Information Sciences 189 (2012) 77–92

Contents lists available at SciVerse ScienceDirect

Information Sciences

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

The sampling method of defuzzification for type-2 fuzzy sets: Experimental evaluation

Sarah Greenfield, Francisco Chiclana *, Robert John, Simon Coupland

Centre for Computational Intelligence, Faculty of Technology, De Montfort University, Leicester LE1 9BH, UK

ARTICLE INFO

Article history:
Received 31 January 2011
Received in revised form 8 November 2011
Accepted 29 November 2011
Available online 4 December 2011

Keywords:
Type-2 fuzzy set
Defuzzification
Sampling method
Type-reduced set
Type-reduction

ABSTRACT

For generalised type-2 fuzzy sets the defuzzification process has historically been slow and inefficient. This has hampered the development of type-2 Fuzzy Inferencing Systems for real applications and therefore no advantage has been taken of the ability of type-2 fuzzy sets to model higher levels of uncertainty. The research reported here provides a novel approach for improving the speed of defuzzification for discretised generalised type-2 fuzzy sets. The traditional type-reduction method requires every embedded type-2 fuzzy set to be processed. The high level of redundancy in the huge number of embedded sets inspired the development of our sampling method which randomly samples the embedded sets and processes only the sample. The paper presents detailed experimental results for defuzzification of constructed sets of known defuzzified value. The sampling defuzzifier is compared on aggregated type-2 fuzzy sets resulting from the inferencing stage of a FIS, in terms of accuracy and speed, with other methods including the exhaustive and techniques based on the α-planes representation. The results indicate that by taking only a sample of the embedded sets we are able to dramatically reduce the time taken to process a type-2 fuzzy set with very little loss in accuracy.

1. Introduction

The main strength of type-2 fuzzy logic is its ability to deal with the second-order uncertainties that arise from several sources [14], among them the fact that the meanings of words are often vague [21, p. 60]. Most researchers concentrate exclusively on interval secondary membership functions [21,23,30] for which an increasing number of applications are being developed [1,5,10,11,15–18,20,25,26,35]. The Karnik–Mendel Iterative Procedure (KMIP) [13,30] is the established technique for defuzzification of interval sets. The capability of the generalised type-2 paradigm to handle uncertainty is explored in [9]. Regrettably interval type-2 fuzzy sets are not able to model uncertainty as fully as their generalised counterparts, as they lack the crucial variability of the third dimension [21]. Our research, therefore, sees developing *generalised* type-2 systems as a challenge for the research community. A triangular type-2 system with a defuzzification algorithm based on the KMIP has been developed by Starczewski [28]; this goes some way towards achieving our goal. Coupland and John [2,3] have exploited geometry to improve the speed of inferencing in generalised type-2 fuzzy sets. In 2008, Liu et al. [19,24] proposed the α-planes method which involves decomposing a generalised type-2 set into a set of α-planes which are horizontal slices akin to interval type-2 sets. This method is used in conjunction with an interval method such as the KMIP, the Greenfield–Chiclana Collapsing Defuzzifier [6], or the Nie–Tan Method [27]. The α-planes/KMIP method has been modified by Zhai and Mendel [36,37] to increase its efficiency. Experiments have shown that the α-planes decomposition introduces slight

* Corresponding author. Tel.: +44 116 207 8413; fax: +44 116 207 8159.
E-mail addresses: sarahg@dmu.ac.uk (S. Greenfield), chiclana@dmu.ac.uk (F. Chiclana), rij@dmu.ac.uk (R. John), simonc@dmu.ac.uk (S. Coupland).

doi:10.1016/j.ins.2011.11.042

inaccuracies [7]; this is touched on in Section 7.2. Further inaccuracies are introduced by the associated interval method, as all the alternatives (apart from the interval exhaustive method) are approximations [6]. Independently, Wagner and Hagras have introduced the notion of zSlices [29], a concept very similar to α-planes.

There are five stages to any Fuzzy Inferencing System (FIS): fuzzification, antecedent computation, implication, aggregation and defuzzification (Fig. 1). In the case of a type-2 FIS, defuzzification consists of two parts — *type-reduction* and defuzzification proper. Type-reduction is the procedure by which a type-2 fuzzy set is converted to a type-1 fuzzy set. This set is then defuzzified to give a crisp number. The type-2 FIS differs from its type-1 counterpart firstly in that it employs type-2 fuzzy sets, and secondly in there being an extra stage of type-reduction. It is this additional stage of type-reduction that has been a processing bottleneck in type-2 fuzzy inferencing because it relies on finding the centroid of an extraordinarily large number of type-1 fuzzy sets (embedded sets). The purpose of the sampling method of defuzzification is to alleviate this bottleneck.

The traditional type-reduction method requires every embedded type-2 fuzzy set to be processed. The high level of redundancy in the huge number of embedded sets inspired the development of our sampling method which randomly samples the embedded sets and processes only the sample. In this paper we show theoretically that the concept of sampling embedded sets is valid. Test are performed on symmetrical sets of known defuzzified values for the purpose of illustration, confirming the theoretical setting of the sampling algorithm. After this, the method is applied to two asymmetric, FIS generated sets, and the results compared with three other defuzzification techniques. We show, both theoretically and practically, that the sampling method gives a good approximation to type-reduction. It is a direct method i.e. it does not involve horizontal decomposition of the type-2 fuzzy set. In summary, the sampling method is an original, computationally efficient technique which employs randomised sampling during the type-reduction stage.

The remainder of the paper is structured as follows: The next section introduces concepts essential to the understanding of the rest of the paper. Following that, Section 3 discusses the exhaustive method of defuzzification. In Section 4 the sampling method is presented. Section 5 describes the design of experiments to evaluate the sampling defuzzifier, and Section 6 presents and discusses the results of the experiments. In Section 7 it is proved that the sampling method can be employed in practical cases using just one random sample of embedded sets of sufficiently large size for the Central Limit Theorem to be applicable. This section includes a practical application of the method and its comparison with three different defuzzification strategies in terms of accuracy and speed. Section 8 discusses the parameter choices required of the user of the sampling method. Lastly, Section 9 concludes the paper.

2. Preliminaries

To make the paper self-contained, the main concepts that will be used are introduced here.

2.1. Mathematical definition of a type-2 fuzzy set

Let X be a universe of discourse. A fuzzy set A on X is characterised by a membership function $\mu_A : X \to [0,1]$. A fuzzy set A on X can be expressed as follows [31]:

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

An alternative mathematical representation of fuzzy set A with continuous universe of discourse is

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

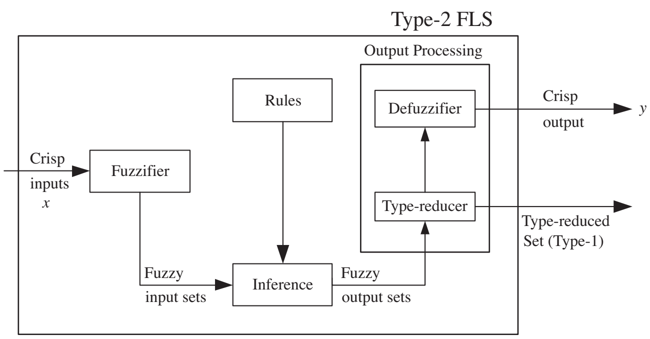


Fig. 1. Type-2 FIS (from [21]).

When the universe of discourse is discrete the fuzzy set A is represented as

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

Note that the membership grades of A are crisp numbers. This type of fuzzy set is also referred to as a type-1 fuzzy set. In the following we will use the notation $U = [0,1]$.

Let $\widetilde{P}(U)$ be the set of fuzzy sets in U. A type-2 fuzzy set $\widetilde{A}$ in X is a fuzzy set whose membership grades are themselves fuzzy [32–34]. This implies that $\mu_{\widetilde{A}}(x)$ is a fuzzy set in U for all x, i.e. $\mu_{\widetilde{A}} : X \to \widetilde{P}(U)$ and

$$\widetilde{A} = \{(x, \mu_{\widetilde{A}}(x));\ \mu_{\widetilde{A}}(x) \in \widetilde{P}(U)\ \forall x \in X\}. \tag{4}$$

It follows that $\forall x \in X\ \exists J_x \subseteq U$ such that $\mu_{\widetilde{A}}(x) : J_x \to U$. Applying (1), we have:

$$\mu_{\widetilde{A}}(x) = \{(u, \mu_{\widetilde{A}}(x)(u));\ \mu_{\widetilde{A}}(x)(u) \in U \forall u \in J_x \subseteq U\}. \tag{5}$$

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

Putting (4) and (5) together we have

$$\widetilde{A} = \{(x, (u, \mu_{\widetilde{A}}(x)(u)));\ \mu_{\widetilde{A}}(x)(u) \in U,\ \forall x \in X \wedge \forall u \in J_x \subseteq U\}. \tag{6}$$

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

2.2. Discretisation

Geometrically the type-2 fuzzy set may be viewed as a surface in space represented by (x,u,z) co-ordinates within a unit cube. Conventionally, discretisation is the first step in creating a computer representation of a fuzzy set (of any type). It is the procedure by which a continuous set is converted into a discrete set through 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 (*Vertical slice*). A *vertical slice* is a plane which intersects the x-axis (primary domain) and is parallel to the u-axis (secondary domain).

Definition 2 (*Footprint of uncertainty*). The *footprint of uncertainty (FOU)* of a type-2 fuzzy set is the projection of the set onto the x–u plane.

Definition 3 (*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 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.

2.2.1. Standard method of discretisation

In this discretisation technique (Fig. 2) 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 [12]. 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.

2.2.2. Grid method of discretisation

An alternative method valid for all type-2 fuzzy sets is the *grid method of discretisation* (Fig. 3). In this straightforward approach the x–u plane 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 [6].[1]

The sampling method of defuzzification is applicable irrespective of the discretisation technique employed. However the grid method was adhered to in the type-2 fuzzy sets prepared for the testing regime described below (Sections 5 and 7).

[1] The grid method is more general than the standard one in that it is applied to the whole theoretical domains X and U, i.e. it is not necessary to identify the actual domains of the membership function $\mu_{\widetilde{A}}$ and the secondary membership functions $\mu_{\widetilde{A}}(x)$ prior to discretisation. However, if the FOU has a narrow section, the discretisation has to be made finer in order to represent the fuzzy set adequately.

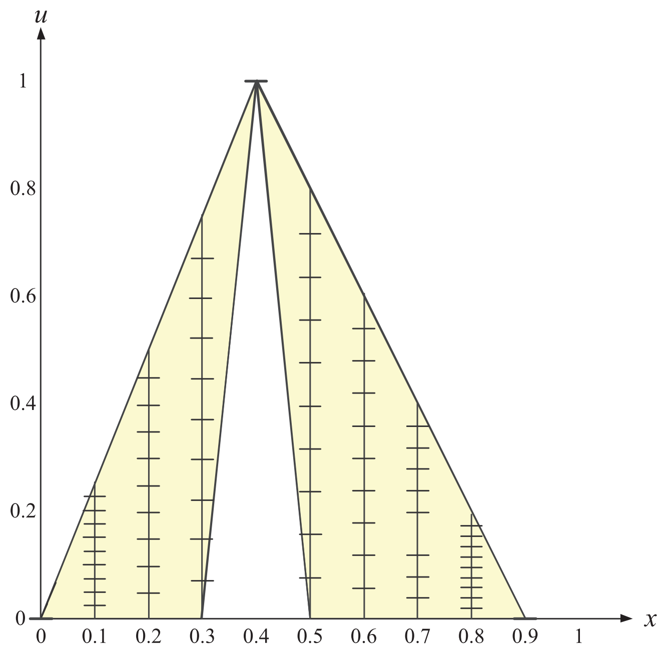


Fig. 2. x–u plane under the standard method of discretisation. Primary domain degree of discretisation = 0.1.

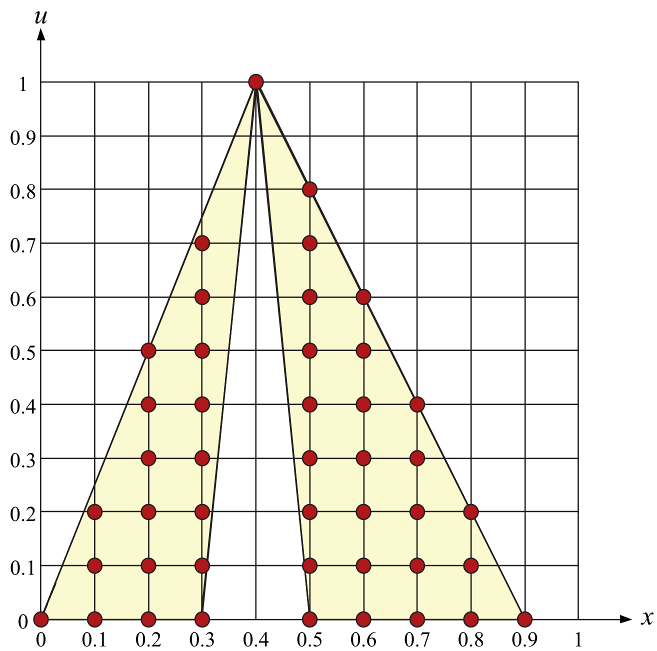


Fig. 3. x–u plane under the grid method of discretisation. Primary domain degree of discretisation = 0.1; secondary domain degree of discretisation = 0.1.

3. Defuzzification of a type-2 fuzzy set

For type-2 fuzzy sets, the defuzzification process has two stages. Firstly, through a procedure known as *type-reduction*, a type-1 set is derived. This set is known as the *type-reduced set (TRS)* [13]. Defuzzifying the type-1 TRS is relatively straightforward, and this is the second stage of type-2 defuzzification.

3.1. Embedded sets

Type-reduction is dependent on the concept of an *embedded type-2 set* [21,22]. An embedded type-2 set (or 'embedded set' for short) is a special kind of type-2 fuzzy set. It relates to the type-2 fuzzy set in which it is embedded in this way: For every primary domain value, x, there is a unique secondary domain value, u, plus the associated secondary membership grade that is determined by the primary and secondary domain values, $\mu_{\widetilde{A}}(x)(u)$.

Example 1. In Fig. 4 we have identified two embedded sets of a type-2 fuzzy set with primary and secondary domain degree of discretisation of 0.1. Embedded set $\widetilde{P}$ is represented by pentagonal, pointed flags, and embedded set $\widetilde{Q}$ is symbolised by quadrilateral shaped flags.

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

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

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

Definition 4 (*Embedded set* [21]). Let $\widetilde{A}$ be a type-2 fuzzy set in X. For discrete universes of discourse X and U, *an embedded type-2 set* $\widetilde{A}_e$ of $\widetilde{A}$ is defined as the following type-2 fuzzy set

$$\widetilde{A}_e = \sum_{i=1}^{N} [\mu_{\widetilde{A}}(x_i)(u_i)/u_i]/x_i u_i \in J_{x_i} \subseteq U \ \wedge \ x_i \in X.$$

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

3.2. Algorithm for type-reduction

The TRS is defined via the application of Zadeh's Extension Principle, and only after the primary domain X has been discretised.

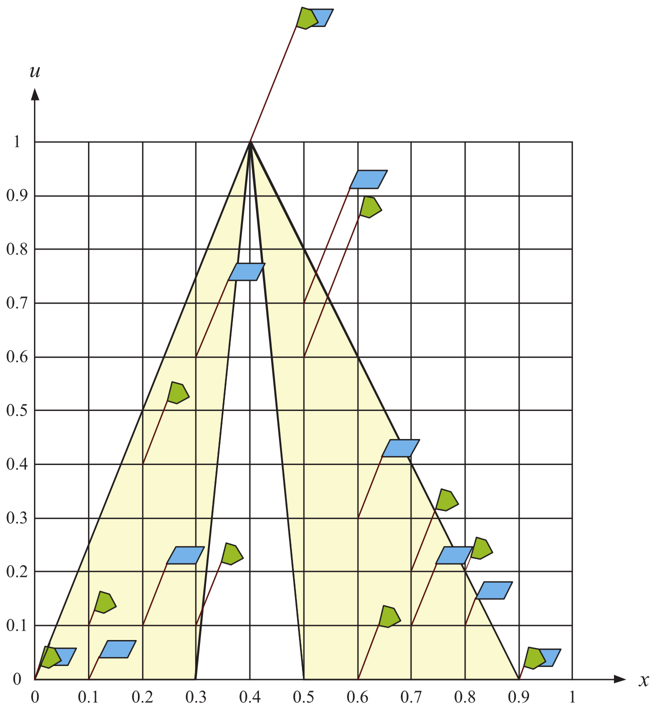


Fig. 4. Two embedded sets, indicated by different flag styles. The flag height reflects the secondary membership grade. The degree of discretisation of primary and secondary domains is 0.1. The shaded region is the FOU.

Definition 5 (*Type-reduced set* [13]). The TRS associated with a type-2 fuzzy set $\widetilde{A}$ with primary domain X discretised into N points is

$$C_{\widetilde{A}} = \int_{u_1 \in J_{x_1}} \cdots \int_{u_N \in J_{x_N}} [\mu_{\widetilde{A}}(x_1)(u_1) * \cdots * \mu_{\widetilde{A}}(x_N)(u_N)] \Bigg/ \frac{\sum_{i=1}^{N} x_i \cdot u_i}{\sum_{i=1}^{N} u_i}. \tag{7}$$

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, the minimum t-norm ($\wedge$) is used [13, p. 201].

Obviously, the above definition of the TRS is meaningful only when X is numeric in nature. The TRS is a type-1 fuzzy set in U and its computation in practice requires the secondary domain U to be discretised as well. Algorithm 1, adapted from Mendel [21, pp. 249–250], is used to compute the TRS of a type-2 fuzzy sets. This stratagem has become known as the *exhaustive method*, as every embedded set is processed [8].

Algorithm 1. Type-reduction of a discretised type-2 fuzzy set to a type-1 fuzzy set.

```
Input: a discretised generalised type-2 fuzzy set
Output: a discrete type-1 fuzzy set
forall embedded sets do
    find the minimum secondary membership grade (z);
    calculate the primary domain value (x) of the type-1 centroid of the type-2 embedded set;
    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};
end
forall primary domain (x) values do
    select the maximum secondary grade {make each x correspond to a unique secondary domain value};
end
```

Example 2. This example relates to the embedded sets introduced in Example 1. Embedded set $\widetilde{P}$ has minimum secondary grade $z_{\widetilde{P}} = 0.1$ and primary domain value of its type-1 centroid $x_{\widetilde{P}} = 0.4308$:

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

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

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

4. The sampling method of defuzzification

4.1. Rationale for the sampling defuzzifier

There are two striking features of a type-2 fuzzy set's embedded sets. The high cardinality of the totality of embedded sets is the reason for the computational bottleneck — the source of the problem we are trying to overcome. Their inherent redundancy points us towards a solution.

A faithful implementation of the type-reduction algorithm (Section 3.2) requires that every embedded set be processed. The number of embedded sets within a type-2 set is $\prod_{i=1}^{N} M_i$, where N is the number of vertical slices into which the primary domain has been discretised, and M_i is the number of elements on the ith slice. On any method of discretisation, the finer the discretisation, the better the representation of a given fuzzy set, but the greater the number of embedded sets generated. A reasonably fine degree of discretisation can give rise to astronomical numbers of embedded sets.[2] For instance, when a prototype type-2 FIS was invoked using a primary and secondary degree of discretisation of 0.02, the number of embedded sets generated was of the order of 2.9×10^{63}. With such enormous numbers of embedded sets, the exhaustive method algorithm that works its way one by one through every embedded set is impractical [7].

Continuous type-2 fuzzy sets are associated with an infinite number of embedded sets, and therefore the centroid values obtained via Algorithm 1 are in fact estimates of the real centroid values. Indeed, discretisation can be seen as a form of sampling of the continuous type-2 fuzzy set. In what follows, we describe the sampling defuzzification algorithm.

[2] For a discretised type-2 fuzzy set, the number of embedded sets is always finite.

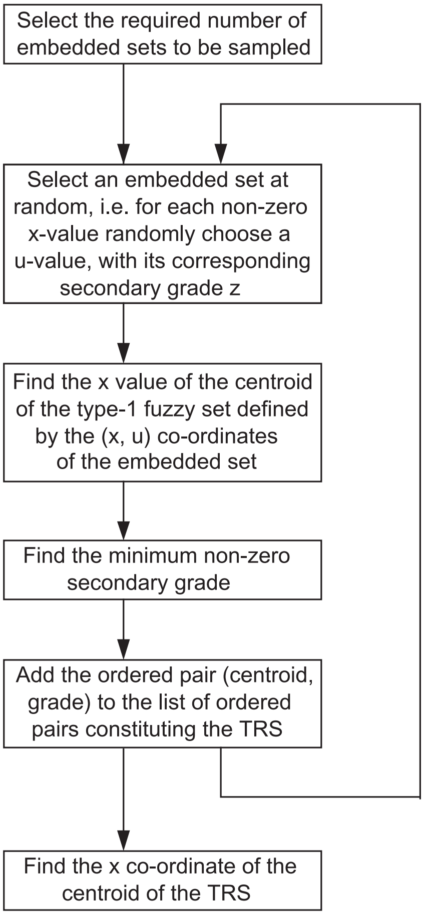


Fig. 5. Flow diagram of the sampling method.

4.2. The sampling defuzzification algorithm

As mentioned above, the essential difference between the sampling method and the exhaustive method of defuzzification is that instead of all the embedded sets participating in type-reduction, a sample is randomly constructed. Fig. 5 sets out the sampling defuzzification process as a flow diagram. It can be seen that stages 6–8 of the type-reduction algorithm presented in Section 3.2 have been omitted. This is because the chances of two primary domain values being equal are low.

4.2.1. Random selection of an embedded set

Because the enumeration of all the possible embedded sets is not practical, a process of *random construction* is employed to select a sample of 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 Fig. 6). 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.

4.2.2. User-selected parameters

For the user of the sampling method there are two decisions to be made:

1. The **degrees of discretisation** for both the primary and secondary domains must be chosen by the user in order to run the FIS and so create the type-2 fuzzy set to be defuzzified. Intuitively it is to be expected that the finer the degree of discretisation, the more accurate the defuzzification results.
2. The **defuzzifier sample size**, i.e. the number of embedded sets (n), is a parameter selected by the user. It is anticipated that a higher number of embedded sets will result in a better accuracy of defuzzification results.

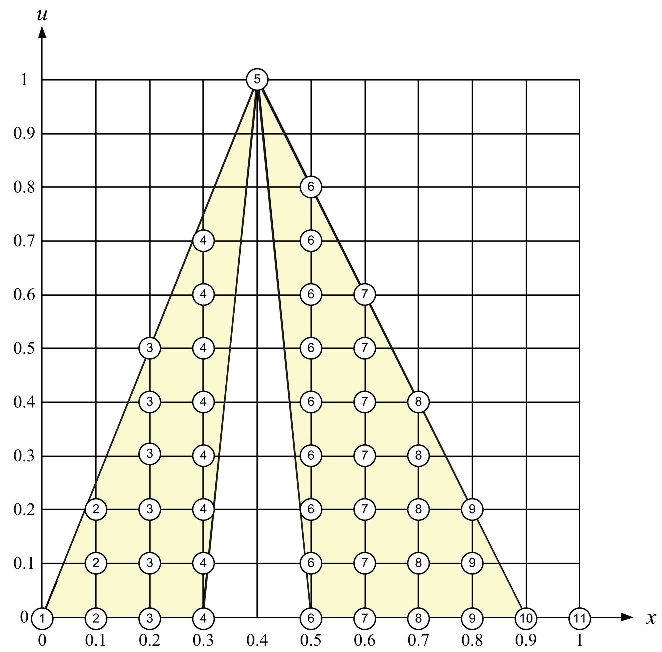


Fig. 6. The grid intersections (circles) within the FOU (shaded area) are the secondary domain values available for constructing an embedded set. They are numbered according to the vertical slice upon which they lie.

Having selected the primary and secondary degrees of discretisation and the sample size, the embedded sets are randomly selected. The sampling method, despite having the extra stages indicated in Algorithm 2, is radically simpler computationally than the exhaustive method.

Algorithm 2. Sampling TRS (in conjunction with the grid method of discretisation).

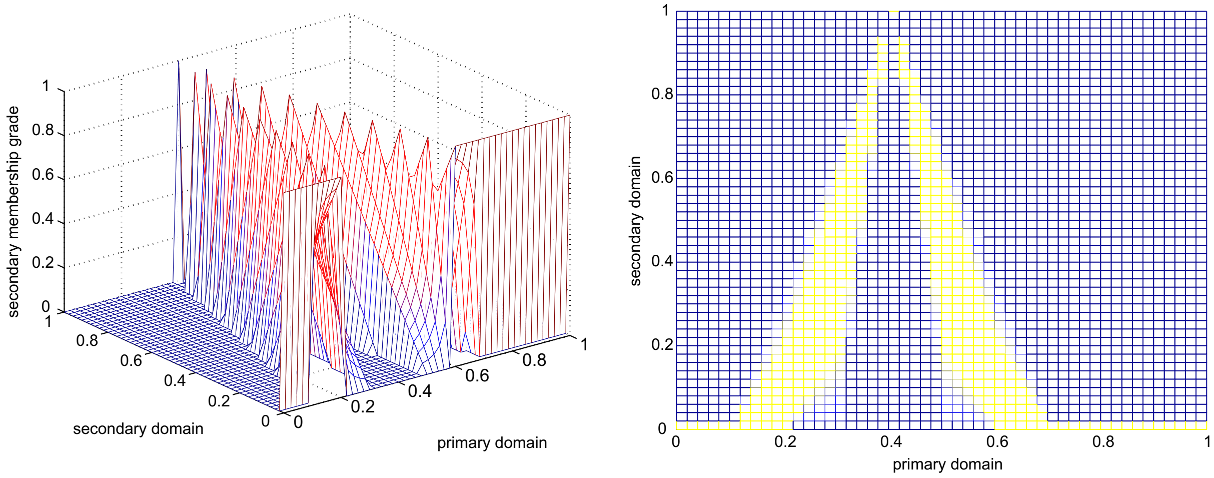
Input: a discretised generalised type-2 fuzzy set
Output: a discrete type-1 fuzzy set
1 select the primary domain degree of discretisation {normally pre-selected};
2 select the secondary domain degree of discretisation {normally pre-selected};
3 select the sample size;
4 **repeat**
5 | randomly select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

5. 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 $\widetilde{A}$. For our experiments, E_n will be our sample space.[3] In this sample space we define the following random variable $X : E_n \rightarrow [0,1]$, where $X(e_n)$ is the centroid of the type-1 fuzzy set deriving from 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 [4, p. 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.

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 a type-2 fuzzy set.

[3] Sampling is the nature of the method described in this paper; 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.



(a) 3-D representation (b) FOU

Fig. 7. Type-2 fuzzy set: triangular primary MF, triangular secondary MFs; degree of discretisation primary and secondary domains is 0.02; defuzzified value = 0.4.

5.1. Test sets

Two test sets were specially constructed using Matlab™. 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 Figs. 7 and 8, 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 (Fig. 7).
Gaussian primary membership function. This symmetrical test set was centred on the *x*-axis to give a defuzzified value of 0.5 (Fig. 8).

Hypothesis testing was carried out in relation to both test sets to ascertain whether or not the estimates of the defuzzified values obtained from the 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 it is $\mu_0 = 0.5$. As we shall see, in both cases the estimates were found to be highly satisfactory.

5.2. The test runs

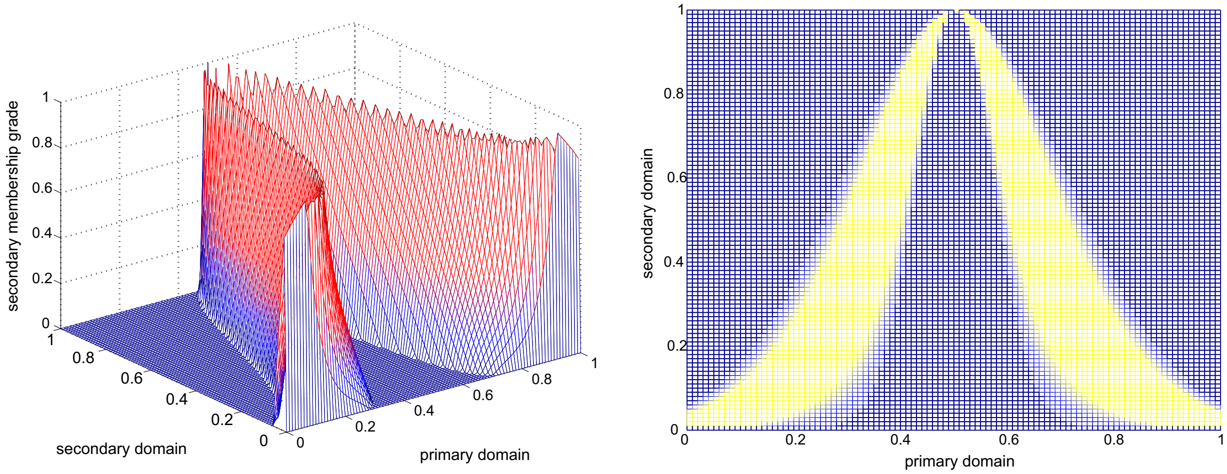
Within each experiment the degree of discretisation[4] and defuzzifier sample size were varied in the same way. Each experiment involved 20 runs, i.e. each test set was defuzzified 20 times using 5 degrees of discretisation (0.1, 0.05, 0.02, 0.01 and 0.0005) and 4 defuzzifier sample sizes (100, 1000, 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]

6. Evaluation of the sampling method: experimental results

The tests were tabulated as Tables 1 and 2. The defuzzification times were not recorded in the tables, as for short runs they were found to be unreliable. (This is attributable to the time taken by operating system processes, which though

[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. 8. Type-2 fuzzy set: Gaussian primary MF, triangular secondary MFs; degree of discretisation primary and secondary domains is 0.01; defuzzified value = 0.5.

negligible in proportion to longer runs, is significant in relation to short runs.) 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 took 0.000848 s, and a sample size of 100,000 took 0.842263 s. 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 took 0.001750 s, and a sample size of 100,000 took 1.733598 s.

6.1. 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

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

has a *Student's t-distribution* with $N - 1$ degrees of freedom, where s is the sample standard deviation. The t-values (t_0) are also provided in Tables 1 and 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.

6.2. 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.

6.3. 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 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.

Table 1
Triangular primary MF, defuzzified value = 0.4.

Degree of discretisation	No. 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

7. Practical application of the sampling defuzzifier

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

Let us assume that E represents the set of all embedded sets. We note that $E = E_1$. According to Algorithm 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 [4, p. 270]: If $\{X_i | i = 1,2,\ldots,n\}$ are n independent random variables with means and variances $\{(\nu_i, \tau_i^2) \mid i = 1,2,\ldots,n\}$, 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 $\hat{w}_i = z_i / \sum_i^n z_i$ and $0 < z_i \leqslant 1$ 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 5), clearly leads to the conclusion that $\nu = \mu$. Therefore, theoretically, the sampling method could be employed in practical cases using just one random sample of embedded

Table 2
Gaussian primary MF, defuzzified value = 0.5.

Degree of discretisation	No. 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

sets of sufficiently large size for the Central Limit Theorem to be applicable. In the next subsection we illustrate the method's application employing a defuzzifier sample size of 100, and it is compared with three different type-2 defuzzification strategies in relation to accuracy and speed.

7.1. Sampling estimation of asymmetric sets

The encouraging results from the previous test runs using symmetrical test sets allows the application of the sampling method to estimate defuzzified values in the more practical cases where they are unknowable other than through defuzzification. Two FIS generated aggregated sets were chosen as test sets because of the complexity and lack of symmetry evident in their graphs.

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 3. The aggregated set produced by the Shopping FIS is depicted in Fig. 9.

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 summarised in Table 4. Fig. 10 shows the aggregated set created by the Washing Powder FIS.

Table 3
Shopping FIS rules.

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

Under coarse discretisation (e.g. a degree of discretisation of 0.1), exhaustive defuzzification is practicable, enabling corroboration of the sampling method. Accordingly the two test sets were created using Matlab™ at a degree of discretisation of 0.1 for both the primary and secondary domains. Each was then defuzzified using four strategies[6]:

1. The sampling method, employing a sample size of 100,
2. the exhaustive method,
3. the α-planes representation combined with the accurate exhaustive method as the interval defuzzifier, using 100 α-planes, and
4. the α-planes representation combined with the approximate KMIP as the interval defuzzifier, using 100 α-planes.

7.2. Defuzzification results

The comparative results are recorded in Tables 5 and 6.

Accuracy: These results confirm that the sampling defuzzifier gives a close estimate to the true defuzzified value as calculated through exhaustive defuzzification. Indeed the errors of the sampling method were less than those of both the α-planes/interval exhaustive and the α-planes/KMIP combinations. The discrepancies between the generalised exhaustive method and the α-planes/interval exhaustive method are attributable to the α-planes decomposition alone, whereas the errors in the α-planes/KMIP combination derive from both the α-planes decomposition and the KMIP.

Efficiency: All timings were much slower than those of the previous set of experiments, since Matlab™ runs more slowly than C. The sampling times were considerably faster than the exhaustive and α-planes/interval exhaustive combination times. The α-planes/KMIP combination was faster still,[7] but slightly less accurate.

8. Parameter choices in applying the sampling method

In any FIS implementation, there are two desirable characteristics of the defuzzification process, namely *speed* and *accuracy*. These characteristics are influenced by the two essential parameter choices of *degree of discretisation* and *sample size*.

- **Speed of defuzzification.** Speed of defuzzification is dependent upon:
1. The hardware used.
2. The software implementation, including choice of programming language.
3. The degree of discretisation, both primary and secondary. The finer the discretisation, the more elements an embedded set contains, which increases the processing time.
4. The sample size.
- **Accuracy.** Accuracy of results depends on the primary and secondary degrees of discretisation and the sample size. Generally either a finer discretisation or a larger sample size leads to greater accuracy.

From the point of view of the user of an FIS employing the sampling defuzzification method, there are two essential parameter choices: *degree of discretisation* and *sample size*.

- **Degrees of discretisation.** The primary and secondary degrees of discretisation are normally selected by the user before invoking the FIS. The finer the discretisation, the closer the approximation of the type-2 fuzzy set to a continuous type-2 set, and hence the more accurate the results. However the drawback to fine discretisation is that the processing effort is increased throughout every stage of the FIS.

[6] All four defuzzification algorithms were coded in Matlab™. The sets were defuzzified on a PC with an Intel (R) Core (TM)2 Duo CPU, a clock speed of 2.93 GHz, a 2.96 GB RAM, running the MS Windows XP Professional 2002 operating system. As in the case of the symmetrical sets (Section 5) the defuzzification programs were run as processes with priorities higher than that of the operating system.

[7] Using the centroid-flow algorithm [36,37] would increase the speed further.

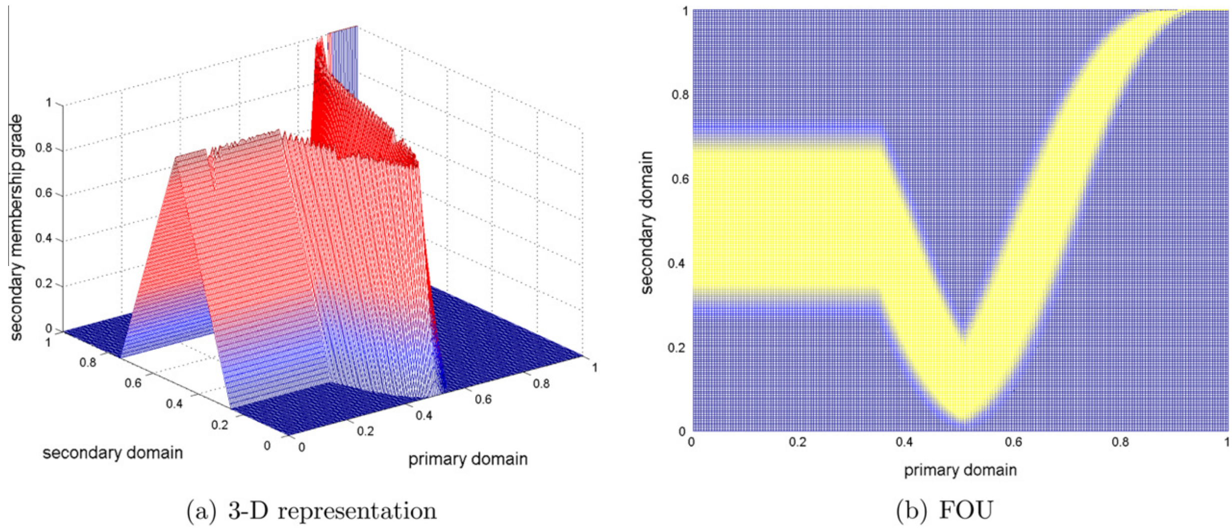


Fig. 9. Shopping FIS test set: aggregated type-2 fuzzy set; degree of discretisation of the primary and secondary domains is 0.005.

Table 4
Washing powder FIS rules.

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

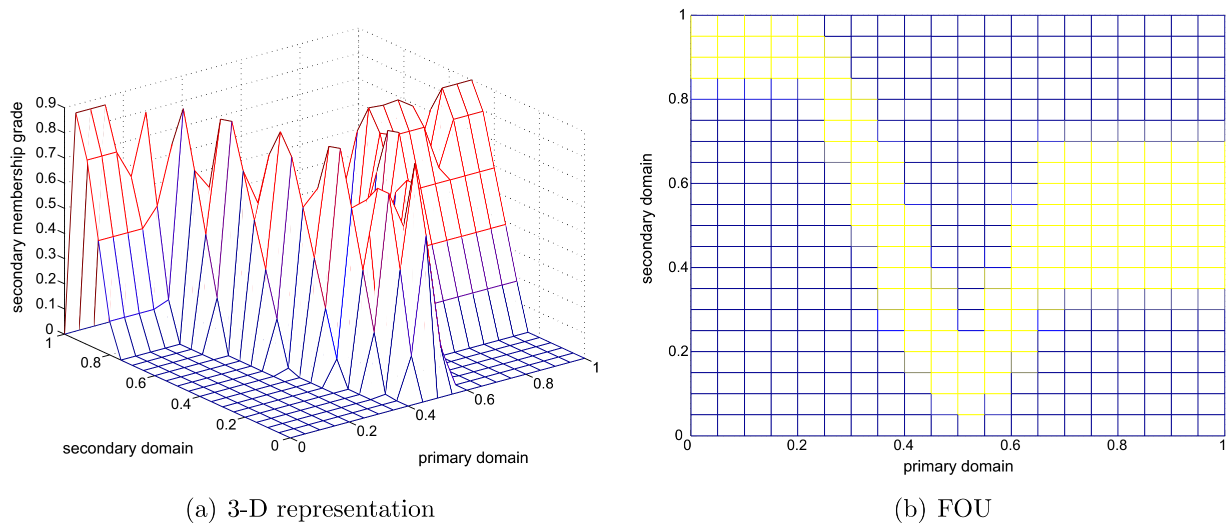


Fig. 10. Washing powder FIS test set: aggregated type-2 fuzzy set; degree of discretisation of the primary and secondary domains is 0.05.

- **Sample size.** The choice of sample size is a compromise between the degree of accuracy required and the defuzzification speed desired. The sample size is the number of embedded sets to be processed, and also the number of elements in the type-reduced set. These both have a bearing on processing time, though the latter's effect is marginal.

Table 5
Defuzzification of the shopping FIS test set.

Method	Defuzzified value	Time taken (s)
Sampling using sample size of 100	0.5935520275	0.094562
Exhaustive	0.5954110452	17.644032
α-Planes/interval exhaustive using 100 α-planes	0.5915223779	8.732145
α-Planes/KMIP using 100 α-planes	0.5863653930	0.028012

Table 6
Defuzzification of the washing powder FIS test set.

Method	Defuzzified value	Time taken (s)
Sampling using sample size of 100	0.4213446619	0.049105
Exhaustive	0.4224023169	84.293682
α-Planes/interval exhaustive using 100 α-planes	0.4294763696	6.418442
α-Planes/KMIP using 100 α-planes	0.4190722100	0.026175

9. Conclusion

In this paper we have considered the computationally challenging problem of defuzzification of generalised type-2 fuzzy sets. The sampling method takes a random sample of the embedded sets and processes only those sets. In the paper we have presented detailed theoretically informed experiments on defuzzification of two type-2 fuzzy sets of known defuzzified values. The results are interesting in that they indicate that this approach trades off very little in accuracy for a massive improvement in speed. Surprisingly, we found the effect of degree of discretisation on accuracy to be slight. In future research we would like to corroborate this using other test sets. From the experiments we are able to provide guidance on the various parameters that need to be selected and the likely impact on performance.

The sampling defuzzifier was compared on aggregated type-2 fuzzy sets resulting from the inferencing stage of a FIS, in terms of accuracy and speed, with other methods including the exhaustive and techniques based on the α-planes representation. The sampling method performed best in accuracy, while in terms of speed it was second after the α-planes/KMIP combination. In all cases, the results indicate that by taking only a sample of the embedded sets we are able to dramatically reduce the time taken to process a type-2 fuzzy set with very little loss in accuracy.

Acknowledgements

The authors thank Dr. Peter Innocent and Dr. Matías Gámez Martínez for their helpful insights during the preparation of this paper, and Ben Passow for his advice about Matlab™ diagrams.

References

[1] J.R. Castro, O. Castillo, P. Melin, A. Rodríguez-Díaz, A hybrid learning algorithm for a class of interval type-2 fuzzy neural networks, Information Sciences 179 (13) (2009) 2175–2193.
[2] S. Coupland, R.I. John, Fuzzy logic and computational geometry, in: Proceedings of RASC 2004, Nottingham, England, 2004, pp. 3–8.
[3] S. Coupland, R.I. John, Geometric type-1 and type-2 fuzzy logic systems, IEEE Transactions on Fuzzy Systems 15 (1) (2007) 3–15.
[4] M.H. DeGroot, Probability and Statistics, second ed., Addison Wesley, 1989.
[5] J.M. Garibaldi, P.A. Birkin, A novel dual-surface type-2 controller for micro robots, in: Proceedings of FUZZ-IEEE 2010, Barcelona, Spain, 2010, pp. 359–366.
[6] S. Greenfield, F. Chiclana, S. Coupland, R.I. John, The collapsing method of defuzzification for discretised interval type-2 fuzzy sets, Information Sciences 179 (13) (2008) 2055–2069.
[7] S. Greenfield, F. Chiclana, S. Coupland, R.I. John, Type-2 defuzzification: two contrasting approaches, in: Proceedings of FUZZ-IEEE 2010, Barcelona, 2010, doi:10.1109/FUZZY.2010.5584007.
[8] S. Greenfield, F. Chiclana, R.I. John, Type-reduction of the discretised interval type-2 fuzzy set, in: Proceedings of FUZZ-IEEE 2009, Jeju Island, Korea, 2009, pp. 738–743.
[9] S. Greenfield, R.I. John, The uncertainty associated with a type-2 fuzzy set, in: Rudolf Seising (Ed.), Views on Fuzzy Sets and Systems from Different Perspectives Philosophy and Logic, Criticisms and Applications, Studies in Fuzziness and Soft Computing, vol. 2, Springer-Verlag, 2009, pp. 471–483.
[10] H. Hagras, C. Wagner, Introduction to interval type-2 fuzzy logic controllers — towards better uncertainty handling in real world applications, IEEE Systems, Man and Cybernetics eNewsletterIssue 27.
[11] E.A. Jammeh, M. Fleury, C. Wagner, H. Hagras, M. Ghanbari, Interval type-2 fuzzy logic congestion control for video streaming across IP networks, IEEE Transactions on Fuzzy Systems 17 (5) (2009) 1123–1142.
[12] N.N. Karnik, An introduction to type-2 fuzzy logic systems, Tech. Rep., University of Southern California, June 1998. <http://sipi.usc.edu/ mendel/report>.
[13] N.N. Karnik, J.M. Mendel, Centroid of a type-2 fuzzy set, Information Sciences 132 (2001) 195–220.
[14] N.N. Karnik, J.M. Mendel, Q. Liang, Type-2 fuzzy logic systems, IEEE Transactions on Fuzzy Systems 7 (6) (1999) 643–658.

[15] E. Kayacan, O. Kaynak, R. Abiyev, J. Tørresen, M. Høvin, K. 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, Barcelona, Spain, 2010, pp. 1125–1132.
[16] C. Leal-Ramírez, O. Castillo, P. Melin, A. Rodríguez-Díaz, Simulation of the bird age-structured population growth based on an interval type-2 fuzzy cellular structure, Information Sciences 181 (3) (2011) 519–535.
[17] L. Leottau, M. Melgarejo, Implementing an interval type-2 fuzzy processor onto a DSC 56F8013, in: Proceedings of FUZZ-IEEE 2010, Barcelona, Spain, 2010, pp. 1939–1942.
[18] O. Linda, M. Manic, Interval type-2 fuzzy voter design for fault tolerant systems, Information Sciences 181 (14) (2011) 2933–2950.
[19] F. Liu, An efficient centroid type-reduction strategy for general type-2 fuzzy logic system, Information Sciences 178 (9) (2008) 2224–2236.
[20] R. Martínez, O. Castillo, L.T. Aguilar, Optimization of interval type-2 fuzzy logic controllers for a perturbed autonomous wheeled mobile robot using genetic algorithms, Information Sciences 179 (13) (2009) 2158–2174.
[21] J.M. Mendel, Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions, Prentice-Hall PTR, 2001.
[22] J.M. Mendel, R.I. John, Type-2 fuzzy sets made simple, IEEE Transactions on Fuzzy Systems 10 (2) (2002) 117–127.
[23] J.M. Mendel, On answering the question "Where do I start in order to solve a new problem involving interval type-2 fuzzy sets?", Information Sciences 179 (19) (2009) 3418–3431.
[24] J.M. Mendel, F. Liu, D. Zhai, α-Plane representation for type-2 fuzzy sets: Theory and applications, IEEE Transactions on Fuzzy Systems 17 (5) (2009) 1189–1207.
[25] G.M. Méndez, M. de los Angeles Hernandez, Hybrid learning for interval type-2 fuzzy logic systems based on orthogonal least-squares and back-propagation methods, Information Sciences 179 (13) (2009) 2146–2157.
[26] S.M. Miller, V. Popova, R. John, M. Gongora, An interval type-2 fuzzy distribution network, in: Proceedings of the 2009 IFSA World Congress/EUSFLAT Conference, Lisbon, 2009, pp. 697–702.
[27] M. Nie, W.W. Tan, Towards an efficient type-reduction method for interval type-2 fuzzy logic systems, in: Proceedings of FUZZ-IEEE 2008, Hong Kong, 2008, pp. 1425–1432.
[28] J.T. Starczewski, Efficient triangular type-2 fuzzy logic systems, International Journal of Approximate Reasoning 50 (5) (2009) 799–811.
[29] C. Wagner, H. Hagras, Toward general type-2 fuzzy logic systems based on zSlices, IEEE Transactions on Fuzzy Systems 18 (4) (2010) 637–660.
[30] D. Wu, J.M. Mendel, Enhanced Karnik–Mendel algorithms, IEEE Transactions on Fuzzy Systems 17 (4) (2009) 923–934.
[31] L. Zadeh, Fuzzy sets, Information and Control 8 (3) (1965) 338–353.
[32] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning, Information Sciences 8 (1975) 199–249.
[33] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning – II, Information Sciences 8 (1975) 301–357.
[34] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning – III, Information Sciences 9 (1975) 43–80.
[35] M. Zaher, H. Hagras, A. Khairy, M. Ibrahim, A type-2 fuzzy logic based model for renewable wind energy generation, in: Proceedings of FUZZ-IEEE 2010, Barcelona, Spain, 2010, pp. 511–518.
[36] D. Zhai, J.M. Mendel, Centroid of a general type-2 fuzzy set computed by means of the centroid-flow algorithm, in: Proceedings of FUZZ-IEEE 2010, Barcelona, 2010, pp. 895–902.
[37] D. Zhai, J.M. Mendel, Computing the centroid of a general type-2 fuzzy set by means of the centroid-flow algorithm, IEEE Transactions on Fuzzy Systems 19 (3) (2011) 401–422.