Journal of Electrical Engineering & Technology Vol. 6, No. 6, pp. 853~866, 2011 ***853***
http://dx.doi.org/10.5370/JEET.2011.6.6.853

Identification of Fuzzy Inference Systems Using a Multi-objective Space Search Algorithm and Information Granulation

Wei Huang, Sung-Kwun Oh†, Lixin Ding*, Hyun-Ki Kim*** and Su-Chong Joo§**

Abstract – We propose a multi-objective space search algorithm (MSSA) and introduce the identification of fuzzy inference systems based on the MSSA and information granulation (IG). The MSSA is a multi-objective optimization algorithm whose search method is associated with the analysis of the solution space. The multi-objective mechanism of MSSA is realized using a non-dominated sorting-based multi-objective strategy. In the identification of the fuzzy inference system, the MSSA is exploited to carry out parametric optimization of the fuzzy model and to achieve its structural optimization. The granulation of information is attained using the C-Means clustering algorithm. The overall optimization of fuzzy inference systems comes in the form of two identification 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 the polynomial type) and parameter identification (viz. the apexes of membership function). The structure identification is developed by the MSSA and C-Means, whereas the parameter identification is realized via the MSSA and least squares method. The evaluation of the performance of the proposed model was conducted using three representative numerical examples such as gas furnace, NOx emission process data, and Mackey-Glass time series. The proposed model was also compared with the quality of some "conventional" fuzzy models encountered in the literature.

Keywords: Multi-objective Space Search Algorithm (MSSA), Information Granulation (IG), Least squares Method (LSM), Fuzzy Inference System (FIS)

1. Introduction

Over the past decades, fuzzy modeling has been used in many fields of engineering, particularly in medical engineering, and even in the social sciences. There have been a number of diverse approaches to fuzzy modeling. In the early 1980s, linguistic modeling [1] was proposed as a primordial identification method for fuzzy models. Tong et al. [2] and Xu et al. [3] examined different approaches for fuzzy models. While appealing with respect to basic topology (a modular fuzzy model composed of a series of rules) [4], these models still need formal solutions in terms of structural optimization, say, a construction of the underlying fuzzy sets or the information granules viewed as the basic building blocks of any fuzzy model. Oh and Pedrycz [5] proposed some enhancements to the model, but the problem of finding "good" initial parameters for the fuzzy set in the rules remains open. Since then, several genetical identification methods for fuzzy models have been proposed. Chung and Kim [6] and others [7] discussed the use of genetic algorithms for fuzzy models. However, while all the methods reported above are based on information granulaton (IG) and optimization algorithms, there is still a lack of adequate investigation on the solution space explored.

There are two main and conflicting objectives in the design of fuzzy systems: accuracy and complexity. The objective of any effective learning method is to develop an accurate, simple, and interpretable fuzzy model. In the 1990s, the emphasis of modeling was placed on accuracy maximization. Various approaches have been proposed to improve the accuracy of fuzzy models using evolutionary and population-based optimization such as Genetic Algorithms (GA) and Particle Swarm Optimization [8-11]. These methods usually help improve the accuracy of the resulting fuzzy model. The complexity of the model increases as a result of the accuracy maximization. Some researchers have attempted to optimize simultaneously the accuracy and complexity of fuzzy models [12-13]. However, this objective is impossible to achieve due to the existence of the accuracy-complexity tradeoff. Recently, accuracy maximization and complexity minimization have often been cast in the setting of multi-objective optimization. A number of evolutionary algorithms (EAs)

† Corresponding Author: Dept. of Electrical Engineering, The University of Suwon, Korea. (ohsk@suwon.ac.kr)
* State Key Laboratory of Software Engineering, Wuhan University, China.
** School of Computer and Communication Engineering, Tianjin Univerity of Technology, China. (huangwabc@163.com)
*** Dept. of Electrical Engineering, The University of Suwon, Korea
§ Dept. of Computer Engineering, Wonkwang University, Korea. (scjoo@wonkwang.ac.kr)
Received: September 7, 2010; Accepted: March 11, 2011

have been developed to solve multi-objective optimization problems such as Micro-GA [14], NSGA-II [15], and so on [16-18]. These EAs are population-based algorithms which may explore different portions of the Pareto front simultaneously. As a result, multi-objective optimization (MOO) techniques have been applied to design fuzzy systems exhibiting high accuracy and significant interpretability [19, 20]. Nevertheless, when dealing with the IG-based fuzzy model, previous studies lack an optimization vehicle which considers not only the solution space being explored but also the techniques of MOO.

In the current study, we present a multi-objective space search algorithm (MSSA) and introduce a fuzzy identification of fuzzy inference systems using the MSSA and IG. MSSA was also used to maximize the accuracy of the Information Granule-based Fuzzy Inference System (IG-FIS) and to solve the problems presented earlier. The optimization is of multi-objective character so we have to deal with the simplicity and accuracy of the model. To reflect the multi-objective character of the design, we consider the mean squared error (root mean squared error) to quantify accuracy, structure complexity, and the total number of polynomial coefficients in the consequence part of the fuzzy rules in order to demostrate the simplicity of the model. The optimization process consists of two identification phases: structure identification and parameter identification. Information granulation is realized using Hard C-Means (HCM), the MSSA, and the Least squares Method (LSM). HCM is used to determine the initial parameters of the fuzzy model such as the initial location of the apexes of the membership functions and the prototypes of the polynomial functions used in the premise and consequence parts of the fuzzy rules. On the other hand, the MSSA and LSM are used to adjust the initial values of the parameters. In short, the MSSA is employed to carry out parametric optimization of the fuzzy model and to realize its structural optimization.

The organization of the paper reflects its main objectives. Section 2 introduces the design of the IG-FIS. Section 3 presents the MSSA and a multi-objective optimization of IG-FIS using the MSSA. Section 4 reports the experimental results. Finally, Section 5 presents the conclusions.

2. A Design of the IG-Based Fuzzy Model

In essence, information granules are viewed as highly related collections of objects (particularly data points) drawn together by some criteria of proximity, similarity, or functionality. Granulation of information is an inherent and omnipresent activity of human beings carried out with the aim of obtaining a better and more effective insight into a problem and arriving at its efficient solution. In particular, granulation of information aims to break down the problem at hand into smaller ones, making it easier to solve. This way, we have to partition the task into a series of well-defined subproblems (modules) with a far lower computational complexity than the original one. The identification procedure for fuzzy models is categorized into identification activities that deal with the development of the premise and the consequence part of rules. The identification completed at the premise level consists of two main steps. First, we select the input variables x_1, x_2, …, x_k of the rules. Second, we form fuzzy partitions (by specifying fuzzy sets of well-defined semantics such as low, high, etc.) of the spaces over which these individual variables are defined. In this sense, this phase is all about information granulation of the elements of the fuzzy partitions we are interested in when developing any rule-based model. The number of fuzzy sets constructed implies directly the number of rules of the model itself. In addition, the membership functions of the information granules have to be determined.

The identification of the premise part is completed in the following manner:

Given a set of data U={x_1, x_2, …, x_l ; y}, where $x_k=[x_{1k}, \ldots, x_{mk}]^T$, $y=[y_1, \ldots, y_m]^T$, l is the number of variables, and m is the number of data.

[Step 1] Arrange a set of data **U** into data set $\mathbf{X}_k$ composed of the corresponding input and output data.

$$\mathbf{X}_k=[\mathbf{x}_k ; \mathbf{y}] \tag{1}$$

[Step 2] Run the K-Means to determine the centers (prototypes) $\mathbf{v}_{kg}$ within the data set $\mathbf{X}_k$.

[Step 2-1] Arrange data set $\mathbf{X}_k$ into c-clusters (in essence, this is effectively the granulation of information)

[Step 2-2] Calculate the centers $\mathbf{v}_{kg}$ of each cluster.

$$\mathrm{v}_{kg} = \{v_{k1},\ v_{k2}, \ldots, v_{kc}\} \tag{2}$$

[Step 3] Partition the corresponding input space using the prototypes of the clusters $\mathbf{v}_{kg}$. Associate each cluster with some meanings (semantics) such as small, large, and so on.

[Step 4] Set the initial apexes of the membership functions using the prototypes $\mathbf{v}_{kg}$.

For consequence identification, we consider the initial values of the polynomial functions based on the information granulation realized for the consequence and premise part.

[Step 1] Find a set of data included in the fuzzy space of the j-th rule.

[Step 2] Compute the prototypes $\mathbf{V}_j$ of the data set by taking the arithmetic mean of each rule.

$$\mathrm{V}_j = \{V_{1j}, V_{2j}, \ldots, V_{kj}; M_j\} \tag{3}$$

[Step 3] Set the initial values of polynomial functions with the center vectors $\mathbf{V}_j$.

The identification of the conclusion parts of the rules deals with the selection of their structures (Type 1, Type 2, Type 3, and Type 4) which is followed by the determination of the respective parameters of the local functions occurring in it. The consequence part of the rule extended from a typical fuzzy rule in the Takagi–Sugeno–Kang fuzzy model has the following form:

$$\begin{aligned} &R^j : If\ x_1\ is\ A_{1c}\ and \cdots and\ x_k\ is\ A_{kc} \\ &then\ y_j - M_j = f_j(x_1, \cdots, x_k) \end{aligned} \tag{4}$$

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):

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

Type 4 (Modified Quadratic Inference):

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

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_{kj} and M_j are the center values of the input and output data, respectively, and n is the number of fuzzy rules.

The calculation of the numeric output of the model based on the activation (matching) levels of rules relies on the following expressions:

$$\begin{aligned} y^* &= \frac{\sum_{j=1}^{n} w_{ji} y_i}{\sum_{j=1}^{n} w_{ji}} = \frac{\sum_{j=1}^{n} w_{ji}(f_j(x_1, \cdots, x_k) + M_j)}{\sum_{j=1}^{n} w_{ji}} \\ &= \sum_{j=1}^{n} \hat{w}_{ji}(f_j(x_1, \cdots, x_k) + M_j) \end{aligned} \tag{5}$$

where y^* is the inferred output value, and w_{ji} is the premise level of matching R^j (activation level). Given the normalized value of w_{ji}, we use an abbreviated notation to describe an activation level of rule R^j as follows:

$$\hat{w}_{ji} = \frac{w_{ji}}{\sum_{j=1}^{n} w_{ji}}, \quad \hat{w}_{ji} = \frac{A_{j1}(x_{1i}) \times \cdots \times A_{jk}(x_{ki})}{\sum_{j=1}^{n} A_{j1}(x_{1i}) \times \cdots \times A_{jk}(x_{ki})} \tag{6}$$

The consequence parameters a_{jk} can be determined by the standard LSM, which leads to the following expression:

$$\hat{\mathbf{a}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} \tag{7}$$

In the case of Type 2 scheme,

$$\begin{aligned} &\hat{\mathbf{a}} = [a_{10} \cdots a_{n0}\ a_{11} \cdots a_{n1} \cdots a_{1k} \cdots a_{nk}]^T, \\ &\mathbf{X} = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_i \quad \cdots \quad \mathbf{x}_m]^T, \\ &\mathbf{x}_i^T = [\ \hat{w}_{1i} \cdots \hat{w}_{ni} \\ &(x_{1i} - V_{11})\hat{w}_{1i} \cdots (x_{1i} - V_{1n})\hat{w}_{ni} \cdots (x_{ki} - V_{k1})\hat{w}_{1i} \cdots \\ &(x_{ki} - V_{kn})\hat{w}_{ni}\ ], \end{aligned}$$

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

3. Multi-objective Optimization of the IG-Based FIS

Many real-world optimization problems [21] come with multiple objectives which not only interact but may also be in conflict. The main objective of the multi-objective optimization algorithm is to determine a Pareto-optimal set. This optimal set balances the tradeoffs among the conflicting objectives. Multi-objective optimization generates the Pareto front which is the set of non-dominated solutions. A solution is non-dominated if improving one objective of the solution is impossible without worsening at least one of the other objectives presented in the problem. In the current work, we utilize the MSSA as the optimization vehicle of IG-FIS.

3.1 MSSA

First, we discuss a space search algorithm (SSA) with a single object. The SSA is an adaptive heuristic optimization algorithm whose search method comes with the analysis of the solution space [22]. To illustrate the idea of the SSA, let us consider why an evolutionary algorithm (such as the well-known genetic algorithm) can find the optimal solution. In fact, a precondition should be satisfied when the evolutionary algorithm can find the optimal solution. The precondition is that in most local areas, a point (solution) and the other points located in the point's adjacent space have similar values of the objective function (fitness values). In other words, in most local areas, a solution with better fitness is closer to the optimal solution. Moreover, if we take the entire space as the biggest local area into consideration, the precondition can be satisfied for any target optimization problems. Based on this observation, we may give rise to a space search mechanism to update the current solutions. The role of this space search is to generate new solutions from old ones. The search method is based on the operator of the space search, which generates two basic steps: generate new subspace (local area) and search the new space. The latter is realized by generating randomly a new solution (individual) located in this space. Regarding the generation of a new space, we consider two cases: (a) space search based on *M* selected solutions (denoted 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). The core issue is how to determine the adjacent space based on M solutions. For convenience, a solution X can be presented in another way $X=(x_1,x_2,......,x_n)$, where n is the index of the dimension. Regarding M solutions, we use the following representations: $X^k=(x_1^k,x_2^k,......,x_n^k)$, $k=1,2,......,M$. To adjust the size of the new generating subspace, we use the coefficients $a_i\in[l,u]$ as parameters, where *l* and *u* are the given numbers. Suppose V is the new generating space, X^{new} is a new feasible solution generated randomly on the basis of V, where $X^{new}=(x_1^{new},x_2^{new},......,x_n^{new})$, and *S* is the entire feasible solution space. The new space can be determined by the following expression:

$$V=\{X^{new}\left|\; x_i^{new}=\sum_{k=1}^{M}a_i x_i^k \cup X^{new}\in S\,,\ where \sum_{i=1}^{M}a_i=1, l\le a_i\le u\}\right. \tag{8}$$

Fig. 1 depicts the different subspaces generated by *M* solutions using different parameters when the index of the dimension of a feasible solution is equal to 2. In this case, M solutions can be presented as

$$X^k=(x_1^k,x_2^k)\,,$$

where $x_i^k\in[l_i,u_i], i=1,2$, and $k=1,2,......,M$. The four points $x_1^{\min}$, $x_2^{\min}$, $x_1^{\max}$, $x_2^{\max}$ are the minimum of x_1^k, the minimum of x_2^k, the maximum of x_1^k, and the maximum of x_2^k, respectively. The search space *V* is equal to S_1 in case of $a_i\in[0,1]$. S_2 is the search space of *V* in case of $a_i\in[l,u]$, where $l<0$, and $u>1$. In the present study, we search the adjacent space S_2 and set $a_i\in[-1,2]$. We generate the new space V_1 based on the following expression:

$$V_1=\{X^{new}\left|\; x_i^{new}=\sum_{k=1}^{M}a_i x_i^k \cup X^{new}\in S\,,\ where \sum_{i=1}^{M}a_i=1, -1\le a_i\le 2\}\right. \tag{9}$$

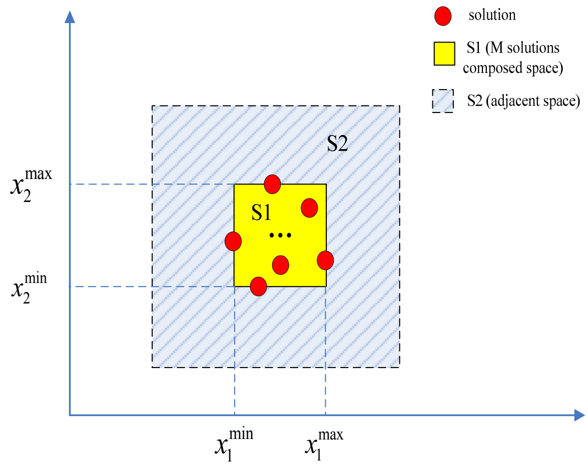


Fig. 1. Different spaces generated from M solutions using different parameters (Case I)

In Case II, the space search operation is based on a given solution which 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. Assume that the best solution in the current solution set is denoted by the following:

$$X^{best}=(x_1^{best},x_2^{best},......,x_n^{best})$$

where $x_i^{best}\in[l_i,u_i], i=1,2,......,n.$ We generate the new space V_2 based on the following expression:

$$V_2=\{X^{new}\mid X^{new}=(x_1^{best},x_2^{best},......,x_{i-1}^{best},x_i^{new},x_{i+1}^{best},......,x_n^{best})\ ,\ where\ x_i^{new}\ne x_i^{best}\cup x_i^{new}\in[l_i,u_i]\} \tag{10}$$

Fig. 2(a) illustrates the new space generated from the current best solution when the index of the dimension of a feasible solution is equal to 2. The current best solution can be presented as follows:

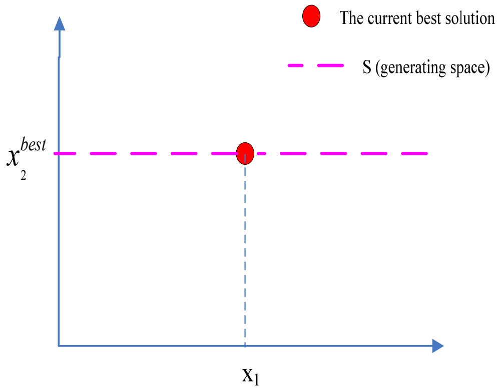
$$X^{best}=(x_1^{best},x_2^{best})\,,$$

where $x_i^{best}\in[l_i,u_i]$, $i=1,2$.

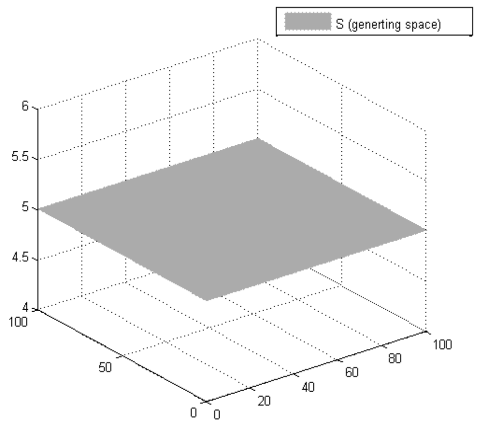
In this case, the value of x_2^{best} is the same as the corresponding value of X^{best}, whereas the value of x_1 ranges from l_1 to u_1. The new space is in essence a line in the solution space. Moreover, we can use the extension space based on V_2 by adding the point x_i^{new}, where $i\in\{1,2,......,n\}$. For example, assuming that the dimension of a feasible solution is equal to 3, the extension space V_3 is expressed as follows:

$$V_3=\{X^{new}\mid X^{new}=(x_1^{best},x_2^{best},...,x_{i-1}^{best},x_i^{new},x_{i+1}^{best},...,x_{j-1}^{best},x_j^{new},x_{j+1}^{best},......,x_n^{best}),\ where\ x_i^{new}\ne x_i^{best}\cup x_j^{new}\ne x_j^{best}\cup x_i^{new}\in[l_i,u_i]\cup x_j^{new}\in[l_i,u_i]\} \tag{11}$$

For convenience, we have to consider that the index of the dimension of a feasible solution is equal to 3. Suppose that the current best solution is denoted by $X^{best} = (x_1^{best}, x_2^{best}, x_3^{best})$, where $x_3^{best} = 5$, $l_i = 0$, $u_i = 100, i = 1, 2, 3$. The new space generated by the extension space such as V_3 is shown in Fig. 2(b).



(a) Use of V_2 in the case of two dimensions



(b) Use of V_3 in the case of three dimensions

Fig. 2. Space generated from a given solution (Case II)

With the understanding of the SSA, we can develop the MSSA. So far, several techniques have been incorporated into multi-objective optimization algorithms in order to improve convergence to the Pareto front ands to produce a well-distributed Pareto front. These techniques include elitism, diversity operators, mutation operators, and constraint handling. The technique of a non-dominant sort and the crowding distance in MSSA are based on the NSGAII [15].

Fig. 3 illustrates the overall flowchart of MSSA. The operator of the search space is realized by the following two basic steps: generate a new subspace (local area) and search the new space. The non-dominated sort is realized using estimation of the crowding distance among solutions in the current solution set *S*. The detailed overall algorithm can be outlined in the following sequence of steps:

Step 1. Initialize (generate randomly) the solution set, $P = (X^1, X^2, ..., X^m)$, where m is the index of the dimension.

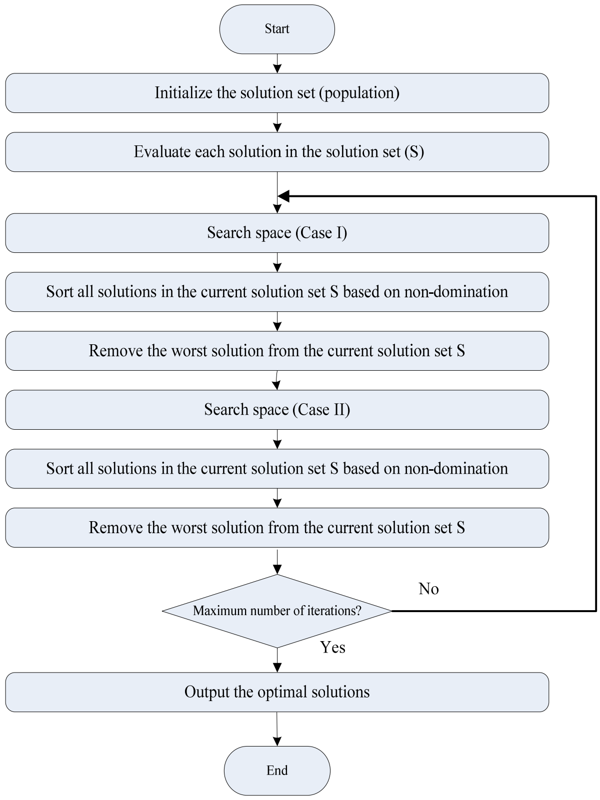


Fig. 3. Overall flowchart of the MSSA

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

Step 3. Search space based on *M* solutions (Case I).

Step 3.1 Select *M* number solutions randomly from the current solution set, where *M* is a given number.

Step 3.2 Generate a new subspace from the *M* solutions according to (2).

Step 3.3 Generate a new solution randomly in the new space.

Step 3.4 Add the new solution in the solution set.

Step 4. Sort the current population based on non-domination.

Step 5. Remove the worst solution from the current solution set.

Step 6. Search space based on the current best solution (Case II).

Step 6.1 Select the current best solution from the current solution set.

Step 6.2 Generate a new subspace based on the current best solution according to (3).

Step 6.3 Generate a new solution randomly in the new space.

Step 6.4 Add the new solution in the solution set.

Step 7. Sort the current population based on non-domination.

Step 8. Remove the worst solution from the current solution set.

Step 9. If the number of iterations equals to N, go to step 3, where N is a given large number (terminal condition).

Step 10. Report the optimal solutions.

3.2 Arrangement of Solutions in the MSSA

When running the optimization method, we distinguish the two main categories of adjustment: sequential [23] and successive tuning [24]. In sequential tuning, structural and parametric optimization are carried out sequentially. First, structural optimization is completed, and then we proceed with the parametric optimization 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, whereas parametric optimization is applied to the fuzzy model derived through structural optimization. In other words, 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 in parameters such as the number of membership functions, number of input, polynomial order, and a collection of specific subsets of input variables. Then parametric optimization is carried out to fine-tune the apexes of the membership functions.

Fig. 4 depicts the arrangement of solutions in the MSSA-based sequential tuning method. The first part of structural optimization is separated from its second part which is used for parametric optimization. The size of the solutions for the 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 the structurally optimized fuzzy inference system. In short, from the viewpoint of structure identification, only one fixed parameter set is considered to carry out the overall structural optimization of the fuzzy model. This set consists of the assigned apexes of the membership functions obtained by C-Means clustering. From the viewpoint of parameter identification, only one structurally optimized model 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 the search space for the structural and parametric optimization is restricted strictly to the sequential tuning method.

To address the problem, we present the MSSA-based successive tuning method. In this method, we achieve both structural and parametric optimization of the model simultaneously. Fig. 5 shows the arrangement of solutions used for the successive tuning method. The second part of parameter identification is linked up with the first part of structure identification within a solution (an individual).

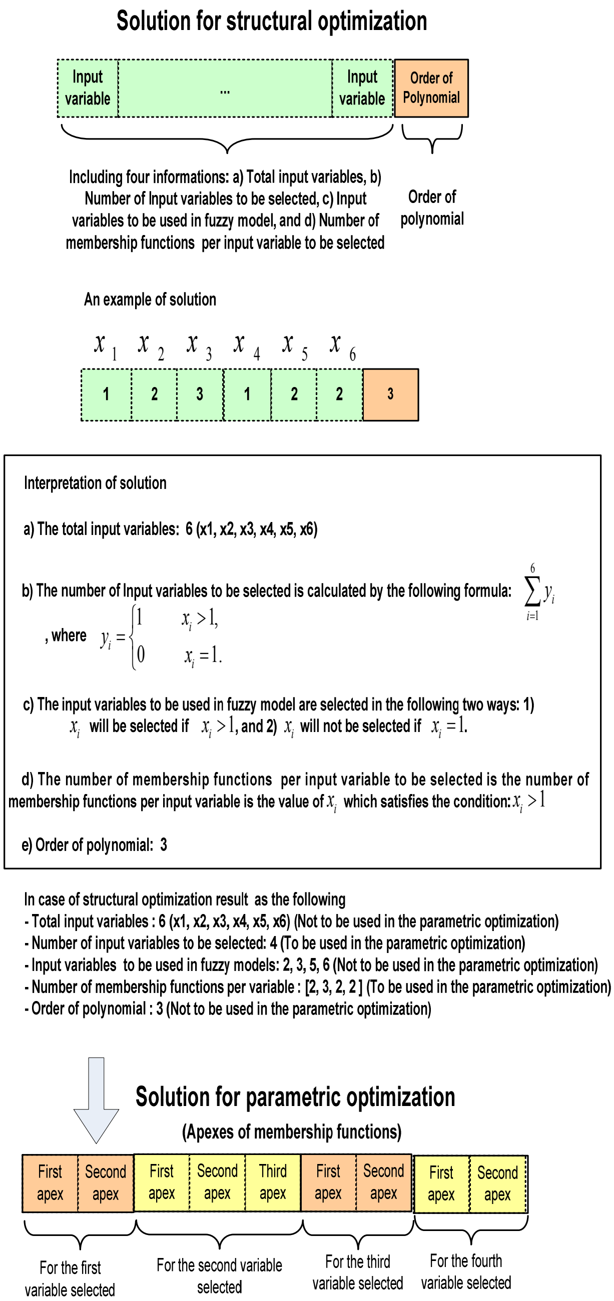


Fig. 4. Arrangement of solutions in the sequential tuning method

The size and arrangement of the first part for structure identification is the same as those in the sequential tuning method, whereas the size of the second part of parameter identification is determined by considering both the number of the system's input variables and the number of membership functions used in their representation. In the successive tuning method, a stochastic variable (a variant identification ratio; see Fig. 6) used within a modified simple search space operator in the MSSA is used to support an efficient successive tuning, including both the structural and parametric optimization of the model. During the initial generations of the SSA, the space search

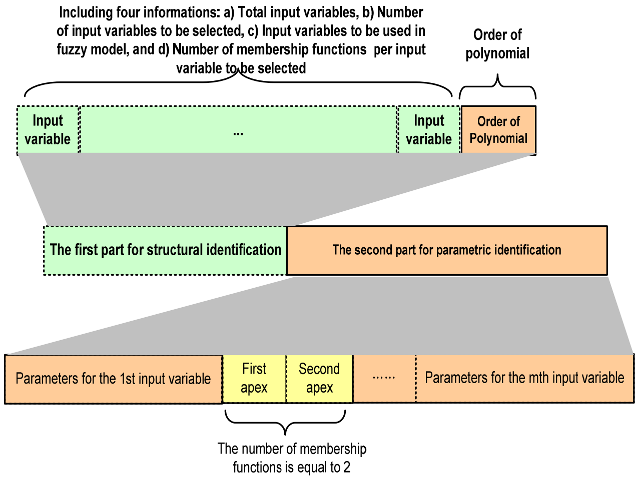


Fig. 5. Arrangement of solutions in the successive tuning method

> *While { the termination conditions are not met }*
> *Select M solutions (parent individuals) from the current solution set, where M is a given number.*
> *Generate random variable (r1).*
> *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 the solution space within the first part of solutions for structural optimization.*
> *Else*
> *Search the solution space within the second part of solutions for parametric optimization.*
> *End IF*
> *Complete the space search operation.*
> *End while*

Fig. 6. Scheme of the space search operator in the MSSA in pseudo-code

operator is assigned with a higher probability to the solution region involving the first part, which is responsible for structural optimization. This probability becomes lower when dealing with a region of the solution involving the second part responsible for parametric optimization. This way, the optimization becomes focused predominantly on 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, which is responsible for parametric optimization, gradually increases. In this sense, the optimization of the IG-based fuzzy set model becomes focused predominantly on parametric optimization.

In the first step of the sequential tuning method, the "topology (structure)-only search with fixed parameters" is carried out for optimization. In the next step, the "parameter-only search with fixed topology (structure)" is conducted for optimization. In the successive tuning method, the second part related to the parametric optimization of the model is connected serially with the first part related to the structural optimization of the model. Therefore, a "simultaneous topology/parameter search" is carried out for optimization. Moreover, the successive tuning method enables us to consider a much more extensive topology/parameter search space for optimization compared with the sequential tuning method. The space search operator in the MSSA for the successive tuning method realized using a variant identification ratio is used. Parameters such as gen, maxgen, and λ should be given. gen is an index of the current generation, whereas maxgen represents the maximum number of generations used in the algorithm; λ serves as some adjustment coefficient whose values can determine a variant identification ratio (p) for both structural and parametric optimization. The scheme of the space search operator in the MSSA algorithm is shown in Fig. 6.

3.3 Objective functions of IG-FIS

Three objective functions are used to evaluate the accuracy and complexity of an IG-FIS: performance indexes, entropy of partition, and the total number of the coefficients of the polynomials to be estimated. Once the input variables of the premise part have been specified, the optimal consequence parameters that minimize the assumed performance index can be determined.

We consider two performance indexes, the standard root mean squared error (RMSE) and the mean squared error (MSE).

$$PI(or\ E\_PI) = \begin{cases} \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - y_i^*)^2}, & (RMSE) \\ \frac{1}{m}\sum_{i=1}^{m}(y_i - y_i^*)^2. & (MSE) \end{cases} \quad (12)$$

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

The accuracy criterion E includes both the training data and the testing data, and comes as a convex combination of the two components:

$$E = \theta \times PI + (1 - \theta) \times E\_PI \quad (13)$$

PI and E_PI denote the performance index for the training data and testing data, respectively. θ is a weighting factor that allows us to strike a sound balance between the performance of the model for the training and testing data. Depending on 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 the testing data are taken into account. Moreover, they are assumed to exhibit the same effect 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 α establishes a certain tradeoff between the approximation and generalization aspects of the fuzzy model.

As a measure for evaluating the structure complexity of a model, we consider the following partition criterion:

$$H = \prod_{i=1}^{n} F_i \tag{14}$$

where *n* is the total number of selected input variables, and F_i is the number of membership functions for the *i*th corresponding input variable.

As a simplicity criterion, we consider the consequence part of the local models, which is computed as follows:

$$N = \sum_{j=1}^{n} C_i ,$$

$$C_i = \begin{cases} 1 & \text{if type of local model is constant} \\ 1+l & \text{if type of local model is linear form} \\ 1+l+(l^2-l)/2+l & \text{if type of local model is quadratic form} \\ 1+l+(l^2-l)/2 & \text{if type of local model is modified quadratic form} \end{cases} \tag{15}$$

where C_i is the number of coefficients of the *i*th polynomial, and l stands for the number of input variables.

In a nutshell, we find the Pareto optimal sets and Pareto front by minimizing {E, H, N} using the MSSA. This leads to interpretable, simple, and accurate fuzzy models.

4. Experimental Studies

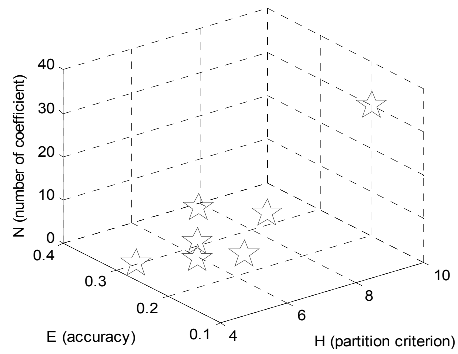
This section reports on comprehensive numeric studies illustrating the design of the fuzzy model. Three well-known data sets are used. Each data set is divided into two parts of the same size. PI denotes the performance index for the training data, and E_PI represents the testing data. In all considerations, the weighting factor θ was set to 0.5.

The parameters of the MSSA are set as follows. We use 100 generations and a size of 200 populations (solutions) 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 current best solution. In the simultaneous tuning method, λ is set as 2.0.

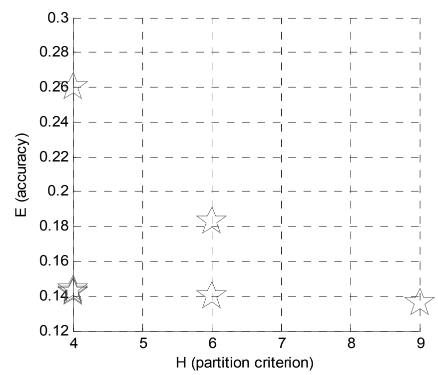
4.1 Gas Furnace Process

The first well-known data set is the time-series data of a gas furnace utilized by Box and Jenkins. The time-series data which consist of 296 input-output pairs resulting from the gas furnace process have been studied extensively in the literature [1-5]. 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 $[u(t-3), u(t-2), u(t-1), y(t-3), y(t-2), y(t-1)]$. Carbon density $y(t)$ is used as output variable. MSE is considered as the performance index.

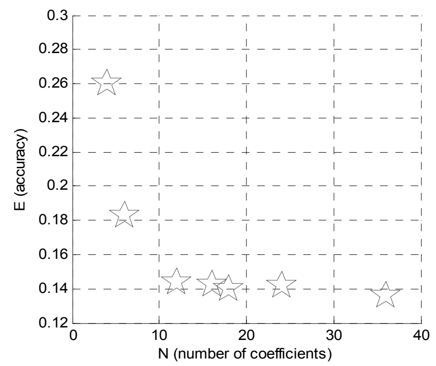
The gas furnace process is partitioned into two parts. The first 148 pairs are used as the training data, whereas the remaining 148 pairs are used as the testing data set for assessing predictive performance. Fig. 7 illustrates the Pareto fronts generated using the MSSA in the case of the



(a) Pareto front (three dimensions)



(b) Pareto front (two dimensions: H and E)



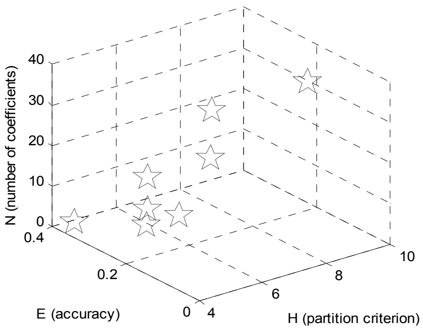
(c) Pareto front (two dimensions: N and E)

Fig. 7. Pareto front produced by the MSSA in the case of the sequential tuning method (Gas)

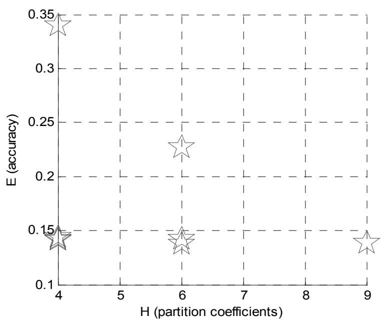
sequential tuning method. Generally, as expected, by increasing the number of coefficients or rules, the accuracy of IG-FIS is increased. Table 1 summarizes the performance values of the solution (individual) objective functions (E, H, N) of the IG-FIS.

Table 1. Optimal solutions using the sequential tuning method (Gas)

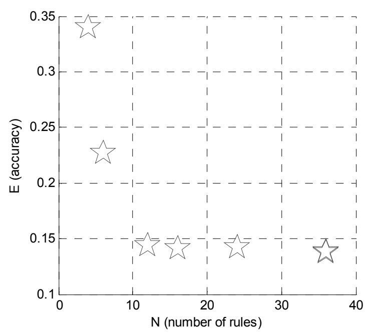
	Selected input variables	No. of MFs	Type	PI	E_PI	Objective values		
						E	H	N
1	1,6	3,3	4	0.015	0.257	0.136	9	36
2	1,6	2,2	3	0.017	0.266	0.142	4	24
3	5,6	3,2	2	0.087	0.193	0.140	6	18
4	5,6	2,2	4	0.087	0.197	0.142	4	16
5	5,6	2,2	2	0.091	0.196	0.144	4	12
6	5,6	2,3	1	0.119	0.247	0.183	6	6
7	1,6	2,2	1	0.097	0.424	0.260	4	4



(a) Pareto front (three dimensions)



(b) Pareto front (two dimensions: H and E)



(c) Pareto front (two dimensions: N and E)

Fig. 8. Pareto front produced by the MSSA in the case of the simultaneous tuning method (Gas)

Fig. 8 depicts the Pareto fronts generated using the MSSA in the case of the simultaneous tuning method. The results of the proposed model are summarized in Table 2.

Table 2. Optimal solutions using the simultaneous tuning method (Gas)

	Selected input variables	No. of MFs	Type	PI	E_PI	Objective values		
						E	H	N
1	5,6	3,3	4	0.074	0.202	0.138	9	36
2	1,6	2,3	3	0.015	0.260	0.138	6	36
3	1,6	2,3	4	0.018	0.266	0.142	6	24
4	1,6	2,2	3	0.017	0.266	0.142	4	24
5	5,6	2,2	4	0.088	0.195	0.141	4	16
6	5,6	2,2	2	0.091	0.196	0.144	4	12
7	1,6	2,3	1	0.063	0.391	0.227	6	6
8	2,6	2,2	1	0.167	0.512	0.339	4	4

Table 3 presents a comparative analysis of some existing models. We consider the performance values of the individual objective functions (E, H, N) of the IG-FIS within the Pareto optimal set (refer to Tables 1 and 2, marked shadow). As can be seen, the proposed model compares favorably both in terms of accuracy and prediction capabilities.

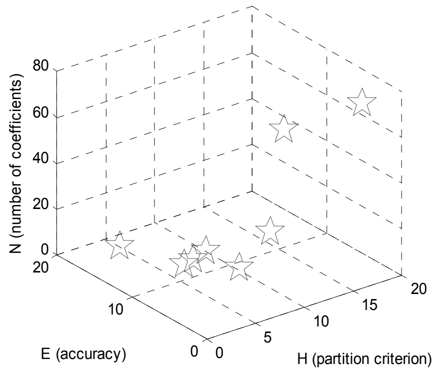
Table 3. Results of selected models (Gas)

Model		Pit	PI	E_PI	No.of rules
Pedrycz's model [1]		0.776			20
Tong's model [2]		0.469			19
Xu's model [3]		0.328			25
Sugeno's model [4]		0.355			6
Oh et al.'s Model [5]	Simplified		0.024	0.328	4
	Linear		0.022	0.326	4
			0.021	0.364	6
HCM+GA [7]	Simplified		0.035	0.289	4
			0.022	0.333	6
	Linear		0.026	0.272	4
			0.020	0.264	6
Our model	Sequential tuning		0.017	0.266	4
	Simultaneous tuning		0.015	0.260	6

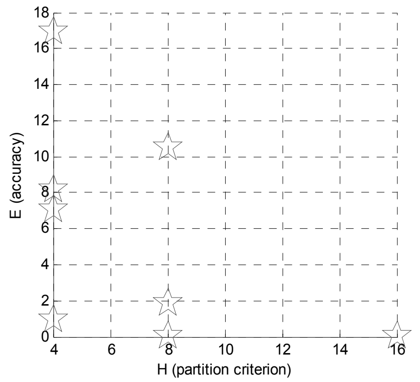
4.2 NOx Emission Process Data

The NOx emission process is also modeled using the data of gas turbine power plants. A NOx emission process of a GE gas turbine power plant located in Virginia, USA is chosen in the 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 NOx, and the performance index is MSE defined by Eq. (12).

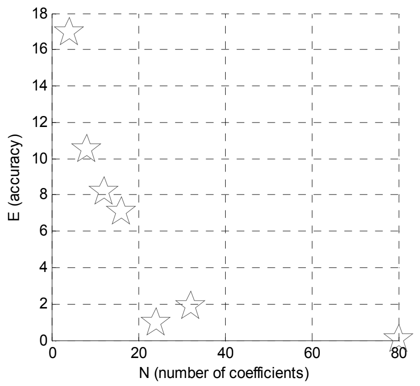
Fig. 10 depicts the Pareto fronts generated using the MSSA in the case of the simultaneous tuning method. The results of the proposed model are shown in Table 5.



(a) Pareto front (three dimensions)



(b) Pareto front (two dimensions: H and E)

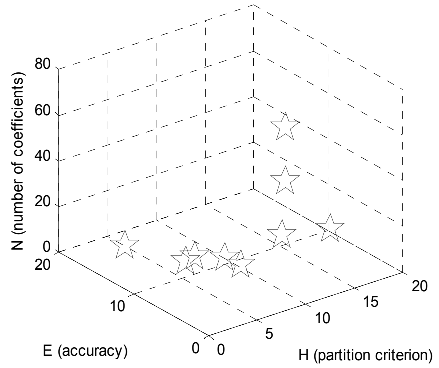


(c) Pareto front (two dimensions: N and E)

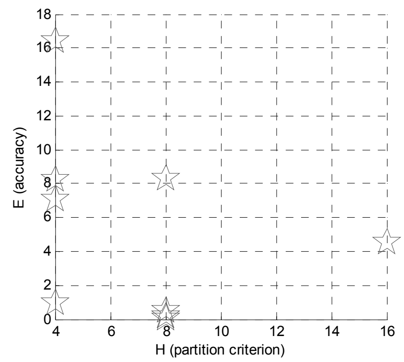
Fig. 9. Pareto front produced by the MSSA in the case of the sequential tuning method (NOx)

Table 4. Optimal solutions using the sequential tuning method (NOx)

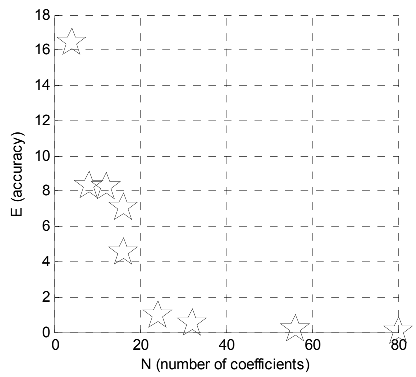
	Selected input variables	No. of MFs	Type	PI	E_PI	Objective values E	H	N
1	2,3,4,5	2,2,2,2	2	0.017	0.084	0.051	16	80
2	1,4,5	2,2,2	3	0.022	0.084	0.053	8	80
3	2,4,5	2,2,2	2	0.907	2.843	0.019	8	32
4	1,4	2,2	3	1.026	0.831	0.928	4	24
5	4,5	2,2	4	5.520	8.534	7.027	4	16
6	4,5	2,2	2	6.703	9.671	8.187	4	12
7	1,4,5	2,2,2	1	9.221	11.78	10.50	8	8
8	4,5	2,2	1	13.60	20.23	16.92	4	4



(a) Pareto front (three dimensions)



(b) Pareto front (two dimensions: H and E)



(c) Pareto front (two dimensions: N and E)

Fig. 10. Pareto front produced by the MSSA in the case of the simultaneous tuning method (NOx)

Table 5. Optimal solutions using the simultaneous tuning method (NOx)

	Selected input variables	No. of MFs	Type	PI	E_PI	Objective values E	H	N
1	1,4,5	2,2,2	3	0.0163	0.057	0.037	8	80
2	1,4,5	2,2,2	4	0.0565	0.220	0.138	8	56
3	1,4,5	2,2,2	2	0.502	0.521	0.511	8	32
4	1,4	2,2	3	1.022	0.829	0.926	4	24
5	1,2,4,5	2,2, 2,2	1	2.445	6.628	4.536	16	16
6	4,5	2,2	4	5.530	8.523	7.026	4	16
7	4,5	2,2	2	6.715	9.661	8.188	4	12
8	2,4,5	2,2,2	1	5.474	11.10	8.288	8	8
9	4,5	2,2	1	13.998	18.83	16.41	4	4

Table 6 illustrates the results of the comparative analysis of the proposed model with the other models. The selected values of the performance indexes of the IG-FIS are marked in Tables 4 and 5. As can be seen, the proposed model outperforms several previous fuzzy models known in the literature.

Table 6. Results of the selected models (NOx)

Model		PI	E_PI	No. of rules
Regression model		17.68	19.23	
Hybrid FS-FNNs [25]		2.806	5.164	30
Hybrid FR-FNNs [26]		0.080	0.190	32
Multi-FNN [27]		0.720	2.205	30
Hybrid rule-based FNNs[28]		3.725	5.291	30
Choi's model [29]		0.012	0.067	18
Our model	Sequential tuning	0.017	0.084	16
	Simultaneous tuning	0.016	0.057	8

4.3 Chaotic Mackey–Glass Time Series

A chaotic time series is generated by the chaotic Mackey–Glass differential delay equation as

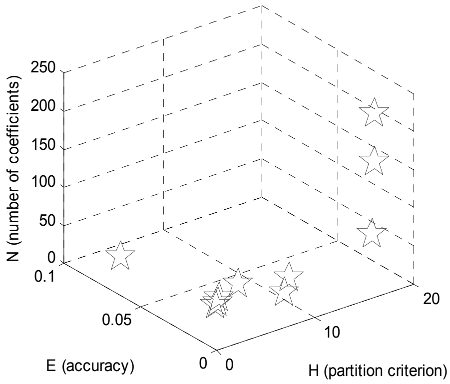
$$\dot{x}(t) = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t)$$

The prediction of the future values of this series introduces 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 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. To come up with a quantitative evaluation of the fuzzy model, we use the standard RMSE performance index as described by (12).

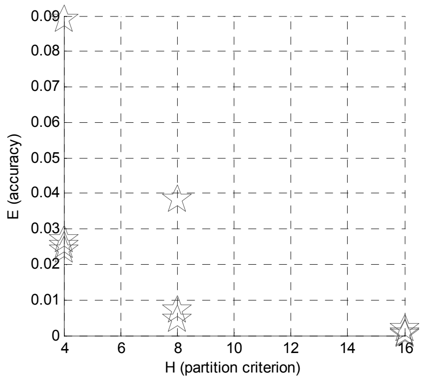
We consider the Mackey–Glass data set, which is divided into two separate parts. The first 500 data pairs were used as the training data set for IG-based FIS, whereas the remaining 500 pairs were used as the testing data set for assessing the predictive performance.

Figs. 11 and 12 depict the Pareto fronts generated using the MSSA in the case of the sequential and simultaneous tuning methods, respectively. The values of the performance index in the case of the sequential tuning method are presented in Table 7, whereas those in the case of the simultaneous tuning method are shown in Table 8.

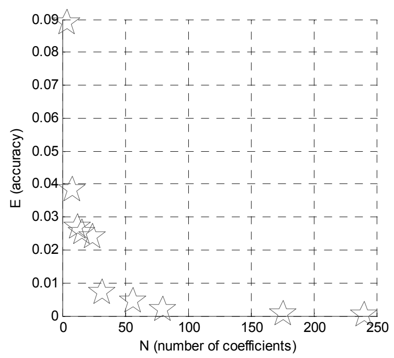
The identification error of the proposed model is compared with those of some other models, as shown in Table 9. From this table, the performance of the proposed model is evidently better than that of the other models in terms of approximation and prediction capabilities.



(a) Pareto front (three dimensions)



(b) Pareto front (two dimensions: H and E)

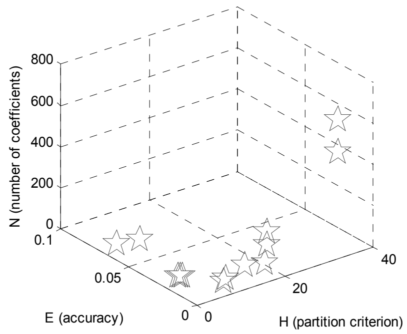


(c) Pareto front (two dimensions: N and E)

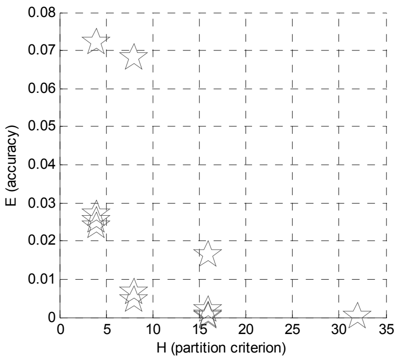
Fig. 11. Pareto front produced by the MSSA in the case of the sequential tuning method (Mackey)

Table 7. Optimal solutions using the sequential tuning method (Mackey)

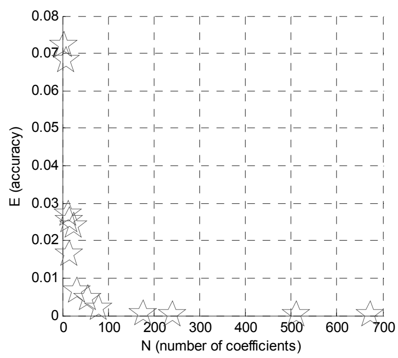
	Selected input variables	No.of MFs	Type	PI	E_PI	Objective values E	H	N
1	1,3,4,6	2,2,2,2	3	0.000107	0.000156	0.000131	16	240
2	1,3,4,6	2,2,2,2	4	0.000325	0.000333	0.000329	16	176
3	2,3,4,5	2,2,2,2	2	0.00185	0.00181	0.00183	16	80
4	3,4,5	2,2,2	4	0.00448	0.00448	0.00448	8	56
5	3,4,5	2,2,2	2	0.00713	0.00686	0.00700	8	32
6	4,6	2,2	3	0.0240	0.0238	0.0239	4	24
7	4,6	2,2	4	0.0253	0.0253	0.0253	4	16
8	4,6	2,2	2	0.0267	0.0269	0.0268	4	12
9	3,4,5	2,2,2	1	0.0385	0.0377	0.0381	8	8
10	2,4	2,2	1	0.0894	0.0883	0.0888	4	4



(a) Pareto front (three dimensions)



(b) Pareto front (two dimensions: H and E)



(c) Pareto front (two dimensions: N and E)

Fig. 12. Pareto front produced by the MSSA in the case of the simultaneous tuning method (Mackey)

Table 8. Optimal solutions using the simultaneous tuning method (Mackey)

	Selected input variables	No.of MFs	Type	PI	E_PI	Objective values E	H	N
1	1,2,3,4,6	2,2,2,2,2	3	0.0000184	0.000129	0.0000738	32	672
2	1,3,4,5,6	2,2,2,2,2	4	0.0000143	0.000109	0.0000618	32	512
3	1,3,4,6	2,2,2,2	3	0.000110	0.000142	0.000126	16	240
4	1,3,4,6	2,2,2,2	4	0.000324	0.000324	0.000324	16	176
5	2,3,4,5	2,2,2,2	2	0.00186	0.00176	0.00181	16	80
6	3,4,5	2,2,2	4	0.00445	0.00431	0.00438	8	56
7	1,4,6	2,2,2	2	0.00656	0.00643	0.00650	8	32
8	3,4,5,6	2,2,2,2	1	0.0166	0.0160	0.0163	16	16
9	4,6	2,2	3	0.0241	0.0237	0.0239	4	24
10	4,6	2,2	4	0.0253	0.0253	0.0253	4	15
11	4,6	2,2	2	0.0269	0.0269	0.0269	4	12
12	1,3,5	2,2,2	1	0.0687	0.0674	0.0680	8	8
13	3,5	2,2	1	0.0726	0.0714	0.0720	4	4

Table 9. Results of the selected models (Mackey)

Model		PI_t	PI	E_PI	NDEI	No.of rules
Support vector regression model [30]			0.023	1.028	0.0246	
Multivariate adaptive regression splines [30]			0.019	0.316	0.0389	
Standard neural networks			0.018	0.411	0.0705	15
RBF neural networks			0.015	0.313	0.0172	15
Wang's model [31]		0.004				7
		0.013				23
ANFIS [32]			0.0016	0.0015	0.007	16
FNN model [33]			0.014	0.009		
Incremental type multilevel FRS [34]			0.0240	0.0253		25
Aggregated type multilevel FRS [34]			0.0267	0.0256		36
Hierarchical TS-FS[35]			0.0120	0.0129		28
Our model	Sequential tuning		0.00011	0.00016	0.0013	16
	Simultaneous tuning		0.00011	0.00014	0.0007	16

5. Concluding Remarks

The current work contributes to research on the hybrid optimization of fuzzy inference systems in the following aspects. First, we proposed a multi-objective space search algorithm. The MSSA using a technique of non-dominated sort and crowding distance is designed on the basis of a space search algorithm. This is an optimization algorithm whose search method comes with the analysis of the solution space. Second, we introduced the fuzzy identification of fuzzy inference systems based on the MSSA and LSM. Instead of single objective optimization of the fuzzy inference system, the MSSA is used to conduct parametric optimization of the fuzzy model and to realize its structural optimization in the design of the IG-based fuzzy model. Numerical experiments using three well-known data sets show that the model constructed using the MSSA has better performance compared with the fuzzy model reported in the literature.

Acknowledgements

This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government (NRF-2010-D00065), Korean Ministry of Education, Science, and Technology Grant (The Regional Core Research Program/Center of Healthcare Technology Development), and the GRRC Program of Gyeonggi Province [GRRC SUWON 2011-B2, Center for U-city Security & Surveillance Technology].

References

[1] W. Pedrycz, "An identification algorithm in fuzzy

relational system". *Fuzzy Sets Syst.*, vol. 13, pp, 153-167, 1984.

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

[3] C. W. Xu., Y. Zailu, "Fuzzy model identification self-learning for dynamic system" *IEEE Trans. Syst., Man cybern.*, vol. 17, no 4, pp, 683-689, 1987.

[4] M. Sugeno, T. Yasukawa, "Linguistic modeling based on numerical data." *in Proceedings of IFSA'91 Brussels, Computer, Management & System Science,* pp, 264-267. 1991.

[5] S. K. Oh., W. Pedrycz, "Identification of Fuzzy Systems by means of an Auto-Tuning Algorithm and Its Application to Nonlinear Systems," *Fuzzy Sets and Syst.*, vol. 115, no 2, pp, 205-230, 2000.

[6] W.Y Chung, W. Pedrycz, E.T Kim, "A new two-phase approach to fuzzy modeling for nonlinear function approximation," *IEICE Trans. Info. Syst.*, vol. 9, pp, 2473-2483, 2006.

[7] 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.

[8] W.Y. Chung, E.T. Kim, "A new two-phase approach to fuzzy modeling for nonlinear function approximation," *IEICE Trans. Inform. Syst.*, vol. E89-D, no. 9, pp. 2473–2483, 2006.

[9] F.J. Lin, L.T. Teng, J.W. Lin, S.Y. Chen, "Recurrent Functional-Link-Based Fuzzy-Neural-Network-Controlled Induction-Generator System Using Improved Particle Swarm Optimization," *IEEE Trans. Indust. Elect.*, vol. 56, no. 5, pp. 1557-1577, 2009.

[10] W. Pedrycz, K.C Kwak, "Linguistic models as a framework of user-centric system modeling," *IEEE Trans. Syst., man cybern. –PART A : Systems and humans*, vol. 36, no. 4, pp. 727–745, 2006.

[11] A. Bastian, "Identifying fuzzy models utilizing genetic programming," *Fuzzy Sets and Syst.*, vol. 112, pp. 333-350, 2000.

[12] Y. Jin, "Fuzzy modeling of high-dimensional systems: complexity reduction and interpretability improvement," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 2, pp. 212–221, 2000.

[13] M. Setnes, H. Roubos, "GA-based modeling and classification: complexity and performance," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 5, pp. 509–522, 2000.

[14] C. Coello, G. Pulido, "Multiobjective optimization using a micro-genetic algorithm," *in Proceedings of Genetic and Evolutionary Computation Conference (GECCO 2001)*, pp. 274-282, 2001.

[15] K. Deb, S. Agrawal, A. Pratab, S. Agarwal, T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.* , vol. 6, no.2, pp. 182-197, 2002.

[16] G. Avigad, A. Moshaiov, "Interactive Evolutionary Multiobjective Search and Optimization of Set-Based Concepts," *IEEE Trans. Syst., Man cybern.-Part B*, vol. 38, nol. 2, pp. 381-403, 2008.

[17] C. Coello, G. Pulido, M. Salazar, "Handling multiobjectives with particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, pp. 256-279, 2004.

[18] G.G. Yen, W.F. Leong, "Dynamic Multiple Swarms in Multiobjective Particle Swarm Optimization," *IEEE Trans. Syst., Man Cybern.-PART A*, vol. 39, nol. 4, pp. 890-911, 2009.

[19] L.J. Herrera, H. Pomares, I. Rojas, O. Valenzuela, and A. Prieto, "TaSe, a Taylor series-based fuzzy system model that combines interpretability and accuracy," *Fuzzy Sets and Syst.*, vol. 153, pp. 403-427, 2005.

[20] M. Delgado, M.P. Ceullar, and M.C. Pegalajar, "Multiobjective Hybrid Optimization and Training of Recurrent Neural Networks,*" IEEE Trans. Syst., Man cybern. –Part B*, vol. 38, nol. 2, pp. 381-403, 2008.

[21] W. Huang, L. Ding, "Project-Scheduling problem with random time-dependent activity duration times," *IEEE Transactions on Engineering Management*, vol. 58, no. 2, pp. 377-387, May 2011.

[22] W. Huang, L. Ding, S.K. Oh, C.W. Jeong, S.C. Joo, "Identification of fuzzy inference system based on information granulation," *KSII Transactions on Internet and Information Systems*, vol. 4, no. 4, pp. 575-594, August 2010.

[23] 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

[24] K.J. Park, W. Pedrycz, S.K. Oh, "A genetic approach to modeling fuzzy systems based on information granulation and successive generation-based evolution method", *Simulation Modelling Practice and Theory*, vol. 15, pp, 1128-1145, 2007.

[25] S.K. Oh, W. Pedrycz, H.S. Prak, "Hybrid identification in fuzzy-neural networks", *Fuzzy Set System*, vol. 138, Issue 2, pp, 399-426, 2003.

[26] H.S. Park, S.K. Oh, "Fuzzy relation-based fuzzy neural-networks using a hybrid identification algorithm", *Int. J. Cont., Autom., Syst.*, vol. 1, Issue 2, pp. 289-300, 2003.

[27] H.S. Park, S.K. Oh, "Multi-FNN identification based on HCM clustering and evolutionary fuzzy granulation", *Int. J. Cont., Autom., Syst.*, vol. 1, Issue 2, pp. 194-202, 2003.

[28] 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, Issue 4, pp. 247-263, 2002.

[29] 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.
[30] In: P.R. Krishnaiah., L.N. Kanal (Eds.), "Classification, Pattern Recognition, and Reduction of Dimensionality", *Handbook of Statistics*, vol. 2, North-Holland, Amsterdam, 1982.
[31] L. X. Wang., J. M. Mendel, "Generating fuzzy rules from numerical data with applications", *IEEE Trans. Syst., man cybern.,* vol. 22, pp. 1414-1427, 1992.
[32] J.S.R Jang, "ANFIS: adaptive-network-based fuzzy inference system", *IEEE Trans. Syst., man cybern.*, vol. 23, no. 3, pp. 665-685, 1993.
[33] L.P. Maguire, B. Roche, T.M. McGinnity, L.J. McDaid, "Predicting a chaotic time series using a fuzzy neural, network", *Inform. Sci.*, vol. 112, pp. 125-136, 1998.
[34] J.C. Duan., F.-L. Chung, "Multilevel fuzzy relational systems: structure and identification", *Soft Comput.*, vol. 6, pp. 71-86, 2002.
[35] Y. Chen., B. Yang., A. Abraham, "Automatic design of hierarchical Takagi-Sugeno type fuzzy systems using evolutionary algorithms", *IEEE Trans. Fuzzy Systems*, vol. 15, no. 3, pp. 385-397, 2007.

Wei Huang He received his MS degree from the School of Information Engineering, East China Institute of Technology, China in 2006, and his Ph.D. degree from the State Key Laboratory of Software Engineering, Wuhan University, China in 2011. He is currently a lecturer in the School of Computer and Communication Engineering, Tianjin University of Technology, Tianjin, China. His research interests include evolutionary computation, operations research, fuzzy system, fuzzy-neural networks, natural language processing, and software reliability.

Sung-Kwun Oh He received his B.Sc., M.Sc., and Ph.D. degrees in Electrical Engineering from Yonsei University, Seoul, Korea in 1981, 1983, and 1993, respectively. From 1983 to 1989, he was a Senior Researcher of the R&D Laboratory of Lucky-Goldstar Industrial Systems Co., Ltd. From 1996 to 1997, he was a Postdoctoral Fellow in the Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, MB, Canada. He is currently a Professor in 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 also currently serves as the 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 ICASE*, South Korea.

Lixin Ding He received his B.Sc. and MSc. degrees from the Department of Applied Mathematics, Hunan University, China in 1989 and 1992, and his Ph.D. degree from the State Key Laboratory of Software Engineering, Wuhan University, China in 1998. From 1998 to 2000, he was a Postdoctoral Fellow in the Department of Armament Science and Technology, 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.

Hyun-Ki Kim He received his B.S., M.S., and Ph.D. degrees in Electrical Engineering from Yonsei University, Seoul, Korea, in 1977, 1985, and 1991, respectively. From 1999 to 2003, he worked as the Chairman of the Korea Association of Small Business Innovation Research. He is currently a Professor in the Department of Electrical Engineering, Suwon University, Korea. His research interests include system automation and intelligent control. He also currently serves as the Chief Editor of the Journal of Korea Association of Small Business Innovation Research.

Su-Chong Joo He received his BS degree in Computer Engineering from Won-Kwang University, South Korea in 1986, and his M.S. and Ph.D. degrees from the Department of Computer Science and Engineering, Chung-Ang University, South Korea in 1988 and 1992, respectively. Currently, he is a professor at the Division of Electrical, Electronic, and Information Engineering in Won-Kwang University, South Korea. His main research interests include distributed real-time computing, distributed object modeling, system optimization, multimedia database systems, and healthcare applications. From July 1993 to August 1994, he was a Post-Doctoral Fellow at the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst. Also, from December 2002 to January 2005, he was a research professor at the Department of Electrical Engineering and Computer Science, University of California, Irvine. He is a member of KISS, IASTED, IEEE, and IEEE computer society.