A Dynamically Constrained Multiobjective Genetic Fuzzy System for Regression Problems

Pietari Pulkkinen and Hannu Koivisto

***Abstract*—In this paper, a multiobjective genetic fuzzy system (GFS) to learn the granularities of fuzzy partitions, tuning the membership functions (MFs), and learning the fuzzy rules is presented. It uses dynamic constraints, which enable three-parameter MF tuning to improve the accuracy while guaranteeing the transparency of fuzzy partitions. The fuzzy models (FMs) are initialized by a method that combines the benefits of Wang–Mendel (WM) and decision-tree algorithms. Thus, the initial FMs have less rules, rule conditions, and input variables than if WM initialization were to be used. Moreover, the fuzzy partitions of initial FMs are always transparent. Our approach is tested against recent multiobjective and monoobjective GFSs on six benchmark problems. It is concluded that the accuracy and interpretability of our FMs are always comparable or better than those in the comparative studies. Furthermore, on some benchmark problems, our approach clearly outperforms some comparative approaches. Suitability of our approach for higher dimensional problems is shown by studying three benchmark problems that have up to 21 input variables.**

***Index Terms*—Genetic fuzzy systems (GFSs), initialization, accuracy, interpretability, Mamdani fuzzy models (FMs).**

I. Introduction

INTERPRETABILITY-accuracy tradeoff of fuzzy models (FMs) has recently attained a lot of research interest [1]–[9]. Since it is not possible to maximize these contradicting objectives simultaneously, multiobjective evolutionary algorithms (MOEAs) have recently been used to find a Pareto optimal set of FMs that present different tradeoffs between the objectives. These approaches are also called multiobjective genetic fuzzy systems (GFS) [10], [11].

Accuracy is often measured by mean-squared error (MSE) when regression problems are considered. However, there is no exact measure for interpretability of FMs [2] and it tends to be somewhat subjective. Nevertheless, the definition by Ishibuchi and Yamamoto [12] is often used. It defines interpretability by four factors: 1) transparency of fuzzy partitions; 2) complexity of FMs (e.g., the number of fuzzy rules and input variables); 3) complexity of fuzzy-rule base (e.g., type of rules and the number of rule conditions); and 4) complexity of fuzzy reasoning (e.g., defuzzification method).

Factor 1) is often satisfied by using fixed fuzzy partitions (uniformly distributed or known by *a priori* knowledge) [3], [12]. However, *a priori* knowledge is often not available. Furthermore, if fuzzy partitions do not present the real distribution of data, the accuracy of FMs is deteriorated [13]. Thus, it is important to not only optimize the rules and rule conditions, but also the membership-function (MF) parameters. However, this increases the search space and may deteriorate the transparency of fuzzy partitions.

There are also studies in which fuzzy partitions are not fixed and factor 1) is taken into account by other means. Merging of highly similar fuzzy sets was used in [14] and [15] to improve the transparency of fuzzy partitions. Parameters of a fuzzy set that cover another fuzzy set were automatically adjusted in [4]. Penalties were issued in [5], if the intersection point of two fuzzy sets was not between user-specified boundaries. This approach not only avoided highly overlapping fuzzy sets, but also ensured that the whole universe of discourse (UoD) was strongly covered. The approach [5] was extended in [16] to reduce the effects of relaxed covering [4]. Here, [16] is followed; however, instead of minimizing the penalties, dynamic constraints are used to ensure that the fuzzy partitions are always transparent. This increases the selection pressure and improves the search efficiency [17].

This paper deals with regression (or function estimation) problems, which have not yet received as much research efforts as classification problems [6]. We apply Mamdani FMs [18], which are also called linguistic FMs. When regression problems are considered, the population is usually initialized randomly or by Wang and Mendel (WM) method [19]. Unfortunately, random initialization does not guarantee a good starting point for further optimization, and WM method usually leads to high number of rules and rule conditions when high-dimensional problems and/or problem with many data points are considered. Recently, we proposed a decision-tree (DT) based initialization method for regression problems [20], which reduces the number of input variables and leads to less rules and rule conditions than WM initialization. However, it does not necessarily create transparent fuzzy partitions. WM algorithm, on the other hand, creates rules for *a priori* given fuzzy partitions; thus, transparency of fuzzy partitions is usually high. Here, we combine the benefits of WM and DT initialization. Therefore, the initial fuzzy partitions are transparent, and the initial FMs contain less rules, rule conditions, and input variables than when WM algorithm is used.

The initial population is then optimized by multiobjective GFS that uses dynamic constraints to ensure the transparency of fuzzy partitions. It also reduces the number of rules, rule conditions, MFs, and input variables. The proposed initialization method and multiobjective GFS therefore aid to satisfy the aforementioned factors [1)–3)]. Factor 4), which is the complexity of fuzzy reasoning, is taken into account by applying simple-weighted-average-defuzzication method.

Manuscript received March 9, 2009; revised June 15, 2009 and September 18, 2009; accepted November 24, 2009. First published December 15, 2009; current version published February 5, 2010.

The authors are with the Department of Automation Science and Engineering, Tampere University of Technology, Tampere 33101, Finland (e-mail: pietari.pulkkinen@tut.fi; hannu.koivisto@tut.fi).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TFUZZ.2009.2038712

TABLE I
SECOND-GENERATION MULTIOBJECTIVE GFSS APPLIED TO IDENTIFICATION OF LINGUISTIC FMS

Problem type	Year	Reference	MFs tuning	Rule selection	Rule learning	Input variable selection: Initialization	Learning	Transparent fuzzy partition
Classification	2007	[1]	No	No	**Yes**	No	**Yes**	Not always
	2008	[7]	**Yes**	No	**Yes**	**Yes**	**Yes**	Not always
	2008	[16]	**Yes**	No	**Yes**	**Yes**	**Yes**	**Yes**
Regression	2007	[3]	No	No	**Yes**	No	**Yes**	**Yes**
	2008-9	[21], [22]	**Yes**	No	**Yes**	No	**Yes**	**Yes**
	2007-9	[8], [23]	**Yes**	**Yes**	No	No	No	Not always
	2009	[24]	No	No	**Yes**	No	**Yes**	**Yes**
	2009	[25]	**Yes**	No	**Yes**	No	**Yes**	**Yes**
	2009	[26]	**Yes**	No	**Yes**	No	**Yes**	**Yes**
	2009	[27]	**Yes**	No	No	No	No	**Yes**
	2009	[20]	**Yes**	No	**Yes**	**Yes**	**Yes**	Not always
	Proposed approach		**Yes**	No	**Yes**	**Yes**	**Yes**	**Yes**

Our multiobjective GFS is tested on a set of nine benchmark problems having 2 up to 21 input variables. For six of them, there are results of other recently proposed GFSs available. Our results are compared to them, and it is shown that our results are comparable or better in terms of accuracy and interpretability.

This paper is organized as follows. First, a brief survey of recently proposed multiobjective GFSs is given. Based on this, novelty of our multiobjective GFS is clearly pointed out. Then, the interpretability of FMs is discussed and a special attention is paid to transparency of fuzzy partitions. Then, in Section IV, the proposed initialization method is introduced. After this, in Section V, dynamically constrained multiobjective GFS is presented. The results comparisons are performed in Section VI and conclusions are given in Section VII.

II. MULTIOBJECTIVE GENETIC FUZZY SYSTEMS FOR LINGUISTIC-FUZZY-MODEL IDENTIFICATION: STATE OF THE ART

Recently, several researchers have focused on designing multiobjective GFSs to identify of compact and accurate linguistic FMs. Ishibuchi's research group has published several papers that consider fuzzy classification. Nonetheless, until recently, there were hardly any papers that considered multiobjective GFSs in regression problems [23].

Table I presents multiobjective GFSs for classification and regression problems. For the sake of brevity, it includes only the recent approaches that apply the second-generation MOEAs (e.g., the nondominated sorting genetic algorithm II (NSGA-II), the strength pareto evolutionary algorithm 2 (SPEA2), and the pareto archived evolution strategy (PAES)). It also excludes those approaches that apply first-order Takagi–Sugeno FMs. In this table, rule selection means that a rule is either included or not included into an FM, whereas rule learning means that appropriate rule conditions are learned by GFS. It is seen that usually either rule learning or rule selection is applied, and there is only one approach [27] that applies neither of them.

MFs of fuzzy rules are taken from four different fuzzy partitions in [1], which means that the resulting global fuzzy partitions are not always transparent. Granularities of global fuzzy partitions are learnt in [24], which improves the transparency. The most trivial way to obtain transparent fuzzy partitions is to use evenly distributed uniformly shaped MFs, like in [3]. However, MFs tuning is often applied because it usually improves the accuracy. Unfortunately, it often deteriorates the transparency of fuzzy partitions. In the area of regression problems, there are some methods [21], [22], [25]–[27] that apply MFs tuning and have appropriately considered this factor. One of them [27] is a context-adaptation approach that only performs MFs tuning, requiring the whole rule base to be provided by the user. MF parameters are learnt using a linguistic two-tuple tuning scheme [9] in [21] and [22]. Piecewise-linear-transformation techniques are applied in [25] and a wrapper-based embedded process is used in [26]. The approaches [8], [20], [23] apply conventional three-parameter MFs tuning with static constraints, which does not guarantee transparency of fuzzy partitions.

In this paper, three-parameter MFs tuning with dynamic constraints is applied. The search space is therefore larger compared to two-tuple representation, which only modifies the lateral displacements of the MFs. On the other hand, it is excepted that the proposed approach improves the accuracy. Moreover, because of dynamic constraints, it is guaranteed that the whole UoD is strongly covered and there is no highly overlapping MFs. Our approach also does not require that MFs are uniformly shaped as long as the transparency conditions, which are introduced later in Section III-A, are met. In some cases, uniformly shaped MFs can actually be misleading if they do not present the real distribution of the data. In some cases, it is therefore necessary that some fuzzy sets are, for example, wide, whereas some others are narrow. Finally, granularities of global fuzzy partitions are also learnt by our approach. These properties guarantee that our approach maintains the transparency of fuzzy partitions at a good level.

Input-variable selection before applying GFS (i.e., in initialization phase) reduces the number of parameters to be optimized. This has been applied by some approaches; however, in the field of regression, there is only one approach [20] that applies this. Usually, regression problems with 2 up to 10 input variables are studied in the literature, and therefore, the role of input-variable selection is not crucial. However, in this paper, its role becomes more important as problems up to 21 input variables are studied.

The difference between the proposed approach and the approach [16] is more than just a different problem type. Transparency of fuzzy partitions was obtained in [16] by minimizing a transparency index. It means that the transparency indexes of FMs in population may be very different. There may be some

FMs with highly transparent fuzzy partitions and some other FMs with unacceptable fuzzy partitions. Naturally, by constraining the range in which the value of transparency index can vary reduces the variation. However, in this case, the offspring population will usually contain some infeasible FMs (FMs for which the transparency index is not acceptable). This deteriorates the search efficiency of GFS. In this paper, transparency of fuzzy partitions is guaranteed by dynamic constraints. This reduces the number of fitness objectives by one, which increases the selection pressure [17].

Based on this brief analysis, it can be concluded that the proposed multiobjective GFS is novel. Indeed, to the best of our knowledge, there exist no multiobjective GFS applicable to regression, which performs rule learning and three-parameter MFs tuning, while preserving transparency of fuzzy partitions. Moreover, input variables are selected in two ways. First, during the initialization phase. Second, during the multiobjective-GFS-search process, which can select input variables among the remaining ones after the initialization.

III. Interpretability of Fuzzy Models

As mentioned previously, in this paper, the factors 2) and 3) of the interpretability definition [12] are satisfied by minimizing the complexity of FMs and factor 4) by application of simple-weighted-average defuzzification. However, because the MFs are tuned, factor 1)—transparency of fuzzy partitions—requires a special attention. In the next section, a definition for this is given. It applies only to input variables, because in this paper, singleton output MFs are used. Because singleton MFs can be presented with only one parameter, it is sufficient to apply static constraints, introduced later in Section V-B, to maintain the transparency of output partition at a good level.

A. Transparency of Fuzzy Partitions

As in [27], this paper uses the transparency definition by de Oliveira [28], which states that a transparent fuzzy partition must meet the conditions, which are given as follows.

1) The number of MFs per variable is moderate.
2) MFs are distinguishable, i.e., two MFs do not present the same or almost the same linguistic meaning.
3) Each MF is normal. An MF is normal if it has membership value 1 at least at one point of UoD.
4) UoD is strongly covered. At least one MF receives a membership value β (where $\beta > 0$) at any point of UoD.

Condition 1) is easily met by constraining the maximum number of MFs to a moderate number (for example, 9). Also, condition 3) is met by applying normal MFs and genetic operators that do not alter their normality. Meeting conditions 2) and 4) is more challenging. In this paper, it is considered that they are met if globally defined MFs are used and the following conditions are met.

1) *Symmetry condition:* The shapes of all MFs are symmetrical. For example, Gaussian MF and generalized-bell (gbell) MF are symmetrical by definition. Also, other MF types, such as triangular and trapezoidal MFs can be easily made symmetrical.

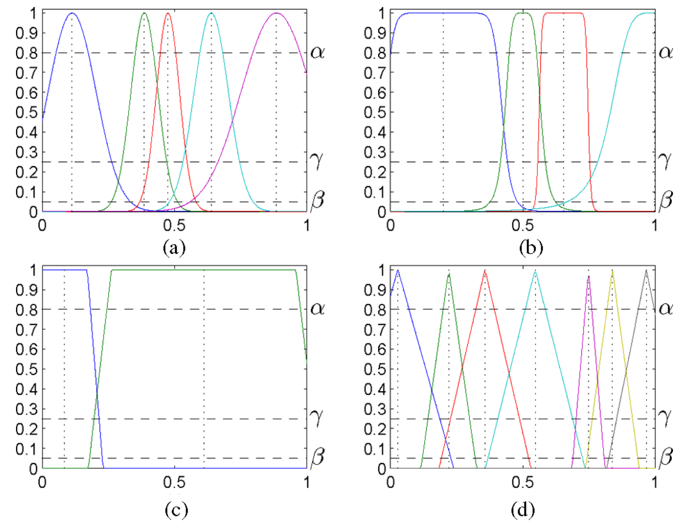


Fig. 1. Examples of fuzzy partitions that are considered to be transparent. MF centers are marked with dotted vertical lines. (a) Gaussian MFs. (b) gbell MFs. (c) Symmetrical trapezoidal MFs. (d) Symmetrical triangular MFs.

2) *α-condition:* At any intersection point of two MFs, the membership value is at most α.
3) *γ-condition:* At the center of each MF, no other MF receives membership value larger than γ. Center of an MF depends on which MF type is used. For gbell MF (with parameters a, b, and c) and Gaussian MF (with parameters c and σ), center is the parameter c. For triangular MF (with parameters $a < b < c$), b is the center. For trapezoidal MF (with parameters $a < b < c < d$), center is $b + ((c - b)/2)$ (see also Fig. 1).
4) *β-condition:* UoD is strongly covered, i.e., at each point of UoD, at least one MF has membership value at least β.

Fig. 1 shows examples of fuzzy partitions with settings $\beta = 0.05$, $\gamma = 0.25$, and $\alpha = 0.8$. Section III-B describes how β, γ, and α must generally be selected in order to apply the dynamic-tuning strategy.

In this paper, gbell MFs are used. They are defined as

$$\mu(x; a, b, c) = \frac{1}{1 + |((x - c)/a)|^{2b}} \tag{1}$$

where a, b, and c define the width, shape, and center of an MF, respectively. As gbell MFs are symmetrical, first of the previous conditions is met. Fulfillment of the rest three conditions rely largely on computing the values of x, for which an MF receives a certain membership value μ. Because of the symmetry of gbell MFs, any membership value $\mu \in (0, 1)$ is received on the left and right side of the center c. These points are denoted here by I_{L} and I_{R}

$$I_{\mathrm{L}}(\mu, \mathbf{p}) = c - a\,(\kappa(\mu))^{1/2b}, \qquad \mu \in (0, 1) \tag{2}$$

$$I_{\mathrm{R}}(\mu, \mathbf{p}) = c + a\,(\kappa(\mu))^{1/2b}, \qquad \mu \in (0, 1) \tag{3}$$

where $\mathbf{p} = [a, b, c]^T$ is a vector containing the MF parameters and

$$\kappa(\mu) = \frac{1 - \mu}{\mu}, \qquad \mu \in (0, 1). \tag{4}$$

Equations (2) and (3) are used to formulate the α, γ, and β conditions. For the sake of clarity, each of them is split into two parts, denoted here by left and right. They ensure the fulfillment of the conditions on the left or right side of the center of an MF, respectively. Let the active MFs of a variable be indexed as $j = 1, \ldots, M_A$, where M_A is the number of currently active MFs of that variable. It will be shown later that our multiobjective GFS maintains the ordering of MFs, i.e., if $i > j$, then $c_i > c_j$, where c_i and c_j are the gbell parameters c of MFs i and j. Moreover, in this paper, the fuzzy partitions with only one MF are not allowed, because they are not considered transparent. Hence, throughout this paper, it is known that if $j = 1$, then MF is the leftmost MF and its neighboring MF is $j + 1$. If $1 < j < M_A$, then MF is in the middle of neighboring MFs $j - 1$ and $j + 1$. Finally, if $j = M_A$, then MF is the rightmost MF and the neighboring MF is $j - 1$. Thus, the transparency conditions can be written as follows

Right α-condition

$$I_R(\alpha, \mathbf{p}_j)) \leq I_L(\alpha, \mathbf{p}_{j+1}), \qquad \text{if } j < M_A.$$

Left α-condition

$$I_L(\alpha, \mathbf{p}_j)) \geq I_R(\alpha, \mathbf{p}_{j-1}), \qquad \text{if } j > 1.$$

Right γ-condition

$$I_R(\gamma, \mathbf{p}_j) \leq c_{j+1} \wedge c_j \leq I_L(\gamma, \mathbf{p}_{j+1}), \qquad \text{if } j < M_A.$$

Left γ-condition

$$I_L(\gamma, \mathbf{p}_j) \geq c_{j-1} \wedge c_j \geq I_R(\gamma, \mathbf{p}_{j-1}), \quad \text{if } j > 1.$$

$$\text{Right } \beta\text{-condition: } \begin{cases} I_R(\beta, \mathbf{p}_j) \geq I_L(\beta, \mathbf{p}_{j+1}), & \text{if } j < M_A \\ I_R(\beta, \mathbf{p}_j) \geq \chi_{\text{high}}, & \text{if } j = M_A \end{cases}$$

$$\text{Left } \beta\text{-condition: } \begin{cases} I_L(\beta, \mathbf{p}_j)) \leq I_R(\beta, \mathbf{p}_{j-1}), & \text{if } j > 1 \\ I_L(\beta, \mathbf{p}_j) \leq \chi_{\text{low}}, & \text{if } j = 1 \end{cases}$$

where the variable range is $\chi = \chi_{\text{high}} - \chi_{\text{low}}$, where χ_{low} and χ_{high} are the lower and upper bounds of the variable, respectively. These conditions are the basis of the proposed dynamic constraints, which require that the fuzzy partitions of initial FMs are transparent. Thus, two simple partition algorithms to create transparent fuzzy partitions are introduced next.

B. Partitioning Algorithm to Create Evenly Distributed Fuzzy Partition

This algorithm creates a fuzzy partition consisting of M_A evenly distributed uniformly shaped MFs, and it is only used when creating the first FM of the initial population. Because MFs are uniformly shaped, the gbell parameter a for each MF j is

$$a_j = a_{\text{even}} = \frac{\chi}{2(M_A - 1)}, \qquad j = 1, \ldots, M_A. \tag{5}$$

It is required that each $a_j \geq a_{\min} = 0.025\chi$ to avoid very narrow MFs. This limits the maximum value of M_A to 21; however, in practice, more than nine MFs are hardly ever assigned. The minimum value of M_A is 2. Centers are distributed evenly as

$$c_1 = \chi_{\text{low}}, \quad \text{and} \quad c_j = c_{j-1} + \frac{\chi}{M_A - 1}, \quad j = 2, \ldots, M_A. \tag{6}$$

Assigning the values for a and c according to (5) and (6) guarantees that UoD is strongly covered and the membership value of each MF pair at their intersection point is 0.5. Thus, $0 < \beta < 0.5$ and $0.5 < \alpha < 1$ must be selected in order to apply the dynamic-tuning strategy. Because the membership value at each intersection point is 0.5, the β and α conditions are fulfilled. Moreover, because gbell MFs are symmetrical, the symmetry condition is satisfied as well. The γ-condition requires that at the center of each MF, no other MF receives membership value larger than γ. This algorithm selects b, such that, at the center of each MF, the neighboring MF(s) receive the membership value $\gamma^* = 0.05$. Thus, $\gamma^* < \gamma < 0.5$ must be selected in order to apply the dynamic-tuning strategy. The following formula for selecting b can be derived by starting from either (2) or (3):

$$b_j = \frac{\ln \kappa(\gamma^*)}{2\ln(d_{\text{center},j}/a_j)}, \qquad j = 1, \ldots, M_A \tag{7}$$

where

$$d_{\text{center},j} = \begin{cases} \min(c_j - c_{j-1}, c_{j+1} - c_j), & \text{if } 1 < j < M_A \\ c_{j+1} - c_j, & \text{if } j = 1 \\ c_j - c_{j-1}, & \text{if } j = M_A \end{cases} \tag{8}$$

denotes the minimum distance from c_j to the nearest center(s) of neighboring MF(s).

Because MFs are evenly distributed, $d_{\text{center},j} = \chi/(M_A - 1)\ \forall j$. Thus, (7) can be written as

$$b_j = \frac{\ln \kappa(\gamma^*)}{\ln 4}, \qquad j = 1, \ldots, M_A. \tag{9}$$

There is no upper limit for the value of b in the sense that larger b values will not violate the transparency conditions. However, very large b values are not desired as they make gbell MFs similar to crisp sets and because b is the exponent in (1). Therefore, value of b for each MF is defined by (9) by this algorithm.

C. Partitioning Algorithm to Create Unevenly Distributed Fuzzy Partition

As there is no *a priori* knowledge about the distribution of MFs, it is also beneficial to create unevenly distributed nonuniformly shaped MFs. The following algorithm is used for this purpose, and it is applied to create the fuzzy partitions of the rest FMs of the initial population and as a part of genetic operators. It selects c and a as follows:

$$a_1 = \max(a_{\min}, r_1 a_{\text{even}}), \qquad \text{and} \qquad c_1 = \chi_{\text{low}} \tag{10}$$

$$a_j = \max\left(a_{\min}, r_j\left(\frac{(2j-1)a_{\text{even}} - (c_{j-1} + a_{j-1})}{2}\right)\right) \tag{11}$$

where $j = 2, \ldots, M_A - 1$

$$c_j = c_{j-1} + a_{j-1} + a_j, \qquad j = 2, \ldots, M_A - 1 \tag{12}$$

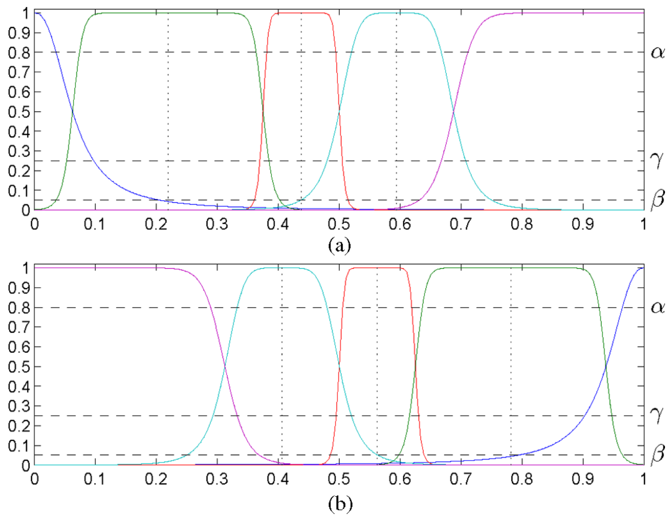


Fig. 2. Example of (a) unevenly distributed fuzzy partition and (b) its inverse.

$$a_{M_{\mathrm{A}}} = \chi_{\mathrm{high}} - (c_{M_{\mathrm{A}}-1} + a_{M_{\mathrm{A}}-1}), \quad \text{and} \quad c_{M_{\mathrm{A}}} = \chi_{\mathrm{high}} \tag{13}$$

where $r_1, r_2, \ldots, r_{M_{\mathrm{A}}-1} \in [0, 1]$ are random real numbers; a_{even} and $a_{\min}$ were defined in the previous section. It can be easily verified that by selecting $r_1 = r_2 = \cdots = r_{M_{\mathrm{A}}-1} = 1$, this algorithm is identical to the algorithm in the previous section.

Unlike in the previous partition algorithm, here, parameter b values are randomly selected from interval [1, 10]. However, they are not allowed to be less than the corresponding minimum values computed according to (7). Thus, it is guaranteed that at the center of each MF, the neighboring MF(s) receive the membership value less than or equal to γ^*.

It is seen from (10), (11), and (13) that the more narrow MFs are more likely to be located on the left side of the range and the wider MFs on the right side of the range. There is, naturally, no justification for this. Hence, by uniform chance, the parameters are defined either by (10)–(13) or by their inversion as follows:

$$a_j^* = a_{M_{\mathrm{A}}-j+1}, \qquad b_j^* = b_{M_{\mathrm{A}}-j+1}, \qquad c_j^* = \chi_{\mathrm{high}} - c_{M_{\mathrm{A}}-j+1} \tag{14}$$

where $j = 1, \ldots, M_{\mathrm{A}}$.

As an example, consider creating a fuzzy partition with five MFs in range [0, 1]. From (5), it follows that $a_{\mathrm{even}} = 1/8$. Let $r_1 = 1/2, r_2 = 1, r_3 = 1/2$, and $r_4 = 1/2$. Thus, $a_1 = a_5^* = 1/16$, $c_1 = 0$, $a_2 = a_4^* = 5/32$, $c_2 = 7/32, a_3 = a_3^* = 1/16, c_3 = 7/16, a_4 = a_2^* = 3/32, c_4 = 19/32, a_5 = a_1^* = 5/16$, and $c_5 = 1$ and $c_1^* = 0, c_2^* = 13/32, c_3^* = 9/16, c_4^* = 25/32$, and $c_5^* = 1$. The minimum values for $b_1 = b_5^*, b_2 = b_4^*, b_3 = b_3^*, b_4 = b_2^*$, and $b_5 = b_1^*$, according to (7), are 1.1752, 4.3755, 1.6067, 2.8820, and 5.6114, respectively. Fig. 2(a) shows the resulting partition when centers are computed according to (10)–(13), whereas Fig. 2(b) depicts the resulting partition when (14) is used instead. It is seen that although MFs are nonuniformly shaped and unevenly distributed, the fuzzy partitions are transparent and reasonable linguistic values could be given. In Fig. 2, $\beta = 0.05$, $\gamma = 0.25$, and $\alpha = 0.8$.

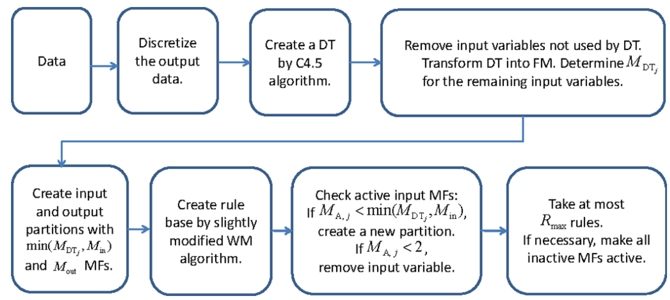


Fig. 3. Procedure of creating the first FM of the population.

IV. Population Initialization

Whenever a GFS is used, the population needs to be initialized first. In order to reduce the search space, it is desirable that the initialization method is able to select the relevant input variables. Thus, in [15], the C4.5 [29] DT-based method for classification problems was proposed. Recently, in [20], it was made suitable for regression problems. Although this method is capable of selecting relevant input variables, its main limitations are that: 1) it does not guarantee transparent fuzzy partitions and 2) it may create far more rules than necessary when applied to noisy datasets.

In this paper, DT initialization is neither used to create the rule base nor to initialize MF parameters, but to select relevant input variables, to reduce the number of input MFs, and rule conditions. MF parameters are determined by the introduced partition algorithms (see Section III-B and Section III-C), which guarantee transparency of fuzzy partitions. Rule base is created by slightly modified WM algorithm [19]. The proposed two modifications are that: 1) when a data point is matched to MFs in order to generate a rule, the data point is not always matched to MFs of all possible input variables. Instead, it is first classified by the constructed DT, and only those input variables that were used by DT to classify the data point are used for matching and 2) as WM algorithm may create large number of rules for datasets with many data points and/or input variables, the generated rules are divided among the members of initial population and only a portion of them is allowed to be included into one FM.

A. Creation of the First Fuzzy Method of the Population

The procedure of creating the first FM is shown in Fig. 3. It is started by discretizing the continuous output data in order to apply C4.5 algorithm. This is done by dividing the output to M_{out} crisp regions. Each continuous output value falls into one of these M_{out} regions and it is replaced with corresponding class label $S \in \{1, \ldots, M_{\mathrm{out}}\}$, which represents these regions. Then, C4.5 algorithm can be applied and a DT constructed.

All input variables which are not used by DT are then removed. Then, fuzzy partitions for the remaining input variables and for the output are created. A user is required to provide the number of input MFs M_{in} and the number of output MFs M_{out}. However, the DT can be used to limit the number of input MFs. First, the DT is transformed into an FM, according to [15]. After this, the number of MFs for each input variable j in the resulting FM is checked and denoted by M_{DT_j}. Then, instead of partitioning each input variable with M_{in} MFs, each input partition is created with $\min(M_{\mathrm{DT}_j}, M_{\mathrm{in}})$ MFs. The output is partitioned with M_{out} MFs. These partitions consist of uniformly shaped evenly distributed MFs and are created by the algorithm introduced in Section III-B.

Then, a slightly modified WM algorithm is used. As mentioned previously, when a rule is generated, each data point is first classified by the constructed DT and only those input variables that were used by DT to classify the data point are used for matching and become conditions of the generated rule. All other parts of the classical WM algorithm remain unchanged.

After creating the rule base, the number of active MFs $M_{\mathrm{A},j}$ for each input variable j is checked (an MF is active if it is part of at least one of the rules). If $M_{\mathrm{A},j} < \min(M_{\mathrm{DT}_j}, M_{\mathrm{in}})$, then there is a gap in fuzzy partition and the whole UoD is not strongly covered. If this is the case and if $M_{\mathrm{A},j} \geq 2$, then a new evenly distributed partition with $M_{\mathrm{A},j}$ MFs is created. If $M_{\mathrm{A},j} < 2$, then input variable j is removed and $M_{\mathrm{A},j}$ is set to 0. The maximum number of MFs, i.e., $M_{\max_j} = M_{\mathrm{A},j}$, that each FM of the population can use in input variable j is determined by this phase. Also, all the input variables that are not removed until now form the set of candidate input variables. The number of these remaining input variables is denoted by n_{s}.

The generated rule base may contain large amount of rules. However, in this paper, each FM can contain at most $R_{\max} = 30$ rules. If the rule base has more than $R_{\max}$ rules, then $R_{\max}$ rules are randomly selected out of it. Otherwise, the rule base is taken as a whole. If rules are randomly selected, it may result into some gaps in the fuzzy partition, which is not allowed. In this case, it is required that the number of active MFs for each input variable must be $M_{\max_j}$ and the number of active output MFs must be M_{out}. If this is not the case, then $\max(M_{\max_1}, M_{\max_2}, \ldots, M_{\max_{n_{\mathrm{s}}}}, M_{\mathrm{out}})$ randomly selected rules are replaced with some rules, thus making all the inactive MFs active. In this paper, these rules are created, such that, in the first of the rules, all antecedents and the consequent are 1. In the second rule, they are all 2. This is continued until all inactive MFs have become active. Of course, it must be taken care that the antecedents for input variable j are at most $M_{\max_j}$ and, for the consequent, at most M_{out}. This rule replacement is necessary only if the rule base contains more than $R_{\max}$ rules. Otherwise, it is certain that there are no gaps in the fuzzy partition.

B. Mamdani Fuzzy Model and Its Coding for Multiobjective-Genetic-Fuzzy-System Optimization

The original dataset contains n input variables; however, the initialization method selects $n_{\mathrm{s}} \leq n$ of them. Therefore, a dataset with D data points is denoted as $Z = [X \quad \mathbf{y}]$, where X is $D \times n_{\mathrm{s}}$ input matrix, and $\mathbf{y}$ is $D \times 1$ output vector. The first FM and all other FMs in this paper are Mamdani FMs. Mamdani fuzzy rules are expressed as

$$R_i : \text{If } x_1 \text{ is } B_{i,1} \ \ldots \text{ and } x_{n_{\mathrm{s}}} \text{ is } B_{i,n_{\mathrm{s}}}, \text{ then } C_i$$

where $B_{i,j}$, with $j = 1, \ldots, n_{\mathrm{s}}$ and $i = 1, \ldots, R$, is an input fuzzy set, C_i is an output fuzzy set, and R is the number of rules. To reduce the computational costs, the output of FMs is computed by approximation of centroid of gravity method [3], [30] as

$$\hat{y}_k = \frac{\sum_{i=1}^{R} \beta_i(\mathbf{x}_k)\bar{C}_i}{\sum_{i=1}^{R} \beta_i(\mathbf{x}_k)}, \qquad k = 1, \ldots, D \tag{15}$$

where $\bar{C}_i$ is the center value of C_i, and $\beta_i(\mathbf{x}_k) = \prod_{j=1}^{n_{\mathrm{s}}} B_{i,j}(x_{k,j})$ is the degree of rule activation. When the slightly modified WM algorithm was used to create the rule base, gbell output MFs were used. However, at the optimization phase, application of gbell MFs is not necessary anymore, since $\bar{C}$ is the only output MF parameter affecting the outcome. Therefore, all gbell output MFs are replaced with singleton MFs as

$$\mu(x, \bar{C}) = \begin{cases} 1, & \text{if } x = \bar{C} \\ 0, & \text{if } x \neq \bar{C} \end{cases}$$

where $\bar{C}$ is the corresponding gbell MF parameter c. For the purpose of multiobjective GFS optimization, the antecedents of the rule base are presented with an integer-coded matrix A. It specifies for each rule $i = 1, \ldots, R$ that MF is used for input variable $j = 1, \ldots, n_{\mathrm{s}}$

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n_{\mathrm{s}}} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n_{\mathrm{s}}} \\ \vdots & \vdots & \ddots & \vdots \\ a_{R,1} & a_{R,2} & \cdots & a_{R,n_{\mathrm{s}}} \end{bmatrix} \tag{16}$$

$a_{i,j} \in \{0, 1, \ldots, M_{\max_j}\}$, where $M_{\max_j}$ is the maximum-number MFs in input variable j. If $a_{i,j} = 0$, input variable j is not used in rule i. Input variable j is not used in an FM if $\forall i, a_{i,j} = 0$, and rule i is not used in an FM if $\forall j, a_{i,j} = 0$. Input MF parameters to which each $a_{i,j}$ is referring are defined in a real-coded matrix P as

$$P = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,\delta} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,\delta} \\ \vdots & \vdots & \ddots & \vdots \\ p_{\rho,1} & p_{\rho,2} & \cdots & p_{\rho,\delta} \end{bmatrix} \tag{17}$$

where ρ is the number of parameters used to define an MF. In this paper, $\rho = 3$, because gbell MFs are used. The maximum number of MFs in an FM is denoted by $\delta = \sum_{j=1}^{n_{\mathrm{s}}} M_{\max_j}$. Thus, for any $a_{i,j} \neq 0$, the corresponding input MF parameters are $p_{1,l}, p_{2,l}$, and $p_{3,l}$, where

$$l = \begin{cases} a_{i,j}, & \text{if } j = 1 \\ a_{i,j} + \sum_{k=1}^{j-1} M_k, & \text{if } j > 1. \end{cases} \tag{18}$$

Similarly as A states the input MFs used in the rules, an integer-coded vector $\mathbf{s}$ defines the output MFs (singletons) used in the rules. Formally, $\mathbf{s} = [s_1, s_2, \ldots, s_R]^T$, where $s_i \in \{1, \ldots, M_{\mathrm{out}}\}$, with $i = 1, \ldots, R$. The maximum number of output MFs is denoted by M_{out}. The output MF parameters to which each s_i is referring are defined in a real-coded vector $\mathbf{o} = [o_1, o_2, \ldots, o_{M_{\mathrm{out}}}]^T$. The total number of parameters to be optimized by a multiobjective GFS is $\theta = Rn_{\mathrm{s}} + \rho\delta + R + M_{\mathrm{out}}$, i.e., the sum of the cardinalities of A, P, $\mathbf{s}$, and $\mathbf{o}$.

C. Mamdani-Fuzzy-Model Coding: An Example

Let us assume that the first FM of the initial population has four rules and uses two input variables x_1 and x_2, which are partitioned, respectively, with three and two gbell MFs. The output is partitioned with four singleton MFs. Both input variables and the output are in the range of [0, 1]. The partitions are uniformly shaped and evenly distributed. The rule base is given as follows.

Rule$_1$*:* If x_1 is 1 and x_2 is 1, then output is 1.
Rule$_2$*:* If x_1 is 2 and x_2 is 2, then output is 2.
Rule$_3$*:* If x_1 is 2 and x_2 is 1, then output is 3.
Rule$_4$*:* If x_1 is 3 and x_2 is 2, then output is 4.

This FM is coded as

$$A = \begin{bmatrix} 1 & 1 \\ 2 & 2 \\ 2 & 1 \\ 3 & 2 \end{bmatrix}, \qquad \mathbf{s} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}, \qquad \mathbf{o} = \begin{bmatrix} 0 \\ 1/3 \\ 2/3 \\ 1 \end{bmatrix}$$

$$P = \begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.5 & 0.5 \\ 2.124 & 2.124 & 2.124 & 2.124 & 2.124 \\ 0 & 0.5 & 1 & 0 & 1 \end{bmatrix}$$

where the first, second, and third row of P contain the gbell parameters a, b, and c, respectively. The first three columns of P contain the gbell parameters of the three MFs of x_1 and the rest two columns contain the gbell parameters of the two MFs of x_2. These parameters are computed according to the algorithm in Section III-B.

D. Creation of the Rest of the Population

The first FM defines the maximum number of rules, maximum number of input variables, and maximum number of MFs per input variable for all the rest $N_{\mathrm{pop}} - 1$ FMs of the population, where N_{pop} is the population size.

If the rule base generated by slightly modified WM algorithm has more than $R_{\max}$ rules, it means that some of the randomly selected rules in the first FM were replaced in order to avoid gaps in the fuzzy partition. In this case, one of the $N_{\mathrm{pop}} - 1$ FMs receives the rule base (i.e., A and $\mathbf{s}$) of the first FM without any replacements. Then, A and $\mathbf{s}$ of the rest $N_{\mathrm{pop}} - 2$ FMs are created by randomly selecting $R_{\max}$ rules from the generated rule base.

If the generated rule base has at most $R_{\max}$ rules, then the rule conditions A of $N_{\mathrm{pop}} - 1$ FMs are created by modifying the rule conditions of the first FM by replacing them with random conditions [7]. However, do-not-care conditions (i.e., conditions that are 0) are not allowed here, as it was pointed out in [8] that it is easier to obtain compact than accurate FMs. Rule consequents $\mathbf{s}$ for all $N_{\mathrm{pop}} - 1$ FMs are the same as in the first FM.

After creating A and $\mathbf{s}$ of the rest $N_{\mathrm{pop}} - 1$ FMs, the input MF parameters P are assigned based on A of each individual FM. For each input variable j of each FM, the number of active MFs $M_{\mathrm{A},j}$ is first checked. If $M_{\mathrm{A},j} \geq 2$, then a new unevenly distributed fuzzy partition with $M_{\mathrm{A},j}$ MFs is created by using the algorithm in Section III-C. If $M_{\mathrm{A},j} < 2$, then all nonzero rule conditions, if any, of that input variable are forced to zero. After this, this input variable has no active MFs, and the value of MF parameters for this input variable can be assigned to any value. However, if the genetic operators at a later stage cause at least two MFs to be active, then the value of these parameters is determined by the algorithm in Section III-C. Finally, the output MF parameters $\mathbf{o}$ for all $N_{\mathrm{pop}} - 1$ FMs are the same as in the first FM.

E. Creation of the Rest of the Population: An Example

Let us return to the example from Section IV-C and consider creating one of the rest $N_{\mathrm{pop}} - 1$ FMs. Since the initial FM has only $4 \leq R_{\max}$ rules, the rules are created by modifying the rules of the first FM. Assume that as a result, the condition *If* x_1 *is 1* of the first rule was changed to *If* x_1 *is 3*. Now, the FM has no rule in which the condition *If* x_1 *is 1* is part of. Thus, the input MF 1 of x_1 is inactive and a new unevenly distributed partition is created with two MFs and assigned to input MFs 2 and 3 of x_1, such that their order is maintained. Similarly, a new unevenly distributed partition with two MFs is also created for x_2, which still has two active MFs. The following could be the result after these operations:

$$A = \begin{bmatrix} \mathbf{3} & 1 \\ 2 & 2 \\ 2 & 1 \\ 3 & 2 \end{bmatrix}, \qquad \mathbf{s} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}, \qquad \mathbf{o} = \begin{bmatrix} 0 \\ 1/3 \\ 2/3 \\ 1 \end{bmatrix}$$

$$P = \begin{bmatrix} \mathit{0.25} & \mathbf{0.3} & \mathbf{0.7} & \mathbf{0.8} & \mathbf{0.2} \\ \mathit{2.124} & \mathbf{3} & \mathbf{9} & \mathbf{7} & \mathbf{4} \\ \mathit{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} \end{bmatrix}$$

where the operated parameters are indicated with boldface. The parameter values of input MF 1 in x_1 are indicated with italics, because they are currently not important. If at some point of optimization, MF 1 becomes active again, the values will be assigned by the algorithm in Section III-C. Before this, none of the genetic operators will operate on these parameters.

V. Dynamically Constrainted Multiobjective Genetic Fuzzy System

After the initialization, the further optimization is performed by popular NSGA-II [31]. Other parts of the algorithm are left unchanged; however, the original genetic operators are replaced with operators applying dynamic constraints, thus ensuring transparency of fuzzy partitions.

A. Fitness Objectives

Two objectives are to be minimized, which are as follows.

1) $\mathrm{MSE} = (1/2D)\sum_{k=1}^{D}(y_k - \hat{y}_k)^2$, where y_k and $\hat{y}_k$ are the actual and predicted outputs for data point k, and D is the number of data points. This objective is actually $\mathrm{MSE}/2$, but it is denoted here as MSE, which is quite common in the field of GFSs.

2) Number of active-rule conditions (total rule length): $R_{\rm cond}$.

The MSE objective is constrained, such that, each FM need to have $\text{MSE} \leq 1.5 \times \text{MSE}_{\rm initial}$, where $\text{MSE}_{\rm initial}$ is the MSE of the first FM of the initial population created in Section IV-A. This constraint is fairly easy to meet as it will be seen in Section VI that the accuracy can be significantly improved by multiobjective GFS optimization. However, it guarantees that the population does not contain some FMs only because they are very compact. Their accuracy must be reasonable as well.

B. Static Constraints for Output Membership Functions

As singleton output MFs are used, there is only one parameter to be optimized (lateral displacement). Therefore, they can be constrained by allowing them to slightly move left/right from their initial positions. The applied static constraints for output MF parameters are

$$\chi_{\rm low} - \frac{\chi}{M_{\rm out}-1} \leq o_1 \leq \chi_{\rm low} + \frac{\chi}{2(M_{\rm out}-1)}$$

$$\chi_{\rm low} + \frac{(2k-3)\chi}{2(M_{\rm out}-1)} \leq o_k \leq \chi_{\rm low} + \frac{(2k-1)\chi}{2(M_{\rm out}-1)}$$

where $k = 2, \ldots, M_{\rm out} - 1$

$$\chi_{\rm h} - \frac{\chi}{2(M_{\rm out}-1)} \leq o_{M_{\rm out}} \leq \chi_{\rm high} + \frac{\chi}{M_{\rm out}-1}.$$

This way of tuning resembles lateral-tuning method [9], and it guarantees transparency of output fuzzy partition to a good level.

C. Dynamic Constraints to Ensure the Transparency of Input Fuzzy Partition

This section presents the dynamic constraints guaranteeing transparent input fuzzy partition in case that a parameter of an MF is modified. The genetic operators assuring transparent input fuzzy partition in case that the number of MFs is altered are introduced later in Section V-D. A prerequisite for these dynamic constraints is that initially (i.e., before modification) the input fuzzy partition is transparent. This is guaranteed by the two partition algorithms, which have already been introduced in Section III-B and Section III-C.

MF parameters are modified one at a time. After each modification, the resulting fuzzy partition must satisfy the transparency conditions defined in Section III-A. As the initial fuzzy partitions are created by the algorithms in Section III-B and III-C, the ordering of MFs is initially known. The ordering is also known after each modification, because the dynamic constraints and the genetic operators in Section V-D do not allow to change it. Therefore, for any two MFs with parameters a_i, b_i, and c_i, and a_j, b_j, and c_j, where $j, i, \in [1, M_{\rm A}]$, with $i \neq j$, of an input variable that currently has $M_{\rm A}$ active MFs, it is guaranteed that if $i > j$, then $c_i > c_j$ and *vice versa*. This is beneficial to design the dynamic constraints.

Besides the dynamic constraints, some static constraints also need to be satisfied; $a_j \in [a_{\rm low}, a_{\rm high}]$, where $a_{\rm low} = 0.005\chi$, and $a_{\rm high} = \chi$. Furthermore, $b_j \geq b_{\rm low} = 1$, and it is preferred that $b_j \leq b_{\rm high} = 10$; however, due to partition algorithms, it may be that initially $b_j > b_{\rm high}$. In this case, b_j is not allowed to increase anymore. Finally, $c_j \in [c_{\rm low}, c_{\rm high}]$, where $c_{\rm low} = \chi_{\rm low}$, and $c_{\rm high} = \chi_{\rm high}$. Next, the dynamic constraints are introduced. They can all be derived starting from (2) and (3).

1) Dynamic Constraints for Parameter a: If a_j is *increased* (i.e., MF j becomes wider), the upper limit satisfying the γ-condition is

$$a_{\gamma,j} = \frac{d_{{\rm center},j}}{(\kappa(\gamma))^{1/2b_j}}$$

where $\kappa(\gamma)$ and $d_{{\rm center},j}$ are computed according to (4) and (8), respectively.

The upper limit satisfying the α-condition is

$$a_{\alpha,j} = \frac{d_{\alpha,j}}{(\kappa(\alpha))^{1/2b_j}}$$

where

$$d_{\alpha,j} = \begin{cases} \min(I_{\rm L}(\alpha, \mathbf{p}_{j+1}) - c_j, c_j - I_{\rm R}(\alpha, \mathbf{p}_{j-1})), & \text{if } 1 < j < M_{\rm A} \\ c_j - I_{\rm R}(\alpha, \mathbf{p}_{j-1}), & \text{if } j = M_{\rm A} \\ I_{\rm L}(\alpha, \mathbf{p}_{j+1}) - c_j, & \text{if } j = 1 \end{cases}$$

is the minimum distance from c_j to the point in which a neighboring MF receives membership value α. $I_{\rm L}$, and $I_{\rm R}$ are computed according to (2) and (3), respectively.

If a_j is *decreased* (i.e., MF j becomes more narrow), the lower limit satisfying the β-condition is

$$a_{\beta,j} = \begin{cases} \dfrac{d_{\beta,j}}{(\kappa(\beta))^{1/2b_j}}, & \text{if } d_{\beta,j} > 0 \\ a_{\rm low}, & \text{if } d_{\beta,j} \leq 0 \end{cases}$$

where

$$d_{\beta,j} = \begin{cases} \max(I_{\rm L}(\beta, \mathbf{p}_{j+1}) - c_j, c_j - I_{\rm R}(\beta, \mathbf{p}_{j-1})), & \text{if } 1 < j < M_{\rm A} \\ \max(\chi_{\rm high} - c_j, c_j - I_{\rm R}(\beta, \mathbf{p}_{j-1})), & \text{if } j = M_{\rm A} \\ \max(c_j - \chi_{\rm low}, I_{\rm L}(\beta, \mathbf{p}_{j+1}) - c_j), & \text{if } j = 1 \end{cases}$$

is computed depending on the location of MF j. If $d_{\beta,j} \leq 0$, UoD will be strongly covered regardless of the decrement in the value of a_j. In this case, the lower limit satisfying the β-condition is simply the static constraint $a_{\rm low}$.

Combining the constraints yields to

$$\max(a_{\rm low}, a_{\beta,j}) \leq a_j \leq \min(a_{\gamma,j}, a_{\alpha,j}, a_{\rm high}).$$

2) Dynamic Constraints for Parameter b: If b_j is *increased* (i.e., MF j becomes crisper), the following upper limit guarantees the fulfillment of α-condition:

$$b_{\alpha,j} = \begin{cases} \dfrac{\ln \kappa(\alpha)}{2 \ln (d_{\alpha,j}/a_j)}, & \text{if } d_{\alpha,j} < a_j \\ b_{\rm high}, & \text{if } d_{\alpha,j} \geq a_j. \end{cases}$$

If $d_{\alpha,j} \geq a_j$, MF j receives at most a membership value α at any intersection point, regardless of the increment in the value of b_j. In this case, the upper limit is the static constraint $b_{\rm high}$.

The following upper limit guarantees the fulfillment of β-condition:

$$b_{\beta,j} = \begin{cases} \dfrac{\ln \kappa(\beta)}{2\ln\left(d_{\beta,j}/a_j\right)}, & \text{if } d_{\beta,j} > a_j \\ b_{\text{high}}, & \text{if } d_{\beta,j} \leq a_j. \end{cases}$$

If $d_{\beta,j} \leq a_j$, MF j receives at least a membership value β at the intersection point(s) of its neighboring MF(s), which is regardless of the increment in b_j. In this case, the upper limit is the static constraint b_{high}.

If b_j is *decreased* (i.e., MF j becomes fuzzier), the following lower limit satisfies the γ-condition:

$$b_{\gamma,j} = \frac{\ln \kappa(\gamma)}{2\ln(d_{\text{center},j}/a_j)}.$$

Combining the constraints yields to[1]

$$\begin{aligned} &\max(b_{\text{low}}, b_{\gamma,j}) \leq b_j, && \text{if } b_j \geq b_{\text{high}} \\ &\max(b_{\text{low}}, b_{\gamma,j}) \leq b_j \leq \min(b_{\text{high}}, b_{\alpha,j}, b_{\beta,j}), && \\ & && \text{if } b_j < b_{\text{high}}. \end{aligned}$$

3) Dynamic Constraints for Parameter c: If c_j is *increased* (MF j is moving toward right), the following upper limit guarantees the fulfillment of α-condition (only the right α-condition needs to be taken into account):

$$c^{+}_{\alpha,j} = \begin{cases} c_{\text{high}}, & \text{if } j = M_{\text{A}} \\ c_j + (I_{\text{L}}(\alpha, \mathbf{p}_{j+1}) - I_{\text{R}}(\alpha, \mathbf{p}_j)), & \text{if } j < M_{\text{A}}. \end{cases}$$

Furthermore, the following upper limit guarantees the fulfillment of β-condition (only the left β-condition needs to be taken into account):

$$c^{+}_{\beta,j} = \begin{cases} c_j + (I_{\text{R}}(\beta, \mathbf{p}_{j-1}) - I_{\text{L}}(\beta, \mathbf{p}_j)), & \text{if } j > 1 \\ c_j + (\chi_{\text{low}} - I_{\text{L}}(\beta, \mathbf{p}_j)), & \text{if } j = 1. \end{cases}$$

Finally

$$c^{+}_{\gamma,j} = \begin{cases} c_{\text{high}}, & \text{if } j = M_{\text{A}} \\ c_j + \min(c_{j+1} - I_{\text{R}}(\gamma, \mathbf{p}_j), I_{\text{L}}(\gamma, \mathbf{p}_{j+1}) - c_j), & \\ & \text{if } j < M_{\text{A}} \end{cases}$$

is the upper limit guaranteeing the fulfillment of γ-condition (only the right γ-condition needs to be taken into account).

If the value of c is *decreased* (MF j is moving toward left), the applied constraints are

$$c^{-}_{\alpha,j} = \begin{cases} c_j - (I_{\text{L}}(\alpha, \mathbf{p}_j) - I_{\text{R}}(\alpha, \mathbf{p}_{j-1})), & \text{if } j > 1 \\ c_{\text{low}}, & \text{if } j = 1 \end{cases}$$

$$c^{-}_{\beta,j} = \begin{cases} c_j - (I_{\text{R}}(\beta, \mathbf{p}_j) - \chi_{\text{high}}), & \text{if } j = M_{\text{A}} \\ c_j - (I_{\text{R}}(\beta, \mathbf{p}_j) - I_{\text{L}}(\beta, \mathbf{p}_{j+1})), & \text{if } j < M_{\text{A}} \end{cases}$$

$$c^{-}_{\gamma,j} = \begin{cases} c_j - \min(I_{\text{L}}(\gamma, \mathbf{p}_j) - c_{j-1}, c_j - I_{\text{R}}(\gamma, \mathbf{p}_{j-1})), & \\ & \text{if } j > 1 \\ c_{\text{low}}, & \text{if } j = 1. \end{cases}$$

Combining the constraints yields to

$$\max\left(c^{-}_{\alpha,j}, c^{-}_{\beta,j}, c^{-}_{\gamma,j}\right) \leq c_j \leq \min\left(c^{+}_{\alpha,j}, c^{+}_{\beta,j}, c^{+}_{\gamma,j}\right).$$

[1]Recall that b_j can be increased only if $b_j < b_{\text{high}}$.

D. Genetic Operators

Five mutation and crossover operators are used. Some of them are not always applicable; therefore, when mutation or crossover is applied, one of the currently applicable operators is randomly selected by uniform chance. Crossover is applied with probability $P_c = 0.1 + (G/G_{\text{Tot}})$, where G is the current generation, and G_{Tot} is the total number of generations. If crossover was applied, mutation is applied with probability $P_m = 0.1$, and if crossover was not applied, mutation is always applied. This strategy is similar to strategy applied in [3].

Upper and lower limits for each modified parameter are computed according to Sections V-B and C and denoted by L_{upper} and L_{lower}. Number of currently active MFs in an input variable is denoted by M_{A} and a random real number by $r \in [0, 1]$.

1) Mutation Operators: Operator 1 modifies the parameters of input MFs. First, the number of input variables that have at least two active MFs is determined. This number is denoted here by n_{active}. Then, out of n_{active} input variables, n_{select} of them are randomly selected, where $n_{\text{select}} \in [1, n_{\text{active}}]$ is a random integer. From each of these n_{select} input variables, an active MF is randomly selected. Then, for each of them, a gbell parameter (a, b, or c) is randomly selected. They are denoted by $p_{i,l}$, where i is 1, 2, or 3 depending upon which gbell parameter is modified, and l is the index of an active MF in P [see (17) and (18)]. Each $p_{i,l}$ is replaced by randomly selecting one of the following replacement formulas: $p_{i,l} \leftarrow p_{i,l} + r(L_{\text{upper}} - p_{i,l})$ or $p_{i,l} \leftarrow p_{i,l} - r(p_{i,l} - L_{\text{lower}})$.

Operator 2: The mutation operator 1 modifies input MF parameters individually; however, sometimes more drastic modification may be necessary. Therefore, this operator selects an input variable for which $M_{\text{A}} \geq 2$ and creates a new unevenly distributed partition with M_{A} MFs using the algorithm defined in Section III-C.

Operator 3 modifies the rule base by randomly selecting n_{rulecond} rule conditions $a_{i,j}$ [see (16)], where $n_{\text{rulecond}} \in [1, 10]$ is a random integer. The selected rule conditions are replaced with random rule conditions; however, as it is easier to obtain compact than accurate FMs [8], this operator favors nonzero-replacement conditions during the first half of the total number of generations G_{Tot}. Therefore, if $G < G_{\text{Tot}}/2$, then the probability that a replacement condition is selected from $[0, M_{\max_j}]$ is $P_z = 2G/G_{\text{Tot}}$, and the probability that it is selected from $[1, M_{\max_j}]$ is $1 - P_z$. When $G \geq G_{\text{Tot}}/2$, replacement conditions are always selected from $[0, M_{\max_j}]$.

The resulting input fuzzy partition may not be transparent if some MFs have become active or inactive, thus resulting into highly overlapping MFs or gaps in the fuzzy partition. Thus, the set of these input variables that use different MFs in the rules than before this operator is determined. Then, M_{A} for each of these input variables is determined. For these input variables for which $M_{\text{A}} \geq 2$, new unevenly distributed partition with M_{A} MFs is created. If $M_{\text{A}} < 2$, all nonzero conditions, if any, of that input variable are forced to zero. This operation is called *repair operator*, and it guarantees transparency of input fuzzy partition.

Operator 4 modifies a consequent s_i, where $i = 1, \ldots, R$, of a randomly selected active rule by replacing it by random consequent chosen from $[1, M_{\text{out}}]$. A rule is active if it has at least one nonzero-rule condition.

Operator 5 modifies the lateral displacement of a randomly selected active-output-MF center (an output MF is active if it is used in at least one of the active rules). The selected output-MF center o_i, where $i = 1, \ldots, M_{\text{out}}$, is replaced by randomly selecting one of the following formulas: $o_i \leftarrow o_i + r(L_{\text{upper}} - o_i)$ or $o_i \leftarrow o_i - r(o_i - L_{\text{lower}})$.

2) Crossover Operators: All five crossover operators randomly select two FMs as parents and produce two FMs as children. They replace their parents in the offspring population. The crossover operators 1, 4, and 5 resemble the mutation operators 1, 4, and 5.

Operator 1 modifies the parameters of active input MFs using BLX-0.5 crossover [23], [32]. It can be applied to input variables, which have the same amount (at least 2) of active MFs in both parents. The number of input variables meeting these requirements is denoted by n_{active}. Out of them, n_{select} are randomly selected, where $n_{\text{select}} \in [1, n_{\text{active}}]$ is a random integer. For each of these n_{select} input variables, an active MF $j \in [1, M_{\text{A}}]$ is randomly selected (the same j from both parents). Then, from each of these selected active MFs, a gbell parameter (a, b, or c) is randomly selected (the same parameter from both parents).

Let p^1_{i,l_1} and p^2_{i,l_2} denote the selected parameters from parents 1 and 2, respectively. The index i is 1, 2, or 3 depending on which gbell parameter is selected [see (17)]. The indexes l_1 and l_2 are determined according to (18). The parameters are replaced by randomly selecting either $p^k_{i,l_k} \leftarrow p^k_{i,l_k} + r(\min(I, L_{\text{upper}} - p^k_{i,l_k}))$ or $p^k_{i,l_k} \leftarrow p^k_{i,l_k} - r(\min(I, p^k_{i,l_k} - L_{\text{lower}}))$, where $k = 1$ and 2, and $I = 0.5|p^1_{i,l_1} - p^2_{i,l_2}|$.

Operator 2: First, an input variable, for which at least one of the parents has at least two active MFs, is randomly selected. After this, all rule conditions and input MF parameters of this input variable are pairwisely swapped. Therefore, child 1 receives all the parameters of parent 1, except rule conditions and input MF parameters of the selected input variable, which are received from parent 2. Likewise, child 2 gets all the parameters of parent 2, except rule conditions and input MF parameters of the selected input variable, which are received from parent 1.

Operator 3 swaps some rules of the parents. It is applicable to those rules that are active in at least one of the parents. Out of these rules, N_{select} of them are selected and their rule conditions are pairwisely swapped (N_{select} is a random integer chosen from $[1, 5]$).

After this operator, input fuzzy partitions may not be transparent. Therefore, for both children separately, the same repair operator as with the mutation operator 3 is applied.

Operator 4 modifies the rule consequents s_i, where $i = 1, \ldots, R$. This operator is possible for those rules that are active in at least one of the parents. The operator selects one of these rules randomly and swaps consequents of this rule.

Operator 5 modifies the lateral displacement of output MF centers. This operator is possible for those output MF centers that are active in both of the parents. Out of them, one is randomly selected from both parents (the same from both parents). They are denoted here by o^1_i and o^2_i, where $i = 1, \ldots, M_{\text{out}}$. They are replaced by randomly selecting one of the following formulas: $o^k_i \leftarrow o^k_i + r(\min(I, L_{\text{upper}} - o^k_i))$ or $o_i \leftarrow o^k_i - r(\min(I, (o^k_i - L_{\text{lower}}))$, where $k = 1$ and 2, and $I = 0.5|o^1_i - o^2_i|$.

TABLE II
PROPERTIES OF THE DATASETS AND THE APPLIED PARAMETERS

Data	Input variables	Data points	M_{in}	M_{out}
Electrical Length (Ele1)	2	495	5	5
Electrical Maintenance (Ele2)	4	1056	5	5
Mackey-Glass (MG)	4	500	5	5
Lorenz	4	500	5	5
Abalone	8	4177	3	3
Box-Jenkins Gas Furnace (Gas)	10	290	5	5
Treasury	15	1049	3	3
Mortgage	15	1049	3	3
Computer Activity (Computer)	21	8192	3	3

VI. EXPERIMENTS

Our multiobjective GFS is validated using nine datasets, which represent different number of input variables and data points (see Table II). For all datasets, five-fold cross-validation was repeated six times (6×5CV) with different random seeds. The data partitions for Ele1, Ele2, Abalone, Mortgage, Treasury, and Computer problems were downloaded from KEEL Website.[2] MG and Lorenz datasets were generated according to [3] and [20]. Finally, Gas dataset was obtained from the Website of Greg Reinsel.[3] For Mackey–Glass (MG), Lorenz, and Gas problems, the same data partitions as in the comparative study [20] were used. C4.5 was run with its default parameters defined in [29]. Population size was fixed to 100 and the number of generations was altered, such that, the same amount of fitness evaluations was used as in the comparative studies. The settings $\alpha = 0.8$, $\beta = 0.05$, and $\gamma = 0.25$ are used in the experiments performed in Section VI-B–F. Furthermore, in VI-G, experiments with $\alpha = 0.6$, $\beta = 0.4$, and $\gamma = 0.1$ will be performed in order to study the tradeoff between transparency of fuzzy partitions and accuracy.

For six of the datasets (Ele1, Ele2, MG, Lorenz, Abalone, and Gas), there exist results of one or more recent GFSs presented in Table III. For these problems, the number of input and output MFs (M_{in} and M_{out}) were selected the same as in the comparative studies. For treasury, mortgage, and computer problems, our method is compared against a baseline method. For these higher dimensional problems, M_{in} and M_{out} were both set to 3 in order to reduce the search space.

Since MOEAs are applied, it is interesting to visualize the Pareto fronts. However, it is not meaningful to visualize the Pareto fronts of all 30 CV runs for each dataset. The averaged results of the ith most accurate FMs from each of the 30 Pareto fronts were shown in [8] for five of the most accurate FMs (i.e., $i = 1, \ldots, 5$). These averages were computed,

[2] http://sci2s.ugr.es/keel/datasets.php
[3] http://www.stat.wisc.edu/~reinsel/bjr-data/index.html

TABLE III
PROPERTIES OF THE COMPARATIVE GFSS

Ref. (Year)	Name	MFs tuning	Rule selection	Rule learning	Input variable selection: Initialization	Learning	Transparency of fuzzy partition
[6] (2007)	GL	2-tuple	No	No	No	No	Best
	GL+S	2-tuple	**Yes**				Best
	GLA	3-tuple	No				Good
	GLA+S	3-tuple	**Yes**				Good
[8], [23] (2007-9)	TS-NSGA-II	3-parameter + SSI	**Yes**	No	No	No	Average
	TS-SPEA2$_{\text{Acc}}$						
	TS-NSGA-II$_A$						
	TS-NSGA-II$_U$						
	TS-SPEA2						
	TS-SPEA2$_{\text{Acc}^2}$						
[20] (2009)	C4.5+NSGA-II	3-parameter + SLI	No	**Yes**	**Yes**	**Yes**	Poor
	This paper	Dynamic 3-parameter	No	**Yes**	**Yes**	**Yes**	Best

SSI and SLI stand for static constraints with small- and large-variation intervals, respectively.

such that, none of the 30 Pareto fronts were excluded from computing the averages. Thus, in each of the Pareto fronts, there were at least five distinct FMs. In this paper, the maximum value of i ($i_{\max}$) is not the same for all datasets, but depends on the minimum number of distinct FMs on the 30 Pareto fronts. More formally, $i_{\max} = \min(L_1, L_2, \ldots, L_{30})$, where L_j, with $j = 1, \ldots, 30$, is the number of distinct FMs on the jth Pareto front of a given dataset. Thus, the length of the averaged Pareto front equals to the length of the shortest Pareto front of the 30 runs.

Besides the Pareto fronts, the number of rules R, rule conditions R_{cond}, input MFs, and the number of input variables F for some of the ith most accurate FMs are tabulated. Moreover, the unequal variance t-test[4] (denoted by t) with 95% confidence is reported for the MSE_{trn} and MSE_{tst}. The same notations as in [6], [8], and [23] are used; $\star$ stands for the best averaged result in the column, $+$ means that the performance of the corresponding row is worse than the best result, and $=$ means that there is no significant statistical difference compared to the best result.

A. Comparative Genetic Fuzzy Systems

The comparative approaches global lateral tuning (GL), global lateral tuning with rule selection (GL+S), global lateral amplitude tuning (GLA), and global lateral amplitude tuning with rule selection (GLA+S) minimize only one objective, namely, MSE, whereas the rest minimize two or more objectives simultaneously and obtain a set of Pareto optimal FMs. All GFSs use globally defined MFs. The approaches [6], [8], [23] create the initial populations using WM algorithm, whereas in [20], C4.5 algorithm is used. In this paper, the initial population is created by a method combining the benefits of C4.5 and WM algorithms.

Performance of GFS designs depends on their individual components, such as initialization method and MFs tuning strategy. For example, by applying different initialization methods, performance of a GFS can be significantly improved or deteriorated. This is because appropriate initialization eases the derivation of better FMs due to reduction in the search space [15]. Because this paper and the comparative studies apply different initialization methods, the purpose of the results comparisons is not to assess the superiority of any individual components, but to assess the superiority of different approaches as a whole. Assessing the superiority of individual components is, of course, important, but requires another study in the future. It should be noted, however, that the results comparisons can be considered fair, because the same amount of fitness evaluations, the same data partitions, and the same amount of input and output MFs are used, as in the comparative studies. Also, our approach does not require any more *a priori* knowledge about the datasets than the comparative methods.

To evaluate the transparency of fuzzy partitions, we follow [6], which states that two-tuple representation leads to more transparent fuzzy partition than three-tuple representation. Moreover, three-tuple representation is more transparent than classic three-parameter representation with static constraints. In [8] and [23], static constraints were defined, such that, MF parameters can vary within small intervals, whereas in [20], larger intervals were used. Therefore, we consider the transparency of fuzzy partitions in [20], which is the poorest among the comparative GFSs. Both two-tuple presentation and the proposed dynamic constraint approach maintain the transparency of fuzzy partitions at a good level. Since the approaches are quite different and because transparency of fuzzy partitions is a subjective matter, it is difficult to judge which one of them yields into more transparent fuzzy partitions. Therefore, their interpretability is considered equal.

B. Estimating the Length of Low-Voltage Lines (Ele1)

For this problem, 50 000 fitness evaluations were used in this paper and in [6]. Table IV shows that GLA+S has the lowest MSE_{trn}, and our most accurate FM (Final-1) has practically the same value. There is no statistical difference between the lowest MSE_{trn} and three of our most accurate FMs (Final-1, Final-2, and Final-3). The lowest MSE_{tst} is obtained by GLA, but again, there is no statistical difference between the lowest MSE_{tst} and three of our most accurate FMs. There is no clear

[4] Also called Welch's t-test [33], [34]. If our multiobjective GFS could be compared to other GFSs in all problems, nonparametric tests would be preferred.

TABLE IV
RESULTS COMPARISON FOR ELE1 PROBLEM

Ref.	Method	R	R_{cond}	MFs	F	$\mathrm{MSE}_{\mathrm{trn}}$	σ_{trn}	t	$\mathrm{MSE}_{\mathrm{tst}}$	σ_{tst}	t
[6]	GL	12.4	24.8	N/A	2.0	166674	11480	+	189216	**14743**	=
	GL+S	**9.0**	18	N/A	2.0	160081	**7316**	+	189844	22448	=
	GLA	12.4	24.8	N/A	2.0	157604	9158	=	**185810**	18812	$\star$
	GLA+S	10.2	20.4	N/A	2.0	**155404**	9264	$\star$	189472	20393	=
This paper	Initial	14.4	28.2	9.0	2.0	272050	17861	+	310160	45892	+
	Final-1	12.3	19.8	8.9	2.0	155785	9356	=	192319	31315	=
	Final-2	11.5	17.8	8.9	2.0	156911	10064	=	192813	31128	=
	Final-3	11.1	**16.2**	8.7	2.0	160178	12273	=	195730	40453	=

TABLE V
RESULTS COMPARISON FOR ELE2 PROBLEM

Ref.	Method	R	R_{cond}	MFs	F	$\mathrm{MSE}_{\mathrm{trn}}$	σ_{trn}	t	$\mathrm{MSE}_{\mathrm{tst}}$	σ_{tst}	t
					50000 fitness evaluations						
[6]	GL	65	260	N/A	4.0	23064	1479	+	25654	2611	+
	GLA	65	260	N/A	4.0	17950	1889	+	21212	2686	+
	GLA+S	49.4	197.6	N/A	4.0	17538	2391	+	21491	4168	+
	GL+S	49.1	196.4	N/A	4.0	18801	2669	+	22586	3550	+
[23]	TS-NSGA-II$_{\mathrm{Acc}}$	48.1	192.4	N/A	4.0	16321	1636	+	20423	3138	+
	TS-NSGA-II	41	164	N/A	4.0	14488	**965**	+	18419	3054	+
	TS-SPEA2$_{\mathrm{Acc}}$	34.5	138	N/A	4.0	11081	1186	=	14161	2191	+
	TS-SPEA2	33	132	N/A	4.0	13272	1265	+	17533	3226	+
This paper	Initial	26.2	61.6	15.4	3.4	74719	15065	+	74274	12069	+
	Final-1	25.0	51.5	14.3	3.1	**10861**	1436	$\star$	**12336**	**2065**	$\star$
	Final-2	24.8	49.7	14.2	**3.0**	11314	1490	=	12731	2103	=
	Final-3	24.5	47.5	14.2	**3.0**	11942	1602	+	13551	2560	+
	Final-4	24.1	45.6	13.9	**3.0**	12742	2016	+	14080	3199	+
	Final-5	**23.9**	**44.0**	13.8	**3.0**	13571	2362	+	15100	3165	+
					100000 fitness evaluations						
[8]	TS-SPEA2$_{\mathrm{Acc}^2}$-1	29.8	119.2	N/A	4.0	10325	1121	+	13935	2759	+
	TS-SPEA2$_{\mathrm{Acc}^2}$-2	28.3	113.2	N/A	4.0	10496	1126	+	14268	2925	+
	TS-SPEA2$_{\mathrm{Acc}^2}$-3	27.0	108.0	N/A	4.0	10835	1191	+	14460	2782	+
	TS-SPEA2$_{\mathrm{Acc}^2}$-4	25.9	103.6	N/A	4.0	11217	1307	+	14806	3069	+
	TS-SPEA2$_{\mathrm{Acc}^2}$-5	24.9	99.6	N/A	4.0	12194	2078	+	15417	3328	+
This paper	Final-1	24.9	49.1	14.3	3.1	**9366**	**887**	$\star$	**10429**	**1646**	$\star$
	Final-2	24.3	46.5	14.2	3.1	9619	957	=	10713	1816	=
	Final-3	23.3	44.0	14.0	3.0	10007	1117	+	11270	1819	=
	Final-4	22.9	41.7	13.6	3.0	10339	1199	+	11663	1773	+
	Final-5	**22.2**	**39.4**	13.4	**2.9**	10948	1412	+	12261	2190	+

difference between different approaches for this problem, because the search space is small due to small amount of input variables. It is also noticed that although M_{in} was set to 5, the initial FM uses on average nine input MFs. Therefore, one of the input variables is usually partitioned with four and the other one with five input MFs.

C. Estimating the Maintenance Costs of Medium-Voltage Lines (Ele2)

This problem is more interesting as it contains four input variables. First, our multiobjective GFS was run for 50 000 fitness evaluations and compared to [6] and [23], which use the same amount of fitness evaluations. Table V shows that our multiobjective GFS has the lowest MSE in train and test sets. There is also statistical difference between our approach and all other approaches when $\mathrm{MSE}_{\mathrm{tst}}$ is considered. When $\mathrm{MSE}_{\mathrm{trn}}$ is considered, there is statistical difference between our approach and all other approaches, except TS-SPEA2$_{\mathrm{Acc}}$. Our FMs can also be considered as the most interpretable, because they are clearly the most compact, and the transparency of fuzzy partitions is at least the same as in the comparative FMs (see also Table III).

Our approach was also run for 100 000 fitness evaluations (the same amount as in [8]). Table V shows that our FMs are the most accurate according to t-test. They are also clearly more compact than the FMs in [8]. Finally, because in [8], three-parameter MFs tuning with static constraints was used, the fuzzy partitions of our FMs can be considered more transparent.

D. Predicting the Age of Abalone

This problem has eight input variables and a very high noise level. According to [8], usually the learning methods yields into similar accuracy. Thus, it may not be possible to improve the accuracy, but only to improve the interpretability, compared to existing methods in the literature. In this paper and also in the comparative study [8], the number of fitness evaluations was set to 100 000. According to Table VI, there is no clear difference in accuracy between different GFSs. The lowest $\mathrm{MSE}_{\mathrm{trn}}$ was obtained by TS-SPEA2$_{\mathrm{Acc}}$ and the lowest $\mathrm{MSE}_{\mathrm{tst}}$ by our approach (Final-1). On the other hand, our approach presents a significant improvement in interpretability. Our FMs are clearly more compact than the comparative FMs. They have much less rule conditions and use much less input variables. Furthermore, according to Table III, our fuzzy partitions can be considered more transparent than the fuzzy partitions in [8].

TABLE VI
RESULTS COMPARISON FOR ABALONE PROBLEM

Ref.	Method	R	R_{cond}	MFs	F	$\mathrm{MSE}_{\mathrm{trn}}$	σ_{trn}	t	$\mathrm{MSE}_{\mathrm{tst}}$	σ_{tst}	t
[8]	TS-NSGA-II	22.4	179.2	N/A	8.0	2.398	0.084	=	2.526	0.242	=
	TS-SPEA2$_{\mathrm{Acc}}$	22.2	177.6	N/A	8.0	**2.368**	0.085	⋆	2.511	0.263	=
	TS-NSGA-II$_A$	22.1	176.8	N/A	8.0	2.404	0.098	=	2.535	0.265	=
	TS-NSGA-II$_U$	21.8	174.4	N/A	8.0	2.407	0.082	=	2.520	0.237	=
	TS-SPEA2	20.0	160.0	N/A	8.0	2.383	0.078	=	2.518	0.246	=
	TS-SPEA2$_{\mathrm{Acc}^2}$	18.6	148.8	N/A	8.0	2.372	0.075	=	2.517	0.230	=
This paper	Initial	30.0	130.6	16.4	6.6	7.946	2.879	+	7.906	2.742	+
	Final-1	20.5	43.3	10.7	4.4	2.389	**0.074**	=	**2.423**	0.173	⋆
	Final-2	19.9	40.9	10.6	4.4	2.393	**0.074**	=	2.426	0.175	=
	Final-3	19.2	39.0	10.5	4.3	2.400	0.078	=	2.432	**0.171**	=
	Final-4	18.4	35.7	10.0	4.1	2.409	0.083	=	2.437	0.184	=
	Final-5	**17.8**	**32.8**	9.9	**4.0**	2.419	0.090	+	2.444	0.197	=

TABLE VII
RESULTS COMPARISON FOR MG, LORENZ, AND GAS PROBLEMS

Problem	Method	R	R_{cond}	MFs	F	$\mathrm{MSE}_{\mathrm{trn}}$	σ_{trn}	t	$\mathrm{MSE}_{\mathrm{tst}}$	σ_{tst}	t
MG $F=4$	C4.5+NSGA-II [20]	**12.0**	**26.1**	17.9	4.0	5.3e-4	9.0e-5	+	6.6e-4	1.8e-4	+
	Initial	30.0	103.5	19.6	4.0	5.6e-3	1.6e-3	+	5.6e-3	1.6e-3	+
	Final-1	27.4	64.8	18.8	4.0	**1.9e-4**	**5.1e-5**	⋆	**2.3e-4**	**6.3e-5**	⋆
	Final-8	24.5	44.7	**17.5**	**3.8**	3.2e-4	1.5e-4	+	3.8e-4	1.5e-4	+
Lorenz $F=4$	C4.5+NSGA-II [20]	**6.6**	**9.5**	**8.8**	2.3	0.144	0.042	+	0.233	0.089	+
	Initial	18.0	35.8	11.4	2.6	2.281	1.021	+	2.371	0.823	+
	Final-1	15.6	25.6	10.7	2.3	**0.076**	**0.016**	⋆	**0.129**	**0.047**	⋆
	Final-4	13.2	19.4	10.2	**2.2**	0.101	0.036	+	0.169	0.084	+
Gas $F=10$	C4.5+NSGA-II [20]	**6.8**	**11.0**	**10.0**	3.7	0.104	0.014	+	0.140	0.040	+
	Initial	23.0	60.8	13.8	4.0	0.664	0.243	+	0.651	0.189	+
	Final-1	20.4	39.3	12.3	3.3	**0.060**	**0.007**	⋆	**0.080**	**0.022**	⋆
	Final-9	16.8	25.0	10.8	**2.9**	0.077	0.024	+	0.092	0.033	=

E. Mackey–Glass, Lorenz, and Gas Problems

Our multiobjective GFS is compared to our former multiobjective GFS [20], which was run for 210 000 fitness evaluations on these problems. The same amount of fitness evaluations is used here. Table VII shows that our most accurate FMs are significantly more accurate than the most accurate FMs of our former study. On the other hand, they also contain much more rules and rule conditions than FMs in [20].

The least accurate FMs of the averaged Pareto fronts for each problem are also presented and denoted by Final-8, Final-4, and Final-9. One can notice that they are still more accurate than the most accurate FMs in [20]. On the other hand, they are also more complex with regards to number of rules and rule conditions. The number of input variables and the number of MFs is approximately the same. Table III shows that the FMs in [20] have the worst transparency of fuzzy partitions and our FMs have the best.

F. Higher Dimensional Problems: Treasury, Mortgage, and Computer Activity

Our approach was run for 100 000 fitness evaluations on these problems. To the best of our knowledge, there are no results of other GFSs available for these problems.[5] Nonetheless, it is important to include a baseline method in order to have an idea about the performance of our approach. Thus, *Genfis3*, a fuzzy-*c*-means (FCM) clustering-based method was used to identify Mamdani FMs. This method is part of MATLAB's Fuzzy Logic Toolbox 2. All settings, besides the type of FM, were kept at their default values and 6×5CV with the same data partitions as with our multiobjective GFS was performed.

Table VIII shows that our FMs are significantly more accurate than the comparative FMs. Moreover, they have less input variables and input MFs than the comparative FMs. The comparative FMs usually have less rules, but more rule conditions, than our FMs. By visual inspection, it was noticed that the fuzzy partitions by Genfis3 often contain many highly overlapping MFs and the UoD may not be strongly covered.

G. Fuzzy Partition Transparency Versus Accuracy Tradeoff

The experiments in Section VI-B–VI-F were performed with $\alpha = 0.8$, $\beta = 0.05$, and $\gamma = 0.25$. If one requires higher transparency of fuzzy partitions, the settings $\alpha = 0.6$, $\beta = 0.4$, and $\gamma = 0.1$ could be used. The 6×5CV procedures for all nine problems were repeated with these settings. The averaged results of the most accurate FMs are shown in Table IX along with the best and the worst results from Tables IV–VIII. In Fig. 4, the averaged Pareto fronts for five of the studied problems are shown. It is seen from Table IX and Fig. 4 that by improving transparency of fuzzy partitions, accuracy is deteriorated, but remains at a reasonable level.

Transparency of fuzzy partitions is evaluated against a fuzzy partition, which has three desirable properties: 1) The membership values at the intersections of neighboring MFs are always

[5] At the time of writing the final version of this paper, this statement no longer holds true. There are recently published results available for some [26] and all [22] of these problems. However, the experimental setup in those papers differ significantly from the experimental setup of this paper. Thus, our results are not compared to them.

TABLE VIII
RESULTS COMPARISON FOR HIGHER DIMENSIONAL PROBLEMS

Problem	Method	R	R_{cond}	MFs	F	MSE_{trn}	σ_{trn}	t	MSE_{tst}	σ_{tst}	t
Treasury $F=15$	Genfis3	8.8	132.0	132.0	15.0	1.440	0.300	+	1.928	1.465	+
	Initial	8.8	19.2	7.4	3.4	1.203	0.437	+	1.128	0.810	+
	Final-1	**7.2**	**10.2**	**4.6**	**2.0**	**0.059**	**0.024**	$\star$	**0.093**	**0.082**	$\star$
Mortgage $F=15$	Genfis3	**8.8**	132.0	132.0	15.0	0.924	0.112	+	1.312	1.007	+
	Initial	19.4	55.0	10.6	4.4	0.737	0.444	+	0.449	0.150	+
	Final-1	16.5	**31.4**	**8.6**	**3.4**	**0.023**	**0.009**	$\star$	**0.039**	**0.027**	$\star$
Computer $F=21$	Genfis3	**3.2**	**67.2**	67.2	21.0	127.31	5.40	+	127.50	3.72	+
	Initial	30.0	219.7	38.8	15.6	452.73	494.18	+	453.80	496.53	+
	Final-1	25.4	82.4	**17.5**	**7.9**	**4.71**	**0.52**	$\star$	**4.86**	**0.46**	$\star$

TABLE IX
AVERAGED RESULTS OF THE MOST ACCURATE FMs USING $\alpha = 0.6$, $\beta = 0.4$, AND $\gamma = 0.1$

									Tables IV - VIII											
									Best according to MSE_{tst}						Worst according to MSE_{tst}					
Problem	R	R_{cond}	MFs	F	MSE_{trn}	σ_{trn}	MSE_{tst}	σ_{tst}	R	R_{cond}	MFs	F	MSE_{trn}	MSE_{tst}	R	R_{cond}	MFs	F	MSE_{trn}	MSE_{tst}
Ele1	12.9	20.7	9.0	2.0	160387	8513	194666	32353	12.4	24.8	N/A	2.0	157604	185810	11.1	16.2	8.7	2.0	160178	195730
Ele2_{50000}	25.1	48.9	14.3	3.2	15890	1675	17536	3028	25.0	51.5	14.3	3.1	10861	12336	65	260	N/A	4.0	23064	25654
Ele2_{100000}	24.4	44.2	13.8	3.0	13721	1804	15435	2781	24.9	49.1	14.3	3.1	9366	10429	24.9	99.6	N/A	4.0	12194	15417
MG	26.9	56.7	18.5	3.9	3.2e-4	1.0e-4	3.7e-4	1.3e-4	27.4	64.8	18.8	4.0	1.9e-4	2.3e-4	12.0	26.1	17.9	4.0	5.3e-4	6.6e-4
Lorenz	15.6	24.0	10.4	2.2	0.163	0.028	0.231	0.087	15.6	25.6	10.7	2.3	0.076	0.129	6.6	9.5	8.8	2.3	0.144	0.233
Abalone	21.8	44.5	10.6	4.3	2.407	0.075	2.452	0.182	20.5	43.3	10.7	4.4	2.389	2.423	22.1	176.8	N/A	8.0	2.404	2.535
Gas	20.1	37.7	12.3	3.3	0.075	0.010	0.095	0.023	20.4	39.3	12.3	3.3	0.060	0.080	6.8	11.0	10.0	3.7	0.104	0.140
Treasury	7.4	11.0	5.0	2.2	0.070	0.037	0.101	0.079	7.2	10.2	4.6	2.0	0.059	0.093	8.8	132.0	132.0	15.0	1.440	1.928
Mortgage	17.1	34.1	8.9	3.6	0.033	0.017	0.056	0.047	16.5	31.4	8.6	3.4	0.023	0.039	8.8	132.0	132.0	15.0	0.924	1.312
Computer	26.4	87.6	17.9	7.8	5.12	0.53	5.54	0.83	25.4	82.4	17.5	7.9	4.71	4.86	3.2	67.2	67.2	21.0	127.31	127.50

The best and the worst results in Tables IV–VIII exclude the results of initial FMs.

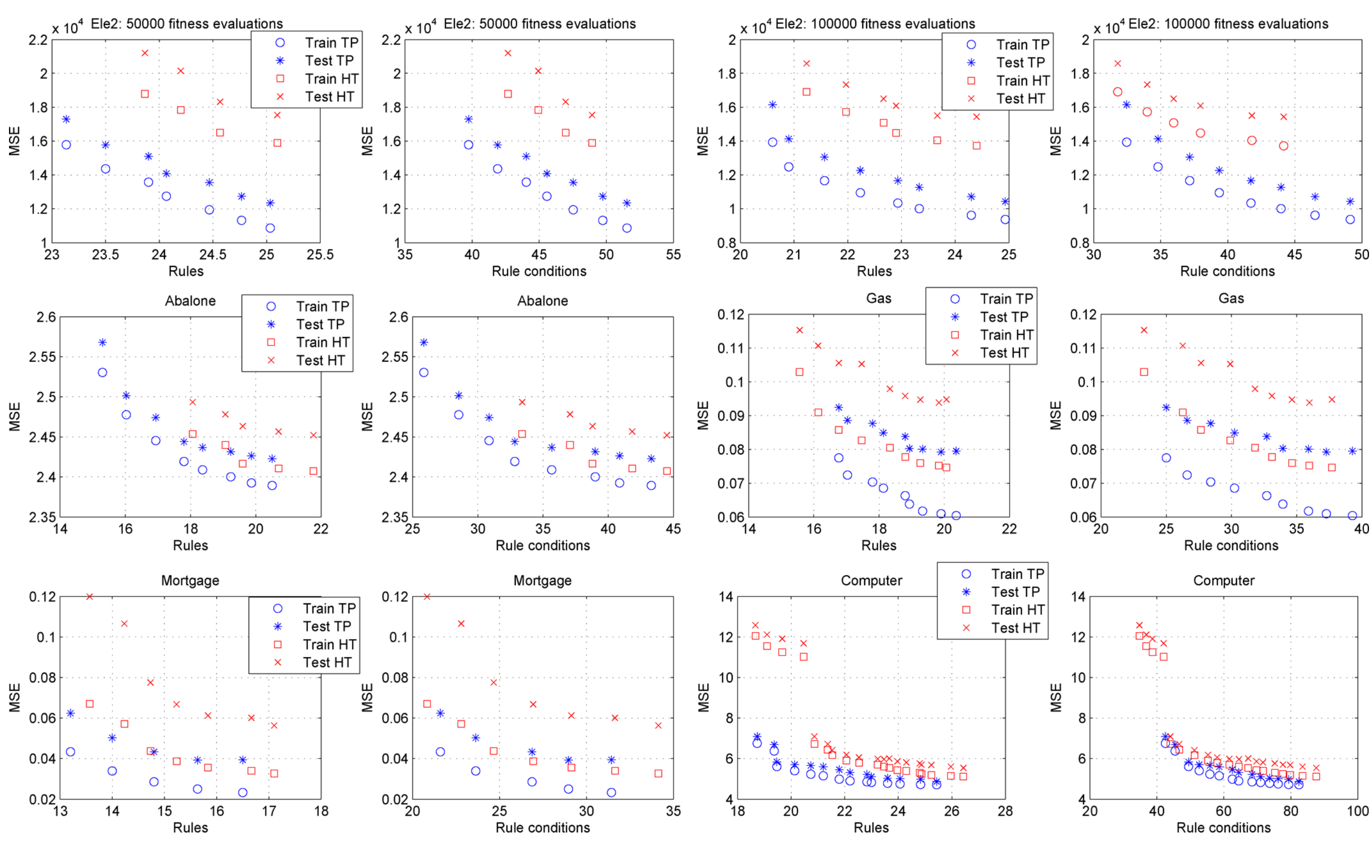
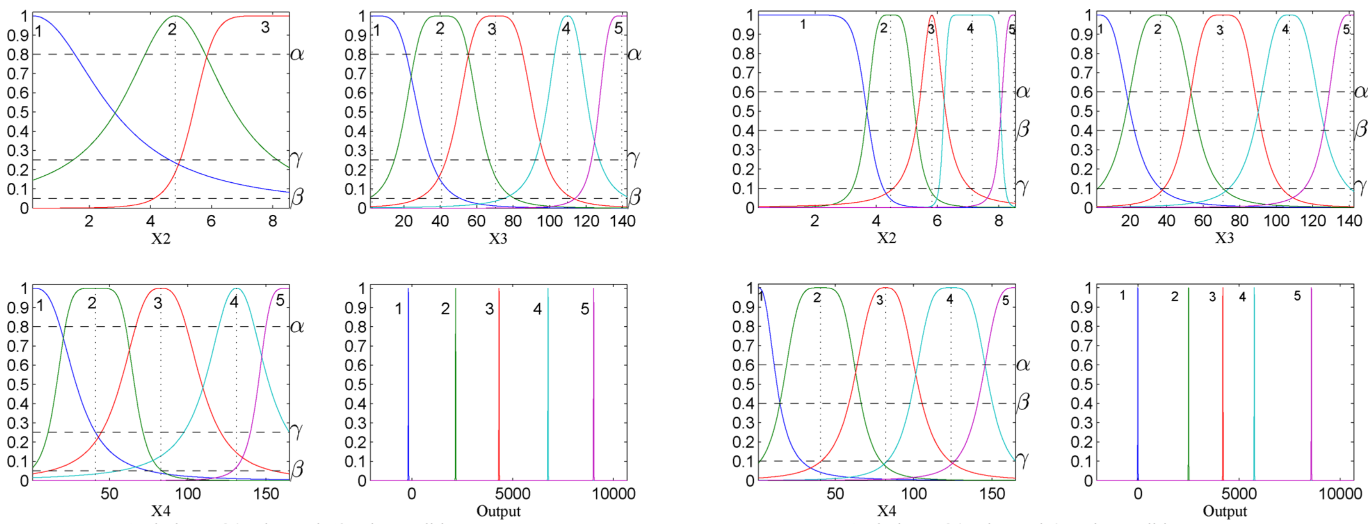


Fig. 4. Averaged Pareto fronts over 30 CV runs for Ele2 with 50 000 and 100 000 fitness evaluations, Abalone, Gas, Mortgage, and Computer problems. TP stands for transparent fuzzy partitions obtained by $\alpha = 0.8$, $\beta = 0.05$, and $\gamma = 0.25$, whereas HT stands for highly transparent fuzzy partitions obtained by $\alpha = 0.6$, $\beta = 0.4$, and $\gamma = 0.1$.

TABLE X
COMPARISON OF THE AVERAGED FUZZY-PARTITION QUALITY INDEXES OF THE MOST ACCURATE FMs AND THE AVERAGE LENGTH OF THE PARETO FRONTS WITH DIFFERENT SETTINGS OF α, β, AND γ

α	β	γ		Ele1	Ele2_{50000}	Ele2_{100000}	MG	Lorenz	Abalone	Gas	Mortgage	Treasury	Computer
			Q_{Int}	0.32	0.28	0.29	0.28	0.30	0.23	0.28	0.26	0.23	0.24
			Q_{Mid}	0.22	0.22	0.23	0.24	0.23	0.11	0.21	0.16	0.13	0.14
0.8	0.05	0.25	Q_{Ext}	0.08	0.05	0.12	0.05	0.05	0.03	0.05	0.05	0.01	0.02
			N_{D}	10.1	12.2	14.0	15.7	11.3	19.5	18.7	12.2	4.3	30.5
			$\sigma_{N_{\text{D}}}$	3.6	2.9	3.2	4.4	4.4	5.7	4.8	5.0	1.6	6.4
			Q_{Int}	0.10	0.09	0.09	0.10	0.10	0.08	0.09	0.08	0.09	0.08
			Q_{Mid}	0.10	0.09	0.09	0.10	0.10	0.07	0.09	0.08	0.08	0.08
0.6	0.4	0.1	Q_{Ext}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
			N_{D}	11.1	11.2	13.2	15.0	11.8	20.7	17.6	14.8	5.4	31.6
			$\sigma_{N_{\text{D}}}$	3.4	3.4	3.2	4.7	4.8	6.3	4.9	4.6	1.9	6.2



Rule base: 24 rules and 50 rule conditions.

X2	X3	X4	Output	X2	X3	X4	Output
	1	1	1		2	4	3
1	4	1	1		4	2	4
	2	1	2		3	3	4
3	2		2	2	4	3	4
2	2	3	2		4	3	4
	1	2	2	2	5		4
	2	2	2		3	4	5
	1	3	2	3		5	5
	2	3	3		4	4	5
	3	2	3		5	3	5
2	2	5	3		3	5	5
	3		3			5	5

Rule base: 24 rules and 45 rule conditions.

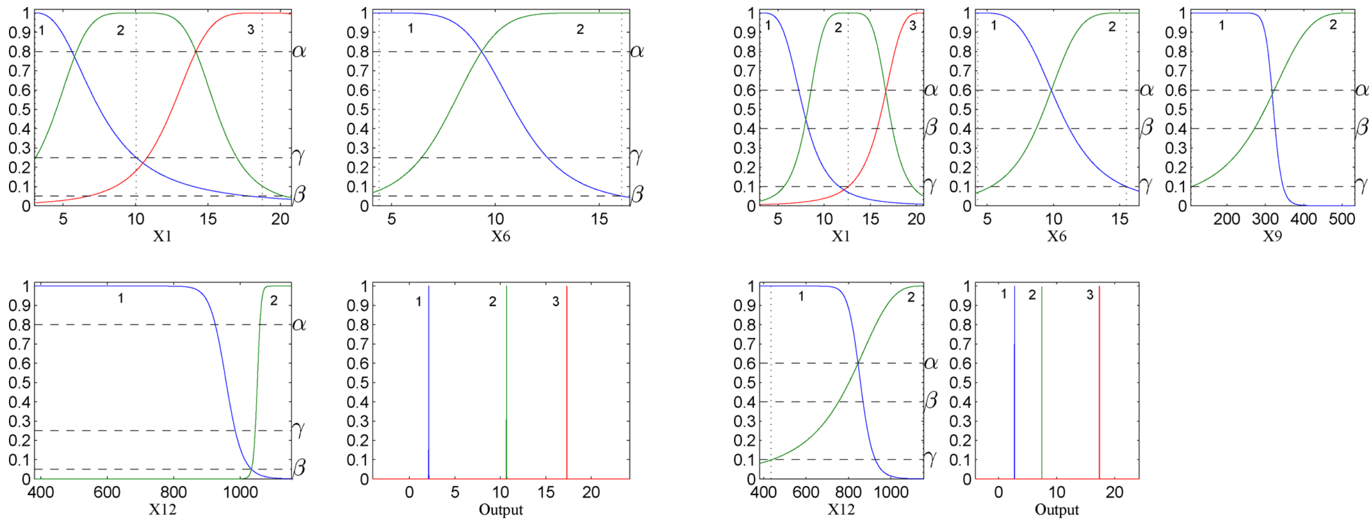
X2	X3	X4	Output	X2	X3	X4	Output
1	1		1		3	3	4
3	2		1	4	2		4
	1	1	1		4	2	4
	2	2	2		5	2	4
	1	2	2		4	3	4
2	2		2			5	5
	2	1	2		3	4	5
	3		3		4	4	5
	1	3	3		3	5	5
	3	2	3		4	3	5
	2	4	3	5	5		5
	2	3	3		5		5

Fig. 5. Ele2 (50 000 fitness evaluations). Examples of the most accurate FMs of one run using the same data partition. (Left) $\alpha = 0.8$, $\beta = 0.05$, $\gamma = 0.25$, $\text{MSE}_{\text{trn}} = 13277$, $\text{MSE}_{\text{tst}} = 12884$, $Q_{\text{Int}} = 0.27$, $Q_{\text{Mid}} = 0.23$, and $Q_{\text{Ext}} = 0.00$. (Right) $\alpha = 0.6$, $\beta = 0.4$, $\gamma = 0.1$, $\text{MSE}_{\text{trn}} = 18272$, $\text{MSE}_{\text{tst}} = 19439$, $Q_{\text{Int}} = 0.09$, $Q_{\text{Mid}} = 0.10$, and $Q_{\text{Ext}} = 0.00$.

0.5; 2) in the center of an MF, all other MFs receive membership value 0; and 3) at the extreme points χ_{low} and χ_{high} of UoD, one MF receives membership value 1. Three quality indexes are therefore computed for each fuzzy partition: 1) Q_{Int}: the maximum absolute difference from the desired intersection membership value 0.5; 2) Q_{Mid}: the maximum membership value of an MF in the center of another MF; and 3) Q_{Ext}: the maximum absolute difference from the desired membership value 1 at the extreme points of UoD. For a strong fuzzy partition, $Q_{\text{Int}} = Q_{\text{Mid}} = Q_{\text{Ext}} = 0$. One must, however, note that even a strong fuzzy partition can be poorly transparent, for example, when some of the MFs are very close to each other. These quality indexes do not take into account this kind of transparency aspects.

Table X compares the averaged quality-index values of the most accurate FMs for different settings of α, β, and γ. Moreover, the average number N_{D} and standard deviation $\sigma_{N_{\text{D}}}$ of distinct FMs on a Pareto front are shown. It is clearly seen that with the settings $\alpha = 0.6$, $\beta = 0.4$, and $\gamma = 0.1$, more transparent fuzzy partitions are obtained (i.e., the quality-index values are lower). The average length of Pareto fronts is, however, not clearly affected by the settings, but depends on the characteristics of each problem. As the number of rule conditions is one of the two fitness objectives, the Pareto fronts tend to be longer if the number of rule conditions in initial FM is high (see Tables IV–VIII).

Figs. 5 and 6 show examples of the most accurate FMs for Ele2 and Mortgage problems with different settings of α, β,



Rule base: 11 rules and 23 rule conditions.

X1	X6	X12	Output
1	1		1
1			1
1	1	1	1
2	1		1
2			2
2	1	1	2
1	2		2
2	2	2	2
	2	2	2
2	2		3
3	2		3

Rule base: 11 rules and 25 rule conditions.

X1	X6	X9	X12	Output
2	2	2	1	1
2	1			1
1	1			1
1		1	2	1
	1		1	2
1	2		1	2
1			1	2
1	2			3
2			2	3
3				3
2	2			3

Fig. 6. Mortgage: Examples of the most accurate FMs of one run using the same data partition. (Left) $\alpha = 0.8$, $\beta = 0.05$, $\gamma = 0.25$, $\text{MSE}_{\text{trn}} = 0.028$, $\text{MSE}_{\text{tst}} = 0.072$, $Q_{\text{Int}} = 0.35$, $Q_{\text{Mid}} = 0.11$, and $Q_{\text{Ext}} = 0.00$. (Right) $\alpha = 0.6$, $\beta = 0.4$, $\gamma = 0.1$, $\text{MSE}_{\text{trn}} = 0.036$, $\text{MSE}_{\text{tst}} = 0.090$, $Q_{\text{Int}} = 0.09$, $Q_{\text{Mid}} = 0.10$, and $Q_{\text{Ext}} = 0.00$.

and γ. It is seen that the fuzzy partitions are more transparent when $\alpha = 0.6$, $\beta = 0.4$, and $\gamma = 0.1$. One may notice that our approach performs input-variable selection, rule learning, granularity learning, and MF-parameters tuning. For example, it can be seen that one of the input variables for Mortgage problem is partitioned with three MFs, whereas the others are partitioned with two MFs. Moreover, these example FMs for Mortgage problem use only three or four input variables, even though the problem has 15 input variables.

VII. Conclusion

A dynamically constrained multiobjective GFS to learn the granularities of fuzzy partitions, tuning the MFs, and learning the fuzzy rules was proposed. It uses dynamic constraints, which enable application of three-parameter MFs tuning to improve the accuracy without deteriorating the transparency of fuzzy partitions. A new initialization method was also proposed. It combines the benefits of WM and DT algorithms, and reduces the number of rules, rule conditions, and input variables, while preserving the transparency of fuzzy partitions. Being a heuristic and suboptimal method, its main purpose is not to obtain very accurate and compact initial FMs, rather, its main purpose is to reduce the search space and, therefore, to ease the further optimization.

Nine benchmark problems having 2 up to 21 input variables were studied, and our multiobjective GFS was tested against 11 recently proposed multiobjective and monoobjective GFSs on six of these nine problems. It was seen that our approach always results into at least comparable accuracy and interpretability with the comparative approaches. Moreover, on some benchmark problems, it clearly outperformed some of the comparative approaches. On the rest three datasets, which have up to 21 input variables, it was tested against a FCM clustering method. It was seen that our FMs are more accurate and interpretable than the FMs obtained by FCM.

Our approach is suitable for both lower and higher dimensional problems. Suitability to higher dimensional problems is aided by the initialization method, which usually reduces the number of input variables. Naturally, if none of the input variables can be removed in initialization phase, the search space will be larger. This poses a challenge to any GFS and requires further research. By our approach, fuzzy partitions with different levels of transparency can be obtained by different settings of α, β, and γ. It was shown that there exists a clear tradeoff between transparency of fuzzy partitions and accuracy. Finally, in this paper, regression problems were considered. However, our approach can be made suitable for classification problems as well [35].

References

[1] H. Ishibuchi and Y. Nojima, "Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning," *Int. J. Approx. Reason.*, vol. 44, no. 1, pp. 4–31, Jan. 2007.

[2] C. Setzkorn and R. C. Paton, "On the use of multi-objective evolutionary algorithms for the induction of fuzzy classification rule systems," *BioSystems*, vol. 81, no. 2, pp. 101–112, 2005.
[3] M. Cococcioni, P. Ducange, B. Lazzerini, and F. Marcelloni, "A Pareto-based multi-objective evolutionary approach to the identification of Mamdani fuzzy systems," *Soft Comput.*, vol. 11, no. 11, pp. 1013–1031, Sep. 2007.
[4] H. Wang, S. Kwong, Y. Jin, W. Wei, and K. Man, "Multi-objective hierarchical genetic algorithm for interpretable fuzzy rule-based knowledge extraction," *Fuzzy Sets Syst.*, vol. 149, no. 1, pp. 149–186, Jan. 2005.
[5] M.-S. Kim, C.-H. Kim, and J.-J. Lee, "Evolving compact and interpretable Takagi–Sugeno fuzzy models with a new encoding scheme," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 5, pp. 1006–1023, Oct. 2006.
[6] R. Alcalá, J. Alcalá-Fdez, M. J. Gacto, and F. Herrera, "Rule base reduction and genetic tuning of fuzzy systems based on the linguistic 3-tuples representation," *Soft Comput.*, vol. 11, no. 5, pp. 401–419, Mar. 2007.
[7] P. Pulkkinen and H. Koivisto, "Fuzzy classifier identification using decision tree and multiobjective evolutionary algorithms," *Int. J. Approx. Reason.*, vol. 48, no. 2, pp. 526–543, Jun. 2008.
[8] M. J. Gacto, R. Alcalá, and F. Herrera, "Adaptation and application of multi-objective evolutionary algorithms for rule reduction and parameter tuning of fuzzy rule-based systems," *Soft Comput.*, vol. 13, no. 5, pp. 419–436, Mar. 2009.
[9] R. Alcalá, J. Alcalá-Fdez, and F. Herrera, "A proposal for the genetic lateral tuning of linguistic fuzzy systems and its interaction with rule selection," *IEEE Trans. Fuzzy Syst.*, vol. 15, no. 4, pp. 616–635, Aug. 2007.
[10] O. Cordón, F. Gomide, F. Herrera, F. Hoffmann, and L. Magdalena, "Ten years of genetic fuzzy systems: Current framework and new trends," *Fuzzy Sets Syst.*, vol. 141, no. 1, pp. 5–31, Jan. 2004.
[11] F. Herrera, "Genetic fuzzy systems: Taxonomy, current research trends and prospects," *Evol. Intell.*, vol. 1, no. 1, pp. 27–46, Mar. 2008.
[12] H. Ishibuchi and T. Yamamoto, "Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining," *Fuzzy Sets Syst.*, vol. 141, no. 1, pp. 59–88, Jan. 2004.
[13] H. Huang, M. Pasquier, and C. Quek, "Optimally evolving irregular-shaped membership function for fuzzy systems," in *Proc. IEEE Congr. Evol. Comput.*, Vancouver, BC, Canada, Jul. 2006, pp. 11078–11085.
[14] M. Setnes, R. Babuŝka, U. Kaymak, and H. R. van N. Lemke, "Similarity measures in fuzzy rule base simplification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 28, no. 3, pp. 376–386, Jun. 1998.
[15] J. Abonyi, J. A. Roubos, and F. Szeifert, "Data-driven generation of compact, accurate, and linguistically-sound fuzzy classifiers based on a decision-tree initialization," *Int. J. Approx. Reason.*, vol. 32, no. 1, pp. 1–21, 2003.
[16] P. Pulkkinen, J. Hytönen, and H. Koivisto, "Developing a bioaerosol detector using hybrid genetic fuzzy systems," *Eng. Appl. Artif. Intell.*, vol. 21, no. 8, pp. 1330–1346, Dec. 2008.
[17] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *Proc. IEEE Congr. Evol. Comput.*, Hong Kong, Jun. 2008, pp. 2424–2431.
[18] E. H. Mamdani and S. Assilian, "An experimental in linguistic synthesis with a fuzzy logic controller," *Int. J. Man–Mach. Stud.*, vol. 7, pp. 1–13, 1975.
[19] L.-X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-22, no. 6, pp. 1414–1427, Nov./Dec. 1992.
[20] P. Pulkkinen and H. Koivisto, "A genetic fuzzy system with inconsistent rule removal and decision tree initialization," in *Applications of Soft Computing, AISC 58*, J. Mehnen, A. Tiwari, M. Köppen, and A. Saad, Eds. Berlin, Germany: Springer-Verlag, 2009, pp. 275–284.
[21] P. Ducange, R. Alcalá, F. Herrera, B. Lazzerini, and F. Marcelloni, "Knowledge base learning of linguistic fuzzy rule-based systems in a multi-objective evolutionary framework," in *HAIS 2008* (Lecture Notes in Artificial Intelligence 5271), E. Corchado and A. Abraham, Eds. Berlin, Germany: Springer-Verlag, 2008, pp. 747–754.
[22] R. Alcalá, P. Ducange, F. Herrera, B. Lazzerini, and F. Marcelloni, "A multiobjective evolutionary approach to concurrently learn rule and data bases of linguistic fuzzy-rule-based systems," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 5, pp. 1106–1122, Oct. 2009.
[23] R. Alcalá, M. J. Gacto, F. Herrera, and J. Alcalá-Fdez, "A multi-objective genetic algorithm for tuning and rule selection to obtain accurate and compact linguistic fuzzy rule-based systems," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 15, no. 5, pp. 539–557, 2007.
[24] M. Antonelli, P. Ducange, B. Lazzerini, and F. Marcelloni, "Learning concurrently partition granularities and rule bases of Mamdani fuzzy systems in a multi-objective evolutionary framework," *Int. J. Approx. Reason.*, vol. 50, no. 7, pp. 1066–1080, Jul. 2009.
[25] M. Antonelli, P. Ducange, B. Lazzerini, and F. Marcelloni, "Learning concurrently granularity, membership function parameters and rules of Mamdani fuzzy rule-based systems," in *Proc. IFSA-EUSFLAT*, Lisbon, Portugal, Jul. 2009, pp. 1033–1038.
[26] J. Casillas, "Embedded genetic learning of highly interpretable fuzzy partitions," in *Proc. IFSA-EUSFLAT*, Lisbon, Portugal, Jul. 2009, pp. 1631–1636.
[27] A. Botta, B. Lazzerini, F. Marcelloni, and D. C. Stefanescu, "Context adaptation of fuzzy systems through a multi-objective evolutionary approach based on a novel interpretability index," *Soft Comput.*, vol. 13, no. 5, pp. 437–449, Mar. 2009.
[28] J. V. de Oliveira, "Semantic constraints for membership function optimization," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 29, no. 1, pp. 128–138, Jan. 1999.
[29] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann, 1993.
[30] B. Zhang and J. Edmunds, "On fuzzy logic controllers," in *Proc. Int. Conf. Control*, Edinburgh, U.K., Mar. 1991, pp. 961–965.
[31] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[32] F. Herrera, M. Lozano, and A. M. Sánchez, "A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study," *Int. J. Intell. Syst.*, vol. 18, pp. 309–338, 2003.
[33] B. L. Welch, "The generalization of "student's" problem when several different population variances are involved," *Biometrika*, vol. 34, no. 1/2, pp. 28–35, Jan. 1947.
[34] G. D. Ruxton, "The unequal variance t-test is an underused alternative to student's t-test and the Mann–Whitney U test," *Behav. Ecol.*, vol. 17, no. 4, pp. 688–690, 2006.
[35] P. Pulkkinen, "A multiobjective genetic fuzzy system for obtaining compact and accurate fuzzy classifiers with transparent fuzzy partitions," in *Proc. 8th Int. Conf. Mach. Learning Appl.*, Miami Beach, FL, Dec. 2009, pp. 89–94.

Pietari Pulkkinen received the M.Sc. degree from Tampere University of Technology, Tampere, Finland, in 2006.

He is currently with the Tampere University of Technology. His research interests include soft-computing methods, especially multiobjective genetic fuzzy systems, and applying them to real-world problems. He has authored or coauthored five international journal articles and four international conference papers. He is a Reviewer for several international journals, such as, the *International Journal of Approximate Reasoning*.

Mr. Pulkkinen is a Reviewer of the IEEE TRANSACTIONS ON FUZZY SYSTEMS.

Hannu Koivisto received the M.Sc. degree in electrical engineering and the Doctor of Technology degree from Tampere University of Technology (TUT), Tampere, Finland, in 1978 and 1995, respectively.

From 2002 to 2007, he was the Head of the Automation and Control Institute, TUT, where he has been a Professor since 1999, and a Professor with the Department of Automation Science and Engineering. His current research interests include applied intelligent data-analysis and neurofuzzy computation, modern-telecommunication-based automation, and system theoretic approach to supply-chain modeling and control. He has authored or coauthored more than 90 publications on these topics. He was a Reviewer of various journal and conference articles.

Prof. Koivisto is a Member of the International Federation of Automatic Control Technical Committee 3.2 (Computational Intelligence in Control).