty bound R. The defuzzified value is taken to be the mean of L and R. The Wu-Mendel Approximation approximates these values to E_L and E_R respectively. E_L is mid way between L_O, the left outer-bound and L_I, the left inner-bound. Similarly E_R is mid way between R_I, the the right inner-bound, and R_O, the right outer-bound. As with the KMIP, the defuzzified value is taken to be the mean of E_L and E_R.

Input: a discretised interval type-2 fuzzy set
Output: approximations to the endpoints of the TRS

set x_i $i-1,2\ldots,N$ to be the domain values of the vertical slices ;
set L_i to be the lower membership grade of J_i ;
set U_i to be the upper membership grade of J_i ;
set L_I to be the left inner-bound ;
set R_I to be the right inner-bound ;
set L_O to be the left outer-bound ;
set R_0 to be the right outer-bound ;
set E_L to be the left endpoint ;
set E_R to be the right endpoint ;
calculate $l = \frac{\sum_i L_i x_i}{\sum_i L_i}$ {defuzzify the lower membership function} ;
calculate $u = \frac{\sum_i U_i x_i}{\sum_i U_i}$ {defuzzify the upper membership function} ;
calculate $L_I = min(l,u)$;
calculate $R_I = max(l,u)$;
calculate $L_O = L_I - \frac{\sum_i (U_i - L_i)}{\sum_i U_i \cdot \sum_i L_i} \cdot \frac{\sum_i L_i x_i \cdot \sum_i U_i(1-x_i)}{\sum_i L_i x_i + \sum_i U_i(1-x_i)}$;
calculate $R_O = R_I + \frac{\sum_i (U_i - L_i)}{\sum_i U_i \cdot \sum_i L_i} \cdot \frac{\sum_i U_i x_i \cdot \sum_i L_i(1-x_i)}{\sum_i U_i x_i + \sum_i L_i(1-x_i)}$;
calculate $E_L = \frac{L_O + L_I}{2}$;
calculate $E_R = \frac{R_O + R_I}{2}$;

Algorithm 2.5: The Wu-Mendel Approximation.

2.4.4 The Nie-Tan Method

Nie and Tan [47] describe an efficient type-reduction method for interval type-2 fuzzy sets, which involves taking the mean of the lower and upper membership functions of the interval set, so creating a type-1 fuzzy set. Symbolically, $\mu_N(x_i) = \frac{1}{2}(\mu_L(x_i) + \mu_U(x_i))$, where N is the resultant type-1 fuzzy set (Algorithm 2.6).

```
Input: a discretised interval type-2 fuzzy set
Output: a discrete type-1 fuzzy set (the Nie-Tan Set)
forall the vertical slices do
    find the mean of the lower and upper membership grades ;
    pair each mean with the domain value of the vertical slice ;
end
```

Algorithm 2.6: The Nie-Tan Method.

2.4.5 The Type-1 OWA Based Approach

The interval counterpart of Chiclana and Zhou's generalised type-1 OWA based approach (Subsection 2.3.6) is that the application of the α-level type-1 OWA leads to the TRS of an interval type-2 fuzzy set [6]. This equivalence underpins a strategy for interval type-reduction.

2.5 Chronology of the Research

Table 2.1 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 both the interval and the generalised cases. Such an evaluation is reported on in Chapter 5.

Summary

This chapter has surveyed the available defuzzification techniques for type-2 fuzzy sets. Several investigators have applied themselves to the problem of the type-2 defuzzification bottleneck; a variety of strategies have been proposed as solutions. The next chapter (Part II, Chapter 3) presents the novel sampling method of generalised defuzzification.

DATE	AUTHORS	METHOD	REFERENCE	PUBLISHER/PUBLICATION
2001	Jerry M. Mendel	**Exhaustive**	[43]	Prentice-Hall PTR
February 2001	Nilesh N. Karnik Jerry M. Mendel	**KMIP**	[33]	Information Sciences
October 2002	Hongwei Wu Jerry M. Mendel	**Wu-Mendel Approximation**	[51]	IEEE Transactions on Fuzzy Systems
July 2007	Luís Alberto Lucas Tania M. Centeno Myriam R. Delgado	**VSCTR**	[42]	Proc. FUZZ-IEEE 2007
June 2008	Maowen Nie Woei Wan Tan	**Nie-Tan**	[47]	Proc. FUZZ-IEEE 2008
June 2008	Sarah Greenfield Robert I. John	Stratified TRS	[23]	Proc. IPMU 2008
May 2008	Feilong Liu	**α-Planes Representation**	[41]	Information Sciences
June 2009	Sarah Greenfield Francisco Chiclana Simon Coupland Robert I. John	**Collapsing**	[16]	Information Sciences
July 2009	Sarah Greenfield Francisco Chiclana Robert I. John	**CORL**	[18]	Proc. IFSA-EUSFLAT 2009
June 2011	Dongrui Wu Maowen Nie	**EIASC**	[50]	Proc. FUZZ-IEEE 2011
July 2011	Francisco Chiclana Shang-Ming Zhou	Type-1 OWA	[6]	Proc. EUSFLAT-LFA 2011
April 2012	Sarah Greenfield Francisco Chiclana Robert I. John Simon Coupland	**Sampling**	[21]	Information Sciences

Table 2.1. Chronology of publication of defuzzification methods. The methods shown in bold have been implemented and evaluated as reported in Chapter 5.

Part II

ORIGINAL STRATEGIES FOR TYPE-2 DEFUZZIFICATION

Chapter 3

Generalised Defuzzification: The Sampling Method

3.1 Introduction to the Sampling Method

We have seen (Section 1.4) that the exhaustive method is impractical, defeated by the extreme computational complexity arising from the proliferation of embedded sets. In response to this computational bottleneck, the sampling method, also known as the *sampling defuzzifier* [21], was devised as a cut-down version of the exhaustive method[1]. 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 2.1 are in fact estimates of the real centroid values. Therefore discretisation in itself may be seen as a form of sampling of the continuous type-2 fuzzy set.

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 (represented by circles in Figure 3.2). The construction of an embedded set requires the selection of a secondary domain (u) 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[2].

[1]Most of the work presented in this chapter is to be found in the literature as [21, 24] and [17].

[2]It is possible to amend the basic sampling algorithm to ensure uniqueness of the embedded sets in the sample. We term this strategy *elite sampling* (Subsection 3.5.1).

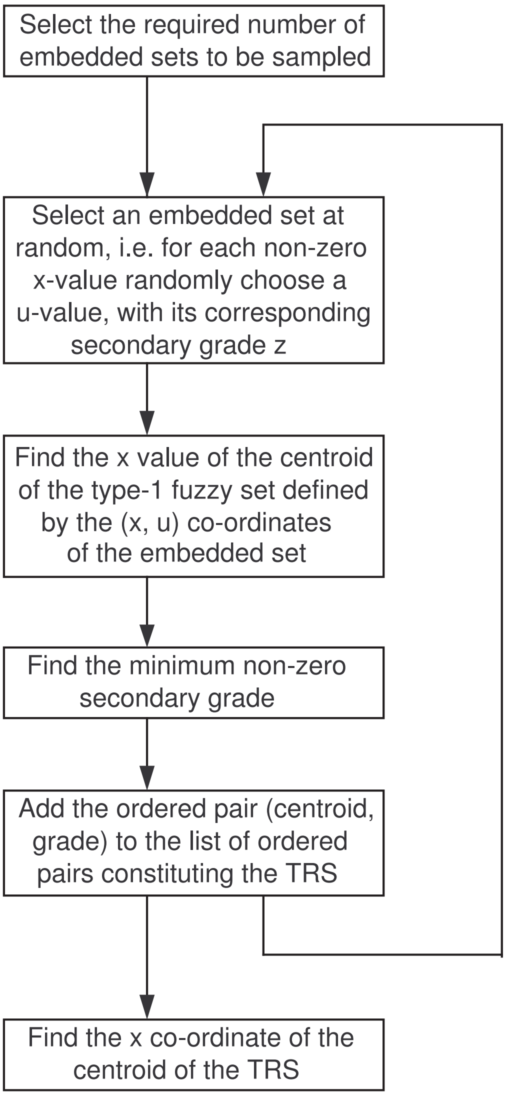


Fig. 3.1. Flow diagram of the Sampling Method.

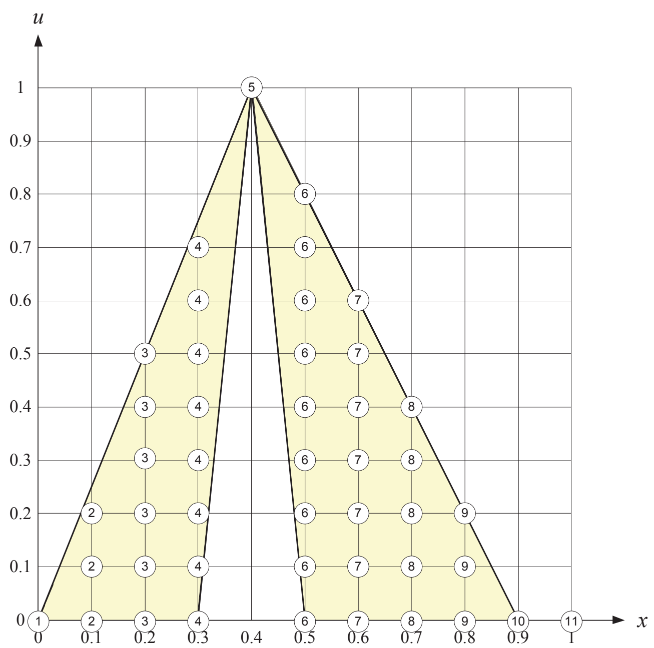


Fig. 3.2. The grid intersections (circles) within the FOU (shaded area) are the secondary domain values available for constructing an embedded set. They are numbered according to the vertical slice upon which they lie.

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.

The Sampling Algorithm The user having selected the necessary parameters, the embedded sets are randomly selected and processed (Algorithm 3.1). The sampling method, despite having the extra stages indicated in the algorithm, is radically simpler computationally than the exhaustive method.

```
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 2.1 ;
until the sample size is reached;
```

Algorithm 3.1: TRS obtained through sampling (in conjunction with the grid method of discretisation).

3.2 Evaluation of the Sampling Method: Experimental Design

Let E_n be the set of all subsets e_n of n embedded sets of a type-2 fuzzy set $\widetilde{A}$. It follows that each e_n is itself a type-2 fuzzy set. For our experiments, E_n will be our sample space[3]. In this sample space we define the following random variable $X : E_n \to [0,1]$, where $X(e_n)$ is the centroid of the type-1 fuzzy set derived from the type-reduction of e_n. We are interested in the distribution of X, and more specifically, as we will see later, in its mean. Because X is bounded then it is obvious that its mean, μ, and variance, σ^2, exist. The Central Limit Theorem [9, page 275] states that if a *large* random sample (of size N) is taken from a distribution with finite variance σ^2, then the sample mean distribution will be approximately a normal distribution with the same original distribution mean and a variance of σ^2/N. 'Large', in the context of the Central Limit Theorem, means 'over 30', therefore the sample sizes used in the experiments are all above 30.

[3]Sampling is the nature of the method described in this chapter; the sample space is the set of embedded sets. However the experimental evaluation also involves sampling; in this case the sample space is E_n.

In this section we describe in detail the design of the experiments conducted to validate our claim: The use of a sample of embedded sets rather than the whole set of embedded sets is sufficient to obtain ‘good’ estimates of the centroid of type-2 fuzzy sets.

3.2.1 Test Sets

Two test sets were specially constructed using Matlab^{TM}. They were devised to have reflectional symmetry, which makes their defuzzified values readily apparent, hence allowing the sampling method to be tested for accuracy. Their primary membership functions are the widely used Gaussian and triangular. In both cases the secondary membership functions are triangular, a shape which is often used in generalised type-2 fuzzy sets. The two sets are depicted in Figures 3.3 and 3.4, together with their associated FOUs.

Triangular Primary Membership Function This symmetrical test set was positioned off-centre on the *x*-axis to give a defuzzified value of 0.4 (Figure 3.3).

Gaussian Primary Membership Function This symmetrical test set was centred on the *x*-axis to give a defuzzified value of 0.5 (Figure 3.4).

Hypothesis testing was carried out in relation to both test sets to ascertain whether or not the estimates of the defuzzified values obtained from sampling method were statistically significant:

$$H_0 : \mu = \mu_0$$
$$H_1 : \mu \neq \mu_0.$$

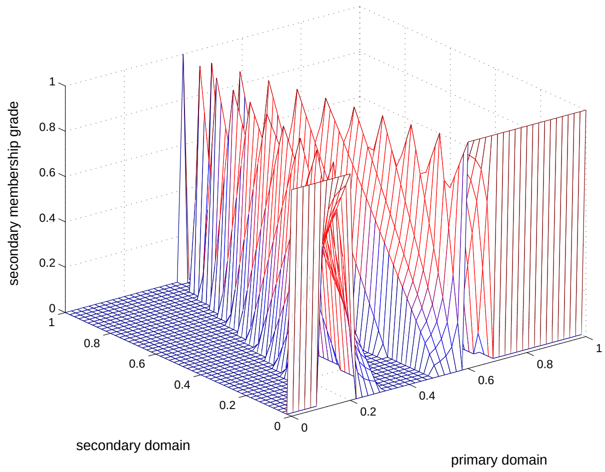
For the first test set $\mu_0 = 0.4$, while for the second test set $\mu_0 = 0.5$. As we shall see, in both cases the estimates were found to be highly satisfactory.

3.2.2 The Test Runs

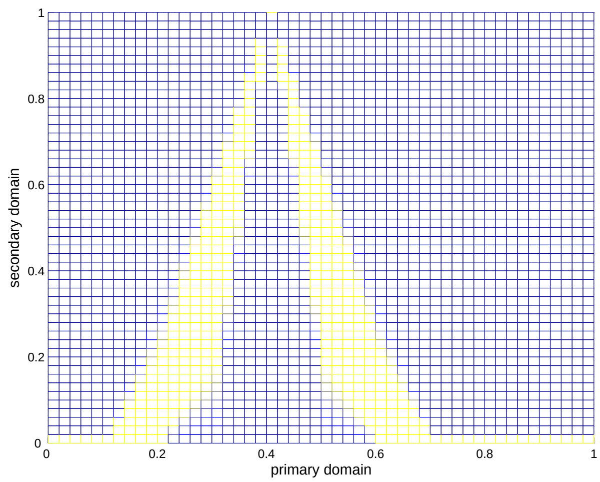
Within each experiment the degree of discretisation[4] and defuzzifier sample size were varied in the same way. Each experiment involved 20 runs, i.e. each test set was defuzzified 20 times using 5 degrees of discretisation (0.1, 0.05, 0.02, 0.01 and 0.0005) and 4 defuzzifier sample sizes of embedded sets (100, 1000, $10,000$, and $100,000$). For each one of the possible 20 combinations of the previous two parameters, the experimental sample size was set at $N = 1000$, a size sufficiently large for the Central Limit Theorem to be applicable. For these parameter settings, both the sample mean (i.e. mean defuzzified value) and sample standard deviation were calculated[5].

[4]The primary and secondary domains were assigned equal degrees of discretisation throughout.

[5]The sampling defuzzifier was coded in C and tested on a PC with a Pentium 4 CPU, a clock speed of 3.00 GHz, and a 0.99 GB RAM, running the MS Windows XP Professional operating system. 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.

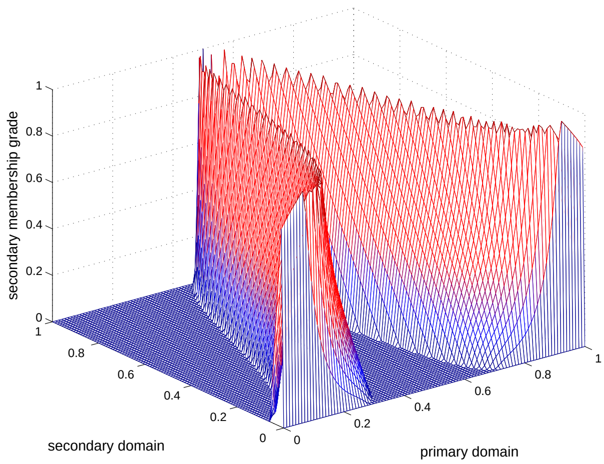


(a) 3-D representation

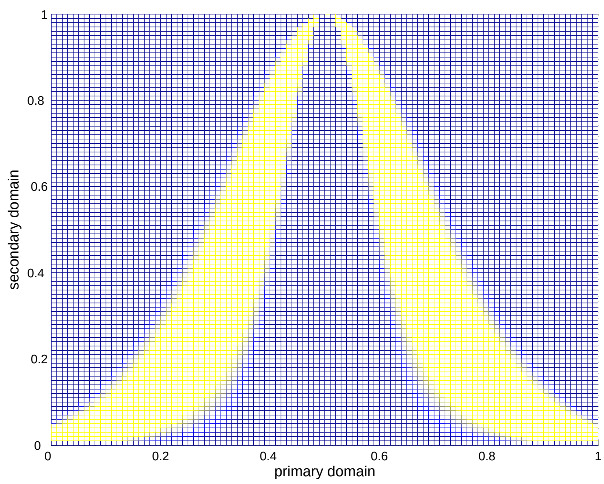


(b) FOU

Fig. 3.3. Type-2 fuzzy set: Triangular primary membership function, triangular secondary membership functions; degree of discretisation of primary and secondary domains is 0.02; defuzzified value = 0.4.



(a) 3-D representation



(b) FOU

Fig. 3.4. Type-2 fuzzy set: Gaussian primary membership function, triangular secondary membership functions; degree of discretisation of primary and secondary domains is 0.01; defuzzified value = 0.5.

3.3 Evaluation of the Sampling Method: Experimental Results

The tests were tabulated as Tables 3.1 and 3.2. The defuzzification times were not recorded in the tables, as for short runs they were found to be unreliable. [6] Those timings considered reliable were encouragingly speedy. For example, defuzzifying the Gaussian test set at a degree of discretisation of 0.01, a sample size of 100 embedded sets took 0.000848 seconds, and a sample size of 100000 embedded sets took 0.842263 seconds. A more computationally demanding example was an FIS generated test set at a degree of discretisation of 0.005, for which a sample size of 100 embedded sets took 0.001750 seconds, and a sample size of 100000 embedded sets took 1.733598 seconds.

3.3.1 Hypothesis Testing

Note that the variance of the random variable defined above is unknown in practice, and therefore we need to estimate it when applying our sampling method. For sufficiently large sample sizes (N), it is established that the statistic

$$t = \frac{\bar{x} - \mu}{s/\sqrt{N}}$$

has a *Student's t-distribution* [9, page 394] with $N-1$ degrees of freedom, where s is the sample standard deviation. The t-values (t_0) are also provided in Tables 3.1 and 3.2. The critical region, at the level of significance $\alpha = 0.05$, is $|t| > 1.96$, and therefore in all cases we have not found sufficient evidence to reject the null hypothesis.

3.3.2 Effect of Number of Embedded Sets on Accuracy

According to the Central Limit Theorem, the standard deviation of the sample mean is smaller than the population standard deviation. An increase of the sample size by a factor of 10 has the effect of decreasing the standard deviation of the sample mean by a factor of $\sqrt{10}$, i.e. if we divide the standard deviations of two samples from the same population with size n and $10n$, the value we would obtain should be close to 3.16. This value *is* approximately obtained in our experiment, which illustrates clearly the effect the number of embedded sets has on the accuracy of the estimated centroid.

3.3.3 Effect of Degree of Discretisation in Accuracy

For each particular number of embedded sets used in our experiment, we note that the lower the degree of discretisation the narrower in general are the 95% degree of confidence intervals for the mean. There is one exception to this in the second test set when the degree of discretisation is

[6] This is attributable to the time taken by operating system processes, which though negligible in proportion to longer runs, is significant in relation to short runs.

0.05. However, in all cases when the degree of discretisation is fixed, the width of the confidence interval decreases when the number of embedded sets increases.

Our experiments on test sets of known defuzzified values have shown that through the sampling method an enormous improvement in speed may be achieved, with no significant loss of accuracy. In the next section we look at the practical application of the method.

3.4 Practical Application of the Sampling Defuzzifier

The sampling defuzzifier is intended as a method that is run *once* using a pre-selected sample size of embedded sets. In what follows, we argue that for the practical application of the sampling method, we need only to select a random sample of embedded sets.

Let us assume that E represents the set of all embedded sets. We note that $E = E_1$ (Section 3.2). According to Algorithm 2.1, associated with each embedded set $e^i \in E$ is its centroid value $x_i \in [0,1]$ and minimum secondary membership grade z_i. The following random variable $C : E \to [0,1]$, with $C(e^i) = x_i$ can be defined. Again, owing to the boundedness property of C we know that its mean, ν, and variance, τ^2, exist.

The following result is well known [9, page 270]: If $\{X_i | i = 1,2,\ldots,n\}$ are n independent random variables with expectations and variances $\{(\nu_i, \tau_i^2) | i = 1,2,\ldots,n\}$, then $X = \sum_i^n \hat{w}_i X_i$, with $\hat{w}_i$ constants, is a random variable with the following mean and variance

$$\left(\sum_i^n \hat{w}_i \nu_i, \sum_i^n {\hat{w}_i}^2 \tau_i^2 \right).$$

Let us assume that $C_1, C_2, \ldots, C_n$ is a random sample from a population with mean, ν, and variance, τ^2, and let $Y = \sum_i^n \hat{w}_i C_i$, with $\hat{w}_i = w_i / \sum_i^n w_i$ and $0 < w_i \leq 1$ for all i. The mean and variance of Y are

$$\left(\nu, \tau^2 \left[\sum_i^n {\hat{w}_i}^2 \right] \right).$$

The construction of Y and the definition of the random variable X (Section 3.2) leads to the conclusion that $\nu = \mu$. Therefore, theoretically the sampling method could be applied in practical cases using just one random sample of embedded sets of sufficiently large size for the Central Limit Theorem to be applicable.

3.5 Variations on the Sampling Method

Since its initial publication in 2005 [24], the sampling method has been adapted, resulting in two variations, *importance sampling* [40] and *elite sampling*.

DEGREE OF DISCRETI-SATION	NUMBER OF EMBEDDED SETS	MEAN OF DEFUZZIFIED VALUES	STANDARD DEVIATION	t_0	$t_{0.05}$=1.96	SD1/SD2 RATIO	95% CONFIDENCE INTERVAL RANGE
0.1	100	0.399970	0.001838	-0.52	fail to reject H_0	3.03	0.00720496
0.1	1000	0.400021	0.000606	1.10	fail to reject H_0	3.54	0.00237552
0.1	10,000	0.400001	0.000171	0.18	fail to reject H_0	3.29	0.00067032
0.1	100,000	0.400003	0.000052	1.82	fail to reject H_0		0.00020384
0.05	100	0.400010	0.001372	0.23	fail to reject H_0	3.22	0.00537824
0.05	1000	0.399988	0.000426	-0.89	fail to reject H_0	3.23	0.00166992
0.05	10,000	0.399995	0.000132	-1.20	fail to reject H_0	3.22	0.00051744
0.05	100,000	0.399999	0.000041	-0.77	fail to reject H_0		0.00016072
0.02	100	0.399969	0.000891	-1.10	fail to reject H_0	3.26	0.00349272
0.02	1000	0.400011	0.000273	1.27	fail to reject H_0	2.97	0.00107016
0.02	10,000	0.400001	0.000092	0.34	fail to reject H_0	3.17	0.00036064
0.02	100,000	0.400000	0.000029	0.00	fail to reject H_0		0.00011368
0.01	100	0.400011	0.000659	0.53	fail to reject H_0	3.20	0.00258328
0.01	1000	0.400007	0.000206	1.07	fail to reject H_0	3.22	0.00080752
0.01	10,000	0.400001	0.000064	0.49	fail to reject H_0	3.20	0.00025088
0.01	100,000	0.400000	0.000020	0.00	fail to reject H_0		0.00007840
0.005	100	0.399981	0.000485	-1.24	fail to reject H_0	3.34	0.00190120
0.005	1000	0.399995	0.000145	-1.09	fail to reject H_0	3.09	0.00056840
0.005	10,000	0.400001	0.000047	0.67	fail to reject H_0	3.13	0.00018424
0.005	100,000	0.400000	0.000015	0.00	fail to reject H_0		0.00005880

Table 3.1. Triangular primary membership function, defuzzified value = 0.4.

DEGREE OF DISCRETI-SATION	NUMBER OF EMBEDDED SETS	MEAN OF DEFUZZIFIED VALUES	STANDARD DEVIATION	t_0	$t_{0.05}$=1.96	SD1/SD2 RATIO	95% CONFIDENCE INTERVAL RANGE
0.1	100	0.499986	0.001794	-0.25	fail to reject H_0	3.21	0.00703248
0.1	1000	0.499987	0.000559	-0.74	fail to reject H_0	3.14	0.00219128
0.1	10,000	0.499993	0.000178	-1.24	fail to reject H_0	3.12	0.00069776
0.1	100,000	0.500002	0.000057	1.11	fail to reject H_0		0.00022344
0.05	100	0.499963	0.002264	-0.52	fail to reject H_0	3.11	0.00887488
0.05	1000	0.499998	0.000728	-0.09	fail to reject H_0	3.21	0.00285376
0.05	10,000	0.499999	0.000227	-0.14	fail to reject H_0	3.29	0.00088984
0.05	100,000	0.499997	0.000069	-1.37	fail to reject H_0		0.00027048
0.02	100	0.499954	0.001390	-1.05	fail to reject H_0	3.33	0.00544880
0.02	1000	0.499988	0.000417	-0.91	fail to reject H_0	3.09	0.00163464
0.02	10,000	0.499995	0.000135	-1.17	fail to reject H_0	3.14	0.00052920
0.02	100,000	0.500000	0.000043	0.00	fail to reject H_0		0.00016856
0.01	100	0.500015	0.000896	0.53	fail to reject H_0	3.22	0.00351232
0.01	1000	0.499996	0.000278	-0.46	fail to reject H_0	3.02	0.00108976
0.01	10,000	0.500002	0.000092	0.69	fail to reject H_0	3.17	0.00036064
0.01	100,000	0.499999	0.000029	-1.09	fail to reject H_0		0.00011368
0.005	100	0.500009	0.000680	0.42	fail to reject H_0	3.21	0.00266560
0.005	1000	0.500005	0.000212	0.75	fail to reject H_0	3.16	0.00083104
0.005	10,000	0.500001	0.000067	0.47	fail to reject H_0	3.19	0.00026264
0.005	100,000	0.500000	0.000021	0.00	fail to reject H_0		0.00008232

Table 3.2. Gaussian primary membership function, defuzzified value = 0.5.

3.5.1 Elite Sampling

The sampling algorithm (Algorithm 3.1; Figure 3.1) allows a given domain value to be associated with more than one secondary grade. However in *elite sampling* (Algorithm 3.2), 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 is designed to be more accurate than basic sampling in situations where there are a significant number of repetitions of embedded sets in the sample. However elite sampling is more computationally complex than basic sampling. In Chapter 5 basic and elite sampling are contrasted for accuracy and speed.

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 2.1 ;
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

Algorithm 3.2: TRS obtained through elite sampling (in conjunction with the grid method of discretisation).

3.5.2 Importance Sampling

Linda and Manic's importance sampling method [40], is a refinement of the basic sampling method which employs uniform sampling. In importance sampling a specific probability distribution function is used. Experiments demonstrate [40] that importance sampling markedly reduces the variance of the sample, giving a performance superior to that of uniform sampling. Moreover comparison of the FIS output surfaces showed responses to be smoother and more stable using the importance sampling technique.

Summary

In this chapter we have seen how the sampling method is motivated by the desirability of cutting down on the number of embedded sets contributing to the TRS. This approach enormously reduces the computational complexity of type-2 defuzzification. Though an approximation, we have demonstrated experimentally that this method is of high accuracy. We shall see in Chapter 5 how the sampling method compares with other generalised methods for both speed and accuracy. The next chapter will present an interval defuzzification technique, the Greenfield-Chiclana Collapsing Defuzzifier.

Chapter 4

Interval Defuzzification:
The Greenfield-Chiclana Collapsing Defuzzifier

4.1 Introduction to the Collapsing Method

In Section 1.4 we saw how the exhaustive method is impractical, owing to the excessive computational requirement of processing all the numerous embedded sets in the aggregated set. Chapter 3 offered one response to this computational bottleneck, the sampling method. This chapter presents another solution, the **Greenfield-Chiclana Collapsing Defuzzifier (GCCD)**, which is the major contribution to knowledge contained in this thesis. Most of the research described in this chapter is reported in the literature as [16, 18–20], and [15]. This easy to use iterative defuzzification technique for discretised interval type-2 fuzzy sets was developed with two questions in mind:

1. A type-1 fuzzy set is easily defuzzified, in contrast to a type-2 fuzzy set. Could the totality of embedded sets in a type-2 fuzzy set be replaced by just one type-1 fuzzy set with the same defuzzified value as that of the original set, but in a simpler and more efficient way than through exhaustive defuzzification?
2. Type-2 fuzzy sets can be thought of as having been created out of type-1 fuzzy sets through a process of blurring (Subsection 1.1.1). Is it possible for the blurring that transforms a type-1 to a type-2 membership function to be reversed?

The collapsing technique is inextricably allied with the concept of the *Representative Embedded Set (RES)* presented below in Section 4.2; this section also describes how both questions are answered by the collapsing method's creation of an approximation to this representative embedded set. An interesting feature of an FIS is that embedded sets are only employed during the final defuzzification stage. The collapsing method completely eliminates these embedded sets; however the embedded set concept is used in the proof of the results (Subsections 4.3.2 and 4.4.3) upon which the method is based.

The contents of this chapter were originally published in 2009 [16]. In this journal paper the results were presented in two stages:

Simple RES We consider the simple interval case, in which each vertical slice consists of two points, corresponding to the lower membership function and the upper membership function (Sections 4.2 and 4.3).

Interval RES We then consider the more complex interval case, whereby each vertical slice consists of a finite number of points, whose primary membership grades are not necessarily evenly spaced (Section 4.4).

The simple interval case is the case of an interval type-2 fuzzy set as commonly understood. The original intention was to extend the collapsing method to generalised type-2 fuzzy sets. The strategy for achieving this aim was to first consider the case where each vertical slice is discretised into more than 2 points, resulting in an RES which we termed the *Interval RES*. So far the collapsing method has not been extended beyond the interval case; this is still under consideration as a topic for future research.

In this chapter the notation '$||A||$' is used for the *scalar cardinality* of A (Appendix B).

4.2 The Representative Embedded Set

Consider an *interval* type-2 fuzzy set discretised such that all primary memberships consist of 2 points. It is helpful to think of the interval type-2 fuzzy set as a *blurred* type-1 membership function [44, page 118], creating lower and upper membership functions (Definitions 1.3 and 1.4). The collapsing technique is the reversal of this blurring process in order to derive a type-1 fuzzy set from an interval type-2 fuzzy set. The type-1 fuzzy set's membership function is calculated so that its defuzzified value is equal to that of the interval type-2 fuzzy set. A type-1 fuzzy set is easily defuzzified, and to do so would be to find the defuzzified value of the original interval type-2 fuzzy set. Hence the collapsing process reduces the computational complexity of interval type-2 defuzzification. We term this special type-1 fuzzy set the *representative embedded set*. It is a *representative set* because it has the same defuzzified value as the original interval type-2 fuzzy set. It is an embedded set because it lies within the FOU of the interval type-2 fuzzy set.

We formally define the concepts of a representative set and a representative embedded set[1].

Definition 4.1 (Representative Set)**.** *Let $\tilde{F}$ be a type-2 fuzzy set with defuzzified value $X_{\tilde{F}}$. Then the type-1 fuzzy set R is a* Representative Set (RS) *of $\tilde{F}$ if its defuzzified value (X_R) is equal to that of $\tilde{F}$, i.e. $X_R = X_{\tilde{F}}$.*

[1]We show below that approximations to these sets exist in the simple and interval cases. We suspect they also exist for the generalised case, but this is as yet unproven

Definition 4.2 (Representative Embedded Set)**.** *Let $\tilde{F}$ be a type-2 fuzzy set with defuzzified value $X_{\tilde{F}}$. Then type-1 fuzzy set R is a* representative embedded set (RES) *of $\tilde{F}$ if it is a RS of $\tilde{F}$* and *its membership function lies within the FOU of $\tilde{F}$.*

We begin by considering the case of an interval type-2 fuzzy set whose vertical slices are discretised into 2 points.

4.3 SimpleRES: RES of an Interval Type-2 Fuzzy Set with Primary Membership Discretised into 2 Points

An interval type-2 fuzzy set is a type-2 fuzzy set in which every secondary membership grade takes the value 1. Because of this, such a set is completely specified by its FOU (Definition 1.2). In the analysis which follows, to speak in terms of the FOU of an interval type-2 fuzzy set is equivalent to referring to the interval type-2 fuzzy set itself. It is assumed that the domain is discretised into an arbitrary number m of vertical slices, and that each vertical slice is discretised into 2 points, which are the end-points of its primary membership.

The objective of this analysis is to derive an expression for the membership function of an RES in terms of the upper and lower membership functions of the interval type-2 fuzzy set to be defuzzified. We term this RES the *Simple RES*. Our strategy is two-stage:

1. We derive a formula for the special case of the interval FOU which has only one blurred vertical slice. We call this the *Simple Solitary Collapsed Slice Lemma*.
2. We generalise this formula to the typical interval FOU with secondary membership function having 2 points, one corresponding to the upper membership function (U) and the other one to the lower membership function (L). We call this the *Simple Representative Embedded Set Approximation*. It is an approximation because not every embedded set is taken into account in deriving the RES.

4.3.1 Solitary Collapsed Slice Lemma (SCSL)

In this subsection we concentrate on the derivation of the RES for a special case of an interval FOU formed by (upwardly) blurring the membership function of a type-1 fuzzy set (A) at a single domain value x_I (Figure 4.1), to create a vertical slice which is an interval as opposed to a point. This interval is the primary membership at x_I. The FOU formed by this blurring is depicted in Figure 4.2, and consists of the shaded triangular region plus the line A. We derive a formula for the membership function of the RES of this somewhat unusual interval FOU, in terms of the original type-1 membership function and the amount of blurring.

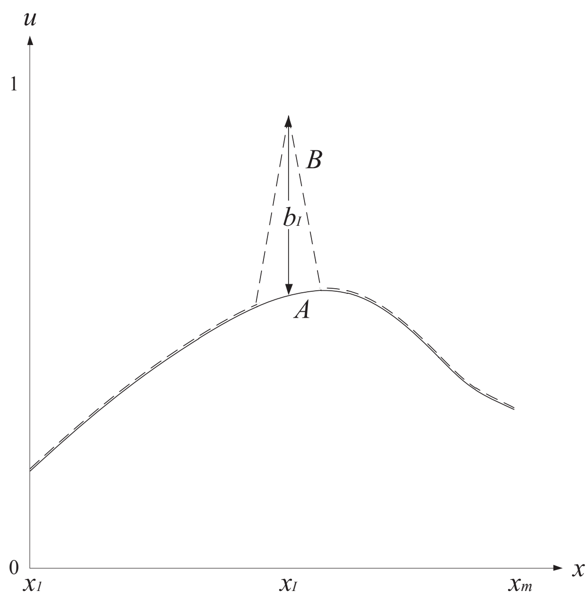


Fig. 4.1. At $x = x_I$ the membership function of type-1 fuzzy set A has been blurred, increasing the membership grade by the amount b_I, creating a new type-1 fuzzy set B.

Let A be a non-empty type-1 fuzzy set that has been discretised into m vertical slices (at x_1, x_2, $\ldots$, x_m). We calculate X_A, the defuzzified value of A, by finding the centroid (Appendix B) of A:

$$X_A = \frac{\sum_{i=1}^{i=m} \mu_A(x_i)x_i}{\sum_{i=1}^{i=m} \mu_A(x_i)} = \frac{\sum_{i=1}^{i=m} \mu_A(x_i)x_i}{\|A\|}.$$

Now suppose the membership function of A is blurred upwards at domain value x_I, so that x_I, instead of corresponding to the point $\mu_A(x_I)$, corresponds to the co-domain range $[\mu_A(x_I), \mu_A(x_I) + b_I]$. Let B (Figure 4.1) be the type-1 fuzzy set whose membership function is the same as that of A apart from at the point x_I, for which $\mu_B(x_I) = \mu_A(x_I) + b_I$. X_B, the defuzzified value of B, may be calculated:

$$X_B = \frac{\sum \mu_B(x_i)x_i}{\sum \mu_B(x_i)} = \frac{\sum \mu_A(x_i)x_i + b_I x_I}{\sum \mu_A(x_i) + b_I} = \frac{\|A\|X_A + b_I x_I}{\|A\| + b_I} = X_A + \frac{b_I(x_I - X_A)}{\|A\| + b_I}.$$

Let $\tilde{F}$ (Figure 4.2) be an interval type-2 fuzzy set whose lower membership function is A and upper membership function is B. Exhaustive defuzzification (Subsection 2.3.1) requires that *all* the embedded sets of a type-2 fuzzy set be processed to form the type-reduced set. $\tilde{F}$ contains only two embedded sets, namely A and B. Therefore, we find the defuzzified value of $\tilde{F}$ by calculating

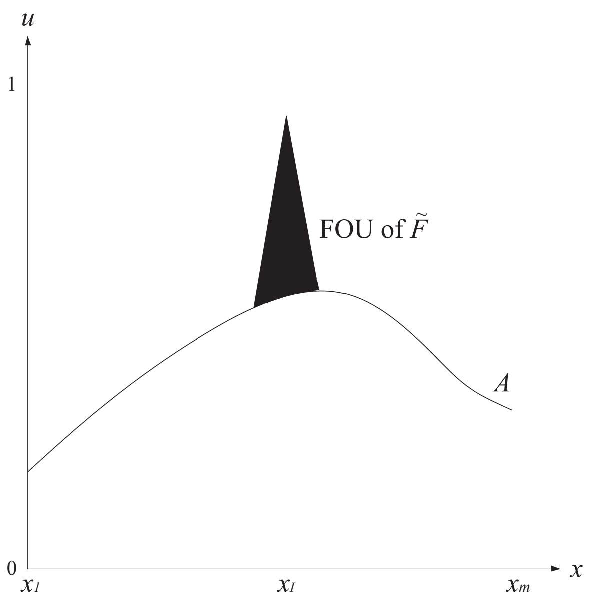


Fig. 4.2. FOU of interval type-2 fuzzy set $\tilde{F}$, which consists of the original line of type-1 fuzzy set A, plus the triangular region.

the mean of X_A and X_B, i.e. $\frac{1}{2}(X_A + X_B)$. Let $X_{\tilde{F}}$ be the defuzzified value of $\tilde{F}$. $X_{\tilde{F}}$ will be expressed in terms of $\|A\|$, X_A, x_I and b_I, all of which are known values:

$$X_{\tilde{F}} = \frac{1}{2}(X_A + X_B) = \frac{1}{2}\left(X_A + X_A + \frac{b_I(x_I - X_A)}{\|A\| + b_I}\right) = X_A + \frac{b_I(x_I - X_A)}{2(\|A\| + b_I)}.$$

Let R be the RES of $\tilde{F}$ such that the membership function of R is the same as that of A for all domain values x_i apart from x_I. At this point the membership function deviates from that of A so that $\mu_R(x_I)$ takes the value $\mu_A(x_I) + r_I$. Figure 4.3 depicts the membership function of R. Following the same chain of reasoning as in the derivation of X_B, we work out an expression for X_R in terms of $\|A\|$, X_A, x_I and r_I:

$$X_R = X_A + \frac{r_I(x_I - X_A)}{\|A\| + r_I}.$$

The defuzzified values X_R and $X_{\tilde{F}}$ are by definition equal, and by equating these values we are able to obtain a formula for r_I in terms of $\|A\|$ and b_I:

$$X_R = X_{\tilde{F}} \Rightarrow r_I = \frac{b_I \|A\|}{2\|A\| + b_I}.$$

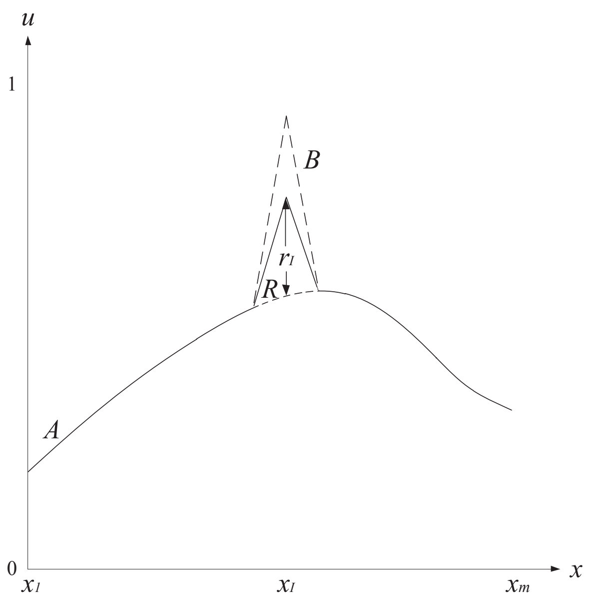


Fig. 4.3. R, the representative embedded set of $\tilde{F}$, is indicated by the undashed line.

We have arrived at the membership function of R, and in so doing proved the Simple Solitary Collapsed Slice Lemma (Simple SCSL)[2]:

Lemma 4.1 (Simple Solitary Collapsed Slice Lemma)**.** *Let A be a non-empty discretised type-1 fuzzy set which has been blurred upwards by amount b_I at a single point x_I to form the FOU of interval type-2 fuzzy set $\tilde{F}$. Then R, the RES of $\tilde{F}$, has a membership function such that*

$$\mu_R(x_i) = \begin{cases} \mu_A(x_i) + \dfrac{\|A\| b_I}{2\|A\| + b_I} & \textit{if } i = I, \\ \mu_A(x_i) & \textit{otherwise.} \end{cases}$$

4.3.2 Simple RESA

We extend the Simple Solitary Collapsed Slice Lemma to the typical situation in which every point of the type-1 membership function has been blurred. First we present the concept behind the approximation: How an interval type-2 fuzzy set may be collapsed to create an approximation to an RES.

[2]The Simple SCSL, where only one slice is collapsed, gives an exact result, the RES. We go on to show how when more than one slice is collapsed, an approximation to the RES is obtained — the Representative Embedded Set Approximation.

In Subsection 4.3.1, we have considered an extremely atypical interval FOU whose membership function follows the course of a type-1 fuzzy set apart from at one point x_I, at which its membership grade opens up to form a secondary domain $[\mu(x_I), \mu(x_I) + b_I]$. We have done this to provide a simple yet illustrative example of the collapsing process, as a basis for generalisation to the typical interval FOU. The Simple SCSL (Subsection 4.3.1) tells us how to calculate the RES for this special case of an interval type-2 fuzzy set.

Now we proceed to look at the typical interval FOU, in which the upper membership grade is greater than the lower membership grade at a minimum of 2 points. The difference between the lower and upper membership grades at any given point is the amount of blur (b_i) at that point, i.e. $\mu_U(x_i) - \mu_L(x_i) = b_i$. The Simple SCSL does not apply in this situation. However, this lemma may be applied repeatedly to FOUs assembled in stages using slices taken from the interval type-2 fuzzy set.

Collapsing the 1st FOU to Form RES R_1

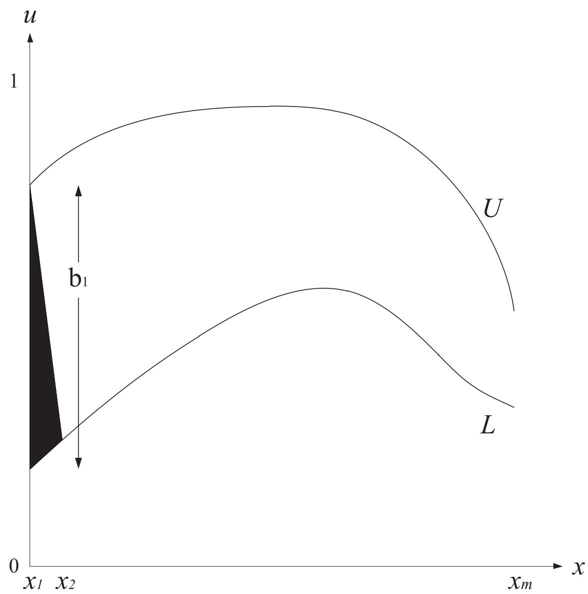


Fig. 4.4. The first slice in interval type-2 fuzzy set $\tilde{F}$.

The first interval FOU to be collapsed (Figure 4.4) comprises the slice at x_1, plus the rest of the lower membership function L, (represented by the shaded triangular region plus the line L). The lower membership function of the FOU is the line L, and the upper membership function starts

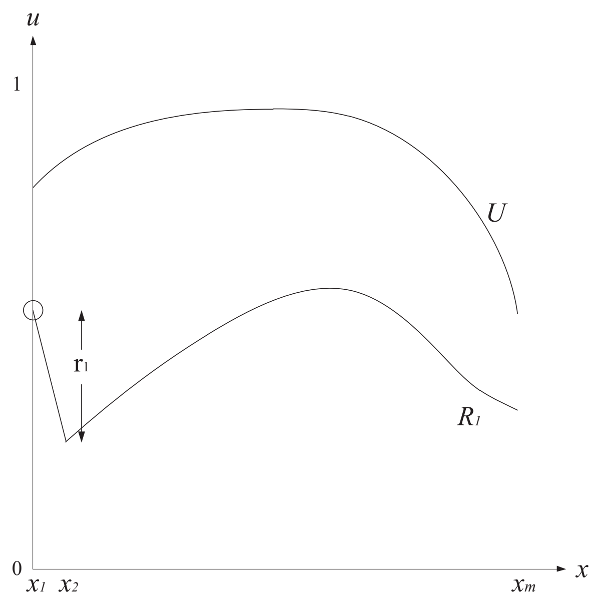


Fig. 4.5. The first slice collapsed, creating RESA R_1 for the interval type-2 fuzzy set $\tilde{F}$. The circle indicates the first tuple of the RESA.

(at x_1) at the line U, but immediately descends to L (slice x_2), after which it follows the course of L (slices $x_2 \ldots x_m$). The Simple SCSL tells us that this interval type-2 fuzzy set may be collapsed into its RES R_1, depicted in Figure 4.5. The collapse increases the membership grade $\mu_L(x_1)$ by r_1 to $\mu_{R_1}(x_1)$.

Collapsing the 2^{nd} FOU to Form RES R_2

We now move on to the second FOU. Figure 4.6 shows this FOU before it is collapsed. The Simple SCSL is re-applied, but instead of the lower membership function being L, it is now R_1. The RES of the second FOU is R_2, which is depicted in Figure 4.7.

Collapsing the $(k+1)^{th}$ FOU to Form RES $R_{(k+1)}$

Suppose FOUs $1,\ldots,k$ have been collapsed in turn, with R_k being the most recently formed RES. Then it is the turn of the $(k+1)^{th}$ FOU to be collapsed. The lower membership function is R_k. This situation prior to the $(k+1)^{th}$ FOU's collapse is represented in Figure 4.8; the situation after the collapse in Figure 4.9.

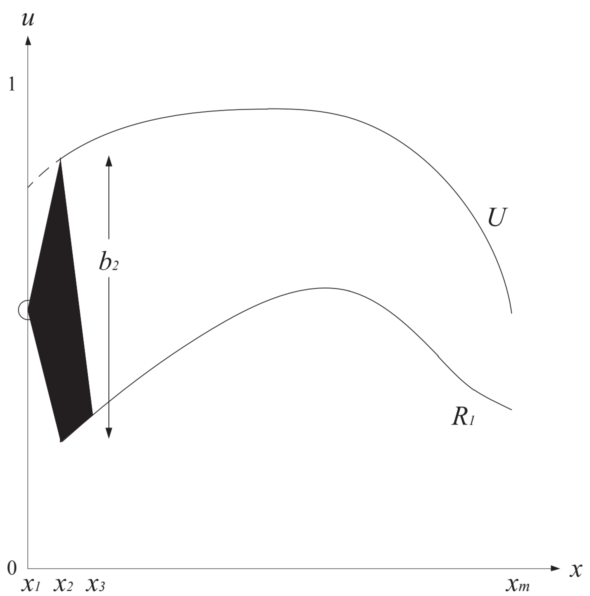


Fig. 4.6. For the interval type-2 fuzzy set $\tilde{F}$, the first slice is collapsed, and the second slice is shown. The circle indicates the first tuple of the RESA.

Collapsing the m^{th} FOU to Form an Approximation for the RES of the Entire Interval Type-2 Fuzzy Set

Suppose FOUs $1,\ldots,m-1$ have been collapsed in turn, with R_{m-1} being the most recently formed RES. Then it is the turn of the m^{th} FOU to be collapsed. The lower membership function is R_{m-1}, and the slice to be collapsed is slice m at x_m. After the collapse the new lower membership function is R_m. As the m^{th} slice is the final slice, then R_m is the RES (R) of the FOU of R_m. R_m is an approximation to the RES of $\tilde{F}$, the original type-2 fuzzy set, because not all the embedded sets have been taken into account simultaneously.

We now state and prove the Simple Representative Embedded Set Approximation (Simple RESA).

Theorem 4.1 (Simple Representative Embedded Set Approximation)**.** *The membership function of the* embedded set *R derived by dynamically collapsing slices of a discretised type-2 interval fuzzy set $\tilde{F}$, having lower membership function L and upper membership function U, is:*

$$\mu_R(x_i) = \mu_L(x_i) + r_i$$

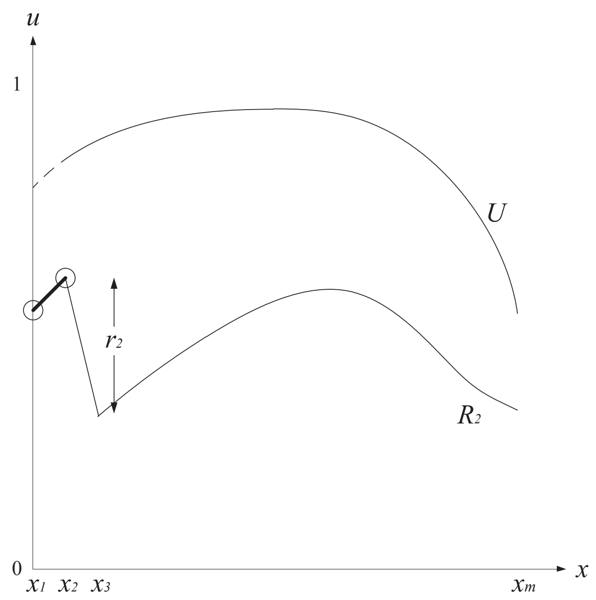


Fig. 4.7. Slices 1 and 2 collapsed, creating RESA R_2 for the interval type-2 fuzzy set $\tilde{F}$. The circles indicate the first two tuples of the RESA.

with

$$r_i = \frac{\left(\|L\| + \sum_{j=1}^{i-1} r_j\right) b_i}{2\left(\|L\| + \sum_{j=1}^{i-1} r_j\right) + b_i},$$

and $b_i = \mu_U(x_i) - \mu_L(x_i),\ r_0 = 0.$

Proof. Proof by induction on the number of collapsing vertical slices (k) will be used. As discussed above, let R_1 be the type-1 fuzzy set formed by collapsing slice 1, R_2 by collapsing slices 1 and 2, and R_i by collapsing slices 1 to i. R_m is the approximate RES, R, of $\tilde{F}$.

Basis (Collapsing the 1st slice to form R_1): Figures 4.4 and 4.5 depict the collapse of the first slice. The resultant RES is R_1. For R_1, $i = 1$, and $\sum_1^{i-1} r_j = 0$. We need to prove that

$$\mu_{R_1}(x_1) = \mu_L(x_1) + \frac{\|L\| b_1}{2\|L\| + b_1},$$

but this is actually what we have when we apply the Simple SCSL for $i = 1$.

Induction hypothesis: Assume the theorem is true for R_k, i.e. that slices $1, \ldots, k$ have been col-

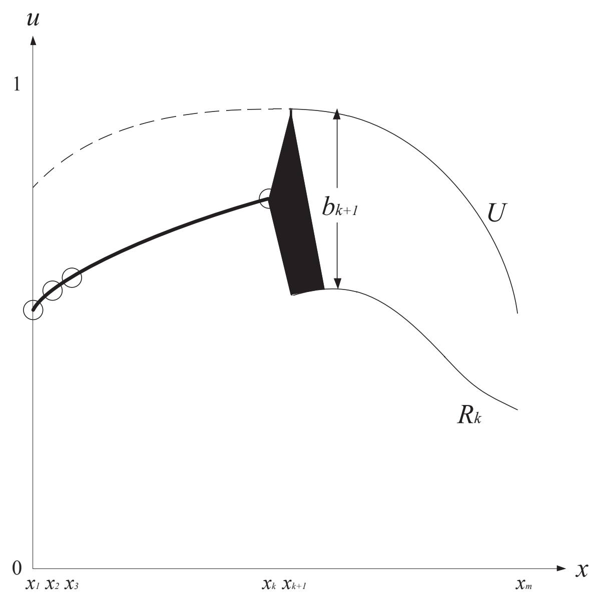


Fig. 4.8. Slices 1 to k collapsed, slice $(k+1)$ about to be collapsed, for interval type-2 fuzzy set $\tilde{F}$. The circles indicate the tuples of the RESA.

lapsed to form type-1 fuzzy set R_k, and that

$$\mu_{R_k}(x_i) = \mu_L(x_i) + \frac{\left(\|L\| + \sum_{j=1}^{i-1} r_j\right) b_i}{2\left(\|L\| + \sum_{j=1}^{i-1} r_j\right) + b_i}.$$

(In this formula, for $i > k$, $b_i = 0$.)

Induction Step: Now we collapse slice $(k+1)$, which is a single slice. Applying the solitary collapsed slice lemma to R_k we obtain:

$$r_{(k+1)} = \frac{\|R_k\| b_{(k+1)}}{2\|R_k\| + b_{(k+1)}}.$$

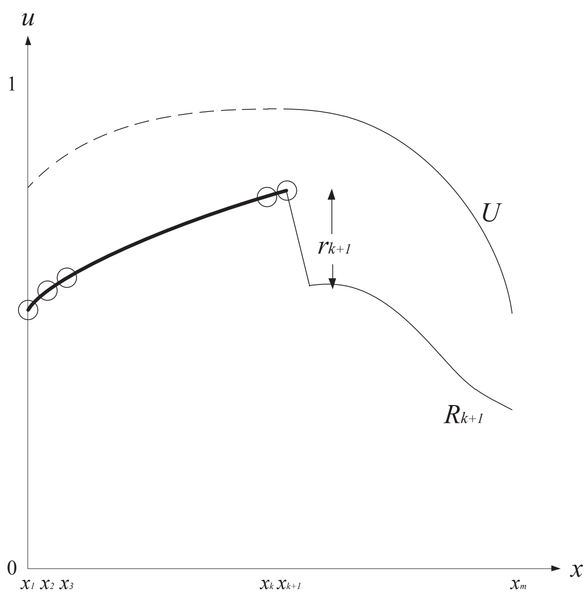


Fig. 4.9. Slices 1 to $(k+1)$ collapsed, creating RESA $R_{(k+1)}$ for interval type-2 fuzzy set $\tilde{F}$. The circles indicate the tuples of the RESA.

We need to prove that $\forall\ i$

$$\mu_{R_{(k+1)}}(x_i) = \mu_L(x_i) + \frac{\left(\|L\| + \sum_{j=1}^{i-1} r_j\right) b_i}{2\left(\|L\| + \sum_{j=1}^{i-1} r_j\right) + b_i}.$$

The proof will be split into three cases.

Case 1: $1 \le i \le k$ In this case we know that $\mu_{R_{(k+1)}}(x_i) = \mu_{R_k}(x_i)$. Applying the induction hypothesis we obtain:

$$\mu_{R_{(k+1)}}(x_i) = \mu_{R_k}(x_i) = \mu_L(x_i) + \frac{\left(\|L\| + \sum_{j=1}^{i-1} r_j\right) b_i}{2\left(\|L\| + \sum_{j=1}^{i-1} r_j\right) + b_i}.$$

Case 2: $i = k+1$

$$\mu_{R_{(k+1)}}(x_i) = \mu_L(x_i) + r_i = \mu_L(x_i) + r_{(k+1)} = \mu_L(x_i) + \frac{\|R_k\| b_i}{2\|R_k\| + b_i}.$$

We know that

$$\|R_k\| = \sum_{j=1}^{m} \mu_{R_k}(x_j) = \sum_{j=1}^{k} (\mu_L(x_j) + r_j) + \sum_{j=k+1}^{m} \mu_L(x_j) = \|L\| + \sum_{j=1}^{k} r_j,$$

and therefore we obtain

$$\mu_{R_{(k+1)}}(x_i) = \mu_L(x_i) + \frac{\left(\|L\| + \sum_{j=1}^{k} r_j\right) b_i}{2\left(\|L\| + \sum_{j=1}^{k} r_j\right) + b_i}.$$

Since $k = i-1$,

$$\mu_{R_{(k+1)}}(x_i) = \mu_L(x_i) + \frac{\left(\|L\| + \sum_{j=1}^{i-1} r_j\right) b_i}{2\left(\|L\| + \sum_{j=1}^{i-1} r_j\right) + b_i}.$$

Case 3: $i > k+1$

$$\mu_{R_{(k+1)}}(x_i) = \mu_{R_k}(x_i) = \mu_L(x_i) + \frac{\left(\|L\| + \sum_{j=1}^{i-1} r_j\right) b_i}{2\left(\|L\| + \sum_{j=1}^{i-1} r_j\right) + b_i}.$$

Again, in this last expression, for $i > k+1$, $b_i = 0$. Therefore the induction hypothesis is true for $k+1$.

Conclusion: We conclude that $\forall\, i$,

$$\mu_R(x_i) = \mu_L(x_i) + \frac{\left(\|L\| + \sum_{j=1}^{i-1} r_j\right) b_i}{2\left(\|L\| + \sum_{j=1}^{i-1} r_j\right) + b_i}.$$

□

4.4 Interval RES: RES of an Interval Set Discretised into n Points

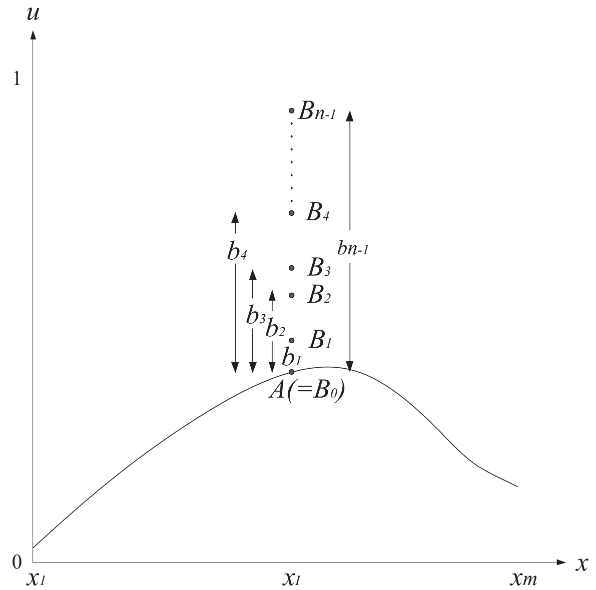


Fig. 4.10. A vertical slice, discretised into more than 2 co-domain points.

In this section, we shall derive the RES for an interval FOU, $\tilde{F}$, formed by (upwardly) blurring the membership function of a type-1 fuzzy set (A) at a single domain value x_I, to create a vertical slice which is an interval as opposed to a single point ($\mu_A(x_I)$), discretised with $n (n \geq 2)$ points $B_0 (= \mu_A(x_I)), B_1, B_2, \ldots, B_{n-1}$ at distance $b_0 (= 0), b_1, b_2, \ldots, b_{n-1}$ from $\mu_A(x_I)$ (Figure 4.10). We seek an approximate formula for the membership function of the RES of this interval FOU, in terms of the original type-1 membership function and $b_1, b_2, \ldots, b_{n-1}$.

Exhaustive defuzzification requires that *all* the embedded sets of a type-2 fuzzy set be processed to form the type-reduced set. $\tilde{F}$ contains n embedded sets, namely $A(= B_0)$, B_1, B_2, $\ldots$, B_{n-1}. We therefore find the defuzzified value of $\tilde{F}$ by calculating the mean of X_A and $X_{B_1}, X_{B_2}, \ldots,$ $X_{B_{n-1}}$, where X_{B_i} is the defuzzified value of B_i, i.e. $X_{\tilde{F}} = \frac{1}{n}\left(X_A + X_{B_1} + X_{B_2} + \ldots + X_{B_{n-1}}\right)$.

4.4.1 Interval Solitary Collapsed Slice Lemma

Let R be the RES of $\tilde{F}$ such that the membership function of R is the same as that of A for all domain values x_i apart from x_I. At this point the membership function deviates from that of A so that $\mu_R(x_I)$ takes the value $\mu_A(x_I)+r_I$. From Subsection 4.3.1 we have:

$$X_R = X_A + \frac{r_I(x_I - X_A)}{\|A\| + r_I},$$

and

$$X_{B_i} = X_A + \frac{b_i(x_I - X_A)}{\|A\| + b_i}, \forall i = 1, \ldots, n-1.$$

We know that

$$X_R = \frac{1}{n}\left(X_A + X_{B_1} + X_{B_2} + \ldots + X_{B_{n-1}}\right),$$

from which we obtain

$$X_A + \frac{r_I(x_I - X_A)}{\|A\| + r_I} = \frac{1}{n}\sum_{i=0}^{n-1}\left(X_A + \frac{b_i(x_I - X_A)}{\|A\| + b_i}\right),$$

and simplifying,

$$\frac{r_I}{\|A\| + r_I} = \frac{1}{n}\sum_{i=0}^{n-1}\frac{b_i}{\|A\| + b_i}.$$

Denoting $C = \sum_{i=0}^{n-1}\frac{b_i}{\|A\| + b_i}$ we get

$$r_I = \frac{C \cdot \|A\|}{n - C}.$$

Using the notation $w_i = \frac{1}{\|A\| + b_i}$ and $\bar{w}_i = \frac{w_i}{\sum_{i=0}^{n-1} w_i}$ we finally arrive at[3]:

$$r_I = \sum_{i=0}^{n-1} \bar{w}_i b_i.$$

r_I is a normalised weighted average of $b_0, \ldots b_{(n-1)}$, the weight corresponding to b_i (w_i) being proportional to $\frac{1}{\|A\| + b_i}$. $\mu_A(x_i) + r_i$ is therefore located between $\mu_A(x_i)$ (B_0) and $\mu_A(x_i) + b_{(n-1)}$

[3] The stages of this deduction are:

$$n - C = \sum_{i=0}^{n-1} 1 - \sum_{i=0}^{n-1}\frac{b_i}{\|A\| + b_i} = \sum_{i=0}^{n-1}\left(1 - \frac{b_i}{\|A\| + b_i}\right) = \sum_{i=0}^{n-1}\frac{\|A\|}{\|A\| + b_i} = \|A\| \cdot \sum_{i=0}^{n-1}\frac{1}{\|A\| + b_i} = \|A\| \cdot \sum_{i=0}^{n-1} w_i.$$

$$C = \sum_{i=0}^{n-1}\frac{b_i}{\|A\| + b_i} = \sum_{i=0}^{n-1} w_i \cdot b_i.$$

$$r_I = \frac{C \cdot \|A\|}{n - C} = \frac{\|A\| \cdot \sum_{i=0}^{n-1} w_i \cdot b_i}{\|A\| \cdot \sum_{i=0}^{n-1} w_i} = \sum_{i=0}^{n-1}\frac{w_i}{\sum_{i=0}^{n-1} w_i} \cdot b_i = \sum_{i=0}^{n-1} \bar{w}_i \cdot b_i.$$

$(B_{(n-1)})$.

This result is the Simple SCSL generalised for the case where the primary membership is discretised into more than 2 points. We call it the Interval Solitary Collapsed Slice Lemma (Interval SCSL):

Lemma 4.2 (Interval Solitary Collapsed Slice Lemma)**.** *Let $\tilde{F}$ be the interval FOU formed by (upwardly) blurring the membership function of a type-1 fuzzy set (A) at a single domain value x_I, to create a vertical slice which is an interval as opposed to a single point ($\mu_A(x_I)$), discretised with $n (n \geq 2)$ primary membership grades $B_0^I, B_1^I, B_2^I, \ldots, B_{n-1}^I$ at distances b_0^I $(= 0)$, b_1^I, b_2^I, $\ldots$, b_{n-1}^I from $\mu_L(x_I)$. Then R, the RES of $\tilde{F}$, has a membership function such that*

$$\mu_R(x_j) = \begin{cases} \mu_A(x_I) + r_I & \text{if } j = I, \\ \mu_A(x_j) & \text{otherwise,} \end{cases}$$

where $r_I = \sum_{i=0}^{n-1} \bar{w}_i^I \cdot b_i^I$, $\bar{w}_i^I = \dfrac{w_i^I}{\sum_{i=0}^{n-1} w_i^I}$, *and* $w_i^I = \dfrac{1}{\|A\| + b_i^I}$.

4.4.2 The Interval SCSL as a Generalisation of the Simple SCSL

In the following we show that Lemma 4.2, the Interval Solitary Collapsed Slice Lemma, generalises Lemma 4.1, the Simple Solitary Collapsed Slice Lemma.

In the simple case we have at a single domain value x_I, $n = 2$ and primary membership grades B_0^I, B_1^I at distances $b_0^I = 0$, $b_1^I = b_I = \mu_U(x_I) - \mu_L(x_I)$ from $\mu_L(x_I)$. In this case, we have:

$$w_0^I = \frac{1}{\|A\| + b_0^I} = \frac{1}{\|A\|},$$

$$w_1^I = \frac{1}{\|A\| + b_I}, \text{ and}$$

$$w_0^I + w_1^I = \frac{2\|A\| + b_I}{\|A\|(\|A\| + b_I)}.$$

$$r_I = \frac{w_0^I b_0^I + w_1^I b_1^I}{w_0^I + w_1^I} = \frac{\frac{1}{\|A\| + b_I} b_I}{\frac{2\|A\| + b_I}{\|A\|(\|A\| + b_I)}} = \frac{\|A\| b_I}{2\|A\| + b_I}.$$

4.4.3 Interval RESA

Corresponding to the Interval SCSL, the Interval Representative Embedded Set Approximation is obtained following a similar line of reasoning to that employed in Subsection 4.3.2:

Theorem 4.2 (Interval Representative Embedded Set Approximation)**.** *Let $\tilde{F}$ be an interval type-2 fuzzy set with lower and upper membership functions, L and U. Let us assume that the domain of $\tilde{F}$ is discretised into N points $x_1,\dots,x_N$, with associated primary memberships J_{x_I} discretised into n ($n>2$) primary membership grades $B_0^I, B_1^I, B_2^I, \dots, B_{n-1}^I$ at distances b_0^I $(=0)$, b_1^I, b_2^I, …, b_{n-1}^I $(=\mu_U(x_I)-\mu_L(x_I))$ from $\mu_L(x_I)$. The membership function of the* representative embedded set R *approximates to:*

$$\mu_R(x_I) \approx \mu_L(x_I) + r_I \ \forall I = 1,\dots,N,$$

where $\forall I : r_I = \sum_{i=0}^{n-1} \bar{w}_i^I \cdot b_i^I$; $\bar{w}_i^I = \dfrac{w_i^I}{\sum_{i=0}^{n-1} w_i^I}$; $w_i^I = \dfrac{1}{\|L\| + \Re_{I-1} + b_i^I}$; *and* $\Re_{I-1} = \sum_{k=0}^{I-1} r_k$ *with* $\Re_0 = 0$.

Result 4.3 provides the formula for calculating the approximate defuzzified value of an interval type-2 fuzzy set using the collapsing method.

Theorem 4.3 (Defuzzified Value of a Discretised Interval Type-2 FS)**.** *Let $\tilde{F}$ be an interval type-2 fuzzy set with lower and upper membership functions, L and U. Let us assume that the domain of $\tilde{F}$ is discretised into N points $x_1,\dots,x_N$, with associated primary memberships J_{x_I} discretised into n ($n>2$) primary membership grades $B_0^I, B_1^I, B_2^I, \dots, B_{n-1}^I$ at distances b_0^I $(=0)$, b_1^I, b_2^I, …, b_{n-1}^I $(=\mu_U(x_I)-\mu_L(x_I))$ from $\mu_L(x_I)$. The defuzzified value of $\tilde{F}$ approximates to:*

$$X_{\tilde{F}} \approx X_L + \frac{\sum_{I=1}^{N} r_I\,(x_I - X_L)}{\|L\| + \sum_{I=1}^{N} r_I},$$

where

$\forall I : r_I = \sum_{i=0}^{n-1} \bar{w}_i^I \cdot b_i^I$; $\bar{w}_i^I = \dfrac{w_i^I}{\sum_{i=0}^{n-1} w_i^I}$; $w_i^I = \dfrac{1}{\|L\| + \Re_{I-1} + b_i^I}$; $\Re_{I-1} = \sum_{k=0}^{I-1} r_k$ *with* $\Re_0 = 0$; *and* X_L *is the centroid of L.*

4.5 Variants of the Collapsing Method

The calculation of the Simple Representative Embedded Set Approximation is an iterative procedure; Theorem 4.1 (the Simple RESA) is the collapsing formula. The proof of this theorem presented *a version* of collapsing — the most intuitive variant, whereby the slices are collapsed in the order of increasing domain value ($x=0$ to $x=1$). We term this *collapsing forward*. However slice collapse may be performed in any slice order giving slightly different RESAs. If the domain of the interval type-2 fuzzy set is discretised into m vertical slices, the number of permutations of these slices is $m!$ [10, page 139]. Therefore there must be $m!$ RESAs obtainable by varying the order of slice collapse. The question that then presents itself is, "Does the order in which the slices are collapsed affect the accuracy of the method?" This question is investigated in this section [18].

There are four fundamental variants, which we term *forward*, *backward*, *outward* and *inward*. Inward and outward may each be approached in two different ways. For the inward variant, slice collapse might start from the left (*inward left*) or from the right (*inward right*). The last slice to be collapsed is in the middle. For the outward variant, the first slice collapsed is in the middle[4], but the second slice may be to the right (*outward right*) or to the left (*outward left*). Added to these, there are three composite variants:

Collapsing forward-backward which is the mean of the defuzzified values found by collapsing forward and collapsing backward,

collapsing inward right-left which is the mean of the defuzzified values found by collapsing inward right and collapsing inward left, and

collapsing outward right-left which is the mean of the defuzzified values found by collapsing outward right and collapsing outward left.

4.5.1 Experimental Comparison of Collapsing Variants

Our methodology was to run different collapsing variants against each other to see which gave the most accurate results. For this purpose three interval test sets with known defuzzified values were employed:

Symmetric Horizontal Test Set The lower membership function is the line $y = 0.2$; the upper membership function the line $y = 0.8$. The shape of this test set may be described as a horizontal stripe. The symmetry of this set tells us that its defuzzified value is 0.5. This set is depicted in Figure 4.11.

Symmetric Triangular Test Set This is a normal test set. The lower and upper membership functions are both triangular in shape, both with vertices at $(0.4, 1)$. The symmetry of this set reveals its defuzzified value to be 0.4. Figure 4.12 is a graphical representation of this test set.

Asymmetric Gaussian Test Set This test set was deliberately designed to be asymmetrical, and hence a more realistic simulation of an FIS aggregated set. Both the lower and upper membership functions are Gaussian. As this set has no symmetry, exhaustive defuzzification (Subsection 2.3.1) had to be employed to determine the benchmark defuzzified value, which, as would be expected, varies slightly with the degree of discretisation. Owing to the limitations of computers the exhaustive technique only works for 21 slices or fewer. Figure 4.13 depicts this test set.

[4] We always employ an odd number of slices, giving a determinate middle slice.

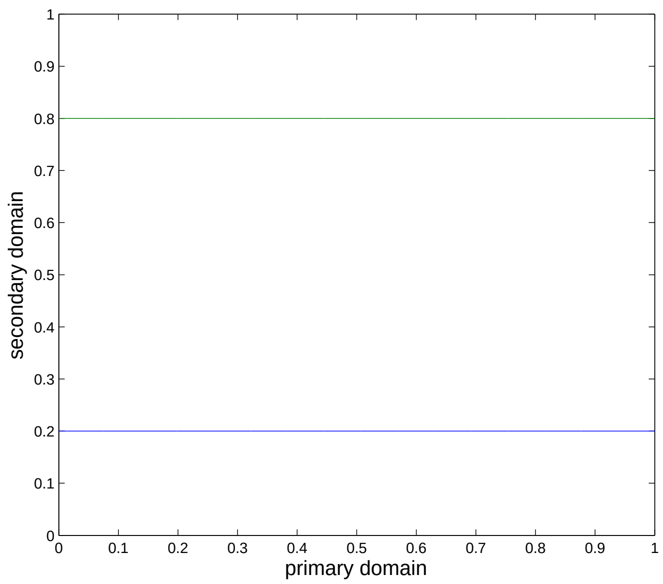


Fig. 4.11. Horizontal test set.

A preliminary set of tests was performed on the fundamental variants: forward, backward, inward, outward, and the composite variant forward-backward. Following these tests further tests were performed on the two best performing variants.

Preliminary Tests Table 4.1 gives the results for the horizontal test set; Table 4.2 gives the associated errors. Table 4.3 shows the triangular test set results, and Table 4.4 the errors. The defuzzification results for the Gaussian test set are shown in Table 4.5, with the errors in Table 4.6. For all three test sets, the best performing variant was outward, followed by inward, then forward and backward. For the symmetrical sets (horizontal and triangular), evidence of the experimental work carried out suggests that the errors of collapsing forward were equal and opposite to those of collapsing backward. Therefore in these cases we would expect collapsing forward-backward to give exact results. This has been confirmed by experiments. For the Gaussian test set, backward performed more poorly than forward. In this case the composite of forward-backward performed worse than forward, though better than backward.

Further Tests The outward variant may be performed in two ways, outward right and outward left. Collapsing outward right-left is the mean of collapsing right and collapsing left. The results and associated errors for the three versions of the outward variant as applied to the three test sets are shown in Tables 4.7 to 4.9.

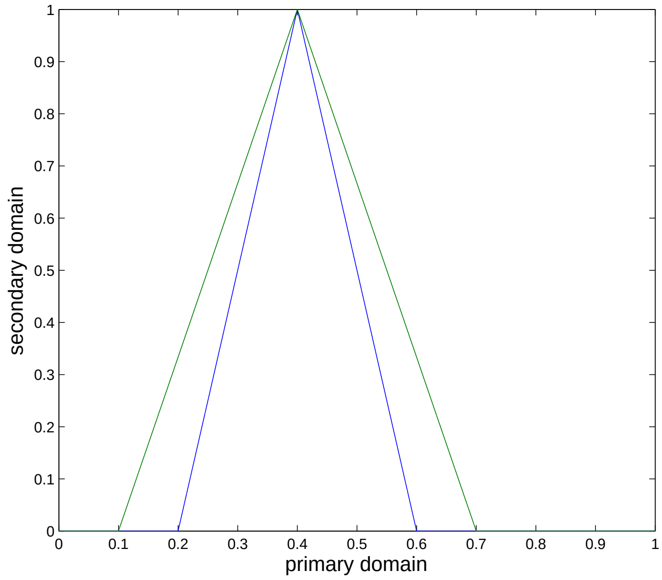


Fig. 4.12. Triangular test set.

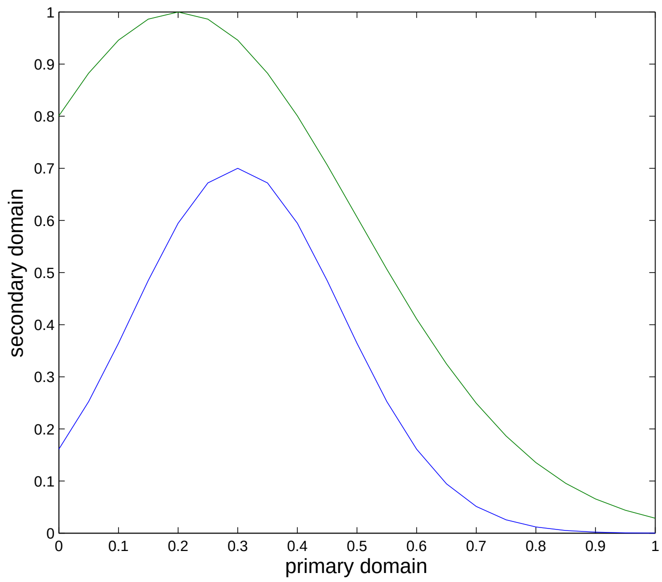


Fig. 4.13. Gaussian test set.

For the symmetrical horizontal test set, outward right and outward left gave rise to equal but opposite errors. For the composite outward right-left, these errors cancelled to zero.

The triangular test set, though symmetrical, was not placed symmetrically about $x = 0.5$. The errors of collapsing right and collapsing left were of equal sign and either equal or very close in quantity. When the errors were not equal, those of outward left were marginally smaller than those of outward right.

For the Gaussian test set, the errors were all of negative sign. At all degrees of discretisation, outward left gave the best results, outward right-left gave the second best results, and outward right the worst.

For two of the three test sets outward left outperformed outward right. Our conjecture is that the position of the centroid is an important factor affecting which performs better out of outward right and outward left. This topic requires further research using a wider range of test sets, but for now we conclude that the optimum strategy is collapsing outward right-left.

In this section we have demonstrated experimentally that the most accurate variant of the collapsing defuzzifier is the composite Collapsing Outward Right-Left (CORL). We shall use CORL in the comparative tests described in Chapter 5.

4.5.2 Why is Outward the Most Accurate Variant?

This explanation is based on the symmetrical horizontal test set. As each slice is collapsed, $\|L\| + \sum_{j=1}^{j=i-1} r_j$ in both the numerator and denominator of the collapsing formula (Equation (4.1)) increases, which means that as the collapse progresses, the r_i for each collapsed slice i is a closer approximation to $\frac{1}{2}b_i$, i.e. half the 'blur' term. Thus with every successive collapsed slice, the RESA tends towards the midline of L and U, as shown in Figure 4.14 for the forward and backward variants.

For the symmetrical horizontal test set, we take the RES to be the midline of L and U for two reasons. Firstly, by symmetry we would expect the RES to be a horizontal line. Secondly, as the number of slices is increased (either as the collapse progresses, or as the degree of discretisation is made finer), the RESA gets closer to the midline of L and U.

Therefore, as the slices are collapsed, the RESA approaches the RES. This means that the earlier slices in the RESA deviate more from the RES than the later ones. To get the best results, the collapse needs to proceed symmetrically. Both the inward and outward variants meet this criterion; the inaccuracies are distributed symmetrically. However the greatest inaccuracy is associated with the first collapsed slice. To achieve maximum accuracy, the ideal place for this first slice to be positioned is centrally, as the effect on the defuzzified value obtained is then minimal. For this reason outward (Figure 4.15) gives a more accurate defuzzified values than inward. We would expect the same reasoning to apply to all type-2 fuzzy test sets. However further investigation,

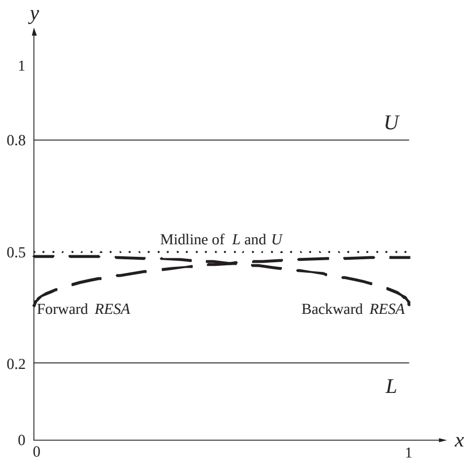


Fig. 4.14. Forward RESA and Backward RESA.

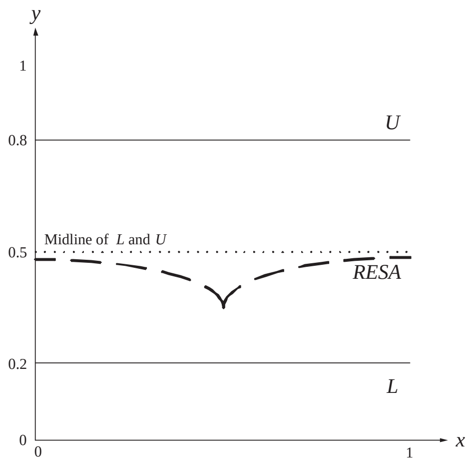


Fig. 4.15. Outward RESA.

using radically contrasting test sets, is planned in the future.

4.6 Continuous Type-2 Fuzzy Sets Approached through Finer Discretisation

Although this thesis primarily concerns *discretised* type-2 fuzzy sets, light may be shed on the continuous case by investigating what happens as the degree of discretisation is reduced. For interval type-2 fuzzy sets it is easily demonstrated [19] that in the continuous case the RESA and Nie-Tan Set (NTS) are identical: The Nie-Tan method computes $\mu_N(x_i) = \frac{1}{2}(\mu_L(x_i) + \mu_U(x_i))$. As the degree of discretisation becomes finer, $||L||$ in the collapsing formula (Equation 4.1) tends to infinity, making the expression $\|L\| + \sum_{j=1}^{j=i-1} r_j$ also tend to infinity. r_i therefore increases, with $\frac{b_i}{2}$ as its upper bound. Thus in the continuous case the collapsing defuzzifier computes

$$\mu_R(x_i) = \mu_L(x_i) + \frac{1}{2}(\mu_U(x_i) - \mu_L(x_i)) = \mu_L(x_i) + \frac{1}{2}\mu_U(x_i) - \frac{1}{2}\mu_L(x_i) = \frac{1}{2}(\mu_L(x_i) + \mu_U(x_i)).$$

This means that in the continuous case the GCCD and Nie-Tan Method are equivalent as they compute the same type-1 fuzzy set. Evidence from [14, 15] and Tables C.2, E.2, F.2, G.2 and H.2 shows that the GCCD and the Nie-Tan defuzzified values both approach the exhaustive defuzzified value as discretisation becomes finer. However this trend is not apparent in Table D.2. This is probably attributable to discretisation effects. We believe that the TRS and the NTS give the same defuzzified value in the continuous case, but this conjecture is as yet unproven. If, in the continuous case, the TRS and NTS are equivalent (in the sense of defuzzifying to the same value),

DEGREE OF DISCRETISATION	COLLAPSING FORWARD	COLLAPSING BACKWARD	COLLAPSING INWARD	COLLAPSING OUTWARD
0.1	0.5038320922	0.4961679078	0.4993086838	0.4995891494
0.05	0.5019998917	0.4980001083	0.4998049953	0.4998891777
0.02	0.5008177226	0.4991822774	0.4999665227	0.4999815068
0.01	0.5004115350	0.4995884650	0.4999914377	0.4999953154
0.005	0.5002064040	0.4997935960	0.4999978353	0.4999988213
0.002	0.5000827100	0.4999172900	0.4999996513	0.4999998107
0.001	0.5000413793	0.4999586207	0.4999999126	0.4999999526
0.0001	0.5000041401	0.4999958599	0.4999999991	0.4999999995
0.00001	0.5000004140	0.4999995860	0.5000000000	0.5000000000

Table 4.1. Defuzzified values obtained by collapsing the symmetrical horizontal test set. By symmetry the defuzzified value is 0.5.

then since we have proved that the continuous RESA and the continuous NTS are the same type-1 set, then the continuous RESA and the continuous TRS give the same defuzzified value i.e. the continuous RESA *is* the RES (Section 4.2).

Summary

This chapter introduced the Greenfield-Chiclana Collapsing Defuzzifier. The formulae for its associated concept, the RESA, was derived in the simple and interval cases. The most accurate variant of the collapsing method was shown experimentally to be CORL. The GCCD was originally envisaged as a generalised method, the simple and interval RESAs being stages towards the development of the generalised RESA. However the imperative for generalising the RESA was obviated by Liu's α-planes representation [41] (published at about the same time as the collapsing method), which generalises *any* interval method.

The next chapter (Part III, Chapter 5) reports on experimental comparisons of the various methods for speed and accuracy.

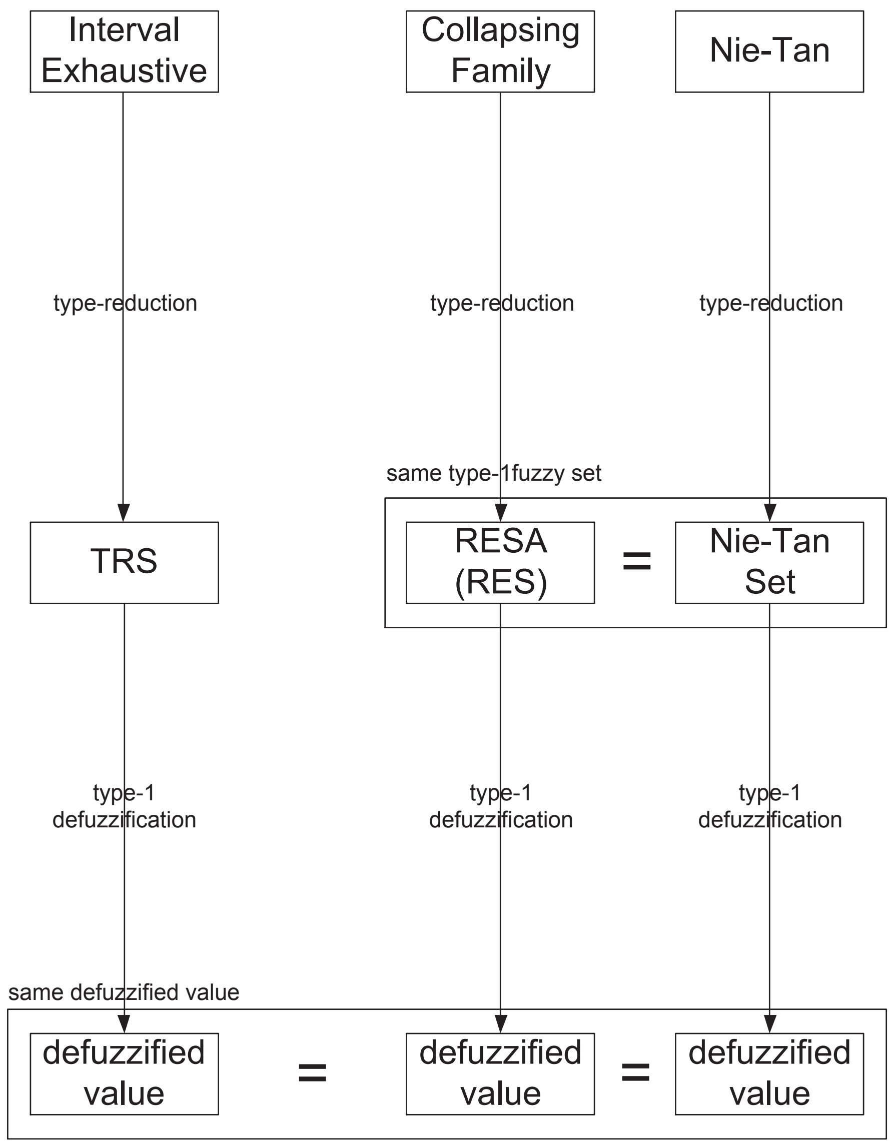


Fig. 4.16. Relationships between the interval methods in the continuous case.

DEGREE OF DISCRETISATION	COLLAPSING FORWARD	COLLAPSING BACKWARD	COLLAPSING INWARD	COLLAPSING OUTWARD
0.1	0.0038320922	-0.0038320922	-0.0006913162	-0.0004108506
0.05	0.0019998917	-0.0019998917	-0.0001950047	-0.0001108223
0.02	0.0008177226	-0.0008177226	-0.0000334773	-0.0000184932
0.01	0.0004115350	-0.0004115350	-0.0000085623	-0.0000046846
0.005	0.0002064040	-0.0002064040	-0.0000021647	-0.0000011787
0.002	0.0000827100	-0.0000827100	-0.0000003487	-0.0000001893
0.001	0.0000413793	-0.0000413793	-0.0000000874	-0.0000000474
0.0001	0.0000041401	-0.0000041401	-0.0000000009	-0.0000000005
0.00001	0.0000004140	-0.0000004140	0.0000000000	0.0000000000

Table 4.2. Errors incurred in collapsing the symmetrical horizontal test set. Error = collapsing defuzzified value − known defuzzified value of 0.5.

DEGREE OF DISCRETISATION	COLLAPSING FORWARD	COLLAPSING BACKWARD	COLLAPSING INWARD	COLLAPSING OUTWARD
0.1	0.4001359091	0.3998640909	0.4001131909	0.3998916916
0.05	0.4000597189	0.3999402811	0.4000498280	0.3999505451
0.02	0.4000230806	0.3999769194	0.4000195457	0.3999808751
0.01	0.4000115326	0.3999884674	0.4000098170	0.3999904744
0.005	0.4000057773	0.3999942227	0.4000049299	0.3999952381
0.002	0.4000023153	0.3999976847	0.4000019784	0.3999980943
0.001	0.4000011585	0.3999988415	0.4000009904	0.3999990469
0.0001	0.4000001159	0.3999998841	0.4000000992	0.3999999047
0.00001	0.4000000116	0.3999999884	0.4000000099	0.3999999905

Table 4.3. Defuzzified values obtained by collapsing the symmetrical triangular test set.

DEGREE OF DISCRETISATION	COLLAPSING FORWARD	COLLAPSING BACKWARD	COLLAPSING INWARD	COLLAPSING OUTWARD
0.1	0.0001359091	-0.0001359091	0.0001131909	-0.0001083084
0.05	0.0000597189	-0.0000597189	0.0000498280	-0.0000494549
0.02	0.0000230806	-0.0000230806	0.0000195457	-0.0000191249
0.01	0.0000115326	-0.0000115326	0.0000098170	-0.0000095256
0.005	0.0000057773	-0.0000057773	0.0000049299	-0.0000047619
0.002	0.0000023153	-0.0000023153	0.0000019784	-0.0000019057
0.001	0.0000011585	-0.0000011585	0.0000009904	-0.0000009531
0.0001	0.0000001159	-0.0000001159	0.0000000992	-0.0000000953
0.00001	0.0000000116	-0.0000000116	0.0000000099	-0.0000000095

Table 4.4. Errors incurred in collapsing the symmetrical triangular test set. Error = collapsing defuzzified value − known defuzzified value of 0.4.

DEG. OF DISC.	EXHAUSTIVE DEFUZZIFIED VALUE	COLLAPSING FORWARD	COLLAPSING BACKWARD	COLLAPSING INWARD	COLLAPSING OUTWARD	COLLAPSING FORWARD BACKWARD
0.5	0.2899142309	0.2947300898	0.4090097593	0.2940174555	0.2884666838	0.3518699246
0.25	0.2906756945	0.2925398791	0.3712146394	0.2923170555	0.2900969651	0.3318772592
0.125	0.3043413255	0.3052741624	0.3526142975	0.3051864643	0.3041285835	0.3289442299
0.1	0.3074987724	0.3082433183	0.3477346996	0.3081777251	0.3073450280	0.3279890090
0.0625	0.3125118626	0.3129728510	0.3393585073	0.3129362993	0.3124323840	0.3261656791
0.05	0.3142610070	0.3146278507	0.3362363800	0.3145998182	0.3142020426	0.3254321154

Table 4.5. Defuzzified values obtained by collapsing the Gaussian test set.

DEG. OF DISC.	EXHAUSTIVE DEFUZZIFIED VALUE	COLLAPSING FORWARD	COLLAPSING BACKWARD	COLLAPSING INWARD	COLLAPSING OUTWARD	COLLAPSING FORWARD BACKWARD
0.5	0.2899142309	0.0048158589	0.1190955284	0.0041032246	-0.0014475471	0.0619556937
0.25	0.2906756945	0.0018641846	0.0805389449	0.0016413610	-0.0005787294	0.0412015647
0.125	0.3043413255	0.0009328369	0.0482729720	0.0008451388	-0.0002127420	0.0246029044
0.1	0.3074987724	0.0007445459	0.0402359272	0.0006789527	-0.0001537444	0.0204902366
0.0625	0.3125118626	0.0004609884	0.0268466447	0.0004244367	-0.0000794786	0.0136538165
0.05	0.3142610070	0.0003668437	0.0219753730	0.0003388112	-0.0000589644	0.0111711084

Table 4.6. Errors incurred in collapsing the Gaussian test set. Error = collapsing defuzzified value − exhaustive defuzzified value.

	COLLAPSING DEFUZZIFIED VALUES			**ERRORS**		
DEGREE OF DISC.	OUTWARD RIGHT	OUTWARD LEFT	OUTWARD RIGHT-LEFT	OUTWARD RIGHT	OUTWARD LEFT	OUTWARD RIGHT-LEFT
0.1	0.4995891494	0.5004108506	0.5000000000	-0.0004108506	0.0004108506	0.0000000000
0.05	0.4998891777	0.5001108223	0.5000000000	-0.0001108223	0.0001108223	0.0000000000
0.02	0.4999815068	0.5000184932	0.5000000000	-0.0000184932	0.0000184932	0.0000000000
0.01	0.4999953154	0.5000046846	0.5000000000	-0.0000046846	0.0000046846	0.0000000000
0.005	0.4999988213	0.5000011787	0.5000000000	-0.0000011787	0.0000011787	0.0000000000
0.002	0.4999998107	0.5000001893	0.5000000000	-0.0000001893	0.0000001893	0.0000000000
0.001	0.4999999526	0.5000000474	0.5000000000	-0.0000000474	0.0000000474	0.0000000000
0.0001	0.4999999995	0.5000000005	0.5000000000	-0.0000000005	0.0000000005	0.0000000000
0.00001	0.5000000000	0.5000000000	0.5000000000	0.0000000000	0.0000000000	0.0000000000

Table 4.7. Defuzzified values and errors obtained for the symmetrical horizontal test set, collapsed outward. By symmetry, the defuzzified value is 0.5.

	COLLAPSING DEFUZZIFIED VALUES			**ERRORS**		
DEGREE OF DISC.	OUTWARD RIGHT	OUTWARD LEFT	OUTWARD RIGHT-LEFT	OUTWARD RIGHT	OUTWARD LEFT	OUTWARD RIGHT-LEFT
0.1	0.3998916916	0.3998916916	0.3998916916	-0.0001083084	-0.0001083084	-0.0001083084
0.05	0.3999505451	0.3999522908	0.3999514180	-0.0000494549	-0.0000477092	-0.0000485820
0.02	0.3999808751	0.3999812363	0.3999810557	-0.0000191249	-0.0000187637	-0.0000189443
0.01	0.3999904744	0.3999905679	0.3999905212	-0.0000095256	-0.0000094321	-0.0000094788
0.005	0.3999952381	0.3999952617	0.3999966681	-0.0000047619	-0.0000047383	-0.0000047501
0.002	0.3999980943	0.3999980981	0.3999980962	-0.0000019057	-0.0000019019	-0.0000019038
0.001	0.3999990469	0.3999990479	0.3999990474	-0.0000009531	-0.0000009521	-0.0000009526
0.0001	0.3999999047	0.3999999047	0.3999999047	-0.0000000953	-0.0000000953	-0.0000000953
0.00001	0.3999999905	0.3999999905	0.3999999905	-0.0000000095	-0.0000000095	-0.0000000095

Table 4.8. Defuzzified values and errors obtained for the symmetrical triangular test set, collapsed outward. By symmetry, the defuzzified value is 0.4.

		COLLAPSING DEFUZZIFIED VALUES			ERRORS		
DEGREE OF DISC.	DEFUZZIFIED VALUE	OUTWARD RIGHT	OUTWARD LEFT	OUTWARD RIGHT-LEFT	OUTWARD RIGHT	OUTWARD LEFT	OUTWARD RIGHT-LEFT
0.5	0.2899142309	0.2884666838	0.2890645675	0.2887656257	-0.0014475471	-0.0008496634	-0.0011486052
0.25	0.2906756945	0.2900969651	0.2902918203	0.2901943927	-0.0005787294	-0.0003838742	-0.0004813018
0.125	0.3043413255	0.3041285835	0.3041906758	0.3041285835	-0.0002127420	-0.0001506497	-0.0001816959
0.1	0.3074987724	0.3073450280	0.3073862653	0.3073656467	-0.0001537444	-0.0001125071	-0.0001331257
0.0625	0.3125118626	0.3124323840	0.3124493111	0.3124408476	-0.0000794786	-0.0000625515	-0.0000710150
0.05	0.3142610070	0.3142020426	0.3142130418	0.3142075422	-0.0000589644	-0.0000479652	-0.0000534648

Table 4.9. Defuzzified values and errors obtained for the Gaussian test set, collapsed outward.

Part III

EVALUATION OF THE TYPE-2 DEFUZZIFICATION METHODS

Chapter 5

Evaluation of Type-2 Defuzzification Methods

5.1 Comparing and Contrasting the Strategies

In this section the similarities and differences between the strategies presented in Parts I and II are examined.

For three reasons the exhaustive method is in a class of its own:

1. It is the only *precise* method. Inaccuracies are engendered through the process of discretisation, but beyond that, the exhaustive method introduces no further imprecision.

2. It is the only *inefficient* method, indeed its staggering inefficiency was the motivation for the research into alternative methods reported in this thesis.

3. Owing to its inefficiency, it is the only strategy *not regarded as a practical defuzzification method*. For coarsely discretised test sets, it is used as a benchmark in evaluating the accuracy of other methods.

Throughout this thesis a distinction has been made between interval and generalised methods. Using the α-Plane Representation (Subsection 2.3.4) it is possible to generalise any interval method. But for some interval methods there is more than one route to the generalised form. For example, the Nie-Tan Method may be generalised via the α-Plane Representation, or as VSCTR. And of course generalised methods may always be applied to interval sets, though techniques *designed* as interval methods (Section 2.4) permit considerable reduction in computational complexity as they disregard the variable secondary membership grade.

The type-1 OWA based method does not rely on the repeated application of any interval method of defuzzification, unlike generalised methods reliant on the α-Plane Representation. However, as it has yet to be implemented, it has been left out of the comparative study reported in this thesis.

Another link between methods is that one may be a simplified version of another. This applies in two cases. The sampling method is a cut-down version of the exhaustive method as it deals with only a sample of the embedded sets, ignoring the majority. The KMIP family of methods are cut down versions of the interval exhaustive method since the type-reduced set is specified by only two embedded sets — those corresponding to the endpoints of the interval. Figure 5.1 shows the relationships of generalisation and simplification between the methods.

All of the methods discussed in this thesis perform type-reduction, i.e. they create a type-1 fuzzy set from a type-2 fuzzy set. However not all the methods type-reduce to the TRS as defined by Mendel [43]. The techniques that type-reduce to a type-1 fuzzy set other than the TRS are the Nie-Tan Method, VSCTR, the GCCD in all its variants, and the α-Planes Method.

The KMIP and associated methods such as EIASC are search algorithms; they search for the two embedded sets which correspond to the endpoints of the TRS interval. None of the other strategies involve searching.

All the methods apart from the one-pass GCCD are symmetric. With one-pass collapsing, the point where the collapse begins, and the direction in which it travels, make a difference to the defuzzified value obtained by the method (Section 4.5). However the two-pass CORL *is* symmetric.

Most of the methods are expressed as closed formulae. The exceptions are those of the KMIP family and the GCCD family. The KMIP family find the endpoints of the TRS interval through iterative algorithms and the GCCD family employ an iterative formula to find the RESA and hence calculate the defuzzified value. However the GCCD may be written in the form of a closed formula (Theorem 4.3).

Table 5.1 summarises the characteristics of the various methods.

5.2 The Methods' Association with the Concept of Embedded Sets

The Wavy-Slice Representation Theorem (Subsection 2.2) states that a type-2 fuzzy set can be represented as the union of its type-2 embedded sets. The defuzzification methods of Chapter 2 may be split into three groups according to their employment of embedded sets:

The algorithms explicitly refer to embedded sets: The exhaustive method processes every embedded set in a type-2 fuzzy set. The sampling method (Chapter 3) processes a sample of the embedded sets. The KMIP family search for the two embedded sets which correspond to the endpoints of the TRS interval.

The embedded set concept is used in the derivations of the algorithms: The RESA, which is central to the collapsing method (Chapter 4), employs the concept of embedded sets in its proof. The Wu-Mendel Approximation (Subsection 2.4.2) finds approximations to the endpoints of the TRS interval; these endpoints are associated with embedded sets.

Embedded sets have no influence at all on the algorithms: The concept of embedded sets has no bearing whatsoever on the Nie-Tan Method, its generalisation, VSCTR and the type-1

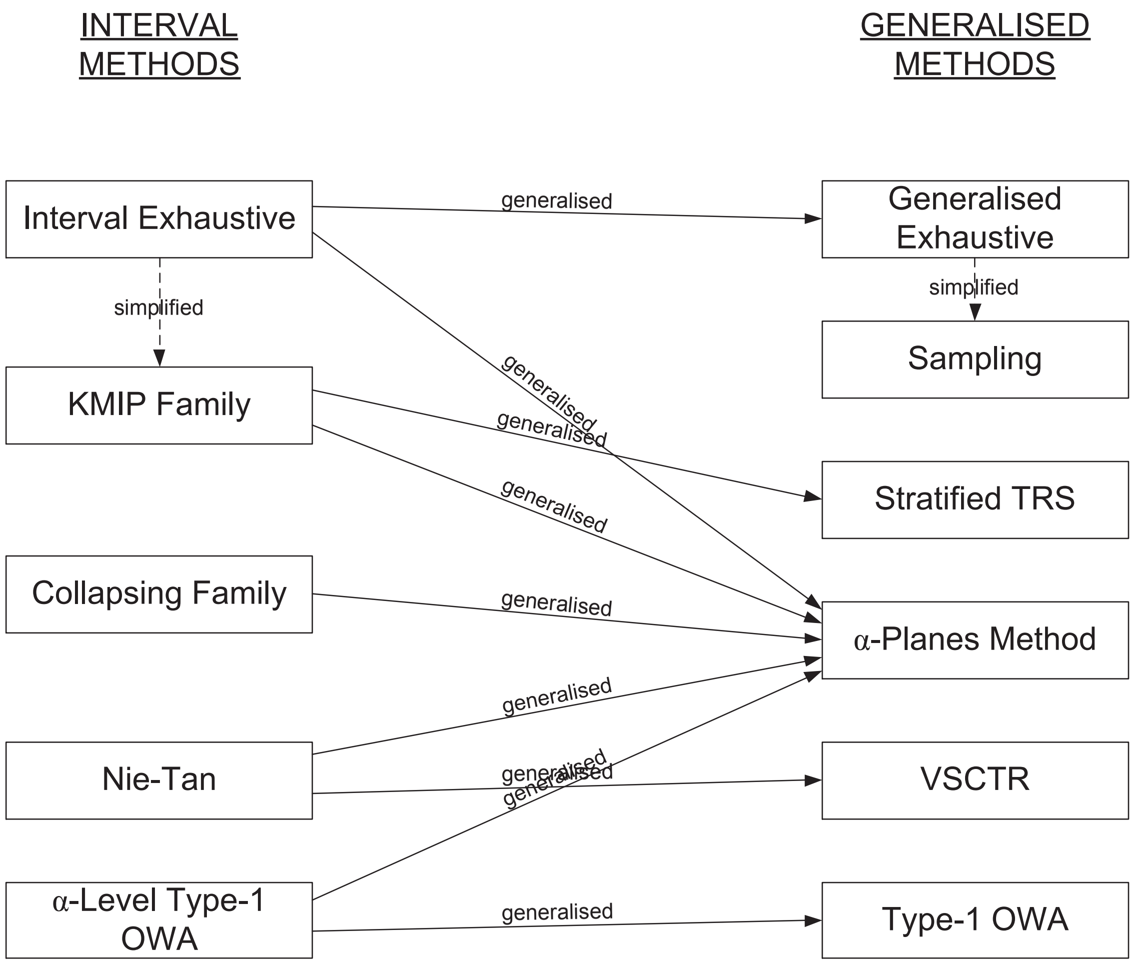


Fig. 5.1. Relationships of generalisation and simplification between the major defuzzification methods.

METHOD	METHOD BY DESIGN?	GENERALISED?	SIMPLIFICATION?	PRECISE?	EFFICIENT?	TYPE-REDUCER?	TYPE-RED. TO TRS?	SEARCH ALGORITHM?	SYMMETRIC?	ITERATIVE?	STAND-ALONE?
Exhaustive	no	yes	no	yes	no	yes	yes	no	yes	no	yes
KMIP	yes	no	yes	no	yes	yes	yes	yes	yes	yes	yes
Wu-Mendel	yes	no	no	no	yes	yes	yes	no	yes	no	yes
Sampling	yes	yes	yes	no	yes	yes	yes	no	yes	no	yes
GCCD (1-pass)	yes	no	no	no	yes	yes	no	no	no	yes	yes
Nie-Tan	yes	no	no	no	yes	yes	no	no	yes	no	yes
VSCTR	yes	yes	no	no	yes	yes	no	no	yes	no	yes
CORL (2-pass)	yes	no	no	no	yes	yes	no	no	yes	yes	yes
EIASC	yes	no	no	no	yes	yes	yes	yes	yes	yes	yes
α-Planes	yes	yes	no	no	yes	yes	no	no	yes	no	no

Table 5.1. Comparing and contrasting the major defuzzification methods for discretised type-2 fuzzy sets.

OWA based method. No mathematical justification has been given for the supposition that the Nie-Tan and VSCTR methods give the same defuzzified value as the exhaustive method, although in Section 4.6 experimental evidence is presented which supports this conjecture.

Figure 5.2 summarises this three-way classification in the form of a Venn Diagram.

5.3 Experimental Evaluation of the Defuzzification Techniques

In the last three chapters several alternative interval and generalised defuzzification strategies have been presented. But which should the application developer choose? In the remainder of this chapter we report on experiments which evaluate the methods by testing them for accuracy and efficiency. The test runs were performed in isolation from the rest of the FIS, on specially created test sets, six interval and six generalised.

For accuracy evaluation, the error of a test run was calculated by finding the difference between the resultant defuzzified value and the benchmark exhaustive defuzzified value for the test set in question. However, in evaluating the α-planes method, the sign of the error was tabulated, as it sheds light on the convergence of the results as the number of α-planes is increased.

The defuzzification methods were coded in Matlab^{TM} and tested on a laptop with an AMD Turion II Neo K645 CPU, a clock speed of 1.6 GHz, and a 4096MB 1333MHz 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.

5.3.1 Interval Defuzzification Techniques

Interval Test Sets Six interval type-2 fuzzy sets were prepared: M, N, S, U, W and X. Test sets M and X were taken from Liu's 2008 *Information Sciences* paper on the α-Planes Representation [41, pages 2230 – 2233], and the remaining four devised so that the group as a whole exhibited a wide range of features (Table 5.2). Graphical representations of the interval test sets are to be found in Appendices C to H. Each test set was discretised into 5, 9, 11, 17, 21, 51, 101, 1001, 10001, and 100001 vertical slices[1].

Methodology for Interval Methods Comparison The GCCD is best thought of as a family of methods as there are a number of variants (Section 4.5). It has been demonstrated practically and theoretically that the two-pass CORL is the most accurate variant (Subsection 4.5.1). Algorithms

[1] It is convenient to use odd numbers of vertical slices so that there is always a middle slice to use as the starting point of CORL.

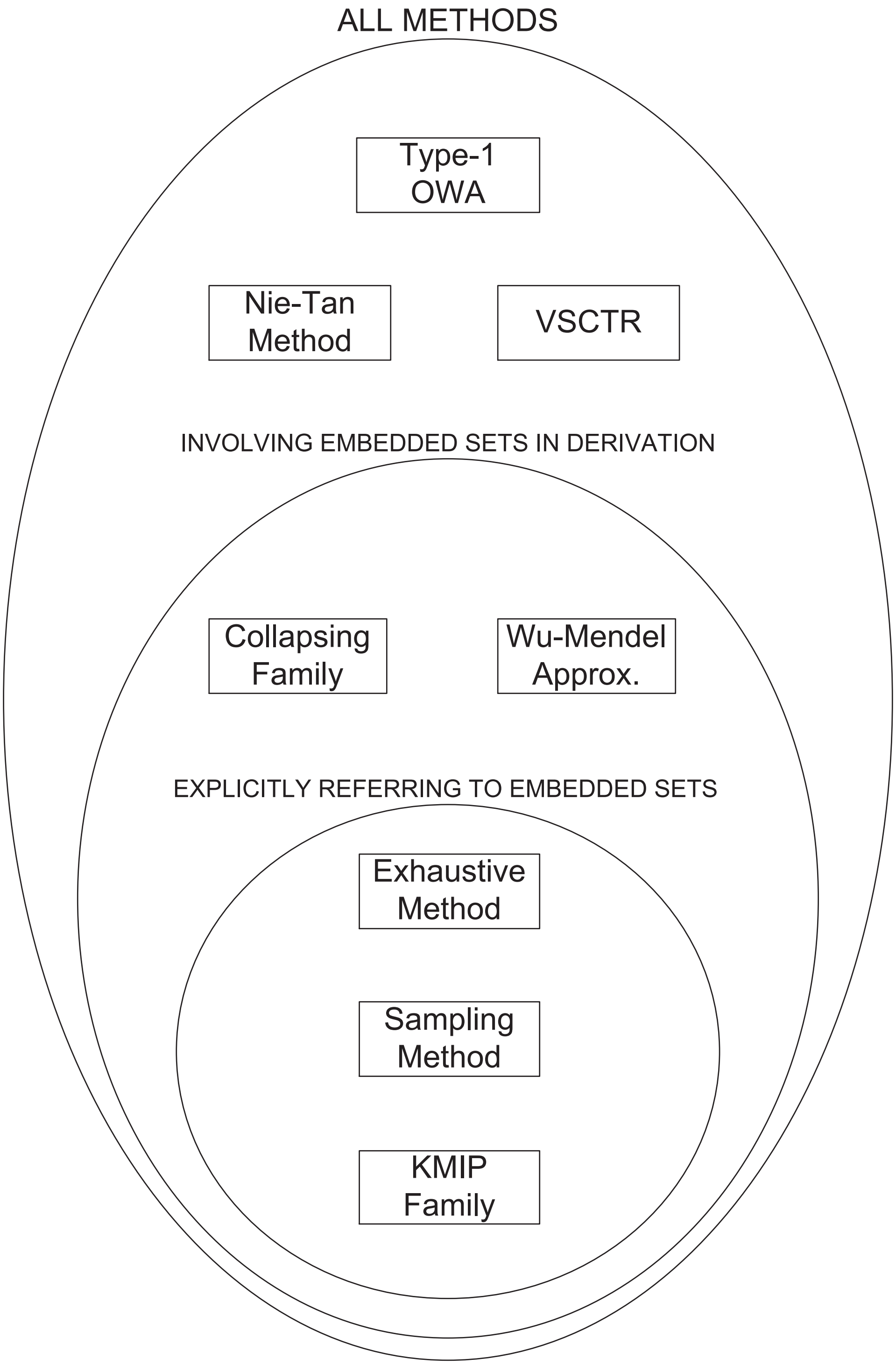


Fig. 5.2. Methods' relationship to the concept of embedded sets.

FEATURE	M	N	S	U	W	X
Symmetrical	no	yes	no	no	no	no
Extreme (low or high) defuzzified value	no	yes	yes	no	no	no
FOU with narrow section	no	yes	yes	yes	no	yes
FOU with wide section	no	no	no	yes	yes	yes
$[0,1]$ as support	yes	no	no	yes	yes	no
Normal	yes	yes	yes	yes	no	yes
Lower membership function normal	no	yes	yes	yes	no	no
Complex shape	yes	no	no	no	yes	yes
Piecewise linear lower and upper membership functions	no	yes	no	no	no	yes
Angle in lower and upper membership functions	yes	yes	no	no	no	yes

Table 5.2. Features of the interval test sets.

based on the KMIP form another family (Subsection 2.4.1). In [50], Wu and Nie have shown that the most efficient version of the KMIP is EIASC (Subsection 2.4.1). Accordingly, the experiments reported in this chapter make use of CORL and EIASC.

Each of the six test sets, at each degree of defuzzification, was defuzzified using each of the four methods of defuzzification to be tested, namely CORL, EIASC, the Nie-Tan Method and the Wu-Mendel Approximation. To provide benchmark values for accuracy the test sets were also defuzzified using the interval exhaustive method, though this technique could only be applied to degrees of discretisation higher than 0.05, i.e. sets with 21 vertical slices or less. For the timings, multiple runs were performed (10000 for slices 5 to 17, and 1000 for 21 slices). The multiple run time was divided by the number of runs to give results of greater accuracy than those that would have been obtained from a single run.

The test regime contained an additional feature. For both the exhaustive interval method and EIASC, the endpoints of the TRS interval were noted. This allowed an assessment of whether EIASC was successful in locating the TRS endpoints.

Accuracy of Interval Methods Tables 5.3 to 5.6 show the rankings of the four interval methods in relation to accuracy. Assessment of accuracy was only possible for sets discretised into 5, 9, 11, 17 and 21 vertical slices, as higher numbers of slices were beyond the ability of the benchmark exhaustive method to process. This meant that for accuracy testing there were 5 test runs per test set. To contrast the overall performance of the methods, a weighting of 4 was assigned to first place, 3 to second place, 2 to third place, and 1 to fourth place. This technique is a form of multi-criteria decision making [4, 5, 27, 28] allowing the inconsistent performances of the methods to be reflected in their total scores[2]. Table 5.7 is a summary from which it can be seen that CORL is

[2] or *utility values*

the most accurate method, the Nie-Tan Method the second most accurate, EIASC the third most accurate, and the Wu-Mendel Approximation the least accurate. However the tests show all the methods to be adequate as regards accuracy and efficiency.

POSITION	M	N	S	U	W	X	TOTAL	WEIGHTING	WEIGHTED TOTAL
First	4	1	5	5	5	1	21	4	84
Second	0	1	0	0	0	2	3	3	9
Third	1	3	0	0	0	2	6	2	12
Fourth	0	0	0	0	0	0	0	1	0
								GRAND TOTAL	**105**

Table 5.3. Rankings of CORL in relation to accuracy.

POSITION	M	N	S	U	W	X	TOTAL	WEIGHTING	WEIGHTED TOTAL
First	0	5	0	0	0	1	6	4	24
Second	1	0	1	0	2	1	5	3	15
Third	4	0	2	1	2	3	12	2	24
Fourth	0	0	2	4	1	0	7	1	7
								GRAND TOTAL	**70**

Table 5.4. Rankings of EIASC in relation to accuracy.

Though these experiments show CORL to be the superior method in relation to accuracy, the technique's performance was not strong for every test set. CORL was ranked first for accuracy 100% of the time for test sets S, U and W. For test set M, CORL was ranked first 80% of the time. But for test sets N and X, CORL was ranked first only 20% of the time. Even worse, for set N, the 'first' was in fact a 'first equal' with EIASC and the Nie-Tan Method. What might explain the uneven performance of CORL? One factor that stands out immediately is that the sets that CORL performed well with are smooth, whereas the ones for which it performed badly are spiky. For test set M, which is mostly smooth, but contains a downward spike in both its lower and upper membership function, CORL performed quite well $\left(\frac{4}{5}\right)$ but not as well as possible $\left(\frac{5}{5}\right)$. Reassuringly, even for the spiky test sets, CORL did not perform as badly as possible $\left(\frac{0}{5}\right)$. In most cases where CORL did not perform best, the Nie-Tan Method was most accurate; in the remainder of cases EIASC was the most accurate.

The Nie-Tan Method slightly outperformed EIASC for accuracy. This is surprising since the

POSITION	M	N	S	U	W	X	TOTAL	WEIGHTING	WEIGHTED TOTAL
First	1	3	0	0	0	3	7	4	28
Second	4	1	4	2	1	1	13	3	39
Third	0	1	1	2	3	0	7	2	14
Fourth	0	0	0	1	1	1	3	1	3
								GRAND TOTAL	**84**

Table 5.5. Rankings of the Nie-Tan Method in relation to accuracy.

POSITION	M	N	S	U	W	X	TOTAL	WEIGHTING	WEIGHTED TOTAL
First	0	0	0	0	0	0	0	4	0
Second	0	0	0	3	2	1	6	3	18
Third	0	0	2	2	0	0	4	2	8
Fourth	5	5	3	0	3	4	20	1	20
								GRAND TOTAL	**46**

Table 5.6. Rankings of the Wu-Mendel Approximation in relation to accuracy.

Nie-Tan Method is conceptually very simple, and involves embedded sets neither in its processing nor its derivation. The Wu-Mendel Approximation did not come first on any occasion; its best performance was second, but it usually came last.

METHOD	TOTAL WEIGHTED SCORE
CORL	105
Nie-Tan Method	84
EIASC	70
Wu-Mendel Approximation	46

Table 5.7. Overall performance of the interval test sets in relation to accuracy.

Efficiency of Interval Methods In relation to timing, the fastest method was EIASC, followed by the Nie-Tan method, followed by CORL. The Wu-Mendel Approximation was the slowest. However, even when implemented in Matlab^{TM}, a relatively slow running language, at reasonable degrees of discretisation all the methods were sufficiently fast for practical applications. Tables 5.8 to 5.11 show the rankings of the four interval methods in relation to timing.

POSITION	M	N	S	U	W	X	TOTAL	WEIGHTING	WEIGHTED TOTAL
First	0	0	0	0	0	0	0	4	0
Second	0	0	0	0	0	0	0	3	0
Third	4	5	4	5	5	5	28	2	56
Fourth	1	0	1	0	0	0	2	1	2
								GRAND TOTAL	**58**

Table 5.8. Rankings of CORL in relation to timing.

Performance of EIASC as a Search Algorithm How successful was EIASC was in finding the endpoints of the TRS interval? The relevant test results are to be found in Tables C.4 to H.4. For test sets M, N, U and X, and at all degrees of discretisation, EIASC located the endpoints. For test set S (Table E.4), at all degrees of discretisation, EIASC failed to locate either endpoint, but came close in all of the test runs. For test set W (Table G.4), the results were bizarre! At all degrees of discretisation, EIASC failed to locate the endpoints. Astonishingly, the left endpoint had a higher defuzzified value than the right endpoint for all degrees of discretisation. None of the endpoints were anywhere near those of the interval exhaustive method, even if the EIASC left and right endpoints were interchanged. But the strangest feature of all, given these unexpected results,

POSITION	M	N	S	U	W	X	TOTAL	WEIGHTING	WEIGHTED TOTAL
First	5	5	5	5	5	5	30	4	120
Second	0	0	0	0	0	0	0	3	0
Third	0	0	0	0	0	0	0	2	0
Fourth	0	0	0	0	0	0	0	1	0
								GRAND TOTAL	**120**

Table 5.9. Rankings of EIASC in relation to timing.

POSITION	M	N	S	U	W	X	TOTAL	WEIGHTING	WEIGHTED TOTAL
First	0	0	0	0	0	0	0	4	0
Second	5	5	5	5	5	5	30	3	90
Third	0	0	0	0	0	0	0	2	0
Fourth	0	0	0	0	0	0	0	1	0
								GRAND TOTAL	**90**

Table 5.10. Rankings of the Nie-Tan Method in relation to timing.

POSITION	M	N	S	U	W	X	TOTAL	WEIGHTING	WEIGHTED TOTAL
First	0	0	0	0	0	0	0	4	0
Second	0	0	0	0	0	0	0	3	0
Third	1	0	1	0	0	0	2	2	4
Fourth	4	5	4	5	5	5	28	1	28
								GRAND TOTAL	**32**

Table 5.11. Rankings of the Wu-Mendel Approximation in relation to timing.

METHOD	TOTAL WEIGHTED SCORE
EIASC	120
Nie-Tan Method	90
CORL	58
Wu-Mendel Approximation	32

Table 5.12. Overall performance of the interval test sets in relation to timing.

was that the actual defuzzified values were fairly accurate (as compared with other methods tested on the same test set, and with EIASC tested on other test sets). Further research is necessary to find an explanation for this tantalising situation.

Recommended Interval Method Taking all the interval test results into consideration, it is clear that the Wu-Mendel Approximation has nothing to commend it; not only is it the least accurate technique, it is also the slowest. Of the remaining three methods, CORL is to be recommended. Though not *the fastest* method, it is fast[3] and is clearly the most accurate.

5.3.2 Generalised Defuzzification Techniques

We now move on to testing generalised methods, namely the sampling method, VSCTR and the α-planes method. The exhaustive method is used as the standard of accuracy. In the last subsection CORL was established as the interval defuzzification method of choice. For this reason the α-planes method is tested in conjunction with CORL as the interval defuzzifier.

Generalised Test Sets The initial intention was to include Liu's two generalised type-2 fuzzy test sets [41, pages 2230 – 2233], the correlates of interval test sets M and X. However this was not possible, since 1. for Case A (generalised *M*) the secondary membership functions are derived by a random procedure and therefore cannot be recreated, and 2. in Case B (generalised *X*) the secondary membership functions are too similar to interval membership functions for this set to be of value as a generalised test set. Accordingly six FIS generated generalised type-2 fuzzy test sets were created. These are aggregated sets produced by the inferencing stage of *Fuzzer*, a prototype type-2 FIS [12]. 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[4]. 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 5.13 to 5.15. Table 5.16 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 5 rules and 2 inputs which are tabulated in Table 5.13.

[3] CORL's efficiency suffers from its being a two-pass variant of the GCCD.

[4] 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 (Appendices I to N).

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 4 rules and 3 inputs which are summarized in Table 5.14.

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 4 rules and 3 inputs as tabulated in Table 5.15.

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, 10000, 50000 and 100000,
4. the elite sampling method using sample sizes of 50, 100, 250, 500, 750, 1000, 5000, 10000, 50000 and 100000,
5. the α-planes/CORL method using 3, 5, 9, 11, 21, 51, 101, 1001, 10001 and 100001 α-planes, and
6. the α-planes/Interval Exhaustive method using 3, 5, 9, 11, 21, 51, 101, 1001, 10001 and 100001 α-planes (as an evaluation of the accuracy of the α-planes representation itself)[5].

Accuracy of Generalised Methods With 10 alternative sample sizes for both the sampling and the elite sampling strategies, and 10 different numbers of α-planes for the α-planes/CORL method, it was not always possible to rank the performances of the methods. For each test set a figure showing the ranking hierarchy was produced. For those cases where only one or two results obscured a direct ranking of two methods, these results were ignored to create an approximated hierarchy.

HeaterFIS0.125 The sampling and elite sampling methods both outperformed the α-planes/ CORL method. VSCTR was more accurate than sampling. VSCTR was more precise than elite sampling for sample sizes of 1000 or under; for sample sizes of 5000 or over elite sampling was more accurate than VSCTR. Whether sampling outperformed elite sampling or vice versa depended on the degree of discretisation. Figure 5.3 display the methods' ranking in relation to accuracy.

[5] The lengthy processing times prevented defuzzification using 10001 and 100001 α-planes with test sets HeaterFIS0.0625, PowderFIS0.05 and ShoppingFIS0.05.

INPUTS		OUTPUTS
TEMPERATURE	DATE	HEATING
cold	—	high
—	winter	high
hot	not winter	low
—	spring	medium
—	autumn	medium

Table 5.13. Heater 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 5.14. Washing Powder 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 5.15. Shopping FIS rules.

TEST SET	NORMAL FOU	NORMAL SEC. MF	NARROW FOU	NO. OF EMB. SETS
HeaterFIS0.125	yes	no	no	14580
HeaterFIS0.0625	yes	no	yes	13778100
PowderFIS0.1	yes	no	yes	24300
PowderFIS0.05	yes	yes	yes	3840000
ShoppingFIS0.1	yes	yes	no	312500
ShoppingFIS0.05	yes	yes	yes	3840000

Table 5.16. Features of the generalised test sets.

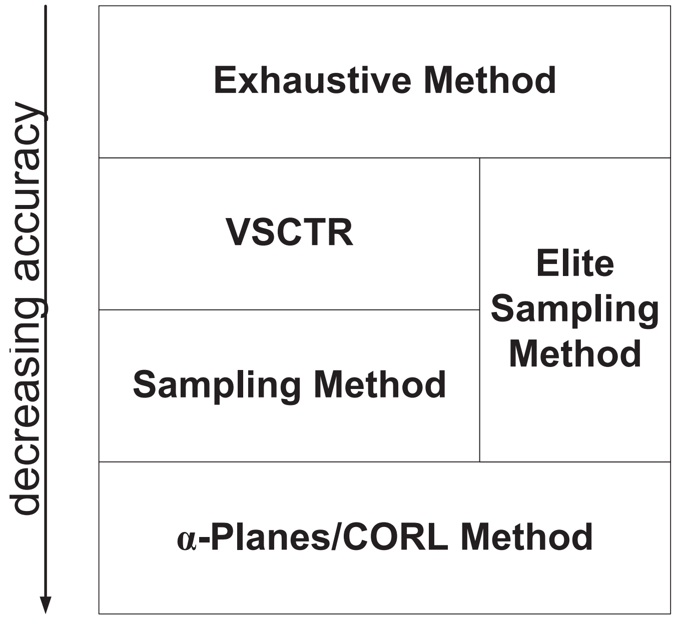


Fig. 5.3. Hierarchy of type-2 defuzzification methods' performance in relation to accuracy, for the Heater0.125 test set. The exhaustive method is used as a benchmark.

HeaterFIS0.0625 The sampling method outperformed VSCTR. VSCTR, the sampling and the elite sampling methods were more accurate than the α-planes/CORL method. The elite sampling technique gave more precise results than VSCTR apart from when sample sizes of 50 and 250 were used. Whether sampling outperformed elite sampling or vice versa depended on the degree of discretisation. Figure 5.4 display the methods' ranking in relation to accuracy.

PowderFIS0.1 The sampling method was more accurate than VSCTR apart from when a sample size of 100 was used. The elite sampling strategy outperformed VSCTR. The α-planes method was more precise than VSCTR apart from when 3 and 5 α-planes were employed. Out of the sampling method, the elite sampling method and the α-planes/CORL technique, the ranking of accuracy depended on the degree of discretisation and the number of α-planes used. Figures 5.5 and 5.6 summarise the methods' ranking in relation to accuracy.

PowderFIS0.05 VSCTR outperformed the sampling method apart from when a sample size of 250 was used. VSCTR was more accurate than the elite sampling method apart from with sample sizes of 50000 and 100000. Whether sampling outperformed elite sampling or vice versa depended on the degree of discretisation. VSCTR, the sampling method, and the elite sampling method outperformed the α-planes/CORL technique. Figures 5.7 and 5.8 display the methods' ranking in relation to accuracy.

ShoppingFIS0.1 VSCTR outperformed the sampling method. VSCTR outperformed the elite sampling method for sample sizes of 100, 250, 750, 1000 and 5000; the elite sampling method was more precise than VSCTR for sample sizes of 50, 500, 10000, 50000 and 100000. VSCTR performed better than the α-planes method apart from when 11 α-planes were used. Out of the

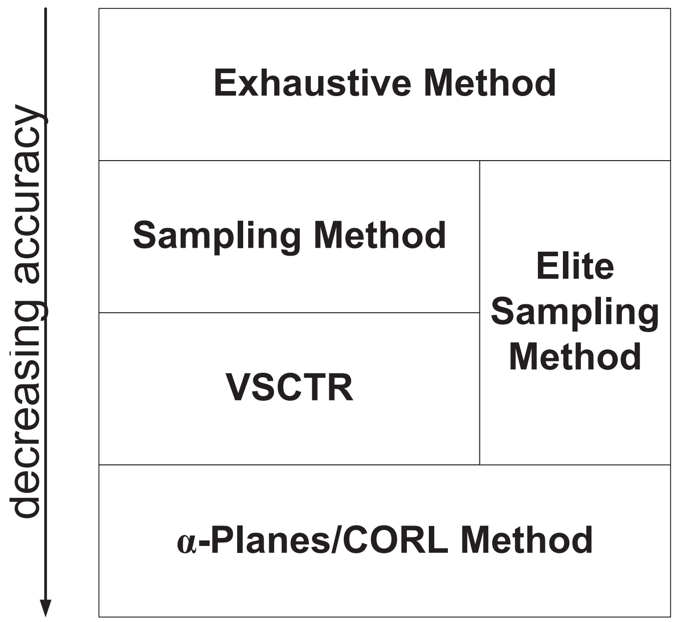


Fig. 5.4. Hierarchy of type-2 defuzzification methods' performance in relation to accuracy, for the Heater0.0625 test set. The exhaustive method is used as a benchmark.

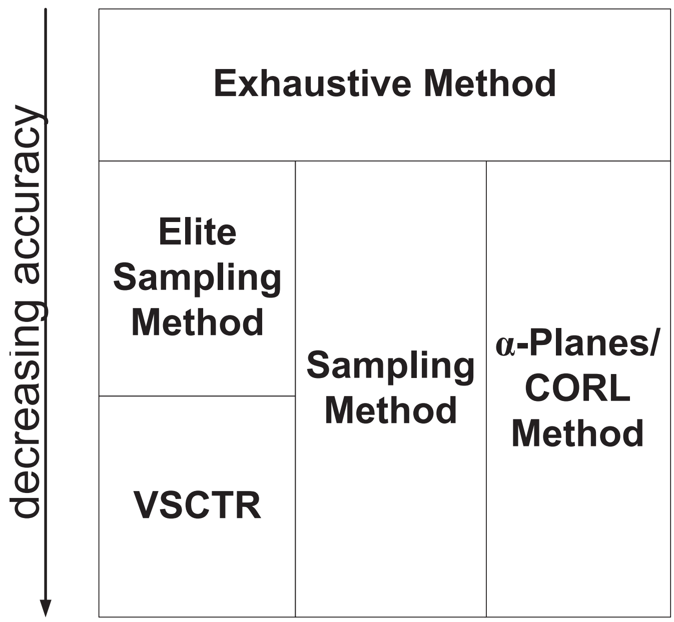


Fig. 5.5. Hierarchy of type-2 defuzzification methods' performance in relation to accuracy, for the Powder0.1 test set. The exhaustive method is used as a benchmark.

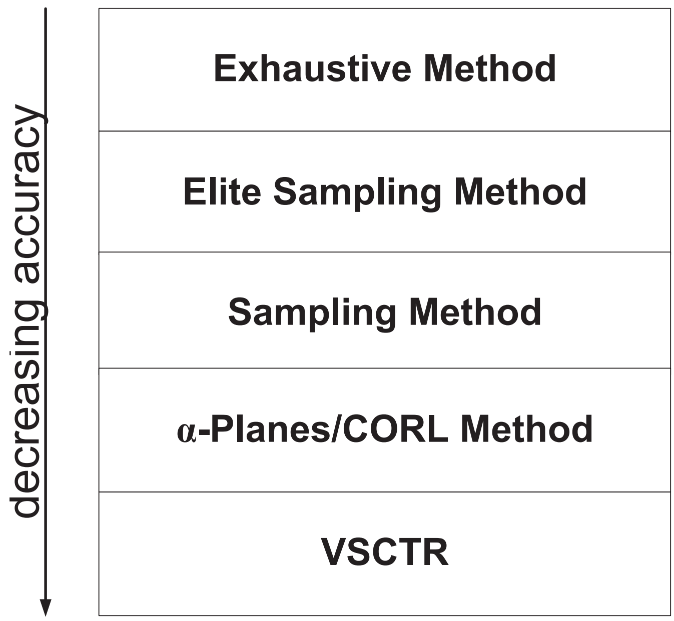


Fig. 5.6. Approximated hierarchy of type-2 defuzzification methods' performance in relation to accuracy, for the Powder0.1 test set. (In this case only one or two results obscured a direct ranking of two methods; these results were ignored.) The exhaustive method is used as a benchmark.

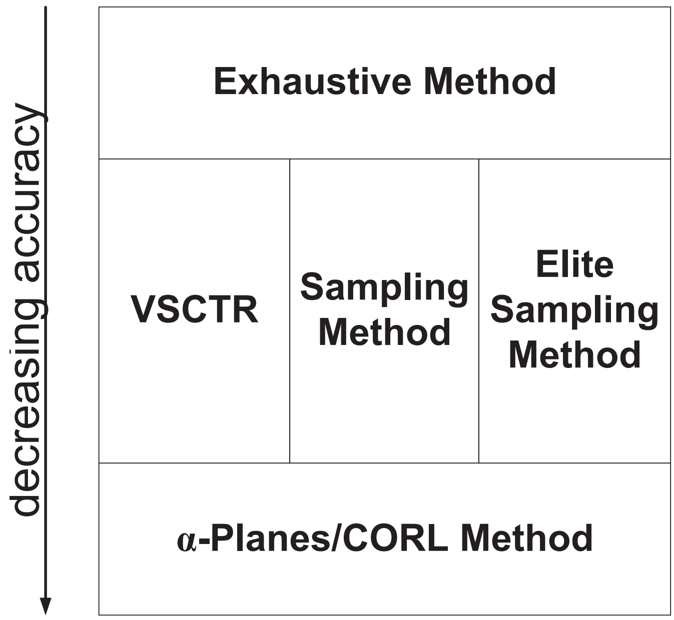


Fig. 5.7. Hierarchy of type-2 defuzzification methods' performance in relation to accuracy, for the Powder0.05 test set. The exhaustive method is used as a benchmark.

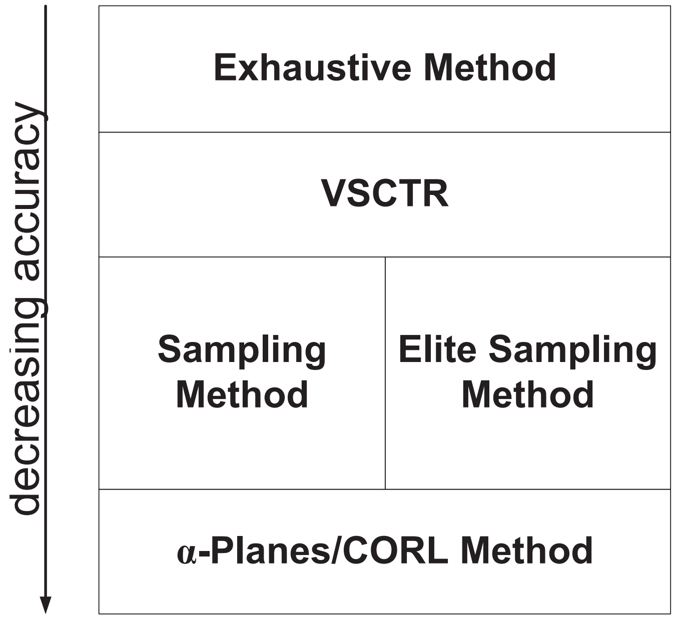


Fig. 5.8. Approximated hierarchy of type-2 defuzzification methods' performance in relation to accuracy, for the Powder0.05 test set. (In this case only one or two results obscured a direct ranking of two methods; these results were ignored.) The exhaustive method is used as a benchmark.

sampling method, the elite sampling method and the α-planes/CORL technique, the ranking of accuracy depended on the degree of discretisation and the number of α-planes. Figures 5.9 and 5.10 summarise the methods' ranking in relation to accuracy.

ShoppingFIS0.05 VSCTR, the sampling method and the elite sampling method outperformed the α-planes method. VSCTR outperformed the sampling method apart from when sample sizes 50, 100 and 1000 were used. The elite sampling method performed better than VSCTR for sample sizes 250, 1000, 5000, 10000, 50000 and 100000. Whether sampling outperformed elite sampling or vice versa depended on the degree of discretisation. Figure 5.11 display the methods' ranking in relation to accuracy.

All the methods tested were satisfactory in relation to speed and accuracy, but some performed better than others. In the majority of test runs, VSCTR proved more accurate than both the sampling approaches and the α-planes method. This is not surprising in the light of the interval experimental results, as the Nie-Tan Method, of which VSCTR is a generalisation, performed well for accuracy in the evaluations of interval methods (Table 5.5).

A possible explanation for VSCTR outperforming the sampling methods is that in the experiments, the grid method of discretisation was employed. During defuzzification under the grid method, many embedded sets are created which prove to be redundant. For example, Figure 5.12 shows a selection of six embedded sets of a type-2 fuzzy set discretised according to the grid method, which all have the same defuzzified value of 0.5. There are many more possible embedded sets with the same defuzzified value. Had the standard method of discretisation been used, would sampling be demonstrated to be more accurate than VSCTR? The answer to this question

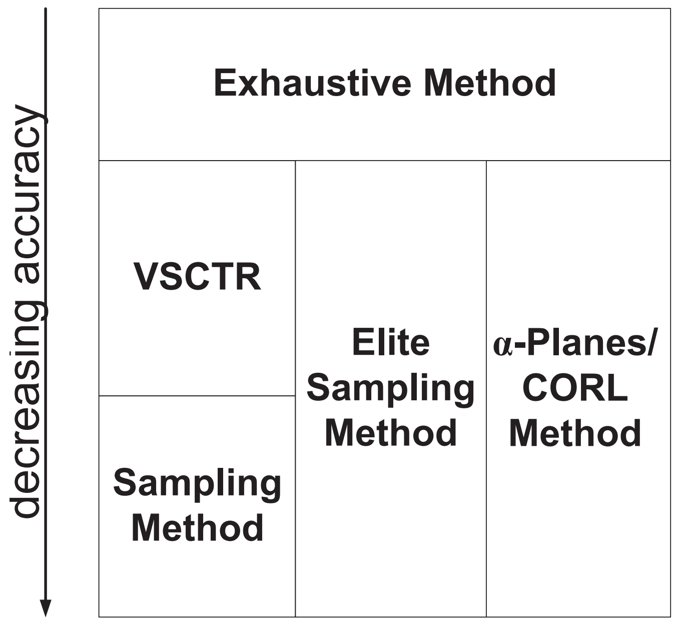


Fig. 5.9. Hierarchy of type-2 defuzzification methods' performance in relation to accuracy, for the Shopping0.1 test set. The exhaustive method is used as a benchmark.

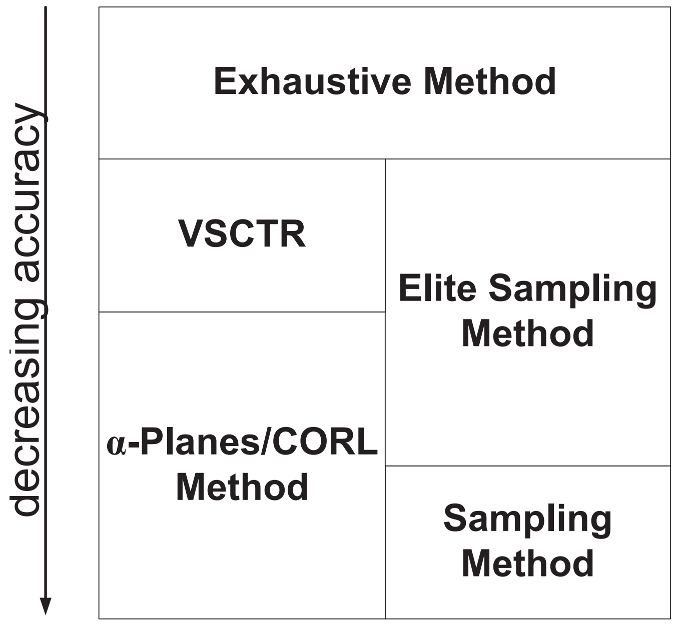


Fig. 5.10. Approximated hierarchy of type-2 defuzzification methods' performance in relation to accuracy, for the Shopping0.1 test set. (In this case only one or two results obscured a direct ranking of two methods; these results were ignored.) The exhaustive method is used as a benchmark.

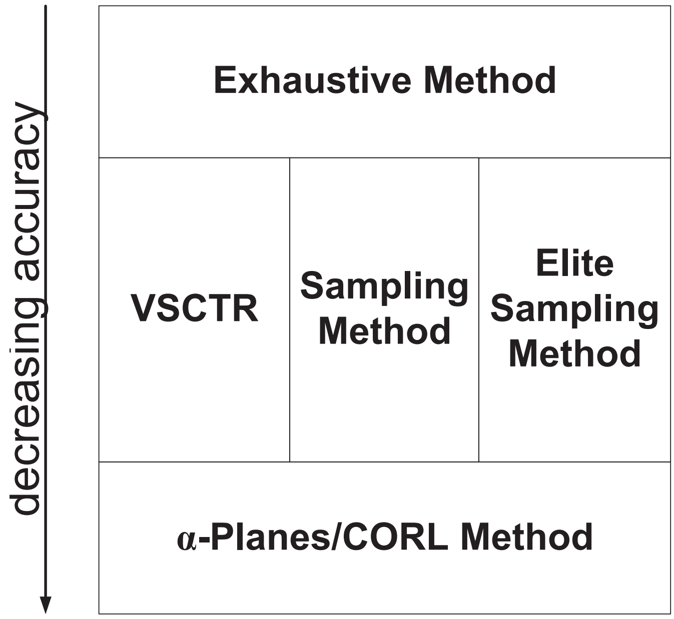


Fig. 5.11. Hierarchy of type-2 defuzzification methods' performance in relation to accuracy, for the Shopping0.05 test set. The exhaustive method is used as a benchmark.

is a topic for future research (Section 6.1).

With non-elite sampling, the sample is skewed, since embedded sets are retained that under exhaustive defuzzification would be eliminated. This leads to inaccuracies even when the sample size is enormous (Tables I.3 and K.3). With elite sampling, many, indeed the majority, of embedded sets are excluded. We term an embedded set that is not excluded a *Non-Redundant Embedded Set (NRES)*. The numbers of NRESs are indicated in Column 3 of Tables I.3 to L.3. In these tables Column 4 shows the number of NRESs as a percentage of the sample size, and Column 5 as a percentage of all the embedded sets of the type-2 fuzzy set. Employing elite sampling leads to a drastically reduced effective sample size, which in turn engenders inaccuracies.

Elite sampling is more accurate than non-elite sampling but is not necessarily worth the extra time it takes, as with generalised methods defuzzification time is more of an issue than with interval methods.

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 I.5, K.5 and M.5) 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 [17] and [13].

Efficiency of Generalised Methods VSCTR is undoubtedly the fastest method for defuzzifica-

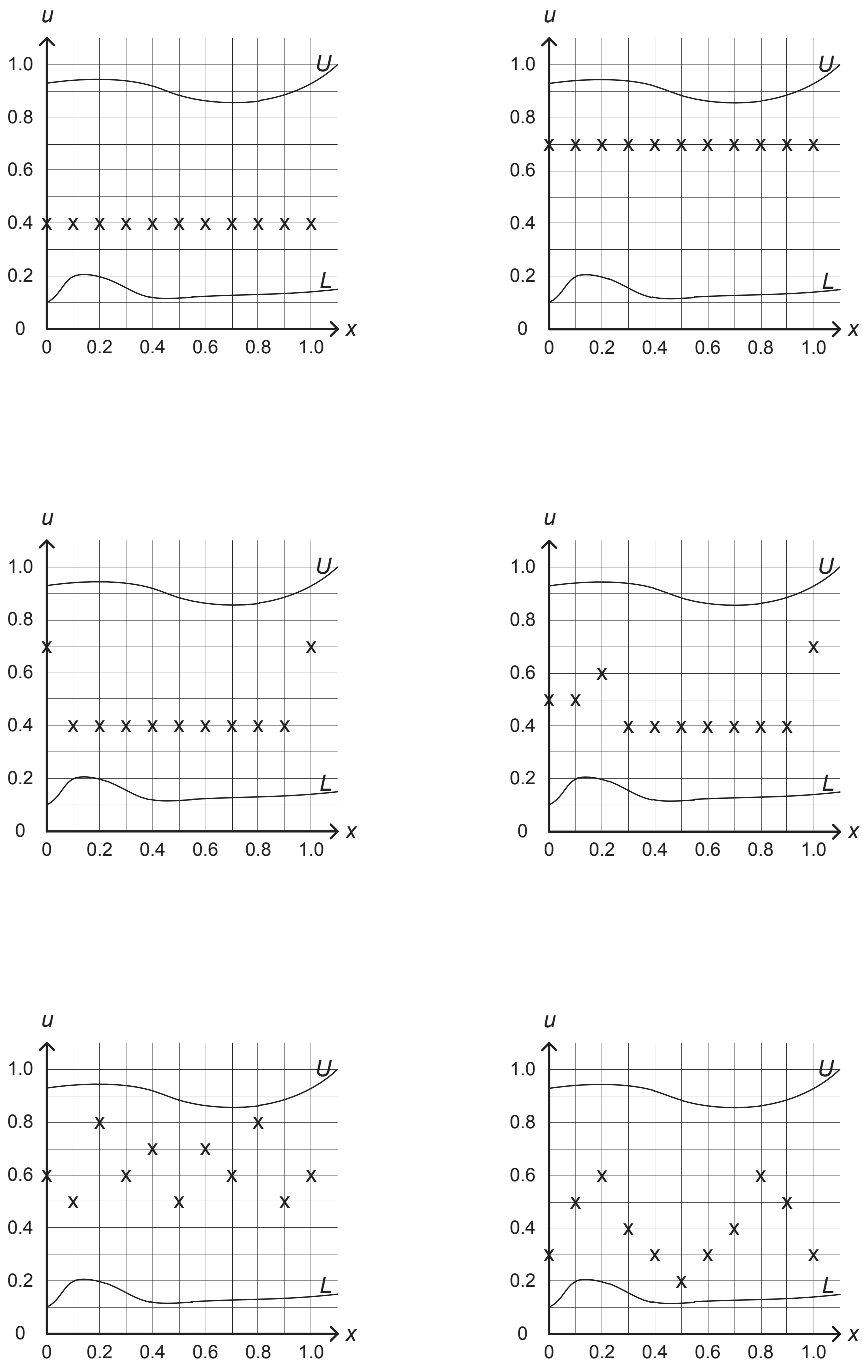


Fig. 5.12. Embedded sets, all of which have the same defuzzified value of 0.5. L is the lower membership function, U the upper membership function, and the FOU is the region between L and U.

tion 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. As the generalisation of the interval Nie-Tan Method, which performed well for speed in the interval test runs (Table 5.10), VSCTR's efficiency is to be expected.

Recommended Generalised Method Assuming that the grid method of discretisation is employed, VSCTR has been shown to be the best performing method of type-2 defuzzification of those evaluated.

Summary

The methods presented in Chapter 2 were compared and contrasted in various respects and then tested for accuracy and efficiency. Overall the best performing interval method was CORL and the worst performing the Wu-Mendel Approximation, with EIASC and the Nie-Tan method coming in between. The best performing generalised method was VSCTR, though in terms of accuracy it was only slightly better than the sampling and the elite sampling methods. The worst performing strategy was the α-planes method.

The next chapter concludes the thesis.

Chapter 6

Conclusions and Discussion

To conclude the thesis, this chapter summarises the themes and results of the preceding chapters and suggests directions for future work. In this thesis two original defuzzification techniques have been presented, the sampling method for generalised type-2 fuzzy sets, and the Greenfield-Chiclana Collapsing Defuzzifier for interval type-2 fuzzy sets. In addition to these two defuzzification techniques, the grid method of discretisation, a straightforward alternative approach for type-2 fuzzy sets, has been introduced.

In 2004, at the commencement of the work reported in this thesis, there were no practical generalised defuzzification methods, and only two interval techniques — the KMIP and the Wu-Mendel Approximation. Accordingly my research hypothesis as stated in Chapter 1 was:

> **The development of discretised, generalised type-2 fuzzy inferencing systems has been impeded by computational complexity, particularly in relation to defuzzification. The development of alternative defuzzification algorithms will resolve this defuzzification bottleneck.**

As evidenced by the publication record over the past seven years, other researchers (working entirely independently) would appear to subscribe to my hypothesis, notably Lucas et al. [42] and Liu [41]. Thus the development of new algorithms for defuzzification of type-2 fuzzy sets has progressed through the work of several people.

Several conclusions may be drawn from this investigation:

1. Through the development of alternative defuzzification algorithms the defuzzification bottleneck for generalised type-2 fuzzy sets has been resolved.

2. VSCTR is the best performing generalised method under the grid method of discretisation of those compared.

3. The experimental evaluation shows the Greenfield-Chiclana Collapsing Defuzzifier to be the best performing interval defuzzification method of those compared.

4. 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.

5. The Karnik-Mendel algorithms, in the EIASC version, cannot be depended upon to find the left and right endpoints of the TRS interval.

6. It has been demonstrated mathematically that in the continuous case the RESA and NTS are identical.

7. There is experimental evidence showing that the GCCD and the Nie-Tan defuzzified values both approach the exhaustive defuzzified value as discretisation becomes finer.

8. Were it to be proven that the GCCD and the Nie-Tan defuzzified values both approach the exhaustive defuzzified value as discretisation becomes finer (Conclusion 7), then it would follow immediately that the continuous RESA is the RES.

Each conclusion will be discussed in turn.

Conclusion 1 (Research Hypothesis): Through the development of alternative defuzzification algorithms the defuzzification bottleneck for generalised type-2 fuzzy sets has been resolved.

All the generalised methods evaluated in Chapter 5 are satisfactory for accuracy and efficiency, though parameters such as sample size and number of α-planes must be relatively small for the defuzzification process to be speedy. The defuzzification bottleneck has therefore been resolved.

Conclusion 2: VSCTR is the best performing generalised method under the grid method of discretisation.

The test regime of Chapter 5 demonstrated VSCTR to be the best performing method for accuracy and efficiency. However no mathematical justification has been provided to show that VSCTR leads to the same defuzzified value as the exhaustive method. The experiments were performed using the grid method of discretisation, and it is conjectured that the sampling method might outperform VSCTR for accuracy on the standard method of discretisation.

Conclusion 3: The Greenfield-Chiclana Collapsing Defuzzifier is the best performing interval defuzzification method.

The collapsing defuzzifier was conceived as a generalised defuzzification technique, to be developed in three stages.

Simple Interval Collapsing: The collapsing formula has been derived, and the method implemented in software.

Interval Collapsing: The collapsing formula has been derived, but the method has not been implemented in software. The derivation of this result was seen as a step towards the ultimate goal of generalised collapsing.

Generalised Collapsing: The collapsing formula has yet to be derived.

To date only interval collapsing has been derived, implemented in code, and tested. Methods based on the KMIP have been, and remain, the accepted interval methods. But the collapsing method poses a challenge to the status quo as it has been shown by the experimental evaluation of Chapter 5 to give more accurate results than the KMIP family of methods. True, EIASC (the version of the Karnik-Mendel algorithms used in the tests) is faster than CORL (the most accurate variant of collapsing), but this extra speed is of little value as CORL is a fast method anyway. The test results therefore challenge the KMIP family's established position.

<u>Conclusion 4:</u> 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.

Experiments described in Chapter 5 demonstrated that as the number of α-planes increased, the α-planes/CORL results do not converge to the value obtained by generalised exhaustive defuzzification. Furthermore even the α-planes/interval exhaustive results fail to converge to this value. The defuzzified values for both the α-planes/CORL and α-planes/interval exhaustive methods are similar and appear to converge to the same number, which is *not* the value obtained from generalised exhaustive defuzzification. This discrepancy reveals an issue with the α-planes method itself.

<u>Conclusion 5:</u> The Karnik-Mendel algorithms, in the EIASC version, cannot be depended upon to find the left and right endpoints of the TRS interval.

The sporadic unreliability of EIASC in finding the left and right endpoints of the TRS interval was reported on in Chapter 5.

<u>Conclusion 6:</u> It has been demonstrated mathematically that in the continuous case the RESA and NTS are identical.

It was proved in Section 4.6 that in the continuous case the RESA and NTS are identical.

Conclusion 7: There is experimental evidence showing that the GCCD and the Nie-Tan defuzzified values both approach the exhaustive defuzzified value as discretisation becomes finer.

Section 4.6 summarises the strong experimental evidence suggesting that the GCCD and the Nie-Tan defuzzified values both approach the exhaustive defuzzified value as discretisation becomes finer. This result has yet to be proved mathematically.

Conclusion 8: Were it to be proved that the GCCD and the Nie-Tan defuzzified values both approach the exhaustive defuzzified value as discretisation becomes finer (Conclusion 6), then it would follow immediately that the continuous RESA is the RES.

If the TRS and NTS can be proved to be equivalent in the continuous case then since it has already been proved that the continuous RESA and the continuous NTS are the same type-1 set, then the continuous RESA and the continuous TRS must defuzzify to the same value. This would imply that the continuous RESA *is* the RES (Section 4.2).

6.1 Further Work

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

Generalising the Collapsing Defuzzifier Extension of the GCCD to generalised type-2 fuzzy sets.

Continuous Type-2 Fuzzy Inferencing

- Complete the proof that the continuous NTS and the continuous TRS have the same defuzzified value.
- Show that the continuous RESA is the RES. It has been demonstrated that the continuous RESA is the same as the continuous NTS. To prove this result, it would be sufficient to prove that the continuous NTS has the same defuzzified value as the continuous TRS.
- Investigate continuous type-2 fuzzy inferencing.

Grid Method of Discretisation Exploration of the implications of the method of discretisation for type-2 fuzzy inferencing.

Stratified TRS Implementation in software of the generalised defuzzification technique based on the stratified structure of the TRS.

Type-1 OWA Based Approach Software implementation of the Type-1 OWA based approach.

Sampling Method Using the Standard Method of Discretisation Investigate the accuracy and efficiency of the sampling method when implemented using the standard method of discretisation.

EIASC Investigate the reasons why EIASC sometimes fails to locate the left and right endpoints of the TRS interval, and why the left and right endpoints can even be reversed.

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.

Summary

The objective of the research presented in this thesis is to reduce the computational complexity of type-2 defuzzification. Two new type-2 defuzzification methods have been presented, the sampling method and the Greenfield-Chiclana Collapsing Defuzzifier. The available type-2 defuzzification techniques have been surveyed, and the main ones coded and tested comparatively for accuracy and efficiency. The sampling method performed well as a generalised defuzzifier, but was outperformed by VSCTR. The Greenfield-Chiclana Collapsing Defuzzifier outperformed the three other interval methods tested, including EIASC, a version of the established Karnik-Mendel Algorithms. The testing revealed discrepancies between actual and expected results for EIASC and the α-planes method.

In addition to the two new defuzzification methods, an alternative strategy for discretising type-2 fuzzy sets has been introduced. This discretisation technique reduces the computational complexity of all stages of the fuzzy inferencing system.

References

[1] Janet Aisbett, John T. Rickard, and David G. Morgenthaler. Type-2 Fuzzy Sets as Functions on Spaces. *IEEE Transactions on Fuzzy Systems*, 18(4):841–844, August 2010.

[2] Philip A. Birkin and Jonathan M. Garibaldi. A Novel Dual-Surface Type-2 Controller for Micro Robots. In *Proceedings of FUZZ-IEEE 2010*, pages 359–366, Barcelona, Spain, 2010.

[3] Max Black. Vagueness: An Exercise in Logical Analysis. *Philosophy of Science*, 4:427 – 455, 1937.

[4] Francisco Chiclana, Francisco Herrera, and Enrique Herrera-Viedma. Integrating Multiplicative Preference Relations in a Multipurpose Decision-Making Model Based on Fuzzy Preference Relations. *Fuzzy Sets and Systems*, 122(2):277–291, September 2001.

[5] Francisco Chiclana, Francisco Herrera, and Enrique Herrera-Viedma. A Note on the Internal Consistency of Various Preference Representations. *Fuzzy Sets and Systems*, 131(1):75–78, September 2002.

[6] Francisco Chiclana and Shang-Ming Zhou. The Type-1 OWA Operator and the Centroid of Type-2 Fuzzy Sets. In *Proceedings of EUSFLAT-LFA 2011*, pages 15–20, Aix-les-Bains, France, July 2011.

[7] Simon Coupland. Type-2 Fuzzy Sets: Geometric Defuzzification and Type-Reduction. In *Proc. FOCI 2007*, pages 622 – 629, Honolulu, Hawaii, USA, April 2007.

[8] Simon Coupland and Robert I. John. Geometric Type-1 and Type-2 Fuzzy Logic Systems. *IEEE Transactions on Fuzzy Systems*, 15(1):3–15, February 2007.

[9] Morris H. DeGroot. *Probability and Statistics*. Addison Wesley, second edition, 1989.

[10] Carol Gibson (ed.). *The Facts on File Dictionary of Mathematics*. Facts on File, New York and Oxford, second edition, 1988.

[11] Andreas P. Engelbrecht. *Computational Intelligence: An Introduction*. Wiley, West Sussex, U.K., 2002.

[12] Sarah Greenfield. Uncertainty, Imprecision and Vagueness. Master's thesis, De Montfort University, 2005.

[13] Sarah Greenfield and Francisco Chiclana. Combining the α-Plane Representation with an Interval Defuzzification Method. In *Proceedings of EUSFLAT-LFA 2011*, pages 920–927, Aix-les-Bains, France, July 2011.

[14] Sarah Greenfield and Francisco Chiclana. Type-Reduction of the Discretised Interval Type-2 Fuzzy Set: Approaching the Continuous Case through Progressively Finer Discretisation. Accepted for publication in the *Journal of Artificial Intelligence and Soft Computing Research*, 2011.

[15] Sarah Greenfield and Francisco Chiclana. Type-Reduction of the Discretised Interval Type-2 Fuzzy Set: What Happens as Discretisation Becomes Finer? In *Proceedings of the IEEE Symposium on Advances in Type-2 Fuzzy Logic Systems 2011*, Paris, April 2011.

[16] Sarah Greenfield, Francisco Chiclana, Simon Coupland, and Robert I. John. The Collapsing Method of Defuzzification for Discretised Interval Type-2 Fuzzy Sets. *Information Sciences*, 179(13):2055–2069, June 2009.

[17] Sarah Greenfield, Francisco Chiclana, Simon Coupland, and Robert I. John. Type-2 Defuzzification: Two Contrasting Approaches. In *Proceedings of FUZZ-IEEE 2010*, Barcelona, July 2010. DOI: 10.1109/FUZZY.2010.5584007.

[18] Sarah Greenfield, Francisco Chiclana, and Robert I. John. The Collapsing Method: Does the Direction of Collapse Affect Accuracy? In *Proceedings of IFSA-EUSFLAT 2009*, pages 980–985, Lisbon, Portugal, July 2009.

[19] Sarah Greenfield, Francisco Chiclana, and Robert I. John. Type-Reduction of the Discretised Interval Type-2 Fuzzy Set. In *Proceedings of FUZZ-IEEE 2009*, pages 738–743, Jeju Island, Korea, August 2009.

[20] Sarah Greenfield, Francisco Chiclana, Robert I. John, and Simon Coupland. The Collapsing Method of Defuzzification for Discretised Interval Type-2 Fuzzy Sets. In *Proc. UKCI 2007*, London, July 2007.

[21] Sarah Greenfield, Francisco Chiclana, Robert I. John, and Simon Coupland. The Sampling Method of Defuzzification for Type-2 Fuzzy Sets: Experimental Evaluation. *Information Sciences*, 189:77–92, April 2012. DOI: 10.1016/j.ins.2011.11.042.

[22] Sarah Greenfield and Robert I. John. Optimised Generalised Type-2 Join and Meet Operations. In *Proc. FUZZ-IEEE 2007*, London, July 2007.

[23] Sarah Greenfield and Robert I. John. Stratification in the Type-Reduced Set and the Generalised Karnik-Mendel Iterative Procedure. In *Proc. IPMU 2008*, pages 1282–1289, Malaga, June 2008.

[24] Sarah Greenfield, Robert I. John, and Simon Coupland. A Novel Sampling Method for Type-2 Defuzzification. In *Proc. UKCI 2005*, pages 120–127, London, September 2005.

[25] Hani Hagras and Christian Wagner. Introduction to Interval Type-2 Fuzzy Logic Controllers — Towards Better Uncertainty Handling in Real World Applications. *IEEE Systems, Man and Cybernetics eNewsletter*, 2009. issue 27.

[26] Petr Hajek. Fuzzy Logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Stanford, Fall 2002. URL http://plato.stanford.edu/archives/fall2002/entries/logic-fuzzy.

[27] Francisco Herrera, Enrique Herrera-Viedma, and Francisco Chiclana. Multiperson Decision Making Based on Multiplicative Preference Relations. *European Journal of Operational Research*, 129:372 – 385, March 2001.

[28] Enrique Herrera-Viedma, Francisco Herrera, and Francisco Chiclana. A Consensus Model for Multiperson Decision Making with Different Preference Structures. *IEEE Transactions on Systems, Man and Cybernetics*, 32:394 – 402, May 2002.

[29] Emmanuel A. Jammeh, Martin Fleury, Christian Wagner, Hani Hagras, and Mohammed Ghanbari. Interval Type-2 Fuzzy Logic Congestion Control for Video Streaming Across IP Networks. *IEEE Transactions on Fuzzy Systems*, 17(5):1123–1142, 2009.

[30] Robert I. John. *Perception Modelling using Type-2 Fuzzy Sets*. PhD thesis, De Montfort University, 2000.

[31] Robert I. John and Simon Coupland. Type-2 Fuzzy Logic: A Historical View. *IEEE Computational Intelligence Magazine*, 2(1):57 – 62, February 2007. DOI: 10.1109/MCI.2007.357194.

[32] Nilesh N. Karnik. An Introduction to Type-2 Fuzzy Logic Systems. Technical report, University of Southern California, June 1998. http://sipi.usc.edu/ mendel/report.

[33] Nilesh N. Karnik and Jerry M. Mendel. Centroid of a Type-2 Fuzzy Set. *Information Sciences*, 132:195 – 220, 2001.

[34] Erdal Kayacan, Okyay Kaynak, Rahib Abiyev, Jim Tørresen, Mats Høvin, and Kyrre Glette. Design of an Adaptive Interval Type-2 Fuzzy Logic Controller for the Position Control of a Servo System with an Intelligent Sensor. In *Proceedings of FUZZ-IEEE 2010*, pages 1125–1132, Barcelona, Spain, 2010.

[35] George J. Klir and Tina A. Folger. *Fuzzy Sets, Uncertainty, and Information*. Prentice-Hall International, 1992.

[36] George J. Klir and Bo Yuan. *Fuzzy Sets and Fuzzy Logic*. Prentice-Hall P T R, 1995.

[37] Bart Kosco. Fuzziness Vs. Probability. *International Journal of General Systems*, 17(1):211–240, 1990.

[38] Werner Van Leekwijck and Etienne E. Kerre. Defuzzification: Criteria and Classification. *Fuzzy Sets and Systems*, 108:159 – 178, 1999. DOI: 10.1016/j.fss.2008.06.018.

[39] Leonardo Leottau and Miguel Melgarejo. Implementing an Interval Type-2 Fuzzy Processor onto a DSC 56F8013. In *Proceedings of FUZZ-IEEE 2010*, pages 1939–1942, Barcelona, Spain, 2010.

[40] Ondrej Linda and Milos Manic. Importance Sampling Based Defuzzification for General Type-2 Fuzzy Sets. In *Proceedings of FUZZ-IEEE 2010*, pages 1943–1949, Barcelona, Spain, July 2010.

[41] Feilong Liu. An Efficient Centroid Type-Reduction Strategy for General Type-2 Fuzzy Logic System. *Information Sciences*, 178(9):2224–2236, 2008.

[42] Luís Alberto Lucas, Tania Mezzadri Centeno, and Myriam Regattieri Delgado. General Type-2 Fuzzy Inference Systems: Analysis, Design and Computational Aspects. In *Proceedings of FUZZ-IEEE 2007*, London, 2007.

[43] Jerry M. Mendel. *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Prentice-Hall PTR, 2001.

[44] Jerry M. Mendel and Robert I. John. Type-2 Fuzzy Sets Made Simple. *IEEE Transactions on Fuzzy Systems*, 10(2):117 – 127, 2002.

[45] Jerry M. Mendel, Feilong Liu, and Daoyuan Zhai. α-Plane Representation for Type-2 Fuzzy Sets: Theory and applications. *IEEE Transactions on Fuzzy Systems*, 17(5):1189 – 1207, 2009.

[46] Simon M. Miller, Viara Popova, Robert John, and Mario Gongora. An Interval Type-2 Fuzzy Distribution Network. In *Proc. 2009 IFSA World Congress/EUSFLAT Conference*, pages 697–702, Lisbon, July 2009.

[47] Maowen Nie and Woei Wan Tan. Towards an Efficient Type-Reduction Method for Interval Type-2 Fuzzy Logic Systems. In *Proceedings of FUZZ-IEEE 2008*, pages 1425–1432, Hong Kong, June 2008.

[48] Christian Wagner and Hani Hagras. Toward General Type-2 Fuzzy Logic Systems based on zSlices. *IEEE Transactions on Fuzzy Systems*, 18(4):637 – 660, 2010.

[49] Dongrui Wu and Jerry M. Mendel. Enhanced Karnik-Mendel Algorithms. *IEEE Transactions on Fuzzy Systems*, 17(4):923–934, August 2009.

[50] Dongrui Wu and 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.

[51] Hongwei Wu and 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):622–639, 2002.

[52] Lotfi A. Zadeh. Fuzzy Sets. *Information and Control*, 8:338 – 353, 1965.

[53] Lotfi A. Zadeh. The Concept of a Linguistic Variable and its Application to Approximate Reasoning. *Information Sciences*, 8:199 – 249, 1975.

[54] Lotfi A. Zadeh. The Concept of a Linguistic Variable and its Application to Approximate Reasoning – II. *Information Sciences*, 8:301 – 357, 1975.

[55] Lotfi A. Zadeh. The Concept of a Linguistic Variable and its Application to Approximate Reasoning – III. *Information Sciences*, 9:43 – 80, 1975.

[56] Mina Zaher, Hani Hagras, Amr Khairy, and Mohamed Ibrahim. A Type-2 Fuzzy Logic based Model for Renewable Wind Energy Generation. In *Proceedings of FUZZ-IEEE 2010*, pages 511–518, Barcelona, Spain, 2010.

[57] Daoyuan Zhai and Jerry M. Mendel. Centroid of a general type-2 fuzzy set computed by means of the centroid-flow algorithm. In *Proceedings of FUZZ-IEEE 2010*, pages 895–902, Barcelona, July 2010. DOI: 10.1109/FUZZY.2010.5584547.

[58] Daoyuan Zhai and 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):401–422, 2011. DOI: 10.1109/TFUZZ.2010.2103076.

[59] Shang-Ming Zhou, Francisco Chiclana, Robert I. John, and Jonathan M. Garibaldi. Type-1 OWA Operators for Aggregating Uncertain Information with Uncertain Weights Induced By Type-2 Linguistic Quantifiers. *Fuzzy Sets and Systems (ISSN 0165-0114)*, 159(24):3281 – 3296, December 2008. DOI: 10.1016/j.fss.2008.06.018.

Appendix A
Operations on Crisp Sets

The operations that may be performed on crisp sets are *union*, *intersection*, and *complement*, which may, as with fuzzy sets, be defined in terms of the characteristic function δ. Let X be a crisp set (the universal set), sets A and B subsets of X, and x an element of X, $(x \in X)$.

Union: The union of A and B contains all the elements in either A or B.

$$\delta_{A\cup B}(x) = \begin{cases} 1 & \text{if } x \in A \text{ or } x \in B, \\ 0 & \text{if } x \notin A \text{ and } x \notin B. \end{cases}$$

Intersection: The intersection of A and B contains all the elements in both A and B.

$$\delta_{A\cap B}(x) = \begin{cases} 1 & \text{if } x \in A \text{ and } x \in B, \\ 0 & \text{if } x \notin A \text{ or } x \notin B. \end{cases}$$

Complement: The complement of A ($\overline{A}$) contains those elements of X that are not in A.

$$\delta_{\overline{A}}(x) = \begin{cases} 1 & \text{if } x \notin A, \\ 0 & \text{if } x \in A. \end{cases}$$

The following three corollaries can easily demonstrated from these definitions [52]:

1. $$A \cup B \Rightarrow \delta_{A\cup B}(x) = max[\delta_A(x), \delta_B(x)].$$

2. $$A \cap B \Rightarrow \delta_{A\cap B}(x) = min[\delta_A(x), \delta_B(x)].$$

3. $$\delta_{\overline{A}}(x) = 1 - \delta_A(x).$$

Crisp sets obey the *Law of Excluded Middle* ($A \cup \overline{A} = X$) and the *Law of Contradiction* ($A \cap \overline{A} = \emptyset$), where $\emptyset$ is the empty set.

Appendix B

Type-1 Fuzzy Sets: Definitions

Definition B.1 (Type-1 Fuzzy Set)**.** *Let X be a universe of discourse. A type-1 fuzzy set A on X is characterised by a membership function* $\mu_A : X \to [0,1]$ *and can be expressed as follows [52]:*

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

Definition B.2 (Type-1 Fuzzy Set with Continuous Universe of Discourse)**.** *A mathematical representation of type-1 fuzzy set A with continuous universe of discourse X is*

$$A = \int_{x \in X} \mu_A(x)/x. \tag{B.2}$$

Definition B.3 (Type-1 Fuzzy Set with Discrete Universe of Discourse)**.** *A mathematical representation of type-1 fuzzy set A with discrete universe of discourse X is*

$$A = \sum_{x \in X} \mu_A(x)/x. \tag{B.3}$$

Note that the membership grades of A are crisp numbers.

Definition B.4 (Support)**.** *The* support *of a type-1 fuzzy set A is the crisp set that contains all the elements of the universal set X that have non-zero membership grades in A* [36, page 21].

This definition can be written as

$$\text{supp}(A) = \{x \in X | \mu_A(x) > 0\}.$$

Definition B.5 (Normal Type-1 Fuzzy Set)**.** *A normal type-1 fuzzy set is a type-1 fuzzy set for which the maximum membership grade is* 1 [36, page 21].

Definition B.6 (Cardinality)**.** *For type-1 fuzzy set A,* $| \ A \ |$*, the cardinality of A, is the number of tuples in A.*

Definition B.7 (Scalar Cardinality)**.** *The* scalar cardinality *of a type-1 fuzzy set A defined on a finite universal set X is the summation of the membership grades of all the elements of* supp(A). *Thus,*

$$\| A \| = \sum_{x \in X} \mu_A(x).$$

Definition B.8 (α-Cut)**.** *"An α-cut of a [type-1] fuzzy set A is a crisp set A_α that contains all the elements of the universal set X that have a membership grade in A greater than or equal to the specified value of α. This definition can be written as*

$$A_\alpha = \{x \in X \mid \mu_A(x) \geq \alpha\}.'' \text{ [35, page 16]}$$

Definition B.9 (Centroid of a Type-1 Fuzzy Set)**.** *Let A be a non-empty type-1 fuzzy set that has been discretised into m vertical slices (at x_1, x_2, ..., x_m). The centroid of A is calculated by this formula:*

$$X_A = \frac{\sum_{i=1}^{i=m} \mu_A(x_i) x_i}{\sum_{i=1}^{i=m} \mu_A(x_i)}.$$

Appendix C

Interval Test Set M

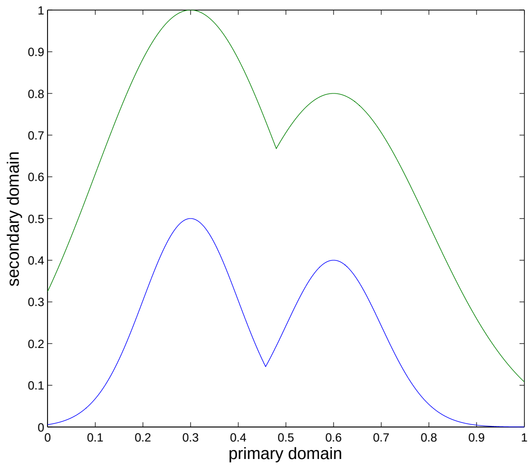


Fig. C.1. Interval Test Set M.

NUMBER OF SLICES	EXHAUSTIVE DEFUZZIFI-CATION	COLLAPSING OUTWARD RIGHT-LEFT	EIASC	NIE-TAN DEFUZZIFI-CATION	WU-MENDEL DEFUZZIFI-CATION
5	0.4199972460	0.4184943624	0.4211829214	0.4206867325	0.4295236614
9	0.4348736430	0.4344474812	0.4370057000	0.4337693820	0.4424349889
11	0.4358519320	0.4355179126	0.4384915906	0.4349819049	0.4433509906
17	0.4372019939	0.4369963064	0.4393687298	0.4366808102	0.4446021161
21	0.4376286265	0.4374646165	0.4397562165	0.4372193093	0.4449897526
51	not possible	0.4386565707	0.4407611544	0.4385614087	0.4458806157
101	not possible	0.4390891348	0.4411369145	0.4390423183	0.4462108484
1001	not possible	0.4394818955	0.4414842336	0.4394772838	0.4465104689
10001	not possible	0.4395214423	0.4415190125	0.4395209818	0.4465405095
100001	not possible	0.4395254018	0.4415224943	0.4395253558	0.4465435162

Table C.1. Defuzzified values for test set M.

NUMBER OF SLICES	EXHAUSTIVE DEFUZZIFI-CATION	ERROR CORL	ERROR EIASC	ERROR NIE-TAN METHOD	ERROR WU-MENDEL APPROX.
5	0.4199972460	0.0015028836	0.0011856754	**0.0006894865**	0.0095264154
9	0.4348736430	**0.0004261618**	0.0021320570	0.0011042610	0.0075613459
11	0.4358519320	**0.0003340194**	0.0026396586	0.0008700271	0.0074990586
17	0.4372019939	**0.0002056875**	0.0021667359	0.0005211837	0.0074001222
21	0.4376286265	**0.0001640100**	0.0021275900	0.0004093172	0.0073611261

Table C.2. Errors for test set M. The lowest errors are shown in bold.

NO. OF SLICES	EXHAUST. DEFUZZIFI-CATION	COLLAPSING OUTWARD RIGHT-LEFT	EIASC	NIE-TAN DEFUZZIFI-CATION	WU-MENDEL DEFUZZIFI-CATION
5	0.00214 secs.	0.0000768 secs.	**0.0000199 secs.**	0.0000363 secs.	0.0000927 secs.
9	0.0359 secs.	0.0000787 secs.	**0.0000200 secs.**	0.0000388 secs.	0.0000997 secs.
11	0.155 secs.	0.0000792 secs.	**0.0000205 secs.**	0.0000358 secs.	0.0000935 secs.
17	86.3 secs.	0.0000831 secs.	**0.0000212 secs.**	0.0000362 secs.	0.000133 secs.
21	5.97 hours	0.000841 secs.	**0.0000230 secs.**	0.0000378 secs.	0.0000917 secs.
51	not possible	0.000978 secs.	0.0000273 secs.	0.0000384 secs.	0.0000920 secs.
101	not possible	0.00125 secs.	0.0000356 secs.	0.0000418 secs.	0.000102 secs.
1001	not possible	0.00515 secs.	0.000180 secs.	0.0000781 secs.	0.000165 secs.
10001	not possible	0.00440 secs.	0.00160 secs.	0.000254 secs.	0.000823 secs.
100001	not possible	0.0609 secs.	0.0211 secs.	0.00650 secs.	0.0203 secs.

Table C.3. Defuzzification times for test set M. The fastest timings are shown in bold.

	INTERVAL EXHAUSTIVE METHOD			**EIASC**		
NO. OF SLICES	LEFT ENDPOINT	RIGHT ENDPOINT	DEFUZZ. VALUE	LEFT ENDPOINT	RIGHT ENDPOINT	DEFUZZ. VALUE
5	0.27671543	0.56565041	0.41999725	0.27671543	0.56565041	0.42118292
9	0.30045034	0.57356106	0.43487364	0.30045034	0.57356106	0.43700570
11	0.30462202	0.57236116	0.43585193	0.30462202	0.57236116	0.43849159
17	0.30800197	0.57073549	0.43720199	0.30800197	0.57073549	0.43936873
21	0.30902603	0.57048641	0.43762863	0.30902603	0.57048641	0.43975622

Table C.4. For test set M, left endpoints, right endpoints and defuzzified values obtained firstly by interval exhaustive defuzzification, and secondly by EIASC.

Appendix D

Interval Test Set N

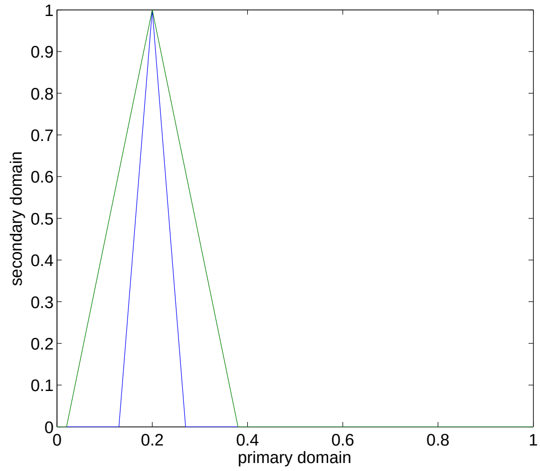


Fig. D.1. Interval Test Set N.

NUMBER OF SLICES	EXHAUSTIVE DEFUZZIFI-CATION	COLLAPSING OUTWARD RIGHT-LEFT	EIASC	NIE-TAN DEFUZZIFI-CATION	WU-MENDEL DEFUZZIFI-CATION
5	0.2500000000	0.2500000000	0.2500000000	0.2500000000	0.2684478216
9	0.2137762063	0.2166152302	0.2135859199	0.2071078431	0.2592275919
11	0.2000000000	0.1996177735	0.2000000000	0.2000000000	0.2094094585
17	0.2003511054	0.1997796246	0.1999873891	0.1998318386	0.2205701965
21	0.2000000000	0.1996245463	0.2000000000	0.2000000000	0.2162988115
51	not possible	0.1997811699	0.2000000000	0.2000000000	0.2191889438
101	not possible	0.1998809711	0.2000000000	0.2000000000	0.2199779857
1001	not possible	0.1999878729	0.2000000000	0.2000000000	0.2199779857
10001	not possible	0.1999987847	0.2000000000	0.2000000000	0.2199779857
100001	not possible	0.1999998784	0.2000000000	0.2000000000	0.2199779857

Table D.1. Defuzzified values for test set N.

NUMBER OF SLICES	EXHAUSTIVE DEFUZZIFI-CATION	ERROR CORL	ERROR EIASC	ERROR NIE-TAN METHOD	ERROR WU-MENDEL APPROX.
5	0.2500000000	**0.0000000000**	**0.0000000000**	**0.0000000000**	0.0184478216
9	0.2137762063	0.0028390239	**0.0001902864**	0.0066683632	0.0454513856
11	0.2000000000	0.0003822265	**0.0000000000**	**0.0000000000**	0.0094094585
17	0.2003511054	0.0005714808	**0.0003637163**	0.0005192668	0.0202190911
21	0.2000000000	0.0003754537	**0.0000000000**	**0.0000000000**	0.0162988115

Table D.2. Errors for test set N. The lowest errors are shown in bold.

NO. OF SLICES	EXHAUST. DEFUZZIFI-CATION	COLLAPSING OUTWARD RIGHT-LEFT	EIASC	NIE-TAN DEFUZZIFI-CATION	WU-MENDEL DEFUZZIFI-CATION
5	0.00215 secs.	0.0000781 secs.	**0.0000203 secs.**	0.0000366 secs.	0.0000913 secs.
9	0.0361 secs.	0.0000847 secs.	**0.0000204 secs.**	0.0000356 secs.	0.0000892 secs.
11	0.151 secs.	0.0000818 secs.	**0.0000222 secs.**	0.0000350 secs.	0.0000900 secs.
17	89.0 secs.	0.0000832 secs.	**0.0000224 secs.**	0.0000361 secs.	0.0000905 secs.
21	6.11 hours	0.0000855 secs.	**0.0000248 secs.**	0.0000381 secs.	0.0000924 secs.
51	not possible	0.000103 secs.	0.0000305 secs.	0.0000381 secs.	0.0000922 secs.
101	not possible	0.000122 secs.	0.0000406 secs.	0.0000442 secs.	0.0000960 secs.
1001	not possible	0.000521 secs.	0.000226 secs.	0.0000629 secs.	0.000167 secs.
10001	not possible	0.00442 secs.	0.00197 secs.	0.000255 secs.	0.000774 secs.
100001	not possible	0.0547 secs.	0.0262 secs.	0.00648 secs.	0.0200 secs.

Table D.3. Defuzzification times for test set N. The fastest timings are shown in bold.

	INTERVAL EXHAUSTIVE METHOD			**EIASC**		
NO. OF SLICES	LEFT ENDPOINT	RIGHT ENDPOINT	DEFUZZ. VALUE	LEFT ENDPOINT	RIGHT ENDPOINT	DEFUZZ. VALUE
5	0.25000000	0.25000000	0.25000000	0.25000000	0.25000000	0.25000000
9	0.16609589	0.26107595	0.21377621	0.16609589	0.26107595	0.21358592
11	0.16293077	0.23076923	0.20000000	0.16923077	0.23076923	0.20000000
17	0.16252575	0.23744903	0.20035111	0.16252575	0.23744903	0.19998739
21	0.16515152	0.23484848	0.20000000	0.16515152	0.23484848	0.20000000

Table D.4. For test set N, left endpoints, right endpoints and defuzzified values obtained firstly by interval exhaustive defuzzification, and secondly by EIASC.

Appendix E

Interval Test Set S

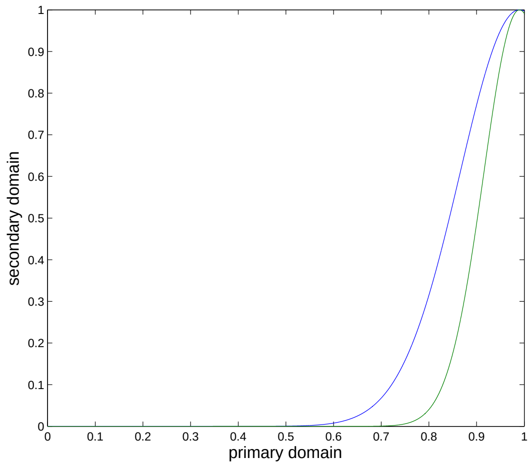


Fig. E.1. Interval Test Set S.

NUMBER OF SLICES	EXHAUSTIVE DEFUZZIFI-CATION	COLLAPSING OUTWARD RIGHT-LEFT	EIASC	NIE-TAN DEFUZZIFI-CATION	WU-MENDEL DEFUZZIFI-CATION
5	0.9819755268	0.9819724508	0.9820266266	0.9808139836	0.9807620182
9	0.9480255757	0.9479604407	0.9498630718	0.9466403800	0.9457130018
11	0.9411164375	0.9410330760	0.9433286138	0.9399863542	0.9388383001
17	0.9309357359	0.9308645243	0.9143284643	0.9301283898	0.9285457815
21	0.9273868277	0.9273228598	0.9109648792	0.9267196454	0.9249480230
51	not possible	0.9183390298	0.9025264762	0.9180843438	0.9157231847
101	not possible	0.9151513040	0.8995751696	0.9150219257	0.9124074799
1001	not possible	0.9121820221	0.8968494023	0.9121689184	0.9092945897
10001	not possible	0.9118794577	0.8965729600	0.9118781458	0.9089759570
100001	not possible	0.9118491428	0.8965452759	0.9118490116	0.9089440168

Table E.1. Defuzzified values for test set S.

NUMBER OF SLICES	EXHAUSTIVE DEFUZZIFI-CATION	ERROR CORL	ERROR EIASC	ERROR NIE-TAN METHOD	ERROR WU-MENDEL APPROX.
5	0.9819755268	**0.0000030760**	0.0000510998	0.0011615432	0.0012135086
9	0.9480255757	**0.0000651350**	0.0018374961	0.0013851957	0.0023125739
11	0.9411164375	**0.0000833615**	0.0022121763	0.0011300833	0.0022781374
17	0.9309357359	**0.0000712116**	0.0166072716	0.0008073461	0.0023899544
21	0.9273868277	**0.0000639679**	0.0164219485	0.0006671823	0.0024388047

Table E.2. Errors for test set S. The lowest errors are shown in bold.

NO. OF SLICES	EXHAUST. DEFUZZIFI-CATION	COLLAPSING OUTWARD RIGHT-LEFT	EIASC	NIE-TAN DEFUZZIFI-CATION	WU-MENDEL DEFUZZIFI-CATION
5	0.00217 secs.	0.0000761 secs.	**0.0000199 secs.**	0.0000349 secs.	0.0000914 secs.
9	0.0358 secs.	0.0000793 secs.	**0.0000205 secs.**	0.0000349 secs.	0.0000906 secs.
11	0.150 secs.	0.0000815 secs.	**0.0000208 secs.**	0.0000371 secs.	0.0000885 secs.
17	87.4 secs.	0.0000814 secs.	**0.0000206 secs.**	0.0000362 secs.	0.0000892 secs.
21	6.06 hours	0.0000906 secs.	**0.0000213 secs.**	0.0000378 secs.	0.0000890 secs.
51	not possible	0.0000987 secs.	0.0000205 secs.	0.0000371 secs.	0.0000928 secs.
101	not possible	0.000124 secs.	0.0000218 secs.	0.0000378 secs.	0.000101 secs.
1001	not possible	0.000516 secs.	0.0000366 secs.	0.0000647 secs.	0.000163 secs.
10001	not possible	0.00430 secs.	0.000226 secs.	0.000242 secs.	0.000746 secs.
100001	not possible	0.0540 secs.	0.00609 secs.	0.00684 secs.	0.0198 secs.

Table E.3. Defuzzification times for test set S. The fastest timings are shown in bold.

	INTERVAL EXHAUSTIVE METHOD			**EIASC**		
NO. OF SLICES	LEFT ENDPOINT	RIGHT ENDPOINT	DEFUZZ. VALUE	LEFT ENDPOINT	RIGHT ENDPOINT	DEFUZZ. VALUE
5	0.96538217	0.99851014	0.98197553	0.96555156	0.99850169	0.98202663
9	0.93020229	0.96943988	0.94802558	0.93041860	0.96930754	0.94986307
11	0.92393723	0.96265983	0.94111644	0.92413641	0.96252082	0.94332861
17	0.91322991	0.95337626	0.93093574	0.91440515	0.91425178	0.91432846
21	0.91003500	0.95022604	0.92738683	0.91103067	0.91089908	0.91096488

Table E.4. For test set S, left endpoints, right endpoints and defuzzified values obtained firstly by interval exhaustive defuzzification, and secondly by EIASC.

Appendix F

Interval Test Set U

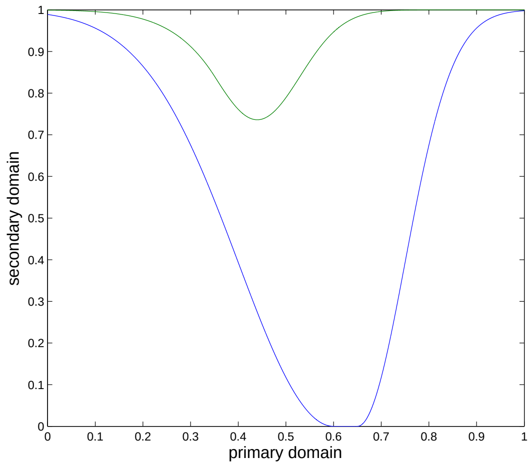


Fig. F.1. Interval Test Set U.

NUMBER OF SLICES	EXHAUSTIVE DEFUZZIFI-CATION	COLLAPSING OUTWARD RIGHT-LEFT	EIASC	NIE-TAN DEFUZZIFI-CATION	WU-MENDEL DEFUZZIFI-CATION
5	0.4883524681	0.4883390379	0.4872834960	0.4898381531	0.4886512972
9	0.4901791343	0.4901282562	0.4882247910	0.4912375749	0.4898251596
11	0.4897646372	0.4897114396	0.4874545055	0.4907033605	0.4891705815
17	0.4896410219	0.4895948384	0.4868477729	0.4902778680	0.4886038758
21	0.4895764854	0.4895352299	0.4866154107	0.4901020305	0.4883730907
51	not possible	0.4894304615	0.4857662498	0.4896778953	0.4878096286
101	not possible	0.4894012488	0.4855181130	0.4895287090	0.4876103532
1001	not possible	0.4893775680	0.4852734241	0.4893906649	0.4874255045
10001	not possible	0.4893753479	0.4852486294	0.4893766612	0.4874067277
100001	not possible	0.4893751274	0.4852461485	0.4893752588	0.4874048471

Table F.1. Defuzzified values for test set U.

NUMBER OF SLICES	EXHAUSTIVE DEFUZZIFI-CATION	ERROR CORL	ERROR EIASC	ERROR NIE-TAN METHOD	ERROR WU-MENDEL APPROX.
5	0.4883524681	**0.0000134302**	0.0010689721	0.0014856850	0.0002988291
9	0.4901791343	**0.0000508781**	0.0019543433	0.0010584406	0.0003539747
11	0.4897646372	**0.0000531976**	0.0023101317	0.0009387233	0.0005940557
17	0.4896410219	**0.0000461835**	0.0027932490	0.0006368461	0.0010371461
21	0.4895764854	**0.0000412555**	0.0029610747	0.0005255451	0.0012033947

Table F.2. Errors for test set U. The lowest errors are shown in bold.

NO. OF SLICES	EXHAUST. DEFUZZIFI-CATION	COLLAPSING OUTWARD RIGHT-LEFT	EIASC	NIE-TAN DEFUZZIFI-CATION	WU-MENDEL DEFUZZIFI-CATION
5	0.00216 secs.	0.0000771 secs.	**0.0000197 secs.**	0.0000371 secs.	0.0000911 secs.
9	0.0361 secs.	0.0000789 secs.	**0.0000204 secs.**	0.0000359 secs.	0.0000926 secs.
11	0.151 secs.	0.0000800 secs.	**0.0000210 secs.**	0.0000365 secs.	0.0000912 secs.
17	87.2 secs.	0.0000834 secs.	**0.0000223 secs.**	0.0000354 secs.	0.0000974 secs.
21	6.17 hours	0.0000872 secs.	**0.0000242 secs.**	0.0000381 secs.	0.0000929 secs.
51	not possible	0.000103 secs.	0.0000310 secs.	0.0000367 secs.	0.0000938 secs.
101	not possible	0.000124 secs.	0.0000390 secs.	0.0000408 secs.	0.000103 secs.
1001	not possible	0.000525 secs.	0.000220 secs.	0.0000609 secs.	0.000209 secs.
10001	not possible	0.00436 secs.	0.00195 secs.	0.000281 secs.	0.000754 secs.
100001	not possible	0.0541 secs.	0.0253 secs.	0.00673 secs.	0.0202 secs.

Table F.3. Defuzzification times for test set U. The fastest timings are shown in bold.

	INTERVAL EXHAUSTIVE METHOD			**EIASC**		
NO. OF SLICES	LEFT ENDPOINT	RIGHT ENDPOINT	DEFUZZ. VALUE	LEFT ENDPOINT	RIGHT ENDPOINT	DEFUZZ. VALUE
5	0.45928005	0.51528694	0.48835247	0.45928005	0.51528694	0.48728350
9	0.45659412	0.51985546	0.49017913	0.45659412	0.51985546	0.48822479
11	0.45441529	0.52049372	0.48976464	0.45441529	0.52049372	0.48745451
17	0.45235204	0.52134351	0.48964102	0.45235204	0.52134351	0.48684777
21	0.45163372	0.52159710	0.48957649	0.45163372	0.52159710	0.48661541

Table F.4. For test set U, left endpoints, right endpoints and defuzzified values obtained firstly by interval exhaustive defuzzification, and secondly by EIASC.

Appendix G

Interval Test Set W

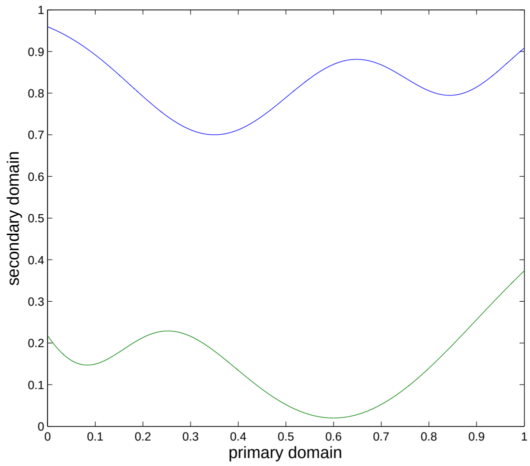


Fig. G.1. Interval Test Set W.

NUMBER OF SLICES	EXHAUSTIVE DEFUZZIFI-CATION	COLLAPSING OUTWARD RIGHT-LEFT	EIASC	NIE-TAN DEFUZZIFI-CATION	WU-MENDEL DEFUZZIFI-CATION
5	0.5113179745	0.5113040394	0.5161402998	0.5077561025	0.5105303188
9	0.5061788619	0.5063274201	0.5073309871	0.5049025262	0.5069806367
11	0.5053236036	0.5054813865	0.5058808491	0.5044228942	0.5062420655
17	0.5040847945	0.5042208959	0.5038132389	0.5036577343	0.5050172344
21	0.5036907192	0.5038093313	0.5031529409	0.5033898118	0.5045764132
51	not possible	0.5028524838	0.5016290930	0.5027185541	0.5034475969
101	not possible	0.5025461348	0.5011408418	0.5024854899	0.5030479895
1001	not possible	0.5022770301	0.5007101291	0.5022715645	0.5026778778
10001	not possible	0.5022504912	0.5006675189	0.5022499508	0.5026403102
100001	not possible	0.5022478411	0.5006632626	0.5022477872	0.5026365478

Table G.1. Defuzzified values for test set W.

NUMBER OF SLICES	EXHAUSTIVE DEFUZZIFI-CATION	ERROR CORL	ERROR EIASC	ERROR NIE-TAN METHOD	ERROR WU-MENDEL APPROX.
5	0.5113179745	**0.0000139351**	0.0048223253	0.0035618720	0.0007876557
9	0.5061788619	**0.0001485582**	0.0011521252	0.0012763357	0.0008017748
11	0.5053236036	**0.0001577829**	0.0005572455	0.0009007094	0.0009184619
17	0.5040847945	**0.0001361014**	0.0002715556	0.0004270602	0.0009324399
21	0.5036907192	**0.0001186121**	0.0005377783	0.0003009074	0.0008856940

Table G.2. Errors for test set W. The lowest errors are shown in bold.

NO. OF SLICES	EXHAUST. DEFUZZIFI-CATION	COLLAPSING OUTWARD RIGHT-LEFT	EIASC	NIE-TAN DEFUZZIFI-CATION	WU-MENDEL DEFUZZIFI-CATION
5	0.00216 secs.	0.0000775 secs.	**0.0000193 secs.**	0.0000355 secs.	0.0000895 secs.
9	0.0360 secs.	0.0000800 secs.	**0.0000192 secs.**	0.0000356 secs.	0.0000902 secs.
11	0.151 secs.	0.0000810 secs.	**0.0000192 secs.**	0.0000354 secs.	0.0000923 secs.
17	87.8 secs.	0.0000831 secs.	**0.0000199 secs.**	0.0000359 secs.	0.0000898 secs.
21	6.17 hours	0.0000852 secs.	**0.0000204 secs.**	0.0000374 secs.	0.0000920 secs.
51	not possible	0.0000977 secs.	0.0000222 secs.	0.0000386 secs.	0.0000925 secs.
101	not possible	0.000127 secs.	0.0000224 secs.	0.0000399 secs.	0.0000991 secs.
1001	not possible	0.000537 secs.	0.0000417 secs.	0.0000624 secs.	0.000160 secs.
10001	not possible	0.00434 secs.	0.000174 secs.	0.000257 secs.	0.000768 secs.
100001	not possible	0.0537 secs.	0.00424 secs.	0.00653 secs.	0.0196 secs.

Table G.3. Defuzzification times for test set W. The fastest timings are shown in bold.

	INTERVAL EXHAUSTIVE METHOD			**EIASC**		
NO. OF SLICES	LEFT ENDPOINT	RIGHT ENDPOINT	DEFUZZ. VALUE	LEFT ENDPOINT	RIGHT ENDPOINT	DEFUZZ. VALUE
5	0.29432676	0.72152802	0.51131797	0.60522021	0.42706039	0.51614030
9	0.29549724	0.71268849	0.50617886	0.55401150	0.46065048	0.50733099
11	0.29747736	0.70777990	0.50532360	0.54356299	0.46819871	0.50588085
17	0.29760690	0.70363458	0.50408479	0.52768133	0.47994515	0.50381324
21	0.29823131	0.70122804	0.50369072	0.52233190	0.48397398	0.50315294

Table G.4. For test set W, left endpoints, right endpoints and defuzzified values obtained firstly by interval exhaustive defuzzification, and secondly by EIASC.

Appendix H

Interval Test Set X

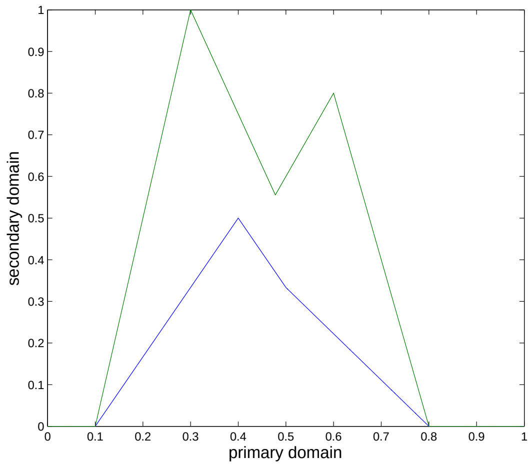


Fig. H.1. Interval Test Set X.

NUMBER OF SLICES	EXHAUSTIVE DEFUZZIFI-CATION	COLLAPSING OUTWARD RIGHT-LEFT	EIASC	NIE-TAN DEFUZZIFI-CATION	WU-MENDEL DEFUZZIFI-CATION
5	0.4213200635	0.4210574784	0.4178281069	0.4149746193	0.4232352457
9	0.4343761342	0.4338356112	0.4344020217	0.4346006144	0.4391219149
11	0.4323638373	0.4318365393	0.4321428571	0.4322643343	0.4373876062
17	0.4312767130	0.4309748555	0.4317442643	0.4312318453	0.4364261865
21	0.4322207919	0.4319600819	0.4325327375	0.4321864324	0.4372554986
51	not possible	0.4319937098	0.4325997229	0.4320857791	0.4371386472
101	not possible	0.4320360065	0.4326071818	0.4320824405	0.4371345749
1001	not possible	0.4320747859	0.4326102449	0.4320794687	0.4371311188
10001	not possible	0.4320789692	0.4326101896	0.4320794379	0.4371310833
100001	not possible	0.4320793907	0.4326101896	0.4320794376	0.4371310830

Table H.1. Defuzzified values for test set X.

NUMBER OF SLICES	EXHAUSTIVE DEFUZZIFI-CATION	ERROR CORL	ERROR EIASC	ERROR NIE-TAN METHOD	ERROR WU-MENDEL APPROX.
5	0.4213200635	**0.0002625851**	0.0034919566	0.0063454442	0.0019151822
9	0.4343761342	0.0005405230	**0.0000258875**	0.0002244802	0.0047457807
11	0.4323638373	0.0005272980	0.0002209802	**0.0000995030**	0.0050237689
17	0.4312767130	0.0003018575	0.0004675513	**0.0000448677**	0.0051494735
21	0.4322207919	0.0002607100	0.0003119456	**0.0000343595**	0.0050347067

Table H.2. Errors for test set X. The lowest errors are shown in bold.

NO. OF SLICES	EXHAUST. DEFUZZIFI-CATION	COLLAPSING OUTWARD RIGHT-LEFT	EIASC	NIE-TAN DEFUZZIFI-CATION	WU-MENDEL DEFUZZIFI-CATION
5	0.00215 secs.	0.0000763 secs.	**0.0000201 secs.**	0.0000356 secs.	0.0000897 secs.
9	0.0362 secs.	0.0000802 secs.	**0.0000202 secs.**	0.0000357 secs.	0.0000894 secs.
11	0.152 secs.	0.0000810 secs.	**0.0000209 secs.**	0.0000356 secs.	0.0000904 secs.
17	89.3 secs.	0.0000820 secs.	**0.0000223 secs.**	0.0000361 secs.	0.0000901 secs.
21	6.15 hours	0.0000845 secs.	**0.0000235 secs.**	0.0000367 secs.	0.0000939 secs.
51	not possible	0.000102 secs.	0.0000296 secs.	0.0000385 secs.	0.0000935 secs.
101	not possible	0.000120 secs.	0.0000387 secs.	0.0000385 secs.	0.0000993 secs.
1001	not possible	0.000503 secs.	0.000206 secs.	0.0000593 secs.	0.000164 secs.
10001	not possible	0.00435 secs.	0.00188 secs	0.000249 secs.	0.000764 secs.
100001	not possible	0.0540 secs.	0.0240 secs.	0.00689 secs.	0.0196 secs.

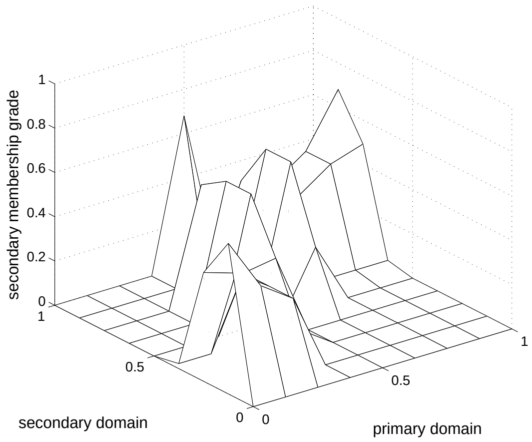
Table H.3. Defuzzification times for test set X. The fastest timings are shown in bold.

	INTERVAL EXHAUSTIVE METHOD			**EIASC**		
NO. OF SLICES	LEFT ENDPOINT	RIGHT ENDPOINT	DEFUZZ. VALUE	LEFT ENDPOINT	RIGHT ENDPOINT	DEFUZZ. VALUE
5	0.34756098	0.48809524	0.42132006	0.34756098	0.48809524	0.41782811
9	0.36775362	0.50105042	0.43437613	0.36775362	0.50105042	0.43440202
11	0.36666667	0.49761905	0.43236384	0.36666667	0.49761905	0.43214286
17	0.36521967	0.49826886	0.43127671	0.36521967	0.49826886	0.43174426
21	0.36628499	0.49878049	0.43222079	0.36628499	0.49878049	0.43253274

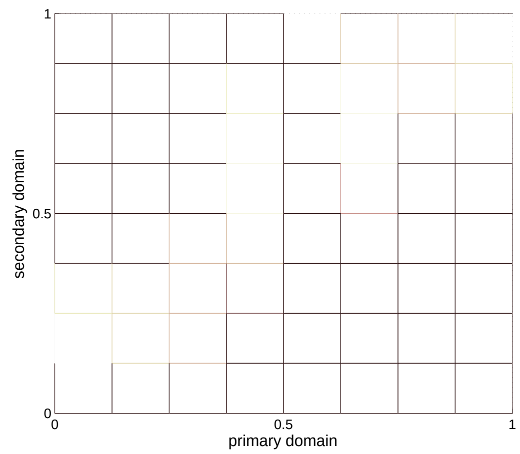
Table H.4. For test set X, left endpoints, right endpoints and defuzzified values obtained firstly by interval exhaustive defuzzification, and secondly by EIASC.

Appendix I

Generalised Test Set Heater0.125



(a) 3-D representation.



(b) FOU.

Fig. I.1. HeaterFIS0.125 — Heater FIS generated generalised test set, domain degree of discretisation 0.125.

EXHAUSTIVE DEFUZZIFIED VALUE	NO. OF EMB. SETS	NO. OF NON-RED. EMB. SETS	EXHAUST-IVE TIMING	VSCTR DEFUZZIFIED VALUE	VSCTR ERROR	VSCTR TIMING
0.6313618377	14580	486	1.37 secs.	0.6327431582	0.001381321	0.000268 secs.

Table I.1. Exhaustive and VSCTR results for the HeaterFIS0.125 test set.

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 secs.
100	0.69%	0.6255373882	0.0058244495 ◁	0.0377 secs.
250	1.71%	0.6299440373	0.0014178004 ◁	0.0937 secs.
500	3.43%	0.6262109521	0.0051508856	0.188 secs.
750	5.14%	0.6263047645	0.0050570732	0.282 secs.
1000	6.86%	0.6246724480	0.0066893897	0.377 secs.
5000	34.29%	0.6251282506	0.0062335871	1.93 secs.
10000	68.59%	0.6256899730	0.0056718647	4.03 secs.
50000	342.94%	0.6252882201	0.0060736176	39.1 secs.
100000	685.87%	0.6254891164	0.0058727213	2.34 mins.

Table I.2. Sampling results for the HeaterFIS0.125 test set. Number of embedded sets = 14580. 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-AGE OF EMB. SETS SAMPLED	NUMBER OF NRESS	NRESS AS PERCENT-AGE OF SAMPLE SIZE	NRESS AS PERCENT-AGE OF ALL ESS	ELITE SAMPLING DEFUZZIFIED VALUE	ELITE SAMPLING METHOD ERROR	ELITE SAMPLING METHOD TIMING
50	0.34%	43	86.00%	0.29%	0.6167008518	0.0146609859	0.0216 secs.
100	0.69%	80	80.00%	0.55%	0.6228289822	0.0085328555	0.0432 secs.
250	1.71%	147	58.80%	1.01%	0.6295092983	0.0018525394	0.108 secs.
500	3.43%	221	44.20%	1.52%	0.6279777881	0.0033840496 ◁	0.215 secs.
750	5.14%	252	33.60%	1.73%	0.6295741240	0.0017877137 ◁	0.328 secs.
1000	6.86%	278	27.80%	1.91%	0.6293312286	0.0020306091 ◁	0.432 secs.
5000	34.29%	408	8.16%	2.80%	0.6306797744	**0.0006820633** ◁	2.16 secs.
10000	68.59%	438	4.38%	3.00%	0.6310563647	**0.0003054730** ◁	5.01 secs.
50000	342.94%	486	0.97%	3.33%	0.6311913249	**0.0001705128** ◁	22.8 secs.
100000	685.87%	486	0.49%	3.33%	0.6313618377	**0.0000000000** ◁	42.9 secs.

Table I.3. Elite sampling results for the HeaterFIS0.125 test set. Number of embedded sets = 14580. 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.

NUMBER OF α-PLANES	NUMBER OF α-PLANES USED	α-PLANES/ CORL DEFUZZIFIED VALUE	α-PLANES/ CORL ERROR	α-PLANES/ CORL TIMING
3	3	0.5974411770	-0.0339206607	0.000566 secs.
5	4	0.6014928819	-0.0298689558	0.000804 secs.
9	7	0.6220020252	-0.0093598125	0.00154 secs.
11	8	0.6202108548	-0.0111509829	0.00178 secs.
21	15	0.6176529687	-0.0137088690	0.00346 secs.
51	36	0.6149638697	-0.0163979680	0.00851 secs.
101	70	0.6146818722	-0.0166799655	0.0169 secs.
1001	682	0.6149166283	-0.0164452094	0.166 secs.
10001	6808	0.6149818425	-0.0163799952	1.77 secs.
100001	68061	0.6149818643	-0.0163799734	59.6 secs.

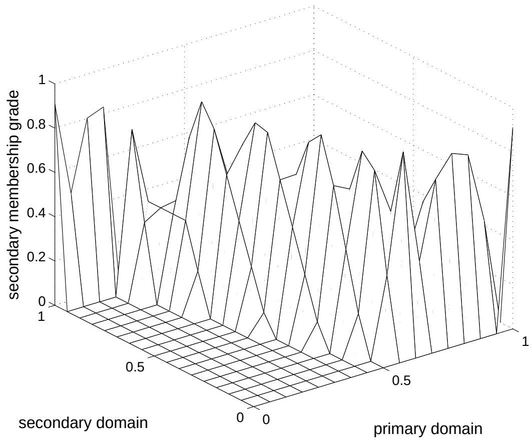
Table I.4. α-planes/CORL results for the HeaterFIS0.125 test set. Exhaustive defuzzified value = 0.6313618377. Error = α-planes/CORL - exhaustive value.

NUMBER OF α-PLANES	NUMBER OF α-PLANES USED	α-PLANES/ INT. EXHAUSTIVE DEFUZZIFIED VALUE	α-PLANES/ INTERVAL EXHAUSTIVE ERROR
3	3	0.5974395543	-0.0339222834
5	4	0.6014844463	-0.0298773914
9	7	0.6219954766	-0.0093663611
11	8	0.6202019617	-0.0111598760
21	15	0.6176441546	-0.0137176831
51	36	0.6149552604	-0.0164065773
101	70	0.6146732228	-0.0166886149
1001	682	0.6149079069	-0.0164539308
10001	6808	0.6149731309	-0.0163887068
100001	68061	0.6149731532	-0.0163886845

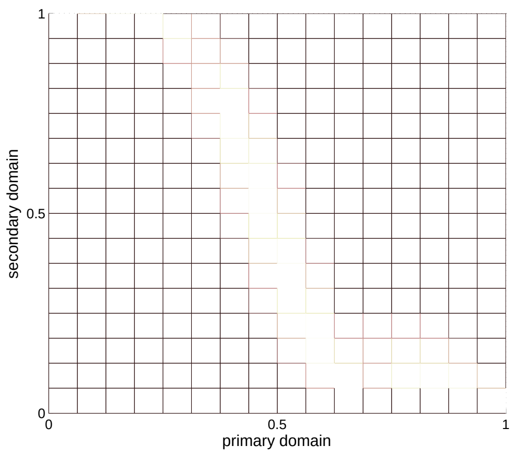
Table I.5. α-planes/interval exhaustive results for the HeaterFIS0.125 test set. Exhaustive defuzzified value = 0.6313618377. Error = α-planes/interval exhaustive value - exhaustive value.

Appendix J

Generalised Test Set Heater0.0625



(a) 3-D representation.



(b) FOU.

Fig. J.1. HeaterFIS0.0625 — Heater FIS generated generalised test set, domain degree of discretisation 0.0625.

EXHAUST. DEFUZZ. VALUE	NO. OF EMB. SETS	NO. OF NON-RED. EMB. SETS	EXHAUST-IVE TIMING	VSCTR DEFUZZ. VALUE	VSCTR ERROR	VSCTR TIMING
0.2621587894	13778100	2774	25.1 mins.	0.2592117473	0.0029470421	0.000453 secs.

Table J.1. Exhaustive and VSCTR results for the HeaterFIS0.0625 test set.

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 secs.
100	0.0007%	0.2643678735	**0.0022090841** ◁	0.0609 secs.
250	0.0018%	0.2639954015	**0.0018366121** ◁	0.152 secs.
500	0.0036%	0.2644544522	**0.0022956628** ◁	0.305 secs.
750	0.0054%	0.2641746630	**0.0020158736** ◁	0.458 secs.
1000	0.0073%	0.2646109558	**0.0024521664** ◁	0.609 secs.
5000	0.0363%	0.2645765948	**0.0024178054**	3.11 secs.
10000	0.0726%	0.2645380675	**0.0023792781**	6.38 secs.
50000	0.3629%	0.2644304187	**0.0022716293**	51.9 secs.
100000	0.7258%	0.2645136689	**0.0023548795**	2.74 mins.

Table J.2. Sampling results for the HeaterFIS0.0625 test set. Number of embedded sets = 13778100. 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	PERCENTAGE OF EMB. SETS SAMPLED	NUMBER OF NRESS	NRESS AS PERCENTAGE OF SAMPLE SIZE	NRESS AS PERCENTAGE OF ALL 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 secs.
100	0.0007%	94	94.00%	0.0007%	0.2650536879	**0.0028948985**	0.0660 secs.
250	0.0018%	217	86.8o%	0.0016%	0.2654055130	0.0032467236	0.165 secs.
500	0.0036%	358	71.60%	0.0026%	0.2645029187	**0.0023441293**	0.330 secs.
750	0.0054%	491	65.47%	0.0036%	0.2642948052	**0.0021360158**	0.495 secs.
1000	0.0073%	606	60.60%	0.0044%	0.2646455386	**0.0024867492**	0.661 secs.
5000	0.0363%	1140	22.80%	0.0083%	0.2637295835	**0.0015707941** ◁	3.32 secs.
10000	0.0726%	1355	13.55%	0.0098%	0.2635483884	**0.0013895990** ◁	6.67 secs.
50000	0.3629%	1809	3.62%	0.0131%	0.2631277384	**0.0009689490** ◁	33.5 secs.
100000	0.7258%	1958	1.96%	0.0142%	0.2629459523	**0.0007871629** ◁	1.12 mins.

Table J.3. Elite sampling results for the HeaterFIS0.0625 test set. Number of embedded sets = 13778100. 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.

NUMBER OF α-PLANES	NUMBER OF α-PLANES USED	α-PLANES/ CORL DEFUZZIFIED VALUE	α-PLANES/ CORL ERROR	α-PLANES/ CORL TIMING
3	3	0.2911992286	0.0290404392	0.000900 secs.
5	5	0.2843138916	0.0221551022	0.00170 secs.
9	9	0.2781833083	0.0160245189	0.00329 secs.
11	11	0.2791783831	0.0170195937	0.00408 secs.
21	20	0.2839726877	0.0218138983	0.00769 secs.
51	47	0.2845058809	0.0223470915	0.0185 secs.
101	92	0.2857499961	0.0235912067	0.0365 secs.
1001	911	0.2836509843	0.0214921949	0.367 secs.
10001	9097	0.2835417182	0.0213829288	3.88 secs.
100001	90961	0.2835490870	0.0213902976	1.94 mins.

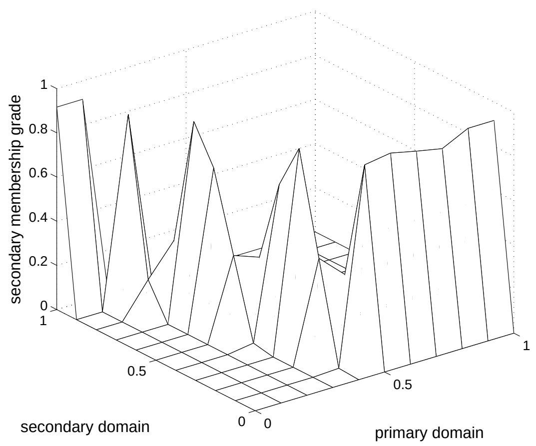
Table J.4. α-planes/CORL results for the HeaterFIS0.0625 test set. Exhaustive defuzzified value = 0.2621587894. Error = α-planes/CORL - exhaustive value.

NUMBER OF α-PLANES	NUMBER OF α-PLANES USED	α-PLANES/ INT. EXHAUSTIVE DEFUZZIFIED VALUE	α-PLANES/ INTERVAL EXHAUSTIVE ERROR
3	3	0.2912056106	0.0290468212
5	5	0.2843202930	0.0221615036
9	9	0.2781887468	0.0160299574
11	11	0.2791839651	0.0170251757
21	20	0.2839784863	0.0218196969
51	47	0.2845118383	0.0223530489
101	92	0.2857559640	0.0235971746
1001	911	0.2836568708	0.0214980814
10001	—	—	—
100001	—	—	—

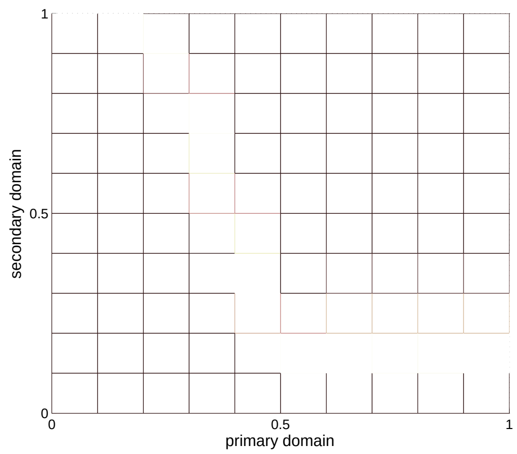
Table J.5. α-planes/interval exhaustive results for the HeaterFIS0.0625 test set. Exhaustive defuzzified value = 0.2621587894. Error = α-planes/interval exhaustive value - exhaustive value.

Appendix K

Generalised Test Set Powder0.1



(a) 3-D representation.



(b) FOU.

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

EXHAUSTIVE DEFUZZIFIED VALUE	NO. OF EMB. SETS	NO. OF NON-RED. EMB. SETS	EXHAUST-IVE TIMING	VSCTR DEFUZZIFIED VALUE	VSCTR ERROR	VSCTR TIMING
0.2806983775	24300	1701	2.55 secs.	0.2646964681	0.0160019094	0.000310 secs.

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

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 secs.
100	0.41%	0.2983068036	0.0176084261	0.0453 secs.
250	1.03%	0.2879898240	**0.0072914465** $\triangleleft$	0.113 secs.
500	2.06%	0.2883575902	**0.0076592127** $\triangleleft$	0.225 secs.
750	3.09%	0.2904003138	**0.0097019363**	0.340 secs.
1000	4.12%	0.2885932629	**0.0078948854**	0.454 secs.
5000	20.58%	0.2893665435	**0.0086681660**	2.32 secs.
10000	41.15%	0.2894760075	**0.0087776300**	4.84 secs.
50000	205.76%	0.2893699018	**0.0086715243**	43.9 secs.
100000	411.52%	0.2896395345	**0.0089411570**	2.46 mins.

Table K.2. Sampling results for the PowderFIS0.1 test set. Number of embedded sets = 24300. 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 '$\triangleleft$' are lower than the corresponding errors for the elite sampling method.

SAMPLE SIZE	PERCENTAGE OF EMB. SETS SAMPLED	NUMBER OF NRESS	NRESS AS PERCENTAGE OF SAMPLE SIZE	NRESS AS PERCENTAGE OF ALL ESS	ELITE SAMPLING DEFUZZIFIED VALUE	ELITE SAMPLING METHOD ERROR	ELITE SAMPLING METHOD TIMING
50	0.21%	48	96.00%	0.20%	0.2924967750	**0.0117983975** ◁	0.0254 secs.
100	0.41%	94	94.00%	0.39%	0.2888488173	**0.0081504398** ◁	0.0510 secs.
250	1.03%	199	79.60%	0.82%	0.2886109164	**0.0079125389**	0.127 secs.
500	2.06%	360	72.00%	1.48%	0.2889039270	**0.0082055495**	0.254 secs.
750	3.09%	477	63.60%	1.96%	0.2879727936	**0.0072744161** ◁	0.381 secs.
1000	4.12%	567	56.70%	2.33%	0.2884496935	**0.0077513160** ◁	0.512 secs.
5000	20.58%	1120	22.40%	4.61%	0.2838675377	**0.0031691602** ◁	2.56 secs.
10000	41.15%	1366	13.66%	5.62%	0.2829860213	**0.0022876438** ◁	5.13 secs.
50000	205.76%	1661	3.32%	6.84%	0.2807352287	**0.0000368512** ◁	25.8 secs.
100000	411.52%	1698	1.70%	6.99%	0.2807555453	**0.0000571678** ◁	51.7 secs.

Table K.3. Elite sampling results for the PowderFIS0.1 test set. Number of embedded sets = 24300. 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.

NUMBER OF α-PLANES	NUMBER OF α-PLANES USED	α-PLANES/ CORL DEFUZZIFIED VALUE	α-PLANES/ CORL ERROR	α-PLANES/ CORL TIMING
3	3	0.3100683482	0.0293699707	0.000653 secs.
5	5	0.2990422650	0.0183438875	0.00123 secs.
9	9	0.2949801671	**0.0142817896**	0.00238 secs.
11	11	0.2860659413	**0.0053675638** *	0.00296 secs.
21	20	0.2903153362	**0.0096169587**	0.00557 secs.
51	47	0.2928824383	**0.0121840608**	0.0133 secs.
101	93	0.2909066603	**0.0102082828**	0.0267 secs.
1001	918	0.2907821474	**0.0100837699**	0.267 secs.
10001	9168	0.2907215619	**0.0100231844**	2.88 secs.
100001	91671	0.2907192214	**0.0100208439**	1.89 mins.

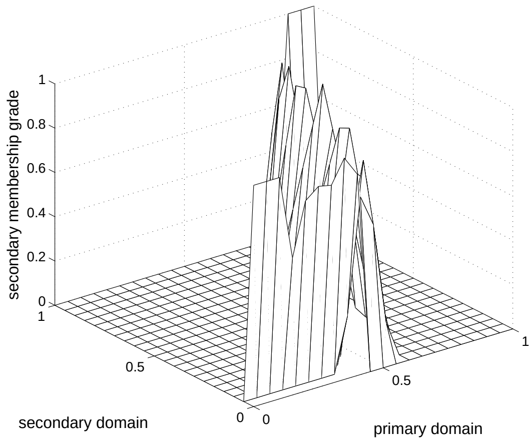
Table K.4. α-planes/CORL results for the PowderFIS0.1 test set. Exhaustive defuzzified value = 0.2806983775. Error = α-planes/CORL - exhaustive value. Errors shown in bold are smaller than the VSCTR error. The error marked '*' is lower than every error for the sampling method.

NUMBER OF α-PLANES	NUMBER OF α-PLANES USED	α-PLANES/ INT. EXHAUSTIVE DEFUZZIFIED VALUE	α-PLANES/ INTERVAL EXHAUSTIVE ERROR
3	3	0.3100714646	0.0293730871
5	5	0.2990446820	0.0183463045
9	9	0.2949820128	0.0142836353
11	11	0.2860677799	0.0053694024
21	20	0.2903173044	0.0096189269
51	47	0.2928844669	0.0121860894
101	93	0.2909086286	0.0102102511
1001	918	0.2907840999	0.0100857224
10001	9168	0.2907235112	0.0100251337
100001	91671	0.2907211701	0.0100227926

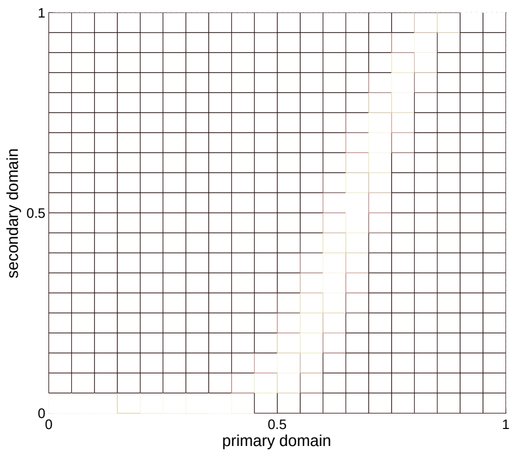
Table K.5. α-planes/interval exhaustive results for the PowderFIS0.1 test set. Exhaustive defuzzified value = 0.2806983775. Error = α-planes/interval exhaustive value - exhaustive value.

Appendix L

Generalised Test Set Powder0.05



(a) 3-D representation.



(b) FOU.

Fig. L.1. PowderFIS0.05 — Powder FIS generated generalised test set, domain degree of discretisation 0.05.

EXHAUSTIVE DEFUZZIFIED VALUE	NO. OF EMB. SETS	NO. OF NON-RED. EMB. SETS	EXHAUST-IVE TIMING	VSCTR DEFUZZ. VALUE	VSCTR ERROR	VSCTR TIMING
0.8180632180	3840000	5093	8.22 mins.	0.8185912163	0.0005279983	0.000555 secs.

Table L.1. Exhaustive and VSCTR results for the PowderFIS0.05 test set.

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 secs.
100	0.003%	0.8173514791	0.0007117389 ◁	0.0648 secs.
250	0.007%	0.8176368830	**0.0004263350** ◁	0.162 secs.
500	0.013%	0.8166109316	0.0014522864	0.323 secs.
750	0.020%	0.8166335918	0.0014296262	0.485 secs.
1000	0.026%	0.8165791599	0.0014840581	0.647 secs.
5000	0.130%	0.8171269807	0.0009362373	3.30 secs.
10000	0.260%	0.8169971802	0.0010660378	6.73 secs.
50000	1.302%	0.8168484040	0.0012148140	54.4 secs.
100000	2.604%	0.8168981632	0.0011650548	2.82 mins.

Table L.2. Sampling results for the PowderFIS0.05 test set. Number of embedded sets = 3840000. 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-AGE OF EMB. SETS SAMPLED	NUMBER OF NRESS	NRESS AS PERCENT-AGE OF SAMPLE SIZE	NRESS AS PERCENT-AGE OF ALL 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 secs.
100	0.003%	94	94.00%	0.0024%	0.8164985098	0.0015647082	0.0709 secs.
250	0.007%	216	86.40%	0.0056%	0.8168936251	0.0011695929	0.178 secs.
500	0.013%	406	81.20%	0.0106%	0.8169408395	0.0011223785 ◁	0.355 secs.
750	0.020%	550	73.33%	0.0143%	0.8168162196	0.0012469984 ◁	0.533 secs.
1000	0.026%	673	67.30%	0.0175%	0.8170654905	0.0009977275 ◁	0.711 secs.
5000	0.130%	1595	31.90%	0.0415%	0.8171645726	0.0008986454 ◁	3.59 secs.
10000	0.260%	2029	20.29%	0.0528%	0.8173314906	0.0007317274 ◁	7.21 secs.
50000	1.302%	3026	6.05%	0.0788%	0.8175959636	**0.0004672544** ◁	36.3 secs.
100000	2.604%	3439	3.44%	0.0896%	0.8177743683	**0.0002888497** ◁	1.22 mins.

Table L.3. Elite sampling results for the PowderFIS0.05 test set. Number of embedded sets = 3840000. 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.

NUMBER OF α-PLANES	NUMBER OF α-PLANES USED	α-PLANES/ CORL DEFUZZIFIED VALUE	α-PLANES/ CORL ERROR	α-PLANES/ CORL TIMING
3	3	0.8371816462	0.0191184282	0.00148 secs.
5	5	0.8132243650	-0.0048388530	0.00244 secs.
9	9	0.8003904509	-0.0176727671	0.00433 secs.
11	11	0.8028981616	-0.0151650564	0.00529 secs.
21	21	0.8000431818	-0.0180200362	0.0101 secs.
51	51	0.7987563133	-0.0193069047	0.0243 secs.
101	101	0.7983826038	-0.0196806142	0.0483 secs.
1001	1001	0.7974846584	-0.0205785596	0.479 secs.
10001	10001	0.7974345629	-0.0206286551	50.5 secs.
100001	100001	0.7974291278	-0.0206340902	2.46 mins.

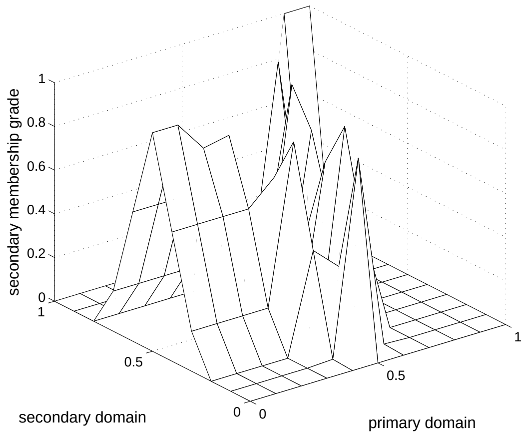
Table L.4. α-planes/CORL results for the PowderFIS0.05 test set. Exhaustive defuzzified value = 0.8180632180. Error = α-planes/CORL - exhaustive value.

NUMBER OF α-PLANES	NUMBER OF α-PLANES USED	α-PLANES/ INT. EXHAUSTIVE DEFUZZIFIED VALUE	α-PLANES/ INTERVAL EXHAUSTIVE ERROR
3	3	0.8371808680	0.0191176500
5	5	0.8132227384	-0.0048404796
9	9	0.8003883556	-0.0176748624
11	11	0.8028960507	-0.0151671673
21	21	0.8000408574	-0.0180223606
51	51	0.7987538800	-0.0193093380
101	101	0.7983801575	-0.0196830605
1001	1001	0.7974821984	-0.0205810196
10001	—	—	—
100001	—	—	—

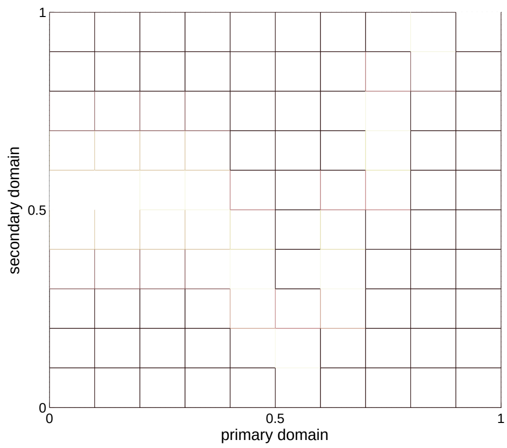
Table L.5. α-planes/interval exhaustive results for the PowderFIS0.05 test set. Exhaustive defuzzified value = 0.8180632180. Error = α-planes/interval exhaustive value - exhaustive value.

Appendix M

Generalised Test Set Shopping0.1



(a) 3-D representation.



(b) FOU.

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

EXHAUSTIVE DEFUZZIFIED VALUE	NO. OF EMB. SETS	NO. OF NON-RED. EMB. SETS	EXHAUST-IVE TIMING	VSCTR DEFUZZ. VALUE	VSCTR ERROR	VSCTR TIMING
0.5954109472	312500	2495	32.9 secs.	0.5939161160	0.0014948312	0.000315 secs.

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

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 secs.
100	0.03%	0.5905449544	0.0048659928	0.0434 secs.
250	0.08%	0.5926005506	0.0028103966 $\triangleleft$	0.108 secs.
500	0.16%	0.5926817464	0.0027292008	0.218 secs.
750	0.24%	0.5923095537	0.0031013935	0.325 secs.
1000	0.32%	0.5934992219	0.0019117253	0.435 secs.
5000	1.60%	0.5931185649	0.0022923823	2.23 secs.
10000	3.20%	0.5929055726	0.0025053746	4.60 secs.
50000	16.00%	0.5933037587	0.0021071885	42.4 secs.
100000	32.00%	0.5933184632	0.0020924840	2.43 mins.

Table M.2. Sampling results for the ShoppingFIS0.1 test set. Number of embedded sets = 312500. Percentage of embedded sets sampled = $\frac{\text{sample size}}{\text{number of embedded sets}} \times 100$. Exhaustive defuzzified value = 0.5954109472. Errors marked '$\triangleleft$' are lower than the corresponding errors for the elite sampling method.

SAMPLE SIZE	PERCENT-AGE OF EMB. SETS SAMPLED	NUMBER OF NRESS	NRESS AS PERCENT-AGE OF SAMPLE SIZE	NRESS AS PERCENT-AGE OF ALL ESS	ELITE SAMPLING DEFUZZIFIED VALUE	ELITE SAMPLING METHOD ERROR	ELITE SAMPLING METHOD TIMING
50	0.02%	50	100.00%	0.016%	0.5954284597	**0.0000175125** ◁	0.0243 secs.
100	0.03%	94	94.00%	0.030%	0.5935585538	0.0018523934 ◁	0.0485 secs.
250	0.08%	210	84.00%	0.067%	0.5911745884	0.0042363588	0.121 secs.
500	0.16%	368	73.60%	0.118%	0.5954734998	**0.0000625526** ◁	0.243 secs.
750	0.24%	481	64.13%	0.154%	0.5935433933	0.0018675539 ◁	0.366 secs.
1000	0.32%	570	57.00%	0.182%	0.5935158606	0.0018950866 ◁	0.486 secs.
5000	1.60%	1134	22.68%	0.363%	0.5937003262	0.0017106210 ◁	2.45 secs.
10000	3.20%	1401	14.01%	0.448%	0.5948734695	**0.0005374777** ◁	4.92 secs.
50000	16.00%	1943	3.89%	0.622%	0.5949611026	**0.0004498446** ◁	24.8 secs.
100000	32.00%	2146	2.15%	0.687%	0.5952072004	**0.0002037468** ◁	49.9 secs.

Table M.3. Elite sampling results for the ShoppingFIS0.1 test set. Number of embedded sets = 312500. 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.

NUMBER OF α-PLANES	NUMBER OF α-PLANES USED	α-PLANES/ CORL DEFUZZIFIED VALUE	α-PLANES/ CORL ERROR	α-PLANES/ CORL TIMING
3	3	0.6151869952	0.0197760480	0.000911 secs.
5	5	0.6018755341	0.0064645869	0.00148 secs.
9	9	0.5932602572	-0.0021506900	0.00261 secs.
11	11	0.5946487587	**-0.0007621885**	0.00322 secs.
21	21	0.5929872008	-0.0024237464	0.00608 secs.
51	51	0.5920148105	-0.0033961367	0.0146 secs.
101	101	0.5919492352	-0.0034617120	0.0289 secs.
1001	1001	0.5914403564	-0.0039705908	0.288 secs.
10001	10001	0.5914134660	-0.0039974812	3.12 secs.
100001	100001	0.5914058776	-0.0040050696	2.13 mins.

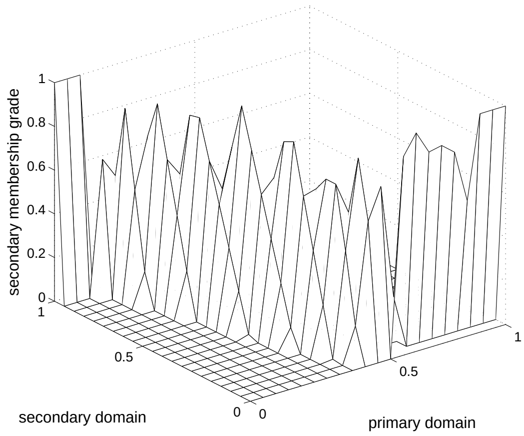
Table M.4. α-planes/CORL results for the ShoppingFIS0.1 test set. Exhaustive defuzzified value = 0.5954109472. Error = α-planes/CORL - exhaustive value. Error shown in bold is smaller than the VSCTR error.

NUMBER OF α-PLANES	NUMBER OF α-PLANES USED	α-PLANES/ INT. EXHAUSTIVE DEFUZZIFIED VALUE	α-PLANES/ INTERVAL EXHAUSTIVE ERROR
3	3	0.6151852147	0.0197742675
5	5	0.6018735720	0.0064626248
9	9	0.5932572151	-0.0021537321
11	11	0.5946460014	-0.0007649458
21	21	0.5929838018	-0.0024271454
51	51	0.5920110566	-0.0033998906
101	101	0.5919454769	-0.0034654703
1001	1001	0.5914366015	-0.0039743457
10001	10001	0.5914097097	-0.0040012375
100001	100001	0.5914021206	-0.0040088266

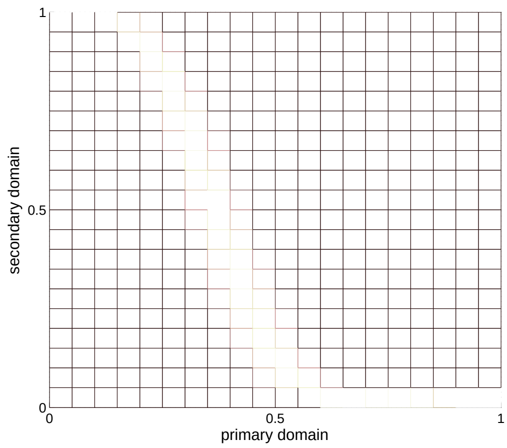
Table M.5. α-planes/interval exhaustive results for the ShoppingFIS0.1 test set. Exhaustive defuzzified value = 0.5954109472. Error = α-planes/interval exhaustive value - exhaustive value.

Appendix N

Generalised Test Set Shopping0.05



(a) 3-D representation.



(b) FOU.

Fig. N.1. ShoppingFIS0.05 — Shopping FIS generated generalised test set, domain degree of discretisation 0.05.

EXHAUST. DEFUZZ. VALUE	NO. OF EMB. SETS	NO. OF NON-RED. EMB. SETS	EXHAUST-IVE TIMING	VSCTR DEFUZZIFIED VALUE	VSCTR ERROR	VSCTR TIMING
0.1821425020	3840000	12347	11.8 mins.	0.1814087837	0.0007337183	0.000552 secs.

Table N.1. Exhaustive and VSCTR results for the ShoppingFIS0.05 test set.

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 secs.
100	0.003%	0.1820101587	**0.0001323433** ◁	0.0656 secs.
250	0.007%	0.1838659287	0.0017234267	0.164 secs.
500	0.013%	0.1831044751	0.0009619731 ◁	0.329 secs.
750	0.020%	0.1829725179	0.0008300159 ◁	0.492 secs.
1000	0.026%	0.1827985154	**0.0006560134** ◁	0.655 secs.
5000	0.130%	0.1830080344	0.0008655324	3.33 secs.
10000	0.260%	0.1831606564	0.0010181544	6.79 secs.
50000	1.302%	0.1830777694	0.0009352674	54.6 secs.
100000	2.604%	0.1830956217	0.0009531197	2.85 mins.

Table N.2. Sampling results for the ShoppingFIS0.05 test set. Number of embedded sets = 3840000. 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-AGE OF EMB. SETS SAMPLED	NUMBER OF NRESS	NRESS AS PERCENT-AGE OF SAMPLE SIZE	NRESS AS PERCENT-AGE OF ALL 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 secs.
100	0.003%	97	97.00%	0.0025%	0.1835334781	0.0013909761	0.0720 secs.
250	0.007%	241	96.40%	0.0063%	0.1820180536	**0.0001244484** ◁	0.180 secs.
500	0.013%	455	91.00%	0.0118%	0.1834734689	0.0013309669	0.360 secs.
750	0.020%	668	89.07%	0.0174%	0.1830600422	0.0009175402	0.542 secs.
1000	0.026%	819	81.90%	0.0213%	0.1828033788	**0.0006608768**	0.723 secs.
5000	0.130%	2538	50.76%	0.0661%	0.1828180409	**0.0006755389** ◁	3.68 secs.
10000	0.260%	3520	35.20%	0.0917%	0.1828462673	**0.0007037653** ◁	7.44 secs.
50000	1.302%	6072	12.14%	0.1581%	0.1826417325	**0.0004992305** ◁	38.2 secs.
100000	2.604%	7209	7.21%	0.1877%	0.1825469839	**0.0004044819** ◁	1.29 mins.

Table N.3. Elite sampling results for the ShoppingFIS0.05 test set. Number of embedded sets = 3840000. 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.

NUMBER OF α-PLANES	NUMBER OF α-PLANES USED	α-PLANES/ CORL DEFUZZIFIED VALUE	α-PLANES/ CORL ERROR	α-PLANES/ CORL TIMING
3	3	0.1628183538	-0.0193241482	0.00153 secs.
5	5	0.1867756350	0.0046331330	0.00248 secs.
9	9	0.1996095491	0.0174670471	0.00442 secs.
11	11	0.1971018384	0.0149593364	0.00542 secs.
21	21	0.1999568182	0.0178143162	0.0103 secs.
51	51	0.2012436867	0.0191011847	0.0248 secs.
101	101	0.2016173962	0.0194748942	0.0496 secs.
1001	1001	0.2025153416	0.0203728396	0.488 secs.
10001	10001	0.2025654371	0.0204229351	5.13 secs.
100001	100001	0.2025708722	0.0204283702	2.44 mins.

Table N.4. α-planes/CORL results for the ShoppingFIS0.05 test set. Exhaustive defuzzified value = 0.1821425020. Error = α-planes/CORL - exhaustive value.

NUMBER OF α-PLANES	NUMBER OF α-PLANES USED	α-PLANES/ INT. EXHAUSTIVE DEFUZZIFIED VALUE	α-PLANES/ INTERVAL EXHAUSTIVE ERROR
3	3	0.1628191321	-0.0193233699
5	5	0.1867772616	0.0046347596
9	9	0.1996116444	0.0174691424
11	11	0.1971039493	0.0149614473
21	21	0.1999591426	0.0178166406
51	51	0.2012461200	0.0191036180
101	101	0.2016198425	0.0194773405
1001	1001	0.2025178016	0.0203752996
10001	—	—	—
100001	—	—	—

Table N.5. α-planes/interval exhaustive results for the ShoppingFIS0.05 test set. Exhaustive defuzzified value = 0.1821425020. Error = α-planes/interval exhaustive value - exhaustive value.