Available online at www.isas.org.in/jisas

JOURNAL OF THE INDIAN SOCIETY OF AGRICULTURAL STATISTICS 64(3) 2010 391-398

Application of Genetic Algorithm for Fitting Self-Exciting Threshold Autoregressive Nonlinear Time-Series Model

M.A. Iquebal[1], Himadri Ghosh[2] and Prajneshu[2]*

[1]*Indian Institute of Pulses Research, Kanpur*

[2]*Indian Agricultural Statistics Research Institute, New Delhi*

Received 25 July 2009; Revised 17 April 2010; Accepted 26 April 2010

SUMMARY

The Self-exciting threshold autoregressive (SETAR) nonlinear time-series model is thoroughly studied. This model is capable of describing cyclical data. The Genetic algorithm (GA), which is a powerful stochastic search and optimization procedure motivated by the principles of genetics and natural selection, is employed for estimation of parameters of the model. As an illustration, India's lac export annual data for the period 1901-2001 is considered for building the model. Forecasting for hold-out data for the years 2002-'08 is also carried out. Superiority of GA-methodology over Search algorithm procedure is demonstrated for the data under consideration.

Keywords: Akaike information criterion, Genetic algorithm, India's annual lac export data, Forecasting for hold-out data, SETAR nonlinear time-series model.

1. INTRODUCTION

Time-series data in various fields, like crop sciences, fisheries and agricultural meteorology quite often depict cyclical fluctuations. Some examples of such behaviour are: India's annual lac production / export data, India's summer monsoon rainfall data, and Population-sizes of several fish species having prey-predator type of interactions. Prajneshu *et al*. (2002) modelled India's lac production cyclical data through Structural time-series (STS) approach and have shown superiority of this approach over the well-known Box-Jenkins autoregressive integrated moving average methodology. However, one limitation of this work is that the underlying model is "linear". During last three decades or so, the area of "Nonlinear time-series modelling" has been rapidly developing. An important subclass of such models is the Self-exciting threshold autoregressive (SETAR) model, proposed by H. Tong in 1980, and discussed in Fan and Yao (2003). A heartening aspect of this model is that it is capable of describing cyclical data having sudden rise and fall.

Nampoothiri and Balakrishna (2000) made attempts to fit SETAR model to monthly coconut oil prices at Cochin market by following "Recursive estimation" method. However, a drawback of this work is that the estimation procedure is adhoc in nature and is not based on any sound statistical optimization principle. Accordingly, the estimates obtained are not, in general, globally optimal. Ghosh *et al*. (2006) fitted the SETAR model by "Search algorithm" (Tong 1995) to describe India's lac export data exhibiting prominent cycles. However, main limitation of this algorithm is that the number of possible models to be searched is extremely large due to the fact that the algorithm entails a complete search technique. Specifically, if the number of autoregressive models is M, largest order is L,

*Corresponding author : Prajneshu
E-mail address : prajneshu@yahoo.co.in

number of threshold values is *S*, and number of delay parameter values is *T*, then the number of models to be computed is $\binom{S}{M-1} L^M T$, which is very large.

The purpose of this paper is to thoroughly study the powerful stochastic optimization technique of Genetic algorithm (GA) for estimation of parameters of SETAR model. GA is a target-oriented parallel search technique and is mainly applied to optimization process searching for universal or nearly universal extreme values. This methodology is then applied to a real data set. The organization of this paper is as follows. After a brief introduction in Section 1, the next section gives a description and identification of SETAR model. Section 3 discusses working principles of GA. This is followed by Section 4 dealing with an illustration of fitting the model through GA to India's lac export time-series data. Finally, Section 5 is concerned with some concluding remarks on the merits of GA and delineation of some research problems for future research work.

2. DESCRIPTION AND IDENTIFICATION OF SETAR MODEL

The SETAR model comprises several linear AR-models and the switch mechanism amongst AR-models is based on delay output and threshold values. The SETAR model is a particular class of Threshold autoregressive (TAR) models, which is a piece-wise linearization of nonlinear models over the state space by introduction of thresholds. In this paper, we shall confine only to SETAR two-regime model, which is described below.

A SETAR (2; k_1, k_2) two-regime model can be expressed as

$$X_t = \begin{cases} a_0^{(1)} + \sum_{i=1}^{k_1} a_i^{(1)} X_{t-i} + \varepsilon_t^{(1)}, \text{ if } X_{t-d} \le r \\ a_0^{(2)} + \sum_{i=1}^{k_2} a_i^{(2)} X_{t-i} + \varepsilon_t^{(2)}, \text{ if } X_{t-d} > r \end{cases} \quad (2.1)$$

where k_1, k_2 are orders of two AR-models corresponding to two regimes $R_i = (r_{i-1}, r_i], i = 1, 2$; $\{a_i^{(1)};\ i = 0, 1, 2, \ldots, k_1\}$ and $\{a_i^{(2)};\ i = 0, 1, 2, \ldots, k_2\}$ are autoregressive coefficients; $\varepsilon_t^{(1)}$, $\varepsilon_t^{(2)}$ are white noise terms; *d* is delay parameter value and *r* is threshold value.

Tong and Lim (1980) proposed the "*search algorithm*" for estimating delay parameter value *d*, threshold value *r*, and parameters of the two AR-terms of SETAR model given in eq. (2.1). The key point is to use pooled Akaike information criterion (AIC), where

$$AIC(t_q) = AIC(\hat{k}_1) + AIC(\hat{k}_2),$$

and

$$AIC(\hat{k}_i) = \min_{0 \le k_i \le L} \{N_i \log (RSS_i(k_i)/N_i) + 2(k_i + 1)\},\ i = 1, 2$$

Here *L* and N_i denote respectively maximum order and number of observations in the two-regimes defined above. The estimation procedure is based on first keeping the delay parameter value as fixed. Then, minimum AIC values for varying thresholds are computed. Finally, the best SETAR model is identified by minimizing the AIC values over varying delay parameter values.

3. ESTIMATION OF PARAMETERS USING GENETIC ALGORITHM

The genetic algorithm (GA), first proposed by Holland (1975), is a stochastic search and optimization procedure motivated by the principles of genetics and natural selection. It combines Charles Darwin's principle of "Natural selection" and "Survival of the fittest" with computer-constructed evolution mechanism to select better species from the original population. This is done by random exchange of information among them, expecting a superior offspring. Besides this, in order to avoid missing some good species and producing a local optimum, several mutations must be processed. Thus the problem encountered in classical optimization may be overcome by using GA.

In this paper, an attempt is made to implement real-coded GA (RCGA) to estimate SETAR parameters optimally. The crossover operator in RCGA is based on coding of solutions, where a solution is represented in terms of binary string(s). The basic concept of GA is to encode the solution into an appropriate range by using finite-length digital string(s) of length *B*. Here the string or chromosome exhibits biological representation

of the optimization problem of interest. A widely used decoding formula is

$$c = L + \left\{A \Big/ \left(2^B - 1\right)\right\} \times (U - L) \qquad (3.1)$$

where c is encoded value of a string, U and L are upper and lower bounds of the optimal solution to be searched, and A denotes decoded value of the string.

The three operators, viz. selection, crossover, and mutation make GA an important tool for optimization. When a string or a solution is created by GA, it is evaluated in terms of its fitness value given by

$$NAIC = [AIC(k_1) + AIC(k_2)]/(N - d) \qquad (3.2)$$

Selection operator of GA is performed to identify good solutions and to eliminate bad solutions from the population (Deb 2002). Since selection operator cannot create a new solution, crossover and mutation operators are used in mating pool to create a new population of solutions. Although, there exists a number of crossover operators in the GA literature, a number of difficulties arise while using the binary-coded GA to handle problems having a continuous search space. One of these is that of Hamming cliffs, which causes artificial hindrance to a gradual search. The other difficulty is its inability to achieve any arbitrary precision in arriving at the optimal solution. Thus, crossover operator used in binary coding needs to be redesigned in order to increase propagation of more meaningful *schemata* pertaining to a continuous search space. From the viewpoint of better search power, solving a real-valued optimization problem using RCGA which follows similar single-point binary crossover operator and applied on real solutions is more efficient than transforming it into binary-coded GA. Similar to binary-coded GA, selection operator using Tournament selection is applied and the set of solutions with improved fitness value is obtained in the next generation. A pair of real-valued decision variable is used to create a new pair of offspring using most popular RCGA, viz. Simulated binary crossover (SBX), which is described below. This is followed by perturbing the variable using mutation operator. Thus, a new set of solutions in the next generation are obtained. The process is continued until desired convergence is achieved.

3.1 SBX Operator

There exists a number of real-parameter GA implementations, where crossover and mutation operators are applied directly on real solutions. The SBX operator works with two parent solutions and creates two offspring. As the name suggests, SBX operator simulates the working principle of single-point crossover operator on binary strings. This operator respects the interval schemata processing in the sense that common interval schemata between parents are preserved in offspring. The procedure of computing offspring $x_i^{(1,t+1)}$ and $x_i^{(2,t+1)}$ from parent solutions $x_i^{(1,t)}$ and $x_i^{(2,t)}$, where $x_i^{(j,t)}$ is value of j^{th} parent (offspring) in t^{th} generation is as follows. A spread factor β_i is defined as ratio of absolute difference in offspring values to that of parents:

$$\beta_i = \left| \frac{x_i^{(2,t+1)} - x_i^{(1,t+1)}}{x_i^{(2,t)} - x_i^{(1,t)}} \right| \qquad (3.3)$$

The probability density function (p.d.f.) of β_i used to create an offspring is derived from the principle of single-point binary crossover and has similar search power in binary-coded GA. Deb and Agrawal (1995) derived the p.d.f. of β_i as

$$P(\beta_i) = \begin{cases} 0.5(\eta_c + 1)\beta_i^{\eta_c}, & \text{if } \beta_i \le 1 \\ 0.5(\eta_c + 1)\beta_i^{-\eta_c - 2}, & \text{otherwise} \end{cases} \qquad (3.4)$$

Fig. 1 shows above p.d.f. with η_c = 2 and 5 for creating offspring from two parent solutions ($x_i^{(1,t)}$ = 2.0 and $x_i^{(2,t)}$ = 5.0) in real space. It may be pointed out that distribution index η_c is a non-negative real number. A large value of η_c gives a higher probability for creating 'near-parent' solutions while its small value allows distinct solution to be selected as offspring.

To perform SBX operator, first a Uniform random number U_i between 0 and 1 is created. Thereafter, ordinate β_{q_i} is found so that the area under probability curve from 0 to β_{q_i} is equal to the chosen random number U_i. Here β_{q_i} represents q_i^{th} quantile of the

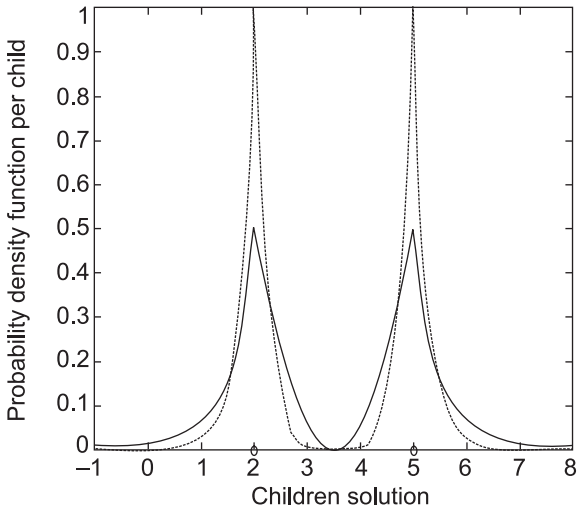

Fig. 1. p.d.f. for creating offspring under an SBX with η_c operator (Solid line for $\eta_c = 2$ and dotted lines for $\eta_c = 5$). Parents are marked with an "o"

p.d.f. given in eq. (3.4) and is calculated by equating area under the probability curve $q_i = u_i$, as follows:

$$\beta_{q_i} = \begin{cases} (2u_i)^{-\eta_c - 1}, & \text{if } u_i \le 0.5 \\ [2(1-u_i)]^{-\eta_c - 1}, & \text{otherwise.} \end{cases} \quad (3.5)$$

Subsequently, offspring are calculated using

$$x_i^{(1,\,t+1)} = 0.5\left\{\left(1+\beta_{q_i}\right)x_i^{(1,\,t)} + \left(1-\beta_{q_i}\right)x_i^{(2,\,t)}\right\} \quad (3.6)$$

$$x_i^{(2,\,t+1)} = 0.5\left\{\left(1-\beta_{q_i}\right)x_i^{(1,\,t)} + \left(1+\beta_{q_i}\right)x_i^{(2,\,t)}\right\} \quad (3.7)$$

It may be noted that two offspring are symmetrically distributed about their respective parent solutions. This is deliberately enforced to avoid bias towards any particular parent solution. Another important aspect of this crossover operator is that, for a fixed η_c, offspring have a spread which is proportional to that of parent solution, i.e.

$$x_i^{(2,\,t+1)} - x_i^{(1,\,t+1)} = \beta_{q_i}\left(x_i^{(2,\,t)} - x_i^{(1,\,t)}\right) \quad (3.8)$$

3.2 Mutation Operator

This operator does the same task as performed by real-parameter crossover operator. The only difference is that the former perturbs every parent parameter solution to create a new population while the latter perturbs two parent parameter solutions at a time to produce two new offspring. With only one parent, a range of perturbation must be predefined. There are several mutation operators, like Polynomial, Random, and Non-uniform. However, one advantage of the Polynomial mutation operator is that the p.d.f. does not change with generations, thereby avoiding local optima. Therefore, in our study, we shall be employing this operator. It mutates i^{th} variable $x_i^{(1,\,t+1)}$ to $y_i^{(1,\,t+1)}$ by the transformation:

$$y_i^{(1,\,t+1)} = x_i^{(1,\,t+1)} + \left\{x_i^{(U)} - x_i^{(L)}\right\}\bar{\delta}_i \quad (3.9)$$

where $\bar{\delta}_i$ is drawn from the p.d.f.

$$P(\delta) = 0.5\,(\eta_m + 1)\,(1-|\delta|)^{\eta_m},\ |\delta| \le 1, \text{ and } \eta_m \ge 0 \quad (3.10)$$

The parameter η_m is responsible for controlling the shape of p.d.f. of $y_i^{(1,\,t+1)}$.

4. AN ILLUSTRATION

India's annual lac export data for the period 1901-2001 is obtained from the website *http://www.shellacepc.com/statistics.html* of "Shellac and Forestry Produce Export Promotion Council, Kolkata" and several issues of "Agricultural Statistics at a Glance, Directorate of Economics and Statistics, New Delhi". Before making attempts to fit the SETAR model using GA, significant features of the data noticed are:

(i) Obvious cycles of approximately 13 years with varying amplitudes.

(ii) The rise period from a local minimum to next local maximum, exceeding descent period from a maximum to next local minimum, thereby showing time irreversibility.

To apply SETAR two-regime model to above data, it is desirable to carry out a preliminary Exploratory data analysis for justifying choice of this model. For the data under consideration, this has already been done in Section 3 by Ghosh *et al*. (2006). Some of the important conclusions from that study are reproduced here. The Directed scatter diagram of the time-series $\{x_t\}$, $t = 1, 2\ldots$, indicated that the joint distribution of (x_t, x_{t-j}), $t = j+1, j+2, \ldots, N$; $j = 1, 2, \ldots, p$ are non-Gaussian. The kernel density estimation revealed that

the data exhibit multimodality. The lag regression function analysis showed a gradual shift from a linear function through a single hump curve back to a linear function with single hump. Periodogram function analysis indicated presence of at least one peak at approximate frequency of 2 cycles per 13 years besides a fundamental frequency of a cycle per 13 years. In view of all this, it is desirable to apply SETAR nonlinear time-series model to Indian lac export data.

The RCGA with SBX and η_c = 2 is used for estimation of parameters of SETAR model. The other RCGA parameters, viz. population size, crossover probability, and mutation probability for minimization of eq. (3.2) are respectively 100, 0.9, 0.01 with number of generations as 100. Relevant computer program is developed using C-language and is appended as an Annexure. The proposed algorithm enables us to select SETAR (2; 1, 1) model, when delay parameter d =1, k_1 =1 and k_2 = 1. The optimal threshold value comes out as 10500 metric tonnes, which is 30th percentile of the data. The best-identified model on the basis of minimum *NAIC* value, viz. 15.63, is

$$X_t = \begin{cases} 2178.870 + 0.718\, X_{t-1}, & \text{if } X_{t-1} \le 10500 \\ 6441.216 + 0.723\, X_{t-1}, & \text{if } X_{t-1} > 10500 \end{cases} \quad (4.1)$$

with Var $\left(\varepsilon_t^{(1)}\right)$ = 67.83 and Var $\left(\varepsilon_t^{(2)}\right)$ = 124.45. The standard errors of parameter estimates $(a_0^{(1)}, a_1^{(1)}\ a_0^{(2)}, a_1^{(2)})$ are respectively computed as (64.29, 0.05, 121.03, 0.09). The fitted SETAR (2; 1, 1) model along with data points is exhibited in Fig. 2.

A mechanistic interpretation of fitted *SETAR* model is as follows. Eq. (4.1) can be rewritten as

$$X_t - X_{t-1} = \begin{cases} 2178.870 - 0.282\, X_{t-1}, & \text{if } X_{t-1} \le 10500 \\ 6441.216 - 0.277\, X_{t-1}, & \text{if } X_{t-1} > 10500 \end{cases} \quad (4.2)$$

In the upper regime, i.e. $X_{t-1} > 10500$, $X_t - X_{t-1}$ tends to be negative, implying decrease in lac export. In the lower regime, i.e. $X_{t-1} \le 10500$, $X_t - X_{t-1}$ tends to be marginally positive, implying slow increase in lac export. This phenomenon leads to cyclicity, which is in agreement with observed lac export data.

4.1 Forecasting for Hold-out Data

Using observed data and fitted model in eq.(4.1), one-step-ahead predicted value x_{N+1} of X_{N+1} are obtained. Now, treating this x_{N+1} as observed value of X_{N+1}, same calculation is repeated and predicted value x_{N+2} of X_{N+2}, and so on, are obtained. Forecast values for years 2002, 2003,…,2008 are computed and reported in Table 1.

Table 1. Forecasting for hold-out data of India's lac export data (metric tonnes) by SETAR two-regime model

Year	Actual value	Forecast values using	
		Search algorithm	Genetic algorithm
2002	7015.00	6274.06	6856.25
2003	5819.00	7407.90	6222.50
2004	10500.00	6215.43	10576.78
2005	8540.00	10804.23	9717.87
2006	9300.00	8922.63	8310.59
2007	7510.00	9652.23	8856.27
2008	7980.00	7933.83	7571.05

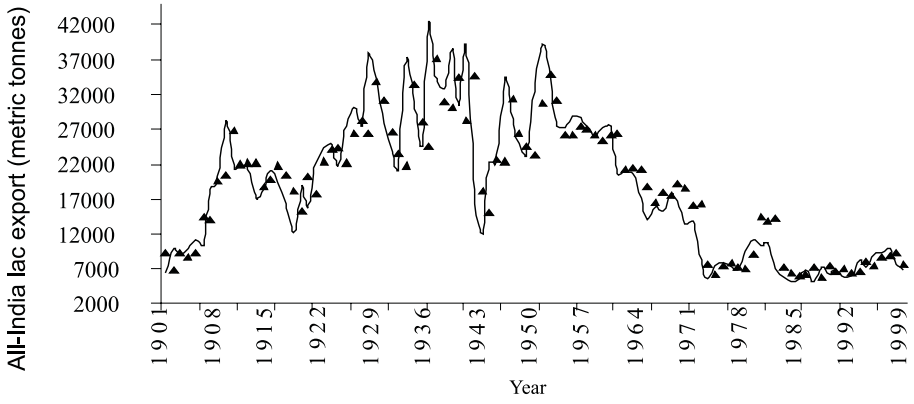

Fig. 2. Fitted SETAR (2; 1, 1) two-regime model for modelling and forecasting India's lac export data

The forecast performance for hold-out data on the basis of Root mean square error and Mean absolute error for fitted SETAR model using search algorithm are respectively computed as 2109.19 and 1659.33. The corresponding values for fitted SETAR model using genetic algorithm are respectively computed as 881.59 and 867.66. Thus, SETAR model fitted using GA technique is superior to that fitted using Search algorithm for forecasting purposes.

To sum up, it may be concluded that, for given data, SETAR two-regime nonlinear time-series model fitted through Genetic algorithm procedure has outperformed that fitted using Search algorithm procedure for modelling as well as forecasting.

5. CONCLUDING REMARKS

In this paper, superiority of Genetic algorithm stochastic optimization technique over Search algorithm technique for fitting of Self-exciting threshold autoregressive nonlinear time-series model to India's lac export data is demonstrated. It may be pointed out that Search algorithm determines final value of threshold (r) only from a set of discrete potential values $\{t_{0.30}, t_{0.40}, t_{0.50}, t_{0.60}, t_{0.70}\}$. On the other hand, Genetic algorithm enables obtaining the optimum value of threshold (r) by using Stochastic optimization algorithm in the entire solution space, thereby ensuring global optimum. Further, given a fixed threshold (r), Search algorithm requires computation of Akaike information criterion values for all combinations of orders of the two regimes. Thus, this algorithm is computationally inefficient. In contrast to this, Genetic algorithm is based on selection criterion, which employs "Selection operator", ensuring thereby that fitness values over successive iterations are non-increasing. In other words, Genetic algorithm is computationally more efficient than Search algorithm.

As future work, possibility of employing Fast rectangular window type algorithm to estimate threshold (r) may be explored. For the purpose of inference, use of Bayesian methods, like MCMC simulation technique may be considered. Another direction of future research may be to develop efficient estimation procedures for fitting Self-exciting threshold autoregressive nonlinear time-series model with three or more regimes.

ACKNOWLEDGEMENT

Authors are grateful to the referee for valuable comments. Drs. Himadri Ghosh and Prajneshu also thank Department of Science and Technology, New Delhi for providing financial assistance during the course of this research work.

REFERENCES

Deb, K. (2002). *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley, Singapore.

Deb, K. and Agrawal, R.B. (1995). Simulated binary crossover for continuous search space. *Compl. Syst.*, **9**, 115-148.

Fan, J. and Yao, Q. (2003). *Nonlinear Time Series: Nonparametric and Parametric Methods*. Springer, New York.

Ghosh, H., Sunilkumar, G. and Prajneshu (2006). Self exciting threshold autoregressive models for describing cyclical data. *Cal. Stat. Assoc. Bull.*, **58**, 115-132.

Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor.

Nampoothiri, C.K. and Balakrishna, N. (2000). Threshold autoregressive model for a time series data. *J. Ind. Soc. Agril. Statist.*, **53**, 151-160.

Prajneshu, Ravichandran, S. and Wadhwa, S. (2002). Structural time-series models for describing cyclical fluctuations. *J. Ind. Soc. Agril. Statist.*, **55**, 70-78.

Tong, H. (1995). *Non-Linear Time Series: A Dynamical System Approach*. Oxford University Press, Oxford.

Tong, H. and Lim, K.S. (1980). Threshold autoregression, limit cycles and cyclical data. *J. Roy. Stat. Soc.,* **B42**, 245-292.

ANNEXURE

OBJECTIVE FUNCTION (TO BE MINIMIZED):

```
void objective(indv)
INDIVIDUAL *indv;
{int i;
   double term3, pi, your_func;
   double g[MAXCONSTR], gsum, x[2*MAXVECSIZE];
   double sig11=0,sig12=0,sig13=0,sig14=0,sig21=0,sig22=0,sig23=0,sig24=0;
   double AIC11,AIC12,AIC13,AIC14,AIC21,AIC22,AIC23,AIC24;
   double AIC_A[8]; double temp1,temp2,r;
   int k1=0,k2=0,d,count1=0,count2=0; int j,k=0;
   double NAIC[1000],NAIC_Sorted[1000],VALUE_R[1000];
   int VALUE_K1[1000],VALUE_K2[1000],VALUE_D[1000];
   FILE *(create file);
   for (i=0; i < nvar_bin; i++)
   x[i] = indv->xbin[i];
   for (i=nvar_bin; i < nvar_bin+nvar_real; i++)
   x[i] = indv->xreal[i-nvar_bin];
#ifdef (define problem number)
   term3=0;
   MINM = 1;
   try1=fopen(path for text file for);
   for(loop for d)
{for (loop for r)
{for (j=1;j<130;j++)
{if (y[j-d]<=r)
{for(k1=1;k1<=4;k1++)
{if (k1==1)
{sig11=sig11+pow((y[j]-x[0]-x[1]*y[j-1]),2);}
   else if (k1==2)
{sig12=sig12+pow((y[j]-x[0]-x[1]*y[j-1]-x[2]*y[j-2]),2);}
   else if (k1==3)
{sig13=sig13+pow((y[j]-x[0]-x[1]*y[j-1]-x[2]*y[j-2]-x[3]*y[j-3]),2);}
   else if (k1==4)
{sig14=sig14+pow((y[j]-x[0]-x[1]*y[j-1]-x[2]*y[j-2]-x[3]*y[j-3]-x[4]*y[j-4]),2);}}}}
   else
{for(k2=1;k2<=4;k2++)
{if (k2==1)
{sig21=sig21+pow((y[j]-x[5]-x[6]*y[j-1]),2);}
   else if (k2==2)
{sig22=sig22+pow((y[j]-x[5]-x[6]*y[j-1]-x[7]*y[j-2]),2);}
   else if (k2==3)
```

```
{sig23=sig23+pow((y[j]-x[5]-x[6]*y[j-1]-x[7]*y[j-2]-x[8]*y[j-3]),2);}
   else if (k2==4)
{sig24=sig24+pow((y[j]-x[5]-x[6]*y[j-1]-x[7]*y[j-2]-x[8]*y[j-3]-x[9]*y[j-4]),2);}}}}
   AIC11=count1*log(sig11)+2*(1+1);
   AIC_A[0]=AIC11;
   AIC12=count1*log(sig12)+2*(2+1);
   AIC_A[1]=AIC12;
   AIC13=count1*log(sig13)+2*(3+1);
   AIC_A[2]=AIC13;
   AIC14=count1*log(sig14)+2*(4+1);
   AIC_A[3]=AIC14;
   AIC21=count2*log(sig21)+2*(1+1);
   AIC_A[4]=AIC21;
   AIC22=count2*log(sig22)+2*(2+1);
   AIC_A[5]=AIC22;
   AIC23=count2*log(sig23)+2*(3+1);
   AIC_A[6]=AIC23;
   AIC24=count2*log(sig24)+2*(4+1);
   AIC_A[7]=AIC24;
   //Calculate AIC Value
for (i=0;i<=3;i++)
{for(j=4;j<=7;j++)
{NAIC[k]=(AIC_A[i]+AIC_A[j]/(100-d));
   NAIC_Sorted[k]=NAIC[k];
   VALUE_K1[k]=i+1;
   VALUE_K2[k]=j-3;
   VALUE_D[k]=d;
   VALUE_R[k]=r;
   k++;}}}}
for(i=0;i<k;i++)
{for(j=0;j<k-1;j++)
{if(AIC[j]>=AIC[j+1])
   AIC[j]=AIC[j+1];
   AIC[j+1]=temp1;}}}
   fprintf(try1,"\nAIC=%4.5lf,",AIC[0]);
   for(i=0;i<k;i++)
   {if(NAIC_Sorted[i]==NAIC[0])
   {break; }}
   fprintf("k1=%d,k2=%d,d=%d,r=%lf",VALUE_K1[i],
                    VALUE_K2[i], VALUE_D[i],VALUE_R[i]);
   fclose(try1);
   your_func=NAIC[0];
```