Novel Fuzzy Inference System (FIS) Analysis and Design Based on Lattice Theory. Part II: Industrial System Modeling

Vassilis G. Kaburlasos, *Member, IEEE*, and Athanasios Kehagias

***Abstract*— We introduce novel (set- and lattice-theoretic) perspectives and tools for the analysis and design of fuzzy inference systems (FIS). We present a FIS, including both fuzzification and defuzzification, as a device for implementing a function $f : \mathsf{R}^N \rightarrow \mathsf{R}^M$. The family of FIS functions has cardinality $\aleph_2 = 2^{\aleph_1}$, where $\aleph_1$ is the cardinality of the set R of real numbers. Hence the FIS family is much larger than polynomials, neural networks, etc.; furthermore a FIS has a capacity for local generalization. A formulation in the context of lattice theory allows us to define the set F^* of fuzzy interval numbers (*FIN*s), which includes both (fuzzy) numbers and intervals. We present a metric d_K on F^*, which can introduce *tunable* nonlinearities. FIS design based on d_K has advantages such as: an alleviation of the curse of dimensionality problem and a potential for improved computer memory utilization. We present a new FIS classifier, namely *granular self-organizing map* (*grSOM*), which we apply to an industrial fertilizer modeling application.**

***Index Terms*— Fuzzy inference system, fuzzy interval number (*FIN*), lattice theory, industrial system modeling, classification**

I. INTRODUCTION

A FUZZY set can be defined on any universe of discourse; however fuzzy sets of real numbers are of particular interest. Many applications use *fuzzy numbers*, i.e. convex, normal fuzzy sets with bounded support. In particular, fuzzy numbers are frequently used in *Fuzzy Inference Systems* (*FIS*s) which use linguistic (fuzzy) rules. Note that much of the popularity of FISs is due to successful automatic control applications [3], [28].

Several authors [4], [37] have employed mathematical lattice theory for knowledge representation, a topic of fundamental significance in artificial intelligence. We have introduced *fuzzy lattice theory* in clustering/classification applications [13], [31], [32] as a cross-fertilization of *mathematical lattice theory* and *fuzzy set theory*. It is remarkable that even though an explicit connection was shown between *mathematical lattices* and *fuzzy sets* since the introduction of fuzzy set theory [44], no tools have been established for FIS analysis and design based on lattice theory. This work engages explicitly mathematical lattice theory for improving FIS analysis and design.

Several authors have compared FIS with various "learning networks" for non-linear function approximation [6], [15]. One way to look at the issue, is to consider a FIS (including both its *fuzzification* and *defuzzification* procedures) as a device which approximates a function $f : \mathsf{R}^N \longrightarrow \mathsf{R}^M$ in a least square error sense [17], [42], [45]. This can then be compared with alternative modeling methods for function approximation including polynomials, spline curves, ARMA models, statistical regressors, multilayer perceptrons, etc. It is important that, as explained below, the family of all FISs has a higher *cardinality* (in the set-theoretic sense) than any of the aforementioned alternatives. Hence FISs can implement "many more" functions than competing families of learning networks[1]; moreover a general FIS is endowed with a capacity for *local* generalization. In other words, a general FIS can implement in principle a far larger number of functions $f : \mathsf{R}^N \longrightarrow \mathsf{R}^M$ than any alternative modeling method, while retaining a capacity for generalization. Additional advantages, as will be explained in this paper, include an alleviation of the "curse of dimensionality" problem, a potential for improved computer memory utilization, etc.

The rest of the paper is organized as follows. In Section II we present some set-theoretic remarks on fuzzy inference systems (FISs). In Section III we present fuzzy interval numbers (*FIN*s). In Section IV, using *FIN*s, we present new perspectives and tools for improved FIS analysis and design, as well as a new FIS classifier. Section V demonstrates the application of these concepts to an industrial modeling problem. We conclude by summarizing and discussing our results in Section VI. Three Appendices summarize useful definitions and results.

II. SOME SET-THEORETIC REMARKS ON FUZZY INFERENCE SYSTEMS

In this section we review conventional FIS principles including novel set-theoretic perspectives.

A FIS includes a knowledge base of fuzzy rules "if A_i then C_i", symbolically $A_i \longrightarrow C_i$, $i = 1, \ldots, L$. The antecedent A_i (IF part) of a rule is typically a conjunction of N fuzzy statements involving N fuzzy sets, moreover the consequent C_i (THEN part) of a rule may be either a fuzzy statement or an algebraic expression. The former is employed by a Mamdani

Manuscript received January 18, 2005; revised September 7, 2005. This work has been supported in part by the third European framework programme: Operational Programme in Education and Initial Vocational Training II, under project Archimedes contract no. 04-3-001/1.

Vassilis G. Kaburlasos is with the Technological Educational Institution of Kavala, Department of Industrial Informatics, Division of Computing Systems, GR-65404 Kavala, Greece. *Email address*: vgkabs@teikav.edu.gr.

Athanasios Kehagias is with Aristotle University of Thessaloniki, Dept. of Math., Phys. & Comp. Sciences, Division of Mathematics, GR-54124 Thessaloniki, Greece. *Email address*: kehagiat@auth.gr.

[1] We make this point precise in Section II (see also [8], [10], [11]).

type FIS [20], whereas the latter is employed by a Sugeno type FIS [39]. The fuzzy sets involved in a FIS are typically *fuzzy numbers*, i.e. convex, normal fuzzy sets with bounded support defined on the real number universe of discourse R.

An input vector $x \in \mathsf{R}^N$ to a FIS activates, in parallel, rules in the knowledge-base by a *fuzzification* procedure; next, an *inference mechanism* produces the consequents of activated rules; then, the partial results are combined; finally, a real number vector is produced by a *defuzzification* procedure. A variety of fuzzy number shapes /inference mechanisms /(de)fuzzification procedures have been proposed in the literature [1], [22], [29], [35], [43].

Hence, a FIS implements a function $f : \mathsf{R}^N \longrightarrow \mathsf{R}^M$, where (1) N and M are integers, and (2) function f is induced from n pairs $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ of training data vectors. It turns out that the design of a FIS typically boils down to a parameter optimization problem, where it is required to minimize the least squares error $\sqrt{\sum_{i=1}^{n} ||f(x_i) - y_i||^2}$ [45], [46]. In contrast to alternative function estimation methods [34], [41], a FIS retains explicitly a linguistic interpretation. Lately, neural implementations of FIS have proliferated [5], [7], [24], [25].

Despite potential drawbacks, such as the *curse of dimensionality*, which occurs when the number of input/output variables increases, it is widely recognized that a FIS can give in practice better results than alternative function approximation methods and, usually, a fuzzy logic explanation is sought. This work proposes, in addition, a set-theoretic explanation.

First let us calculate $card(\mathcal{F})$, the cardinality of the set $\mathcal{F}$ of *all* functions $f : \mathsf{R}^N \longrightarrow \mathsf{R}^M$. Using standard *cardinal arithmetic* [38] we have

$$card(\mathcal{F}) = \aleph_1^{\aleph_1} = (2^{\aleph_0})^{\aleph_1} = 2^{\aleph_0 \aleph_1} = 2^{\aleph_1} = \aleph_2 > \aleph_1.$$

Unfortunately a general function f_0 in $\mathcal{F}$ is practically useless because it lacks a capacity for generalization. More specifically, knowledge of function f_0 values $f_0(x_1), ..., f_0(x_n)$ at a number of points $x_1, ..., x_n$ cannot give any information regarding the value of function f_0 at a different point $x_{n+1} \neq x_i$, $i = 1, \ldots, n$.

Consider now a parametric family of models (characterized by a capacity for generalization), e.g. polynomials, ARMA models, statistical regressors, radial basis function (RBF) networks, multilayer perceptrons, etc. Due to the finite number p of parameters involved in a parametric family of models it follows that the cardinality of any of the aforementioned families equals $\aleph_1^p = (2^{\aleph_0})^p = 2^{\aleph_0 p} = 2^{\aleph_o} = \aleph_1$. It might be thought that $\aleph_1$ is an adequately large number of models to choose a "good" model from, in a practical application. Unfortunately this is not the case. Consider, for instance, the family of polynomials, which includes $\aleph_1$ models. It is well known that a polynomial may not approximate usefully a set $(x_1, y_1), ..., (x_n, y_n)$ of training data due to *overfitting*; hence a different family of models might be sought, e.g. a ARMA model, a multi-layer perceptron, etc. In the aforementioned sense the cardinality $\aleph_1$ (of a family of models) may be *inherently restrictive*.

What about the cardinality of the set of all FISs? To compute this, let us first compute the cardinality of the set F of fuzzy numbers. The next proposition shows the non-obvious result that there are as many fuzzy numbers as there are real numbers.

Proposition 2.1: It is $card(\mathsf{F}) = \aleph_1$, where $\aleph_1$ is the cardinality of the set R of real numbers.

Proof: The proof appears in [11]. The case of fuzzy numbers with continuous membership has been proved in [8]. ■

Now consider, Mamdani type FISs: the rules in a Mamdani type FIS can be interpreted as samples of a function $m : \mathsf{F}^N \longrightarrow \mathsf{F}^M$. Using standard cardinal arithmetic [38] it follows that the cardinality of the set $\mathcal{M}$ of Mamdani type FISs is $card(\mathcal{M}) = \aleph_1^{\aleph_1} = \aleph_2 > \aleph_1$. Likewise, the rules in a Sugeno type FIS can be interpreted as samples of a function $s : \mathsf{F}^N \longrightarrow \mathcal{P}_p$, where $\mathcal{P}_p$ is a family of parametric models (e.g. polynomial linear models) with p parameters. It follows that the cardinality of the set $\mathcal{S}$ of Sugeno type FISs is $card(\mathcal{S}) = \left(\aleph_1^{\aleph_1}\right)^p = \aleph_1^{(2^{\aleph_0})^p} = \aleph_1^{2^{p\aleph_0}} = \aleph_1^{2^{\aleph_0}} = \aleph_1^{\aleph_1} = \aleph_2$. Also, a FIS (of either Mamdani or Sugeno type) has a capacity for local generalization due to the non-trivial (interval) support of the fuzzy numbers involved in FIS antecedents. In other words, an input vector $x = (x_1, \ldots, x_n)$ within the support of a fuzzy rule activates the aforementioned rule.

In conclusion, a FIS (of either Mamdani or Sugeno type) can implement, in principle, $\aleph_2$ functions and in addition it has a capacity for generalization. Hence the class of FISs is preferable to both the "general" class $\mathcal{F}$ of functions (which lacks a capacity for generalization) and to parametric families of models (which have a smaller cardinality). It is understood that the aforementioned advantage of the family of FIS models is theoretical. Nevertheless substantial practical advantages are shown in the context of this work, based on fuzzy interval number (*FIN*) analysis presented in the following section.

Proposition 2.1 also suggests an interesting proposal regarding the preferable fuzzy number membership function shape. A variety of such shapes have been proposed in the literature including triangular, trapezoidal, polynomial, bell-shaped, etc. [5], [22], [29], [36], [43]. Any of the aforementioned shapes is described by a finite number p of parameters; for instance a triangular membership function is described using $p = 3$ parameters. Hence there exist $\aleph_1^3 = \aleph_1$ fuzzy numbers of triangular shape. Likewise, there exist $\aleph_1^p = \aleph_1$ fuzzy numbers of any particular parametric shape. Moreover, since the number of different parametric shapes (e.g. triangular, Gaussian, trapezoidal, etc.) in practice is finite, it follows that we have a set of $\aleph_1$ parametric fuzzy numbers altogether. It follows that using any of the aforementioned families we can generate $\aleph_2$ functions $f : \mathsf{F}^N \rightarrow \mathsf{F}^M$, each function characterized by a (local) capacity for generalization. Hence, in conclusion, proposition 2.1 ultimately implies that any membership function shape enables a FIS to implement, in principle, $\aleph_2$ different functions. In practice triangular membership function shapes are frequently preferable due to their convenient representation using only $p = 3$ parameters.

III. Fuzzy Interval Numbers (*FIN*s) and Metrics

In this section we define *FIN*s and equip them with a metric which will be used in Section IV to introduce metric-based

FIS.

A fuzzy set F on R is called a *fuzzy number* if it satisfies the following properties [16, p.97]:

A1 It is normal (i.e. $\exists x_0 : F(x_0) = 1$);
A2 The a-cut $F_a = \{x : F(x) \geq a\}$ is a closed interval for all $a \in (0,1]$;
A3 The support of F (i.e. the set $\{x : F(x) > 0\} = \cup_{a\in(0,1]} F_a$) is bounded.

It is well known that every fuzzy set is uniqely represented by its a-cuts. Hence, we can define fuzzy numbers in terms of their a-cuts as follows.

Definition 3.1: A *fuzzy number* is a family of sets $\{F_a\}_{a\in[0,1]}$ which satisfy the following conditions.

F1 $F_0 = \mathsf{R}$.
F2 For every $a, b \in [0,1]$ we have: $a \leq b \Rightarrow F_b \subseteq F_a$.
F3 For every set $A \subseteq [0,1]$, letting $b = \sup A$, we have: $\cap_{a\in A} F_a = F_b$.
F4 For every $a \in (0,1]$, F_a is a closed interval.
F5 $\cup_{a\in(0,1]} F_a$ is a bounded interval.
F6 $F_1 \neq \varnothing$.

We denote the set of fuzzy numbers by F .

In the above definition we could have substituted "...is a family of sets..." by "...can be uniquely represented by a family of sets...". Properties **F1-F3** are the usual properties satisfied by the a-cuts of *every* fuzzy set; **F4** corresponds to **A2**, **F5** corresponds to **A3** and, **F6** corresponds to **A1**.

As pointed out above, a FIS implements a mapping from fuzzy numbers to either fuzzy numbers (for Mamdani type FIS) or algebraic expressions (for Sugeno type FIS). From a computational aspect, a FIS can operate with either numerical representations of functions or, using the "a-cuts point of view", with numerical representations of families of closed intervals [40]. We propose to enhance the advantages of the interval representation *by operating with a wider class of sets*. In particular we propose FIS which operate on *generalized intervals*. We introduce the proposed generalization in two steps: First, we discuss *fuzzy intervals*, and then *generalized intervals*.

Many definitions of *fuzzy interval* have appeared in the literature. We choose one (in terms of a-cuts) which has maximum compatibility with the definition of fuzzy number.

Definition 3.2: A *fuzzy interval* is a family of sets $\{F_a\}_{a\in[0,1]}$ which satisfy the conditions **F1** - **F5** above. We denote the set of fuzzy intervals by F_{int} .

In the above definition we could have substituted "...is a family of sets..." by "...can be uniquely represented by a family of sets...". Every fuzzy number satisfies **F1-F5**, hence $\mathsf{F} \subseteq \mathsf{F}_{int}$.

We can operate on fuzzy intervals F and G using the a-cuts representation, i.e. for every $a \in (0,1]$ we can operate on the a-cuts F_a, G_a. Hence it is natural to study the family $\mathfrak{I}$ of all closed intervals on R. More accurately we define

$$\mathfrak{I} = \{[a,b] : a, b \in \mathsf{R}, a \leq b\} \cup \{\emptyset\}$$

It is known [26] that $(\mathfrak{I}, \subseteq, \cup, \cap)$ is a lattice, where the order relationship is set inclusion (with minimum element $\emptyset$)[2].

[2]For convenience, elements from *lattice theory* are summarized in Appendix A; furthermore, Appendix B summarizes elements from *fuzzy lattice theory*.

Especially for nonempty intervals $[a,b]$ and $[c,d]$ the join $\cup$ and meet $\cap$ operations are

$$[a,b] \cup [c,d] = [a \wedge c, b \vee d], \quad [a,b] \cap [c,d] = [a \vee c, b \wedge d]$$

where $a \wedge c = min\{a,c\}$ and $a \vee c = max\{a,c\}$; furthermore, if $a \vee c > b \wedge d$ then $[a,b] \cap [c,d] \doteq \emptyset$.

Our intention is to equip $\mathfrak{I}$ with a sensible metric. This will be useful for the design of metric-based FIS in Section IV. Several metrics between intervals can be defined. However, the commonly used metrics do not serve our purpose well. In particular, difficulties arise with the treatment of nonintersecting intervals. Hence we will introduce a new family of metrics defined in terms of an alternative representation of closed intervals, namely *positive generalized intervals*.

Definition 3.3: A *positive generalized interval of height h* is a function $m^h_{x_1,x_2} : \mathsf{R} \to \{0, h\}$ (where $x_1 \leq x_2$ and $h \in (0,1]$) defined as follows

$$m^h_{x_1,x_2}(x) = \begin{cases} h, & x_1 \leq x \leq x_2 \\ 0, & \text{otherwise.} \end{cases}$$

We will also denote the generalized interval $m^h_{x_1,x_2}$ by the more convenient notation $[x_1, x_2]^h$, where $x_1 \leq x_2$.

Given any fuzzy interval $F \in \mathsf{F}_{int}$, we can use its a-cuts to generate a family of positive generalized intervals as follows: for every $a \in (0,1]$, denote the indicator function of F_a by $\widetilde{F}_a$; then the function $a\widetilde{F}_a$ is a positive generalized interval of height a, *provided that F_a is not empty*! To handle the case of empty sets, we introduce a "degenerate" generalized interval, denoted by ϕ^h, and satisfying

$$\phi^h(x) = 0 \text{ for every } x \in \mathsf{R}.$$

Definition 3.4: M^h_+ is defined to be the family of all positive generalized intervals of height h; M^h_0 is defined as $\mathsf{M}^h_0 = \mathsf{M}^h_+ \cup \{\phi^h\}$.

Obviously, positive generalized intervals (of a given height h) are in a 1-to-1 correspondence with "classic" closed intervals. *Negative* generalized intervals can also be defined (as shown in Appendix C). Note that the set of (positive *and* negative) generalized intervals was introduced in [8], [9], [33] and will be instrumental for the definition of our family of metrics.

Using the 1-to-1 correspondence between M^h_0 and $\mathfrak{I}$, we can equip M^h_0 with an order relation $\preceq$.

Definition 3.5: Given $h \in (0,1]$, we define the relation $\preceq$ on $\mathsf{M}^h_0 \times \mathsf{M}^h_0$ as follows. For all $[a,b]^h$, $[c,d]^h \in \mathsf{M}^h_+$ we have

$$\phi^h \preceq [a,b]^h \quad \text{and} \quad [a,b]^h \preceq [c,d]^h \Leftrightarrow [a,b] \subseteq [c,d]$$

Proposition 3.6: The relation $\preceq$ is an order on M^h_0 and $(\mathsf{M}^h_0, \preceq)$ is a lattice, where the lattice join $\curlyvee$ and meet $\curlywedge$ operations are given (for all $[a,b]^h$, $[c,d]^h \in \mathsf{M}^h_+$) by

$$[a,b]^h \curlyvee [c,d]^h = [a \wedge c, b \vee d]^h, \qquad [a,b]^h \curlyvee \phi^h = [a,b]^h$$
$$[a,b]^h \curlywedge [c,d]^h = [a \vee c, b \wedge d]^h, \qquad [a,b]^h \curlywedge \phi^h = \phi^h.$$

Proof: This is a straightforward consequence of the 1-to-1 correspondence between M^h_0 and $\mathfrak{I}$. ■

Recall from above that in proposition 3.6 it is defined $[a,b]^h \curlywedge [c,d]^h \doteq \emptyset^h$, if $a \vee c > b \wedge d$.

Based on positive generalized intervals, we now define *Fuzzy Interval Numbers* (*FIN*s).

Definition 3.7: A *fuzzy interval number (FIN)* is a family $\{F_h\}_{h\in(0,1]}$ of positive generalized intervals constructed from a fuzzy number $F \in \mathsf{F}$ such that $F_h = h\widetilde{F}_h$, where $\widetilde{F}_h$ is the indicator function of $\{x : F(x) \geq h\}$. The set of all *FIN*s will be denoted by F^*.

*FIN*s have been defined elsewhere as mathematical objects which may have either positive or negative membership functions [9], [12]. Interpretations for *negative FIN*s have been proposed [12], but they are outside the scope of this paper. Since the interest of this work is in fuzzy inference systems (FISs), we have considered here only *positive FIN*s, or *FIN*s for short; the latter are interpreted as fuzzy numbers.

Now we turn to metrics on *FIN*s; these will be built from metrics on positive generalized intervals.

Proposition 3.8: Let $f_h : \mathsf{R} \to \mathsf{R}$ be a strictly increasing function. Define $\overline{d}_h : \mathsf{M}_0^h \times \mathsf{M}_0^h \to \mathsf{R}_0^+$ as follows: for all $[a,b]^h$, $[c,d]^h \in \mathsf{M}_+^h$ set

$$\overline{d}_h([a,b]^h,[c,d]^h) = [f_h(a\vee c)-f_h(a\wedge c)]+[f_h(b\vee d)-f_h(b\wedge d)] \quad (1)$$

$$\overline{d}_h([a,b]^h,\phi^h) = \overline{d}_h(\phi^h,[a,b]^h) = \infty \quad (2)$$

$$\overline{d}_h(\phi^h,\phi^h) = 0 \quad (3)$$

Then $\overline{d}_h$ is a metric on M_0^h.

Proof: The proof is given in Appendix C. Here we only remark that the proof depends on the fact that the set of (positive *and* negative) generalized intervals is a metric lattice. Hence *negative* generalized intervals (which are discussed in Appendix C) are just as important as positive generalized intervals in introducing the metric $\overline{d}_h$. ■

It can be seen from Proposition 3.8 that every strictly increasing function f_h generates a metric on lattice $(\mathsf{M}_0^h, \preceq)$. Function f_h is also called *underlying positive valuation* because f_h is a positive valuation function in the totally ordered lattice R of real numbers (for a definition of a *positive valuation* function see in Appendix A). Let $\mathbf{D}_h$ denote the family of all metrics in M_0^h. A *very large number* of metrics can be defined in the manner of Proposition 3.8.

Proposition 3.9: $\text{card}(\mathbf{D}_h) \geq \aleph_1$.

Proof: In [11]. ■

We can obtain an underlying positive valuation f_h from every nonnegative, integrable *mass function* $m_h : \mathsf{R} \to \mathsf{R}_0^+$ as follows:

$$f_h(x) = \int_0^x m_h(t)dt,$$

where the above integral is positive (negative) for $x > 0$ ($x < 0$). One may use a mass function $m_h(x)$ as a "device" for attaching a weight of significance to a number x in a data dimension. We remark that typical FIS applications in the literature employ solely (and implicitly) mass function $m_1(x) = 1$; the latter corresponds to the linear underlying positive valuation function $f_1(x) = x$. Nevertheless, *alternative* mass functions can be used. For example a constant mass function $m_h(x) = k_h > 0$ generates a linear underlying positive valuation $f_h(x) = k_h x$, which in turn generates the metric $\overline{d}_h([a,b]^h,[c,d]^h)$= $k_h \cdot (|a-c|+|b-d|)$. Furthermore, nonlinear positive valuation functions can be generated from alternative mass functions as demonstrated below. Hence, a mass function can be interpreted as a "weight function" in a data dimension. For instance, a constant mass function $m_h(x) = k_h$ scales all the numbers in a data dimension equally by k_h; whereas, a non-constant mass function $m_h(x)$ scales different numbers in a data dimension differently.

We are ready now to define metrics on the set of fuzzy interval numbers (*FIN*s).

Proposition 3.10: Given a positive number c, define the function $d_K : \mathsf{F}^* \times \mathsf{F}^* \to \mathsf{R}_0^+$ as follows: for every $F, G \in \mathsf{F}^*$ let

$$d_K(F,G) = c\int_0^1 \overline{d}_h(F_h,G_h)dh.$$

Then d_K is a *pseudometric*, i.e. for all $F,G,H \in \mathsf{F}^*$ we have
$d_K(F,F) = 0$, $d_K(F,G) = d_K(G,F)$, and
$d_K(F,H) \leq d_K(F,G) + d_K(G,H)$

Proof: The proof is given in Appendix C. ■

The reason that d_K is not a metric is that two fuzzy interval numbers F, G with $F \neq G$ could differ only on a set of measure zero; hence $d_K(F,G) = 0$ does not necessarily imply that $F = G$. From a practical point of view the latter is unlikely to occur. In any case, from a pseudometric it is possible to obtain a true metric by a standard construction. Namely, we define an equivalence relation R on F^* as follows: F and G are equivalent iff they differ on a set of measure zero. Then d_K is a true metric on the quotient space F^*/R, i.e. d_K is a metric between the equivalence classes of R [2]. The following example demonstrates experimentally the computation of d_K on the plane.

Example. Consider the three fuzzy numbers $E1, E2$, and F, with piecewise linear membership functions, shown in Fig.1(a). Note that the left sides of $E1$ (solid line) and $E2$ (dashed line) coincide, nevertheless the corresponding right sides are clearly different. Note also that both fuzzy numbers $E1$ and $E2$ attain their unique maximum value at $x = 0$. Moreover, fuzzy number F has an isosceles triangular membership function centered at $x = 3$. Two different mass functions are shown in Fig.1 (b) and (c). On the one hand, the mass function $m_h(t) = h$ (shown in Fig.1(b) for $h = 1$) assumes that all the real numbers are equally important; the corresponding positive valuation function is given by $f_h(x) = hx$. On the other hand, the mass function $m_h(t) = 4h\frac{e^{-7(t-0.5)}}{(1+e^{-7(t-0.5)})^2}$ (shown in Fig.1(c) for $h = 1$) emphasizes symmetrically the numbers around $t = 0.5$; the corresponding positive valuation function, namely *logistic function* (in statistics) or *sigmoid function* (in neural computing), is given by $f_h(x) = \frac{h}{1+e^{-7(x-0.5)}}$. Fig.2 displays the metrics $\overline{d}_h(E1_h, F_h)$ and $\overline{d}_h(E2_h, F_h)$ in solid and dashed lines, respectively. In particular the mass function $m_h(t) = h$ (Fig.1(b)) was employed for computing the curves shown in Fig.2(a), whereas the mass function $m_h(t) = 4h\frac{e^{-7(t-0.5)}}{(1+e^{-7(t-0.5)})^2}$ (Fig.1(c)) was employed for computing the curves shown in Fig.2(b). In Fig.2(a) it follows $d_K(E1,F) \approx 3.0 > 2.9754 \approx d_K(E2,F)$, whereas in Fig.2(b) it follows

$d_K(E1, F) \approx 0.3587 < 0.3811 \approx d_K(E2, F)$. Figures Fig.2 (a) and (b) were meant to demonstrate that a mass function can be used as an instrument for tuning, non-linearly, the metric between two fuzzy numbers. Furthermore, note that metric $d_K(E, F)$ can be used for computing a degree of membership of a fuzzy number E in another fuzzy number F using a function $m_F : \mathsf{F} \times \mathsf{F} \rightarrow [0, 1]$ given, for example, by $m_F(E) = \frac{1}{1+d_K(F,E)}$ inspired from [18] where it is attributed to Zimmermann and Zysno.

(Fig.1 goes here, or below)

(Fig.2 goes here, or below)

IV. Novel Perspectives and Tools for FIS Analysis and Design

The results of the previous sections will be employed in this section for enhancing conventional FIS analysis and design.

A. Metric FIS design: Principles

This subsection carries out a discussion for a Mamdani type FIS; nevertheless, the basic arguments can be extended for Sugeno type FIS.

In the heart of a Mamdani type FIS lies a collection of pairs (A_i, C_i) of multi-dimensional fuzzy numbers $A_i \in \mathsf{F}^N$, $C_i \in \mathsf{F}^M$, $i = 1, \ldots, L$. In conventional FIS terminology a pair (A_i, C_i) is interpreted as a fuzzy rule "if A_i then C_i", $i = 1, \ldots, L$. In the context of this work, the collection (A_i, C_i), $i = 1, \ldots, L$ is interpreted as a look-up table for function approximation by interpolation. More specifically, the operation of a conventional FIS can be described using the following five functions:

1. *Fuzzification function* $f_{fz} : \mathsf{R}^N \times \mathsf{F}^N \longrightarrow \mathsf{F}^N_{int}$.
2. *Rule activation function* $f_{ra} : \mathsf{F}^N_{int} \longrightarrow [0, 1]$.
3. *Partial rule inference function* $f_{pri} : [0, 1] \times \mathsf{F}^M \longrightarrow \mathsf{F}^M_{int}$.
4. *Total rule inference function* $f_{tri} : (\mathsf{F}^M_{int})^L \longrightarrow \mathsf{F}^M_{int}$.
5. *Defuzzification function* $f_{dfz} : \mathsf{F}^M_{int} \longrightarrow \mathsf{R}^M$.

Recall that F (F_{int}) denotes the set of fuzzy numbers (intervals); moreover, recall that $\mathsf{F} \subset \mathsf{F}_{int}$ since the height h of a fuzzy number is always $h = 1$, whereas the height h of a fuzzy interval is $h \in (0, 1]$.

The first argument in the fuzzification function $f_{fz}(.,.)$ is an input to the FIS, whereas the second argument is treated as a parameter $A_i \in \mathsf{F}^N$; more specifically, $y_{fz} = f_{fz}(x; A_i)$, where $x \in \mathsf{R}^N$, $A_i \in \mathsf{F}^N$ for $i \in \{1, \ldots, L\}$, and $y_{fz} \in \mathsf{F}^N_{int}$. Likewise, the second argument in the partial rule inference function $f_{pri}(.,.)$ is treated as a parameter $C_i \in \mathsf{F}^M$; more specifically, $y_{pri} = f_{pri}(x; C_i)$, where $x \in [0, 1]$, $C_i \in \mathsf{F}^M$ for $i \in \{1, \ldots, L\}$, and $y_{pri} \in \mathsf{F}^M_{int}$. The meaning of the other three functions f_{ra}, f_{tri}, and f_{dfz} is obvious. It is interesting to point out that for an input $x \in \mathsf{R}^N$ a FIS computes, in parallel, L values of the function f_{pri}; hence, L different $M-$dimensional fuzzy intervals in F^M_{int} are computed. The aforementioned intervals are used as an input to the total rule inference function $f_{tri} : (\mathsf{F}^M_{int})^L \longrightarrow \mathsf{F}^M_{int}$. We remark that the parallel computation of functions f_{pri} has occasioned parallel, e.g. neural, implementations of FISs.

To operate a FIS we need to know both (1) functions f_{fz}, f_{ra}, f_{pri}, f_{tri}, and f_{dfz}, and (2) the pairs of parameters (A_i, C_i), $i \in \{1, \ldots, L\}$. In conclusion, a FIS implements a parametric function $f : \mathsf{R}^N \longrightarrow \mathsf{R}^M$ given by $y = f(x; f_{fz}, f_{ra}, f_{pri}, f_{tri}, f_{dfz}, (A_1, C_1), \ldots, (A_L, C_L))$.

Given n training data pairs (x_i, y_i), where $x_i \in \mathsf{R}^N$ and $y_i \in \mathsf{R}^M$ for $i \in \{1, \ldots, n\}$, the practical question in a Mamdani type FIS design problem is to estimate the parameters f_{fz}, f_{ra}, f_{pri}, f_{tri}, f_{dfz}, $(A_1, C_1), \ldots, (A_L, C_L)$ so as to minimize the least squares error LSE= $\sqrt{\sum_{i=1}^{n} ||f(x_i) - y_i||^2}$. Based on fuzzy logic arguments the aforementioned function parameters f_{fz}, f_{ra}, f_{pri}, f_{tri}, and f_{dfz} are, typically, fixed; it remains to estimate the parameter pairs $(A_1, C_1), \ldots, (A_L, C_L)$, namely "fuzzy rules" in conventional FIS terminology.

From a function analytic point of view, it is legitimate to replace both the *fuzzification* and *rule activation* functions f_{fz} and f_{ra}, respectively, by their composition function $f_{comp} = f_{fz} \circ f_{ra} : \mathsf{R}^N \times \mathsf{F}^N \longrightarrow [0, 1]$, where the second argument of function $f_{comp}(.,.)$ is treated as a parameter $A_i \in \mathsf{F}^N$, $i \in \{1, \ldots, L\}$. This work proposes replacing the composite function $f_{comp} : \mathsf{R}^N \times \mathsf{F}^N \longrightarrow [0, 1]$ by a 'more advanced' fuzzy membership function $f_{adv} : \mathsf{F}^N \times \mathsf{F}^N \longrightarrow [0, 1]$ based on the metric d_K between *FIN*s; for instance, $f_{adv}(x, A) = \frac{1}{1+d_K(x,A)}$ can be used. In the aforementioned manner we overcome inherent drawbacks of conventional FISs as explained in the following.

B. Metric FIS design: The Potential

A drawback for conventional FIS design is the *curse of dimensionality* problem. That is, when the number of input/output variables increases linearly then the number of fuzzy rules increases exponentially. The latter occurs because an input $x \in \mathsf{R}^N$ to a conventional FIS needs to be within the interval support of "at least one" fuzzy rule, otherwise no FIS output is produced. One way to counter the problem is by placing fuzzy rules only where the data typically appear "hoping" that no data will ever appear elsewhere. A safer way for overcoming the curse of dimensionality problem is by using fuzzy rules with long supports, e.g. Gaussian membership functions. Unfortunately a Gaussian membership function with mean μ and standard deviation σ is practically zero outside the interval $[\mu - 3\sigma, \mu + 3\sigma]$.

Using the abovementioned function $f_{adv}(x, A)$ a FIS input x could be beyond all fuzzy rule supports. Hence fewer fuzzy rules may be used without covering the whole input data domain. The latter is demonstrated by an example below in this subsection. An additional advantage of function $f_{adv}(x, A)$ is that a FIS input x can be a fuzzy number $x \in \mathsf{F}^N$ to compensate for ambiguities in the input data. Both aforementioned advantages are attributed to the fact that the fuzzy membership function $f_{adv} : \mathsf{F}^N \times \mathsf{F}^N \rightarrow [0, 1]$ is defined on the '*FIN*s universe of discourse'; hence it is feasible to compute a fuzzy degree of inclusion of a *FIN* F_1 in another *FIN* F_2, even when the interval supports of F_1 and F_2 do not intersect.

A *mass function* has already been presented above as an instrument for introducing nonlinearities. A *mass function* will also be called *underlying mass function*, where the term "underlying" is used as a reminder that a mass function is an *important* function for building a positive valuation; the latter is used in turn for building a metric as shown in Proposition 3.10. Note that the term "underlying" may be dropped and the corresponding function could simply be called *mass function*. Another advantage is a mass function's capacity to maximize the utility of a digital computer's memory as described next.

Subtle theoretical advantages of various algorithms may evaporate in practice, when numerical calculations are carried out on a digital computer, due to round-off errors. For instance, this work has proposed FIS design using a metric-based activation of fuzzy rules, where a metric $d_K(E_i, H_i)$ between two fuzzy numbers E_i and H_i in the i^{th} dimension is calculated as $d_K(E_i, H_i) = c\int_0^1 d_h((E_i)_h, (H_i)_h)dh$, $c > 0$. Hence, the corresponding metric between two N-tuple fuzzy numbers $E = (E_1, \ldots, E_N)$ and $H = (H_1, \ldots, H_N)$ can be calculated using the following *Minkowski metric*

$$d_p(E, H) = [d_K(E_1, H_1)^p + \ldots + d_K(E_N, H_N)^p]^{1/p},$$

where $p \geq 1$ is a selectable integer parameter. Nevertheless, a Minkowski metric assumes an integer parameter value p; hence, only a fairly small number of different Minkowski metrics $d_p(.,.)$ can be used in practice because number $d_{p_0}(E, H)$ is not expected to be different than number $d_{p_0+1}(E, H)$ in a digital computer for fairly large p_0. It follows that a substantial part of a digital computer memory may stay unused. Nevertheless, the capacity to compute metric $d_K(.,.)$ based on a *mass function* leads to a much larger number of metric functions to choose from in a practical application; thus an employment of $d_K(.,.)$ may take better advantage of the existing digital computer memory resources.

The disadvantage of using metric d_K based on a mass function is computational complexity; in particular, the computation of d_K requires the calculation of an extra definite integral. However, there is experimental evidence that the employment of d_K based on (genetically computed) mass functions can improve performance, for instance in classification problems [12]. The following example demonstrates advantages of the proposed novel FIS design.

1) An Extended Example: Consider a simple, Mamdani type FIS inspired from industrial fertilizer production including two linguistic inputs, one linguistic output, and nine fuzzy rules (Fig.3). More specifically, one input variable is 'AN Melt Flow' in m^3/h, the other input variable is 'Recycled Fertilizer' in T/h, moreover the output variable is '(Fertilizer Granule) Diameter' in mm. A linguistic variable obtains fuzzy set values with isosceles triangular membership functions. A triangular fuzzy membership function is denoted by [a,b,c], where 'a' and 'c' indicate a triangle's basis moreover 'b' corresponds to a triangle's top. The 'AN Melt Flow' input variable obtains the values $[12, 15, 18]$, $[17, 20, 23]$, and $[22, 25, 28]$. Moreover, the 'Recycled Fertilizer' input variable obtains the values $[8.5, 10, 11.5]$, $[11, 12.5, 14]$, and $[13.5, 15, 16.5]$. Note that the nine fuzzy rules in Fig.3 "fully cover" the input data domain, in other words any input data pair (x, y) activates at least one fuzzy rule. Finally, the '(Fertilizer Granule) Diameter' output variable obtains the values $[-1.5, 0.5, 2.5]$, $[2, 3.25, 4.5]$, and $[3, 5, 7]$. In this example 'min' conjunction, 'max' disjunction, 'min' implication, 'max' aggregation, and 'centroid' defuzzification have been employed. For a grid of input data pairs (x, y) in the domain $[15, 25] \times [10, 15]$ the output variable surface shown in Fig.5(a) was computed.

(Fig.3 goes here, or below)

Next we reduced the number of rules by an order of magnitude by ignoring six of the rules in Fig.3; hence the FIS in Fig.4 emerged with three fuzzy rules. Note in Fig.4 that an input data pair may be outside all fuzzy rule supports. The latter input data pairs were conventionally mapped to the middle of the output data range $[0.5, 5]$, i.e they were mapped to number $(0.5+5)/2 = 2.75$. The output surface in this case is shown in Fig.5(b); the corresponding mean square error (MSE) equals MSE=0.7888.

(Fig.4 goes here, or below)

The effectiveness of the fuzzy membership function $m_F(x) = 1/(1+d_1(F, x))$ was evaluated next. Fig.5(c) shows the corresponding output surface; the MSE in this case equals MSE=0.6283.

(Fig.5 goes here, or below)

Next we evaluated a combination of (1) standard fuzzy logic FIS techniques, and (2) fuzzy rule activation using the fuzzy membership function $m_F(x) = 1/(1+d_1(F, x))$. More specifically if a rule was activated (in a standard fuzzy logic sense) more than a user-defined threshold T_f then standard fuzzy logic FIS techniques were employed to compute the (real number) output; otherwise, the fuzzy membership function $m_F(x) = 1/(1 + d_1(F, x))$ was employed. The threshold T_f varied from 0 to 1 in steps of 0.05. The corresponding MSE samples are indicated in Fig.6 by solid diamonds (♦). In particular, $T_f = 0$ means that standard fuzzy logic FIS techniques were used for a datum x within a fuzzy rule support, otherwise the fuzzy membership function $m_F(x) = 1/(1 + d_1(F, x))$ was used. With an increasing T_f the MSE initially drops until a global minimum value of MSE=0.6084 at $\widehat{T}_f = 0.35$; then the MSE increases asymptotically, as expected; more specifically, $T_f = 1$ means that the fuzzy membership function $m_F(x) = 1/(1 + d_1(F, x))$ is used all along. Fig.5(d) shows the corresponding output surface for $T_f = 0.35$.

(Fig.6 goes here, or below)

The example above was meant to demonstrate the capacity of the proposed tools. Note that the proposed FIS techniques reduced the MSE significantly by 20.34% from 0.7888 down to 0.6283. It is interesting that a further, marginal improvement by 3.16% from 0.6283 down to 0.6084 resulted in by a combination of standard fuzzy logic FIS techniques with the proposed FIS techniques.

C. CALFIN: An algorithm for computing a FIN

This section summarizes an algorithm for computing a *FIN* from a population of measurements [9].

Algorithm CALFIN

1) Let $x = [x_1, x_2, \ldots, x_N]$ be a vector with real number entries.
2) Order incrementally the entries of vector x.
3) Initially vector pts is empty.
4) function *calfin*(x)
5) { while $(dimension(x) \neq 1)$
6) med:= *median*(x)[3]
7) insert med in vector pts
8) x_left:= elements in vector x less-than number *median*(x)
9) x_right:= elements in vector x larger-than number *median*(x)
10) *calfin*(x_left)
11) *calfin*(x_right)
12) endwhile
13) } //function *calfin*(x)
14) Sort the entries of vector pts incrementally.
15) Let '$\dim(pts)$' denote the dimension of vector pts. Store in vector val, $\dim(pts)/2$ numbers from 0 up to 1 in steps of $2/\dim(pts)$ followed by another $\dim(pts)/2$ numbers from 1 down to 0 in steps of $2/\dim(pts)$.

Algorithm CALFIN computes two vectors, namely pts and val, where vector val includes the degrees of fuzzy membership of the corresponding real numbers in vector pts. A *FIN*'s membership function is constructed by line-segment connecting the points with coordinates in vectors pts and val. Hence, algorithm CALFIN computes a *FIN* with piecewise linear membership function; moreover the aforementioned function is either strictly increasing or strictly decreasing at a point on the corresponding fuzzy number support. Note that the maximum membership function value of 1 is attained at exactly one number.

It turns out that, asymptotically, for large N, $100(1-h)\%$ of the entries in vector pts are in the interval support $\{x : F_h \neq 0\}$, whereas the remaining $100h\%$ entries in vector pts are split equally both to the left and to the right of interval $\{x : F_h \neq 0\}$. In the aforementioned sense the interval $\{x : F_h \neq 0\}$ constitutes, by construction, "an interval of confidence at level-h". Note also that, due to the one-one correspondence between *FIN*s and probabilistic distribution functions (PDFs) [9], a *FIN* can capture statistics of all orders.

[3]The *median*(x) of a vector $x = [x_1, x_2, \ldots, x_N]$ is a number such that half of the N numbers $x_1, x_2, \ldots, x_N$ are smaller than *median*(x) and the other half are larger than *median*(x); for instance, the *median*$([x_1, x_2, x_3])$ with $x_1 < x_2 < x_3$ equals x_2, whereas the *median*$([x_1, x_2])$ with $x_1 < x_2$ was calculated here as *median*$([x_1, x_2]) = (x1 + x2)/2$.

D. grSOM*: A FIS algorithm for function approximation by classification*

A specific FIS, namely *granular Self-Organizing Map* or *grSOM* for short, is presented in the following as a fuzzy extension of Kohonen's Self-Organizing Map (KSOM) to a fuzzy number domain. The learning phase of algorithm *grSOM* is shown first for structure identification, followed by the classification phase for generalization.

Algorithm *grSOM* for structure identification

1) Define the dimensions I and J of a two-dimensional grid of $I \times J$ units. Each unit can store both a N-tuple (weight) W_{ij}, $i = 1, \ldots, I$, $j = 1, \ldots, J$ of fuzzy interval numbers (*FIN*s) as well as a category label $L_{ij} \in \mathsf{L} = \{l_1, \ldots, l_c\}$, where c is the total number of categories.
2) Initialize randomly the weight of each unit by a training datum.
   Repeat steps 3 and 4 below for a user-defined integer number *Nepochs* of epochs.
3) For each input datum $(x_k, y_k) \in \mathsf{F}^N \times \mathsf{L}$, $k = 1, \ldots, n$, do
   • Compute the Minkowski metric $d_1(x_k, W_{ij})$, $i = 1, \ldots, I$, $j = 1, \ldots, J$.
   • Competition among the $I \times J$ units in the grid: Winner is the unit '$I_0 J_0$' whose weight is included "the most" in x_k, i.e.
   $$I_0 J_0 \doteq \arg \max_{i \in \{1, \ldots I\}, j \in \{1, \ldots J\}} (1/(1 + d_1(x_k, W_{ij})).$$
   • Assign input x_k to both the winner unit and to all the units in the neighborhood of the winner.
4) Use algorithm CALFIN to re-compute the weight W_{ij}, $i = 1, \ldots, I$, $j = 1, \ldots, J$ based on the data assigned to the corresponding unit in step-3 of the current epoch.
5) To each unit 'ij', $i = 1, \ldots, I$, $j = 1, \ldots, J$ in the grid, assign the label of the category which provided the majority of the input data to the unit 'ij' in question during all epochs.

The above algorithm employs a "greedy" version of the *grSOM* algorithm, which (greedy version) guarantees full coverage of the training data domain as described in [12]. Nevertheless, a basic difference here is that the algorithm presented in this work bases its decision-making on the fuzzy membership function $1/(1 + d_1(.,.))$, whereas the *grSOM* algorithm in [12] bases its decision-making directly on the Minkowski metric $d_1(.,.)$. In the aforementioned sense the *grSOM* algorithm presented in this work is a fuzzy inference system (FIS), whereas the *grSOM* algorithm in [12] is not.

After termination of the above algorithm a unit 'ij' in the grid with weight W_{ij} is assigned a category label L_{ij}. The following fuzzy rules are induced naturally from the data: "if W_{ij} then L_{ij}", $i = 1, \ldots, I$, $j = 1, \ldots, J$. The testing phase of algorithm *grSOM* is described in the following.

Algorithm *grSOM* for generalization

1) Present a new input $x_0 \in \mathsf{F}^N$.

2) Competition: Compute the Minkowski metric $d_1(x_0, W_{ij})$ for all the weights W_{ij}, $i = 1, \ldots, I$, $j = 1, \ldots, J$ in the grid. Winner is the unit 'I_0J_0' whose weight is included "the most" in x_0, i.e.

$$I_0J_0 \doteq \arg \max_{i \in \{1,\ldots I\}, j \in \{1,\ldots J\}} (1/(1 + d_1(x_0, W_{ij})).$$

3) Assign category label $L_{I_0J_0}$ to input x_0.

We point out that algorithm *grSOM* is an algorithm for classification. Note that function approximation, in particular regression, by classification is a common practice in machine learning [23]. Moreover, note that system modeling by classification based on fuzzy interval numbers (*FIN*s) has already been employed successfully; for instance, in [33] the *FIN* k-Nearest Neighbor (*FINkNN*) classifier has been employed for predicting industrial sugar production by classification.

V. An Industrial Modeling Application

The objective in this section is to demonstrate the advantageous applicability of the proposed techniques in a real world problem. In particular, we describe modeling the *pan granulator* mill in the Phosphoric Fertilizers Industry (PFI) at Nea Karvali, Greece. The industrial problem is outlined in the following.

A. The industrial problem

The industrial production of nitrogenous fertilizers includes sequentially two processes, namely "Wet Process" and "Dry Process". The former process produces an Ammonium Nitrate (AN) solution, which is fed to the latter one. More specifically, a highly concentrated hot AN melt is sprayed to the rotating *pan granulator* from a spraying nozzle manifold. The fertilizer end-product consists of small fertilizer granules each having size in the range of a few millimeters. The aforementioned size, as well, determines industrial fertilizer quality. A desired quality size can be obtained by tuning the values of several pan granulator *operating parameters/variables* including: AN melt flow, AN melt pressure, pan speed/inclination, volume of recycled fertilizer, nozzle location, etc. Optimal parameter values are constantly sought as explained in the following.

The PFI operates around the clock, and a specific fertilizer production order is carried out from within a few hours to within several days. It turns out that an optimal set of parameter values has to be sought after switching production from one fertilizer type to a different one. Moreover, various disturbances during the industrial production may call for additional tuning.

Tuning in the industry is currently carried out manually by human operators by trial-and-error; a robust feedback automatic control mechanism will be helpful. Therefore, a dependable open loop model of the pan granulator will be useful. This section describes a model of the form $d : \mathsf{R}^N \to \mathsf{R}$, where $d(x)$ denotes the average diameter of produced fertilizer granules and x is a N-dimensional vector of pan granulator operating parameters.

B. Data acquisition and model selection

Data samples have been collected during the last five years for several fertilizer types. More specifically, several pan granulator operating variables have been sampled manually every two hours around the clock. In addition, the corresponding average (fertilizer granule) diameter size has been recorded. All the data corresponded to a *steady state* operation of the pan granulator.

The data used in this work included samples of fourteen operating variables involved in the production of fertilizer type CaN26 during late April/early May 2003 in the Phosphoric Fertilizers Industry (PFI). The aforementioned variables are summarized in Table 1 including their corresponding units; in particular note that the units for the *Nozzle Vertical Distance* (rings) and the *Spraying Angle* (lines) in Table 1 are costumized units used in the industry. A total of 174 data vectors had been available. Twenty data vectors including one (or more) missing values were ignored.

(Table 1 goes here, or below)

The development of a "first principles" model was phased out due to the inherent complexity of the industrial process. Instead, we concentrated our efforts on inducing a model from the measurements. There is a number of system modeling alternatives including polynomial models, ARMA models, various neural network models, (AN)FIS models, etc. FIS models are well established in industrial system modeling applications due to both their capacity for dealing with ambiguity and their straightforward linguistic interpretations. Therefore we decided to use a FIS model in the context of this work.

C. Data Preprocessing

In a data preprocessing step some of the operating variables shown in Table 1 were ignored. Note that the selection of significant variables/features, known in the literature as "Type I structure identification problem" may be critical in system modeling applications. Using a recently published variable selection method based on a genetic algorithm [27], we have found the following six most important variables: 1) *AN Melt Flow*, 2) *Recycled Fertilizer*, 3) *AN Melt Pressure*, 4) *AN Melt Temperature*, 5) *Pan Rotation Speed*, and 6) *Nozzle Distance from the pan*. The practical significance of the aforementioned variables was confirmed by human operator experts from the industry. Note that a visual inspection of the operating variables samples revealed that the samples of both the *Coarse*- and the *Fine- Screen Vibration* variables have all been constant equal to 80%, hence our variable selection method ignored the latter variables right.

For illustrative purposes only the first three most significant operating variables were considered below, namely *AN Melt Flow* $[m^3/h]$, *Recycled Fertilizer* $[T/h]$, and *AN Melt Pressure* $[bar]$. The corresponding fertilizer granule sizes were classified in one of four categories, namely *small* (1 mm), *medium-small* (2 mm), *medium-large* (3.5 mm), and *large* (4.5 mm). In conclusion, 154 3-dimensional vectors were used in this work, each vector with a category label.

D. Experiments and results

Classifier *grSOM* was used, first, for structure indentification and, second, for generalization as described in the following. A $I \times J = 4 \times 4$ size grid of units was employed, where each unit had no more than four neighbors. For demonstration purposes a random permutation of 120 3-dimensional vectors was employed for structure identification by clustering using the *grSOM* algorithm, whereas the remaining 34 vectors were employed for generalization. Each unit weight in the grid was initialized by assigning to it, randomly, a training datum; note that a 3-dimensional input vector was dealt with here as a trivial 3-dimensional fuzzy number.

Initially the neighborhood size of the winner unit was chosen large enough so as to enclose half of the grid. Progressively, after a number of epochs, the neighborhood size of the winner unit decreased and, eventually, an input was assigned only to the winner unit. A number $Nepochs = 10,000$ of epochs was user-defined.

Nine fuzzy rules have been computed by algorithm *grSOM*, more specifically 2, 2, 2, and 3 rules have been computed for categories *small*, *medium-small*, *medium-large*, and *large*, respectively. Four of the aforementioned fuzzy rules are shown in Fig.7, where the "IF part" of a fuzzy rule (including three fuzzy numbers) is shown plotted, moreover the corresponding "THEN part" (i.e. a category label) is written explicitly on the top of a plot.

(Fig.7 goes here, or below)

Algorithm *grSOM* for generalization gave a 97% percentage of success in this experiment; only one datum was misclassified in category *medium-small* instead of its correct category *medium-large*. The metrics $d_K(.,.)$ between various fuzzy numbers in Fig.7 are shown in Table 2. In addition, within parentheses in Table 2 are shown the corresponding fuzzy membership values $m_{Ri3}(Rj3) = \frac{1}{1+d_K(Ri3,Rj3)}$, where $i,j \in \{1,2,3,4\}$. More specifically, Table 2 shows the metrics $d_K(.,.)$ between the fuzzy numbers $R13$, $R23$, $R33$, and $R43$ induced by classifier *grSOM* along the *AN Melt Pressure* data dimension using the uniform mass function $m_1(x) = 1$. The interested reader may confirm by visual inspection that the numbers displayed in Table 2 sensibly quantify the "proximity" of the corresponding fuzzy numbers shown in Fig.7.

(Table 2 goes here, or below)

The aforementioned uniform mass function $m_1(x) = 1$ is the one used implicitly throughout the literature. This work has shown above that alternative mass functions can be employed for "distorting" non-linearly the metric between fuzzy numbers. For instance, Fig.8(b) shows mass function $m_2(x) = 0.1 + \frac{e^{-27(x-0.9)}}{[1+e^{-27(x-0.9)}]^2}$ along the *AN Melt Pressure* data dimension, whereas Table 3 shows the corresponding metrics between fuzzy numbers $R13$, $R23$, $R33$, and $R43$, pairwise. Within parentheses in Table 3 are shown the corresponding fuzzy membership values $m_{Ri3}(Rj3) = \frac{1}{1+d_K(Ri3,Rj3)}$, $i,j \in \{1,2,3,4\}$. A comparison of Tables 2 and 3 reveals how an underlying mass function can change drastically the proximity of fuzzy numbers. In particular, note that the distances between the fuzzy numbers $R13$, $R23$, $R33$ have not changed considerably, whereas the distances between fuzzy number $R43$ and the other ones have changed considerably. The reason is that the first three fuzzy numbers stand near the peak of the Gaussian mass function $m_2(t)$ (Fig.8(b)), whereas the fourth fuzzy number $R43$ stands further to the right tail of mass function $m_2(t)$. Note also that Tables 2 and 3 are diagonal symmetric, as expected, due to the *commutative property* of metric $d_K(.,.)$.

(Fig.8 goes here, or below)

(Table 3 goes here, or below)

E. Comparative experimental results

A series of experiments was carried out using different classifiers in order to demonstrate comparatively the effectiveness of the *grSOM* classifier in this real-world application. Five classifiers have been employed, namely (1) *grSOM* using $1/(1 + d_1(.,.))$, as described in this work, (2) *grSOM* using $d_1(.,.)$, as described in [12], (3) Kohonen's SOM (KSOM), (4) a conventional fuzzy inference system (FIS) using fuzzy sets with triangular fuzzy membership functions, and (5) backpropagation. For each classifier a set of ten different data partitions was used; each partition was produced from a random permutation of 154 data vectors, where the first 120 data vectors were used for training and the remaining 34 data vectors were used for testing.

First, classifier *grSOM* was employed as described in this work using the fuzzy membership function $1/(1 + d_1(.,.))$. Second, classifier *grSOM* was employed using the Minkowski metric $d_1(.,.)$ as described in [12]. Third, Kohonen's SOM (KSOM) algorithm was employed. All aforementioned SOM algorithms used a 4×4 grid of units. Fourth, we employed a conventional FIS, which located clusters in the training data and put a fuzzy set with isosceles triangular membership function on a cluster. The latter membership functions were initialized by trivial *FIN*s computed by KSOM; then, both triangle spreads and triangle top locations were fine-tuned using a steepest descent algorithm on the training data. Fifth, a conventional backpropagation neural network was employed with 3 inputs, 5 hidden layer neurons, and 3 (binary) outputs; sigmoid transfer functions were used, furthermore training was carried out using the resilient backpropagation algorithm with mean square error (MSE) target 0.01 and maximum number of training epochs 1,000.

Table 4 summarizes the experimental results. More specifically, Table 4 shows the classification accuracy average as well as the corresponding standard deviation for ten different data partitions. Furthermore, Table 4 shows the average number of engaged grid units /rules as well as the corresponding standard deviations, where applicable.

(Table 4 goes here, or below)

Backpropagation marginally produced the best classification results on the average but without inducing descriptive decision-making knowledge (rules). The two *grSOM* classifiers scored similarly; note that, given the corresponding standard deviations, the marginally better classification accuracy by '*grSOM* using $1/(1 + d_1(.,.))$' does not appear to be statistically significant. The classification performance of the Triangular FIS was slightly inferior, moreover the corresponding number of induced rules was slightly smaller. The latter was attributed to the employed steepest descent algorithm which also included pruning. The worse performance of the Triangular FIS compared to either *grSOM* model was attributed to the restrictive (triangular) shape of the employed fuzzy membership functions; in other words, the nonparametric shape of a general *FIN* used by either *grSOM* model appears to contribute to an improvement of classification performance is this real-world application. Finally, KSOM produced the poorest classification results as well as the largest number of engaged grid units. The better performance of *grSOM* compared to KSOM was attributed to the fact that a *FIN* captures (locally) statistics of all orders in the training data, whereas the KSOM captures only first order statistics as discussed in [12].

The marginally better performance of backpropagation compared to a *grSOM* classifier was attributed to the small number (120) of 3-dimensional data vectors used for training. More specifically, an average of approximately 8 rules per 120 data implies 15 data per rule - Actually, in our experiments, the number of data per rule varied between 7 and 19. Since a *FIN* represents a local probabilistic distribution, it follows that construction of a *FIN* from a small number of data may not represent a data distribution accurately thus deteriorating performance slightly compared to backpropagation. Note that in recent experiments regarding a different problem, where over one hundred data were used for computing a *FIN*, *grSOM* produced better results than backpropagation as it will be shown elsewhere. However, a *grSOM* classifier in this work has clearly produced better results than either Triangular FIS or KSOM classifier (Table 4). The latter was attributed to a more accurate representation of a local data distribution by a *FIN* than by either a triangle (in Triangular FIS) or a single vector (in KSOM); in particular, neither a triangle nor a single vector can represent higher order statistics [12].

The computation time on a standard personal computer platform for backpropagation was under half a minute, whereas KSOM required a few minutes of computation. A *grSOM* classifier appeared to be eight to ten times slower than KSOM. The need for longer training for a *grSOM* classifier is due to the employment of *FIN*s: first, it takes longer to compute a N-dimensional *FIN* vector than to compute a N-dimensional number vector average and, second, it takes longer to compute a distance d_K than to compute the conventional L2 (Euclidean) distance. The computation time for Triangular FIS was measured to be between the corresponding times of KSOM and *grSOM* classifiers, as expected.

Some further computational experiments were carried out to illustrate further advantages.

F. Further advantages

An artificial input datum $X = (X1, X2, X3)$, shown in Fig.8(a), was fed to the *grSOM* for generalization. More specifically, X includes *heterogeneous data* [30], namely a real number ($X1$), a fuzzy set ($X2$) with isosceles triangular membership function, and an interval ($X3$). Note that datum X does not activate, in the conventional FIS sense, any fuzzy rule in Fig.7 because at least one of the entries $X1$, $X2$, or $X3$ is outside the corresponding interval supports of the fuzzy rules in Fig.7. Nevertheless, using the function $1/(1 + d_1(Xj, R_{ij}))$, $i = 1, 2, 3, 4$, $j = 1, 2, 3$ a degree of activation for a fuzzy rule can be computed.

Another advantage demonstrated here is the capacity to employ heterogeneous data including real numbers, fuzzy numbers, and intervals in any combination. In the aforementioned manner we may compensate for ambiguities in the data; the latter is potentially significant in industrial (and other) applications.

A further advantage demonstrated here is the capacity to employ alternative mass functions. For instance the first three lines in Table 5 show the distances $d_K(.,.)$ between entries of vector $X = (X1, X2, X3)$ and the corresponding entries of the four rules in Fig.7 using the mass function $m_1(x) = 1$; the last line in Table 5 shows the distances between interval $X3$ and the corresponding fuzzy numbers of the four fuzzy rules along the *AN Melt Pressure* data dimension using the mass function $m_2(x) = 0.1 + \frac{e^{-27(x-0.9)}}{[1+e^{-27(x-0.9)}]^2}$ (Fig.8(b). Within parentheses in Table 5 are shown the corresponding fuzzy membership values $m_{Rij}(Xj) = \frac{1}{1+d_K(Rij,Xj)}$, where $i \in \{1, 2, 3, 4\}$, $j \in \{1, 2, 3\}$. The interesting point here is that using mass function $m_1(x)$ it follows that $X3$ is clearly "nearest" to fuzzy number $R13$ (the latter is in category *small*), whereas using mass function $m_2(x)$ it follows that $X3$ is clearly "nearest" to fuzzy number $R43$ (the latter is in category *large*). Hence, a mass function can drastically change the outcome of classification.

(Table 5 goes here, or below)

VI. Conclusion

In this work we have introduced new perspectives and useful tools for enhanced FIS analysis and design. It was shown that: a FIS is typically used for implementing a function $f : \mathsf{R}^N \rightarrow \mathsf{R}^M$; the cardinality of the set of FISs equals $\aleph_2$; moreover, a FIS has a capacity for (local) generalization. Fuzzy interval numbers (*FIN*s) were presented as an alternative, computationally tractable representation of (conventional) fuzzy numbers.

More specifically, based on *generalized interval* analysis, a tunable metric d_K was presented between fuzzy numbers; furthermore a *mass function* can be used for tuning d_K. A novel FIS design was proposed based on metric d_K with specific advantages including an alleviation of the curse of dimensionality problem (by generalization beyond fuzzy rule support), a capacity to rigorously cope with heterogeneous data including (fuzzy) numbers and intervals, etc.

A specific FIS algorithm was presented, namely *granular Self-Organizing Map (grSOM)*, for classification. The *grSOM* can induce fuzzy rules involving fuzzy numbers characterized by non-parametric membership functions. This work has demonstrated computationally (using both artificial data and real world data from an industrial modeling application) practical advantages of the proposed techniques over alternative classification models.

There is evidence in the literature that *FIN*s can produce better results than real numbers in classification problems [33]. Further improvements will be sought in the future especially regarding optimization of the (underlying) mass functions; note that it was demonstrated lately that "genetically computed" mass functions can improve performance in classification applications [12]. Automatic control [28] is a promising application domain of the novel tools presented here. In a different direction note that the metric $d_K(.,.)$ can be employed for calculating a metric between type-2 fuzzy sets [14] by the calculation of an additional integral. Furthermore, the tools presented here may be particularly useful for designing FIS classifiers [19] as well as other decision support systems [21].

APPENDIX A: ELEMENTS FROM LATTICE THEORY

Here we give an overview of useful concepts and results from lattice theory.

Definition A.1: Given a set P, a binary relation $\leq$ between elements of P is called a *partial order* if it satisfies the following conditions for all $x, y, z \in P$:

1) *Reflexivity*: $x \leq x$.
2) *Antisymmetry*: $(x \leq y$ and $y \leq x) \Rightarrow x = y$.
3) *Transitivity*: $(x \leq y$ and $y \leq z) \Rightarrow x \leq z$.

We will sometimes write $y \geq x$, which is equivalent to $x \leq y$.

Definition A.2: If $\leq$ is a partial order on P then we say that $(P, \leq)$ is a *partially ordered set* or, equivalently, a *poset*.

Definition A.3: A *lattice* is a poset $(\mathsf{L}, \leq)$ with the additional property that any two of its elements have a greatest lower bound (g.l.b.), and a least upper bound (l.u.b.) in L.

Notation A.4: Given a *lattice* $(\mathsf{L}, \leq$, and any two elements $x, y \in \mathsf{L}$, their g.l.b. is called the *meet* of x and y and denoted by $x \wedge y$; their l.u.b. is called the *join* of x and y and denoted by $x \vee y$.

Definition A.5: We say that x and y are *comparable* when either $x \leq y$ *or* $y \leq x$; otherwise we say that they are *incomparable* and we write $x||y$.

A lattice without incomparable elements is called *totally ordered* lattice. For example, a totally ordered lattice is the set R of real numbers.

Recall the concepts of *metric* and *positive valuation* in the following.

Definition A.6: A *metric* in a set S is a nonnegative real function $d : S \times S \rightarrow \mathsf{R}_0^+$ which, for all $x, y, z \in S$, satisfies:

D1a $d(x, x) = 0$.
D1b $d(x, y) = 0 \Rightarrow x = y$.
D2 $d(x, y) = d(y, x)$.
D3 $d(x, y) \leq d(x, z) + d(z, y)$.

If only conditions **D1a, D2** and **D3** are satisfied, then 'd' is called a *pseudometric*.

Definition A.7: A *valuation* in a lattice $(\mathsf{L}, \leq)$ is a function $v : \mathsf{L} \rightarrow \mathsf{R}$ which, for all $x, y \in \mathsf{L}$, satisfies:

$$v(x) + v(y) = v(x \wedge y) + v(x \vee y).$$

A valuation is called *positive* if, for all $x, y \in \mathsf{L}$, we have

$$x < y \Rightarrow v(x) < v(y).$$

Proposition A.8: Let $(\mathsf{L}, \leq)$ be a lattice and v be a positive valuation; then

$$d(x, y) = v(x \vee y) - v(x \wedge y)$$

is a *metric* .

APPENDIX B: ELEMENTS FROM FUZZY LATTICE THEORY

Here we give an overview of useful concepts and results from fuzzy lattice theory.

Definition B.1: A *fuzzy lattice* is a triple $(\mathsf{L}, \leq, \mu)$, where $(\mathsf{L}, \leq)$ is a lattice and μ is a fuzzy relation $\mu : \mathsf{L} \times \mathsf{L} \rightarrow [0, 1]$ such that

$$\mu(x, y) = 1 \Leftrightarrow x \leq y.$$

When $x \leq y$ then the fuzzy relation μ holds to the maximum degree (i.e. 1) between x and y; but μ may also hold to a lesser degree between x and y even when $x||y$ (i.e. x and y are incomparable). Hence μ can be understood as a *weak (fuzzy) partial order* relation. In particular, μ possesses a very weak form of transitivity: when both $\mu(x, y) = 1$ and $\mu(y, z) = 1$, then we also have $\mu(x, z) = 1$; but if either $\mu(x, y) \neq 1$ or $\mu(y, z) \neq 1$, then $\mu(x, z)$ can take any value in $[0, 1]$. Hence our definition of fuzzy lattice is quite general. A fuzzy relation which can be used to construct a fuzzy lattice is the so-called *inclusion measure*.

Definition B.2: Given a lattice $(\mathsf{L}, \leq)$, an *inclusion measure* is a fuzzy relation $\sigma : \mathsf{L} \times \mathsf{L} \rightarrow [0, 1]$ which satisfies the following conditions for every $x, y, z \in \mathsf{L}$.

C1 $\sigma(x, x) = 1$.
C2 $z \leq x \Rightarrow \sigma(y, z) \leq \sigma(y, x)$.
C3 $x \wedge y < x \Rightarrow \sigma(x, y) < 1$.

Conditions **C1**, **C2**, and **C3** are interpreted as follows. **C1** means that every lattice element is fully included in itself. **C2** stipulates a common-sense "*consistency property*". **C3** requires that when x and y are incomparable then x is included in y to a degree less than one and, also, that when y is strictly included into x then x is included in y to a degree less than one. Note that in every lattice $(\mathsf{L}, \leq)$ we have the equivalence $x \wedge y < x \Leftrightarrow y < x \vee y$ [2]; hence **C3** can be replaced by the following equivalent condition

C3′ $y < x \vee y \Rightarrow \sigma(x, y) < 1$.

Proposition B.3: Let $(\mathsf{L}, \leq)$ be a lattice and $\sigma : \mathsf{L} \times \mathsf{L} \rightarrow [0, 1]$ be an inclusion measure on $(\mathsf{L}, \leq)$. Then $(\mathsf{L}, \leq, \sigma)$ is a fuzzy lattice.

Proposition B.4: Let $(\mathsf{L}, \leq)$ be a lattice and let $v : \mathsf{L} \rightarrow \mathsf{R}$ be a positive valuation. Then both functions

$$k(x, u) = \frac{v(u)}{v(x \vee u)}, \qquad s(x, u) = \frac{v(x \wedge u)}{v(x)}.$$

are inclusion measures.

APPENDIX C: PROOFS

In this Appendix we give the proofs of Propositions 3.8 and 3.10. These proofs depend on some definitions and results established in the companion paper [11]. The most crucial step is the introduction of *positive and negative* generalized intervals. The following definition subsumes Definition 3.3 as a special case.

Definition C.1: Take any $h \in (0, 1]$. A *positive generalized interval of height* h is a function $m^h_{x_1,x_2}$, where $x_1 \leq x_2$, defined by

$$m^h_{x_1,x_2}(x) = \begin{cases} h, & x_1 \leq x \leq x_2 \\ 0, & \text{otherwise.} \end{cases}$$

A *negative generalized interval of height* h is a function $m^h_{x_1,x_2}$, where $x_1 > x_2$, defined by

$$m^h_{x_1,x_2}(x) = \begin{cases} -h, & x_2 \leq x \leq x_1 \\ 0, & \text{otherwise.} \end{cases}$$

We will also denote a generalized interval $m^h_{x_1,x_2}$ by $[x_1, x_2]^h$.

Notation C.2: The family of all positive generalized intervals of height h will be denoted by M^h_+. The family of all negative generalized intervals of height h will be denoted by M^h_-. The family of all (positive and negative) generalized intervals of height h will be denoted by $\mathsf{M}^h = \mathsf{M}^h_+ \cup \mathsf{M}^h_-$.

The rationale for introducing *negative* generalized intervals is the following. As already mentioned in the text, "classic" intervals form a lattice. In this lattice the infimum of two non-intersecting intervals is the *empty interval*. We have found, in various practical applications, that this fact is rather restrictive. Therefore we have endeavored to construct a lattice of intervals where non-intersecting intervals have a nonempty infimum, furthermore a *positive valuation function* exists (for a definition of a *positive valuation function* see in Appendix A). As will be seen in the sequel, negative generalized intervals serve this purpose well with rewarding results.

Definition C.3: Given $h \in (0, 1]$, we define a relation $\preceq$ on $\mathsf{M}^h \times \mathsf{M}^h$ as follows:

if $[a, b]^h \in \mathsf{M}^h_+$, $[c, d]^h \in \mathsf{M}^h_+$ then:

$$[a, b]^h \preceq [c, d]^h \Leftrightarrow [a, b] \subseteq [c, d]$$

if $[a, b]^h \in \mathsf{M}^h_-$, $[c, d]^h \in \mathsf{M}^h_-$ then:

$$[a, b]^h \preceq [c, d]^h \Leftrightarrow [d, c] \subseteq [b, a]$$

if $[a, b]^h \in \mathsf{M}^h_-$, $[c, d]^h \in \mathsf{M}^h_+$ then:

$$[a, b]^h \preceq [c, d]^h \Leftrightarrow [b, a] \cap [c, d] \neq \varnothing$$

In all other cases $[a, b]^h$ and $[c, d]^h$ are *incomparable*, symbolically $[a, b]^h \parallel [c, d]^h$.

Proposition C.4: The relation $\preceq$ is an order on M^h. Moreover $\left(\mathsf{M}^h, \preceq\right)$ is a lattice, where the lattice join (denoted by $\curlyvee$) and the lattice meet (denoted by $\curlywedge$) are given by

$$[a, b]^h \curlyvee [c, d]^h = [a \wedge c, b \vee d]^h, \; [a, b]^h \curlywedge [c, d]^h = [a \vee c, b \wedge d]^h.$$

where $a \wedge c = min\{a, c\}$ and $a \vee c = max\{a, c\}$.

Proof: The proof appears in [11]. ∎

Next we define a metric on M^h. This is effected by a standard lattice-theoretic construction, which makes use of a *positive valuation function*.

Proposition C.5: Let $f_h : \mathsf{R} \rightarrow \mathsf{R}$ be a strictly increasing function. Then the function $v_h : \mathsf{M}^h \rightarrow \mathsf{R}$ given by

$$v_h([a, b]^h) = f_h(b) - f_h(a)$$

is a *positive valuation* in $\left(\mathsf{M}^h, \preceq\right)$. Furthermore, the function $d_h : \mathsf{M}^h \times \mathsf{M}^h \rightarrow \mathsf{R}^+_0$ given by

$$d_h([a, b]^h, [c, d]^h) = [f_h(a \vee c) - f_h(a \wedge c)] + [f_h(b \vee d) - f_h(b \wedge d)]$$

is a *metric* on M^h.

Proof: The proof appears in [11]. ∎

We have shown a metric d_h on M^h. Now we modify d_h to obtain a metric on M^h_0.

Proof of Proposition 3.8. We want to show that $\overline{d}_h$ is a metric on M^h_0. In other words, we must show that

$$\overline{d}_h(F_h, G_h) = 0 \Leftrightarrow F_h = G_h \tag{4}$$

$$\overline{d}_h(F_h, G_h) = \overline{d}_h(G_h, F_h) \tag{5}$$

$$\overline{d}_h(F_h, H_h) \leq \overline{d}_h(F_h, G_h) + \overline{d}_h(G_h, H_h) \tag{6}$$

for $F_h, G_h, H_h \in M^h_0$.

Now, for $F_h, G_h, H_h \in M^h_+$, (4)-(6) are true because $\overline{d}_h$ is identical to d_h on M^h_+ and d_h is a metric on $M^h \supset M^h_+$. Also, $\overline{d}_h\left(\phi^h, \phi^h\right) = 0$ by (3), and $\overline{d}_h\left([a, b]^h, \phi^h\right) = \overline{d}_h\left(\phi^h, [a, b]^h\right)$ by (2). Hence it remains to show

$$\overline{d}_h(F_h, H_h) \leq \overline{d}_h(F_h, G_h) + \overline{d}_h(G_h, H_h)$$

where at least one of F_h, G_h, H_h is ϕ^h. Indeed, for $F_h, H_h \in M^h_+$ we have

$$\overline{d}_h(F_h, H_h) < \overline{d}_h\left(F_h, \phi^h\right) + \overline{d}_h\left(\phi^h, H_h\right) = \infty + \infty$$

Moreover, for any G_h we have

$$\overline{d}_h\left(F_h, \phi^h\right) = \infty = \overline{d}_h(F_h, G_h) + \overline{d}_h\left(G_h, \phi^h\right)$$

This completes the proof. ∎

Proof of Proposition 3.10 We want to show that d_K is a pseudometric on F^*.

From both the definition

$$d_K(F, G) = c \int_0^1 \overline{d}_h(F_h, G_h) dh$$

and the fact that $\overline{d}_h$ is a metric, using standard properties of integrals it follows

$d_K(F, F) = 0$, $d_K(F, G) = d_K(G, F)$, and
$d_K(F, H) \leq d_K(F, G) + d_K(G, H)$.

Nevertheless, from

$$d_K(F,G) = c\int_0^1 \overline{d}_h(F_h,G_h)dh = 0$$

we *cannot* conclude that $F = G$, because we may have $\overline{d}_h(F_h,G_h) \neq 0$ for an isolated point h_0 or, more generally, on a set of measure zero. Hence we have proved that d_K is a pseudometric, but not necessarily a metric.■

Acknowledgement

The data used in this work is a courtesy of Phosphoric Fertilizers Industry (PFI) at Nea Karvali, Greece. We are especially thankful to Mr. Michael Theodorides, head of the Engineering Division for enabling our access to the industry. We also thank Mr. A. Terzis, head of the Nitric Acid Division for supplying the data used in this work.

REFERENCES

[1] I. Batyrshin, O. Kaynak, I. Rudas, "Fuzzy modeling based on generalized conjunction operations," *IEEE Trans. Fuzzy Systems*, vol. 10, no. 5, pp. 678- 683, 2002.

[2] G. Birkhoff, *Lattice Theory*. Providence, RI: American Mathematical Society, Colloquium Publications, vol. 25, 1967.

[3] C. Elkan, "The paradoxical success of fuzzy logic," *IEEE Expert*, vol. 9, no. 4, pp. 3-8, 1994.

[4] B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. Heidelberg, Germany: Springer-Verlag, 1999.

[5] H. Ishibuchi, K. Kwon, and H. Tanaka, "A learning algorithm of fuzzy neural networks with triangular fuzzy weights," *Fuzzy Sets and Systems*, vol. 71, no. 3, pp. 277-293, 1995.

[6] J.-S. R. Jang and C.-T. Sun, "Functional equivalence between radial basis function networks and fuzzy inference systems," *IEEE Trans. Neural Networks*, vol. 4, no. 1, pp. 156-159, 1993.

[7] J.-S. R. Jang and C.-T. Sun, "Neuro-fuzzy modeling and control," *Proceedings of the IEEE*, vol. 83, no. 3, pp. 378-406, 1995.

[8] V.G. Kaburlasos, "Novel fuzzy system modeling for automatic control applications," *Proc.* 4^{th} *Intl. Conf. on Technology & Automation*, Thessaloniki, Greece, 5-6 Oct. 2002, pp. 268-275.

[9] V.G. Kaburlasos, "*FIN*s: Lattice theoretic tools for improving prediction of sugar production from populations of measurements," *IEEE Trans. Systems, Man and Cybernetics – Part B*, vol. 34, no. 2, pp. 1017-1030, 2004.

[10] V.G. Kaburlasos and A. Kehagias, "Novel analysis and design of fuzzy inference systems based on lattice theory," *IEEE International Conference on Fuzzy Systems* (*FUZZ-IEEE 2004*), Budapest, Hungary, 25-29 July 2004, pp. 281-286.

[11] V.G. Kaburlasos and A. Kehagias, "Novel fuzzy inference system (FIS) analysis and design based on lattice theory. Part I: Working principles," *Intl. J. General Systems*, 2006 (*in press*).

[12] V.G. Kaburlasos and S.E. Papadakis, "Granular self-organizing map (grSOM) for structure identification," *Neural Networks*, 2006 (*in press*).

[13] V.G. Kaburlasos and V. Petridis, "Fuzzy Lattice Neurocomputing (FLN) models," *Neural Networks*, vol. 13, no. 10, pp. 1145-1170, 2000.

[14] N.N. Karnik, J.M. Mendel, and Qilian Liang, "Type-2 fuzzy logic systems," *IEEE Trans. Fuzzy Systems*, vol. 7, no. 6, pp. 643-658, 1999.

[15] H.M. Kim and J.M. Mendel, "Fuzzy basis functions: comparisons with other basis functions," *IEEE Trans. Fuzzy Systems*, vol. 3, no. 2, pp. 158-168, 1995.

[16] G. J. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall, PTR, Upper Saddle River, NJ, 1995.

[17] B. Kosko, "Fuzzy systems as universal approximators," *IEEE Trans. Computers*, vol. 43, no. 11, pp. 1329-1333, 1994.

[18] R. Krishnapuram and J.M. Keller, "A possibilistic approach to clustering," *IEEE Trans. Fuzzy Systems*, vol. 1, no. 2, pp. 98-110, 1993.

[19] H.-M. Lee, C.-M. Chen, J.-M. Chen, Y.L. Yu-Lu Jou, "An efficient fuzzy classifier with feature selection based on fuzzy entropy," *IEEE Trans. Systems, Man and Cybernetics – Part B*, vol. 31, no. 3, pp. 426-432, 2001.

[20] E.H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *Intl. J. Man-Machine Studies*, vol. 7, pp. 1-13, 1975.

[21] L. Mikhailov and M.G. Singh, "Fuzzy analytic network process and its application to the development of decision support systems," *IEEE Trans. Systems, Man and Cybernetics – Part C*, vol. 33, no. 1, pp. 33-41, 2003.

[22] S. Mitaim and B. Kosko, "The shape of fuzzy sets in adaptive function approximation," *IEEE Trans. Fuzzy Systems*, vol. 9, no. 4, pp. 637-656, 2001.

[23] T.M. Mitchell, *Machine Learning*. New York, NY: McGraw-Hill, 1997.

[24] S. Mitra and Y. Hayashi, "Neuro-fuzzy rule generation: survey in soft computing framework," *IEEE Trans. Neural Networks*, vol. 11, no. 3, pp. 748-768, 2000.

[25] D. Nauck and R. Kruse, "Neuro-fuzzy systems for function approximation," *Fuzzy Sets and Systems*, vol. 101, no. 2, pp. 261-271, 1999.

[26] H.T. Nguyen and A. Walker. *First course in fuzzy logic*. CRC Press, 1999.

[27] S.E. Papadakis, P. Tzionas, V.G. Kaburlasos, and J.B. Theocharis, "A genetic based approach to the Type I structure identification problem," *Informatica*, vol. 16, no. 3, pp. 365-382, 2005.

[28] K. Passino and S. Yurkovich, *Fuzzy Control*. Reading, MA: Addison Wesley Longman, 1998.

[29] W. Pedrycz, "Why triangular membership functions?," *Fuzzy Sets and Systems*, vol. 64, no. 1, pp. 21-30, 1994.

[30] W. Pedrycz, J.C. Bezdek, R.J. Hathaway, and G.W. Rogers, "Two parametric models for fuzing heterogeneous fuzzy data," *IEEE Trans. Fuzzy Systems*, vol. 6, no. 3, pp. 411-425, 1998.

[31] V. Petridis and V.G. Kaburlasos, "Fuzzy Lattice Neural Network (FLNN): A hybrid model for learning," *IEEE Trans. Neural Networks*, vol. 9, no. 5, pp. 877-890, 1998.

[32] V. Petridis and V.G. Kaburlasos, "Learning in the framework of fuzzy lattices," *IEEE Trans. Fuzzy Systems*, vol. 7, no. 4, pp. 422-440, 1999.

[33] V. Petridis and V.G. Kaburlasos, "*FINkNN*: A Fuzzy Interval Number k-Nearest Neighbor classifier for prediction of sugar production from populations of samples," *Journal of Machine Learning Research*, vol. 4, pp. 17-37, 2003.

[34] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1481-1497, 1990.

[35] T.A. Runkler, "Selection of appropriate defuzzification methods using application specific properties," *IEEE Trans. Fuzzy Systems*, vol. 5, no. 1, pp. 72-79, 1997.

[36] T.A. Runkler and J.C. Bezdek, "Function approximation with polynomial membership functions and alternating cluster estimation," *Fuzzy Sets and Systems*, vol. 101, no. 2, pp. 207-218, 1999.

[37] J.F. Sowa, *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Pacific Grove, CA: Brooks Cole Publishing Co., 2000.

[38] R.R. Stoll, *Set Theory and Logic*. New York, NY: Dover Publications, 1979.

[39] T. Tagaki and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 15, no. 1, pp. 116-132, 1985.

[40] K. Uehara and M. Fujise, "Fuzzy inference based on families of α-level sets," *IEEE Trans. Fuzzy Systems*, vol. 1, no. 2, pp. 111-124, 1993.

[41] V.N. Vapnik, "An overview of statistical learning theory," *IEEE Trans. Neural Networks*, vol. 10, no. 5, pp. 988-999, 1999.

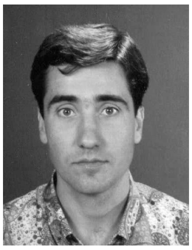
[42] L.X. Wang and J.M. Mendel, "Fuzzy basis functions, universal approximation, and orthogonal least-squares learning," *IEEE Trans. Neural Networks*, vol. 3, no. 5, pp. 807-814, 1992.

[43] C.-H. Wang, W.-Y. Wang, T.-T. Lee, and P.-S. Tseng, "Fuzzy B-spline membership function (BMF) and its applications in fuzzy-neural control," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 25, no. 5, pp. 841-851, 1995.

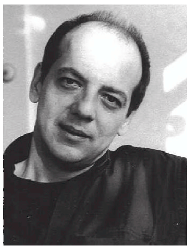
[44] L.A. Zadeh, "Fuzzy sets," *Inform. Contr.*, vol. 8, pp. 338-353, 1965.

[45] X.-J. Zeng and M.G. Singh, "Approximation accuracy analysis of fuzzy systems as function approximators," *IEEE Trans. Fuzzy Systems*, vol. 4, no. 1, pp. 44-63, 1996.

[46] X.-J. Zeng and M.G. Singh, "Knowledge bounded least squares method for the identification of fuzzy systems," *IEEE Trans. Systems, Man and Cybernetics – Part C*, vol. 33, no. 1, pp. 24- 32, 2003.



Vassilis G. Kaburlasos received the Diploma degree from the National Technical University of Athens, Greece, in 1986, and the M.Sc. and Ph.D. degrees from the University of Nevada, Reno, in 1989 and 1992, respectively, all in electrical engineering. He has been participant investigator or leader in 15 research projects funded both publicly and privately in the U.S.A. and in the European Union. He has authored more than 70 scientific papers in indexed journals, refereed conferences and books. Dr. Kaburlasos is currently a Professor with the Department of Industrial Informatics, Technological Educational Institution of Kavala, Greece. His research interests include intelligent machine modeling, design and industrial applications. Prof. Kaburlasos is a member of several professional, scientific, and honor societies around the world including Sigma Xi, Phi Kappa Phi, Tau Beta Pi, Eta Kappa Nu, and the Technical Chamber of Greece.



Athanasios Kehagias was born in Thessaloniki in 1961. He received the Dipl. Ing. Degree in electrical engineering from the School of Engineering of Aristotle University of Thessaloniki in 1984, the M.Sc. in Applied Mathematics from Lehigh University in 1986 and Ph. D. in applied mathematics from Brown University, Providence, RI, in 1989. Since November 1999, he has been with the Department of Mathematics, Physical and Computational Sciences, School of Engineering, Aristotle University of Thessaloniki, Thessaloniki, Greece. His research interests include applications of probability theory, algebra and fuzzy sets.

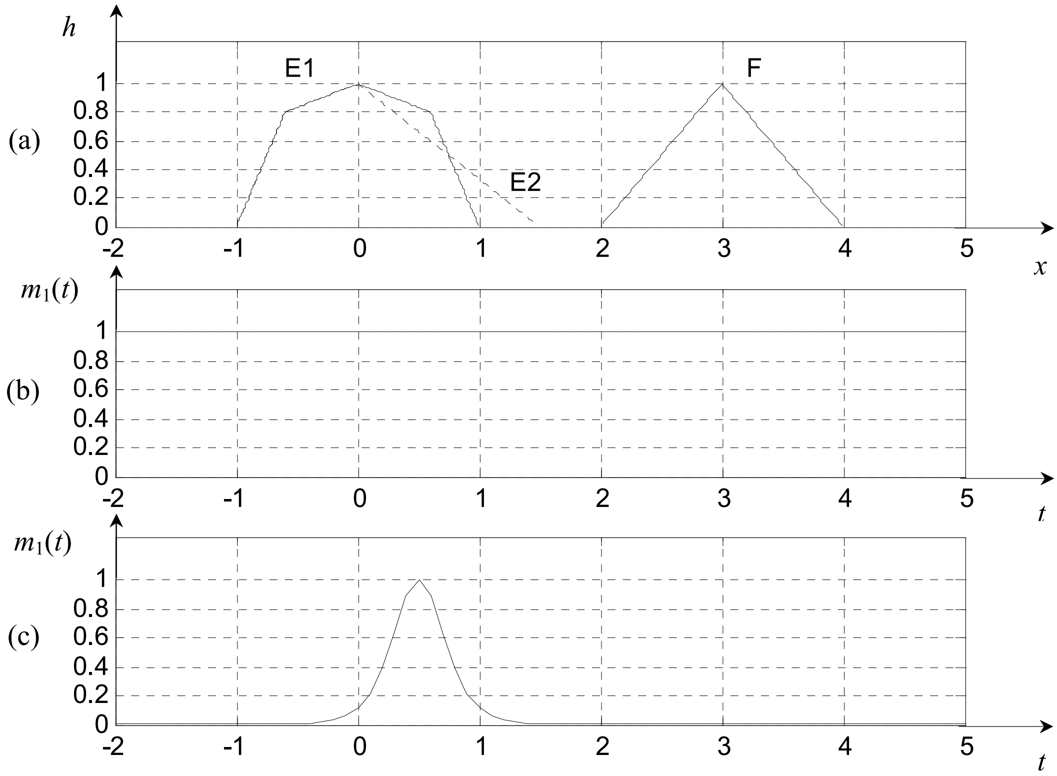
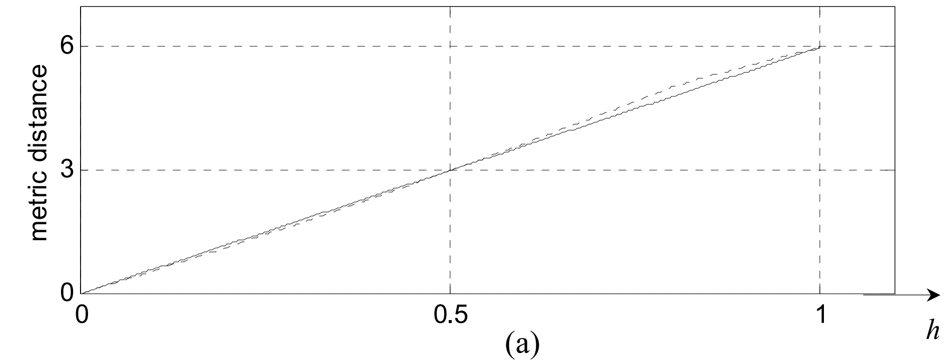


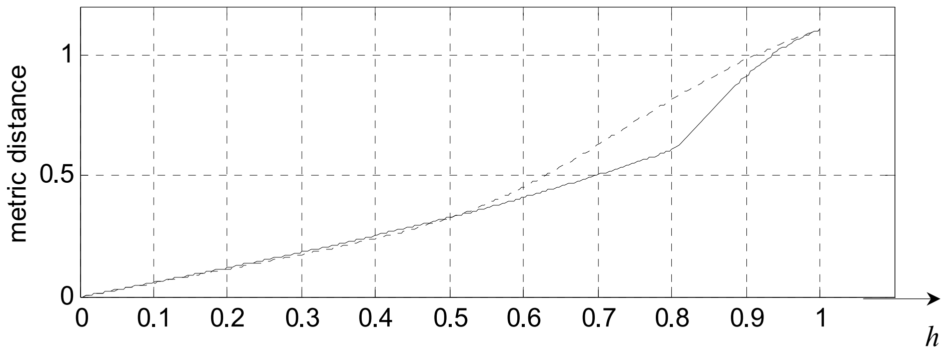
Fig. 1 (a) Three fuzzy numbers E1, E2, and F. The left sides of E1 (solid line) and E2 (dashed line) coincide.

(b) The mass function $m_h(t) = h$, for h=1.

(c) The mass function $m_h(t) = 4he^{-7(t-0.5)}/(1+e^{-7(t-0.5)})^2$, for h=1.



(a)



(b)

Fig. 2 The fuzzy numbers E1, E2, and F mentioned below are shown in Fig.1(a). The metric functions $\overline{d}_h$ ($E1_h$,F_h) and $\overline{d}_h$ ($E2_h$,F_h) are plotted here in solid and dashed lines, respectively. The area under a curve equals the corresponding distance between two fuzzy numbers.

(a) Using the mass function $m_h(t)= h$ (shown in Fig.1(b) for h=1), it turns out that d_K(E1,F) ≈ 3.0 > 2.9754 ≈ d_K(E2,F).

(b) Using the mass function $m_h(t)= 4he^{-7(t-0.5)}/(1+e^{-7(t-0.5)})^2$ (shown in Fig.1(c) for h=1), it turns out that d_K(E1,F)≈ 0.3587 < 0.3811 ≈ d_K(E2,F).

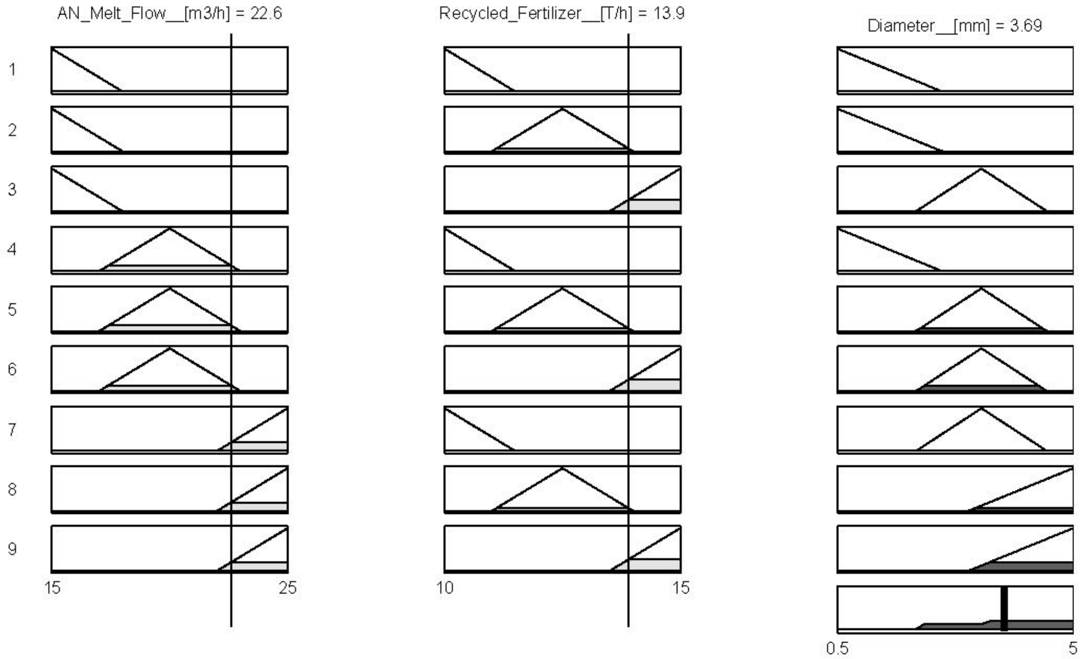


Fig. 3 A simple fuzzy inference system (FIS) with two linguistic input variables (i.e. AN Melt Flow [m^3/h], Recycled Fertilizer [T/h]) and one linguistic output variable (i.e. Diameter [mm]) were used to model industrial fertilizer granule size. Each linguistic variable above may obtain three different fuzzy set values with triangular membership functions. Nine fuzzy rules were used whose antecedent supports fully cover the input data domain. Using ‘min’ conjunction, ‘max’ disjunction, ‘min’ implication, ‘max’ aggregation, and ’centroid ‘ defuzzification the input data pair (22.6, 13.9) is mapped to number 3.69.

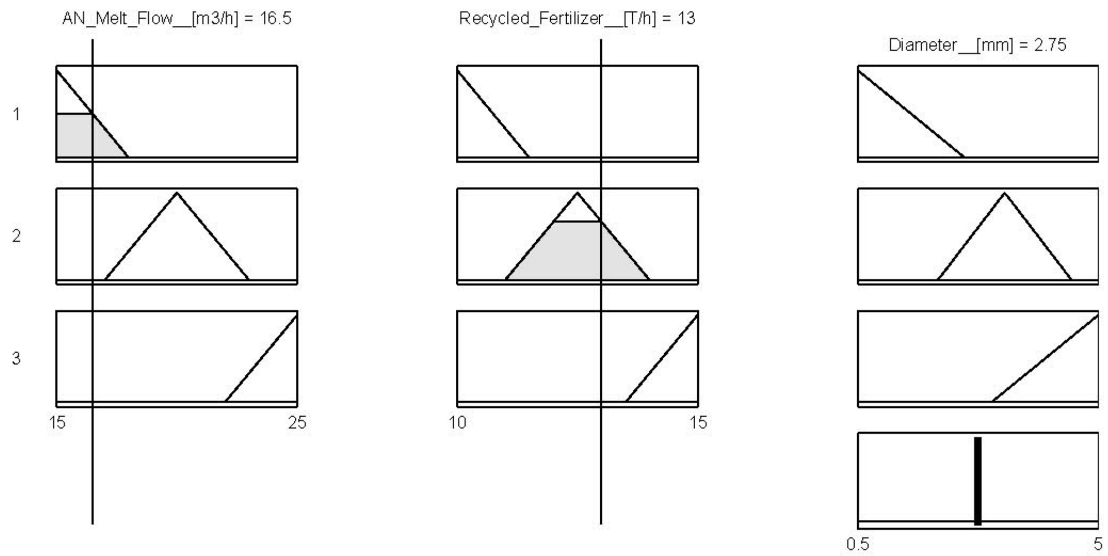


Fig. 4 The fuzzy inference system (FIS) above has stemmed from the FIS in Fig.3 by dropping six of its fuzzy rules, that is an order of magnitude reduction in the number of rules. There exist input data, including the pair (16.5, 13), which do not activate a fuzzy rule. The latter input data were conventionally mapped to the middle of the output data range [0.5, 5], that is the aforementioned data were mapped to number (0.5+5)/2 = 2.75.

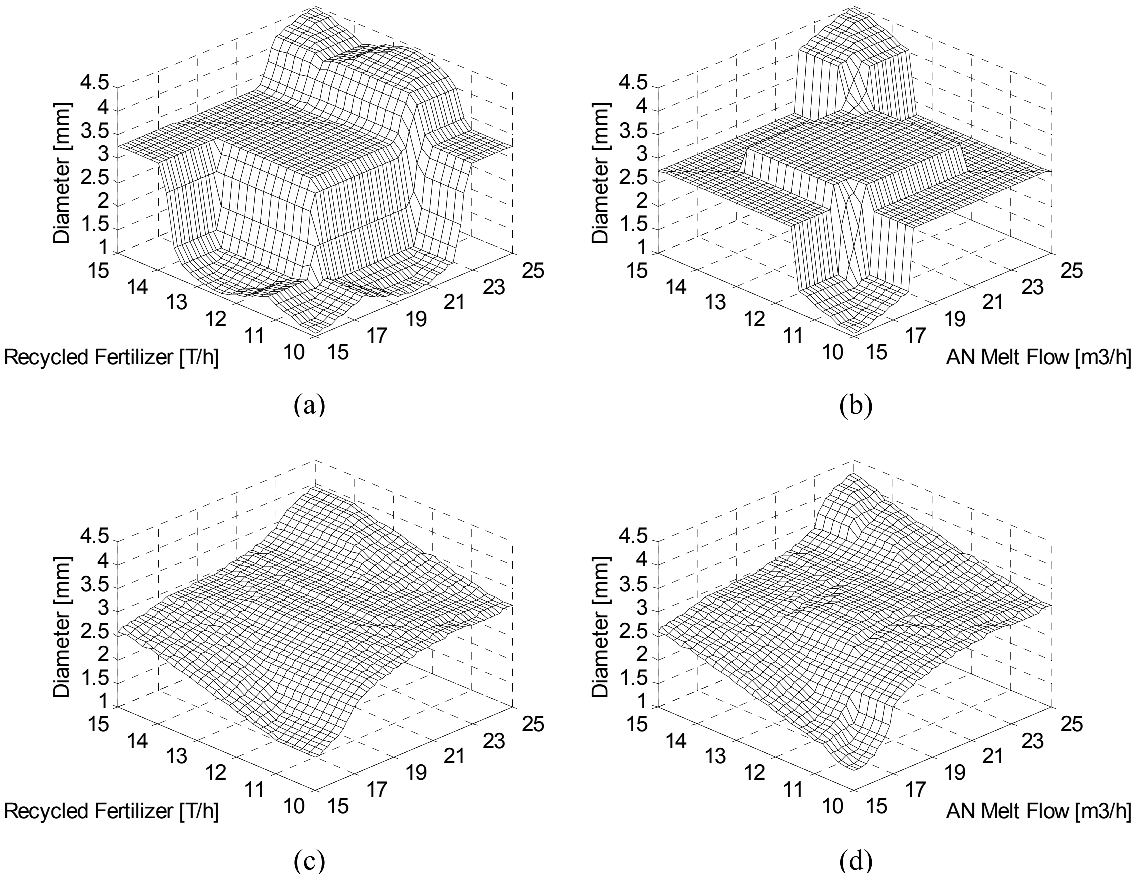


Fig.5 Two-dimensional output functions of different FISs.

(a) Output function of the FIS in Fig.3.

(b) Output function of the FIS in Fig.4.

(c) Output function produced by activating a fuzzy set F of a rule in Fig.4 using the (fuzzy membership) function $m_F(x) = 1/(1+d_1(F,x))$. Hence, a rule can be activated by input data outside its (rule) support.

(d) Output function produced by a thresholded combination of (1) standard fuzzy logic, and (2) fuzzy membership function $m_F(x) = 1/(1+d_1(F,x))$ of the FIS in Fig.4 for threshold T_f=0.35.

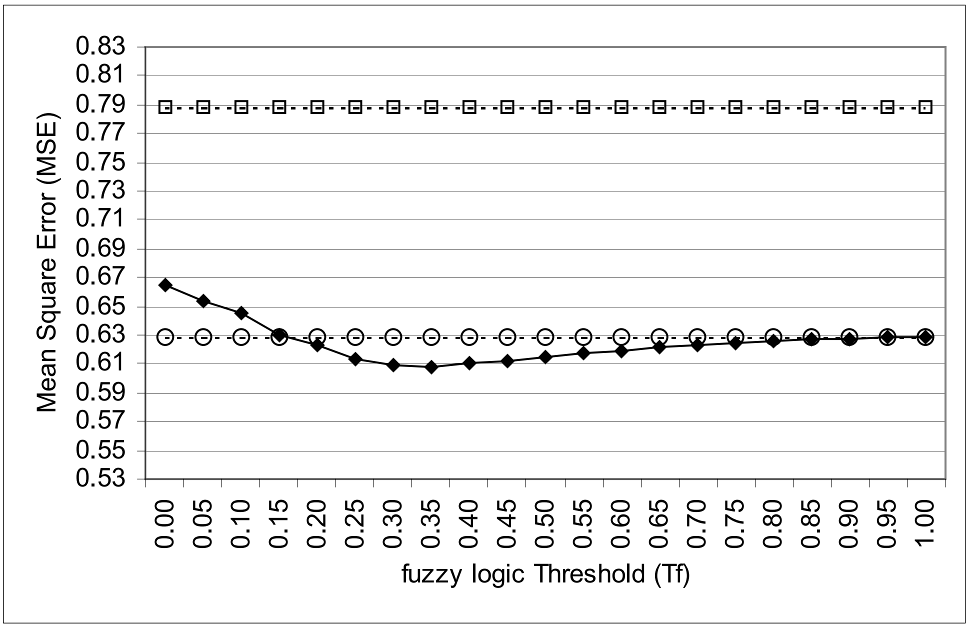


Fig. 6 The three curves above show the mean square error (MSE) of various approximations to the "prototype" surface in Fig.5(a). The top constant line, denoted by empty squares (□), at MSE= 0.7888 shows the MSE of the surface in Fig.5(b). The constant line, denoted by empty circles (○), at MSE= 0.6283 shows the MSE of the surface in Fig.5(c). The curve denoted by solid diamonds (♦) shows the MSE of a thresholded combination of standard fuzzy logic and fuzzy function $m_F(x)= 1/(1+d_1(F,x))$ versus threshold T_f; an optimal (global minimum) value of MSE= 0.6084 at $\widehat{T}_f = 0.35$ corresponds to the surface shown in Fig.5(d).

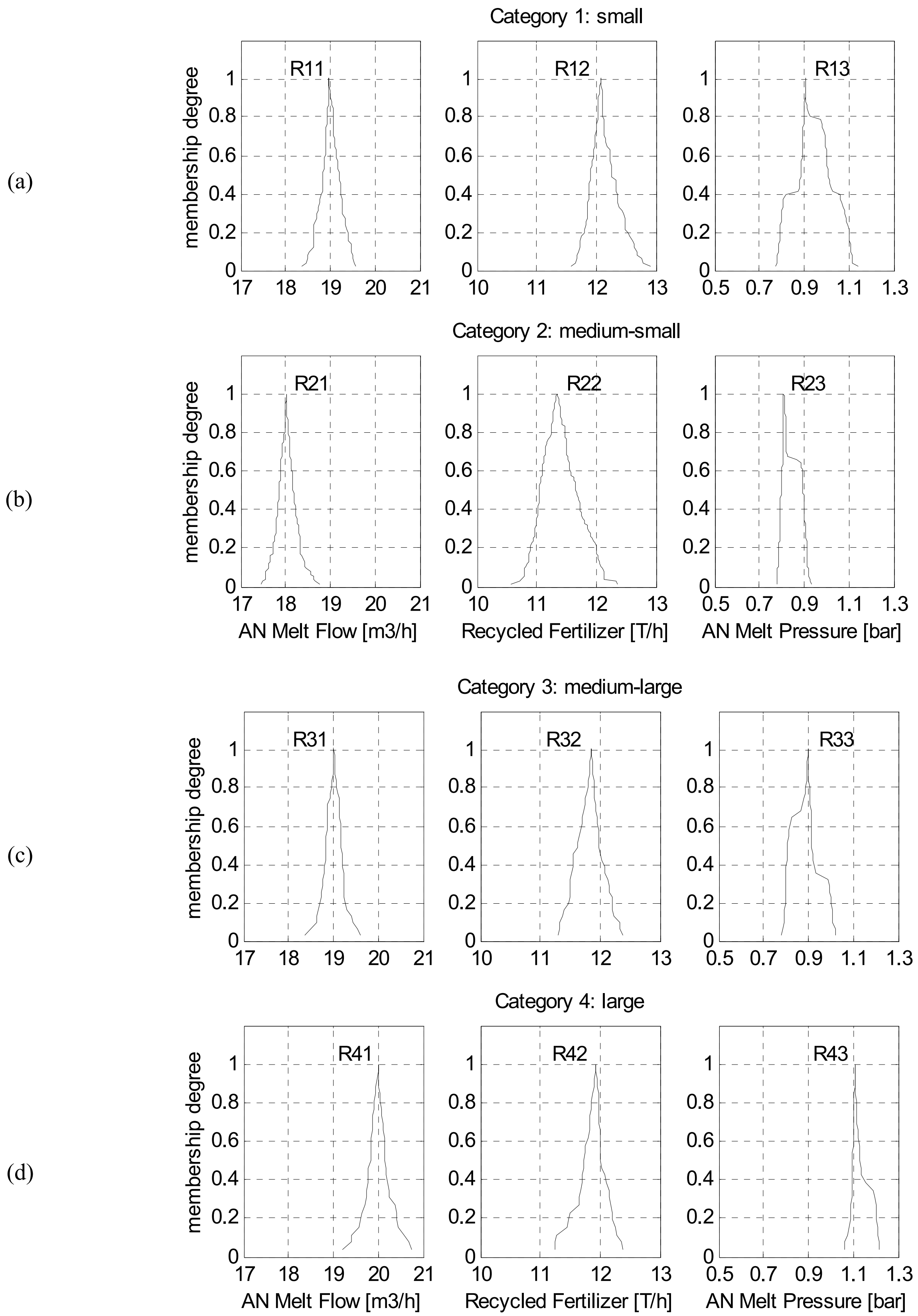


Fig. 7 The "IF parts" of four data-induced fuzzy rules are plotted; the corresponding "THEN part" of a fuzzy rule, that is a category label, is shown on the top of a plot.

(a)

(b)

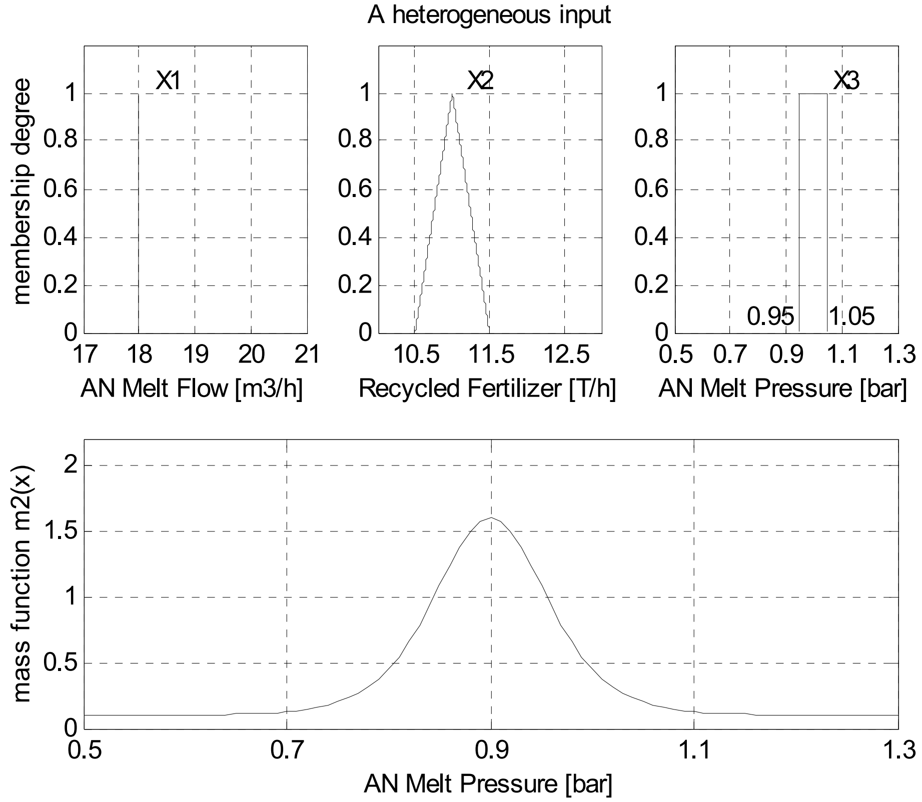


Fig. 8 (a) A FIS input X= (X1, X2, X3) can be *heterogeneous*. In this figure X includes a real number X1=18, a fuzzy set X2 with isosceles triangular membership function, and an interval component X3, respectively.

(b) A mass function can be used for distorting non-linearly the feature space. Using the Gaussian mass function $m2(x)= 0.1+(e^{-27(x-0.9)})/(1+e^{-27(x-0.9)})^2$ above it turns out that interval X3 is nearest to fuzzy number R43 (the latter implies category "large"), more specifically $d_K(X3,R43) = \min_{j\in\{1,2,3,4\}} d_K(X3,Rj3) \cong 0.0392$; whereas using the uniform mass function $m1(x)=1$ it turns out that interval X3 is nearest to fuzzy number R13 (the latter implies category "small"), in particular $d_K(X3,R13) = \min_{j\in\{1,2,3,4\}} d_K(X3,Rj3) \cong 0.0786$.

Table 1 Operating variables available for modeling the operation of the pan granulator in the Phosphoric Fertilizers Industry (PFI).

	Variable name	Unit
1	AN Melt Flow	m^3/h
2	Recycled Fertilizer	T/h
3	AN Melt Temperature	^oC
4	AN Melt Pressure	bar
5	Granulation Temperature	^oC
6	Pan Inclination	degrees
7	Pan Rotation Speed	Hz
8	Nozzle Vertical Distance	rings
9	Nozzle Distance from the pan	cm
10	Scraper Speed	Hz
11	Spraying Angle	lines
12	Coarse Screen Vibration	%
13	Fine Screen Vibration	%
14	$Mg(NO_3)_2$ Supply	%

Table 2 Metrics d_K(Ri3,Rj3), i,j∈{1,2,3,4} between four fuzzy numbers induced by classifier *grSOM* along the *AN Melt Pressure* data dimension. The uniform mass function $m_1(x)=1$ has been employed. Within parentheses is shown the corresponding fuzzy membership value m_{Ri3}(Rj3)= 1/(1+d_K(Ri3,Rj3)), i,j∈{1,2,3,4}.

	R13		R23		R33		R43	
R13	0	(1)	0.105	(0.905)	0.057	(0.946)	0.180	(0.847)
R23	0.105	(0.905)	0	(1)	0.048	(0.954)	0.285	(0.778)
R33	0.057	(0.946)	0.048	(0.954)	0	(1)	0.237	(0.808)
R43	0.180	(0.847)	0.285	(0.778)	0.237	(0.808)	0	(1)

Table 3 Metrics d_K(Ri3,Rj3), i,j∈{1,2,3,4} between four fuzzy numbers induced by classifier *grSOM* along the *AN Melt Pressure* data dimension. A Gaussian mass function $m_2(x)$ has been employed. Within parentheses is shown the corresponding fuzzy membership value m_{Ri3}(Rj3)= 1/(1+d_{K}(Ri3,Rj3)), i,j∈{1,2,3,4}.

	R13		R23		R33		R43	
R13	0	(1)	0.095	(0.913)	0.041	(0.960)	0.105	(0.905)
R23	0.095	(0.913)	0	(1)	0.054	(0.948)	0.200	(0.833)
R33	0.041	(0.960)	0.054	(0.948)	0	(1)	0.146	(0.872)
R43	0.105	(0.905)	0.200	(0.833)	0.146	(0.872)	0	(1)

Table 4 Performance of four classification methods in 10 different random partitions regarding real-world measurements from the Phosphoric Fertilizers Industry (PFI) at Nea Karvali, Greece.

Classification Method	Testing Data Classification Accuracy		no. grid units engaged /rules	
	average	stdv	average	stdv
Backpropagation	97.64	1.24	-	-
grSOM using $1/(1+d_1(.,.))$	97.05	1.38	8.10	1.44
grSOM using $d_1(.,.)$	96.76	1.66	8.30	1.25
Triangular FIS	95.29	1.51	7.50	1.08
KSOM	94.11	1.96	14.80	1.03

Table 5 The first three lines show metrics d_K(Rik,Xk)), i∈{1,2,3,4}, k∈{1,2,3} between the input X= (X1,X2,X3) entries shown in Fig.4(a) and the four fuzzy rule entries shown in Fig.3 using the uniform mass function $m_1(x)=1$. The last line shows the distances between interval X3 and the corresponding fuzzy rule entries using the Gaussian mass function $m_2(x)$. Within parentheses is shown the corresponding fuzzy membership value m_{Rij}(Xj)= 1/(1+d_K(Rij,Xj)), i∈{1,2,3,4}, j∈{1,2,3}.

	R11	R21	R31	R41
X1	0.962 (0.509)	0.196 (0.836)	0.982 (0.504)	1.968 (0.336)

	R12	R22	R32	R42
X2	1.083 (0.480)	0.389 (0.719)	0.825 (0.547)	0.887 (0.529)

	R13	R23	R33	R43
X3	0.078 (0.927)	0.166 (0.857)	0.118 (0.894)	0.119 (0.893)

	R13	R23	R33	R43
X3	0.068 (0.936)	0.159 (0.862)	0.110 (0.900)	0.039 (0.962)

Figure Captions

Fig. 1 (a) Three fuzzy numbers E1, E2, and F. The left sides of E1 (solid line) and E2 (dashed line) coincide.

(b) The mass function $m_h(t)= h$, for h=1.

(c) The mass function $m_h(t)= 4he^{-7(t-0.5)}/(1+e^{-7(t-0.5)})^2$, for h=1.

Fig. 2 The fuzzy numbers E1, E2, and F mentioned below are shown in Fig.1(a). The metric functions $\overline{d}_h(E1_h,F_h)$ and $\overline{d}_h(E2_h,F_h)$ are plotted here in solid and dashed lines, respectively. The area under a curve equals the corresponding distance between two fuzzy numbers.

(a) Using the mass function $m_h(t)= h$ (shown in Fig.1(b) for h=1), it turns out that $d_K(E1,F) \approx 3.0 > 2.9754 \approx d_K(E2,F)$.

(b) Using the mass function $m_h(t)= 4he^{-7(t-0.5)}/(1+e^{-7(t-0.5)})^2$ (shown in Fig.1(c) for h=1), it turns out that $d_K(E1,F) \approx 0.3587 < 0.3811 \approx d_K(E2,F)$.

Fig. 3 A simple fuzzy inference system (FIS) with two linguistic input variables (i.e. AN Melt Flow [m^3/h], Recycled Fertilizer [T/h]) and one linguistic output variable (i.e. Diameter [mm]) were used to model industrial fertilizer granule size. Each linguistic variable above may obtain three different fuzzy set values with triangular membership functions. Nine fuzzy rules were used whose antecedent supports fully cover the input data domain. Using 'min' conjunction, 'max' disjunction, 'min' implication, 'max' aggregation, and 'centroid ' defuzzification the input data pair (22.6, 13.9) is mapped to number 3.69.

Fig. 4 The fuzzy inference system (FIS) above has stemmed from the FIS in Fig.3 by dropping six of its fuzzy rules, that is an order of magnitude reduction in the number of rules. There exist input data, including the pair (16.5, 13), which do not activate a fuzzy rule. The latter input data were conventionally mapped to the middle of the output data range [0.5, 5], that is the aforementioned data were mapped to number (0.5+5)/2 = 2.75.

Fig.5 Two-dimensional output functions of different FISs.

(a) Output function of the FIS in Fig.3.

(b) Output function of the FIS in Fig.4.

(c) Output function produced by activating a fuzzy set F of a rule in Fig.4 using the (fuzzy membership) function $m_F(x)$= $1/(1+d_1(F,x))$. Hence, a rule can be activated by input data outside its (rule) support.

(d) Output function produced by a thresholded combination of (1) standard fuzzy logic, and (2) fuzzy membership function $m_F(x)$= $1/(1+d_1(F,x))$ of the FIS in Fig.4 for threshold T_f=0.35.

Fig. 6 The three curves above show the mean square error (MSE) of various approximations to the "prototype" surface in Fig.5(a). The top constant line, denoted by empty squares (□), at MSE= 0.7888 shows the MSE of the surface in Fig.5(b). The constant line, denoted by empty circles (○), at MSE= 0.6283 shows the MSE of the surface in Fig.5(c). The curve denoted by solid diamonds (♦) shows the MSE of a thresholded combination of standard fuzzy logic and fuzzy function $m_F(x)$= $1/(1+d_1(F,x))$ versus threshold T_f; an optimal (global minimum) value of MSE= 0.6084 at $\widehat{T}_f = 0.35$ corresponds to the surface shown in Fig.5(d).

Fig. 7 The "IF parts" of four data-induced fuzzy rules are plotted; the corresponding "THEN part" of a fuzzy rule, that is a category label, is shown on the top of a plot.

Fig. 8 (a) A FIS input X= (X1, X2, X3) can be *heterogeneous*. In this figure X includes a real number X1=18, a fuzzy set X2 with isosceles triangular membership function, and an interval component X3, respectively.

(b) A mass function can be used for distorting non-linearly the feature space. Using the Gaussian mass function $m2(x)= 0.1+(e^{-27(x-0.9)})/(1+e^{-27(x-0.9)})^2$ above it turns out that interval X3 is nearest to fuzzy number R43 (the latter implies category "large"), more specifically $d_K(X3,R43) = \min_{j\in\{1,2,3,4\}} d_K(X3,Rj3) \cong 0.0392$; whereas using the uniform mass function $m1(x)=1$ it turns out that interval X3 is nearest to fuzzy number R13 (the latter implies category "small"), in particular $d_K(X3,R13) = \min_{j\in\{1,2,3,4\}} d_K(X3,Rj3) \cong 0.0786$.

Table Captions

Table 1
Operating variables available for modeling the operation of the pan granulator in the Phosphoric Fertilizers Industry (PFI).

Table 2
Metrics d_K(Ri3,Rj3), i,j∈{1,2,3,4} between four fuzzy numbers induced by classifier *grSOM* along the *AN Melt Pressure* data dimension. The uniform mass function $m_1(x)=1$ has been employed. Within parentheses is shown the corresponding fuzzy membership value m_{Ri3}(Rj3)= 1/(1+d_K(Ri3,Rj3)), i,j∈{1,2,3,4}.

Table 3
Metrics d_K(Ri3,Rj3), i,j∈{1,2,3,4} between four fuzzy numbers induced by classifier *grSOM* along the *AN Melt Pressure* data dimension. A Gaussian mass function $m_2(x)$ has been employed. Within parentheses is shown the corresponding fuzzy membership value m_{Ri3}(Rj3)= 1/(1+d_K(Ri3,Rj3)), i,j∈{1,2,3,4}.

Table 4
Performance of four classification methods in 10 different random partitions regarding real-world measurements from the Phosphoric Fertilizers Industry (PFI) at Nea Karvali, Greece.

Table 5
The first three lines show metrics d_K(Rik,Xk)), i∈{1,2,3,4}, k∈{1,2,3} between the input X=(X1,X2,X3) entries shown in Fig.4(a) and the four fuzzy rule entries shown in Fig.3 using the uniform mass function $m_1(x)=1$. The last line shows the distances between interval X3 and the corresponding fuzzy rule entries using the Gaussian mass function $m_2(x)$. Within parentheses is shown the corresponding fuzzy membership value m_{Rij}(Xj)= 1/(1+d_K(Rij,Xj)), i∈{1,2,3,4}, j∈{1,2,3}.