International Journal of Scientific Research & Engineering Technology (IJSET)
Vol.7 pp. 8-11

Identification of Nonlinear Systems Using T-S Fuzzy tuned by Backtracking Search Optimization Algorithm

Mourad Turki[#1]

[#] *Higher Institute of Technological Studies of Mahdia , Tunisia*

[1] mouradessturki@yahoo.fr

***Abstract*— This paper proposes a novel metaheuristic optimization algorithm called Backtracking Search Optimization Algorithm BSA to extract T-S model. In this work, the structure and parameters of the fuzzy model are encoded into a particle. Thus, the optimal structure and parameters are achieved simultaneously. The proposed method was compared with others methods such as PSO and CS through a modelling problem.**

***Keywords*— Modeling, TS Fuzzy Systems, Backtracking Search Optimization Algorithm, nonlinear systems**

I. INTRODUCTION

Takagi-Sugeno (T-S) fuzzy models have been used in many fields due to their ablility to handle the complex nonlinear problems and they are more interpretable than the black-box models. A fuzzy model is formed by several linguistic If-Then rules, which approximate the relationship between the input and output [1].

To optimize the Fuzzy models two tasks are needed: the identification of structure and parameters. Because of their global searching capability, evolutionary algorithms (EAs), such as genetic algorithm (GA), genetic programming (GP), evolutionary programming(EP) and evolution strategy (ES), have been employed to optimize the parameters of the fuzzy models [2] - [4]. The particle swarm optimization PSO is also used to extract fuzzy models such as in [5].

In [6] the cuckoo search algorithm CS is used to optimize both the structure and parameters of the fuzzy models.

In this paper a novel method of optimization called Backtracking Search Optimization BSA is proposed to extract fuzzy models and compared with CS and PSO methods.

This paper is organized as follows. Section II introduces the structure of T–S fuzzy model. Backtracking Search Optimization algorithm BSA is described in Section III. Section IV presents the simulation results. Conclusions are given in Section V.

II. STRUCTURE OF T-S MODEL

The Takagi–Sugeno (T–S) fuzzy model was first presented in [7] and is described by the following fuzzy IF–THEN rules:

Rule i :If x_1 is A_i^1 and ... and x_{N_I} is $A_i^{N_I}$ Then

$$y_i = \alpha_i^0 + \alpha_i^1 x_1 + \ldots + \alpha_i^{N_i} x_{N_i} \quad (1)$$

Where $i = 1, \ldots, N_R$, N_R is the number of fuzzy rules ; $x = [x_1, \ldots, x_{N_I}]$ is the input variable, N_I is the dimension of input variable ; $\alpha_i^0, \alpha_i^1, \ldots, \alpha_i^{N_i}$ are the consequent parameters, y_i is an output from the i^{th} fuzzy rule, and A_i^j is a fuzzy variable.

Given the input $x = [x_1, \ldots, x_{N_I}]$, the final output of the fuzzy model is inferred by a weighted mean defuzzification as follows:

$$\hat{y} = \frac{\sum_{i=1}^{N_R} \omega_i y_i}{\sum_{i=1}^{N_R} \omega_i} \quad (2)$$

Where the weight strength ω_i of the i^{th} rule is calculated as:

$$\omega_i = \prod_{j=1}^{n} \mu(A_i^j) \quad (3)$$

Where $\mu(A_i^j)$ is the grade of the membership function (MF) of A_i^j and is characterized by a Gaussian function as

$$\mu_{A_i^j}(x_j) = \exp\left(-\frac{1}{2}\left(\frac{x_j - c_i^j}{\sigma_i^j}\right)^2\right) \quad (4)$$

Where c_i^j and σ_i^j represent the centre (or mean) and the width (or standard deviation) of the MF respectively. c_i^j and σ_i^j are adjustable parameters called the premise parameters [8].

III. BACKTRACKING SEARCH OPTIMIZATION ALGORITHM

Backtracking Search Optimization Algorithm (BSA) is a newly developed stochastic search algorithm, which has been first proposed in [9] by Pinar Civicioglu. Unlike many swarm-intelligence optimization algorithms, BSA has a single

ISSN 1737-9296

control parameter which is not extremely sensitive to the initial value of this parameter. As a population-based iterative evolutionary algorithm (EA), the population generation strategy of BSA includes three operators: selection, mutation and crossover. The searching framework of BSA is described in Fig. 1 with corresponding details stated as follows.

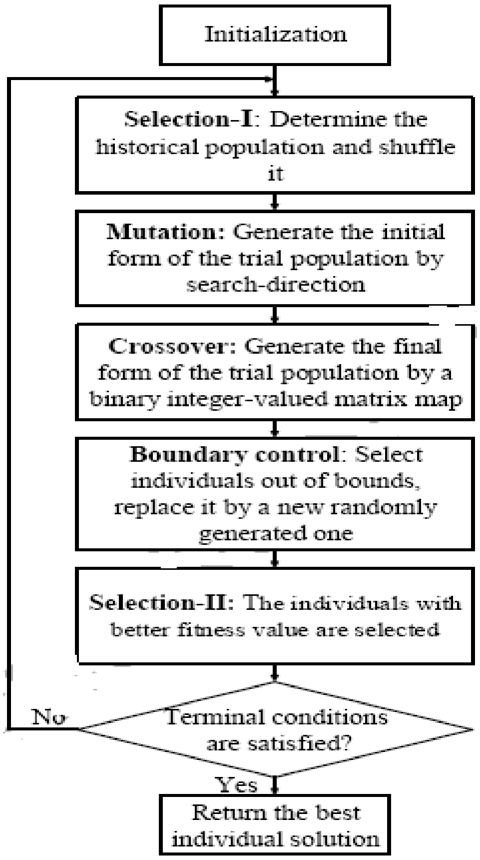


Fig. 1 The framework of BSA

At the first step, the BSA generates two randomly distributed initial populations P and initial historical population oldP according to Eq.(5):

$$\begin{cases} P_{ij} = U(low_j, up_i) \\ oldP_{ij} = U(low_j, up_i) \end{cases} ,i=1,2,\ldots N;\quad j=1,2,\ldots\ldots D \qquad (5)$$

where N , D and U are the population size, the individual dimensionality and the uniform distribution, respectively. Each P_i is a target individual in the population P. Then the first selection determines the historical population *oldP* to be used for calculating the search direction. BSA has the option of redefining *oldP* at the beginning of each iteration through the „if-then" rule in Eq. (6):

$$if\ \ a<b\ \ then\ oldP := P \mid a,b \sim U(0,1) \qquad (6)$$

where := is defined as the update operation. Eq. (6) ensures that BSA designates a population belonging to a randomly selected previous generation as the historical population, which is temporarily recorded until it is changed. After that, the order of the individuals in *oldP* needs to be randomly ordered by a shuffling function, as is illustrated below:

$$oldP:=permuting(oldP) \qquad (7)$$

In order to generate the initial form of the trial population *Mutant* , the mutation operation considers both *P* and *oldP* , where (*oldP* - *P*) is the search-direction matrix. The process is described as:

$$Mutant=P + F.\ (oldP-P) \qquad (8)$$

where the mutation coefficient F = 3.*rndn* , *rndn* ~ $N(0,1)$ (N is the standard normal distribution), controls the amplitude of the search-direction.

The crossover process of BSA generates the final form of the trial population T . Firstly, a binary integer-valued matrix **map** of size $N\times D$ is obtained by:

Map(1:N,1:D)=0 //Initial map
If a<b∣ a,b~U(0,1) then
For I from 1 to N do
$Map_{i,u(1:[mixrate.rnd.D])}=0 \mid u=permuting(<1,2,\ldots,D>)$ (9)
End
Else
For I from 1 to N do, $map_{i,randi(D)}=0$, *end*
End

where *mixrate* is the sole control parameter in BSA (the mix rate parameter), controls the number of elements of individuals that will mutate in a trial by using [*mixrate*.*rdn*.*D*]. Secondly, *T* is updated with:

$$if\ map_{nm}=1,\ (n=1,2,\ldots,N, m=1,2,\ldots\ldots,D),$$
$$Then\quad T_{nm}=P_{nm} \qquad (10)$$

Some individuals of the final trial population T can overflow the allowed search space, so boundary control is rather necessary. Subsequently, the last step comes to the greedy selection. At this stage, the T_i s that have better fitness values than the corresponding P_i s are used to update the P_i s. If the best individual P_{best} has a better fitness value than the global minimum value obtained so far by BSA, the global minimize is updated to be P_{best} , and the global minimum value is updated to be the fitness value of P_{best} .

IV. Simulation results

In this section, the proposed BSA algorithm is used to extract T-S models in the aim to identify a nonlinear plant modelling problem and compared to PSO [10] and CS [6]. The parameters of CS and PSO algorithms are given respectively in Table I and Table II.

ISSN 1737-9296

TABLE I

PARAMETER VALUES

Parameter	Value
Number of nests	20
p_a	0.25
λ	1.5
Step size α	1

TABLE III

PSO PARAMETER VALUES

Parameter	Value
Population size	20
c_1	1.49
c_2	1.49
w_{min}	0.4
w_{max}	0.9

For the BSA algorithm the mixrate=1.For the three methods the commonly used parameters are given as follows: population size = 20, maximum generation=500. Thus the number of evaluations is 10000.

The mean square error MSE is used to test the performance indices, and it is defined as follows:

$$MSE = \frac{1}{n}\sum_{k=1}^{n}(y_{ref}(k) - y(k))^2 \qquad (11)$$

The structure and parameters are all encoded into a particle. As shown in Fig. 2, the particle is represented by a vector which includes the premise parameters, the consequent parameters and the labels. The fuzzy rules are selected as follows:

If l_i >0 then the rule i is active, $i \in$ [1…N], where N is the pre-defined maximum number of rules. In our work, N is set to 5. All the active rules compose the fuzzy inference system.

The nonlinear dynamic plant used is described by the difference equation [10] as follows:

$$y(k) = g\left(y(k-1), y(k-2)\right) + u(k) \qquad (12)$$

where

$$g\left(y(k-1), y(k-2)\right) = \frac{y(k-1)y(k-2)(y(k-1)-0.5)}{1+y(k-1)^2+y(k-2)^2} \qquad (13)$$

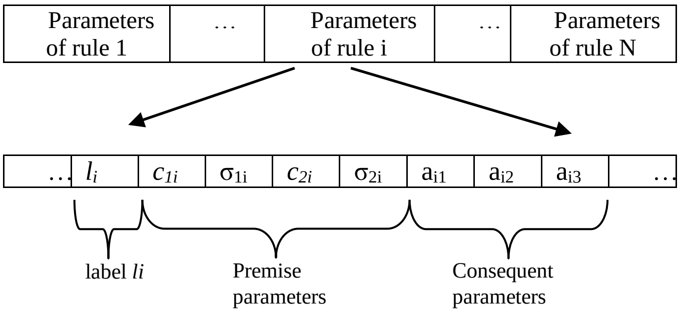


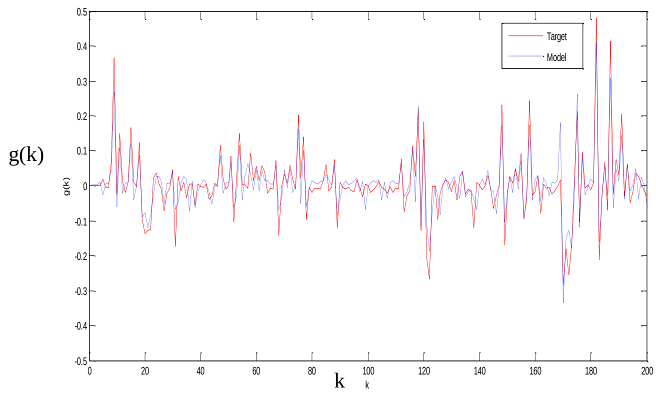
Fig. 2 Structure of a particle

The goal is to approximate the nonlinear part $g\left(y(k-1), y(k-2)\right)$ called the "unforced system" in control literature. It is a two input /single output model. The evolution process was repeated 50 times on a Pentium Core 2 Duo (1.8 GHz CPU) and 2GB of main memory in the same computing environment (MATLAB 2007a). All the coefficients of the consequent linear function are limited to [-10, 10] and the width of the Gaussian function is limited to [0, 5].

The 400 simulated data points were obtained from the nonlinear dynamic plant defined by (12) and (13). The 200 training data were generated using a random input signal u(k) that is distributed in [−1.5, 1.5], and 200 samples of testing data were generated using a sinusoidal input $u(k) = \sin\left(\frac{2\pi k}{25}\right)$.

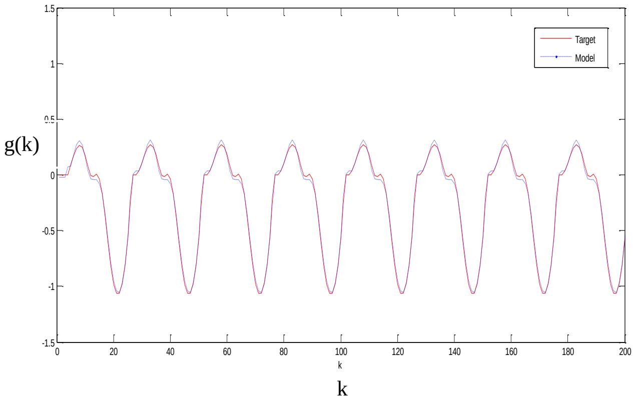
The results of BSA method are compared with PSO and CS. The BSA method has the best MSE in both training and testing stages than PSO method. This demonstrates that BSA gives good results compared to PSO and CS.

Fig. 3 shows the target and CS model output in the training and testing stages. The errors between target output and model output can be seen in Fig. 4. As we can see in Fig. 3, the BSA method has good identification ability and it can predict the output with small errors.



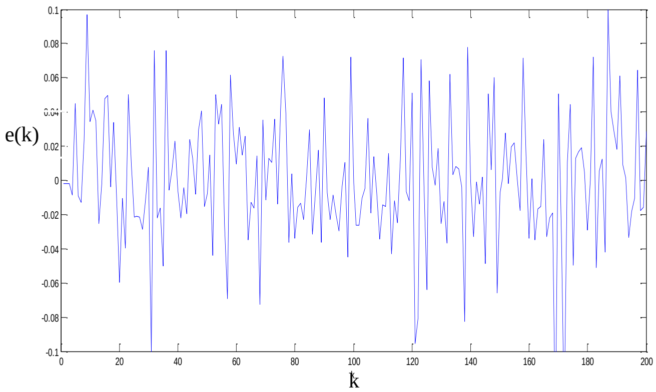
(a)

ISSN 1737-9296

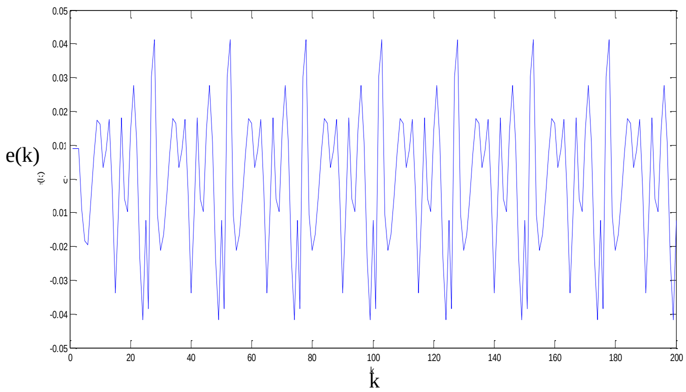


(b)

Fig. 3 The target output and model output for nonlinear plant modelling problem, (a) training stage and (b) testing stage



(a)



(b)

Fig. 4 The errors between target output and model output, (a) training stage and (b) testing stage

V. CONCLUSION

In this paper, the extracting of T–S fuzzy model using Backtracking Search Optimization BSA is presented. The developed T–S fuzzy model has the advantage that the rules structures and both the premises and consequents parameters can be optimised simultaneously. The obtained T–S fuzzy model with BSA method has a smaller MSE than PSO and CS in modelling of nonlinear system. Thus we can conclude that BSA method has a better optimizing accuracy in modelling.

REFERENCES

[1] M. S. Kim, C.H.Kim and J.J. Lee, "Evolving compact and interpretable Takagi-Sugeno Fuzzy models with a new encoding scheme," IEEE Trans. on Systems, Man, and Cybernetics. Part B: Cybernetics, 36(5), 2006, pp 1006-1023.

[2] O.Cordon and F. Herrera, "A Two-Stage Evolutionary Process for Designing TSK Fuzzy Rule-Based Systems, IEEE Trans. Systems., Man, and Cybernetics. Part B: Cybernetics, 29(6), 1999, pp 703-715.

[3] O. Cordon, F.Herrera ,F. Gomide, F. Hoffmann and L. Magdalena, "Ten years of genetic fuzzy systems: current framework and new trends," in Proc. IFSA and 20 th NAFIPS conference , 2001, pp 1241 -1246

[4] S.J. Kang, C.H. Woo, H.S. Hwang and K.B. Woo, "Evolutionary Design of Fuzzy Rule Base for Nonlinear System Modeling and Control," IEEE Trans. Fuzzy Systems, 8(1), 2000, pp 37-45.

[5] A. Khosla, S. Kumar and K.K. Aggarwal, "A framework for identification of fuzzy Models through particle swarm optimization," in: IEEE Indicon Conference, Chennai, India, 2005, pp. 388–391.

[6] M. Turki and A. SAKLY, "Cuckoo search-based TS fuzzy identification of nonlinear systems," In : *Systems and Control (ICSC), 2015 4th International Conference on*. IEEE, 2015. p. 151-154.

[7] T.Takagi and M.Sugeno, "Fuzzy identification of systems and its application to modeling and control," IEEE Trans. On Systems, Man, and Cybernetics. Part B: Cybernetics, 15(1), 1985, pp 116-132.

[8] C.F. Juang, I.F. Chung and C.H. Hsu, "Automatic construction of feedforward/ recurrent fuzzy systems by clustering-aided simplex particle swarm optimization," Fuzzy Sets and Systems, 2007, pp 1979–1996.

[9] P. Civicioglu, Backtracking search optimization algorithm for numerical optimization problems, Applied Mathematics and Computation, 219 (2013) 8121-8144.

[10] L. Wang and J. Yen, "Extracting Fuzzy Rules for System Modeling Using a Hybrid of Genetic Algorithms and Kalman Filter," Fuzzy Sets and Systems , 101 (3), 1999, pp 353–362.

ISSN 1737-9296