# Identification of Fuzzy Inference System Based on Information Granulation

**Wei Huang[1], Lixin Ding[1], Sung-Kwun Oh[2], Chang-Won Jeong[3] and Su-Chong Joo[3]**
[1] State Key Laboratory of Software Engineering, Wuhan University, China
[e-mail: huangwabc@163.com]
[2] Department of Electrical Engineering, University of Suwon, South Korea
[e-mail: ohsk@suwon.ac.kr]
[3] Department of Computer Engineering, Wonkwang University, South Korea
[e-mail: {mediblue, scjoo}@wku.ac.uk]
*Corresponding author: Sung-Kwun Oh

*Received April 19, 2010; revised June 7, 2010; accepted June 28, 2010; published August 27, 2010*

### Abstract

In this study, we propose a space search algorithm (SSA) and then introduce a hybrid optimization of fuzzy inference systems based on SSA and information granulation (IG). In comparison with "conventional" evolutionary algorithms (such as PSO), SSA leads no.t only to better search performance to find global optimization but is also more computationally effective when dealing with the optimization of the fuzzy models. In the hybrid optimization of fuzzy inference system, SSA is exploited to carry out the parametric optimization of the fuzzy model as well as to realize its structural optimization. IG realized with the aid of C-Means clustering helps determine the initial values of the apex parameters of the membership function of fuzzy model. The overall hybrid identification of fuzzy inference systems comes in the form of two optimization mechanisms: structure identification (such as the number of input variables to be used, a specific subset of input variables, the number of membership functions, and polyno.mial type) and parameter identification (viz. the apexes of membership function). The structure identification is developed by SSA and C-Means while the parameter estimation is realized via SSA and a standard least square method. The evaluation of the performance of the proposed model was carried out by using four representative numerical examples such as No.n-linear function, gas furnace, NO.x emission process data, and Mackey-Glass time series. A comparative study of SSA and PSO demonstrates that SSA leads to improved performance both in terms of the quality of the model and the computing time required. The proposed model is also contrasted with the quality of some "conventional" fuzzy models already encountered in the literature.

***Keywords:*** Space search algorithm (SSA), particle swarm algorithm (PSO), information granulation (IG), fuzzy inference system (FIS)

**DOI: 10.3837/tiis.2010.08.008**

## 1. Introduction

In recent years, fuzzy modeling has been utilized in many fields for engineering, medical engineering, and even social science [1]. As for fuzzy model construction, identification of fuzzy rules is one of most important parts in the design of rule-based fuzzy modeling.

Many identification methods for fuzzy models have been studied over the past decades. In the early 1980s, linguistic modeling[2] was proposed as primordial identification methods for fuzzy models. Then Tong et.al [3], C.W.Xu et.al [4] studied differnet approaches for fuzzy models. While appealing with respect to the basic topology (a modular fuzzy model composed of a series of rules) [5], these models still await formal solutions as far as the structural optimization of the model is concerned, say a construction of the underlying fuzzy sets – information granules being viewed as basic building blocks of any fuzzy model. Oh and Pedrycz [6] have proposed some enhancements to the model, yet the problem of finding "good" initial parameters of the fuzzy set in the rules remains open. To solve this problem, several genetically identification methods for fuzzy models have been proposed. Liu et.al [7] , Chung and Kim [8] and others have discussed employing genetic algorithms to fuzzy models, respectively. In a word, evolutionary identification methods have proven to be useful in optimization of such problems.

In this study, we propose a space search algorithm (SSA) and then introduce a hybrid optimization of fuzzy inference systems based on SSA and information granulation (IG). SSA is exploited here to carry out the parameter estimation of the fuzzy models as well as to realize structural optimization. The identification process is comprised of two phases, namely a structural optimization (the number of input variables to be used, a specific subset of input variables, and the number of membership functions) and parametric optimization (apexes of membership function). The SSA and the least square method (LSE) are used in each phase of this sequence. Information granulation is realized with the aid of HCM, SSA and LSM. HCM is used to help determine the initial parameters of the fuzzy model such as the initial location of apexes of the membership functions and the prototypes of the polyno.mial functions being used in the premise and consequence parts of the fuzzy rules, while SSA and LSM is employed to adjust the initial values of the parameters. To evalutate the performance of the proposed model, we exploit two kinds of well-kno.wn data set. A hybrid optimization of fuzzy inference systems based on PSO and IG is also implemented for the comparative study.

## 2. IG-based Fuzzy Model

Granulation of information is an inherent and omnipresent activity of human beings carried out with intent of gaining a better insight into a problem under consideration and arriving at its efficient solution. In particular, granulation of information is aimed at trasforming the problem at hand into serveral smaller and therefore more manageable tasks. The identification of the conclusion parts of the rules deals with a selection of their structure (type 1, type 2, type 3 and type 4) that is followed by the determination of the respective parameters of the local functions occurring there. The conclusion part of the rule that is extended form of a typical fuzzy rule in the TSK (Takagi-Sugeno.-Kang) fuzzy model has the form.

$$R^j : If\ x_1\ is\ A_{1c}\ and \cdots and\ x_k\ is\ A_{kc}\ then\ y_j - M_j = f_j(x_1, \cdots, x_k) \tag{1}$$

Type 1 (Simplified Inference): $f_j = a_{j0}$

Type 2 (Linear Inference): $f_j = a_{j0} + a_{j1}(x_1 - V_{j1}) + \cdots + a_{jk}(x_k - V_{jk})$

Type 3 (Quadratic Inference):

$$f_j = a_{j0} + a_{j1}(x_1 - V_{1j}) + \cdots + a_{jk}(x_k - V_{kj}) + a_{j(k+1)}(x_1 - V_{1j})^2 + \cdots + a_{j(2k)}(x_k - V_{kj})^2 \\ + a_{j(2k+1)}(x_1 - V_{1j})(x_2 - V_{2j}) + \cdots + a_{j((k+2)(k+1)/2)}(x_{k-1} - V_{(k-1)j})(x_k - V_{kj})$$

Type 4 (Modified Quadratic Inference):

$$f_j = a_{j0} + a_{j1}(x_1 - V_{1j}) + \cdots + a_{jk}(x_k - V_{kj}) + a_{j(k+1)}(x_1 - V_{1j})(x_2 - V_{2j}) + \cdots \\ + a_{j(k(k+1)/2)}(x_{k-1} - V_{(k-1)j})(x_k - V_{kj})$$

The calculations of the numeric output of the model, based on the activation (matching) levels of the rules there, rely on the following expression.

$$y^* = \frac{\sum_{j=1}^{n} w_{ji} y_i}{\sum_{j=1}^{n} w_{ji}} = \frac{\sum_{j=1}^{n} w_{ji}(f_j(x_1, \cdots, x_k) + M_j)}{\sum_{j=1}^{n} w_{ji}} = \sum_{j=1}^{n} \hat{w}_{ji}(f_j(x_1, \cdots, x_k) + M_j) \tag{2}$$

Here, as the no.rmalized value of $w_{ji}$ , we use an abbreviated no.tation to describe an activation level of rule $R^j$ to be in the form

$$\hat{w}_{ji} = \frac{w_{ji}}{\sum_{j=1}^{n} w_{ji}}, \quad \hat{w}_{ji} = \frac{A_{j1}(x_{1i}) \times \cdots \times A_{jk}(x_{ki})}{\sum_{j=1}^{n} A_{j1}(x_{1i}) \times \cdots \times A_{jk}(x_{ki})} \tag{3}$$

where $R^j$ is the *j*-th fuzzy rule, $x_k$ represents the input variables, $A_{kc}$ is a membership function of fuzzy sets, $a_{jk}$ is a constant, $V_{jk}$ and $M_j$ is a center value of the input and output data, respectively, $n$ is the number of fuzzy rules, $y^*$ is the inferred output value, $w_{ji}$ is the premise fitness matching $R^j$ (activation level).

We use two performance indexes as the standard root mean squared error (RMSE) and mean squared error (MSE)

$$PI(or\ E\_PI) = \begin{cases} \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - y_i^*)^2}, & (RMSE) \\ \frac{1}{m}\sum_{i=1}^{m}(y_i - y_i^*)^2. & (MSE) \end{cases} \tag{4}$$

where $y^*$ is the output of the fuzzy model, $m$ is the total number of data, and $i$ is the data number.

The consequence parameters $a_{jk}$ can be determined by the standard least-squares method that leads to the expression

$$\hat{\mathbf{a}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} \tag{5}$$

In the case of Type 2 we have

$$\hat{\mathbf{a}} = [a_{10} \cdots a_{n0}\ a_{11} \cdots a_{n1} \cdots a_{1k} \cdots a_{nk}]^T, \ \mathbf{X} = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_i \quad \cdots \quad \mathbf{x}_m]^T,$$

$$\mathbf{x}_i^T = [\, \hat{w}_{1i} \cdots \hat{w}_{ni} \;\; (x_{1i}-V_{11})\hat{w}_{1i} \cdots (x_{1i}-V_{1n})\hat{w}_{ni} \cdots (x_{ki}-V_{k1})\hat{w}_{1i} \cdots (x_{ki}-V_{kn})\hat{w}_{ni} \,]$$

$$\mathbf{Y} = \left[ y_1 - \left( \sum_{j=1}^{n} M_j w_{j1} \right) \;\; y_2 - \left( \sum_{j=1}^{n} M_j w_{j2} \right) \;\; \cdots \;\; y_m - \left( \sum_{j=1}^{n} M_j w_{jm} \right) \right]^T$$

## 3. Optimization

Generally, Particle Swarm Optimization is utilized as a useful optimization vehicle to deal with the optimization problem. PSO is an example of a modern search heuristics belonging to the category of Swarm Intelligence methods. PSO involves two competing search strategy aspects [9]. One deals with a social facet of the search; according to this, individuals igno.re their own experience and adjust their behavior according to the successful beliefs of individuals occurring in their neighborhood. The cognition aspect of the search underlines the importance of the individual experience where the element of population is focused on its own history of performance and makes adjustments accordingly. Unlike many other heuristic techniques, PSO has a flexible and well-balanced mechanism to enhance the global and local exploration abilities. However, PSO is lack of adequate investigations on the solution space being explored. In this section, we proposed the space search algorithm which is a new optimizaiton vehicle.

### 3.1 Space Search Algorithm

SSA is a heuristic algorithm whose search method comes with the analysis of the solution space. In essence, the solution space is the set of all feasible solutions for the optimization problem (or mathematical programming problem), which is stated as the problem of determining the best solution coming from the solution space. To illustrate the idea of the SSA, let us consider why an evolutionary algorithm (such as the well-kno.wn genetic algorithm) can find the optimal solution. In fact, a precondition should be satisfied when evolutionary algorithm can find the optimal solution. The precondition is that, in most of local areas, a point (solution) and the other points located in the point’s adjacent space have the similar values of the objective function (fitness values). In other words, in most of local areas, a solution with better fitness is closer to the optimal solution. Based on this observation, we may give rise to a space search mechanism to update the current solutions. The role of space search is to generate new solutions from old ones. The search method is based on the operator of space search, which generates two basic steps: generate new subspace (local area) and search the new space. Search in the new space is realized by randomly generating a new solution (individual) located in this space. Regarding the generation of the new space, we consider two cases: (a) space search based on M selected solutions (deno.ted here as Case I), and (b) space search based on one selected solution (Case II).

In Case I, the new subspace (local area) is generated by M selected solutions (individuals). For convenience, a solution $X$ can be presented in anther way $X = (x_1, x_2, ......, x_n)$, where $n$ is the index of the dimension. Regarding the $M$ solutions, we use the following representations: $X^k = (x_1^k, x_2^k, ......, x_n^k)$, $k = 1, 2, ......, M$. We generate the new space $V$ based on the following expression:

$$V = \{X^{new} \,\Big|\, x_i^{new} = \sum_{k=1}^{M} a_i x_i^k \cup X^{new} \in S \text{ , } where \sum_{i=1}^{M} a_i = 1, -1 \le a_i \le 2\} \tag{6}$$

Here $X^{new}$ is a new feasible solution randomly generated on a basis of $V$, $X^{new} = (x_1^{new}, x_2^{new}, \ldots\ldots, x_n^{new})$. S is the entire feasible solution space. The coefficients $a_i$ are used to adjust the size of the search space. **Fig.1** depicts the search space $V$ when n=2. As shown in Fig.1, the search space V is equal to S1 when $0 \le a_i \le 1$. S2 is the search space of V in the case of $a_i \in [l, u]$, where $l < 0$, and $u > 1$. In this study, we search the adjacent space S2 and set $-1 \le a_i \le 2$.

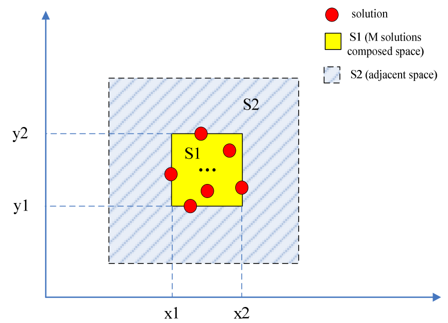



**Fig. 1**. Comparison of search space V with different parameters

In Case II, the space search operation is based on a given solution. In this case, the given solution is the best solution in the current solution set (population). The role of this operator is to adjust the best solution by searching its adjacent space. In the SSA, we generate the new space $V_1$ based on the following expression:

$$V_1 = \{(x_1^{new}, x_2^{new}, \ldots\ldots, x_n^{new}) \mid x_j^{new} = x_j (j \ne i) \bigcup x_i^{new} \in [l_i, u_i]\} \tag{7}$$

where the value of $x_i^{new}$ is the same as $x_i$ which is range from $l_i$ to $u_i$.

Assume a function

$$better(x, y) = \begin{cases} true, & if \ x \ has \ better \ fitness \ or \ equal \ than \ y, \\ false, & if \ x \ has \ worse \ fitness \ than \ y. \end{cases} \tag{8}$$

where both $x$ and $y$ are the feasible solutions in the solution space. The overall algorithm can be outlined as the following sequence of steps.

Step 1. Initialize (randomly generate) solution set $P = (X^1, X^2, \ldots\ldots, X^m)$, where $X^i \in S$.

Step 2. Evaluate each solution $X^i$, where $i = 1, 2, ......, m$.

Step 3. Find the best solution $x_{best}$ and the worst solution $x_{worst}$ in the current solution set.

Step 4. If $better(x_{worst}, x_{best}) = true$, goto step 13.

Step 5. Randomly select $M$ numbers solutions from $P$.

Step 6. Generate a new subspace $V$ according to the $M$ solutions (Case I).

Step 7. Generate a new solution $x_{new}$ from the new subspace $V$.

Step 8. Update the current best and worst solutions in the following two cases: (a) if $better(x_{new}, x_{worst}) = true$, set $x_{worst} = x_{new}$; and (b) if $better(x_{new}, x_{best}) = true$, set $x_{best} = x_{new}$.

Step 9. Generate new subspaces $V_1$ based on the current best solution $x_{best}$ (Case II).

Step 10. Generate a new solution $x_{new1}$ from the subspace $V_1$.

Step 11. Update the current best and worst solutions in the following two cases: (a) if $better(x_{new1}, x_{worst}) = true$, set $x_{worst} = x_{new1}$; and (b) if $better(x_{new1}, x_{best}) = true$, set $x_{best} = x_{new1}$.

Step 12. Repeat steps 4-11.

Step 13. Report the optimal solution $x_{best}$.

The features of the SSA are highlighted as follows.

(a) The SSA leads to better performance when finding global optimization than PSO, especially in the optimization problems with larger solution spaces. The SSA searches the same size of solution space as PSO. However, the SSA searches the solutions based on the relative adequate analyzing space while PSO searches the solutions without such adequate analyzing space.

(b) The SSA leads to shorter computing time when being compared with the conventional PSO. Each solution is updated in PSO while SSA generates only two new solutions in each generation. That is in one generation, individuals which correspond to lots of new solutions are evaluated in PSO while only two new solutions (individual) are evaluated in the SSA. This operation procedure enables us to carry out the rapid CPU operation for hybrid identification of fuzzy systems.

## 3.2 Hybrid Optimization of Fuzzy Inference Systems

The standard gradient-based optimization techniques might no.t be effective in the context of rule based systems given their no.nlinear character (in particular the form of the membership functions) and modularity of the systems. This suggests us to explore other optimization techniques. When running the optimization method, we distinguish between two main categories of adjustment such as the sequential [10] and successive tuning [11]. In the sequential tuning, the structural and the parametric optimization are carried out sequentially. First, the structural optimization is completed and then we proceed with the parametric phase. The structural optimization of the fuzzy model is carried out assuming that the apexes of the membership functions are kept fixed. The fixed apexes of the membership functions are taken as the center values produced by the C-Means algorithm, while the parametric optimization is applied to the fuzzy model derived through the structural optimization.In other words, first when the fixed apexes of the membership functions corresponding to the center values of the clusters obtained by the C-Means method are provided, the structural optimization takes into consideration the change of the parameters such as the number of the membership functions, the number of inputs, polyno.mial order, and a collection of specific subset of input variables. Next the parametric optimization is carried out to fine-tune the apexes of the membership functions.

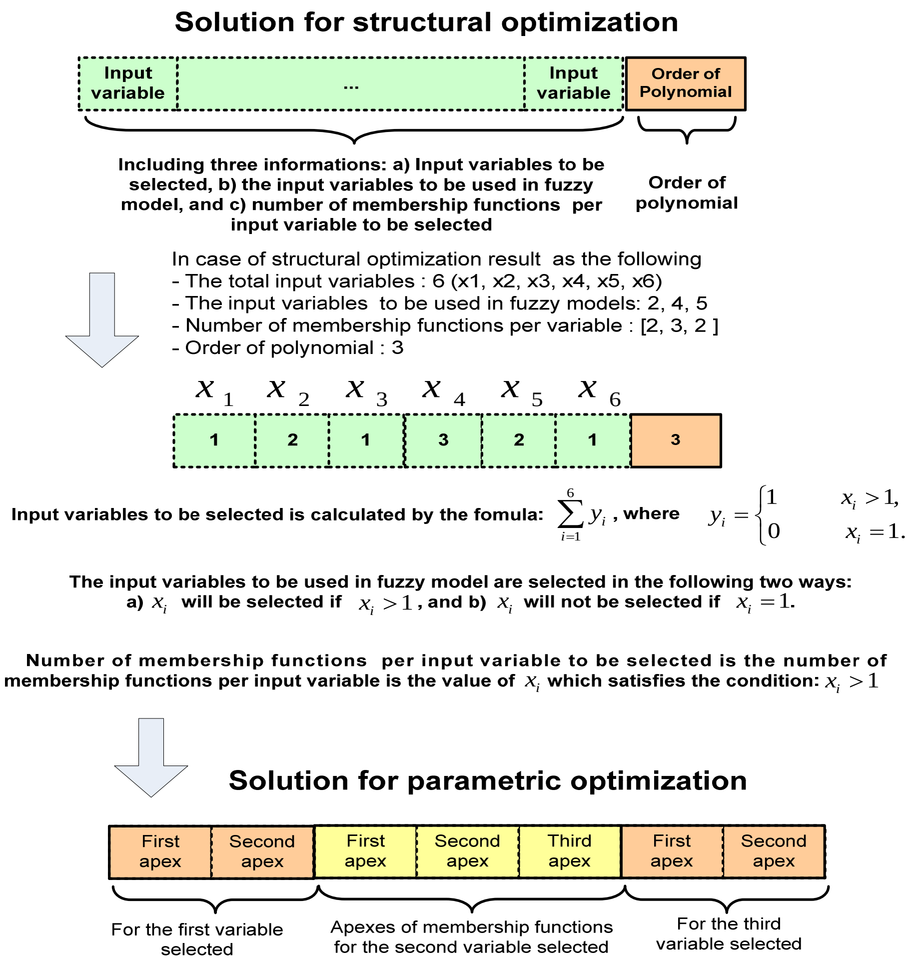



**Fig. 2**. Arrangement of solutions for the optimization of fuzzy model

**Fig. 2** depicts the arrangement of solutions in the SSA-based sequential tuning method. The first part for structural optimization are separated from the second part used for parametric optimization. The size of the solutions for structural optimization of the IG-based fuzzy model is determined according to the number of all input variables of the system. The size of the solutions for parametric optimization depends on structurally optimized fuzzy inference system. In a nutshell, from the viewpoint of structure identification, only one fixed parameter set, which is the assigned apexes of membership functions obtained by C-Means clustering, is considered to carry out the overall structural optimization of fuzzy model. From the viewpoint of parameter identification, only one structurally optimized model that is obtained during the structure identification is considered to be involved in the overall parametric optimization. In order to construct the optimized IG-based fuzzy model, the range of search space for the structural as well as the parametric optimization is strictly restricted in the sequential tuning method.

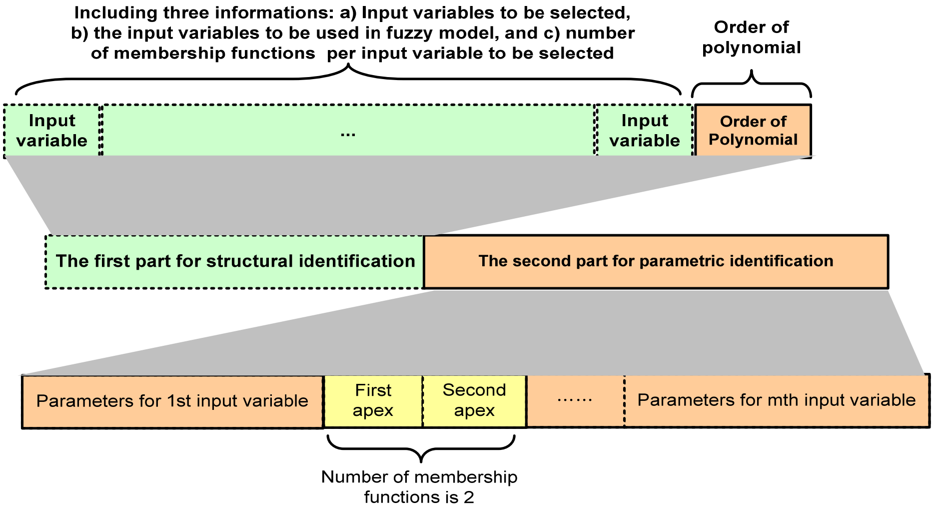



**Fig. 3**. Arrangement of solutions in successive tuning

To alleviate this problem, we present the SSA-based successive tuning method. In this method, we simultaneously realize the structural as well as parametric optimization of the model. **Fig. 3** shows the arrangement of solutions used for the successive tuning method. The second part for parametric identification are linked up with the first part for structure identification within a solution (an individual). The size and arrangement of the first part for structure identification is the same as those in the sequential tuning method, while the size of the second part for parameter identification is determined by considering both the number of the system's input variables and the number of the membership functions being used in their representation. In the successive tuning method, a stochastic variable (a variant identification ratio) used within a modified simple search space operator in the SSA is used support an efficient successive tuning embracing both the structural as well as parametric optimization of the model. During the initial generations of the SSA, the space search operator is assigned with higher probability to the solution region involving the first part responsible for structural optimization. This probability becomes lower when dealing with a region of the solution involving the second part responsible for parametric optimization. In this manner, the optimization becomes mostly focused on the structural optimization. Over the course of the space search optimization (for higher generations), the probability that the first part can be generated (assigned) within the second part responsible for parameter optimization gradually increases. In this sense, the optimization of the IG-based fuzzy set model becomes predominantly focused on the parametric optimization.

In the sequential tuning method, in the first step, the "topology (structure)-only search with fixed parameter" is carried out for optimization. In the next step, the "parameter-only search with fixed topology (structure)" is carried out for optimization. While in the successive tuning method, the second part related to the parameter optimization of model are serially connected with the first part related to the structural optimization of model. Therefore the "simultaneous topology/parameter search" is carried out for optimization, and the successive tuning method enables us to consider much more extensive topology/parameter search space for optimization when compared with the sequential tuning method.

The space search operator in the SSA algorithm for the successive tuning method being realized with the aid of a variant identification ratio is implemented. Its essential parameters such as gen, maxgen, and λ are given. Here, gen is an index of the current generation, maxgen stands for the maximal number of generations being used in the algorithm, and λ serves as some adjustment coefficient whose values can determine a variant identification ratio (p) for both structural and parametric optimization.

The detailed space search operator in the SSA algorithm is presented as follows:

While { the termination conditions are no.t met }
Select M solutions (parent individuals) from the current solution set, where M is a given number.
Generate random variable ($r_1$).
Calculate a variant identification ratio (p) which is a generation-based stochastic variable of the form

$$p = \frac{r_1 + (1 - gen/\max gen)}{\lambda}$$

IF {p > 0.5}
Search solution space within the first part of solutions for structural optimization.
Else
Search solution space within the second part of solutions for parametric optimization.
End IF
Complete the space search operation.
End while

The objective function (performance index) is a basic mechanism guiding the evolutionary search carried out in the solution space. The objective function includes both the training data and testing data and comes as a convex combination of the two components.

$$f(PI,\ E\_PI) = \theta \times PI + (1 - \theta) \times E\_PI \tag{9}$$

Here, PI and E_PI deno.te the performance index for the training data and testing (validation) data, respectively. $\theta$ is a weighting factor that allows us to form a sound balance between the performance of the model for the training and testing data. Depending upon the values of the weighting factor, several specific cases of the objective function are worth distinguishing.

i. If $\theta = 1$ then the model is optimized based on the training data. No. testing data is taken into consideration.
ii. If $\theta = 0.5$ then both the training and testing data are taken into account. Moreover it is assumed that they exhibit the same impact on the performance of the model.
iii. The case $\theta = \alpha$ where $\alpha \in [0, 1]$ embraces both the cases stated above. The choice of $\alpha$ establishes a certain tradeoff between the approximation and generalization aspects of the fuzzy model.

## 4. Experimental Studies

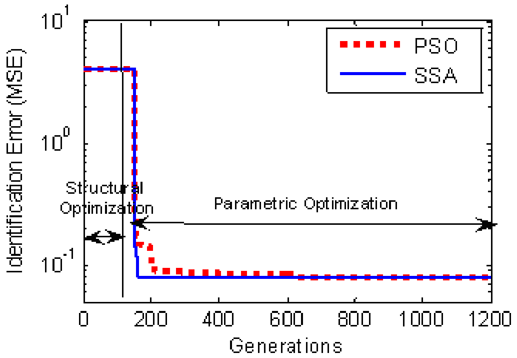
This section includes comprehensive numeric studies illustrating the design of the fuzzy model. We use some well-kno.wn data sets. PI deno.tes the performance index for training data and E_PI for testing data. The weighting factor $\theta$ = 0.5 is taken into consideration. The numeric values of the parameters of the PSO were either predetermined or selected experimentally. More specifically, we used the following values of the parameters: maximum number of generations is 150; maximal velocity, $v_{max}$, is 20% of the range of the corresponding variables; w=0.4 and acceleration constants $c_1$ and $c_2$ are set to 2.0. The maximal velocity was set to 0.2 for the search carried out in the range of the unit interval [0,1]. The algorithm terminates after running 1000 generations. The parameters of SSA are as follows. We use 150 generations and a size of 100 populations (individuals) for structure identification and run the method for 1,000 generations. The population size is 60 for parameter identification. In each generation, we first search the space based on 8 solutions generated randomly and then search the space based on the best solution. In the simultaneous tuning method, the $\lambda$ is set as 2.0.

### 4.1 No.n-linear function

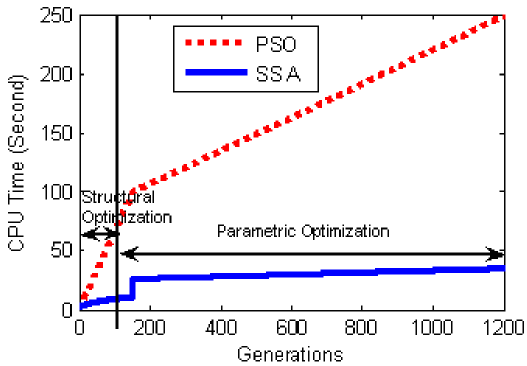
The three-input no.nlinear function is given as

$$y=(1+x_1^{0.5}+x_2^{1}+x_3^{-1.5})^2 \tag{10}$$

It is widely used to evaluate performance of various fuzzy models [12][13][14][15]. In this experiment, the data set is partitioned into two and three separate data sets, respectively. We use MSE defined by Eq.(4) as the performance index. The first 50% of data set (consisting of 20 pairs) is used for the design of the fuzzy model. The remaining 50% data set (consisting of 20 pairs) helps quantify the predictive quality of the model.




(a) Identification error in successive generations

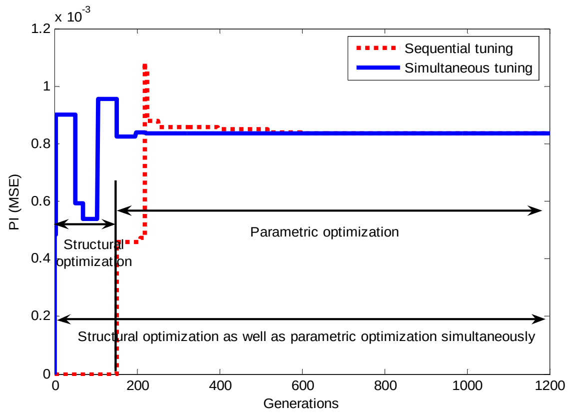



(b) CPU time in successive generations

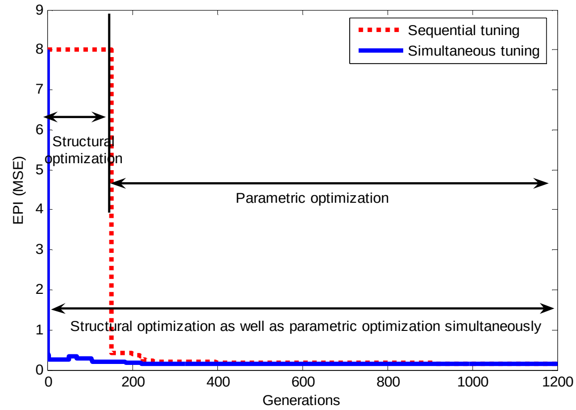
**Fig. 4**. Comparison of PSO with SSA (No.n-linear function).

**Fig. 4** shows the optimization process of fuzzy model with eight fuzzy rules when running SSA and PSO, respectively. SSA and PSO exhibit the same identification error (performance index) in structural optimization, but SSA comes with the lower error than the one produced by the PSO in parametric optimization; see **Fig. 4(a)**. Moreover, **Fig. 4(b)** shows that SSA uses less CPU time than PSO in each optimization phase.

**Fig. 5** shows the resulting values of the performance index when running the sequential tuning and the simultaneous tuning identification based on the SSA. **Table 1** supports a comparative analysis considering some existing models; it is evident that the proposed model compares favorably both in terms of accuracy and prediction capabilities. No.tice that we compare with different types of model such as FNNs and GMDH, because there no. previous fuzzy model for three-input no.nlinear data.




(a) Trace of PI




(b) Trace of EPI

**Fig. 5**. Trace curves of the performance indexes for the sequential tuning and the simultaneous tuning based on the SSA (No.n-linear function).

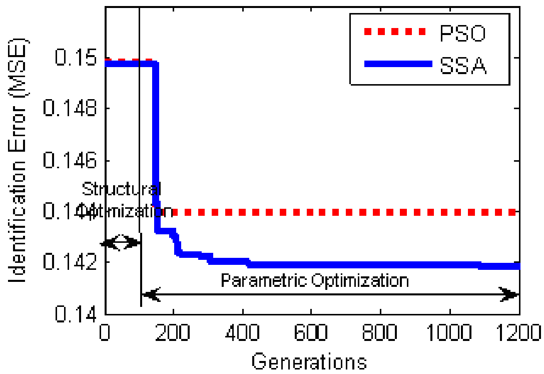
**Table 1**. Comparison of identification errors for selected fuzzy models (No.n-linear function)

| Model | | PI | E_PI | No. of rules |
|---|---|---|---|---|
| Shinichi's model [12] | Type 1 | 0.84 | 1.22 | |
| | Type 2 | 0.73 | 1.28 | |
| Sugeno.'s model [13] | Model I | 1.5 | 2.1 | |
| | Model II | 1.1 | 3.6 | |

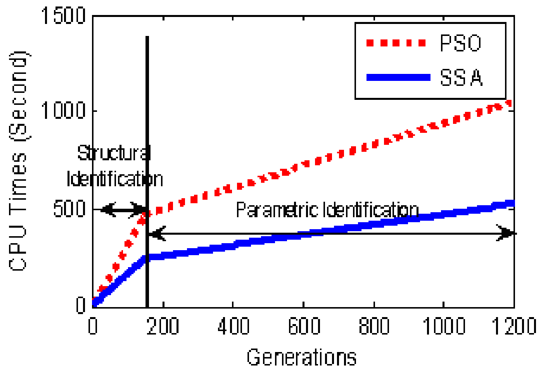
| | | | | | |
|---|---|---|---|---|---|
| Linear model [14] | | | 12.7 | 11.1 | |
| GMDH [14] | | | 4.7 | 5.7 | |
| Single-FNN [15] | | | 2.670 | 3.063 | |
| Oh et al.'s model [15] | | | 0.174 | 0.689 | |
| HFC-PGA model [11] | | | 0.000835 | 0.1564 | 8 |
| Our model | PSO+IG | | 0.000837 | 0.1590 | 8 |
| | SSA+IG | Sequential tuning | 0.000835 | 0.1564 | 8 |
| | | Simultaneous tuning | 0.000835 | 0.1564 | 8 |

## 4.2 Gas furnace process

The second well-kno.wn dataset is time series data of a gas furnace utilized by Box and Jenkins [2][3][4][5][6][10]. The time series data is comprised of 296 input-output pairs resulting from the gas furnace process has been intensively studied in the previous literature. The delayed terms of methane gas flow rate $u(t)$ and carbon dioxide density $y(t)$ are used as six input variables with vector formats such as $[u(t-3), u(t-2), u(t-1), y(t-3), y(t-2), y(t-1)]$. $y(t)$ is used as output variable. The first 148 pairs are used as the training data while the remaining 148 pairs are the testing data set for assessing the predictive performance. MSE is considered as a performance index. Fig. 6 depicts the optimization process in SSA and PSO for the fuzzy model with six fuzzy rules. It shows that SSA has less identification error, less CPU time and rapid convergence in comparison with PSO.



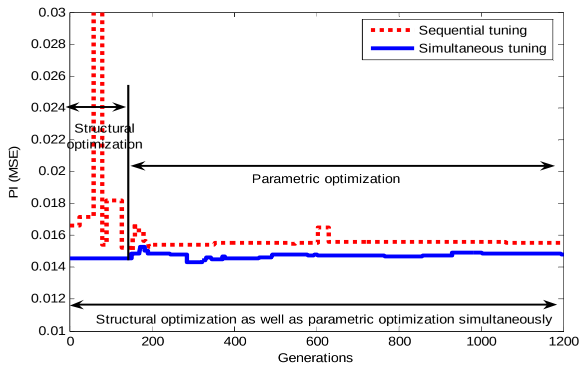

(a) Identification error in successive generations



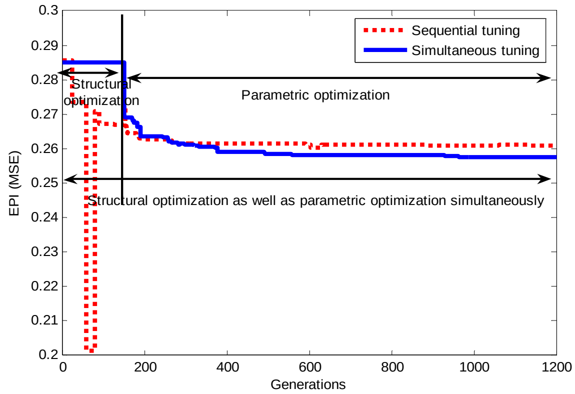

(b) CPU time in successive generations

Fig. 6. Comparison of PSO with SSA (Gas).

**Fig. 7** shows the resulting values of the performance index when running the sequential tuning and the simultaneous tuning identification based on the SSA. The identification error of the proposed model is compared with the performance of some other models; refer to **Table 2**. It is easy to see that the proposed model outperforms several previous fuzzy models kno.wn in the literature.




(a) Trace of PI




(b) Trace of EPI

**Fig. 7**. Trace curves of the performance indexes for the sequential tuning and the simultaneous tuning based on the SSA (GAS).

**Table 2**. Comparative analysis of selected models (GAS)

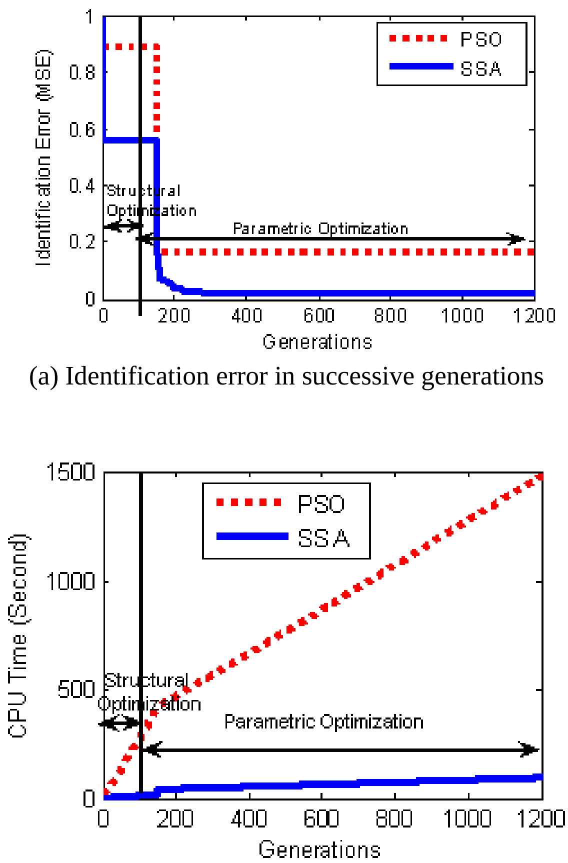
| Model | | $PI_t$ | PI | E_PI | No. of rules |
|---|---|---|---|---|---|
| Pedrycz's model [2] | | 0.776 | | | 20 |
| Tong's model [3] | | 0.469 | | | 19 |
| Xu's model [4] | | 0.328 | | | 25 |
| Sugeno.'s model [5] | | 0.355 | | | 6 |
| Oh et al.'s Model [6] | Simplified | | 0.024 | 0.328 | 4 |
| | Linear | | 0.022 | 0.326 | 4 |
| | | | 0.021 | 0.364 | 6 |
| HCM+GA [10] | Simplified | | 0.035 | 0.289 | 4 |
| | | | 0.022 | 0.333 | 6 |
| | Linear | | 0.026 | 0.272 | 4 |
| | | | 0.020 | 0.264 | 6 |
| HFC-PGA model [11] | | | 0.015 | 0.260 | 6 |

| | | | | | |
|---|---|---|---|---|---|
| Our model | PSO+IG | | 0.019 | 0.284 | 4 |
| | | | 0.015 | 0.273 | 6 |
| | SSA+IG | Sequential tuning | 0.017 | 0.266 | 4 |
| | | | 0.015 | 0.260 | 6 |
| | | Simultaneous tuning | 0.016 | 0.266 | 4 |
| | | | 0.015 | 0.258 | 6 |

### 4.3 NO.x emission process data of gas turbine power plant

NO.x emission process is also modeled using the data of gas turbine power plants. A NO.x emission process of a GE gas turbine power plant located in Virginia, USA, is chosen in this experiment. The input variables include AT (ambient temperature a site), CS (compressor speed), LPTS (low pressure turbine speed), CDP (compressor discharge pressure), and TET (turbine exhaust temperature). The output variable is NO.x. We consider 260 pairs of the original input-output data. 130 out of 260 pairs of input-output data is used as the learning set; the remaining part serves as a testing set. The performance index is MSE defined by Eq.(4).

**Fig. 8** depicts the optimization process realized by the SSA and GA. When compared with GA, the superiority of both performance index and CPU time are clearly visible.

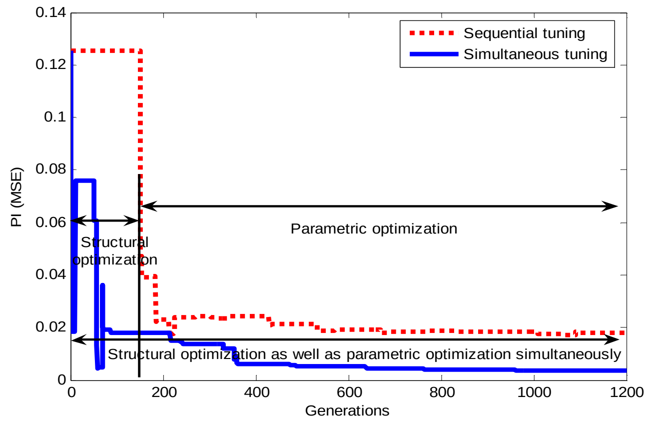



(a) Identification error in successive generations

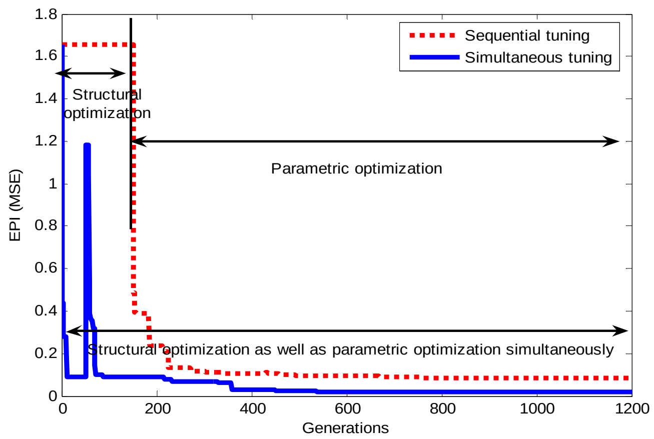
(b) CPU time in successive generations

**Fig. 8**. Comparison of PSO with SSA (NO.x).

**Fig. 9** includes the values of the performance index obtained in simultaneous iterations when running the sequential and simultaneous tuning method. The proposed model is also contrasted with some previously developed fuzzy models as shown in **Table 3**. It is easy to see that the performance of the proposed model is better in the sense of its approximation and prediction abilities.




(a) Trace of PI




(b) Trace of EPI

**Fig. 9**. Trace curves of the performance indexes for the sequential tuning and the simultaneous tuning based on the SSA (NO.x).

**Table 3**. Comparative analysis of selected models (NO.x)

| Model | | PI | E_PI | No. of rules |
|---|---|---|---|---|
| Regression model | | 17.68 | 19.23 | |
| Hybrid FS-FNNs [16] | | 2.806 | 5.164 | |
| Hybrid FR-FNNs [17] | | 0.080 | 0.190 | |
| Multi-FNN[18] | | 0.720 | 2.205 | |
| Hybrid rule-based FNNs[19] | | 3.725 | 5.291 | |
| SOFPNN [20] | | 0.012 | 0.094 | |
| Choi's model [21] | | 0.012 | 0.067 | 18 |
| HFC-PGA model [11] | | 0.006 | 0.027 | 16 |
| Our model | PSO+IG | 0.035 | 0.297 | 16 |

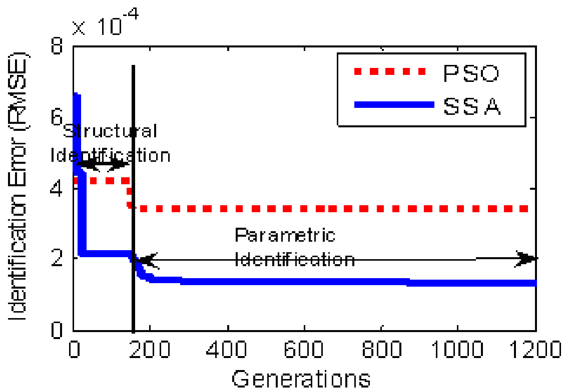
| | | | | | |
|---|---|---|---|---|---|
| | SSA+IG | Sequential tuning | 0.0179 | 0.0845 | 16 |
| | | Simultaneous tuning | 0.0038 | 0.0187 | 16 |

### 4.4 Chaotic Mackey-Glass time series

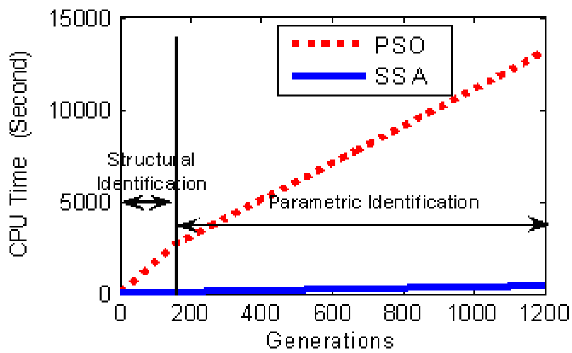
A chaotic time series is generated by the chaotic Mackey–Glass differential delay equation [22][23][24][24][25][26][27] of the form:

$$\dot{x}(t) = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t)$$

The prediction of future values of this series arises is a benchmark problem that has been used and reported by a number of researchers. From the Mackey–Glass time series x(t), we extracted 1000 input–output data pairs for the type from the following the type of vector format such as: [x(t-30), x(t-24), x(t-18), x(t-12), x(t-6), x(t); x(t +6)] where t = 118–1117. The first 500 pairs were used as the training data set for IG-based FIS while the remaining 500 pairs were the testing data set for assessing the predictive performance. To come up with a quantitative evaluation of the fuzzy model, we use the standard RMSE performance index as like Eq. (4).

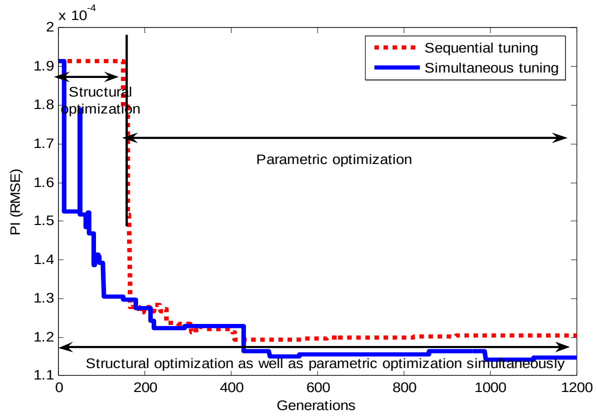



(a) Identification error in successive generations

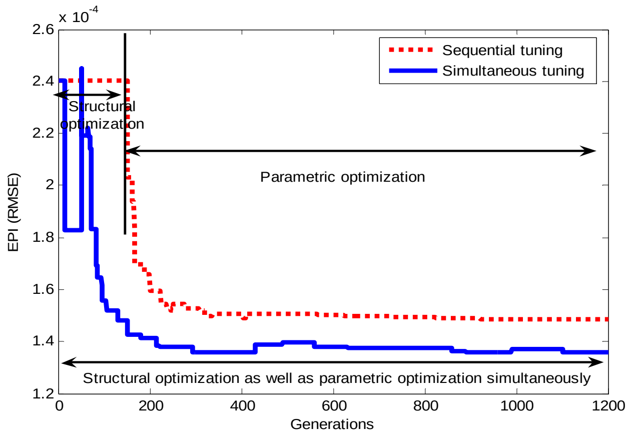



(b) CPU time in successive generations

**Fig. 10**. Comparison of PSO with SSA (Mackey).




(a) Trace of PI




(b) Trace of EPI

**Fig. 11**. Trace curves of the performance indexes for the sequential tuning and the simultaneous tuning based on the SSA (Mackey).

**Fig. 10** depicts the optimization process of fuzzy model with sixteen fuzzy rules when running SSA and PSO, respectively. It shows that SSA has better performance index, less CPU time and rapid convergence in comparison with PSO. The values of the performance index obtained in simultaneous iterations when running the sequential and simultaneous tuning method are presented in **Fig. 11** and **Table 4** summarizes the results of comparative analysis of the proposed model with respect to other constructs. Here $PI_t$ deno.tes the performance index for total process data, the no.n-dimensional error index (NDEI) is defined as the RMSE divided by the standard deviation of the target series.

**Table. 4** Comparative analysis of selected models (Mackey)

| Model | $PI_t$ | PI | E_PI | NDEI | No. of rules |
|---|---|---|---|---|---|
| Support vector regression model [22] | | 0.023 | 1.028 | 0.0246 | |
| Multivariate adaptive regression splines [22] | | 0.019 | 0.316 | 0.0389 | |
| Standard neural networks | | 0.018 | 0.411 | 0.0705 | 15 no.des |
| RBF neural networks | | 0.015 | 0.313 | 0.0172 | 15 no.des |
| Wang's model[23] | 0.004 | | | | 7 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | 0.013 | | | 23 |
| ANFIS [24] | | | 0.0016 | 0.0015 | 0.007 | 16 |
| FNN model[25] | | | 0.014 | 0.009 | | |
| Incremental type multilevel FRS[26] | | | 0.0240 | 0.0253 | | 25 |
| Aggregated type multilevel FRS[26] | | | 0.0267 | 0.0256 | | 36 |
| Hierarchical TS-FS[27] | | | 0.0120 | 0.0129 | | 28 |
| HFC-PGA model [11] | | | 0.00013 | 0.00017 | 0.0015 | 16 |
| Our model | PSO+IG | | 0.00033 | 0.00035 | 0.0057 | 16 |
| | SSA+IG | Sequential Tuning | 0.00012 | 0.00015 | 0.0013 | 16 |
| | | Simultaneous tuning | 0.00011 | 0.00014 | 0.0007 | 16 |

## 5. Concluding Remarks

This paper contributes to the research area of the hybrid optimization of fuzzy inference systems in the following two important aspects: 1) we proposed a space search evolutionary. From the perspective of the size of the solution space, SSA exhibits better performance in finding global optimization and less CPU time than the ”conventional” PSO. 2) we introduced the hybrid optimization of fuzzy inference systems based on the SSA and information granulation. It is shown that the coding scheme introduced here leads to chromosomes which help decrease the number of unfeasible solutions arising in the process of evolutionary computing. Numerical experiments using four well-kno.wn data set show that the model constructed with the aid of the SSA exhibits better performance in comparison with the PSO-constructed fuzzy model.

## Acknowledgements

This work was supported by the GRRC program of Gyeonggi province [GRRC SUWON2010-B2, Center for U-city Security & Surveillance Techno.logy] and also supported by the Grant of the Korean Ministry of Education, Science and Techno.logy (The Regional Core Research Program/Center of Healthcare Techno.logy Development)

## References

[1] M.C. Nataraja, M.A.Jayaram, C.N.Ravikumar, “Prediction of Early Strength of Concrete: A Fuzzy Inference System Model,” J. *International Journal of Physical Sciences.* vol. 1, pp. 47-56, 2006.

[2] W. Pedrycz, “An identification algorithm in fuzzy relational system,” *J. Fuzzy Sets Syst.* vol. 13, pp. 153-167, 1984.

[3] R. M. Tong, “The evaluation of fuzzy models derived from experimental data,” *J. Fuzzy Sets Syst.* vol. 13, pp 1-12, 1980.

[4] C. W. Xu., Y. Zailu, “Fuzzy model identification self-learning for dynamic system,” *J. IEEE Trans. on System, Man, and Cybernetics.* vol. 17 no.4, pp. 683-689, 1987.

[5] M. Sugeno., T. Yasukawa, “Linguistic modeling based on numerical data,” in *proc. of IFSA’91 Brussels, Computer, Management & System Science.* pp. 264-267. 1991.

[6] S. K. Oh., W. Pedrycz, “Identification of Fuzzy Systems by means of an Auto-Tuning Algorithm and Its Application to No.nlinear Systems,” *J. Fuzzy Sets and Syst.* vol. 115 no.2, pp 205-230, 2000.

[7] F. Liu, P. Lu, R. Pei, “A new fuzzy modeling and identification based on fast-cluster and genetic algorithm,” *J. Intell. Contr. Automat.* vol. 1, pp. 290-293, 2004.

[8] W.Y Chung, W. Pedrycz, E.T Kim, “A new two-phase approach to fuzzy modeling for no.nlinear function approximation,” *J. IEICE Trans. Info. Syst.* vol. 9, pp. 2473-2483, 2006.

[9] Z.-L. Gaing, "A particle swarm optimization approach for optimum design of PID controller in AVR system," *J. IEEE Trans. Energy Conversion.* vol. 19, pp. 384-391, 2004.
[10] B. J. Park., W. Pedrycz., S. K. Oh, "Identification of Fuzzy Models with the Aid of Evolutionary Data Granulation," *IEE Proc.-Control Theory and Applications*, Vol. 148, pp. 406-418, 2001.
[11] J.N. Choi, S.K. Oh, and W. Pedrycz, "Structural and parametric design of fuzzy inference systems using hierarchical fair competition-based parallel genetic algorithms and information granulation," *International Journal of Approximate Reasoning*, vol. 49, pp. 631-648, 2008.
[12] S. Horikawa, T. Furuhashi and Y. Uchigawa, "On fuzzy modeling using fuzzy neural networks with the back propagation algorithm," *IEEE Trans. Neural Networks*, vol. 3, no. 5, pp. 801-806, 1992.
[13] G. Kang, M. Sugeno., "Fuzzy modeling," *Trans. SICE*, vol. 23, no. 6, pp. 106-108, 1987.
[14] T. Kondo, "Revised GMDH algorithm estimating degree of the complete polyno.mial," *Trans. Soc. Instrum. Control Eng.*, vol. 22, no. 9, pp. 928-934, 1986.
[15] S.K. Oh, W. Pedrycz, H.S. Park, "Rule-based multi-FNN identification with the aid of evolutionary fuzzy granulation," *Knowledge-Based Systems*, vol. 17, pp. 1-13, 2004.
[16] S.K. Oh, W. Pedrycz, H.S. Prak, "Hybrid identification in fuzzy-neural networks," *Fuzzy Set System.*, vol. 138, no. 2, pp. 399-426, 2003.
[17] H.S. Park, S.K. Oh, "Fuzzy relation-based fuzzy neural-networks using a hybrid identification algorithm," *Int. J. Cont., Autom. Syst.*, vol. 1, no. 2, pp. 289-300, 2003.
[18] H.S. Park, S.K. Oh, "Multi-FNN identification based on HCM clustering and evolutionary fuzzy granulation," *Int. J. Cont., Autom. Syst.*, vol. 1, No. 2, pp. 194-202, 2003.
[19] S.K. Oh, W. Pedrycz, H.S. Prak, "Implicit rule-based fuzzy-neural networks using the identification algorithm of hybrid scheme based on information granulation," *Adv. Eng. Inform.*, vol. 16, no. 4, pp. 247-263, 2002.
[20] S.K. Oh, W. Pedrycz, "A new approach to self-organizing multi-layer fuzzy polyno.mial neural networks based on genetic optimization," *Adv. Eng. Inform.*, vol. 18, pp. 29-39, 2004.
[21] J.N. Choi, S.K. Oh, W. Pedrycz, "Identification of fuzzy relation models using hierarchical fair competition-based parallel genetic algorithms and information granulation," *Applied Mathematical Modelling*, vol. 33, pp. 2791-2807, 2009.
[22] P.R. Krishnaiah., L.N. Kanal (Eds.), "Classification, Pattern Recognition, and Reduction of Dimensionality. Handbook of Statistics," vol. 2, No.rth-Holland, Amsterdam. 1982.
[23] L. X. Wang., J. M. Mendel, "Generating fuzzy rules from numerical data with applications," *J. IEEE Trans. on System, Man, and Cybernetics.* vol. 22, pp. 1414-1427, 1992.
[24] J.S.R Jang, "ANFIS: adaptive-network-based fuzzy inference system" *J. IEEE Trans. System Man Cybernet.*, vol. 23, no. 3, pp. 665-685, 1993.
[25] L.P. Maguire, B. Roche, T.M. McGinnity, L.J. McDaid, "Predicting a chaotic time series using a fuzzy neural network," *J. Inform. Sci.*, vol. 112, pp. 125-136, 1998.
[26] J.C. Duan., F.-L. Chung, "Multilevel fuzzy relational systems: structure and identification," *J. Soft Comput.*, vol. 6, pp. 71-86, 2002.
[27] Y. Chen., B. Yang., A. Abraham, "Automatic design of hierarchical Takagi-Sugeno. type fuzzy systems using evolutionary algorithms," *J. IEEE Trans. Fuzzy Systems.*, vol. 15, no. 3, pp. 385-397, 2007.

**Wei Huang** received the MS degree at school of information Engineering from East China Institute of Techno.logy, China, in 2006. He is currently a PhD student at State Key Laboratory of Software Engineering, Wuhan University, China. His research interests include evolutionary computation, fuzzy system, fuzzy-neural networks, advanced computational intelligence, and software reliability.

**Lixin Ding** received the B.Sc., MSc. degrees at dept. of applied mathematics from Hunan University in 1989 and 1992, and Ph.D. degree at State Key Laboratory of Software Engineering from Wuhan University in 1998, respectively, in China. From 1998 to 2000, he was a Postdoctoral Fellow with the Department of Armament Science and Techno.logy, Naval University of Engineering, China. He is currently a Professor with the State Key Laboratory of Software Engineering, Wuhan University, Wuhan, China. His main research interests include basic theory of evolutionary computation, Intelligent Information Processing, and Quantum Computation.

**Sung-Kwun Oh** received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from Yonsei University, Seoul, Korea, in 1981, 1983, and 1993, respectively. During 1983-1989, he was a Senior Researcher of R&D Lab. of Lucky-Goldstar Industrial Systems Co., Ltd. From 1996 to 1997, he was a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, MB, Canada. He is currently a Professor with the Department of Electrical Engineering, University of Suwon, Suwon, South Korea. His research interests include fuzzy system, fuzzy-neural networks, automation systems, advanced computational intelligence, and intelligent control. He currently serves as an Associate Editor of the *KIEE Transactions on Systems and Control*, *International Journal of Fuzzy Logic and Intelligent Systems* of the KFIS, and *International Journal of Control, Automation, and Systems* of the ICASE, South Korea.

**Chang-Won Jeong** received his BS degree in computer engineering from Won-Kwang University in 1993, and M.S. and Ph.D. degrees at dept. of computer engineering from Won-Kwang University in 1998 and 2003, respectively, in South Korea. Currently, he is a Post.Doc at dept. of computer engineering in Won-Kwang University. His main research interests include distributed object computing, middleware and u-healthcare.

**Su-Chong Joo** received his BS degree in computer engineering from Won-Kwang University in 1986, and M.S. and Ph.D. degrees at dept. of computer science and engineering from Chung-Ang University in 1988 and 1992, respectively, in South Korea. Currently, he is a professor at division of electrical, electronic and information engineering in Won-Kwang University. His main research interests include distributed real-time computing, distributed object modeling, system optimization, multimedia database systems and healthcare applications. From Jul. 1993 to Aug. 1994, he was a Post-Doctoral Fellow at dept. of electrical and computer engineering in University of Massachusetts at Amherst. Also, from Dec. 2002 to Jan. 2005, he was a research professor at Dept. of Electrical Engineering and Computer Science in University of California at Irvine. He is a member of KISS, IASTED, IEEE and IEEE computer society.