Fuzzy Time Series Forecasting

- Developing a new forecasting model based on high order fuzzy time series

AAUE

November 2009

Semester: CIS 4

Author: Jens Rúni Poulsen

Fuzzy Time Series Forecasting

- Developing a new forecasting model based on high order fuzzy time series

Abstract

Numerous Fuzzy Time Series (FTS) models have been proposed in scientific literature during the past decades or so. Among the most accurate FTS models found in literature are the high order models. However, three fundamental issues need to be resolved with regards to the high order models. First, current prediction methods have not been able to provide satisfactory accuracy rates for defuzzified outputs (forecasts). Second, data becomes underutilized as the order increases. Third, forecast accuracy is sensitive to selected interval partitions.

To cope with these issues, a new high order FTS model is proposed in this thesis. The proposed model utilizes aggregation and particle swarm optimization (PSO) to reduce the mismatch between forecasts and actuals. Comparative experiments confirm the proposed model's ability to provide higher accuracy rates than the current results reported in the literature. Moreover, the utilization of aggregation and PSO, to individually tune forecast rules, ensures consistency between defuzzified outputs and actual outputs, regardless of selected interval partitions. As a consequence of employing these techniques, data utilization is improved by: (1) minimizing the loss of forecast rules; (2) minimizing the number of pattern combinations to be matched with future time series data.

Finally, a fuzzification algorithm, based on the trapezoid fuzzification approach, has been developed as a byproduct. The proposed algorithm objectively partitions the universe of discourse into intervals without requiring any user defined parameters.

Author: Jens Rúni Poulsen

Educational Institute: Aalborg University Esbjerg (AAUE)

Semester: CIS 4, Nov. 2009

Supervisor: Daniel Ortiz Arroyo

Jens Rúni Poulsen

Table of Contents

1 Introduction

This research is carried out as part of the CIS 4 semester at AAUE and is concerned with the development of a new forecasting model based on high order fuzzy time series (FTS). Numerous FTS models have been proposed during the past decades or so [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15, 16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33]. The high order FTS models [15,17,32,31, 30] are the most accurate models found in literature vis-à-vis related modalities. Despite this, current publications have not been able to provide satisfactory results for defuzzified outputs (forecasts). Another problem particularly associated to high order models is the underutilization of data that occurs as a result of increasing the model's order. Lastly, current prediction models are sensitive to selected interval partitions. To cope with the shortcomings mentioned here, this project sets out to develop a forecast model based on FTS which: (1) provides higher forecasting accuracy rates than its high order counterparts; (2) improves data utilization; and (3) remains unaffected by selected interval partitions. A secondary objective is to design a fuzzification algorithm based on the trapezoid fuzzification approach[4] which automatically generates interval partitions based on some objective measure.

The thesis is organized as follows. Section 2 deals with relevant theoretical aspects such as fuzzy sets, fuzzy numbers, defuzzification, fuzzy relations, fuzzy aggregation operators, particle swarm optimization (PSO) and basic concepts of FTS. Section 4 provides an overview of related work. The proposed FTS model is presented and comparatively evaluated in section 5 and 6, respectively. Finally, concluding remarks are provided in section 7.

2 Theoretical Foundation

This section reviews various theoretical concepts relevant in the context of this study. The main topics covered here are fuzzy sets, fuzzy numbers, defuzzification, fuzzy relations, fuzzy aggregation operators, PSO and basic FTS concepts.

2.1 Conventional Sets vs Fuzzy Sets

Conventional set theory rests on the notion of a crisp boundary between which elements are members and non-members of a particular set. Thus if someone asks the question of whether an element is in a set, the answer is always yes or no for all elements. For example, if we consider the set of tall people, all persons are either tall or not. There is nothing in between of being tall and not being tall. Basically, conventional sets can be described in two ways; explicitly in a list or implicitly with a predicate. An example of the former could be the finite set $A = \{0, 1, 2, 3\}$. An example of the latter could be the set of all integers larger than 10 which is an infinite set. Either way, we can always answer yes or no to which elements are in a set or not.

The drawback of conventional sets is that many concepts encountered in the real world cannot always be described exclusively by their membership and non-membership in sets. As an example, let us again consider the set of tall people. If we ask a group of people when exactly a person is tall and when exactly he/she is not, we are likely to get a set of deviating answers. This is because there is no crisp boundary between being tall and not being tall (at least not one that is intuitively clear). Stated otherwise, the property of being tall is inherently undecidable. Generally all persons taller than 200 cm satisfy the property of being tall and everyone shorter than 150 cm do not. But what about the membership status of those with a height that lies in between these two extremes? Well some of these people may still be considered tall and some not, the point is however that the further we move down the scale from 200 cm to 150 cm, the answer will not remain as clear-cut for all cases. At some point we will reach a state of ambiguity, that is a state where we cannot explicitly say either yes or no.

Fuzzy set theory [34,35,36] expands the notion of purely crisp sets by assigning membership degrees to set elements so the transition from membership to non-membership is gradual rather than abrupt. Normally, the membership degree of a set element is a real number between 0 and 1. The closer the membership degree is to 1, the more an element belongs to a given set. A membership degree of 0 means that an element is clearly not a member of a particular set. Elements with a membership degree between 0 and 1 are more or less members of a particular set. An example is the property of being tall, where a person may be more or less tall. In conventional set theory, more or less membership is not allowed.

2.2 The Universe of Discourse

All elements in a set are taken from a *universe of discourse* or *universe set* that contains all the elements that can be taken into consideration when the set is formed. In reality there is no such thing as a set or a fuzzy set because all sets are subsets of some universe set, even though the term 'set' is predominantly used. In the fuzzy case, each element in the universe set is a member of the fuzzy set to some degree, even zero. The set of elements that have a non-zero membership is referred to as the *support*. We will use the notation *U* for the *universe set.*

2.3 Fuzzy Subsets

A fuzzy subset *A* in *U* is characterized by a *membership function* (characteristic function) that maps each element in *A* with a real number in the unit interval. Formally, this can be expressed as $\mu_A : U \rightarrow [0,1]$ where the value $\mu_A(x)$ is called the degree of membership of the element x in the fuzzy set *A*. The membership function declares which elements of *U* are members of *A* and which are not. The principle of fuzzifying crisp sets in this manner is called the *extension principle*.

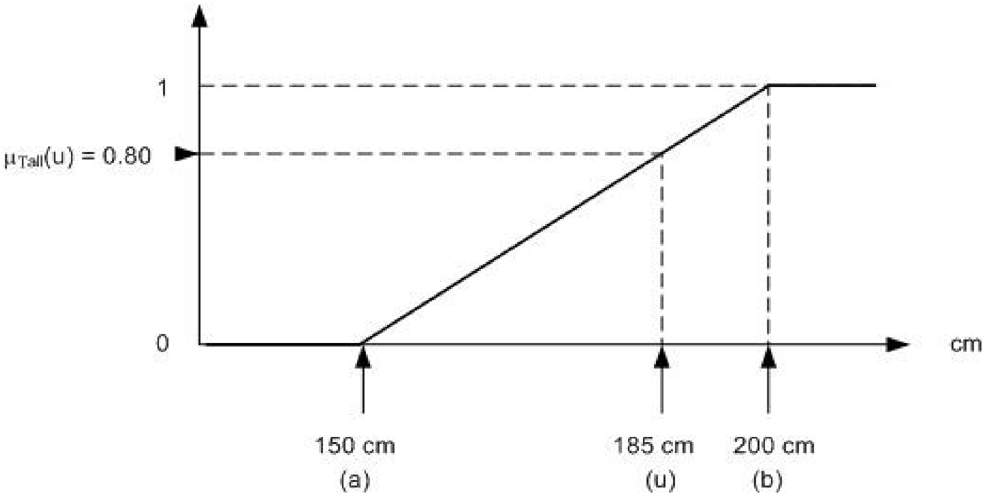

Figure 1. An example of a membership function for the fuzzy set Tall.

A classical example of a fuzzy set is the subset of person's heights considered tall, see figure 1. We refer to this set as Tall. The figure shows that if a person's height is less than or equal to 150 cm (point *a*), the degree of membership in the fuzzy set Tall is equal to zero. This means that all persons whose height is less than or equal to 150 cm are completely excluded from this set. If a person's height is larger than or equal to 200 cm (point *b*), the property of being tall is fully satisfied. Hence the membership degree in Tall is equal to 1. When the height is larger than 150 cm and less than 200 cm, the property of being tall is more or less satisfied. For example, a person who is 185 cm (point *u*) is tall to a degree of 0.8. Mathematically the above membership function (aka characteristic function) can be defined as

$$\mu_{Tall}(x)=\begin{cases}0, & x<a\\ \dfrac{x-a}{b-a}, & a\le x\le b\\ 1, & b<x.\end{cases} \tag{1}$$

In the case where a fuzzy set A is a conventional (crisp) set, the corresponding membership function can be reduced to

$$\mu_A(x)=\begin{cases}1, & x\in A\\ 0, & x\notin A.\end{cases} \tag{2}$$

The above function has only two outputs, 0 or 1. Whenever $\mu_A(x)=1, x$ is a member of *A,* if $\mu_A(x)=0, x$ is declared a non-member of A. Other examples of membership functions commonly used in literature are depicted in figure 2.

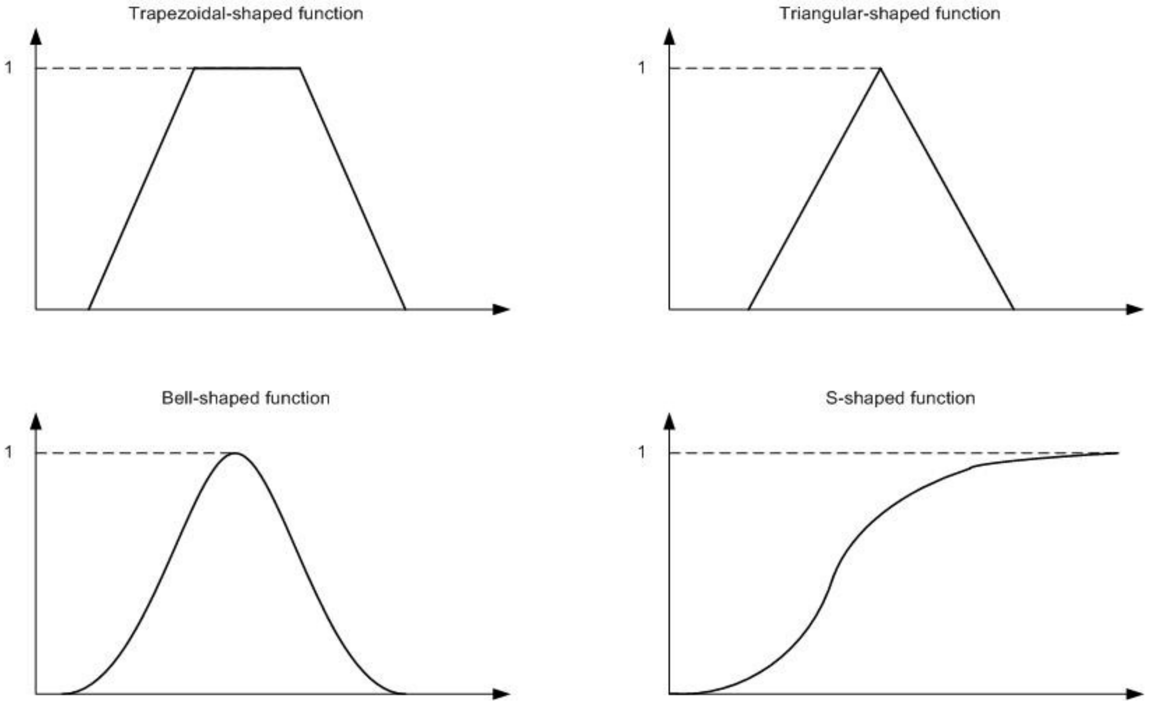

Figure 2. Various shapes of commonly used membership functions.

It needs to be noted that fuzzy membership functions are not necessarily symmetric in nature even though this is not indicated in the figure. Depending on the application, the shape of a membership function may or may not be symmetrical. There is not yet any universal rule or criterion for selecting a membership function for a particular type of fuzzy subset. Rather the choice depends on several factors, for example, the users scientific experience and knowledge or actual needs for the application in question. Whatever membership function is chosen for the problem at hand, will more or less be based on the users subjective measures. However, just as in probability theory and statics, for example, one may assume that a particular function describes some property, like "it is assumed that the membership function in figure 1 describes the property of being tall".

2.3.1 Alpha Cut

An important property of fuzzy sets is the alpha cut (α-cut). Given a fuzzy set A defined on the universal set U and any number in the unit interval, $\alpha \in [0,1]$, the (weak) α-cut, $A_{\bar{\alpha}}$, and the strong α-cut, A_{α}, are the crisp sets which satisfy

$$A_{\bar{\alpha}} = \{x \in A | \mu_A(x) \geq \alpha\}$$
$$A_{\alpha} = \{x \in A | \mu_A(x) > \alpha\}. \tag{3}$$

Less formally the α-cut of a fuzzy set A is the crisp set $A_{\bar{\alpha}}$ that contains all the elements of the universal set U whose membership grades in A are grater than or equal to a given α (or strictly grater than α, if we refer to the strong variant A_{α}). Recall that the support of a fuzzy set A within the universal set U is the crisp set that contains all the elements of U that have non-zero membership grades in A. Hence the support of A is exactly the same as the strong α-cut of A for $\alpha = 0$.

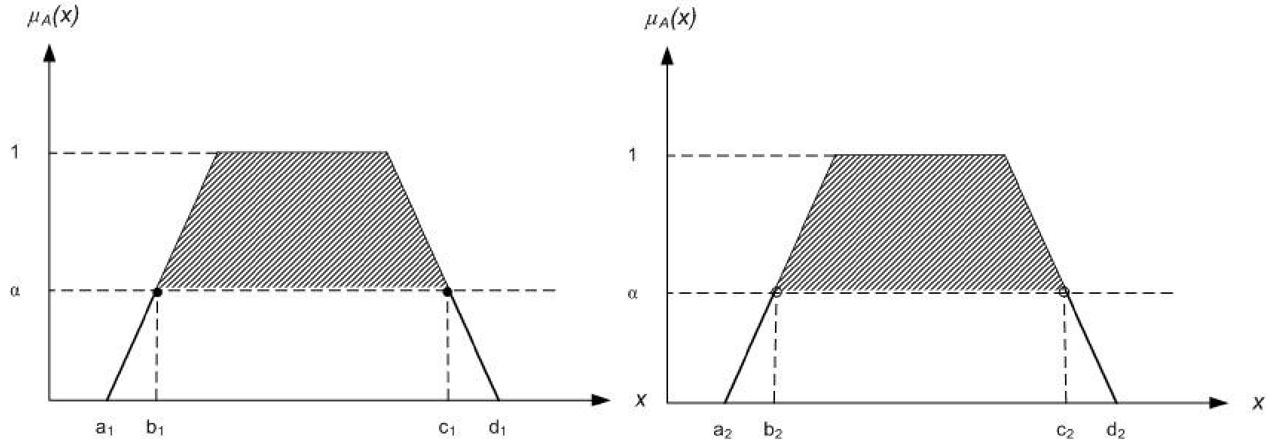

Figure 3. Weak and strong alpha-cut, respectively.

A fuzzy subset A in ℝ is convex iff the sets defined by

$$A_{\bar{\alpha}} = \{x \in A | \mu_A(x) \geq \alpha\} \tag{4}$$

are convex for all α-level sets in the interval [0,1]. Another more direct definition of convexity is the following: For all pairs $x_1, x_2 \in A$ and any $\lambda \in [0,1]$, A is convex iff

$$\mu_A[\lambda x_1 + (1-\lambda) x_2] \geq \min[\mu_A(x_1),\ \mu_A(x_2)], \tag{5}$$

where min denotes the minimum operator.

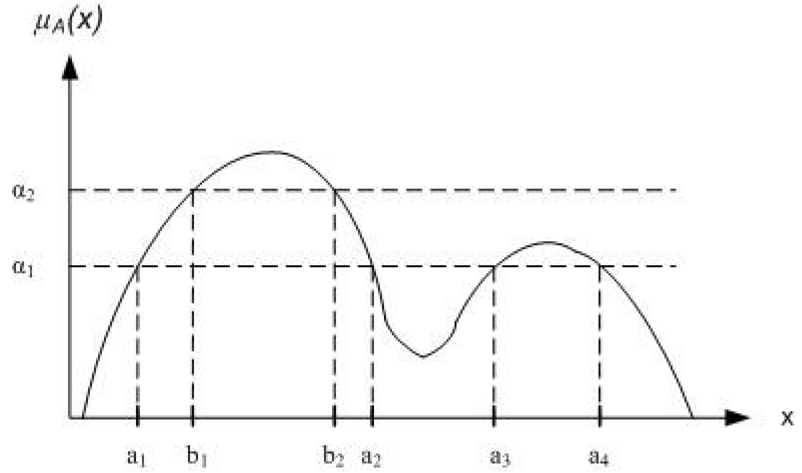

Figure 4. A convex and a non-convex fuzzy subset.

2.4 Representations of Fuzzy Sets

Basically a fuzzy set can be viewed as a collection of ordered pairs

$$A=\{(x_1,\mu(x_1)),(x_2,\mu(x_2)),\ldots,(x_n,\mu(x_n))\}, \tag{6}$$

where element x is a member of A and $\mu(x)$ denotes its degree of membership in A. A single pair $(x,\mu(x))$ is called a fuzzy singleton. Hence a complete fuzzy set can be viewed as the union of its constituent singletons. If a fuzzy set is finite and discrete an often used notation is

$$A=u_1/x_1+u_2/x_2+\ldots+u_n/x_n. \tag{7}$$

It is important to note that neither the slash or the plus sign represent any kind of algebraic operation. The slash links the elements of the support with their grades of membership in *A,* whereas the plus sign indicates that the listed pairs of elements and membership grades collectively form the definition of the set A.

2.5 Operations on Fuzzy Sets

In classical set theory there are three basic operations that can be performed on crisp sets: the complement, intersection and union. These operators also exists in fuzzy set theory in addition to a range of other operators. The standard fuzzy set operators complement, intersection and union are defined by the equations

$$\begin{aligned}&\overline{\mu_A}(x)=1-\mu_A(x)\\&(\mu_A\cap\mu_B)(x)=\min[\mu_A(x),\mu_B(x)]\\&(\mu_A\cup\mu_B)(x)=\max[\mu_A(x),\mu_B(x)].\end{aligned} \tag{8}$$

where *A* and *B* are fuzzy subsets of the universal interval *U.* The two operators, min and max*,* respectively denote the minimum and maximum operators. As can be seen from the respective equation, the min and the max operations of two fuzzy sets μ_A and μ_B is an element-by-element comparison between corresponding elements in the respective sets. In the complement case, each membership value of μ_A is substracted from 1.

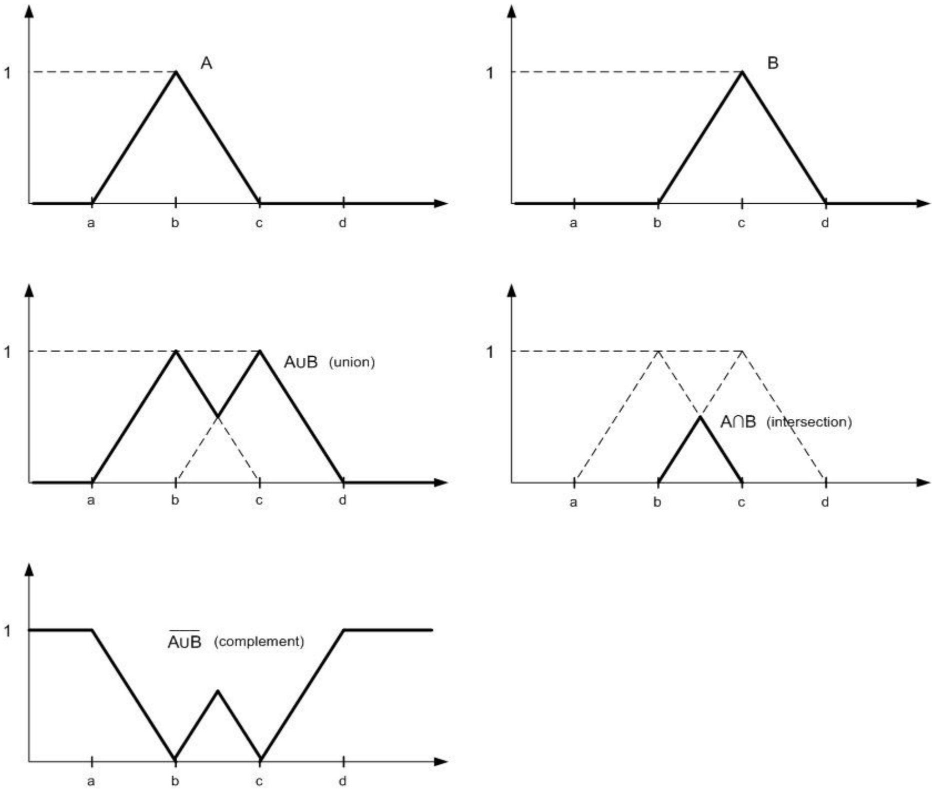

Figure 5. The basic set operations.

As mentioned, there are also a range of other operators in addition to the standard fuzzy set operators. These operators can be categorized as follows: t-norms, averaging operators and t-conorms. An in-debt discussion about these operators can be found in section 2.11.

2.6 Fuzzy Numbers

In this section we will briefly review some frequently used classes of fuzzy numbers.

Definition 1: Fuzzy Number

A fuzzy number A is described as any fuzzy subset of the real line ℝ with membership function μ_A which possesses the following properties[37]:

(a) μ_A is a continuous mapping from ℝ to the closed interval $[0,\omega], 0 \leqslant \omega \leqslant 1$;

(b) $\mu_A(x)=0$, for all $x \in [-\infty, a]$;

(c) μ_A is strictly increasing on $[a,b]$;

(d) $\mu_A(x)=\omega$, for all $x \in [b,c]$, where ω is a constant and $0 < w \leq 1$;

(e) μ_A is strictly decreasing on $[c,d]$;

(f) $\mu_A(x)=0$, for all $x\in[d,\infty]$;

where a, b, c and d are real numbers. Unless elsewhere specified, it is assumed that A is convex and bounded; i.e. $-\infty<a,d<\infty$. If $\omega=1$ in (d), A is a normal fuzzy number, and if $0<w<1$ in (d), A is a non-normal fuzzy number. For convenience, the fuzzy number in definition 1 can be denoted by $A=(a,b,c,d;\omega)$. The image (opposite) of A can be given by $-A=(-d,-c,-b,-a;w)$. Property (a) can also be written as $A:\mathbb{R}\rightarrow[0,1]$ The membership function of μ_A can be expressed as

$$\mu_A(x)=\begin{cases}\mu_A^L(x), & a\le x\le b\\ \omega, & b\le x\le c\\ \mu_A^R(x), & c\le x\le d\\ 0, & otherwise,\end{cases} \tag{9}$$

where $\mu_A^L:[a,b]\rightarrow[0,\omega]$ and $\mu_A^R:[c,d]\rightarrow[0,\omega]$.

Definition 2: Triangular Fuzzy Number

A triangular fuzzy number A is a fuzzy number with a piecewise linear membership function μ_A defined by

$$\mu_A=\begin{cases}\dfrac{x-a_1}{a_2-a_1}, & a_1\le x\le a_{2,}\\ \dfrac{a_3-x}{a_3-a_2}, & a_2\le x\le a_{3,}\\ 0, & otherwise,\end{cases} \tag{10}$$

which can be denoted as a triplet $(a_{1,}a_{2,}a_3)$.

Definition 3: Trapezoidal Fuzzy Number

A trapezoidal fuzzy number A is a fuzzy number with a membership function μ_A denoted by

$$\mu_A=\begin{cases}\dfrac{x-a_1}{a_2-a_1}, & a_1\le x\le a_2\\ 1, & a_2\le x\le a_3\\ \dfrac{a_4-x}{a_4-a_3}, & a_3\le x\le a_4\\ 0, & otherwise,\end{cases} \tag{11}$$

which can be denoted as a quartet $(a_{1,} a_{2,} a_{3,} a_4)$.

2.7 Ranking

Ranking[38,39] is the task of comparing fuzzy subsets (i.e. numbers) and arranging them in a certain order. Especially in decision making situations, appropriate methods are needed to compare and evaluate different alternatives, i.e. which fuzzy number precedes the other in a given situation. When dealing with strictly numerical values, this process is quite simple, since the order can be naturally determined. Fuzzy numbers, on the other hand, cannot be ordered in the same manner because the same natural order does not exist in fuzzy numbers. That is, we cannot explicitly say that a fuzzy number A is larger than another fuzzy number B as in the numerical case. Whether A is larger, smaller or equal to B is a matter interpretation. A simple method for ordering fuzzy subsets consists in the definition of a ranking function *F*, mapping each fuzzy set to the real numbers $\mathbb{R}$, where a natural order exists. Suppose $S=\{\mu_{A1}, \mu_{A2}, \ldots, \mu_{An}\}$ is a set of n convex fuzzy numbers, and the ranking function F is a mapping from S to the real numbers $\mathbb{R}$, i.e. $F: S \rightarrow \mathbb{R}$. Then for any distinct pair of fuzzy numbers $\mu_{Ai}, \mu_{Aj} \in S$, the ranking function can be defined as

$$\begin{aligned} &if\, F(\mu_{Ai}) < F(\mu_{Aj});then\ \mu_{Ai} < \mu_{Aj} \\ &if\, F(\mu_{Ai}) = F(\mu_{Aj});then\ \mu_{Ai} = \mu_{Aj} \\ &if\, F(\mu_{Ai}) > F(\mu_{Aj});then\ \mu_{Ai} > \mu_{Aj}. \end{aligned} \tag{12}$$

This implies, for example, that if $F(\mu_{Ai}) > F(\mu_{Aj})$, the fuzzy number μ_{Ai} is numerically greater than the fuzzy number μ_{Aj}. The higher μ_{Ai} is, the larger $F(\mu_{Aj})$ is. A useful technique for ranking normal fuzzy numbers that are convex, such as triangular- and trapezoid fuzzy numbers, is the centroid, defined by

$$F(\mu_A) = \frac{\int x \mu_A(x)\, dx}{\int \mu_A(x)\, dx}, \tag{13}$$

where $F(\mu_A)$ represents the centroid of the fuzzy set μ_A.

2.8 Linguistic Variables

Fuzzy numbers are frequently used to represent quantitative variables, normally referred to as *linguistic variables* [35,36]. Linguistic variables take words or sentences as values, as opposed to an algebraic variable which takes numbers as values. All values are taken from a *term set* that contains the set of acceptable values/concepts. Each value/concept in the term set is represented by a fuzzy number which is defined over some universe interval, also called a *base variable*. In short this relationship can be expressed as follows: linguistic variable → fuzzy variable → base variable. For example, let *v* be a linguistic variable denoting a person's height. The values of *v*, which are fuzzy variables, could be defined by the term set *T* = {*very short, short, medium tall, tall, very tall*} and the associated base variable could span the interval from 100 to 220 cm.

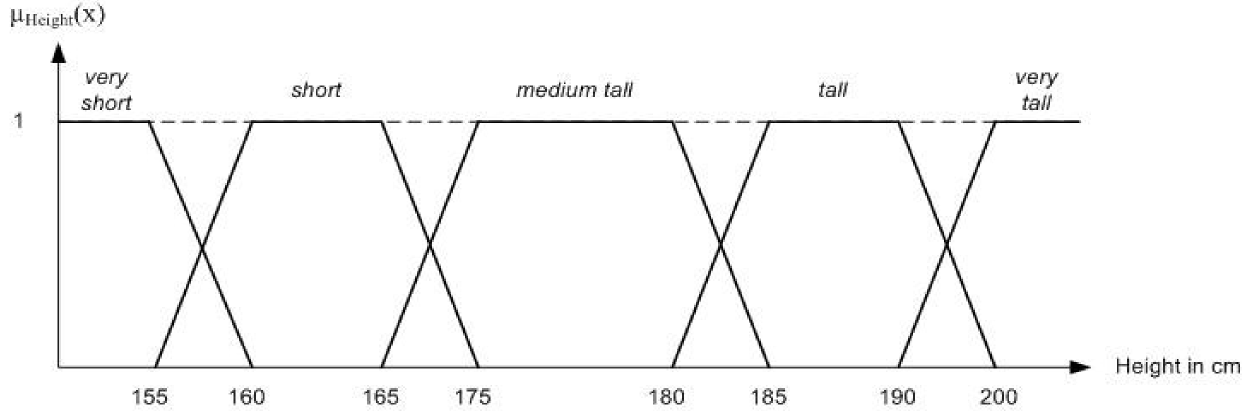

Figure 6. A linguistic representation of the fuzzy set Height.

An example of a linguistic variable is shown in figure 6. Its name is Height and it expresses the height of a person in a given context by five linguistic terms - very short, short, medium tall, tall and very tall. Each of the basic linguistic terms is assigned one of five trapezoidal fuzzy numbers which define the range of the base variable.

2.9 Defuzzification

Two central concepts in fuzzy set theory are fuzzification and its counterpart defuzzification. Fuzzification is the process of converting crisp values into fuzzy values by identifying possible uncertainties or variations in the crisp values. This conversion is represented by membership functions. There are various ways this fuzzification process can be carried out, like intuition, genetic algorithms [34] or neural networks [34]. Defuzzification is the process of converting fuzzy values into crisp ones. In the following we will describe some defuzzification methods found in literature [34,40].

2.9.1 Max-membership principle

In this method, the defuzzified value, *Z*, equals the *x*-value with the highest membership degree. It is given by the expression

$$\mu_A(x_*) \geq \mu_A(x)\, for\, all\; x \in A, \tag{14}$$

where x_* is the value with the highest degree of membership in the fuzzy set *A*. If we consider the following set $A = 0.3/10 + 0.45/12 + 0.6/15 + 0.9/17$. Then $\max(A) = 17$.

2.9.2 Centroid method

This is the most widely used method. It is also referred to as the center of gravity or center of area method. It can be defined by equation 13 when μ_A is continuous. For the discrete case in which μ_A is defined on a finite universal set $\{x_{1,} x_2, \ldots, x_n\}$, the equation is

$$Z = \frac{\sum_{i=1}^{n} \mu_A(x_i)\, x_i}{\sum_{i=1}^{n} \mu_A(x_i)}. \tag{15}$$

Using the example from before, we get

$$Z = \frac{(0.3 \cdot 10) + (0.45 \cdot 12) + (0.6 \cdot 15) + (0.9 \cdot 17)}{0.3 + 0.45 + 0.6 + 0.9} = 14.53$$

2.9.3 Weighted average method

This method is only applicable for symmetrical membership functions. It bears some resemblance to the centroid method, except it only includes the maximum membership value of membership functions. The expression is given as

$$Z = \frac{\sum \max(\mu_A(x))\, x}{\sum \max(\mu_A(x))}, \tag{16}$$

where $\max(\mu_A(x))$ is the maximum membership degree of membership function μ_A and x is the corresponding value. Assume we have two functions, μ_A and μ_B, where $\max(\mu_A) = 0.8$ and $\max(\mu_B) = 0.75$. The corresponding points on the x-axis are *a* and *b*, respectively. Then the defuzzified value *Z* can be obtained as

$$Z = \frac{(a \cdot 0.8) + (b \cdot 0.75)}{0.8 + 0.75}$$

2.9.4 Mean-max membership

This method is similar to the max-membership principle, except the maximum does not necessarily have to be unique. Hence the maximum membership degree may include more than a single point, it may include a range of points. The expression is given as

$$Z = \frac{a + b}{2}, \tag{17}$$

where *a* and *b* are the end points of the maximum membership range.

2.10 Fuzzy Relations

A relation signifies a relationship between set elements of two or more sets. Crisp relations can be defined by a characteristic function which assigns a value from the binary pair {0,1} to each subset of the universe set, where 0 implies no association and 1 implies an association. The Cartesian product of two sets *A* and *B*, denoted $A \times B$, is the set of all possible combinations of the elements in *A* and *B*. All relations are subsets of the Cartesian product which therefore represents the universe set. A fuzzy relation [34,40] is a fuzzy set defined on the Cartesian product of crisp sets. Each element within the relation may then be associated to a degree between 0 and 1, in the same manner as set membership is represented in fuzzy sets. The grade indicates the strength of the relation present between the elements.

To express this more formally, we consider a fuzzy relation between two crisp sets *X* and *Y*. Then a fuzzy relation R is a mapping from $X \rightarrow Y$ from the Cartesian space, $X \times Y$, to the unit interval. The strength of the mapping is expressed by the membership function, $\mu_R(x, y)$, of the relation for all ordered pairs $(x, y) \in X \times Y$. This function can be expressed as

$$\mu_R(x, y) = \mu_{A\times B}(x, y) = \min(\mu_A(x), \mu_B(y)). \tag{18}$$

This means that each fuzzy set can be regarded as a vector of membership values where each value is associated with a particular element in each set. If we consider two fuzzy sets *A,* containing four elements, and *B*, containing five elements. Then *A* is expressed as column vector of size 3×1 and *B* a column vector of size 1×2. The corresponding relation will be a 3×2 matrix. That is, a matrix with four rows and five columns (note: the resulting matrix has the same number of rows as *A* and the same number of columns as *B*). Lets illustrate this by an example where *A* is defined on the universe set $X = \{x_1, x_2, x_3\}$, and *B* is defined on the universe set $Y = \{y_1, y_2\}$. We then have the two following vectors

$$A = 0.4/x_1 + 0.7/x_2 + 0.1/x_3 \text{ and } B = 0.5/y_1 + 0.8/y_2.$$

The resulting matrix is then obtained by taking the minimum of each pair of $(x, y) \in A \times B$. For example, the entries (x_1, y_2) and (x_2, y_1) of the matrix are derived by taking the minimum of the pairs $(0.4, 0.8)$ and $(0.7, 0.5)$. Hence the relational matrix of *A* and *B* looks as

$$A \times B = R = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \end{array} \begin{array}{c} \begin{array}{cc} y_1 & y_2 \end{array} \\ \begin{bmatrix} 0.4 & 0.4 \\ 0.5 & 0.7 \\ 0.1 & 0.1 \end{bmatrix} \end{array}.$$

2.10.1 Fuzzy Relational Compositions

Relations can be combined in various ways by using the union or the intersection operator. A generic way to compose fuzzy relations is to pick the minimum value in a series connection and the maximum value in a parallel connection. Because a relation is itself a set, the basic set operations such as union, intersection, and complement can be applied without modifications. The standard composition R of two fuzzy relations P and Q, normally written as $R = P \circ Q$, is formally defined by

$$R(x,z)=[P \circ Q](x,z)=\max_{y \in Y} \min[\mu_P(x,y), \mu_Q(y,z)], \tag{19}$$

for all $x \in X$ and all $z \in Z$. Less formally this means that the *ij*-entry of the matrix R is derived by combining the *i*th row of P with the *j*th column of Q. Using matrix notation, the same expression can can be written as

$$[r_{ij}] = [p_{ik}] \circ [q_{kj}] = \max \min(p_{ik}, q_{kj})$$

An illustrative example of a max-min composition of two fuzzy sets is shown below

$$P \circ Q = \begin{bmatrix} 0.7 & 0.6 \\ 0.8 & 0.3 \end{bmatrix} \circ \begin{bmatrix} 0.8 & 0.5 & 0.4 \\ 0.1 & 0.6 & 0.7 \end{bmatrix} = \begin{bmatrix} 0.7 & 0.6 & 0.6 \\ 0.8 & 0.5 & 0.4 \end{bmatrix}.$$

For example,

$$\begin{aligned} r_{1,2} &= \max[\min(p_{11}, q_{12}), \min(p_{12}, q_{22})] \\ &= \max[\min(0.7, 0.5), \min(0.6, 0.6)] \\ &= 0.6, \end{aligned}$$

$$\begin{aligned} r_{2,2} &= \max[\min(p_{21}, q_{12}), \min(p_{22}, q_{22})] \\ &= \max[\min(0.8, 0.5), \min(0.3, 0.6)] \\ &= 0.5. \end{aligned}$$

Another related operation is the min-max operations which is derived in an analogous manner to the max-min operation.

A second example of a relational composition is the max-product which is defined by

$$R(x,z)=[P \circ Q](x,z)=\{\max_{y \subset Y}[\mu_P(x,y) * \mu_Q(y,z)]\} \tag{20}$$

for all $x \in X$ and $z \in Z$. Here the min-operator has been replaced by the multiplication operator but the entries are combined between matrices in the same manner. By reusing the previous example, we get

$$P \circ Q = \begin{bmatrix} 0.7 & 0.6 \\ 0.8 & 0.3 \end{bmatrix} \circ \begin{bmatrix} 0.8 & 0.5 & 0.4 \\ 0.1 & 0.6 & 0.7 \end{bmatrix} = \begin{bmatrix} 0.56 & 0.36 & 0.42 \\ 0.64 & 0.4 & 0.32 \end{bmatrix}.$$

For example,

$$\begin{aligned} r_{11} &= max[(p_{11}*q_{11}),(p_{12}*q_{21})] \\ &= max[(0.7*0.8),(0.6*0.1)] \\ &= 0.56, \end{aligned}$$

$$\begin{aligned} r_{23} &= max[(p_{21}*q_{13}),(p_{22}*q_{23})] \\ &= max[(0.8*0.4),(0.3*0.7)] \\ &= max[(0.32),(0.21)] \\ &= 0.32. \end{aligned}$$

A third example of a compositional operation is the max-average composition, defined by

$$R(x,z)=[P\circ Q](x,z)=\{\frac{1}{2}\max_{y\subset Y}[\mu_P(x,y)+\mu_Q(y,z)]\} \tag{21}$$

for all $x\in X$ and $z\in Z$. Compared to the previous expression, it can be seen that the multiplicative operator has been replaced by an additive operation such that we now obtain the maximum of the averages between corresponding pairs. Again by using the same example, we get

$$P\circ Q = \begin{bmatrix} 0.7 & 0.6 \\ 0.8 & 0.3 \end{bmatrix} \circ \begin{bmatrix} 0.8 & 0.5 & 0.4 \\ 0.1 & 0.6 & 0.7 \end{bmatrix} = \begin{bmatrix} 0.75 & 0.6 & 0.65 \\ 0.8 & 0.65 & 0.6 \end{bmatrix}.$$

For example,

$$\begin{aligned} r_{11} &= 1/2\max[(\mu_{p11}+\mu_{q11}),(\mu_{p12}+\mu_{q21})] \\ &= 1/2\max[(0.7+0.8),(0.6+0.1)] \\ &= 1/2\max[(1.5),(0.7)] \\ &= \max[(0.75),(0.35)] \\ &= 0.75, \end{aligned}$$

$$\begin{aligned} r_{23} &= 1/2\max[(\mu_{p21}+\mu_{q13}),(\mu_{p22}+\mu_{p23})] \\ &= 1/2\max[(0.8+0.4),(0.3+0.7)] \\ &= 1/2\max[(1.2),(1.0)] \\ &= \max[(0.6),(0.5)] \\ &= 0.6. \end{aligned}$$

2.11 Aggregation

The purpose of aggregation is to aggregate pieces of data in a desirable way in order to reach a conclusion or final decision. Typically this data is represented by numerical values which make some kind of sense in regard to the application. Hence the aggregation problem is generally regarded as the problem of reducing a series of numerical values into a single representative. Formally an aggregation operator can be defined as function *h* which assigns a real number *y* to any *n*-tuple $(x_{1,}x_2......,x_n)$ of real numbers [41]:

$$y=h(x_{1,}x_2......,x_n). \tag{22}$$

In literature, aggregation operations are generally defined over the unit interval, meaning that it is assumed that both the input and output is restricted to the unit interval. An aggregation operation of dimension $n \geq 1$ can therefore be formally described as mapping over the unit interval

$$h:[0,1]^n \rightarrow [0,1]. \tag{23}$$

The case $n=1$ is represented by the negation operator defined by $\neg h(x)=1-h(x)$. For $n \geq 2$, two classes of operators are of particular interest in fuzzy theory; triangular norms and the averaging operators.

Even though the input/output of aggregation operations often times is restricted to the unit interval, this is not a mandatory characteristic of aggregation operators. Hence the definition above can be extended to arbitrary intervals as well. In this context however, we assume that the inputs/outputs of the aggregation operators discussed in this thesis are from the unit interval, unless otherwise stated.

Logically, certain conditions are expected to be imposed on the function *h* in order for it to qualify as an aggregation operator, although there are different views on which basic properties should be fulfilled, since aggregation operators frequently are designed for certain applications. The most important thing in this case is not whether a given aggregation operator satisfies all of the basic properties associated with aggregation operators, but whether the aggregation operator in question produces a meaningful outcome from an applicative context, which in turn necessitates the presence of certain constraints. Some of the fundamental properties frequently associated with aggregations operators are enlisted below [41]:

1) $h(x)=x$ (**identity when unary**);

2) $h(0,\ldots,0)=0$ and $h(1,\ldots,1)=1$ (**boundary conditions**);

3) $h(x_1,\ldots,x_n) \leq h(y_1,\ldots,y_n)$ (**monotonic increasing**)

   if $x_i \leq y_i$ for all $i \in \mathbb{N}$;

4) *h* is continuous with respect to each of its arguments;

5) $h(x_1,\ldots,x_n)=h(x_{p(i)},\ldots,x_{p(n)})$ for all permutations p (**symmetry**);

6) $h(x,x,\ldots,x)=x$ (**idempotency**);

7) $h(x_1,x_2,x_3)=h(h(x_1,x_2),x_3)=h(x_1,h(x_2,x_3))$ (**associativity**);

8) $h^{[n]}(x_1,\ldots,e,\ldots,x_n)=h^{[n-1]}(x_1,\ldots,x_n)$ (**neutral element**);

9) $h^{[n]}(x_1,...,a,...,x_n)=a$ (**absorbent element**);

10) $\min(x_1,......,x_n) \leq h(x_1,......,x_n) \leq \max(x_1,......,x_n)$ (**compensation**).

Property 1 only applies to unary operators, i.e. operators taking one argument only. According to this property, the aggregated result equals x if h is unary. Property 2 defines the worst/best case behaviour of aggregation operators. If the argument x_i is either completely false ($x_i = 0$) or completely true ($x_i = 1$) for all $i \in \mathbb{N}$, then the aggregated result should reflect the same behaviour. The properties of boundary conditions can easily be extended to input/outputs outside the unit interval. Sometimes the boundary conditions are extended as follows [42]:

2.1) $\forall x \in [0,1]\ h(x,0) = h(1,0) \cdot x$

2.2) $\forall x \in [0,1]\ h(x,1) = (1 - h(1,0)) \cdot x + h(1,0)$.

These extensions add more constrains to the basic requirements of aggregation operators since they exclude every aggregation operator which is not an averaging operator. Property 2.1 presumes the value of $h(x,0)$ to be the weighed arithmetic mean of x and 0, and property 2.2 presumes the value of $h(x,1)$ to be the weighted arithmetic mean of x and 1. Actually the requirements of property 2 are special cases of 2.1 and 2.2 when $x = 0$ and $x = 1$, respectively. Property 3 states that the aggregated result as minimum does not decrease if the argument increases. Property 4 (continuity) ensures that a changes in arguments will not result in discontinuous change in the aggregate value. Especially the properties of 2-4 are considered fundamental to aggregation operators in general [34]. Property 5 (symmetry) is related to the order of arguments in the sense that the order should not have any influence on the aggregated result. This is particularly relevant when all arguments are assumed to be equally important. Another related property associated to n-ary operators with n^2 arguments is *bisymmetry* [41]. This property simply states that it does not matter whether the aggregation is done vertically or horizontally, if h is an n-ary operator. We can write this as

$$\begin{aligned} &h(h(x_{11}, x_{12}, \dots, x_{1n}), h(x_{21}, x_{22}, \dots, x_{2n}), \dots, h(x_{n1}, x_{n2}, \dots, x_{nn})) = \\ &h(h(x_{11}, x_{21}, \dots, x_{n1}), h(x_{12}, x_{22}, \dots, x_{n2}), \dots, h(x_{1n}, x_{2n}, \dots, x_{nn})). \end{aligned} \tag{24}$$

This implies that you can either aggregate the column vectors first and then the outputs thereof or the row vectors first and then the outputs thereof. It should be noted that symmetry and associativity implies bisymmetry, but neither symmetry or associativity is implied by bisymmetry. Property 6 (idempotency) states that if x is aggregated n times, the final outcome will be x as well. This condition may be warranted in cases where x is a fuzzy set, because aggregating equal sets logically implies the same set. Property 7(associativity) reflects the notion that aggregation is done in packages but the order of packages has no influence on the aggregated result. Property 8 (neutral element) assumes the existence of a neutral element e which has no influence on the result when applied. Property 9 (absorbent element) assumes the existence of an absorbent element a which acts as an annihilator. Property 10 (compensation) relies on the assumption that the aggregated result is somewhere between the lowest argument (min) and the highest argument (max). This condition is only valid for averaging operators.

2.11.1 Averaging Operators

A type of operator widely studied in literature are averaging operators. Averaging operators of dimension $n \geq 2$ can be described by the mapping $h: [0,1]^n \rightarrow [0,1]$ which meets the following axiomatic requirements [43,44]: (1) *h* is symmetric; (2) *h* is monotonic increasing; (3) *h* is continuous; (4) *h* is idempotent. It has to be noted that the assumption of symmetry may not always be warranted in every application context. In that case, the assumption of symmetry has to be dropped. A good example of an operator where this property is not meet is the weighted average mean. Some commonly used averaging operators are listed in table 1.

Operator	Equation
The arithmetic mean	$\frac{1}{n}\sum_{i=1}^{n} x_i$
Weighted arithmetic mean	$\sum_{i=1}^{n} w_i \cdot x_i$ where $w_i \in [0,1]$ and $\sum_{i=1}^{n} w_i = 1$
Geometric mean	$\left(\prod_{i=1}^{n} x_i\right)^{\frac{1}{n}}$
Harmonic mean	$\frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}$
Quadratic mean	$\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}$
Median	Sort the arguments in ascending order. If the number of arguments *n* is odd, then the middle value is selected. If *n* is even, then take the mean of the middle pair.
Min and max	$\min(x_1, \ldots, x_n)$ $\max(x_1, \ldots, x_n)$

Table 1. Examples of commonly used averaging operators.

A common characteristic of aggregation operators is that they cover the entire interval between min and max. That is, any aggregation operator, $h(x_1,..., x_n)$, satisfies the following inequalities (aka compensation property) [34] :

$$\forall (x_1, \ldots x_n) \in [0,1]^n: \ \min(x_1, \ldots, x_n) \leqslant h(x_1, \ldots, x_n) \leqslant \max(x_1, \ldots, x_n). \tag{25}$$

To show this, let $x_{\min}$ = min and $x_{\max}$ = max. Since every averaging operator satisfies the properties of monotonicity and idempotency, it also satisfies the inequalities:

$$x_{min}=h(x_{min},\ldots,x_{min}) \leq h(x_1,\ldots,x_n) \leq h(x_{max},\ldots,x_{max}) = x_{max}. \qquad (26)$$

This means that all averaging operators are bounded by the standard fuzzy union and the standard fuzzy intersection operations. Conversely, it follows that all operators bounded by the standard fuzzy union and standard fuzzy intersection are idempotent since

$$\begin{array}{l} x=h(x,\ldots,x)\leq h(x,\ldots,x)\leq h(x,\ldots,x)=x \\ \forall\, x\in[0,1]. \end{array} \qquad (27)$$

2.11.1.1 Generalized Means

Many of the common means belong to the family of the quasi-arithmetic means [41,44], defined as

$$h(x_1,\ldots,x_n)=f^{-1}\left(\frac{1}{n}\sum_{i=1}^{n} x_i\right), \qquad (28)$$

where f is a continuous strictly monotonic function, and f^{1} is its inverse. It can be noted that the geometric mean and the harmonic mean are particular cases of (28) with $f(x)$ = log x and $f(x) = 1/x$, respectively. A particularly noticeable case of quasi-arithmetic means can be obtained by applying the function: $f: x\rightarrow x^{\alpha}$. We can then obtain a quasi-arithmetic mean of the form:

$$h_{\alpha}(x_1,\ldots,x_n)=\left[\frac{1}{n}\sum_{i=1}^{n} x_i^{\alpha}\right]^{\frac{1}{\alpha}}. \qquad (29)$$

This class of means is often referred to as *power means* or *generalized means* because a common group of well-known means can be generalized by changing the α parameter:

- For α = 1 we obtain the arithmetic mean.
- For α = 2 we obtain the square mean (aka euclidean mean)
- For α = -1 we obtain the harmonic mean.
- When α converges towards -∞, h_α converges towards minimum.
- When α converges towards ∞, h_α converges towards maximum.
- When α converges towards 0, h_α converges towards the geometric mean.

The class of power means can be extended with weights as well such that we get weighted power means, defined by the equation:

$$h_\alpha((w_1,x_1),\ldots,(w_n,x_n))=\left[\frac{1}{n}\sum_{i=1}^{n} w_i\cdot x_i^\alpha\right]^{\frac{1}{\alpha}}$$
$$\text{where } w_i\in[0,1]\,\forall\, i \text{ and } \sum_{i=1}^{n} w_i=1\,. \tag{30}$$

Other well-known means can be generalized as well using weighted power means by changing the α parameter:

- For α = 1, h_α equals the weighted arithmetic mean.
- For α = 2, h_α equals the weighted square mean.
- For α = -1, h_α equals the weighted harmonic mean.
- When α converges towards -∞, h_α converges towards minimum.
- When α converges towards ∞, h_α converges towards maximum.
- When α converges towards 0, h_α converges towards the weighted geometric mean.

2.11.1.2 Ordered Weighted Averaging Operators (OWA)

One of the most widely studied operator in fuzzy theory is the OWA operator [43,44]. This operator is mainly used for aggregating scores associated with the satisfaction of multiple criteria. An OWA operator of dimension $n \geq 2$ can be described as the function:

$$OWA(x_{1,}x_{2,}\ldots,x_n)=\sum_{j=1}^{n} w_j\cdot x_{p(j)}$$
$$\text{where } w_i\in[0,1]\text{and}\sum_{i=1}^{n} w_i=1 \tag{31}$$

and p is a permutation that orders the arguments in a non-increasing order: $x_{p(1)}\geq x_{p(2)}\geq\cdots\geq x_{p(n)}$. Some special cases of OWA, when choosing particular weights, are displayed in table 2.

	OWA weights
Maximum	$\begin{cases} w_1 = 1 \\ w_i = 0 \quad \text{if } i \neq 1 \end{cases}$
Minimum	$\begin{cases} w_n = 1 \\ w_i = 0 \quad \text{if } i \neq n \end{cases}$
Arithmetic mean	$w_i = \frac{1}{n} \quad \forall i$
Median	$\begin{cases} w_{\frac{n+1}{2}} = 1 & \text{if } n \text{ is odd} \\ w_{\frac{n}{2}} = \frac{1}{2} \text{ and } w_{\frac{n}{2}+1} = \frac{1}{2} & \text{if } n \text{ is even} \\ w_i = 0 & \text{else.} \end{cases}$

Table 2. Special cases of OWA [41]**.**

An important aspect with respect to OWA is the derivation of appropriate weights which should represent the problem at hand as closely as possible. Two measures of importance in this regard are orness and dispersion, defined by:

$$orness(\vec{w}) = \frac{1}{n-1} \cdot \sum_{i=1}^{n} (n-i) \cdot w_i \tag{32}$$

and

$$dispersion(\vec{w}) = -\sum_{i=1}^{n} w_i \cdot \ln(w_i) \in [0, \ln(n)] \tag{33}$$

The dual measure of orness is referred to as andness, and is defined by $1 - orness(\vec{w})$. The dispersion measure reflects the degree of utilization of the information in the argument vector. The more evenly distributed the weights are, the higher dispersion. A normalized dispersion measure to the unit interval can obtained by dividing by *ln(n)* such that the equation becomes:

$$ndispersion(\vec{w}) = -\frac{1}{\ln(n)} \sum_{i=1}^{n} w_i \cdot \ln(w_i) \in [0,1] \tag{34}$$

Orness and andness can be interpreted as the degree to which an OWA operator represents pure OR (i.e. max) or pure AND (i.e. min), respectively. The degree of orness of the arithmetic mean is 0.5, as can be seen from the following example. Consider the vector of OWA weights $\vec{w} = (0.2, 0.2, 0.2, 0.2, 0.2)$. The orness can then be calculated as:

$$\begin{aligned} orness(\vec{w}) &= \frac{1}{4}(0.8 + 0.6 + 0.4 + 0.2) \\ &= \frac{1}{2}. \end{aligned}$$

So if orness equals 0.5, we obtain neutrality with the arithmetic mean. If orness is strictly less than

0.5, we move towards min, and thereby also the pure AND. If orness is strictly larger 0.5, we move towards max, and thereby the pure OR.

Clearly, it is possible to obtain the same degree of orness for different weight vectors. By using the dispersion measure, we are able to further distinguish between the OWA weights. To illustrate this, consider the two OWA vectors $\vec{w}_1=(0,0,1,0,0)$ and $\vec{w}_2=(0.2,0.2,0.2,0.2,0.2)$. These two vectors respectively correspond to the median and the arithmetic mean. The orness and the dispersion for these vectors are calculated as:

$$\begin{aligned}
orness(\vec{w}_1) &= \frac{1}{5-1}((5-1)\cdot 0+(5-2)\cdot 0+(5-3)\cdot 1+(5-4)\cdot 0+(5-5)\cdot 0)\\
&= 0.5\\
ndispersion(\vec{w}_1) &= -\frac{(0\cdot\ln(0))+(0\cdot\ln(0))+(1\cdot\ln(1))+(0\cdot\ln(0))+(0\cdot\ln(0))}{\ln(5)}\\
&= 0\\
orness(\vec{w}_2) &= 0.5(\text{as already shown})\\
ndispersion(\vec{w}_2) &= -\frac{(0.2\cdot\ln(0.2))+(0.2\cdot\ln(0.2))+(0.2\cdot\ln(0.2))+(0.2\cdot\ln(0.2))+(0.2\cdot\ln(0.2))}{\ln(5)}\\
&= 1
\end{aligned}$$

As can be seen from this example, different results for the normalized dispersion can be obtained, despite the same degree of orness for the two vectors.

2.11.2 Triangular Norms (T-Norms and T-Conorms)

Another class of operators which have been extensively studied in literature, are the so-called triangular norms [41,45] which can be divided into two basic operations, namely the t-norm and its dual the t-conorm. In fuzzy set theory, the t-norm defines the union and the t-conorm defines intersection of fuzzy sets. This makes it possible to use these to characterize the logical connectives of AND and OR, respectively.

A t-norm is a function $T:[0,1]^2\rightarrow[0,1]$ which satisfies the following axioms:

- $T(x,y)=T(y,x)$ (T1) **commutativity**
- $T(x,y)\leqslant T(u,v)$, if $x\leqslant u\wedge y\leqslant v$ (T2) **monotonicity (increasing)**
- $T(x,(T(y,z))=T(T(x,y),z))$ (T3) **associativity**
- $T(x,1)=x$ (T4) **1 as neutral element**

The result of applying the t-norm operator will never be larger than the minimum of arguments. Formally this can be written as:

$$\forall \text{ t-norms } T : T(x, y) \leqslant \min(x, y). \tag{35}$$

We can prove this as follows:

1. From T2 and T4 we get $T(x, y) \leqslant T(x,1) = x$.
2. From T1, T2 and T4 we get $T(x, y) \leqslant T(1, y) = y$.

That is $T(x, y) \leqslant x$ and $T(x, y) \leqslant y$, hence $T(x, y) \leqslant \min(x, y)$.

A t-conorm is a function $S:[0,1]^2 \rightarrow [0,1]$ which satisfies the following axioms:

- $S(x, y) = S(y, x)$ (S1) **commutativity**
- $S(x, y) \leqslant S(u, v)$, if $x \leqslant u \wedge y \leqslant v$ (S2) **monotonicity (increasing)**
- $S(x,(S(y, z)) = S(S(x, y), z))$ (S3) **associativity**
- $S(x,0) = x$ (S4) **0 as neutral element**

From an axiomatic point of view, t-norms and t-conorms only differ with the respect to their boundary conditions or neutral element which is 1 and 0, respectively. The result of applying the t-conorm operator is never less than the maximum of arguments. The formal notation is:

$$\forall \text{ t-conorms } S : S(x, y) \geq \max(x, y). \tag{36}$$

The proof is trivial and analogous to the one shown previously.

Norm operations are always defined as binary operations, but due to their associative properties, they can be generalized for *n* arguments. For example, the multi-argument forms for the min and max operators are:

$$T_{i=1}^{n}(x_i)_{\min} = \min_{i=1}^{n}(x_i) \text{ and } S_{i=1}^{n}(x_i)_{\max} = \max_{i=1}^{n}(x_i), \text{ respectively.} \tag{37}$$

For the algebraic product and the algebraic sum, the multi-argument forms are:

$$T_{i=1}^{n}(x_i)_{ap} = \prod_{i=1}^{n}(x_i) \text{ and } S_{i=1}^{n}(x_i)_{as} = 1 - \prod_{i=1}^{n}(1 - x_i), \text{ respectively.} \tag{38}$$

Generally multi-argument forms are trivial with the algebraic sum as an exception. Therefore, the derivation of the multi-argument form of the algebraic sum is shown as a proof of induction in Appendix I.

2.11.2.1 Duality of T-Norms and T-Conorms

Any t-norm is associated with a dual t-conorm and vice versa [41,45]. A t-norm and a t-conorm are said to be dual if the law of De Morgan is satisfied:

$$\overline{T(x,y)}=S(\overline{x},\overline{y}), \tag{39}$$

where $\overline{x}$ denotes the standard negation, defined by $\overline{x}=1-x$. Some common t-norms and their dual t-conorms are listed in table 3.

	t-norm	**t-conorm**
min and max	min(x, y)	max(x, y)
algebraic product and sum	$x \cdot y$	$x + y - x \cdot y$
Lukasiewicz t-norm and t-conorm	max($x + y$ - 1,0)	min($x + y$,1)
drastic product and sum	$\begin{cases} 0 & \text{if } (x,y)\in[0,1[^2, \\ \min(x,y) & \text{otherwise.} \end{cases}$	$\begin{cases} 1 & \text{if } (x,y)\in]0,1]^2 \\ \max(x,y) & \text{otherwise.} \end{cases}$

Table 3. Common t-norms and their dual t-conorms.

The minimum or min is the largest t-norm. It is also the only idempotent t-norm and thus the only t-norm which is an averaging operator as well. Its dual t-conorm, i.e. the max operator, is the smallest t-conorm. It is the only idempotent t-conorm and thus the only t-conorm which is an averaging operator as well. Hence the min and max respectively define the lower and upper bounds of averaging operators. The drastic product is interesting from the point of view that it yields the smallest t-norm and the largest t-conorm.

2.11.3 Averaging Operators and Triangular Norms in Context

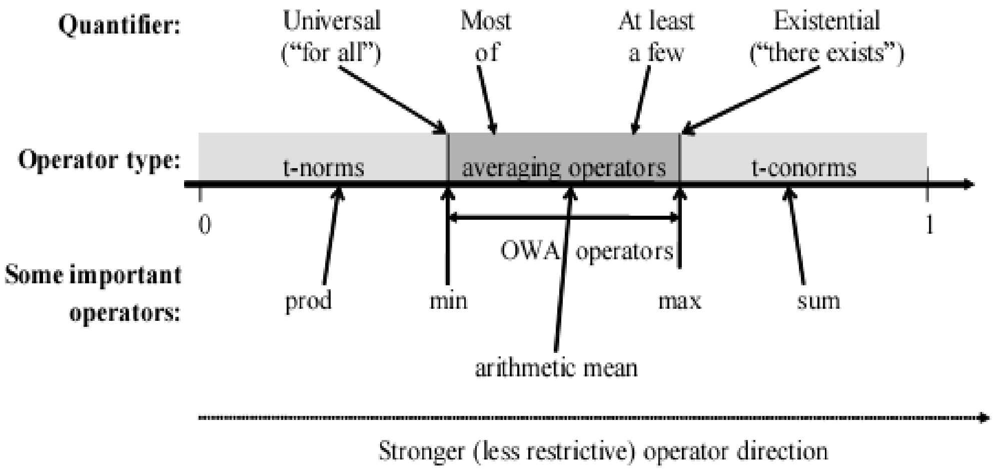

Figure 7. The relationship between triangular norms and averaging operators [44] **.**

Figure 7 summarizes the relationship between the different classes of operators discussed in the previous sections. It can be seen from the figure that the boundary between averaging operators and triangular norms is defined by the min and max operators. Recall that the result of a t-norm operation is always ≤ min, and for t-conorm operations, the result is always ≥ max. In particular, these operators are important when distinguishing between triangular norms and averaging operators because the min and max are the only idempotent triangular norms and thereby the only triangular norms that are averaging operators as well. Moreover these operators are the only associative averaging operators.

Averaging operators satisfy the compensation property which implies that an averaging operator always yield a result between min and max. Weighted averaging operators, like OWA, can be regarded as parametrized ways of moving between min and max. Moving towards min (or 0) corresponds to moving towards pure AND. As we move closer towards AND, the more restrictive the operator becomes, since pure AND requires all arguments to be satisfied. This is equivalent to the universal quantifier which states that all arguments must be fulfilled. At the opposite end of the extreme, we have max, corresponding to the pure OR. This is equivalent to the existential quantifier, which states, that there exists at least one argument which is fulfilled. So, the further we move towards max (or 1), the less restrictive the operator becomes. Between these two extremes different levels of strictness can be specified. For example, a query may be satisfied if "most off" the arguments are fulfilled or "at least a few".

From an axiomatic point of view, triangular norms and averaging operators have the symmetry, monotonicity and continuity axioms in common. The axioms regarding associativity and the existence of a neutral element only applies to triangular norms. In fact, triangular norms cover all aggregating operations which are associative [34]. Idempotency, on the other hand, only applies to triangular norms.

2.12 Particle Swarm Optimization (PSO)

PSO [46,47] is an optimization technique applicable to continuous non-linear functions. It was first introduced by Kennedy and Eberhart in [46]. The algorithm simulates the social behaviours shown by various kinds of organisms such as bird flocking or fish schooling. Imagine a group of birds randomly foraging in an area. The group shares the common goal of locating a single piece food. While foraging, individual birds may learn from the discoveries and past experiences of other birds through social interaction. Each bird synchronizes its movements with group while simultaneously avoiding collisions with other birds. As the search continues, the birds move closer toward the place where the food is by following the bird which is closest to the food.

In PSO, bird flocks are represented as particle swarms searching for the best solution in a virtual search space. A fitness value is associated to each particle which is evaluated against a fitness function to be optimized, and the movement of each particle is directed by a velocity parameter. During each iteration, particles move about randomly within a limited area, but individual particle movement is directed toward the particle which is closest to the optimal solution. Each particle remembers its personal best position (the best position found by the particle itself) as well as the global best position (the best solution found by any particle in the group). The parameters are updated each time another best position is found. This way, the solution evolves as each particle moves about.

Compared to other related approaches such as genetic algorithms and neural networks, PSO it is quite simple and easy to implement. It is initialized with a set of randomly generated particles which in fact are candidate solutions. An iterative search process is then set in motion to improve the set of current solutions. During each iteration, new solutions are proposed by each particle which are individually evaluated against: (1) the particles own personal best solution found in any proceeding iteration and (2) the global best solution currently found by any particle in the swarm. We refer to each candidate solution as a position. If a particle finds a position better than its current personal best position, its personal best position is updated. Moreover, if the new personal best position is better than the current global best position, the global best position is updated. After the evaluation process is completed, each particle updates its velocity and position with the equations:

$$v_i = \omega v_i + c_1 r_1 (\hat{x}_i - x_j) + c_2 r_2 (\hat{g} - x_j) \tag{40}$$

$$x_j = x_j + v_i, \tag{41}$$

where

- v_i is the velocity of particle p_i and is limited to[-V_{max}, V_{max}] where V_{max} is user-defined constant.
- ω is an inertial weight coefficient.
- $\hat{x}_i$ is the current personal best position.
- x_j is the present position.
- $\hat{g}$ is the global best position.
- c_1 and c_2 are user defined constants that say how much the particle is directed towards good positions. They affect how much the particle's local best and global best influence its

movement. Generally c_1 and c_2 are set to 2.

- r_1 and r_2 are randomly generated numbers between 0 and 1.

Note that the velocity controls the motion of each particle. The speed of convergence, can be adjusted by the inertial weight coefficient and the constants c_1, c_2. Whenever computed velocity exceeds its user-defined boundaries, the computed results will be replaced by either -V_{min} or V_{max}. The running procedure of basic PSO algorithm is summarized in pseudo code below.

The basic PSO algorithm

```
for all particles{
        initialize velocities and positions
        }//end for
while stopping criteria is unsatisfied{
        for each particle{
            1. compute velocities by equation (40)
            2. increment positions by equation (41)
               if present fitness value is better than current local best value
            3. update local best positions
               if present fitness value is better than current global best value
            4. update global best positions
        }//end for
}//end while
```

2.13 Fuzzy Time Series and its Concepts

In the following section we will briefly review some of the fundamental concepts of FTS as they originally were conceived in pioneering publications by Song/Chissom [1,2,48] and Chen [3, 31].

Definition 1: Fuzzy Time Series

Let $Y(t)(t=...,0,1,2,...)$, a subset of real numbers, be the universe of discourse on which fuzzy sets $f_i(t)(i=1,2,...)$ are defined. If $F(t)$ is a collection of $f_i(t)(i=1,2,...)$, then $F(t)$ is called a fuzzy time series on $Y(t)(t=...,0,1,2,...)$.

Definition 2: Fuzzy Relation

If there exists a fuzzy relationship $R(t-1,t)$, such that $F(t)=F(t-1)\times R(t-1,t)$, where × represents an operator, then $F(t)$ is said to be caused by $F(t-1)$. The relationship between $F(t)$ and $F(t-1)$ is denoted by

$$F(t-1)\rightarrow F(t). \tag{42}$$

Examples of operators from literature are the max-min composition (see section 2.10.1) [1], the min-max composition [2] and the arithmetic operator [3]. If $F(t-1)=A_i$ and $F(t)=A_j$, the logical relationship between $F(t)$ and $F(t-1)$ is denoted by $A_i \rightarrow A_j$, where A_i is called the left hand side and A_j the right hand side of the fuzzy relation. The variable *t* denotes the time. For example, if *t* = 1973, the fuzzy relationship between $F(t)$ and $F(t-1)$ is given by $F(1972) \rightarrow F(1973)$. Note the right hand side of the fuzzy relation represents the future fuzzy set (forecast). Its crisp counterpart is denoted as $Y(t)$.

Definition 3: *N*-Order Fuzzy Relations

Let $F(t)$ be a fuzzy time series. If $F(t)$ is caused by $F(t-1)$, $F(t-2),\ldots,F(t-n)$, then this fuzzy relationship is represented by

$$F(t-n),\ldots,F(t-2),\ F(t-1) \rightarrow F(t), \tag{43}$$

and is called an *n*-order fuzzy time series. The *n*-order concept was first introduced by Chen in [31]. *N*-order based FTS models are referred to as high order models.

Definition 4: Time-Invariant Fuzzy Time Series

Suppose $F(t)$ is caused by $F(t-1)$ only and is denoted by $F(t-1) \rightarrow F(t)$, then there is a fuzzy relationship between $F(t)$ and $F(t-1)$ which is expressed as the equation:

$$F(t)=F(t-1) \times R(t-1,t). \tag{44}$$

The relation *R* is referred to as a first order model of $F(t)$. If $R(t-1,t)$ is independent of time t, that is, for different times t_1 and t_2, $R(t_1,t_1-1)=R(t_2,t_2-1)$, then $F(t)$ is called a time-invariant fuzzy time series. Otherwise it is called a time-variant fuzzy time series.

Definition 5: Fuzzy Relationship Group (FLRG)

Relationships with the same fuzzy set on the left hand side can be further grouped into a relationship group. Relationship groups are also referred to as fuzzy logical relationship groups or FLRG 's in short. Suppose there are relationships such that

$$\begin{aligned} &A_i \rightarrow A_{j1,} \\ &A_i \rightarrow A_{j2,} \\ &\ldots \\ &A_i \rightarrow A_{jn,} \end{aligned}$$

then they can be grouped into a relationship group as follows:

$$A_i \rightarrow A_{j1\,,}\, A_{j2,\ldots,}\, A_{jn}. \tag{45}$$

The same fuzzy set cannot appear more than once on the right hand side or the relationship group. The term relationship group was first introduced by Chen in [3].

2.14 Conclusion

Various theoretical concepts have been reviewed in this section such as fuzzy sets, fuzzy numbers, defuzzification, fuzzy relations, fuzzy aggregation, PSO and FTS. The main purpose of this discussion has been to provide self-contained study of the underlying theoretical concepts of the forecasting model presented in later sections.

3 Related Work

This section provides an overview of current research. FTS has been subjected to extensive research since first introduced almost 2 decades ago. However, the intention here is not to provide an exhaustive study of every work published. Rather the intention is to provide a general overview of FTS as an independent research field. First we will briefly review Song and Chissom's work [1,2, 48] which is the earliest work published on FTS. Next, a more detailed study is provided of Chen's work published in [3,31], as the work presented in the respective papers are among the most important milestones in this particular field of research. Finally, a brief review of more recent developments is provided.

3.1 Song and Chissom's Work

FTS was originally proposed by Song and Chissom[1,2,48] in a series of papers to forecast student enrollments at the University of Alabama. The motivation for introducing a new forecasting framework, based on fuzzy set theory, was the need to model time series problems when historical data are defined as linguistic values. The first model published was the so-called time-invariant model which comprises the followings steps: (1) define the universe of discourse; (2) partition the universe of discourse into equally lengthy intervals; (3) define fuzzy sets of the universe of discourse; (4) fuzzification of historical data; (5) establish fuzzy relations; (6) forecast by $A_i = A_{i-1} \circ R$, where $\circ$ is the max-min operator (see section 2.10.1); and (7) defuzzify forecasted results.

In step (5), the fuzzy relation *R* is defined by

$$R_i = A_s^T \times A_q, \text{ for all } k \text{ relations } A_s \rightarrow A_q, R = \bigcup_{i=1}^{k} R_i, \tag{46}$$

where × is the min operator, *T* is the transpose operator, and ∪ is the union operator.

Subsequently Song and Chissom proposed the time-variant model, which basically comprises the same steps as its time-invariant counterpart. The most notable difference is the notion of fuzzy relationship in step 5, denoted as $R^w(t, t-1)$, and defined by

$$R_i = f^T(t-i) \times f(t-i+1) \text{ for all } k \text{ fuzzy relations } A_s \rightarrow A_q, R^w(t, t-1) = \bigcup_{i=2}^{w} R_i, \tag{47}$$

where *w* is the window base, *T* is the transpose operator, × is the Cartesian product, and ∪ is the union.

3.2 Chen's Work

A significant drawback of the FTS models developed by Song and Chissom is that they are associated with unnecessary high computational overheads due to complex matrix operations in step 5 and 6. In order to reduce the computation overhead of the time-variant and time-invariant models, Chen [3] proposed a simplified model including only simple arithmetic operations. The step-by-step procedure proposed by Chen is listed as:

1. Partition the universe of discourse into equally lengthy intervals.
2. Define fuzzy sets on the universe of discourse.
3. Fuzzify historical data.
4. Identify fuzzy relationships (FLR's).
5. Establish fuzzy relationship groups (FLRG's).
6. Defuzzify the forecasted output.

In the following it will be demonstrated how the model is used to forecast student enrollments at the University of Alabama. Actual enrollment data for the period 1971 - 1992 are shown in table 4.

Year	Student enrollments	Year	Student enrollments
1971	13055	1982	15433
1972	13563	1983	15497
1973	13867	1984	15145
1974	14696	1985	15163
1975	15460	1986	15984
1976	15311	1987	16859
1977	15603	1988	18150
1978	15861	1989	18970
1979	16807	1990	19328
1980	16919	1991	19337
1981	16388	1992	18876

Table 4. Historical student enrollments 1971 - 1992, at Alabama University.

Step 1: Define the universe of discourse and partition it into equally lengthy intervals

The universe of discourse U is defined as $[D_{min}-D_{1,} D_{max}-D_2]$ where $D_{\min}$ and $D_{\max}$ are the minimum and maximum historical enrollment, respectively. From table 4, we get $D_{min}=13055$ and $D_{max}=19337$. The variables D_1 and D_2 are just two positive numbers, properly chosen by the user. If we let D_1 = 55 and D_2 = 663, we get $U=[13000,20000]$. Chen used seven intervals which is the same number used in most cases observed in literature. Dividing U into seven evenly lengthy intervals u_1, u_2, u_3, u_4, u_5, u_6 and u_7, we get u_1 = [13000, 14000], u_2 = [14000, 15000], u_3 = [15000, 16000], u_4 = [16000, 17000], u_5 = [17000, 18000], u_6 = [18000, 19000] and u_7 = [19000, 20000].

Step 2: Define fuzzy sets on the universe of discourse

Assume $A_1, A_2, \ldots, A_k$ to be fuzzy sets which are linguistic values of the linguistic variable 'enrollments'. Then the fuzzy sets $A_1, A_2, \ldots, A_k$ are defined on the universe of discourse as

$$A_1 = a_{11}/u_1 + a_{12}/u_2 + \ldots + a_{1\mathrm{m}}/u_m,$$
$$A_2 = a_{21}/u_1 + a_{22}/u_2 + \ldots + a_{2\mathrm{m}}/u_m,$$
$$\vdots$$
$$A_k = a_{kl}/u_1 + a_{k2}/u_2 + \ldots + a_{km}/u_m,$$

where $a_{ij} \in [0,1]$, $1 \leqslant i \leqslant k$, and $1 \leqslant j \leqslant m$. The variable a_{ij} represents the membership degree of the crisp interval u_j in the fuzzy set A_i. Prior to defining fuzzy sets on the U, linguistic values should be assigned to each fuzzy set. Chen uses the linguistic values A_1 = (not many), A_2 = (not too many), A_3 = (many), A_4 = (many many), A_5 = (very many), A_6 = (too many) and A_7 = (too many many). Fuzzy sets can be defined on the universe of discourse as follows:

$$A_1 = 1/u_1 + 0.5/u_2 + 0/u_3 + 0/u_4 + 0/u_5 + 0/u_6 + 0/u_7$$
$$A_2 = 0.5/u_1 + 1/u_2 + 0.5/u_3 + 0/u_4 + 0/u_5 + 0/u_6 + 0/u_7$$
$$A_3 = 0/u_1 + 0.5/u_2 + 1/u_3 + 0.5/u_4 + 0/u_5 + 0/u_6 + 0/u_7$$
$$A_4 = 0/u_1 + 0/u_2 + 0.5/u_3 + 1/u_4 + 0.5/u_5 + 0/u_6 + 0/u_7$$
$$A_5 = 0/u_1 + 0/u_2 + 0/u_3 + 0.5/u_4 + 1/u_5 + 0.5/u_6 + 0/u_7$$
$$A_6 = 0/u_1 + 0/u_2 + 0/u_3 + 0/u_4 + 0.5/u_5 + 1/u_6 + 0.5/u_7$$
$$A_7 = 0/u_1 + 0/u_2 + 0/u_3 + 0/u_4 + 0/u_5 + 0.5/u_6 + 1/u_{7.}$$

Step 3: Fuzzify historical data

In this context, fuzzification is the process of identifying associations between the historical values in the dataset and the fuzzy sets defined in the previous step. Each historical value is fuzzified according to its highest degree of membership. If the highest degree of belongingness of a certain historical time variable, say $F(t-1)$, occurs at fuzzy set A_k, then $F(t-1)$ is fuzzified as A_k. To exemplify this, let us fuzzify year 1971. According to table 4, the enrollment in 1971 was 13055 which lies within the boundaries of interval u_1. Since the highest membership degree of u_1 occurs at A_1, the historical time variable $F(1971)$ is fuzzified as A_1. Actual enrollment of 1974 is 14696 which lies within the boundaries of interval u_2. Hence $F(1974)$ is fuzzified as A_2. A complete overview of fuzzified enrollments is shown in the table 5.

Year	Actual enrollment	Interval	Fuzzified enrollment
1971	13055	[13000, 14000]	A_1
1972	13563	[13000, 14000]	A_1
1973	13867	[13000, 14000]	A_1
1974	14696	[14000, 15000]	A_2
1975	15460	[15000, 16000]	A_3
1976	15311	[15000, 16000]	A_3
1977	15603	[15000, 16000]	A_3
1978	15861	[15000, 16000]	A_3
1979	16807	[16000, 17000]	A_4
1980	16919	[16000, 17000]	A_4
1981	16388	[16000, 17000]	A_4
1982	15433	[15000, 16000]	A_3
1983	15497	[15000, 16000]	A_3
1984	15145	[15000, 16000]	A_3
1985	15163	[15000, 16000]	A_3
1986	15984	[15000, 16000]	A_3
1987	16859	[16000, 17000]	A_4
1988	18150	[18000, 19000]	A_6
1989	18970	[18000, 19000]	A_6
1990	19328	[19000, 20000]	A_7
1991	19337	[19000, 20000]	A_7
1992	18876	[18000, 19000]	A_6

Table 5. Fuzzified historical enrollments.

Step 4: Identify fuzzy relationships

Relationships are identified from the fuzzified historical data. If the time series variable $F(t-1)$ is fuzzified as A_k and $F(t)$ as A_m, then A_k is related to A_m. We denote this relationship as $A_k \rightarrow A_m$, where A_k is the current state of enrollment and A_m is the next state of enrollment. From table 5, we can see that year 1971 and 1972 both are fuzzified as A_1, which provides the following relationship: $A_1 \rightarrow A_1$. The complete set of relationships identified from table 5 are presented in the table 6.

$A_1 \rightarrow A_1$	$A_1 \rightarrow A_2$	$A_2 \rightarrow A_3$	$A_3 \rightarrow A_3$
$A_3 \rightarrow A_4$	$A_4 \rightarrow A_4$	$A_4 \rightarrow A_3$	$A_4 \rightarrow A_6$
$A_6 \rightarrow A_6$	$A_6 \rightarrow A_7$	$A_7 \rightarrow A_7$	$A_7 \rightarrow A_6$

Table 6. Fuzzy set relationships.

Note that even though the same relationships may appear more than once, these are ignored since there can only be one relationship of the same kind.

Step 5: Establish fuzzy relationship groups (FLRG's)

If the same fuzzy set is related to more than one set, the right hand sides are merged. We refer to this process as the establishment of FLRG's. For example, we see from table 6 that A_1 is related to itself and to A_2. This provides the following FLRG: $A_1 \rightarrow A_1,\ A_2$. A complete overview of the relationship groups obtained from table 6 is shown table 7.

Group 1:	$A_1 \rightarrow A_1$	$A_1 \rightarrow A_2$	
Group 2:	$A_2 \rightarrow A_3$		
Group 3:	$A_3 \rightarrow A_3$	$A_3 \rightarrow A_4$	
Group 4:	$A_4 \rightarrow A_4$	$A_4 \rightarrow A_3$	$A_4 \rightarrow A_6$
Group 5:	$A_6 \rightarrow A_6$	$A_6 \rightarrow A_7$	
Group 6:	$A_7 \rightarrow A_7$	$A_7 \rightarrow A_6$	

Table 7. FLRG's.

Step 6: Defuzzify the forecasted output

Assume the fuzzified enrollment of $F(t-1)$ is A_j, then forecasted output of $F(t)$ is determined according to the following principles:

1. If there exists a one-to-one relationship in the relationship group of A_j, say $A_j \rightarrow A_k$, and the highest degree of belongingness of A_k occurs at interval u_k, then the forecasted output of $F(t)$ equals the midpoint of u_k.

2. If A_j is empty, i.e. $A_j \rightarrow \emptyset$, and the interval where A_j has the highest degree of belongingness is u_j, then the forecasted output equals the midpoint of u_j.

3. If there exists a one-to-many relationship in the relationship group of A_j, say $A_j \rightarrow A_1, A_2, \ldots, A_n$, and the highest degrees of belongingness occurs at set $u_1,\ u_2, \ldots, u_n$, then the forecasted output is computed as the average of the midpoints $m_1,\ m_2, \ldots, m_n$ of $u_1,\ u_2, \ldots, u_n$. This equation can be expressed as:

$$\frac{m_1 + m_2 + \ldots + m_n}{n}.$$

For example, year 1972 is forecasted using the fuzzified enrollments of 1971. According to table 5, the fuzzified enrollments of year 1971 is A_1. From table 7 it can be seen that A_1 is related to A_1 and A_2. The highest degrees of belongingness of A_1 and A_2 are the sets of u_1 and u_2, where u_1 = [13000, 14000] and u_2 = [14000, 15000]. The midpoints of the intervals, u_1 and u_2, are 13500 and 14500, respectively. Using rule 3, the forecasted enrollment of 1972 is computed as (13500+14500)/2 = 14000. Year 1980 is forecasted using the fuzzified enrollments of 1979 as basis.

Because the fuzzified enrollment of 1979 is A_4, we have the following FLRG: $A_4 \rightarrow A_3, A_4, A_6$. The highest degrees of belongingness for the fuzzy sets A_3, A_4 and A_6 are at intervals u_3 = [15000, 16000], u_4 = [16000, 17000] and u_6 = [18000, 19000], respectively, and the midpoints of u_3, u_4 and u_6 are 15500, 16500 and 18500, respectively. Therefore the forecasted output is calculated as (15500+16500+18500)/3 = 16833. Since there are no empty relationship groups, rule 2 is never applied in this example.

Year	Actual enrollment	Forecasted enrollment	FLRG's	Interval midpoints
1971	13055		$A_1 \rightarrow A_1, A_2$	13500; 14500
1972	13563	14000	$A_1 \rightarrow A_1, A_2$	13500; 14500
1973	13867	14000	$A_1 \rightarrow A_1, A_2$	13500; 14500
1974	14696	14000	$A_2 \rightarrow A_3$	15500
1975	15460	15500	$A_3 \rightarrow A_3, A_4$	15500; 16500
1976	15311	16000	$A_3 \rightarrow A_3, A_4$	15500; 16500
1977	15603	16000	$A_3 \rightarrow A_3, A_4$	15500; 16500
1978	15861	16000	$A_3 \rightarrow A_3, A_4$	15500; 16500
1979	16807	16000	$A_4 \rightarrow A_3, A_4, A_6$	15500; 16500; 18500
1980	16919	16833	$A_4 \rightarrow A_3, A_4, A_6$	15500; 16500; 18500
1981	16388	16833	$A_4 \rightarrow A_3, A_4, A_6$	15500; 16500; 18500
1982	15433	16833	$A_3 \rightarrow A_3, A_4$	15500; 16500
1983	15497	16000	$A_3 \rightarrow A_3, A_4$	15500; 16500
1984	15145	16000	$A_3 \rightarrow A_3, A_4$	15500; 16500
1985	15163	16000	$A_3 \rightarrow A_3, A_4$	15500; 16500
1986	15984	16000	$A_3 \rightarrow A_3, A_4$	15500; 16500
1987	16859	16000	$A_4 \rightarrow A_3, A_4, A_6$	15500; 16500; 18500
1988	18150	16833	$A_6 \rightarrow A_6, A_7$	18500; 19500
1989	18970	19000	$A_6 \rightarrow A_6, A_7$	18500; 19500
1990	19328	19000	$A_7 \rightarrow A_6, A_7$	18500; 19500
1991	19337	19000	$A_7 \rightarrow A_6, A_7$	18500; 19500
1992	18876	19000		

Table 8. Forecasted enrollments for the period 1972 - 1992.

The model explored in so far is referred to as a first order FTS model. Chen later introduced its high order counterpart which incorporates *n*-order relationships in [31]. In the high order variant, relationships of order $n \geq 2$ can be expressed as $A_{i,1}, A_{i,2}, \ldots, A_{i,n} \rightarrow A_{i,n+1}$. For example, a second order relationship is denoted by $A_{i,1}, A_{i,2} \rightarrow A_{i,3}$. A third order relationship is denoted by $A_{i,1}, A_{i,2}, A_{i,3} \rightarrow A_{i,4}$. All second order relationships identified from table 5 are listed in table 9.

$A_1, A_1 \rightarrow A_1$	$A_3, A_3 \rightarrow A_4$	$A_3, A_4 \rightarrow A_6$
$A_1, A_1 \rightarrow A_2$	$A_3, A_4 \rightarrow A_4$	$A_4, A_6 \rightarrow A_6$
$A_1, A_2 \rightarrow A_3$	$A_4, A_4 \rightarrow A_4$	$A_6, A_6 \rightarrow A_7$
$A_2, A_3 \rightarrow A_3$	$A_4, A_4 \rightarrow A_3$	$A_6, A_7 \rightarrow A_7$
$A_3, A_3 \rightarrow A_3$	$A_4, A_3 \rightarrow A_3$	$A_7, A_7 \rightarrow A_6$

Table 9. Second order relationships.

The grouping of relationships is somewhat different compared to the first order variant. In high order models, relations with identical left hand sides are not merged into a single entity in the same manner as in the first order case. To illustrate this, consider the following relationship from table 9, for *t* = 1973:

$$F(1971), F(1972) \rightarrow F(1973) = A_1, A_1 \rightarrow A_1 .$$

An ambiguity occurs in this case because another relation is found with an identical left hand side, namely $A_1, A_1 \rightarrow A_2$. To deal with this ambiguity, the current relationship is extended to a third order relation as follows:

$$F(1970), F(1971), F(1972) \rightarrow F(1973) = \#, A_1, A_1 \rightarrow A_1 .$$

The # symbol indicates null set since $F(1970)$ does not exist. If no other third order relation exists with identical left hand sides, $F(1973)$ can be defuzzified by the same principles as in the first order case, otherwise another *n* + 1 order extension has to be made.

Table 10 shows the performance of different *n*-order models in form of MSE (see equation 54).

Order	MSE
1	407507
2	89093
3	86694
4	89376
5	94539
6	98215
7	104056
8	102179
9	102789

Table 10. Forecast accuracy for different orders.

In some cases, higher forecasting accuracies can be accomplished with higher model orders, as is the case with the enrollment data. But increasing the order from *n* to *n* + 1, does not necessarily result in higher accuracy rates for all cases.

3.3 Other Developments

Generally, the focus of current FTS research has been on the establishment of fuzzy relationships and interval partitions. Early studies, in particular, were entirely devoted to the former issue such as Song/Chissom [1,2,48], Chen [3], Hwang et al [16], Sullivan/Woodall [20]. Other more recent studies dealing with the relationship aspect can also be found, like Tsaur/Woodall [14] and Sing [5].

More recently, interval partitions have received a considerable amount attention in current studies. A major reason for this paradigm shift is the need for formalized approaches to interval partitioning. In early studies, intervals were assumed to be subjectively defined by the user, in the same manner as shown in the example provided of Chen's model [3], in section 3.2. Huarng [19], was probably the first researcher to focus on the interval partition aspect. In [19], Huarng proposed the distribution- and average-based length approaches to determine the lengths of intervals. Furthermore, the study conducted by Huarng in [19], was the first to investigate the influence of interval lengths on forecast results. Other examples of formalized approaches to interval partitioning can be found in [4,10,29].

A common factor shared by the models published in [4,10,19,29], is that interval lengths are determined independently of forecast accuracy. In contrast, Chen/Chung [15] apply a somewhat different strategy where they exploit genetic algorithm (GA) to tune interval lengths in order to improve forecast accuracy. A similar study is published by Kuo et al in [30], where particle swarm optimization (PSO) is exploited in an analogous manner. The studies published in [15] and [30] are highlighted in this project because they have presented the best results currently published in literature. Both of these models will be used as targets for comparison with the respect to the high order model presented in later sections.

Other studies can also be found which both deal with the fuzzy relationship aspect and the interval partition aspect, such as the ones presented in [8,18,33]. The current study belongs to this latter category of projects, as it sets out to develop new and hopefully better ways of creating interval partitions and relational computations.

3.4 Conclusion

Especially two issues seem to be the primary focus of current research. The first is the selection of interval partitions (i.e. the length and number of intervals). The second is the formulation of fuzzy relationships. Both of these factors highly influence forecast accuracy and thus are considered central to FTS. It has been found that current research efforts are united by one common goal: to enhance consistency between forecast rules and the data they derive from. This implies that performance of different FTS models by tradition is evaluated under known conditions. In plain words, it means that forecast rules are validated using the same old data they originate from, rather than validating them on future datasets (see computations of forecasts in section 3.2). To make this project comparable with those of others, we will follow the same principle in the evaluation phase in section 5.

High order models are highlighted in this context, because they are among most accurate models found in literature, and thus are selected as targets for comparison. The findings of this related work study has lead to the identification of the following key problems with regards to high order models: (1) there is a lack of consistency between forecast rules and the data they represent; (2) forecast accuracy is sensitive to selected interval partitions; (3) data becomes underutilized as the model's order increases. In (3), the underutilization of data manifests itself in two ways. First the number forecast rules (fuzzy relationships) reduces as the order increases. Second, the combination of patterns (fuzzy sets) to be matched with future patterns increases with order increments. This, in turn, reduces the probability of finding equivalent pattern combinations in future time series data. Solving the problems (1)-(3) is the primary objective of this project.

A secondary objective is to further improve the trapezoid fuzzification algorithm proposed by Cheng et al in [4]. This objective is motivated by the need for an algorithm capable of generating trapezoidal fuzzy numbers (or intervals) automatically, based on the characteristics in data.

4 Introducing a Modified Fuzzy Time Series Model

In the following sections a modified FTS model is presented. First we will discuss the data fuzzification part. A novel fuzzification algorithm based on the trapezoid fuzzification approach [4] is be presented and evaluated. Next a novel approach to defuzzify forecasted output is presented based on PSO and aggregation. Finally, in section 5, the proposed FTS model will be evaluated by comparing it to other related developments.

4.1 Algorithm Overview

Before elaborating on the details of the forecasting model presented in here, we will initially provide an overview of the algorithmic process. The overall structure of the algorithm discussed in the following sections is depicted in figure 8.

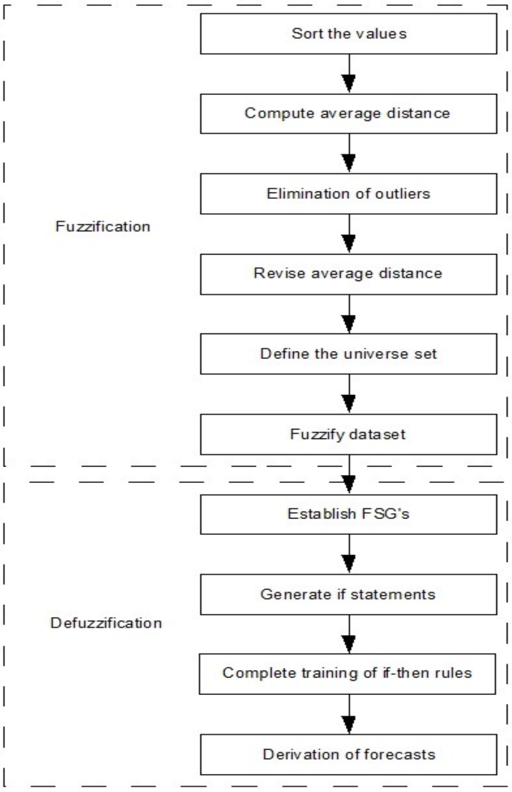

Figure 8. Overall algorithm structure.

The proposed model proposed is divided into two main components, fuzzification and defuzzification. Both of these main components are decoupled which implies that they can be integrated independently with other alternatives. For example, the fuzzification module can be integrated with Chen's first order model [3] or Chen's high order model [31]. This will be demonstrated in section 4.3. The defuzzification module can be integrated with other fuzzification algorithms, such as the ones published in [10,15,18,19].

The fuzzification module can be further decomposed into a six-step process where the first four steps are data preprocessing functions. The fuzzification task itself comprises the last two steps. When data has been fuzzified, it is further processed by the defuzzification module. During the defuzzification phase, data is grouped into patterns which are converted into corresponding if-statements. The if-rules are trained individually to match the data they represent. When training is completed, data is defuzzified by matching the if-then rules with equivalent patterns in the dataset.

4.2 Fuzzifying Historical Data

The fuzzification algorithm (FA) proposed here generates a series of trapezoidal fuzzy sets from a given dataset and establishes associations between the values in the dataset and the fuzzy sets generated. It is inspired by the trapezoid fuzzification approach proposed by Cheng et al in [4]. They introduced an approach where the crisp intervals, generally defined by the user at the initial step of FTS, are replaced with trapezoidal fuzzy sets with overlapping boundaries. This overlap implies that a value may belong to more than one set. If a value belongs to more than one set, it is associated to the set where its degree of membership is highest. The FA introduced here follows the same principles but differs from the approach described by Cheng et al [4] by performing automatically the calculation of the fuzzy intervals/sets. The fuzzification approach published in [4], requires the user to specify the number of sets. This is an undesirable requirement in situations where multiple forecasting problems need to be solved. For example, a grocery store may need forecast information related to thousands of products. The proposed algorithm aims to solve this problem by determining the number of sets on basis of the variations in data.

Another aspect this algorithm attempts to capture, is the notion of a non-static universe set. Whenever values are encountered which fall outside the boundaries of the current universe set, the universe set has to augment accordingly. This aspect, in particular, has not received much attention in current publications. The most likely reason for this is that current modalities rely on the assumption of predetermined outcomes (see section 3.4), and therefore, no revisions of the universe set are required. In real life situations though, future outcomes are rarely known. The basic idea of the algorithm described in the following paragraphs, is to repeat the fuzzification procedure when the dataset is updated. The proposed procedure can be described as a six-step process:

Step 1: Sort the values in the current dataset in ascending order.

Step 2: Compute the average distance between any two consecutive values in the sorted dataset and the corresponding standard deviation.

Step 3: Eliminate outliers from the sorted dataset.

Step 4: Compute the revised average distance between any two remaining consecutive values in the sorted dataset.

Step 5: Define the universe of discourse.

Step 6: Fuzzify the dataset using the trapezoid fuzzification approach [4].

First the values in the historical dataset are sorted in ascending order. Then the average distance between any two consecutive values in the sorted dataset is computed and the corresponding standard deviation. The average distance is given by the equation:

$$AD(x_i \ldots x_n) = \frac{1}{n-1} \sum_{i=1}^{n-1} \left| x_{p(i)} - x_{p(i+1)} \right|, \tag{48}$$

where p is permutation that orders the values ascendantly: $x_{p(i)} \leqslant x_{p(i+1)}$. The standard deviation is computed as

$$\sigma_{AD} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - AD)^2} \tag{49}$$

Both the average distance and standard deviation are used in step 3 to define outliers in the sorted dataset. Outliers are values which are either abnormally high or abnormally low. These are eliminated from the sorted dataset, because the intention here is to obtain an average distance value free of distortions. An outlier, in this context, is defined as a value less than or larger than one standard deviation from average. After the elimination process is completed, a revised average distance value is computed for the remaining values in the sorted dataset, as in step 2. The revised average distance, obtained in step 4, is used in step 5 and 6 to partition the universe of discourse into a series of trapezoidal fuzzy sets. Basically, the intention is to create a series of trapezoidal approximations which capture the generic nature of data as closely as possible, in the sense that we neither want the spread of individual functions to be to narrow or to wide.

In step 5, the universe of discourse is determined. Its lower and upper bound is determined by locating the largest and lowest values in the dataset and augment these by: (1) subtracting the revised average distance from the lowest value and (2) adding the revised average distance to the highest value. More formally, if $D_{\max}$ and $D_{\min}$ are the highest and lowest values in the dataset, respectively, and AD_R is the revised average distance, the universe of discourse U can be defined as $U = [D_{\min} - AD_R, D_{\max} + AD_R]$.

When the U has been determined, fuzzy subsets can be defined on U. Since the subsets are represented by trapezoidal functions, the membership degree, for a given function μ_A and a given value x, is obtained by equation 11.

$$\mu_A = \begin{cases} \frac{x-a_1}{a_2-a_1}, & a_1 \le x \le a_2 \\ 1, & a_2 \le x \le a_3 \\ \frac{a_4-x}{a_4-a_3}, & a_3 \le x \le a_4 \\ 0, & otherwise. \end{cases}$$

Prior to the fuzzification of data, we need to know the number of subsets to be defined on *U*. The number of sets, *n*, is computed by

$$n = \frac{R-S}{2S}, \tag{50}$$

where *R* denotes the range of the universe set and *S* denotes the segment length. Equation 50 originates from the fact that we know the following about *S*:

$$S = \frac{R}{(2n+1)}. \tag{51}$$

The range, *R*, is computed by

$$R = UB - LB, \tag{52}$$

where *UB* and *LB* respectively denote the upper bound ($D_{\max} + AD_R$) and lower bound ($D_{\min} - AD_R$) of *U*. The segment length, *S*, equals the average revised distance AD_R which in turn constitutes the length of left spread (*ls*), core (*c*) and right spread (*rs*) of the membership function (see figure 9). That is, $ls = AD_R$, $c = AD_R$ and $rs = AD_R$.

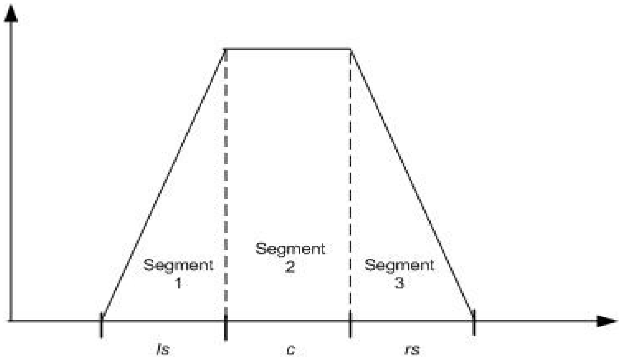

Figure 9. The segments of a trapezoidal fuzzy number.

In short, the task here is to decide how many fuzzy sets to generate when the length of each segment, *S*, and the range, *R*, are known. When the number of sets has been computed, the sets can be defined on *U* and data can be fuzzified which completes the final step of the algorithm.

In the following example, we will fuzzify the first four years of student enrollment in Alabama University. The respective values to be fuzzified are 13055, 13563, 13867 and 14696 (see table 5).

Because the sequence is already in ascending order, the sorting part is omitted. The average distance and the standard deviation are respectively computed as

$$AD=\frac{|13055-13563|+|13563-13867|+|13867-14696|}{3}=\frac{508+304+829}{3}=547$$

and

$$\sigma_{AD}=\sqrt{\frac{(508-547)^2+(304-547)^2+(829-547)^2}{3}}=216.1\approx 216.$$

Next, possible outliers are eliminated. Recall that outliers include the values less than or larger than one standard deviation from AD. This means only the values satisfying the condition:

$$547-216\leqslant x\leqslant 547+216,$$

are taken into consideration when computing the revised average distance. In this case, only one of the three values satisfy the above condition, namely 508. Thus the revised average distance, AD_R, and the segment length, *S*, equals 508. At this point, step 1 - 4 are completed. Prior to defining the universe set *U*, we need to determine the lower bound (*LB*) and the upper bound (*UP*) of *U*. Following equation 52, *LB* and *UP* are computed as

$$LB = 13055 - 508 = 12547$$

$$UB = 14696 + 508 = 15204.$$

Hence *U* = [12547, 15204]. The range, *R*, is computed as difference between *UB* and *LB*. Hence we get 15204 - 12547 = 2657. Finally the number of sets, *n*, is computed as

$$n=\frac{2657-508}{2\cdot 508}=2.12\approx 2.$$

Knowing the universe of discourse and the parameters of *N*, *R* and *S*, the fuzzy sets are generated as shown in figure 10 and table 11.

Fuzzy set	Trapezoidal fuzzy number (*a*, *b*, *c*, *d*)	Crisp interval
A_1	(12547,13055,13602,14149)	u_1 = [13055,13602]
A_2	(13602,14149,14696,15204)	u_2 = [14149,14696]

Table 11. Fuzzifying the first four years of enrollment.

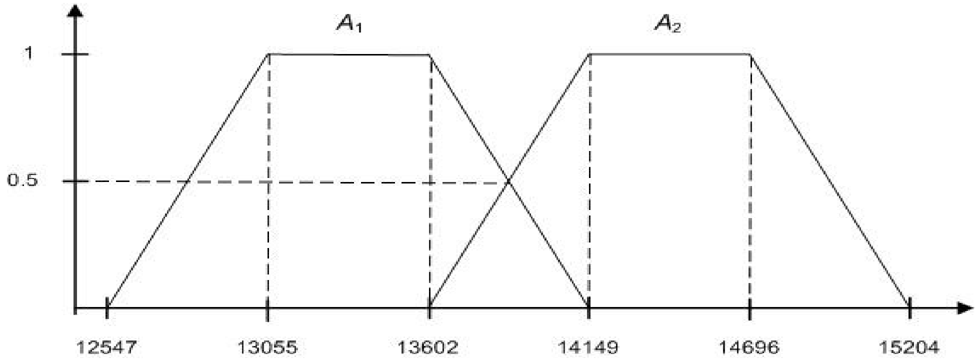

Figure 10. Generated membership functions.

Note the difference between the points *a*, *b*, *c* and *d*, in the fuzzy number A_1 and A_2, in table 11 and figure 10, is not exactly 508. This is because the implemented algorithm adapts the segment length, such that the lowest value in the dataset always appears as the left bound in the first crisp interval, and the highest value in the dataset always appears as the right bound in the last crisp interval. From table 11and figure 10 it can be seen that the lowest of the four values (i.e. 13055), appears as the lower bound of the first crisp interval, u_1, and the highest value (i.e. 14696), appears as the upper bound in the second crisp interval, u_2. Normally these values cannot be matched precisely without adjusting the segment length, due to rounding errors occurring as a result of equation 48 and 50.

Nonetheless, we are now able to fuzzify the first four historical enrollments according to membership functions A_1 and A_2, defined by:

$$A_1 = \begin{cases} 0, & x < 12547 \\ \dfrac{x - 12547}{13055 - 12547}, & 12547 \le x \le 13055 \\ 1, & 13055 \le x \le 13602 \\ \dfrac{14149 - x}{14149 - 13602}, & 13602 \le x \le 14149 \\ 0, & x > 14149. \end{cases}$$

and

$$A_2 = \begin{cases} 0, & x < 13602 \\ \dfrac{x - 13602}{14149 - 13602}, & 13602 \le x \le 14149 \\ 1, & 14149 \le x \le 14696 \\ \dfrac{15204 - x}{15204 - 14696}, & 14696 \le x \le 15204 \\ 0, & x > 15204. \end{cases}$$

Note the intervals overlap so more than one interval may be met. For example, the enrollment for year 1973 is 13867. This value meets both membership functions. The membership degree in A_1 is $0.5155 \approx 0.52$, and in A_2, it is $0.4845 \approx 0.48$. Hence the enrollment for 1973 is fuzzified as A_1. A special case occurs when the membership degree is 0.5, as this implies a that value has the same membership status in two different sets. In such cases, the respective value is associated to both A_1 and A_2.

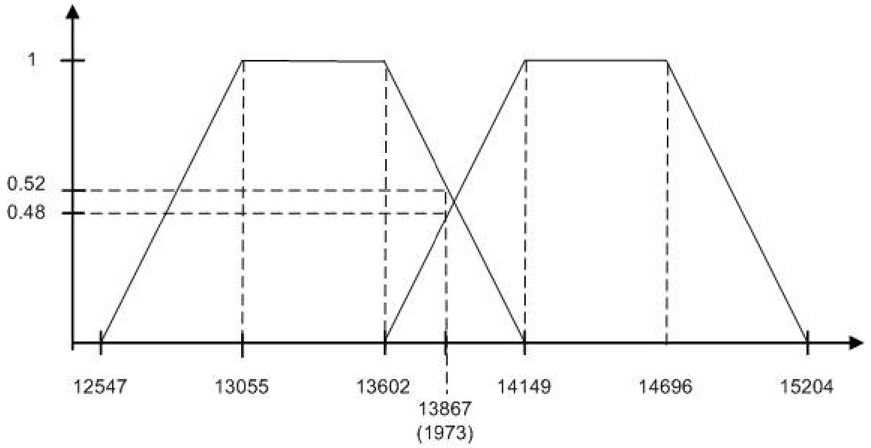

Figure 11. Fuzzifying year 1973.

Year	Enrollment	Fuzzy set	Membership degree
1971	13055	A_1	1
1972	13563	A_1	1
1973	13867	A_1	0.52
1974	14696	A_2	1

Table 12. Fuzzified enrollments 1971 - 1974.

By processing the entire enrollment dataset from table 4, the resultant trapezoidal sets are as shown in table 13. A complete overview of the fuzzified enrollments is shown in table 14.

Fuzzy set	Fuzzy number
A_1	(12861,13055,13245,13436)
A_2	(13245,13436,13626,13816)
A_3	(13626,13816,14007,14197)
A_4	(14007,14197,14388,14578)
A_5	(14388,14578,14768,14959)
A_6	(14768,14959,15149,15339)
A_7	(15149,15339,15530,15720)
A_8	(15530,15720,15910,16101)
A_9	(15910,16101,16291,16482)
A_{10}	(16291,16482,16672,16862)
A_{11}	(16672,16862,17053,17243)
A_{12}	(17053,17243,17433,17624)
A_{13}	(17433,17624,17814,18004)
A_{14}	(17814,18004,18195,18385)
A_{15}	(18195,18385,18576,18766)
A_{16}	(18576,18766,18956,19147)
A_{17}	(18956,19147,19337,19531)

Table 13. Generated fuzzy sets by processing the enrollment data from 1971 - 1992.

Year	Enrollment	Fuzzy Set
1971	13055	A_1
1972	13563	A_2
1973	13867	A_3
1974	14696	A_5
1975	15460	A_7
1976	15311	A_7
1977	15603	A_7
1978	15861	A_8
1979	16807	A_{11}
1980	16919	A_{11}
1981	16388	A_{10}
1982	15433	A_7
1983	15497	A_7
1984	15145	A_6
1985	15163	A_6
1986	15984	A_8
1987	16859	A_{11}
1988	18150	A_{14}
1989	18970	A_{16}
1990	19328	A_{17}
1991	19337	A_{17}
1992	18876	A_{16}

Table 14. Fuzzifying annual enrollments.

Generally it is assumed that the fuzzy sets, $A_1, A_2, \ldots, A_n$, individually represent some linguistic value. With 17 intervals, however, linguistic values may not make much sense. In the model proposed here, this issue is ignored because linguistic values generally do not serve any purpose in FTS what so ever - although they may be useful in certain applicative contexts.

4.3 Evaluating the Proposed Fuzzification Algorithm

In the following section, we're going to evaluate the proposed FA by applying it directly to Chen's model [3,31] for different orders. The first experiment is apply the algorithm to Chen's first order model [3] and compare performance with the one in [4], where the authors also apply their algorithm directly to Chen's model. Next, performance will be evaluated for different model orders by comparing results to the ones reported by Chen in [31].

To evaluate performance across models, the mean squared error (MSE) and mean absolute percentage error (MAPE) are used as performance measures. The respective measures are defined by the equations

$$MAPE = \frac{1}{n} \sum_{t=1}^{n} \frac{\lfloor forecast_t - actual_t \rfloor}{actual_t} \times 100 \tag{53}$$

and

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (forecast_i - actual_i)^2 . \tag{54}$$

From fuzzified data in table 14, we get the following first order relationships:

$A_1 \rightarrow A_2$	$A_8 \rightarrow A_{11}$	$A_6 \rightarrow A_8$
$A_2 \rightarrow A_3$	$A_{11} \rightarrow A_{11}$	$A_{11} \rightarrow A_{14}$
$A_3 \rightarrow A_5$	$A_{11} \rightarrow A_{10}$	$A_{14} \rightarrow A_{16}$
$A_5 \rightarrow A_7$	$A_{10} \rightarrow A_7$	$A_{16} \rightarrow A_{17}$
$A_7 \rightarrow A_7$	$A_7 \rightarrow A_6$	$A_{17} \rightarrow A_{17}$
$A_7 \rightarrow A_8$	$A_6 \rightarrow A_6$	$A_{17} \rightarrow A_{16}$

Table 15. First order relationships.

Moreover, from the data in table 15, we get the following first order FLRG's:

Group 1:	$A_1 \rightarrow A_2$	Group 7:	$A_{11} \rightarrow A_{10}, A_{11}, A_{14}$
Group 2:	$A_2 \rightarrow A_3$	Group 8:	$A_{10} \rightarrow A_7$
Group 3:	$A_3 \rightarrow A_5$	Group 9:	$A_6 \rightarrow A_6, A_8$
Group 4:	$A_5 \rightarrow A_7$	Group 10:	$A_{14} \rightarrow A_{16}$
Group 5:	$A_7 \rightarrow A_6, A_7, A_8$	Group 11:	$A_{16} \rightarrow A_{17}$
Group 6:	$A_8 \rightarrow A_{11}$	Group 12:	$A_{17} \rightarrow A_{16}, A_{17}$

Table 16. First order FLRG's.

To defuzzify forecasted output, we can use the centroid, given by equation 13. For a symmetrical fuzzy number $A_i = (a, b, c, d)$, this computation can be reduced to finding the midpoint of the crisp interval of u_i, given by $[b, c]$. Based on the FLRG's shown in table 16, forecast results are derived as shown in table 17.

Year	**Enrollment**	**Chen** [3]	**Cheng et al** [4]	**Proposed FA**
1971	13055	-	-	-
1972	13563	14000	13531	14230
1973	13867	14000	13912	14230
1974	14696	14000	14673	14230
1975	15460	15500	15435	15541
1976	15311	16000	15435	15541
1977	15603	16000	15435	15541
1978	15861	16000	15435	16196
1979	16807	16000	16958	16196
1980	16919	16833	17211	16196
1981	16388	16833	17211	17507
1982	15433	16833	15435	16196
1983	15497	16000	15435	15541
1984	15145	16000	15435	15541
1985	15163	16000	15435	15541
1986	15984	16000	15435	15541
1987	16859	16000	16958	16196
1988	18150	16833	17211	17507
1989	18970	19000	18861	18872
1990	19328	19000	19242	18872
1991	19337	19000	19052	18872
1992	18876	19000	19052	18872
		-	-	-
MSE		407507	261162	119096
MAPE		3.11%	2.66%	1.42%

Table 17. Comparing forecast results when order = 1.

Forecasted results are calculated according to the same principles shown earlier in section 3.2, in step 6. Referring to table 17, we see that forecast error is reduced when applying the proposed FA to Chen's first order model, since the FA case has the lowest MSE and MAPE vis-à-vis the other models referred to in the table.

In the second experiment, the FA has been applied directly to Chen's high order model [31] for different orders. Experimental results, in form of MSE and MAPE, are presented in table 18.

Order	**MSE** **Chen** [31]	**MSE** **proposed** **FA**	**MAPE** **Chen** [31]	**MAPE** **proposed** **FA**
2	89093	10787	1.62%	0.56%
3	86694	10543	1.56%	0.54%
4	89376	11099	1.57%	0.56%
5	94536	11715	1.65%	0.58%
6	98215	11486	1.68%	0.57%
7	104056	10371	1.74%	0.53%
8	102179	10960	1.70%	0.55%
9	102789	10049	1.68%	0.52%

Table 18. Comparing results for higher orders.

Again, we see better results are obtained when using the proposed FA vis-à-vis the interval partition used by Chen in [31]. We will briefly illustrate how some calculations are derived for the second order case. The second and third order FLRG's, obtained from table 14, are shown in table 19 and 20, respectively.

Group 1:	$A_1, A_2 \rightarrow A_3$	Group 6:	$A_7, A_8 \rightarrow A_{11}$	Group 12:	$A_6, A_6 \rightarrow A_8$
Group 2:	$A_2, A_3 \rightarrow A_5$	Group 7:	$A_8, A_{11} \rightarrow A_{11}$	Group 13:	$A_6, A_8 \rightarrow A_{11}$
Group 3:	$A_3, A_5 \rightarrow A_7$		$A_8, A_{11} \rightarrow A_{14}$	Group 14:	$A_{11}, A_{14} \rightarrow A_{16}$
Group 4:	$A_5, A_7 \rightarrow A_7$	Group 8:	$A_{11}, A_{11} \rightarrow A_{10}$	Group 15:	$A_{14}, A_{16} \rightarrow A_{17}$
Group 5:	$A_7, A_7 \rightarrow A_7$	Group 9:	$A_{11}, A_{10} \rightarrow A_7$	Group 16:	$A_{16}, A_{17} \rightarrow A_{17}$
	$A_7, A_7 \rightarrow A_8$	Group 10:	$A_{10}, A_7 \rightarrow A_7$	Group 17:	$A_{17}, A_{17} \rightarrow A_{16}$
	$A_7, A_7 \rightarrow A_6$	Group 11:	$A_7, A_6 \rightarrow A_6$		

Table 19. Second order FLRG's.

Group 1:	#, $A_1, A_2 \rightarrow A_3$	Group 6:	$A_7, A_7, A_7 \rightarrow A_8$	Group 11:	$A_{11}, A_{10}, A_7 \rightarrow A_7$	Group 16:	$A_6, A_8, A_{11} \rightarrow A_{14}$
Group 2:	$A_1, A_2, A_3 \rightarrow A_5$	Group 7:	$A_7, A_7, A_8 \rightarrow A_{11}$	Group 12:	$A_{10}, A_7, A_7 \rightarrow A_6$	Group 17:	$A_8, A_{11}, A_{14} \rightarrow A_{16}$
Group 3:	$A_2, A_3, A_5 \rightarrow A_7$	Group 8:	$A_7, A_8, A_{11} \rightarrow A_{11}$	Group 13:	$A_7, A_7, A_6 \rightarrow A_6$	Group 18:	$A_{11}, A_{14}, A_{16} \rightarrow A_{17}$
Group 4:	$A_3, A_5, A_7 \rightarrow A_7$	Group 9:	$A_8, A_{11}, A_{11} \rightarrow A_{10}$	Group 14:	$A_7, A_6, A_6 \rightarrow A_8$	Group 19:	$A_{14}, A_{16}, A_{17} \rightarrow A_{17}$
Group 5:	$A_5, A_7, A_7 \rightarrow A_7$	Group 10:	$A_{11}, A_{11}, A_{10} \rightarrow A_7$	Group 15:	$A_6, A_6, A_8 \rightarrow A_{11}$	Group 20:	$A_{16}, A_{17}, A_{17} \rightarrow$ #

Table 20. Third order FLRG's.

Generally the forecasting part is trivial, except for a few special cases. For example, when forecasting year 1977, an ambiguity occurs. To see this, consider the following second order relationship from table 14, for $t = 1977$:

$$F(t-2),\ F(t-1) \rightarrow F(t) = F(1975),\ F(1976) \rightarrow F(1977) = A_7, A_7 \rightarrow A_7.$$

The above relationship matches group 5 in table 19. But two other identical left hand sides exist for group 5. Therefore, we need to find the corresponding third order relation. Again, for t = 1977, we get the following third order FLRG from table 14:

$$F(t-3), F(t-2),\ F(t-1) \rightarrow F(t) = F(1974),\ F(1975),\ F(1976) \rightarrow F(1977) = A_5, A_7, A_7 \rightarrow A_7.$$

At this point, no ambiguities are found in table 20 for the corresponding FLRG. Hence the forecasted result, $Y(1977)$, is computed as the midpoint of the crisp interval, u_7, in the fuzzy set A_7 = (15149,15339,15530,15720). The computation yields

$$Y(1977) = \frac{15339 + 15530}{2} \approx 15435.$$

In the current example, we have demonstrated how forecast accuracy is influenced by different interval partitions. From my own standpoint, this is an additional drawback of current FTS models. The reason for this is rooted in the assumption that intervals should reflect the same characteristics as the data they represent, as this is more useful from a data analytical perspective. So, although comparative results presented in the current section favour the proposed FA in terms of forecast accuracy, this has not been the primary goal. Rather the goal has been to develop a method which objectively determines interval partitions without requiring any user intervention.

4.4 Defuzzifying Output

In the previous sections, the fuzzification part has been discussed. In the following section we will present a novel approach to defuzzify forecasted output. The defuzzification method presented here comprises the following steps:

Step 1: Establish fuzzy set groups (FSG's).

Step 2: Convert the FSG's into corresponding if statements.

Step 3: Train the if-then rules.

Step 4: Derive forecasts.

Before we go into details with the individual steps, it is important to understand how defuzzified output is computed. First recall from the definition provided in section 2.13, that an n-order fuzzy relationship is denoted as $F(t-n), \ldots, F(t-2), F(t-1) \rightarrow F(t)$, where F represents a fuzzified forecast value at time t. In traditional FTS, it is assumed that the left hand side of the fuzzy relation is fuzzified in the same manner as the right hand side. For example, if F, on the left hand side, represents a trapezoidal set, then F, on the right hand, side represents a trapezoidal set as well.

In the modified version introduced here, this notion has been revised such that $F(t)$ is given by the following defuzzification operator, $Y(t)$, defined by

$$Y(t)=\sum_{i=1}^{n} a_{t-i}\cdot w_i \\ \text{where } w_i\in[0,1] \tag{55}$$

and a_{t-i} denotes the actual value at time t - i. Otherwise stated, the defuzzified output is the weighted sum of the actual values from time $(t-n)$ to $t-1$, where n depends on the time series span. For example, if n = 2, we have

$$Y(t)=(a_{t-1}\cdot w_1)+(a_{t-2}\cdot w_2). \tag{56}$$

One question needing to be addressed is how the defuzzification operator deployed here should be interpreted from a fuzzy logical point of view. The thought here is simply to consider the weights as a fuzzy relationship between past values (inputs) and the future value (output). Each w_i represents the strength of the causal relationship between a given input and some unknown output. The closer w_i is to 1, the stronger the relationship and vice versa.

It has to be stressed that the defuzzification operator introduced here is not an aggregation operator from a traditional point of view, since it does not satisfy all of the basic conditions of aggregation (see Appendix II). The proposed operator has been specifically adapted to solve the problem at hand because none of the other operators discussed earlier have been found useful in this context. Averaging operators, for example, never produce outputs less than the minimum value of arguments or larger than the maximum value of arguments. In the current situation, this requirement is undesirable due to the fact that future demand patterns often fluctuate beyond the boundaries of previous min and max values. To illustrate this, we need to take a closer look at the enrollment data in table 17. For t = 1973 and n = 2, we get, a_{1972} = 13563 and a_{1971} = 13055. Assuming $Y(t)$ is an averaging operator, output is restricted to the interval [13055,13563]. However actual output for t = 1973 is 13867 which is out of reach by any averaging operator. Consider another case for t = 1981 and n = 2. We then get a_{1980} = 16919 and a_{1979} = 16807. If $Y(t)$ is the min operator, we get min(a_{1980}, a_{1979}) = 16807, and, if $Y(t)$ is the max operator, we get max(a_{1980}, a_{1979}) = 16919. But actual output for t = 1981 is 16388 which also is unreachable by any averaging operator. As a consequence, a basic requirement for the defuzzification operator proposed here, is that it covers a broader interval than min and max. A reasonable assumption with regards to the bounds of arguments, a_{t-i}, is that they are within the limits of the defined universe set.

4.4.1 Establishment Fuzzy Set Groups (FSG's)

In conventional FTS, fuzzy relationships are identified immediately after data have been fuzzified. However, in the model presented here, the right hand side of the fuzzy relation is not known until the weights have been determined. So, instead of identifying relationships and establishing FLRG's, we establish fuzzy set groups (FSG's). The purpose of the FSG establishment is to partition historical data into unique sets of sub patterns which subsequently are converted into corresponding if statements. During the first pass of the algorithm, consecutive sets are grouped pairwise. Table 21 show the fuzzified data in table 14 grouped in this manner. Every FSG appears in chronological order.

Label	FSG	Label	FSG
1	$\{A_1, A_2\}$	12	$\{A_7, A_7\}$
2	$\{A_2, A_3\}$	13	$\{A_7, A_6\}$
3	$\{A_3, A_5\}$	14	$\{A_6, A_6\}$
4	$\{A_5, A_7\}$	15	$\{A_6, A_8\}$
5	$\{A_7, A_7\}$	16	$\{A_8, A_{11}\}$
6	$\{A_7, A_7\}$	17	$\{A_{11}, A_{14}\}$
7	$\{A_7, A_8\}$	18	$\{A_{14}, A_{16}\}$
8	$\{A_8, A_{11}\}$	19	$\{A_{16}, A_{17}\}$
9	$\{A_{11}, A_{11}\}$	20	$\{A_{17}, A_{17}\}$
10	$\{A_{11}, A_{10}\}$	21	$\{A_{17}, A_{16}\}$
11	$\{A_{10}, A_7\}$		

Table 21. Establishment of FSG's.

To exemplify the principles of grouping, consider year 1971, 1972 and 1973 which respectively are fuzzified as A_1, A_2 and A_3 (see table 14). The pairwise grouping of sets is carried out in the following order:

$$\{F(t-2), F(t-1)\} = \{A_{i,t-2}, A_{i,t-1}\}.$$

The above group, with two elements, is referred to as a second order FSG. By following this principle, the following two second order FSG's are derived:

$$\{F(1971), F(1972)\} = \{A_1, A_2\} \text{ and } \{F(1972), F(1973)\} = \{A_2, A_3\}.$$

In table 21, these groups are labelled as 1 and 2, respectively.

Ultimately, the goal of grouping sets in this manner is to obtain a series of FSG's free of ambiguities. An ambiguity occurs, in this context, if two or more FSG's contain the same combination of elements/sets - i.e. they are not unique. From table 21, it can be seen that not all FSG's are unique. Note that the FSG's labelled as 5, 6 and 12 are identical, as is the case with 8 and 16. In order to obtain a series of disambiguated FSG's, we extend the ambiguous FSG's to third order FSG's, by including the previous set in the corresponding time series. For a second order FSG, the combination ,$\{F(t-2), F(t-1)\}$, is extended to include $F(t-3)$, so the respective FSG now equals the following third order FSG: $\{F(t-3), F(t-2), F(t-1)\}$. Table 22 shows the extensions of the ambiguous FSG's identified in table 21.

Label	{*F*(*t* - 2), F(*t* - 1)}	*F*(*t* - 3)	Extended FSG {*F*(*t* - 3),*F*(*t* - 2), F(*t* - 1)}
5	$\{A_7, A_7\}$	A_5	$\{A_5, A_7, A_7\}$
6	$\{A_7, A_7\}$	A_7	$\{A_7, A_7, A_7\}$
8	$\{A_8, A_{11}\}$	A_7	$\{A_7, A_8, A_{11}\}$
12	$\{A_7, A_7\}$	A_{10}	$\{A_{10}, A_7, A_7\}$
16	$\{A_8, A_{11}\}$	A_6	$\{A_6, A_8, A_{11}\}$

Table 22. Extending ambiguous FSG's.

The extension process is continued until a unique combination of elements is obtained for each FSG. From table 22, we see that only a single extension is required to obtain a unique combination of elements in this particular case. An updated overview of the FSG's in table 21 is shown in table 23.

Label	FSG	Label	FSG
1	$\{A_1, A_2\}$	12	$\{A_{10}, A_7, A_7\}$
2	$\{A_2, A_3\}$	13	$\{A_7, A_6\}$
3	$\{A_3, A_5\}$	14	$\{A_6, A_6\}$
4	$\{A_5, A_7\}$	15	$\{A_6, A_8\}$
5	$\{A_5, A_7, A_7\}$	16	$\{A_6, A_8, A_{11}\}$
6	$\{A_7, A_7, A_7\}$	17	$\{A_{11}, A_{14}\}$
7	$\{A_7, A_8\}$	18	$\{A_{14}, A_{16}\}$
8	$\{A_7, A_8, A_{11}\}$	19	$\{A_{16}, A_{17}\}$
9	$\{A_{11}, A_{11}\}$	20	$\{A_{17}, A_{17}\}$
10	$\{A_{11}, A_{10}\}$	21	$\{A_{17}, A_{16}\}$
11	$\{A_{10}, A_7\}$		

Table 23. Disambiguated FSG's.

4.4.2 Converting FSG's into if statements

Defuzzified output, $Y(t)$, is obtained by matching historical patterns with a corresponding if-then rule. The if statements are generated on basis of the content of the FSG's. This task is fairly simple as the sequence of elements of each FSG is the same as they appear in time. That is, for any FSG of size *n*, the elements appear in the same sequence as in the corresponding time series:

$$F(t-n), F(t-n+1), \ldots, F(t-1).$$

Each FSG can be therefore easily be transformed into if-then rules of the form:

$$\text{if } (F(t-1)=A_{i,t-1} \wedge F(t-2)=A_{i,t-2} \wedge \ldots \wedge F(t-n+1)=A_{i,t-n+1} \wedge F(t-n)=A_{i,t-n});$$
$$\text{then } w_{1,t-1}=? \wedge w_{2,t-2}=? \wedge \ldots \wedge w_{n-1,t-n+1}=? \wedge w_{n,t-n}=?$$

For practical reasons, the sequence of conditions in the if-statement appear in reversed order compared to their equivalent FSG's. For example, an FSG of the form:

$$\{A_{i,t-2}, A_{i,t-1}\},$$

is converted into an equivalent if-rule of the form:

$$\text{if } (F(t-1)=A_{i,t-1} \wedge F(t-2)=A_{i,t-2}).$$

When a rule is matched, the resultant weights are returned and the forecasted value, $Y(t)$, is computed according equation 55. To illustrate this, suppose we need to find a matching if-then rule when forecasting the enrollment for year 1973. From table 14, we get $F(1971)=A_1$ and $F(1972)=A_2$ for t = 1973. Now, assume the following if-then rule already exists in the current rule base:

$$\text{if } (F(t-1)=A_1 \wedge F(t-2)=A_1);$$
$$\text{then } w_{1,t-1}=0.6488 \wedge w_{2,t-2}=0.3882.$$

The above rule is then matched as:

$$\text{if } (F(1973-1)=A_1 \wedge F(1973-2)=A_1);$$
$$\text{then } w_{1,1972}=0.6488 \wedge w_{2,1971}=0.3882.$$

Using equation 55, the forecasted enrollment for year 1973 is computed as

$$Y(1973)=(13563 \cdot 0.6488)+(13055 \cdot 0.3882)=13867.62 \approx 13868.$$

By processing all of the data in table 23, a series of incomplete if statements are generated as shown in table 24. In order to determine the weights, we utilize PSO (see section 2.12) to train the rules individually to match the data they represent.

Rule	Matching part
1	if $(F(t-1)=A_2 \wedge F(t-2)=A_1)$
2	if $((F-1)=A_3 \wedge F(t-2)=A_2)$
3	if $(F(t-1)=A_5 \wedge F(t-2)=A_3)$
4	if $(F(t-1)=A_7 \wedge F(t-2)=A_5)$
5	if $(F(t-1)=A_7 \wedge F(t-2)=A_7 \wedge F(t-3)=A_5)$
6	if $(F(t-1)=A_7 \wedge F(t-2)=A_7 \wedge F(t-3)=A_7)$
7	if $(F(t-1)=A_8 \wedge F(t-2)=A_7)$
8	if $(F(t-1)=A_{11} \wedge F(t-2)=A_8 \wedge F(t-3)=A_7)$
9	if $(F(t-1)=A_{11} \wedge F(t-2)=A_{11})$
10	if $(F(t-1)=A_{10} \wedge F(t-2)=A_{11})$
11	if $(F(t-1)=A_7 \wedge F(t-2)=A_{10})$
12	if $(F(t-1)=A_7 \wedge F(t-2)=A_7 \wedge F(t-3)=A_{10})$
13	if $(F(t-1)=A_6 \wedge F(t-2)=A_7)$
14	if $(F(t-1)=A_6 \wedge F(t-2)=A_6)$
15	if $(F(t-1)=A_8 \wedge F(t-2)=A_6)$
16	if $(F(t-1)=A_{11} \wedge F(t-2)=A_8 \wedge F(t-3)=A_6)$
17	if $(F(t-1)=A_{14} \wedge F(t-2)=A_{11})$
18	if $(F(t-1)=A_{16} \wedge F(t-2)=A_{14})$
19	if $(F(t-1)=A_{17} \wedge F(t-2)=A_{16})$
20	if $(F(t-1)=A_{17} \wedge F(t-2)=A_{17})$
21	if $(F(t-1)=A_{16} \wedge F(t-2)=A_{17})$

Table 24. Generated if rules in chronological order.

4.4.3 Training the if-then rules with PSO

In the following we are going to provide an example of how PSO is utilized to tune the weights in the defuzzification operator in equation 55. The user defined parameters are set as follows[1]:

- The inertial coefficient, ω, equals 1.4.
- The self confidence and social confidence coefficient, c_1 and c_2, respectively, both equals 2.
- The minimum and maximum velocity is limited to [-0.01,0.01].
- The minimum and maximum position is limited to [0,1].
- The number of particles equals five.

The fitness function employed here is the squared error (SE), defined by

[1] The parameters are selected based experimental results.

$$SE = (forecast_t - actual_t)^2 \tag{57}$$

Basically the idea is to evaluate the aggregated result, $Y(t)$, against the actual outcome at time *t*, and adjust the weights in the defuzzification operator such that the squared error is minimized. By minimizing SE for each *t*, MSE is minimized as well. In the following example, the stopping criteria is defined by setting the minimum SE to 3 and the maximum number of iterations to 500[2].

During the first step of the algorithm, the weights (positions) are initialized. Note we assume the existence of a stronger relationship between actual output and the more recent observations in the time series data. So, if $F(t-1)$ is fuzzified as A_i and $F(t-2)$ as A_j, a stronger relationship is assumed to exist between A_i and $Y(t)$ than between A_j and $Y(t)$. Therefore, relatively higher weights are assigned to the most recent observations when positions are initialized. Applying this approach, w_{t-i} will usually remain larger than w_{t-i+1} at the point of termination. Table 25 and 26 respectively show the initial positions and velocities of all particles for a matching rule *R*.

Particle	**Position 1 (w_1)**	**Position 2 (w_2)**	**SE**
1	0.75	0.5	8,024,473
2	0.75	0.5	8,024,473
3	0.75	0.5	8,024,473
4	0.75	0.5	8,024,473
5	0.75	0.5	8,024,473

Table 25. Initial positions of all particles.

In the example above, rule 1 in table 24 is trained. As can be seen from table 25, the personal best positions are the same for all particles during initialization. Hence the personal best positions equals the global best position for all particles.

Particle	v_1	v_2
1	0.0049	0.0011
2	0.0032	0.0065
3	0.0034	0.0081
4	0.0023	0.0009
5	0.0007	0.0048

Table 26. Randomized initial velocities of all particles.

When all particles and velocities have been initialized, the velocities are updated before positions are incremented. Velocities are updated according to equation 40. The computations yield:

[2] Stopping criteria is determined on basis of experimental results.

$$
\begin{aligned}
v_{1,1} &= (1.4\cdot 0.0049)+2\cdot r_1(0.75-0.75)+2\cdot r_2(0.75-0.75) &&= 0.0069\\
v_{1,2} &= (1.4\cdot 0.0011)+2\cdot r_1(0.5-0.5)+2\cdot r_2(0.5-0.5) &&= 0.0015\\
v_{2,1} &= (1.4\cdot 0.0032)+2\cdot r_1(0.75-0.75)+2\cdot r_2(0.75-0.75) &&= 0.0045\\
v_{2,2} &= (1.4\cdot 0.0065)+2\cdot r_1(0.5-0.5)+2\cdot r_2(0.5-0.5) &&= 0.0091\\
v_{3,1} &= (1.4\cdot 0.0034)+2\cdot r_1(0.75-0.75)+2\cdot r_2(0.75-0.75) &&= 0.0048\\
v_{3,2} &= (1.4\cdot 0.0081)+2\cdot r_1(0.5-0.5)+2\cdot r_2(0.5-0.5) &&= 0.0113\\
v_{4,1} &= (1.4\cdot 0.0023)+2\cdot r_1(0.75-0.75)+2\cdot r_2(0.75-0.75) &&= 0.0032\\
v_{4,2} &= (1.4\cdot 0.0009)+2\cdot r_1(0.5-0.5)+2\cdot r_2(0.5-0.5) &&= 0.0013\\
v_{5,1} &= (1.4\cdot 0.0007)+2\cdot r_1(0.75-0.75)+2\cdot r_2(0.75-0.75) &&= 0.0001\\
v_{5,2} &= (1.4\cdot 0.0048)+2\cdot r_1(0.5-0.5)+2\cdot r_2(0.5-0.5) &&= 0.0067.
\end{aligned}
$$

Positions are incremented according to equation 41. Incremented positions after the first iteration are shown in table 27.

Particle	w_1	w_2	SE
1	0.7549	0.5011	8,488,885
2	0.7532	0.5065	8,767,574
3	0.7534	0.5081	8,907,895
4	0.7523	0.5009	8,269,618
5	0.7507	0.5048	8,438,491

Table 27. The positions of all particles after the first iteration.

After the first iteration, none of the computed SE values in table 27 are less than 8,024,473. Thus no personal best positions nor global best positions are reached at this point. At some point, the stopping criteria is met and the algorithm terminates. The personal best positions of all particles after termination are listed in table 28.

Particle	w_1	w_2	SE
1	0.6738	0.3699	10159
2	0.6854	0.3502	1
3	0.6686	0.3662	325
4	0.6724	0.3482	40597
5	0.6879	0.3383	14522

Table 28. The personal best positions of all particles after termination.

According to table 28, particle 2 has the global best position. Hence the weights associated to rule *R* equals 0.6854 and 0.3502. The pseudo code for the training algorithm is shown on page 57.

PSO algorithm for training of the if-then rules

```
Precondition: a set of untrained if-then rules
Postcondition: a set of trained if-then rules

for all rules $R_{id}$ {
    1. if a matching pattern $id = \{A_{id,t-1} \wedge \ldots \wedge A_{id,t-n-1} \wedge A_{id,t-\mathrm{n}}\}$ is found for rule $R_{id}$ {
        1.1. retrieve actual value, $a_{t-i}$, from dataset, from index $i = 0$ to $n$.
        1.2. initialize global best fitness value as $SE_{global\_best} = +\infty$
for each particle, $p_i$, from index $i = 1$ to $z$ {
        1.3. initialize position, $w_{ij}$, from index $j = 1$ to $n$.
        1.4. initialize velocity, $v_{ij}$, from index $j = 1$ to $n$.
        1.5. compute defuzzified output, $Y(t)_i$, by
             $Y(t)_i = \sum_{j=1}^{n} a_{t-j} \cdot w_{ij}$.
        1.6. compute squared error, $SE_i$, by
             $SE_i = (Y(t)_i - actual_t)^2$.
        1.7. initialize local best fitness value, $SE_{local\_best}$, as
             $SE_{local\_best} = SE_i$
        1.8. initialize local best position, $local\_best_{ij}$, as
             $local\_best_{ij} = w_{ij}$ from $j = 1$ to $n$
        1.9. if $SE_i < SE_{global\_best}$ {
             1.9.1. update global best fitness value, $SE_{global\_best}$, value as
                    $SE_{global\_best} = SE_i$
             1.9.2. update global best position, $global\_best_j$, as
                    $global\_best_j = w_{ij}$ from $j = 1$ to $n$
             }//if
}//for
while stopping criteria is unsatisfied{
        for each particle, $p_i$, from index $i = 1$ to $z$ {
        1.10. update velocity, $v_{ij}$, from $j = 1$ to $n$ by
              $v_{ij} = \omega.\cdot v_{ij} + c_1 \cdot r_1(local\_best_{ij} - w_{ij}) + c_2 \cdot r_2(global\_best_j - w_{ij})$
        1.11. if $V_{\min} > v_{ij}$
              1.11.1. set $v_{ij} = V_{\min}$
        1.12. if $V_{\max} < v_{ij}$
              1.12.1. set $v_{ij} = V_{\max}$
        1.13. update position from index $i = 1$ to $n$ by
              $w_{ij} = w_{ij} + v_{ij}$
        1.14. goto step 1.5.
        1.15. goto step 1.6.
        1.16. if $SE_i < SE_{\mathrm{local\_best}}$ {
              1.16.1. update local best fitness value by
                      $SE_{\mathrm{local\_best}} = SE_i$
              1.16.2. update local best position by
                      $local\_best_{ij} = w_{ij}$ from $i = 1$ to $n$
              }//if
        1.17. if $SE_i < SE_{global\_best}$ {
              1.17.1. update global best fitness value by
                      $SE_{global\_best} = SE_i$
              1.17.2. update global best position by
                      $global\_best_{ij} = w_{ij}$ from $i = 1$ to $n$
              }//if
        }//for
}//while
    2. update then-part of rule $R_{id}$ as
       $(w_{id,t-1} = global\_best_1 \wedge \ldots \wedge w_{id,t-n-1} = global\_best_{n-1} \wedge w_{id,n} = global\_best_n)$
}//for
```

After the weights have been optimized via PSO, the 'blanks' in the then part can be filled. The partially completed if-rules from table 24 are shown in fully completed form in table 29. The fixed parameters supplied to the training algorithm are equivalent to the those listed on page 54.

Label	Matching part	Weights
1	$\text{if}(F(t-1)=A_2 \wedge F(t-2)=A_1)$	then w_1 = 0.6488 and w_2 = 0.3882
2	$\text{if}((F-1)=A_3 \wedge F(t-2)=A_2)$	then w_1 = 0.6586 and w_2 = 0.4102
3	$\text{if}(F(t-1)=A_5 \wedge F(t-2)=A_3)$	then w_1 = 0.667 and w_2 = 0.408
4	$\text{if}(F(t-1)=A_7 \wedge F(t-2)=A_5)$	then w_1 = 0.6395 and w_2 = 0.369
5	$\text{if}(F(t-1)=A_7 \wedge F(t-2)=A_7 \wedge F(t-3)=A_5)$	then w_1 = 0.4411, w_2 = 0.3158 and w_3 = 0.2699
6	$\text{if}(F(t-1)=A_7 \wedge F(t-2)=A_7 \wedge F(t-3)=A_7)$	then w_1 = 0.4638, w_2 = 0.4645 and w_3 = 0.0978
7	$\text{if}(F(t-1)=A_8 \wedge F(t-2)=A_7)$	then w_1 = 0.6695 and w_2 = 0.3967
8	$\text{if}(F(t-1)=A_{11} \wedge F(t-2)=A_8 \wedge F(t-3)=A_7)$	then w_1 = 0.4379, w_2 = 0.3892 and w_3 = 0.2171
9	$\text{if}(F(t-1)=A_{11} \wedge F(t-2)=A_{11})$	then w_1 = 0.1604 and w_2 = 0.8137
10	$\text{if}(F(t-1)=A_{10} \wedge F(t-2)=A_{11})$	then w_1 = 0.5497 and w_2 =0.3798
11	$\text{if}(F(t-1)=A_7 \wedge F(t-2)=A_{10})$	then w_1 = 0.5997 and w_2 =0.3809
12	$\text{if}(F(t-1)=A_7 \wedge F(t-2)=A_7 \wedge F(t-3)=A_{10})$	then w_1 = 0.4151, w_2 = 0.3966 and w_3 = 0.1582
13	$\text{if}(F(t-1)=A_6 \wedge F(t-2)=A_7)$	then w_1 = 0.6194 and w_2 = 0.3731
14	$\text{if}(F(t-1)=A_6 \wedge F(t-2)=A_6)$	then w_1 = 0.7524 and w_2 = 0.302
15	$\text{if}(F(t-1)=A_8 \wedge F(t-2)=A_6)$	then w_1 = 0.3869 and w_2 = 0.704
16	$\text{if}(F(t-1)=A_{11} \wedge F(t-2)=A_8 \wedge F(t-3)=A_6)$	then w_1 = 0.4668, w_2 = 0.3847 and w_3 = 0.2725
17	$\text{if}(F(t-1)=A_{14} \wedge F(t-2)=A_{11})$	then w_1 = 0.654 and w_2 = 0.4212
18	$\text{if}(F(t-1)=A_{16} \wedge F(t-2)=A_{14})$	then w_1 = 0.635 and w_2 = 0.4012
19	$\text{if}(F(t-1)=A_{17} \wedge F(t-2)=A_{16})$	then w_1 = 0.6202 and w_2 = 0.3874
20	$\text{if}(F(t-1)=A_{17} \wedge F(t-2)=A_{17})$	then w_1 = 0.5932 and w_2 = 0.3831
21	$\text{if}(F(t-1)=A_{16} \wedge F(t-2)=A_{17})$	then w_1 = ? and w_2 = ?

Table 29. Generated if-then rules in chronological order.

4.5 Conclusion

This section has presented a modified high order FTS model. Introductory, a novel FA was proposed based on the trapezoid fuzzification approach [4]. The proposed algorithm can be applied to any FTS model incorporating interval partitions. The algorithm is regarded as an improvement of similar work [4] in the sense that fuzzification is carried out automatically. Experimental results indicate that forecast accuracy can improved using the proposed FA although this was not the goal per se. Actually the main intention has been to develop an approach where interval partitions are determined objectively without the need of user intervention. Based on current test results, it is believed this goal has been achieved.

The emphasis of the work presented here has been to improve consistency between forecast rules and the data they derive from. In order to achieve this, a defuzzification operator is proposed ad hoc. In traditional models, output is defuzzified via interval (or fuzzy set) operations, whereas in the proposed model, defuzzified output is a weighted sum of actual values. By utilizing PSO and aggregation, forecast rules can be individually tuned to match the data they represent, regardless of the chosen interval partitions. Overall experimental results are presented in the next section.

5 Experimental Results

In section 4, the concepts of a new FTS model were discussed but we have not yet demonstrated overall model performance in terms of forecast accuracy. The purpose of this section is to evaluate the performance of the proposed model vis-à-vis other related prediction models. Performance is compared by the same principle as previously shown in the thesis, namely by evaluating the performance of forecast rules, using the same dataset they derive from.

5.1 Comparing different FTS models

Year	Actual Enrollment	Chen (order = 3) [31]	Li / Cheng [8]	Sing (order = 3) [32]	Chen/Hsu [17]	Chen/Chung (order = 9) [15]	KUO et al (order = 9) [30]	Proposed model
1971	13055	-	-	-	-	-	-	-
1972	13563	-	13500	-	13750	-	-	-
1973	13867	-	13500	-	13875	-	-	13868
1974	14696	14500	14500	14750	14750	-	-	14696
1975	15460	15500	15500	15750	15375	-	-	15460
1976	15311	15500	15500	15500	15313	-	-	15309
1977	15603	15500	15500	15500	15625	-	-	15602
1978	15861	15500	15500	15500	15813	-	-	15861
1979	16807	16500	16500	16500	16834	16846	-	16806
1980	16919	16500	16500	16500	16834	16846	16890	16919
1981	16388	16500	16500	16500	16416	16420	16395	16390
1982	15433	15500	15500	15500	15375	15462	15434	15434
1983	15497	15500	15500	15500	15375	15462	15505	15497
1984	15145	15500	15500	15250	15125	15153	15153	15143
1985	15163	15500	15500	15500	15125	15153	15153	15163
1986	15984	15500	15500	15500	15938	15977	15971	15982
1987	16859	16500	16500	16500	16834	16846	16890	16859
1988	18150	18500	18500	18500	18250	18133	18124	18150
1989	18970	18500	18500	18500	18875	18910	18971	18971
1990	19328	19500	19500	19500	19250	19334	19337	19328
1991	19337	19500	19500	19500	19250	19334	19337	19336
1992	18876	18500	18500	18750	18875	18910	18882	18875
		-	-	-	-	-	-	-
MSE		86694	85040	76509	5611	1101	234	1
MAPE		1.53	1.53	1.41	0.36	0.15	0.014	0.006

Table 30. Comparing different fuzzy time series models.

Different FTS models are compared in terms of MSE and MAPE in table 30. All of the FTS models referenced in the table, are among those with the highest forecasting accuracy found in literature. The MSE and MAPE of the proposed model is 1 and 0.006, respectively. Both measures are lower than for any of the referenced models in the table. Based on these results, it can be concluded that the proposed model outperforms any existing FTS model in the training phase.

Introductory, in section 1, it was argued that the number forecast rules decreases as the order increases in high order models. This is evident from table 30, when considering the forecasted enrollments by the two models by Chen/Chung [15] and Kuo et al [30], as it can be noted that the first 7 - 8 years of enrollment are not forecasted. Moreover, by increasing the order, additional combinations of patterns (fuzzy sets) have to be matched which reduces the probability of finding equivalent pattern combinations in future data.

5.2 Conclusion

Comparative experiments conducted so far show that the proposed model outperforms its counterparts. However there is not sufficient evidence to conclude whether this is a good forecasting method in general, since this requires more extensive research. Because this method of forecasting is inherently rule based, its practical usefulness highly depends on its abilities to derive matching forecast rules, and consistency of those, under unknown conditions. This, however, has not been the focus area of the current project nor any other related research found in literature. So, as for now, this aspect of FTS remains virtually unexplored.

6 Final Conclusion

This project contributes to current research in two ways. First, a novel approach has been developed which combines aggregation and PSO. By combining these techniques, forecast rules can be individually tuned to match the data they represent, regardless of selected interval partitions. It has been found that the individual tuning of rules reduces the need to increase the model's order to improve forecast accuracy, as opposed to the models recently published by the authors in [15] and [30]. As a consequence, better data utilization is achieved in form of: (1) an increased number of forecast rules; (2) fewer pattern combinations to be matched with future time series data. The second contribution to current research, is a fuzzification algorithm, developed as a byproduct. The algorithm, which is a further improvement of the work published in [4], uses an objective measure to automatically generate interval partitions. Experimental results indicate that forecast accuracy may be improved, using the proposed fuzzification approach. All in all, comparative experiments confirm the proposed model's superiority over its counterparts under known conditions. However true performance under unknown conditions has yet to be confirmed.

7 References

[1] Q. Song and B.S. Chissom, Forecasting enrollments with fuzzy time series - part I, Fuzzy Sets and Systems 54 (1993), pp. 1-9.

[2] Q. Song and B.S. Chissom, Forecasting enrollments with fuzzy time series - part II, Fuzzy Sets and Systems 62 (1994), pp. 1-8.

[3] S.M. Chen, Forecasting enrollments based on fuzzy time series, Fuzzy Sets and Systems 81 (1996), pp. 311-319.

[4] C.H. Cheng, J.R. Chang and C.A. Yeh, Entropy-based and trapezoid fuzzification fuzzy time series approaches for forecasting IT project cost, Technological Forecasting & Social Change 73 (2006), pp. 524-542.

[5] S.R. Sing, A robust method of forecasting based on fuzzy time series, Applied Mathematics and Computation 188 (2007), pp. 472-484.

[6] K.H. Huarng, T.H.K Yu and Y.W. Hsu, A multivariate heuristic model for fuzzy-time series forecasting, Systems, Management and Cybernetics 37 (2007), pp. 836-846.

[7] T.A. Jilani and S.M.A. Burney, A refined fuzzy time series model for stock market forecasting, Statistical Mechanics and its Applications 387 (2008), pp. 2857-2862.

[8] S.T. Li and Y.C. Cheng, Deterministic fuzzy time series model for forecasting enrollments, Computer and Mathematics with Applications 53 (2007), pp. 1904-1920.

[9] C.H. Cheng, J.W. Wang and C.H. Li, Forecasting the number of outpatient visits using a new fuzzy time series based on weighted transitional matrix, Expert Systems with Applications 34 (2008), pp. 2568-2575.

[10] K. Huarng and T.H.K. Yu, Ratio-based lengths of intervals to improve fuzzy time series forecasting, Systems, Management and Cybernetics 36 (2006), pp. 328-340.

[11] S.M. Chen and J.R. Hwang, Temperature prediction using fuzzy time series, Systems, Management and Cybernetics 30 (2000), pp. 263-275.

[12] T.A. Jiliani, S.M.A. Burney and C. Ardil, Multivariate high order fuzzy time series forecasting for car road accidents, Int. Journal of Computational Intelligence 4 (2008), pp. .

[13] S.T. Li, Y.C. Cheng and S.Y. Lin, A fcm-based deterministic forecasting model for fuzzy time series, Computers and Mathematics with Applications 56 (2008), pp. 3052-3063.

[14] R.C. Tsaur, J.C. O Yang and H.F. Wang, Fuzzy relation analysis in fuzzy time series model, Computers and Mathematics with Applications 49 (2005), pp. 539-548.

[15] S.M. Chen and N.Y. Chung, Forecasting enrollments using high-order fuzzy time series and genetic algorithms, Int. Journal of Intelligent Systems 21 (2006), pp. 485-501.

[16] S.M. Chen, J.R. Hwang and C.H. Lee, Handling forecasting problems using fuzzy time series, Fuzzy Sets and Systems 100 (1998), pp. 217-228.

[17] S.M. Chen and C.C. Hsu, A new method to forecast enrollments using fuzzy time series, Int. Journal of Applied Science and Engineering 2 (2004), pp. 234-244.

[18] K. Huarng and H.K. Yu, A dynamic approach to adjusting lengths of intervals in fuzzy time series forecasting, Intelligent Data Analysis 8 (2004), pp. 3-27.

[19] K. Huarng, Effective lengths of intervals to improve forecasting in fuzzy time series, Fuzzy Sets and Systems 123 (2001), pp. 387-394.

[20] J.S. Sullivan and W.H. Woodall, A comparison of fuzzy forecasting, Fuzzy Sets and Systems 64 (1994), pp. 279-293.

[21] H.S. Lee and M.T. Chou, Fuzzy forecasting based on fuzzy time series, Int. Journal of Computer Matemathics 81 (2004), pp. 781-789.

[22] K. Huarng and H.K. Yu, A type 2 fuzzy time series model for stock index forecasting, Physica A 353 (2005), pp. 445-462.

[23] H.K. Yu, Weighted fuzzy time series models for TAIEX forecasting, Physica A 349 (2005), pp. 609-624.

[24] S.T. Li and Y.P. Chen, Natural partioning-based forecasting model fore fuzzy time series, Fuzzy Systems 3 (2004), pp. 1355-1359.

[25] C.C. Tsai and S.J. Wu, Forecasting enrollments with high-order fuzzy time series, Fuzzy Information Processing Society (2000), pp. 196-200.

[26] K. Huarng and T.H.K. Yu, The application of neural networks to forecast fuzzy time series, Physica A 363 (2006), pp. 481-491.

[27] T.L. Chen, C.H. Cheng and H.J. Teoh, High-order fuzzy time series based on multi period adaption model for forcasting stock markets, Physica A 387 (2008), pp. 876-888.

[28] S.T. Li and Y.P. Chen, Natural partitioning based forecasting model for fuzzy time-series, 2004 IEEE Int. Conference on Fuzzy Systems 3 (2004), pp. 1355-1359.

[29] C.H. Cheng, G.W. Cheng and J.W. Wang, Multi-attribute fuzzy time series method based on fuzzy clustering, Expert Systems with Applications 34 (2008), pp. 1235-1242.

[30] I.H. Kuo, S.J. Horng, T.W. Kao, T.L. Lin, C.L. Lee and Y. Pan, An improved method for forecasting enrollments based on fuzzy time series and particle swarm optimization, Expert Systems with Applications 36 (2009), pp. 6108-6117.

[31] S.M. Chen, Forecasting enrollments based on high-order fuzzy time series, Cybernetics and Systems: An Int. Journal 33 (2002), pp. 1-16.

[32] S.R. Sing, A simple time variant method for fuzzy time series forecasting, Cybernetics and

Systems: An Int. Journal 38 (2007), pp. 305-321.

[33] H.K. Yu, A refined fuzzy time-series model for forecasting, Physica A 346 (2004), pp. 657-681.

[34] G.J. Klir and B. Yuan, Fuzzy sets and fuzzy logic: theory and applications, ISBN: 0-13-101171-5, Prentice Hall, 1995.

[35] J. Janzen, Tutorial on fuzzy logic, http://www.iau.dtu.dk/~jj/pubs/logic.pdf (validated October 28 2009).

[36] L.A. Zadeh, Fuzzy sets, Information and Control 8 (1965), pp. 338-353.

[37] D. Dubois and H. Prade, Operations on fuzzy numbers, Int. Journal of Systems Science (1978), pp. 613-626.

[38] T.S. Liou and M. J. Wang, Ranking fuzzy numbers with integral value, Fuzzy Sets and Systems 50 (1992), pp. 247-555.

[39] R. R. Yager, Ranking fuzzy subsets over the unit interval, Proc. 1978 CDC (1978), pp. 1435-1437.

[40] S.N. Sivanandam, S. Sumathi and S.N. Deepa, Introduction to fuzzy logic using MATHLAB, ISBN: 103-540-35780-7, Springer-Verlag, 2007.

[41] M. Detyniecki, Fundamentals on aggregation operators, http://www-poleia.lip6.fr/~marcin/papers/Dety_AGOP_01.pdf (validated October 28 2009).

[42] G. Mayor and E. Trillas, On the representation of some aggregation functions, Proc. of ISMVL (1986), pp. 111-114.

[43] H.L. Larsen, Importance weighted OWA aggregation of multicriteria queries, Proc. of the North American Fuzzy Information Processing Society Conference (1999), pp. 740-744.

[44] H.L. Larsen, Fuzzy knowledge operators: averaging operators (AAUE lecture note), http://aaue.dk/~legind/FL_E2006/FL-04/FL-04%20noter.pdf (validated October 28 2009).

[45] H.L. Larsen, Fuzzy knowledge operators: triangular norm operators (AAUE lecture note), http://aaue.dk/~legind/FL_E2006/FL-03/FL-03%20Noter.pdf (validated October 28 2009).

[46] J. Kennedy and R. Eberhart, Particle swarm optimization, Proc. of IEEE Int. Conference on Neural Network (1995), pp. 1942-1948.

[47] J. Kennedy, R. Eberhart and Y. Shi, Swarm intelligence, ISBN: 1-55860-595-9, Morgan Kaufman, 2001.

[48] Q. Song and B.S. Chissom, Fuzzy time series and its models, Fuzzy Sets and Systems 54 (1993), pp. 269-277.

8 Appendix I

In this section we will show how the multi-argument form of the algebraic sum, in equation 38, is derived as a proof of induction.

Step 1.

First we must show that statement $S_{i=1}^{n}(x_i)_{as} = 1-\prod_{i=1}^{n}(1-x_i)$ is true for the base case, $n=2$. For $n=2$ we get:

$$S_{i=1}^{2}(x_i)_{as} = x_1+x_2-x_1\cdot x_2=1-(1-x_1)(1-x_2) = 1-\prod_{i=1}^{n}(1-x_i).$$

That is, the statement holds for *n* = 2.

Step 2.

Assume that the same statement holds for *n* = *k*. We must then show that the statement is true for *n* = *k* + 1. Now let $y=S_{i=1}^{k}(x_i)_{as} = 1-\prod_{i=1}^{n}(1-x_i)$. Then we get $S_{i=1}^{k+1}(x_i) = y+x_{k+1}-y\cdot x_{k+1}$, or:

$$\begin{aligned}
S_{i=1}^{k+1}(x_i) &= \left(1-\prod_{i=1}^{k}(1-x_i)\right)+x_{k+1}-\left(1-\prod_{i=1}^{k}(1-x_i)\right)\cdot x_{k+1}\\
&= 1-\prod_{i=1}^{k}(1-x_i)+x_{k+1}-\left(x_{k+1}-x_{k+1}\cdot\prod_{i=1}^{k}(1-x_i)\right)\\
&=1-\prod_{i=1}^{k}(1-x_i)+x_{k+1}-x_{k+1}+x_{k+1}\cdot\prod_{i=1}^{k}(1-x_i)\\
&=1-\prod_{i=1}^{k}(1-x_i)+x_{k+1}\cdot\prod_{i=1}^{k}(1-x_i)\\
&=1-\left(\prod_{i=1}^{k}(1-x_i)\right)\cdot(1-x_{k+1})\\
&=1-\prod_{i=1}^{k+1}(1-x_i).
\end{aligned}$$

Since $S_{i=1}^{n}(x_i)_{as} = 1-\prod_{i=1}^{n}(1-x_i)$ applies for *n* = *k* + 1, it remains true for every positive integer *n* by the induction principle which concludes the proof.

9 Appendix II

In this section we will examine whether the defuzzification operator in equation 55 is an aggregation operator in classical fuzzy logic sense. Recall from section 2.11 that an aggregation operator is a real function *h*, mapped over the unit interval:

$$h:[0,1]^n \rightarrow [0,1]$$

which as minimum satisfies the following conditions:

1. $h(0,\ldots,0)=0$ and $h(1,\ldots,1)=1$ (**boundary conditions**);
2. $h(x_1,\ldots,x_n) \leq h(y_1,\ldots,y_n)$, if $x_i \leq y_i$ for all $i \in \mathbb{N}$; (**monotonic increasing**)
3. *h* is continuous with respect to each of its arguments;

To see whether condition 1 holds, we assume the operator in 55 only accepts arguments between 0 and 1. From equation 55 we know that the weights, w_i, are subjected to the condition $0 \leqslant w_i \leqslant 1$. This implies that the largest possible output is obtained when $w_i = 1$ for all $i \in \mathbb{N}$. Therefore we select $w_i = 1$ for all arguments a_i. Now we can easily see that the lower boundary condition is satisfied since $a_1 \cdot 1 + a_2 \cdot 1 + \ldots + a_n \cdot 1 = 0$, if $a_i = 0$ for all $i \in \mathbb{N}$. Regarding the upper boundary condition, it can easily be seen that it holds for the unary case (i.e. $n = 1$), since $a_1 \cdot 1 = 1$, if $a_1 = 1$. However for n ≥ 2, it does not hold since the statement $a_1 \cdot 1 + a_2 \cdot 1 + \ldots + a_n \cdot 1 = 1$ is not true, if $a_i = 1$ for all $i \in \mathbb{N}$. Hence the operator in equation 55 is not an aggregation operator with regards to condition 1. Monotonicity and continuity are trivially satisfied.