FORECASTING METHOD BASED ON HIGH ORDER FUZZY TIME SERIES AND SIMULATED ANNEALING TECHNIQUE

F. Radmehr [1*] & N.S. Gharneh [2]

Department of Industrial Engineering
Amirkabir University of Technology, Iran
[1] f.radmehr@aut.ac.ir, [2] nshams@aut.ac.ir

ABSTRACT

This paper proposes a fuzzy forecasting problem to forecast the Alabama University enrolment dataset. A novel simulated annealing heuristic algorithm is used to promote the accuracy of forecasting. The algorithm enjoys two new neighbourhood search operators called 'subtitle' and 'adjust'. A Taguchi method is also used as an optimisation technique to tune the different parameters and operators of the proposed model comprehensively. The experimental results show that the proposed model is more accurate than existing models.

OPSOMMING

Die navorsing handel oor 'n voorgestelde wasige vooruitskatting vir studente-inskrywing by die Universiteit van Alabama. 'n Nuwerwetse louter heuristiese simulasie-algoritme word gebruik. Die model word onder andere beheer deur 'n Taguchi optimiseringstegniek. Die modelresultate toon 'n verbetering op ander bestaande metodes.

[1] Graduate student, Department of Industrial Engineering, Amirkabir University of Technology, Tehran, Iran.
* Corresponding author.

1. INTRODUCTION

It is indisputable that forecasting plays an important role in our daily life. It is also used in many areas for decision-making, such as stock markets, university enrolments, and weather forecasting.

Various techniques for time series forecasting have been evolved in recent decades. Compared with other models, ARMA and ARIMA-based models are prominent and highly useful. However, they cannot deal with time series vagueness and linguistic terms [1]. In addition, these statistical methods could not perform appropriately on time series with a small amount of data [2]. To deal with such deficiencies, fuzzy time series have been developed and widely applied.

Fuzzy time series (FTS) were first introduced by Song & Chissom [1,3,4]. They developed this model based on fuzzy logic. In recent years, FTS models have attracted the attention of many researchers because of their advantages: better performance in some real forecasting problems [2], dealing with data in linguistic terms [2], and their ability to integrate with heuristic knowledge and models [5].

One of the most critical issues in FTS models is the style of their universe of discourse partitioning, which affects their performance in forecasting [6]. Therefore, one of the main purposes of this study is to propose a new simulated annealing (SA) based model to find the right length of the intervals, thus improving forecasting results. A Taguchi method is applied to tune the parameters of the model.

The structure of this paper is as follows: Section 2 provides a review of the literature related to fuzzy time series models. Section 3 examines the concept of fuzzy time series. The proposed model and its algorithm will be described in Section 4. An evaluation and implementation of the model and its results are presented in Section 5. Section 6 concludes the paper.

2. LITERATURE REVIEW

The concept of a fuzzy set theory was first introduced by Zadeh [7]. Since then, the adoption of fuzzy logic has been extensive. Song & Chissom [1,3,4] presented time variant and time invariant fuzzy time series models, which improved the forecasting approach. Chen [8] simplified these models by using simple mathematical operations, because of their simplicity and better forecasting accuracy. This model became especially popular among scientists.

Chen considered the number 7 to be the suitable number of intervals. This researcher divided the universe of discourse into seven equal intervals, representing the linguistic values of very low, low, slightly low, medium, slightly high, high, and very high. These intervals are the simplest and the easiest to use that do not take into account the type and distribution of data. Chen revisited his idea in 2002, investigating the problem for seven to 14 intervals. He worked with equal intervals in his article, simply changing the numbers. He also presented a high-order fuzzy time series model in this study [9].

Huarng was the first to study the length of intervals, pointing out that length affects forecast accuracy in fuzzy time series, and proposing a method with distribution-based length and average-based length to reconcile this issue [6]. Since then, partitioning the universe of discourse has become one of the important issues in this area. Various studies have been done in this field, some of which this paper introduces.

For example, Jilani [10] investigated fuzzy intervals, dividing the universe of discourse into seven equal parts and sorting them in descending order, based on the number of data in each interval. He then divided the first interval into four parts, the second interval into three, the third interval into two parts, and then used the rest of the intervals as they appeared in the data to forecast the University of Alabama's enrolment data.

Cheng [11] is another researcher to have addressed the universe of discourse partition. He not only used a new heuristic method to determine appropriate fuzzy intervals, but also improved the prediction model. He applied expected adaptation to forecast Alabama enrolments and TAIFEX data. In this model, the intervals are first divided into seven equal parts, and the average data that is in each interval is calculated. Then those intervals in which the data are above average are divided into two parts. This process, like the previous one, attempts to homogenise the intervals and divides the intervals with more data.

Huarng [5] pioneered the application of heuristic knowledge to fuzzy time series models. Integrating heuristic models with FTS models intensified their forecasting performance, making them popular among scientists. Heuristic models have also been used in universe of discourse partitioning in some studies. Lee et al. [12,13] introduced two methods based on fuzzy time series, the genetic algorithm, and simulated annealing heuristics to forecast temperature and the TAIFEX. Chen & Chung [14,15] developed first order and high order fuzzy time series models by using the genetic algorithm for enrolment forecasting. Kuo et al. [16] used a PSO-based model called HPSO to achieve well-tuned intervals forecasting Alabama enrolments. Park et al. [17] studied a two-factor high-order fuzzy time series using the PSO method, applying the model to TAIFEX and KOSPI 200 datasets.

Yolcu et al. [18] proposed a new approach using a single-variable constrained optimisation to determine the ratio for the length of intervals. To find the correlation between stock volume and stock market price, Chu et al. [19] proposed a dual-factor method. Cheng et al. [20] proposed another novel method that incorporated trend weighting into the fuzzy time-series models of Chen and Yu, and applied this method to explore the extent to which the innovation diffusion of ICT products is described using this procedure. Wang & Chen [21] presented a new method to predict temperature and TAIFEX using automatic clustering and two-factor high-order fuzzy time series for temperature prediction. Teoh et al. [22] proposed a hybrid fuzzy time series model using the cumulative probability distribution approach (CPDA) and rough set rule induction techniques. Leu & Lee [23] presented another fuzzy time series model using distance-based fuzzy time series to forecast the exchange rate. Egrioglu et al. [24] proposed a hybrid high-order approach to analyse seasonal fuzzy time series, determining the order of the model using the Box-Jenkins model. Bahrepour et al. [25] developed an adaptive ordered fuzzy time series that employs an adaptive order selection algorithm to compose the rule structure and partition the universe of discourse into unequal intervals based on a fast self-organising strategy. They employed it on the FOREX market. Liu & Wei [26] proposed a model to forecast seasonal data.

The literature shows that quite a lot of work has been done on the effective partitioning of the system. In our view, using simulated annealing (SA) to partition the universe of discourse has been considered once, as discussed above. In this paper, we will use SA-FTS to show how SA is able to increase the performance of the system.

3. FUZZY TIME SERIES

As discussed above, there has been considerable growth in fuzzy time series models in forecasting approaches and decisions in recent years. They have all tried to deal with forecasting problems – and they have all been applied according to different definitions. In what follows the concept and definitions of fuzzy time series are explained.

Definition 1. [1,3,4] Let $Y(t)(t=\ldots, 0, 1, 2, \ldots)$, a subset of real numbers, be the universe of discourse by which fuzzy sets $f_j(t)$ are defined. If $F(t)$ is a collection *of* $f_1(t)$, $f_2(t)$. . . then $F(t)$ is called a fuzzy time-series defined on $y(t)$.
Definition 2. [1,3,4] If there is a fuzzy relationship $R(t\text{-}1,t)$, such that $F(t)=F(t\text{-}1)\times R(t\text{-}1, t)$, where × is an operator, then $F(t)$ is said to be caused by $F(t\text{-}1)$. The relationship between $F(t)$ and $F(t\text{-}1)$ can be denoted by $F(t\text{-}1) \rightarrow F(t)$.

Definition 3. [1,3,4] Suppose $F(t\text{-}1)=A_i$ and $F(t)=A_j$, a fuzzy logical relationship is defined as $A_i \rightarrow A_j$; where A_i is named as the left-hand side of the fuzzy logical relationship and A_j the right-hand side.

Definition 4. [1,3,4] Fuzzy logical relationships with the same fuzzy set on the left-hand side can be further grouped into a fuzzy logical relationship group. Suppose there are fuzzy logical relationships such that:

$A_i \rightarrow A_{j1}$
$A_i \rightarrow A_{j2}$
…

then they can be grouped into a fuzzy logical relationship group $A_i \rightarrow A_{j1}, A_{j2}, \ldots$

Definition 5. [1,3,4] Suppose $F(t)$ is caused by $F(t\text{-}1)$ only, and $F(t) = F(t\text{-}1) \times R(t\text{-}1, t)$. For any t, if $R(t\text{-}1, t)$ is independent of t, then $F(t)$ is named a time-invariant fuzzy time series; otherwise it is a time-variant fuzzy time series.

Definition 6. [1,3,4] In a multivariate fuzzy inference process, an m-factor k-order fuzzy time series is a time series consisting of m factors that are of k^{th} order. For example, a two-factor k^{th} order fuzzy time series with X as the primary and Y as the second factor could be stated as:

$$\begin{aligned} &if\ \ (X_1 = x_1, Y_1 = y_1), (X_2 = x_2, Y_2 = y_2), \\ &\ldots(X_k = x_k, Y_k = y_k) then (X_{k+1} = x_{k+1}) \end{aligned} \tag{1}$$

Chen [8] developed a new model based on the above definitions, and simplified Song & Chissom's model. Chen's model provided a better forecasting performance and better integration with heuristic knowledge, compared with Song & Chissom's model [5,8].

However, Chen's model also has a drawback: it omits fuzzy logic relations repetitiveness. To overcome this drawback, this study uses the frequency weighted method [2]. In this case the fuzzy logic relationship (FLR) that occurred often in the past will likely happen often in the future. The procedure of forecasting with fuzzy time series is described as follows:

Step 1. Define the universe of discourse and intervals:
Let the U_{min} and U_{max} be the minimum and maximum value of historical data, let $U = [u_{\min}, u_{\max}]$ be the universe of discourse. Then U is partitioned into several intervals.

Step 2. Define fuzzy sets based on the intervals, and fuzzify the historical data.
Let U be the universe of discourse, where $U=\{u_1, u_2, u_3, \ldots, u_n\}$. A fuzzy set A_i is defined by:

$$A_i = \{(u_1, z_{A_i}(u_1)), (u_2, z_{A_i}(u_2)), \ldots, (u_n, z_{A_i}(u_n))\} \tag{2}$$

where z_{A_i} is the membership function of fuzzy set A_i, and $z_{A_i}(u_k)$ is the degree of belongingness of u_k in A_i. $z_{A_i}(u_k) \epsilon [0,1];\ 1 \le i \le m; 1 \le k \le n.$

A price or a datum is fuzzified to A_k, if the maximum membership function is under A_k. As in the literature, the present study uses a triangular membership function to define fuzzy sets A_i. To present the procedure here, let the number of intervals be seven. Thus the fuzzy sets are defined by the following equation:

$$\begin{aligned} A_1 &= 1/u_1 + 0.5/u_2 + 0/u_3 + 0/u_4 + 0/u_5 + 0/u_6 + 0/u_7 \\ A_2 &= 0.5/u_1 + 1/u_2 + 0.5/u_3 + 0/u_4 + 0/u_5 + 0/u_6 + 0/u_7 \\ A_3 &= 0/u_1 + 0.5/u_2 + 1/u_3 + 0.5/u_4 + 0/u_5 + 0/u_6 + 0/u_7 \\ A_4 &= 0/u_1 + 0/u_2 + 0.5/u_3 + 1/u_4 + 0.5/u_5 + 0/u_6 + 0/u_7 \\ A_5 &= 0/u_1 + 0/u_2 + 0/u_3 + 0.5/u_4 + 1/u_5 + 0.5/u_6 + 0/u_7 \\ A_6 &= 0/u_1 + 0/u_2 + 0/u_3 + 0/u_4 + 0.5/u_5 + 1/u_6 + 0.5/u_7 \\ A_7 &= 0/u_1 + 0/u_2 + 0/u_3 + 0/u_4 + 0/u_5 + 0.5/u_6 + 1/u_7 \end{aligned} \tag{3}$$

Step 3. Establish fuzzy logical relationships, and group them based on the left hand side. According to Definitions 2 and 3, in first order models the fuzzy relationship F(t-1) →*F(t)* could be denoted by $A_i \rightarrow A_j$, in which A_i and A_j are the corresponding linguistic values of F(t-1) and F(t) respectively. Therefore, the way of evoking all fuzzy logical relationships under *n* order is to find any relationship consisting of the type $F(t-n), F(t-n+1), F(t-n+2), \ldots F(t-1) \rightarrow F(t)$. Then by replacing the corresponding linguistic values, *n* order fuzzy relationships were established as $A_i, A_j, A_k \ldots A_l \rightarrow A_z$.

The FLR with the same left-hand side could establish a FLR group.

$A_i, A_j \rightarrow A_k;$
$A_i, A_j \rightarrow A_l;$
…
$A_j, A_z \rightarrow A_m;$
$A_j, A_z \rightarrow A_n;$
…

⇒

$A_i, A_j \rightarrow A_k; A_l; \ldots;$
$A_j, A_z \rightarrow A_m; A_n; \ldots;$
…

Step 4. Assign weights.
All FLRs could establish a fluctuation matrix based on the repetitiveness of each FLR in the historical data. Equation (4) represents a *1×n* fluctuation matrix for a specific FLR, in which n denotes the number of fuzzy sets and w_i $1 \leq i \leq n$ indicates the frequency of that specific FLR with the right hand side of A_i $1 \leq i \leq n$ in the historical dataset.

$$flactuation\,matrix = \begin{bmatrix} w_1 & w_2 & w_3 & \ldots & w_4 \end{bmatrix} \tag{4}$$

To obtain a frequency weighted matrix, the fluctuation matrix should be standardised by equation (5).

$$W_n(t) = [\frac{w_1}{\sum_{k=1}^{n} w_k} \quad \frac{w_2}{\sum_{k=1}^{n} w_k} \quad \frac{w_3}{\sum_{k=1}^{n} w_k} \quad \ldots \quad \frac{w_4}{\sum_{k=1}^{n} w_k}] \tag{5}$$

Finally, a frequency weighted matrix for each FLR group is calculated. The weight of a FLR is based on the repetitiveness of that FLR in the past.

Step 5. Calculate the forecast values.
To obtain the forecast value at time t, we will use Eq. (6) where $W_n(t-1)$ is a frequency weighted matrix for a corresponding FLR group of time t-1, *and* $L_{df}(t-1)$ is a deffuzified matrix of fuzzy sets midpoints.

$$F(t) = L_{df}(t-1) \times W_n(t-1) \tag{6}$$

4. PROPOSED MODEL

Some studies have shown that intervals of different lengths influence the forecasting accuracy [6,16]. This paper also intends to integrate the SA heuristic with Chen's model to achieve a better performance in forecasting.

4.1 Simulated annealing

Simulated annealing (SA) is an optimisation heuristic that searches the solution space by generating a stochastic neighbourhood. In most heuristics, local optimality is a critical problem. The SA algorithm tries to overcome this problem by accepting bad solutions with a decreasing probability [27,28].

The algorithm starts from a definite initial temperature called (Ts). During the process, temperature (T) decreases gradually by the mechanism of a cooling schedule until it reaches the final temperature (Te). In each iteration the algorithm generates a new solution in the neighbourhood of the current solution. Then by comparing the fitness of the neighbour with the current solution, SA decides whether or not to accept the solution. The algorithm accepts superior neighbours as the current solution; but in the case of inferior solutions, it accepts the solution by the probability that is generated by the Boltzmann function (-*Δ/kT*). In this function, k is the constant part, T is the current temperature, and *Δ* is the difference between the current and the new solution's fitness [27,28].

4.1.1 Encoding scheme

In this section the encoding scheme is described. Let the number of intervals be n, with U_{min},U_{max} as the lower and upper bands of the universe of discourse respectively. The solution X is a vector consisting of n-1 components $X = (x_1, x_2, \dots, x_{n-1})$ where $x_i \geq x_{i-1}$, $1 \leq i \leq n-1$. Each element of solution x_i is called a break-point. Using this solution, n intervals will be produced as follows: int_1=(U_{min}, x_1), int_2=(x_1, x_2), …and int_n=(x_{n-1}, U_{max}). Figure 1 is a graphical representation of a solution with seven unequal intervals.

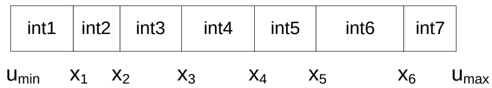

Figure 1: Graphical representation of solution (n=7)

4.1.2 Neighbourhood search structure

The neighbourhood search structure (NSS) is a procedure that generates a new solution that slightly changes the current solution. A variety of NSSs is available and applied to the SA heuristic algorithm. In order to intensify and diversify the search space, two different NSSs are taken into consideration.

- *Substitute operator:* the position of randomly chosen break-points regenerated within range (u_{min},u_{max}). In other words, this procedure replaces a randomly-chosen interval with another newly-generated interval. The mechanism of this NSS is a random number *(r_i)* generated for n-1 break-points. This random number is compared with a constant predefined value *(α)*, if $r_i > \alpha$. The break-point will be replaced by another newly-generated break-point *(y_i)* where $u_{min} < y_i < u_{max}$. Figure 2 is a graphical representation of this procedure in a seven-interval solution where *α=0.5*.

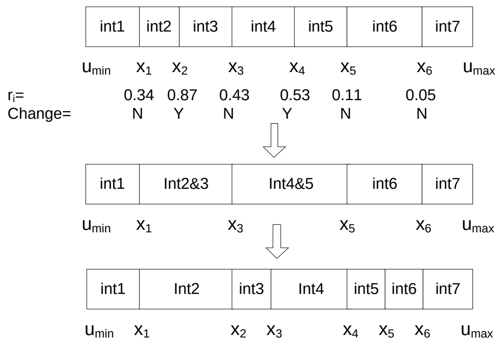

Figure 2: The substitute structure

- *Adjust operator:* the position of a randomly-chosen break-point (x_i) regenerated within the range $(\max(u_{min}, x_{i-1}), \min(u_{max}, x_{i+1}))$. In other words, this procedure aims to adjust a randomly-chosen break-point between the previous and the next break-points. The mechanism of this procedure can be explained thus: β is a predefined value that demonstrates the number of break-points that will be adjusted. So randomly-chosen break-points (x_i) will be replaced with new break-points (y_i), where $\max(u_{min}, x_{i-1}) < y_i < \min(u_{max}, x_{i+1})$. Figure 3 is a sample graphical representation of *Adjust operator* on a seven-interval solution where β=1.

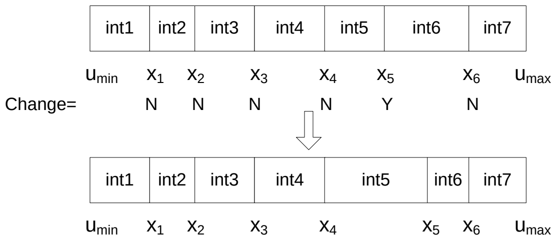

Figure 3: The adjust structure

- *Mix operator*: Finally, these two operators – one of which tries to diversify (substitute), while the other tends to intensify (adjust) the search space – must be combined, with the third operator used as a mixer. The mechanism of this operator is thus: A random number (*r*) is generated and compared with a predefined value such as $\gamma \epsilon$ [0,1]. If $r \leq \gamma$ the *substitute* operator will apply to the current solution; otherwise the *adjust* operator will be applied.

4.1.3 Cooling schedule

The main goal of using a cooling schedule is to control the SA heuristic's behaviour. As explained above, to avoid becoming trapped in local optimums, the SA heuristic accepts a worse solution using a probability that is generated by the Boltzman function. This probability depends on a temperature that decreases steadily by a mechanism called a cooling schedule. There are three types of cooling schedule in the literature: *linear*, *exponential*, and *hyperbolic*. Table 1 presents the formula and the procedure of each mechanism, where T_s, T_e, and *N* are the starting temperature, the stopping temperature, and the number of temperatures between T_s and T_e respectively. By conducting some initial experiments, linear and exponential cooling schedules result in the proposed model's better performance. Following the procedure described in [29], the starting temperature over the range 4 to 10 is appropriate for the problem; the stopping temperature is also determined over the range 0.001 to 0.1.

Table 1: Cooling schedules and their formulas

Cooling schedule	Formula
Linear	$T_i = T_0 - i\frac{T_s - T_e}{N}, \quad i = 1,2,\dots,N$
Exponential	$T_i = \frac{A}{i+1} + B, \; A = \frac{(T_s - T_e)(N+1)}{N} \; B = T_s - A, \quad i = 1,2,\dots,N$
Hyperbolic	$T_i = \frac{1}{2}(T_s - T_e)\left(1 - tgh\left(\frac{10i}{N} - 5\right)\right) + T_e, \quad i = 1,2,,\dots N$

4.1.4 Boltzman function

In order to define the Boltzman function it is necessary first to determine the objective function of the model. The objective of forecasting models is to reduce forecasting errors. There are different ways of calculating error in the literature. One of these methods, which is prevalent in the literature, is the mean square error (MSE) [16]. E(x) indicates the forecasting error of solution X, which is calculated by a definition of MSE as follows:

$$E(x) = \frac{\sum_{i=1}^{n}(F_i - A_i)^2}{n} \tag{7}$$

in which *n* denotes the number of forecasted data, F_i denotes the *ith* forecasted data, and A_i denotes the actual corresponding data. The function $P(E(X^n), E(X^c), T) = \exp(\frac{-\Delta E_i}{T_i})$ is often taken to be a Boltzman function, in which $E(X^n)$, $E(X^c)$, T_i, denotes a new solution error, a current solution error, and the temperature at *ith* iteration respectively.

The MSE performance measure depends on the amount of forecasted data. In order to handle different test problems with different ranges of data, a dimensionless Boltzman function is needed; therefore, ΔE is defined as:

$$\Delta E = \frac{E(X^{new}) - E(X^{current})}{E(X^{current})} \quad (8)$$

The pseudo-code of the proposed SA is described in Figure 4.

```
T=Ts
x= initialisation of the solution          Initialise a solution with randomly-generated break-
                                            points.
Xbest=x
while t≥Te do                               while stopping criterion is not met do
    for iter=1 to itermax do                neighbourhood search in each temperature
        if rand() <γ then                   mix operator
            x^n=substitute (x)              apply substitute operator on current solution
        else
            x^n= adjust (x)                 apply adjust operator on current solution
        end if
        if e(x^n) < e(x) then               acceptance criterion
            x=x^n
            if e(x^n) < e(xbest) then       comparison with best solution
                xbest=x^n
            endif
        else
            if rand() < exp (-Δ/t) then     accepting inferior solution with a probability
                x=x^n
            endif
        endif
    endfor
    t= temperature decreasing               temperature reduction by cooling schedule
end while
```

Figure 4: Pseudo-code of proposed SA

4.2 Parameter calibration

There are parameters in the SA heuristic that need to be tuned, as each of them has an effect on the algorithm's performance. Therefore, designing parameters appropriately makes them more **e**fficient – depending, of course, on the nature of the problem. It is important to know that when a model or problem changes, the parameters must change. According to the literature, many researchers do not apply any tuning method to their problems. This study aims to design an experimental test to tune all parameters.

There are so many approaches to the design of experimental tests. The most widely used – and comprehensive – method is full factorial. However, it is extremely time-consuming in problems with a large number of parameters, such as those included in this study, because the method examines every possible combination of factors. For that reason, fractional factorial experiments (FFE) were developed [30] to reduce the required number of experimental tests. The Taguchi method uses the orthogonal array to investigate the combination of factors and to measure the main effect of each one by designing a small number of experimental tests [31].

Taguchi divides factors into two basic groups: *controllable* and *noise* (uncontrollable) factors.

There is no direct control over noise factors. The Taguchi method is used to minimise the effect of the noise factors, and to set the optimum level of the controllable factors, based on the concept of robustness [32]. The Taguchi method also demonstrates the relative significance of each factor in its impact on the objective function.

A transformation of the repetition data to another value is the measure of variation developed by Taguchi. Because of this transformation, which produces a signal-to-noise ratio (S/N), the Taguchi method is considered to be a robust design [32]. In this method, the terms 'signal' and 'noise' indicate the desirable value (response variable) and undesirable value (standard deviation) respectively. The method tends to maximise the signal-to-noise ratio.

In this method, objective functions have been divided into three main groups: *the-smaller-the-better*, *the-larger-the-better*, and *nominal-is-best*. Since our forecasting objective function is classified as 'the-smaller-the-better', the corresponding signal-to-noise ratio [32] is:

$$\frac{S}{N} ratio = -10 \log_{10}(objective\ function)^2 \tag{9}$$

For this problem, eight factors are considered to be SA control factors: *cooling schedule, starting temperature, stopping temperature, value of desired temperature, value of neighbourhood search in each temperature, substitute operator parameter (α), adjust operator parameter (β)* and *mix operator (γ)*. Table 2 sets out the controllable factors and their related levels.

Table 2: Factors and corresponding levels

Factor		Levels
A	Cooling schedule (C.S) {A(1): exponential, A(2): linear}	2
B	Starting temperature (T_s) {B(1): 4, B(2):8 B(3):10}	3
C	Value of desired temperature (T_n) {C(1):100, C(2):200, C(3):300}	3
D	Neighbourhood search in each temperature (NST) {D(1): 30, D(2):80, D(3):200}	3
E	Stopping temperature (T_e) {E(1):0.001, E(2)=0.01, E(3)=0.1}	3
F	Mix operator parameter (γ) {E(1):0.1, E(2):0.2, E(3):0.3, E(4):0.4, E(5):0.5, E(6):0.6}	6
G	Substitute operator parameter (α) {F(1):0.4, F(2):0.5, F(3):0.6, F(4):0.7 F(5):0.8, F(6):0.9}	6
H	Adjust operator parameter (β) {G(1):1, G(2):2, G(3):3, G(4):4, G(5):5, G(6):6}	6

The cumulative degree of freedom for these factors is 24, so the appropriate orthogonal array for this problem should have at least 24 rows and 8 columns. For the orthogonal array in this case, two modifications need to be done. First, the omission of two columns with three levels is necessary.

Second, a two-level factor of the case (factor A: *cooling schedule*) lacks one level compared with L_{36}. To offset this gap, one optionally selected existing level of the related factor needs to be allotted to that extra level. So the third level of factor A is set as a repetition of level 1. Table 3 shows 36 experimental tests that were designed according to the modified L_{36}.

Taguchi-designed experiments are conducted using a three-order fuzzy time series model on the Alabama University enrolment as a prominent test problem in the area of fuzzy time series. To get more reliable information, each trial is implemented three times. In all the experiments, Matlab 7.1 is used on a PC with a 2.4 GHz Intel Core 2 Duo processor and 4 GB of RAM. In order to compare the experimental tests, relative percentage deviation (RPD) is used as a performance measure, calculated as follows:

$$RPD = \frac{trial_{sol} - min_{sol}}{min_{sol}}.100\% \quad (10)$$

where min_{sol} is the best result obtained for a given test problem among the experimental trials, and $trial_{sol}$ is the objective value (MSE) obtained in an experimental trial.

The results obtained for each experiment were transformed into a S/N ratio. Figure 4 shows the average S/N ratio for each level. As shown in Figure 5, A(1), B(1), C(3), D(3), E(1), H(2) are the optimum levels of factors A, B, C, D, E, H respectively. Determining the optimal level of factors F and G needs more investigation, for which, the RPD needs to be analysed.

The results of the average RPD are shown in Figure 6. This analysis strongly confirmed the optimal levels for factors A, B, C, D, E, and H, and it was found that F(4) and G(4) were the superior levels, with a minimum RPD for factors E and F respectively. Thus the chosen levels are as follows:

- Cooling schedule: exponential, starting temperature: 4.
- Value of desired temperature: 300.
- Neighbourhood search in each temperature: 200.
- Stopping temperature: 0.001.
- Mix operator parameter: 0.4.
- Substitute operator parameter: 0.7.
- Adjust operator parameter: 2.

Table 3: The modified orthogonal array L36

Trial	Levels of control factors							
	A	B	C	D	E	F	G	H
1	A(1)	B(1)	C(1)	D(1)	E(1)	F(1)	G(1)	H(1)
2	A(1)	B(1)	C(2)	D(2)	E(2)	F(1)	G(2)	H(2)
3	A(1)	B(1)	C(2)	D(3)	E(3)	F(2)	G(4)	H(6)
4	A(1)	B(1)	C(3)	D(2)	E(3)	F(3)	G(6)	H(4)
5	A(1)	B(2)	C(1)	D(1)	E(2)	F(3)	G(4)	H(5)
6	A(1)	B(2)	C(2)	D(2)	E(1)	F(6)	G(3)	H(5)
7	A(1)	B(2)	C(2)	D(3)	E(3)	F(5)	G(5)	H(1)
8	A(1)	B(2)	C(3)	D(3)	E(1)	F(2)	G(2)	H(3)
9	A(1)	B(3)	C(1)	D(2)	E(3)	F(4)	G(6)	H(3)
10	A(1)	B(3)	C(1)	D(3)	E(2)	F(5)	G(3)	H(4)
11	A(1)	B(3)	C(3)	D(1)	E(1)	F(4)	G(5)	H(2)
12	A(1)	B(3)	C(3)	D(1)	E(2)	F(6)	G(1)	H(6)
13	A(2)	B(1)	C(1)	D(1)	E(2)	F(2)	G(5)	H(4)
14	A(2)	B(1)	C(2)	D(2)	E(2)	F(6)	G(5)	H(3)
15	A(2)	B(1)	C(3)	D(2)	E(3)	F(5)	G(1)	H(5)
16	A(2)	B(1)	C(3)	D(3)	E(1)	F(3)	G(3)	H(2)
17	A(2)	B(2)	C(1)	D(1)	E(3)	F(1)	G(3)	H(3)
18	A(2)	B(2)	C(2)	D(3)	E(1)	F(4)	G(1)	H(4)
19	A(2)	B(2)	C(3)	D(2)	E(2)	F(4)	G(4)	H(1)
20	A(2)	B(2)	C(3)	D(3)	E(2)	F(1)	G(6)	H(6)
21	A(2)	B(3)	C(1)	D(2)	E(1)	F(5)	G(2)	H(6)
22	A(2)	B(3)	C(1)	D(3)	E(3)	F(6)	G(4)	H(2)
23	A(2)	B(3)	C(2)	D(1)	E(1)	F(2)	G(6)	H(5)
24	A(2)	B(3)	C(2)	D(1)	E(3)	F(3)	G(2)	H(1)
25	A(1)	B(1)	C(1)	D(3)	E(1)	F(6)	G(6)	H(1)
26	A(1)	B(1)	C(1)	D(3)	E(2)	F(4)	G(2)	H(5)
27	A(1)	B(1)	C(2)	D(1)	E(3)	F(4)	G(3)	H(6)
28	A(1)	B(1)	C(3)	D(1)	E(1)	F(5)	G(4)	H(3)
29	A(1)	B(2)	C(1)	D(2)	E(1)	F(3)	G(5)	H(6)
30	A(1)	B(2)	C(1)	D(2)	E(3)	F(2)	G(1)	H(2)
31	A(1)	B(2)	C(2)	D(1)	E(2)	F(5)	G(6)	H(2)
32	A(1)	B(2)	C(3)	D(1)	E(3)	F(6)	G(2)	H(4)
33	A(1)	B(3)	C(2)	D(2)	E(1)	F(1)	G(4)	H(4)
34	A(1)	B(3)	C(2)	D(3)	E(2)	F(3)	G(1)	H(3)
35	A(1)	B(3)	C(3)	D(2)	E(2)	F(2)	G(3)	H(1)
36	A(1)	B(3)	C(3)	D(3)	E(3)	F(1)	G(5)	H(5)

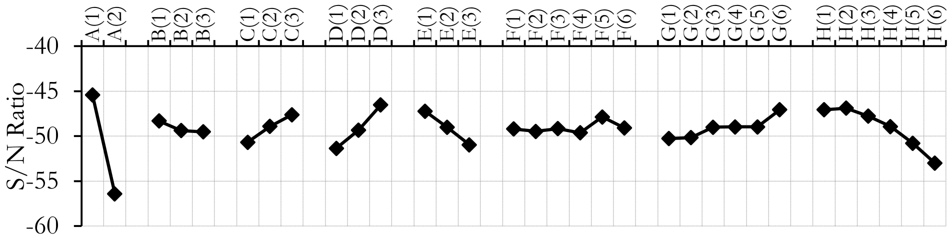

Figure 5: Mean S/N ration for each level of factors

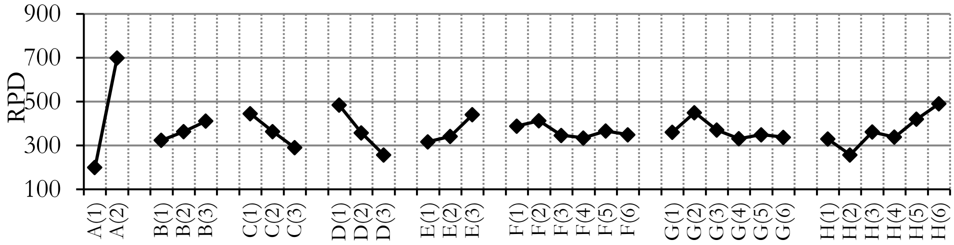

Figure 6: The mean RPD for each level of factors

Table 4: S/N ratio and RPD for each level of factors

Factor and levels	Mean S/N ratio	Mean RPD
A(1)	-45.404	199.9822
A(2)	-56.4162	699.337
B(1)	-48.3167	323.9288
B(2)	-49.3857	363.617
B(3)	-49.5218	411.7555
C(1)	-50.6879	444.7059
C(2)	-48.9248	364.1762
C(3)	-47.6114	290.4192
D(1)	-51.3608	484.5255
D(2)	-49.3264	357.5176
D(3)	-46.537	257.2582
E(1)	-47.2308	317.1252
E(2)	-49.0199	341.3405
E(3)	-50.9734	440.8356
F(1)	-49.2062	388.4628
F(2)	-49.4657	413.2079
F(3)	-49.1662	346.6948
F(4)	-49.6347	334.5277
F(5)	-47.8713	365.9912
F(6)	-49.104	349.7183
G(1)	-50.2645	360.2004
G(2)	-50.1616	449.9538
G(3)	-49.0085	370.5036
G(4)	-48.9786	331.2433
G(5)	-48.9753	349.2642
G(6)	-47.0597	337.4372
H(1)	-47.0746	329.7754
H(2)	-46.8824	257.5625
H(3)	-47.7762	361.9836
H(4)	-48.9362	338.6425
H(5)	-50.7834	420.1407
H(6)	-52.9955	490.4979

To examine the relative significance of the factors in terms of their impact on the objective function, an analysis of variance (ANOVA) was applied to the S/N ratio. The results are given in Table 4. The cooling schedule (A) factor has the largest impact on the objective function with a relative significance of 63.53%. After this factor, the next most important ones were: adjust operator parameter (H), at 10.41%; neighbour search in each temperature (D), 8.89%; stopping temperature (E), 5.17%; and number of desired temperature (C), 3.40%. The remaining factors – B, F, and G – had the least impact on the quality of the objective function.

Table 5: ANOVA for S/N ratio

S.V.	D.F.	S.S.	M.S.	F	P. x	Cu.
A	1	9.71E+02	970.62	330.48	63.53	63.53
B	2	10.9266	5.46	1.86	0.33	63.86
C	2	57.6671	28.83	9.82	3.40	67.26
D	2	141.2271	70.61	24.04	8.89	76.14
E	2	84.572	42.29	14.40	5.17	81.31
F	5	12.1206	2.42	0.83	0.17	81.48
G	5	40.5572	8.11	2.76	1.70	83.18
H	5	173.2995	34.66	11.80	10.41	93.59
Error	11	32.3071	2.94			
Total	35	1.52E+03				

5. EXPERIMENTAL EVALUATION

The proposed model was applied to the Alabama University enrolment data. A comparison of the results from the proposed model, from Kuo et al. [16], and from Chen & Chung [14], based on the different number of intervals and first-order forecasting, is shown in Table 6.

Table 6: Comparison of results from proposed model and others under the different values of interval and first-order forecasting

Methods	Number of intervals						
	8	9	10	11	12	13	14
Proposed model	121052	85777	53115	40814	28962	20745	16643
Chen & Chung [14]	132963	96244	85486	55742	54248	42497	35324
Kuo et al. [16]	119962	90527	60722	49257	34709	24687	22965

The main difference between the three methods is the heuristic algorithms that were used to forecast. Chen & Chung [14] used the genetic algorithm; Kuo et al. [16] used the PSO heuristic; while this study has proposed a method that benefits from the SA heuristic algorithm. This model is well-tuned by the Taguchi orthogonal arrays method. From Table 7 it is clear that the proposed model offers a better performance in achieving the appropriate set of intervals.

In the next step, a comparison is carried out between the proposed model and existing high-order models. The results are given in Table 7. They clearly indicate that the proposed model offers the greatest accuracy among all existing models.

Table 7: Different high order models with seven intervals

Order	Chen [9]	C & C [15]	Singh [33]	Kuo [16]	SA-FTS
2	89093	67834		67123	53533
3	86694	31123	133700	31644	27608
4	89376	32009		23271	20561
5	94539	24948		23534	21426
6	98215	26980		23671	21831
7	104056	26969		20651	18900
8	102179	22387		17106	15196
9	102789	18734		17971	15573

Table 8: Comparison of results from different first-order forecasting models

Year	Actual	Chen [8]	Huarng [5]	C &C [15]	Kuo[16]	SA-FTS
1971	13055					
1972	13563	14 000	14 000	13 714	13 555	13706
1973	13867	14 000	14 000	13 714	13 994	13706
1974	14696	14 000	14 000	14 880	14 711	14749
1975	15460	15 500	15 500	15 467	15 344	15341
1976	15311	16 000	15 500	15 172	15 411	15346
1977	15603	16 000	16 000	15 467	15 411	15346
1978	15861	16 000	16 000	15 861	15 411	15923
1979	16807	16 000	16 000	16 831	16 816	16839
1980	16919	16 833	17 500	17 106	17 140	17046
1981	16388	16 833	16 000	16 380	16 464	16400
1982	15433	16 833	16 000	15 464	15 505	15455
1983	15497	16 000	16 000	15 172	15 411	15346
1984	15145	16 000	15 500	15 172	15 411	15346
1985	15163	16 000	16 000	15 467	15 344	15341
1986	15984	16 000	16 000	15 467	16 018	15923
1987	16859	16 000	16 000	16 831	16 816	16839
1988	18150	16 833	17 500	18 055	18 060	18007
1989	18970	19 000	19 000	18 998	19 014	19059
1990	19328	19 000	19 000	19 300	19 340	19197
1991	19337	19 000	19 500	19 149	19 340	19197
1992	18876	19 000	19 000	19 149	19 014	19059
MSE		407 507	226 611	35 324	22 965	16643

Table 8 contains a comparison of the forecasted enrolments generated by the proposed model and existing first-order fuzzy time series, with different number of intervals, such as Chen [8], Huarng [5], Chen & Chung [15] and Kuo et al. [16]. Three of the models – Chen & Chung [15], Kuo et al. [16], and the proposed model – used first order fuzzy forecast rules and 14 intervals to forecast enrolments.

As the final step, the proposed model was compared with other high-order fuzzy forecasting models such as Chen [9], Singh [33], Chen & Chung [15], and Kuo et al. [16]. The results are shown in Table 9, where two of the models [15, 16] and the proposed model use ninth-order fuzzy rules and 14 intervals to forecast enrolments.

Table 9: Comparison of results for different high order models

Year	Actual data	Singh [33]	Chen [9]	C & C [15]	Kuo [16]	SA-FTS
1971	13 055					
1972	13 563					
1973	13 867					
1974	14 696	14 286	14 500			
1975	15 460	15 361	15 500			
1976	15 311	15 468	15 500			
1977	15 603	15 512	15 500			
1978	15 861	15 582	15 500			
1979	16 807	16 500	16 500	16 846		
1980	16 919	16 361	16 500	16 846	16 890	16890
1981	16 388	16 362	16 500	16 420	16 395	16386
1982	15 433	15 744	15 500	15 462	15 434	15434
1983	15 497	15 560	15 500	15 462	15 505	15497
1984	15 145	15 498	15 500	15 153	15 153	15153
1985	15 163	15 306	15 500	15 153	15 153	15153
1986	15 984	15 442	15 500	15 977	15 971	15986
1987	16 859	16 558	16 500	16 846	16 890	16890
1988	18 150	17 187	18 500	18 133	18 124	18152
1989	18 970	18 475	18 500	18 910	18 971	18972
1990	19 328	19 382	19 500	19 334	19 337	19328
1991	19 337	19 487	19 500	19 334	19 337	19328
1992	18 876	18 744	18 500	18 910	18 882	18877
MSE		133 700	86 694	1101	234	159

All of the experimental results clearly confirmed that the new proposed SA is superior to the other algorithms in its different fuzzy order rules and number of intervals.

6. CONCLUSION

This paper dealt with a new forecasting model based on the simulated annealing (SA) heuristic and fuzzy time series (FTS) to forecast the Alabama University's enrolment dataset. The proposed metaheuristic struck a compromise between intensification and diversification by using two neighbourhood search structures called 'adjust' and 'substitute' operators. To enhance the quality of the simulated annealing, a comprehensive comparison has been made between different parameters of the algorithm to obtain their precise calibration by means of the Taguchi method. To investigate the model's effectiveness in comparison with other models in the literature, the experimental results revealed the superiority of the proposed simulated annealing based model (SA-FTS).

The SA-FTS was tested on Alabama enrolments under one factor condition. It could be applied to other problems with two or more factors, such as temperature prediction and stock indices, in further research.

REFERENCES

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

[2] **Tsaur, R.C., Yang, J.C. & Wang**, **H.F.** 2005. Fuzzy relation analysis in fuzzy time series model, *Computers and Mathematics with Applications,* 49 (4), pp 539-548.

[3] **Song, Q. & Chissom**, **B.S.** 1993. Fuzzy time series & its models, *Fuzzy Sets & Systems,* 54 (3), pp 269-277.

[4] **Song, Q. & Chissom**, **B.S.** 1994. Forecasting enrollments with fuzzy time series – Part II, *Fuzzy Sets & Systems,* 62 (1), pp 1-8.

[5] **Huarng, K.** 2001. Heuristic models of fuzzy time series for forecasting, *Fuzzy Sets & Systems,* 123 (3), pp 369-386.

[6] **Huarng, K.** 2001. Effective lengths of intervals to improve forecasting in fuzzy time series, *Fuzzy Sets & Systems,* 123, pp 387-394.

[7] **Zadeh, L.A.** 1965. Fuzzy sets, *Information & Control,* 8, pp 338-353.

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

[9] **Chen, S.M.** 2002. Forecasting enrollments based on high-order fuzzy time series, *Cybernetics & Systems: An International Journal,* 33, pp 1-16.

[10] **Jilani, T.A.** 2007, Fuzzy metric approach for fuzzy time series forecasting based on frequency density based partitioning, in World Academy of Science, Engineering & Technology Conference.

[11] **Cheng, C.H.** 2008. Fuzzy time-series based on adaptive expectation model for TAIEX forecasting, *Expert Systems with Applications,* 34, pp 1126-1132.

[12] **Lee, L.W., Wang, L.H. & Chen**, **S.M.** 2007. Temperature prediction & TAIFEX forecasting based on fuzzy logical relationships & genetic algorithms, *Expert Systems with Applications,* 33 (33), pp 539-550.

[13] **Lee, L.W., Wang, H.F. & Chen**, **S.M.** 2008. Temperature prediction & TAIFEX forecasting based on high-order fuzzy logical relationships & genetic simulated annealing techniques, *Expert Systems with Applications,* 34 (1), pp 328-336.

[14] **Chen, S.M. & Chung**, **N.Y.** 2006. Forecasting enrollments of students by using fuzzy time series & genetic algorithms, *International Journal of Information & Management Sciences,* 17, pp 1-17.

[15] **Chen, S.M. & Chung**, **N.Y.** 2006. Forecasting enrollments using high-order fuzzy time series & genetic algorithms: Research Articles, *International Journal of Information & Management Sciences,* 21, pp 485-501.

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

[17] **Park, J.I., Lee, D.J., Song, C.K. & Chun**, **M.G.** 2010. TAIFEX & KOSPI 200 forecasting based on two-factors high-order fuzzy time series & particle swarm optimization, *Expert Systems with Applications,* 37, pp 959-967.

[18] **Yolcu, U., Egrioglu, E., Uslu, V.R., Basaran, M.A. & Aladag, C.H.** 2009. A new approach for determining the length of intervals for fuzzy time series, *Applied Soft Computing,* 9 (2), pp 647-651.

[19] **Chu, H.H., Chen, T.L., Cheng, C.H. & Huarng**, **C.C.** 2009. Fuzzy dual factor time-series for stock index forecasting, *Expert Systems with Applications,* 36 (1), pp 165-171.

[20] **Cheng, C.H., Chen, Y.S. & Wu, Y.L.** 2009. Forecasting innovation diffusion of products using trend-weighted fuzzy time-series model, *Expert Systems with Applications,* 36 (2), pp 1826-1832.

[21] **Wang, N.Y. & Chen, S.M.** 2009. Temperature prediction & TAIFEX forecasting based on automatic clustering techniques & two-factors high-order fuzzy time series, *Expert Systems with Applications,* 36 (2), pp 2143-2154.

[22] **Teoh, H.J., Cheng, C.H., Chu, H.H. & Chen, J.S.** 2008. Fuzzy time series model based on probabilistic approach & rough set rule induction for empirical research in stock markets, *Data & Knowledge Engineering,* 67 (1), pp 103-117.

[23] **Leu, Y., Lee, C.P. & Jou, Y.Z.** 2009. A distance-based fuzzy time series model for exchange rates forecasting, *Expert Systems with Applications,* 36 (4), pp 8107-8114.

[24] **Egrioglu, E., Aladag, C.H., Yolcu, U., Basaran, M.A. & Uslu, V.R.** 2009. A new hybrid approach based on SARIMA & partial high order bivariate fuzzy time series forecasting model, *Expert Systems with Applications,* 36 (4), pp 7427-7434.

[25] **Bahrepour, M., Akbarzadeh-T, M.R., Yaghoobi, M. & Naghibi-S, M.B.** 2011. An adaptive ordered fuzzy time series with application on FOREX, *Expert Systems with Applications,* 38 (1), pp 475-485.

[26] **Liu, H.T. & Wei, M.L.** 2010. An improved forecasting method for seasonal time series, *Expert Systems with Applications,* 37 (9), pp 6310-6318.

[27] **Kirkpatrick, S., Gellat, C.D. & Vecchi, M.P.** 1983. Optimization by simulated annealing, *Science,* 220 (4598), pp 671-680.

[28] **Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. & Teller, E.** 1953. Equations of state calculations by fast computing machines, *Journal of Chemical Physics,* 21, pp 1087-1092.

[29] **Crama, Y. & Schyns, M.** 2003. Simulated annealing for complex portfolio selection problems, *European Journal of Operational Research,* 150, pp 546-571.

[30] **Cochran, W.G. & Cox, G.M.** 1992. *Experimental designs,* 2nd edition. Wiley.

[31] **Ross, R.J.** 1989. *Taguchi techniques for quality engineering.* McGraw-Hill.

[32] **Phadke, M.S.** 1989. *Quality engineering using robust design.* Prentice-Hall.

[33] **Singh, S.R.** 2007. A simple method of forecasting based on fuzzy time series, *Applied Mathematics & Computation,* 186, pp 330-339.