2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

Genetic Algorithm-Aided Dynamic Fuzzy Rule Interpolation

Nitin Naik, Ren Diao and Qiang Shen
Department of Computer Science
Institute of Mathematics, Physics and Computer Science
Aberystwyth University, UK
Email: {nkn, rrd09, qqs}@aber.ac.uk

***Abstract*—Fuzzy rule interpolation (FRI) is a well established area for reducing the complexity of fuzzy models and for making inference possible in sparse rule-based systems. Regardless of the actual FRI approach employed, the interpolative reasoning process generally produces a large number of interpolated rules, which are then discarded as soon as the required outcomes have been obtained. However, these interpolated rules may contain potentially useful information, e.g., covering regions that were uncovered by the original sparse rule base. Thus, such rules should be exploited in order to develop a dynamic rule base for improving the overall system coverage and efficacy. This paper presents a genetic algorithm based dynamic fuzzy rule interpolation framework, for the purpose of selecting, combining, and promoting informative, frequently used intermediate rules into the existing rule base. Simulations are employed to demonstrate the proposed method, showing better accuracy and robustness than that achievable through conventional FRI that uses just the original sparse rule base.**

I. Introduction

Fuzzy rule interpolation (FRI) [1], [2], [3], [4] is of particular significance for reasoning in the presence of insufficient knowledge. Given a sparse rule base, if an observation has no overlap with antecedent values, no rule can be invoked in classical fuzzy inference, and therefore no consequence can be derived. FRI techniques can support inference in such cases. Most existing FRI systems, regardless of their underlying theory and implementation, tend to process a large amount of interpolated rules, which are generally discarded once the outcomes in response to the given observations are derived. However, interpolated rules may contain potentially useful information, e.g., covering regions that were not covered by the original rule base. If dynamically and intelligently maintained these rules may help greatly improve the overall interpolative coverage and efficacy. This process can be especially beneficial if the frequently appearing observations are of high similarity, where a dynamically created rule may reduce the overheads of interpolation.

A number of techniques [5], [6], [7], [8], [9] exist in the field of dense fuzzy rule-based systems and adaptive fuzzy control, which support dynamic modifications to a given dense rule base. There are also approaches developed for the automatic generation of fuzzy rule-based models [10], [11], [12], using techniques such as neural network [10], genetic algorithm [11], [12], etc. These techniques learn from the data in order to refine a given rule-based system. They can maintain a concurrent, real time rule base for inference and thus, entail more appropriate reasoning results. Unfortunately, such approaches are not directly applicable to sparse rule-based systems due to their assumption of fully covered rules, as well as the underlying computational differences between the use of compositional rule of inference and rule interpolation.

A dynamic fuzzy rule interpolation approach [13] has been introduced to better exploit the interpolation results provided by a given FRI method. However, this approach relies heavily on the use of the standard k-means clustering algorithm. Yet, for many application problems, it is difficult to predict the value of k (the number of clusters) [14]. This paper improves upon the original approach, by employing a genetic algorithm (GA) based clustering technique [15], [16], [17] in place of k-means clustering. In this work, the collection of the interpolated rules is pre-partitioned into hyper-cubes (multi-dimensional blocks), in order to reduce the search complexity of the GA process. The non-empty hyper-cubes are then identified and used as the input into GA. After a certain number of generations, GA identifies a "best" chromosome (cluster arrangement) based on a given fitness function such as the Dunn Index [18]. Here, a chromosome is viewed as a combination of strong and weak clusters, where the weak clusters are merged into the closest strong clusters in order to obtain the final result. In the end, the densest clusters that have accumulated a sufficient number of candidate rules are selected for rule aggregation and promotion.

The remainder of this paper is organised as follows. Section II introduces the theoretical underpinnings of FRI and GAs, and explains the scale and move transformation based approach to FRI (T-FRI) that is used in the current implementation of the GA-based dynamic FRI. Section III illustrates the proposed method. Section IV provides an example, demonstrating the procedures of the proposed approach, and verifies its correctness and accuracy by comparing its outputs to those of conventional FRI. Finally, Section V concludes the paper and suggests a number of future areas of extension.

II. Theoretical Background

A. Transformation-Based Fuzzy Rule Interpolation

This section provides an outline of T-FRI, including both the underlying concepts and the interpolation steps. For simplicity, in this work, fuzzy sets are represented using triangular membership functions. Suppose that an original, sparse rule base $\mathbb{R}$ exists, with rules $R_i \in \mathbb{R}$ and an observation O:

R_i: IF x_1 is $A_{i,1}$, $\cdots$, and x_j is $A_{i,j}$, $\cdots$, and X_N is $A_{i,N}$,

978-1-4799-2072-3/14/$31.00 ©2014 IEEE

THEN y is B_i
O: $A_{\circ,1}$, $\cdots$, $A_{\circ,j}$, $\cdots$, $A_{\circ,N}$

where $A_{i,j} = (a_0, a_1, a_2)$ is the triangular linguistic term for rule R_i, defined on the domain of the antecedent variable x_j, $j \in \{1, \cdots, N\}$, where N is the total number of antecedents, and B_i is the consequent. The observed fuzzy set of variable x_j is denoted by $A_{\circ,j}$. The representative value $rep(A)$ of a triangular fuzzy set is defined as the mean of the X coordinates of the triangle's three points: the left and right extremities of the support a_0, a_2 (with membership values $= 0$), and the normal point a_1 (with membership value $= 1$).

$$rep(A) = (a_0 + a_1 + a_2)/3 \tag{1}$$

The following outlines the core of the T-FRI, more details can be found in [19], [20].

1) Determine the Closest Rules for the Observation: The distance between R_i and O is determined by computing the aggregated distance of all antecedent variables:

$$d(R_i, O) = \sqrt{\sum_{j=1}^{N} d_j^2}, \quad d_j = \frac{d(A_{i,j}, A_{\circ,j})}{range_{x_j}} \tag{2}$$

where $d(A_{i,j}, A_{\circ,j}) = |rep(A_{i,j}) - rep(A_{\circ,j})|$ is the distance between two fuzzy sets in the j^{th} antecedent dimension, with $range_{x_j} = \max x_j - \min x_j$ over the domain of the variable x_j. $d_j \in [0, 1]$ is therefore the normalised result of the otherwise absolute distance measure, so that distances are compatible with each other across different variable domains. The M, $M \geq 2$ rules which have the least distance measurements, with regard to the observed values $A_{\circ,j}$ are then chosen to perform the interpolation in order to obtain the conclusion $B_\circ$.

2) Construct the Intermediate Rule: Let the normalised displacement factor $\omega_{i,j}$, as shown in Eqn. 3, denote the weight of the j^{th} antecedent of the i^{th} rule:

$$\omega_{i,j} = \frac{\omega_{i,j}^{\dagger}}{\sum_{i=1}^{M} \omega_{i,j}^{\dagger}} \tag{3}$$

As $A_{i,j}$ and $A_{\circ,j}$ may totally coincide with each other, the value of $d(A_{i,j}, A_{\circ,j})$ may equal 0. This will make $\omega_{i,j}^{\dagger}$ to be infinite. So, the following non-increasing function is used to represent the weight:

$$\omega_{i,j}^{\dagger} = \exp^{-d(A_{i,j}, A_{\circ,j})} \tag{4}$$

The so-called intermediate fuzzy terms $A_j^{\dagger\dagger}$ are constructed from the antecedents of the M rules.

$$A_j^{\dagger\dagger} = \sum_{i=1}^{M} \omega_{i,j} A_{i,j} \tag{5}$$

These are then shifted to $A_j^{\dagger}$ such that they have the same representative values as those of $A_{\circ,j}$:

$$A_j^{\dagger} = A_j^{\dagger\dagger} + \delta_j range_{x_j} \tag{6}$$

where δ_j is the bias between $A_{\circ,j}$ and ${A_j}^{\dagger}$ on the j^{th} variable domain:

$$\delta_j = \frac{rep(A_{\circ,j}) - rep({A_j}^{\dagger})}{range_{x_j}} \tag{7}$$

Similar to Eqn. 6, the shifted intermediate consequence $B^{\dagger}$ can be computed, with the parameters ω_{B_i} and δ_B being aggregated from the corresponding values of ${A_j}^{\dagger}$, such that:

$$\omega_{B_i} = \frac{1}{N} \sum_{j=1}^{N} \omega_{i,j}, \ \delta_B = \frac{1}{N} \sum_{j=1}^{N} \delta_j \tag{8}$$

3) Scale Transformation: Let $A_j^{\dagger\dagger} = (a_0{}^{\dagger\dagger}, a_1{}^{\dagger\dagger}, a_2{}^{\dagger\dagger})$ denote the fuzzy set generated by the scale transformation in the j^{th} antecedent dimension. By using the scale rate s_j, the current support of $A_j^{\dagger}$, $(a_0{}^{\dagger}, a_2{}^{\dagger})$ is transformed into a new support $(a_0{}^{\dagger\dagger}, a_2{}^{\dagger\dagger})$, such that $a_2{}^{\dagger\dagger} - a_0{}^{\dagger\dagger} = s_j \times (a_2^{\dagger} - a_0^{\dagger})$.

$$\begin{cases} a_0{}^{\dagger\dagger} = \frac{a_0{}^{\dagger}(1+2s_j) + a_1{}^{\dagger}(1-s_j) + a_2{}^{\dagger}(1-s_j)}{3} \\ a_1{}^{\dagger\dagger} = \frac{a_0{}^{\dagger}(1-s_j) + a_1{}^{\dagger}(1+2s_j) + a_2{}^{\dagger}(1-s_j)}{3} \\ a_2{}^{\dagger\dagger} = \frac{a_0{}^{\dagger}(1-s_j) + a_1{}^{\dagger}(1-s_j) + a_2{}^{\dagger}(1+2s_j)}{3} \\ s_j = \frac{a_2{}^{\dagger\dagger} - a_0{}^{\dagger\dagger}}{a_2{}^{\dagger} - a_0{}^{\dagger}} \end{cases} \tag{9}$$

From the above and the given observation terms $A_{\circ,j}$, the scaling factor s_B for the consequent is calculated using Eqn. 10:

$$s_B = \frac{\sum_{j=1}^{N} s_j}{N} \tag{10}$$

4) Move Transformation: $A_j^{\dagger\dagger}$ is subsequently moved using the move rate m_j as given in Eqn. 11, so that the final transformed fuzzy set matches the shape of the observed value $A_{\circ,j}$.

$$\begin{cases} m_j = \frac{3(a_0 - a_0{}^{\dagger\dagger})}{a_1{}^{\dagger\dagger} - a_0{}^{\dagger\dagger}}, \ a_0 \geq a_0{}^{\dagger\dagger} \\ m_j = \frac{3(a_0 - a_0{}^{\dagger\dagger})}{a_3{}^{\dagger\dagger} - a_2{}^{\dagger\dagger}}, \ \text{otherwise} \end{cases} \tag{11}$$

From this, the move factor m_B for the consequent is calculated such that:

$$m_B = \frac{\sum_{j=1}^{N} m_j}{N} \tag{12}$$

The final interpolated result $B_\circ$ can now be estimated by applying the scale and move transformation to $B^{\dagger}$, using the parameters s_B, and m_B.

B. Genetic Algorithms

Genetic Algorithms (GAs) are class of stochastic search procedures that are inspired by Darwinian principle of survival of the fittest individuals and natural selection [21]. Their operation is dependent on two important operators: crossover and mutation. The population (the set of chromosomes) is initially generated randomly and their members are then selected for reproductive process with respect to their fitness values. The chromosomes with higher fitness values have better chances to reproduce. The reproductive process is repeated until desired conditions are met, such as a desired fitness level, or a maximum number of generations. The generic procedure of GAs can be summarised as follows [15], [16]:

- *Initialisation:* Generate random population $\mathbb{P}$ of $|\mathbb{P}|$ chromosomes $[X_1, X_2, ..., X_{|\mathbb{P}|}]$, where each chromosome X_i is an order collection of genes $= [x_1^i, \ldots, x_r^i, x_{r+1}^i, \ldots, x_{|X_i|}^i]$.

- *Fitness Calculation:* Evaluate the fitness $f(X_i)$ of each chromosome X_i in the population $\mathbb{P}$, $\forall i \in \{1, ..., |\mathbb{P}|\}$.
- *Chromosome Selection:* Select two parent chromosomes X_p and X_q from a population $\mathbb{P}$ according to their fitness (the better fitness, the bigger chance to be selected).
- *Crossover:* With a crossover rate δ_c, cross over the parents X_p and X_q to form new offsprings ($X_p^{'}$ and $X_q^{'}$). If no crossover was performed, the offsprings are exact copies of their parents.
- *Mutation:* With a mutation rate δ_m, mutate the offsprings ($X_p^{'}$ and $X_q^{'}$) at each locus (position in chromosome).
- *Acceptance:* The new offsprings ($X_p^{'}$ and $X_q^{'}$) then together form the new population $\mathbb{P}_{new}$, and are used for the subsequent generation.
- *Repeat:* If the given condition is not satisfied, repeat the process from the step-*Fitness Calculation*.
- *Termination:* If the termination condition is satisfied, stop, and return the best chromosome X^{best} in the final population.

III. GA-Based Dynamic Fuzzy Rule Interpolation

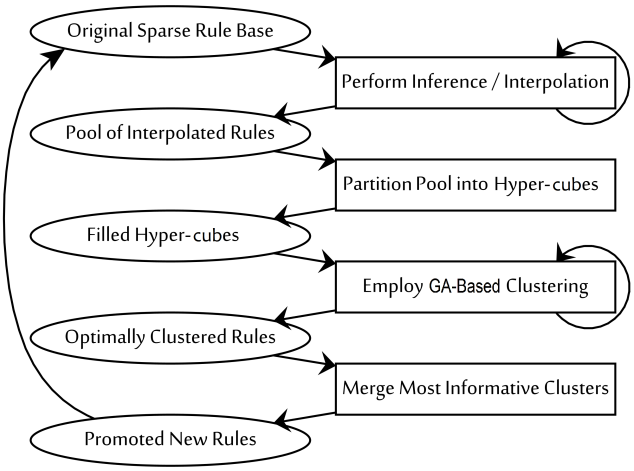

Fig. 1. Procedure of GA-Based Dynamic FRI

This section describes the proposed GA-based dynamic FRI, the overall process of its working is presented in Fig. 1. Generally speaking, there initially exists a set of original (sparse) rules $\mathbb{R}$. While running the FRI system, an interpolation mechanism such as T-FRI gradually fills a pool of interpolated rules $\mathbb{R}'$. The domains of those antecedents appearing in $\mathbb{R}'$ are partitioned into a set of hyper-cubes $\mathbb{H}$. These hyper-cubes are examined to find all non-empty blocks $\mathbb{H}^*$, so that the GA-based clustering algorithm can be employed to find the "best" clustering arrangement leading to a set of strong hyper-cubes $\mathbb{H}^1$ and another of weak hyper-cubes $\mathbb{H}^0$. The strong hyper-cubes are candidate cluster centres in the final clustering outcome. The weak ones are the hyper-cubes that have much less concentration of rules, which are merged into the strong hyper-cubes in order to form the final arrangement. Using GA-based clustering allows the best clusters to be determined without the need to pre-specify the number of clusters k, which is otherwise required by the standard k-means clustering method [14]. After the clustering process, the clusters that have accumulated a sufficient number of interpolated rules (say, more than a certain threshold σ) are selected. Finally, an aggregation process is applied to those selected clusters, in order to construct and promote new rules to become members of the rule base $\mathbb{R}$.

This approach is intuitive and no restriction is imposed over the use of any specific FRI method. The main benefit is to greatly reduce the overheads of interpolating similar, commonly observed values once similar cases have been dealt with, so that only straightforward application of the compositional rule of inference is needed to be carried out.

The following details the key procedures involved in this approach, including antecedent partitioning, interpolated rule clustering and rule promotion. In this work, without losing generality the distance $d(R_p, R_q)$ between two rules R_p ans R_q is defined by:

$$d(R_p, R_q) = \sqrt{\sum_{1}^{N} \frac{(rep(A_{p,i}) - rep(A_{q,i}))^2}{range_{x_i}}} \tag{13}$$

A. Partitioning of Input Space

A grid-based partitioning method is used to identify the uncovered regions of $\mathbb{R}$ and the most frequently interpolated areas that are covered by $\mathbb{R}'$. The antecedent domain is partitioned into a set of hyper-cubes $\mathbb{H}$, by dividing the value ranges of the antecedent variables. A given rule R, which may be an original rule R_k or an interpolated rule $R_k^{'}$ is then assigned to the hyper-cube H_p by checking whether its antecedent values lie within the boundaries of H_p:

$$R \in H_p \text{ if } rep(A_{k,j}) \in [\min H_{p,j}, \max H_{p,j}), j \in \{1, \ldots, N\} \tag{14}$$

where $A_{k,j}$ is the value of the j^{th} antecedent of the rule R.

Ideally, the total number of hyper-cubes and their sizes should be dynamically adjusted according to the current state of the (increasing less) sparse rule base. However, for simplicity, the pre-determined partitions are considered in the current implementation, where the input dimensions are evenly divided into η intervals. The total number of hyper-cubes $|\mathbb{H}|$ is therefore η^N. Whilst all the hyper-cubes are checked, only the non-empty hyper-cubes $\mathbb{H}^*$ are to be used for the later clustering process.

$$\mathbb{H}^* \subseteq \mathbb{H}, \forall H \in \mathbb{H}^*, |H| \neq 0 \tag{15}$$

B. Clustering of Interpolated Rules

A modified genetic algorithm as given in Alg. 1 is used for clustering, which groups similar interpolated rules $R' \in H$ together, forming the clusters. In this work, the customization and implementation of the GA is specified as follows:

Algorithm 1: Genetic Algorithm for Clustering

$\mathbb{P}_{new}$, new population
$X_i^{'}, i^{th}$ chromosome of $\mathbb{P}_{new}$
$X^{best}, X^{best} \in \mathbb{P}, f(X^{best}) = \max\limits_{\forall X \in \mathbb{P}} f(X)$
$f(X_i)$, fitness value of X_i
δ_c , crossover rate
r , random number, where $0 \leq r \leq 1$
k_{max} , maximum number of generations
while $(k_{max} \neq 0)$ **do**
  $\mathbb{P}_{new} = \phi$
  for $(i = 1; i < |\mathbb{P}|; i + 2)$ **do**
    $X_i^{'} = roulettewheelselection(\mathbb{P})$
    $X_{i+1}^{'} = roulettewheelselection(\mathbb{P})$
    if $(r < \delta_c)$ **then**
      $X_i^{'} = crossover(X_i^{'}, X_{i+1}^{'}, true)$
      $X_{i+1}^{'} = crossover(X_i^{'}, X_{i+1}^{'}, false)$
    $X_i^{'} = mutate(X_i^{'})$
    $X_{i+1}^{'} = mutate(X_{i+1}^{'})$
    $\mathbb{P}_{new} = \mathbb{P}_{new} + [X_i^{'}, X_{i+1}^{'}]$
  $k_{max} = k_{max} - 1$
  $\mathbb{P} = \mathbb{P}_{new}$
return X^{best}

1) Chromosome and Population Representation: The length of chromosome $|X|$ is set to the total number of non-empty hyper-cubes $|\mathbb{H}^*|$. Each chromosome is a sequence of $0s$ and $1s$ as shown in Fig. 2. The gene 0 represents weak cluster at that position, and the gene 1 represents a potential presence of a cluster, or a strong cluster. The initial population $\mathbb{P}\ [X_1, X_2, ..., X_{|\mathbb{P}|}]$ is generated randomly, to start the GA search process, where the size of the population $|\mathbb{P}|$ is adjusted in relation to the number of non-empty hyper-cubes $\mathbb{H}^*$. In the GA literature, a population between 20 to 30 chromosomes is typically employed in implementation though a longer population may be utilised [22], [23]. Being a preliminary investigation, a fixed population size is adopted herein.

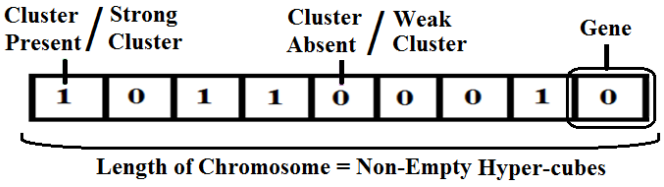

Fig. 2. Chromosome Representation in GA-based Dynamic FRI

2) Fitness Calculation: The fitness function is a problem-dependent parameter in GAs, which decides on the quality of individual chromosomes. In this work, a chromosome represents a potential cluster arrangement, and the Dunn Index (DI) [18] is utilised to assess its quality on the basis of cluster isolation and compactness. A higher value of DI indicates a more favourable result:

$$f(X_i) = \min_{p,q \in \{1,\ldots,i\}, p \neq q} \{ \frac{m_{pq}}{\max_{r \in \{1,\ldots,i\}} s_r} \} \tag{16}$$

where s_r and m_{pq} are the intra-cluster (compactness) and inter-cluster (isolation) distance measurements, respectively:

$$s_r = \sqrt{\sum_{R' \in C_r} \frac{d(R', \mu_r)^2}{|C_r|}}, \ m_{pq} = d(\mu_p, \mu_q) \tag{17}$$

In the above, C_r is the r^{th} cluster, the distance between a given interpolated rule R' and the centroid μ_q of a cluster C_q is calculated in a way similar to Eq. 13 should that:

$$\forall R'_j, R'_k \in \mathbb{R}', d(R'_j, \mu_q) = d(R'_k, \mu_q) \tag{18}$$

where

$$d(R', \mu_q) = \sqrt{\sum_{1}^{N} (rep(A'_i) - \mu_{q,i})^2}, R' \in \mathbb{R}' \tag{19}$$

3) Selection, Crossover and Mutation: Based on the fitness values, parent chromosomes are selected to generate offsprings in the next population using the roulette wheel selection algorithm [24], as summarised in Alg. 2. In roulette wheel selection, each chromosome is assigned a segment of roulette wheel, with a size proportional to its fitness value. Naturally, the bigger the fitness value is, the larger the segment will be.

Algorithm 2: roulettewheelselection($\mathbb{P}$)

$\mathbb{P} = [X_1, ..., X_{|\mathbb{P}|}]$, population
X_i, i^{th} chromosome of population $\mathbb{P}$
$f(X_i)$, fitness value of X_i
r , random number, where $0 \leq r \leq 1$
$threshold = r \times \sum_{i=1}^{|\mathbb{P}|} f(X_i)$
for $\forall i \in \{1, \ldots, |\mathbb{P}|\}$ **do**
  if $(threshold > 0)$ **then**
    $threshold = threshold - f(X_i)$
  else
    return X_i

Crossover and mutation control the generation of offsprings. Crossover process exchanges information between two parent chromosomes while generating the two offsprings. The rate of crossover δ_c is generally high at about $70\% - 95\%$ [22]. The mutation operation tries to avoid premature convergence and explore potential alternative solution regions. However, high mutation rate δ_m has a negative impact on the search ability of the GA and therefore, is set to a very low value [22].

4) Termination: The entire reproductive process is repeated until the maximum number of generations $k_{\max}$ is reached. When the GA terminates, the best chromosome X^{best} of the final population is treated as the search outcome.

5) Cluster/Hyber-cube Merging and Filtering: As previously explained, the "best" chromosome indicates the best clustering strategy determined by the GA. It shows whether a given hyper-cube is to be assigned as a candidate cluster centre (a strong hyper-cube $H^1 \in \mathbb{H}^1$), with which one or more weak hyper-cubes $H^0 \in \mathbb{H}^0$ may be merged subsequently. This arrangement is awarded with the highest fitness value

Algorithm 3: crossover($X_i^{'}$,$X_{i+1}^{'}$, $left$)

$X_i = [x_1^i, \ldots, x_r^i, x_{r+1}^i, \ldots, x_{|X|}^i]$
x_r^i, r^{th} gene of i^{th} chromosome X_i
r , random integer, where $1 \leq r \leq |\mathbb{P}|$
if $(left = true)$ **then**
  return $[x_1^i, \ldots, x_r^i] + [x_{r+1}^{i+1}, \ldots, x_{|X_{i+1}^{'}|}^{i+1}]$
else
  return $[x_1^{i+1}, \ldots, x_r^{i+1}] + [x_{r+1}^i, \ldots, x_{|X_i^{'}|}^i]$

Algorithm 4: mutate($X_i^{'}$)

r , random number, where $0 \leq r \leq 1$
δ_m , mutation rate
x_j^i, j^{th} gene of i^{th} chromosome X_i
for $\forall j \in \{1, \ldots, |X_i^{'}|\}$ **do**
  if $(r < \delta_m)$ **then**
    $x_j^i = \neg x_j^i$
return $X_i^{'}$

(as judged by metrics such as the Dunn Index shown in Eq. 16) over the search process concerned, thereby forming the final clustering outcome. The process which combines the strong and weak hyper-cubes/clusters is outlined in Alg. 5. A selection process is then carried out in order to choose one or more clusters of rules as the candidates for rule promotion. This process may be implemented as picking the clusters which contain more than σ rules, and in case of a tie, the most compact (see Eq. 17) clusters will be selected.

Algorithm 5: merge($\mathbb{H}^1$, $\mathbb{H}^0$)

$H_i^0 \in \mathbb{H}^0$, i^{th} weak hyper-cube
$H_{H_i^0}^1 \in \mathbb{H}^1$, the closest strong hyper-cube to H_i^0
$\mu_{H_i^0}$, centroid of hyper-cube H_i^0
for $\forall H_i^0 \in \mathbb{H}^0, i \in \{1, ..., |\mathbb{H}^0|\}$ **do**
  find $H_{H_i^0}^1 = \arg\min_{H^1 \in \mathbb{H}^1} |\mu_{H^1} - \mu_{H_i^0}|$
  $H_{H_i^0}^1 = H_{H_i^0}^1 \cup H_i^0$
return $\mathbb{H}^1$

C. Dynamic Rule Promotion

Following the clustering and selection process, a group of informative rules $R' \in C_q \subseteq H^*$ are taken for further generalisation in an effort to form a new, aggregated rule, which is pure robust and is hereafter referred to as R^*. This work adopts a weighted combination method, using the cluster centroid μ_q to compute the contributions from the individual candidate rules. Similar to the process of constructing intermediate rules as described in the T-FRI approach [19], [20], a matrix w_{ij} of dimension ($|C_q|$, $N+1$) is used. It indicates the weight of A'_{ij} of an interpolated rule $R'_i \in C_q$ regarding the jth antecedent A_j^* of R^*:

$$w_{i,j} = \frac{1}{d(A'_{i,j}, \mu_{q,j})}, i \in \{1, \ldots, |C_q|\}, j \in \{1, \ldots, N\} \quad (20)$$

and that of B'_i to B^*:

$$w_{i,N+1} = \frac{1}{d(B'_i, \mu_{q,N+1})} \quad (21)$$

The normalised weights can also be obtained:

$$w'_{i,j} = \frac{w_{i,j}}{\sum_{i=1}^{|C_q|} w_{i,j}} \quad (22)$$

From this, the components of the dynamically promoted new rule R^* is constructed as follows:

$$A_j^* = \sum_{i=1}^{|C_q|} w'_{i,j} A'_{i,j},\ j \in \{1, \ldots, N\}, B^* = \sum_{i=1}^{|C_q|} w'_{i,N+1} B'_i$$

This newly promoted R^* is then added to the original (sparse) rule base such that $\mathbb{R} := \mathbb{R} \cup \{R^*\}$, while the rules involved in the aggregation process are removed from the pool of interpolated rules: $\mathbb{R}' := \mathbb{R}' \setminus C_q$. This partitioning-clustering-promotion procedure is applied for all hyper-cubes satisfying $|H_p^*| \geq \sigma$. The entire dynamic FRI process may repeat until the original rule base reaches a state with sufficient coverage of the problem domain. The resultant, complete algorithm for dynamic interpolation supported by a GA is given in Alg. 6.

Algorithm 6: GA-based Dynamic Interpolation($\mathbb{R}$, $\mathbb{R}^{'}$,σ)

$\mathbb{R}$, original sparse rule base
$\mathbb{R}^{'}$, interpolated rule base
R^*, dynamically generated new rule
$\mathbb{H}$, all partitioned hyper-cubes
$\mathbb{H}^* = \mathbb{H}^1 \cup \mathbb{H}^0$, set of non-empty hyper-cubes
$\mathbb{H}^1$, set of strong hyper-cubes
$\mathbb{H}^0$, set of weak hyper-cubes
$\mathbb{C}$, set of clusters
C_i, i^{th} cluster of $\mathbb{C}$
σ , threshold for promoting new rules
$\mathbb{H} = partition(\mathbb{R}^{'})$
$\mathbb{H}^* = \{H | H \in \mathbb{H}, |H| \neq 0\}$
$\mathbb{H}^1 = GA(\mathbb{H}^*)$
$\mathbb{C} = merge(\mathbb{H}^1, \mathbb{H}^0)$
for $\forall C_i \in \mathbb{C}$ **do**
  if $|C_i| > \sigma$ **then**
    $R^* = aggregate(C_i)$
    $\mathbb{R} = \mathbb{R} \cup \{R^*\}$
    $\mathbb{R}' = \mathbb{R}' \setminus C_i$

D. Complexity Analysis

The proposed dynamic approach can be decomposed into three core parts: rule base partitioning, GA-based clustering, and rule promotion. The complexity of the rule base partitioning procedure is shown in Eq. 23, which depends on the number of rules in the interpolated rule base $|\mathbb{R}^{'}|$, the number of rule antecedents N, and the number of partition intervals η:

$$O_{\text{partition}} = O(|\mathbb{R}^{'}| N \eta) \quad (23)$$

The complexity of the GA-based clustering operation given in Eq. 24 is affected by the maximum number of generations

$k_{\max}$, the size of the population $|\mathbb{P}|$, and the complexity of the fitness evaluation O_{fitness}. Additional factors such as the use of genetic operators [22], [24] also play a role, but their impact varies depending on their actual implementations. Thus,

$$O_{\text{ga}} = O(|\mathbb{P}|k_{\max}) \cdot O_{\text{fitness}} \tag{24}$$

where

$$O_{\text{fitness}} = O(|\mathbb{H}| + |\mathbb{R}^{'}|^2 + \frac{|\mathbb{R}^{'}|^2}{|\mathbb{H}|^2} + \mathbb{H}^2) \tag{25}$$

The fitness complexity O_{fitness} combines the cost of chromosome transformation: $O(|\mathbb{H}|)$, the cost of the hyper-cube merging process: $O(|\mathbb{R}^{'}|^2)$, and finally the complexity of the Dunn Index calculation, which is based on both intra- and inter-cluster distance calculations. The complexity of rule promotion depends on the number of clusters $|\mathbb{C}|$ (derived from the output of GA), the number of interpolated rules $|\mathbb{R}^{'}|$, and the number of antecedent dimensions N, such that

$$O_{\text{promotion}} = O(\frac{|\mathbb{R}^{'}|N}{|\mathbb{C}|}) \tag{26}$$

The overall complexity of the proposed GA-based dynamic fuzzy rule interpolation approach is therefore the sum of the three above, i.e., $O_{\text{partition}} + O_{\text{ga}} + O_{\text{promotion}}$.

IV. Simulation-based Evaluation

A numerical example is employed to demonstrate the process of the proposed approach, as well as to evaluate its performance. A function of three crisp input variables, shown in Eq. 27 is chosen to populate a sparse rule base $\mathbb{R}$ of size 100.

$$y = 1 + \sqrt{x_1} + \frac{1}{x_2} + \frac{1}{\sqrt{x_3^3}},\ x_1, x_2, x_3 \in [1, 20] \tag{27}$$

An initial fuzzy rule is generated by fuzzifying the crisp inputs and their associated function output, where a numerical value a is converted to a fuzzy set A with a support length of 1: $A = (a-0.5, a, a+0.5), Rep(A) = a$. This provides a simple non-linear rule base suitable for the purpose of this preliminary investigation. The experiment in this section invokes three different values of η, where the antecedent dimensions are evenly partitioned into $\eta \in \{4, 5, 6\}$ intervals, as a result, $4^3 = 64$, $5^3 = 125$, and $6^3 = 216$, hyper-cubes can be created. The parameters of the GA are set to the following values: crossover rate $\delta_c = 0.7$, mutation rate $\delta_m = 0.05$, population size $|\mathbb{P}| = 20$, and maximum generation $k_{\max} = 100$.

A. GA-Based Clustering Results

The GA-based clustering algorithm is performed over 500 interpolated rules, where 90, 132 and 167 new rules have been promoted for intervals 4, 5, and 6, respectively. The representative values of the consequent of the dynamically promoted rules are recorded. They are then compared to the results of conventional interpolation ($\epsilon_{\%dvi}$), and to the ground truths calculated using the base function ($\epsilon_{\%dvt}$). The differences between conventional interpolation and the ground truths ($e_{\%ivt}$) are also provided. Here the percentage error $\epsilon_{\%} = \epsilon / range_y$ is calculated relative to the range of the consequent variable. Since stochastic elements are present in the initial rule generation, as well as within the clustering procedure, the GA dynamic process is repeated 50 times for each set of the parameter values. Table I shows the averaged value $\epsilon_{\%}$ and the standard deviations of $\epsilon_{\%}$.

TABLE I. GA-Based Clustering Results

	$\eta = 4$			$\eta = 5$			$\eta = 6$		
	$\epsilon_{\%dvi}$	$\epsilon_{\%dvt}$	$\epsilon_{\%ivt}$	$\epsilon_{\%dvi}$	$\epsilon_{\%dvt}$	$\epsilon_{\%ivt}$	$\epsilon_{\%dvi}$	$\epsilon_{\%dvt}$	$\epsilon_{\%ivt}$
AVG	2.68	2.24	2.07	2.38	**1.24**	2.45	3.47	**2.06**	3.74
SD	2.77	2.01	3.35	2.70	**1.25**	2.63	3.03	**1.97**	3.73

According to the simulation results, for $\eta = 5$ and 6, the implemented algorithm promotes more accurate rules, with derived consequent values closer to the ground truth, than those obtainable using conventional interpolation. For this example problem, the best parameter configuration is $\eta = 5$, which produces both accurate and stable rules. For the configuration of $\eta = 6$, the promoted rules are also closer to the ground truth than the outcomes obtained by the conventional T-FRI. These results imply that the rules promoted using intervals $\eta = 5$ and $\eta = 6$, once added to the rule base, will not only avoid the need of future interpolations of similar observations, but also improve the inference accuracy (i.e., the quality of the rule base) overall. Note that large intervals ($\eta = 4$) do not yield good quality rules for this experimental scenario. This is as can be expected, because the size of the individual hyper-cubes are too large to form any meaningful clustering arrangement. The use of the GA also greatly relaxes the needs to specify decent starting conditions, since its stochastic mechanisms are insensitive to the initial states.

B. Rule Base Fulfilment

An extended dynamic rule promotion process is also performed for the same intervals $\eta \in \{4, 5, 6\}$ but with a different number of interpolation rules 250, 500, and 750, respectively. The aim is to observe the level of fulfilment of the sparse regions in the rule base, assuming the proposed dynamic process is in its normal operation (i.e., perform interpolation consecutively). Fig. 3 illustrates graphically the number of fulfilled regions $\mathbb{H}^*$ in $\mathbb{R}$, and the number of rules $|\mathbb{R}|$ in relative to the number of iterations carried out. Here, the same partitioning process is carried out on the original (sparse) rule base, which acts as a preliminary yet compatible way of measuring rule base coverage. The graphs show the values of $|\mathbb{R}|$ and $|\mathbb{H}^*|$ varying throughout the whole process as rules having been promoted may be subsequently removed as new interpolated rules are recorded in the following iterations.

The coverage improves gradually over time as new rules are promoted and added to $\mathbb{R}$. For the case of $\eta = 4$, all original sparse rule-base regions are filled in 38 iterations with 238 original rules as the final size of the rule base. However, when $\eta = 5$ and $\eta = 6$, the rule base can not fully cover the problem space but is closer to fulfilling all regions. In case of the interval $\eta = 5$, 120 hyper-cubes are filled in 125 hyper-cubes through 118 iterations with 525 as the final size of the rule base. Similarly, in case of the interval $\eta = 6$, 193 hyper-cubes are filled in 216 hyper-cubes through 127 iterations with 570 original rules in the final rule base.

Both sets of experiments (accuracy and fulfilment), once analysed together, help to reach the conclusion that the initial, sparse rule base is gradually refined into a denser rule base. The overall accuracy of the resultant rule base is also improved.

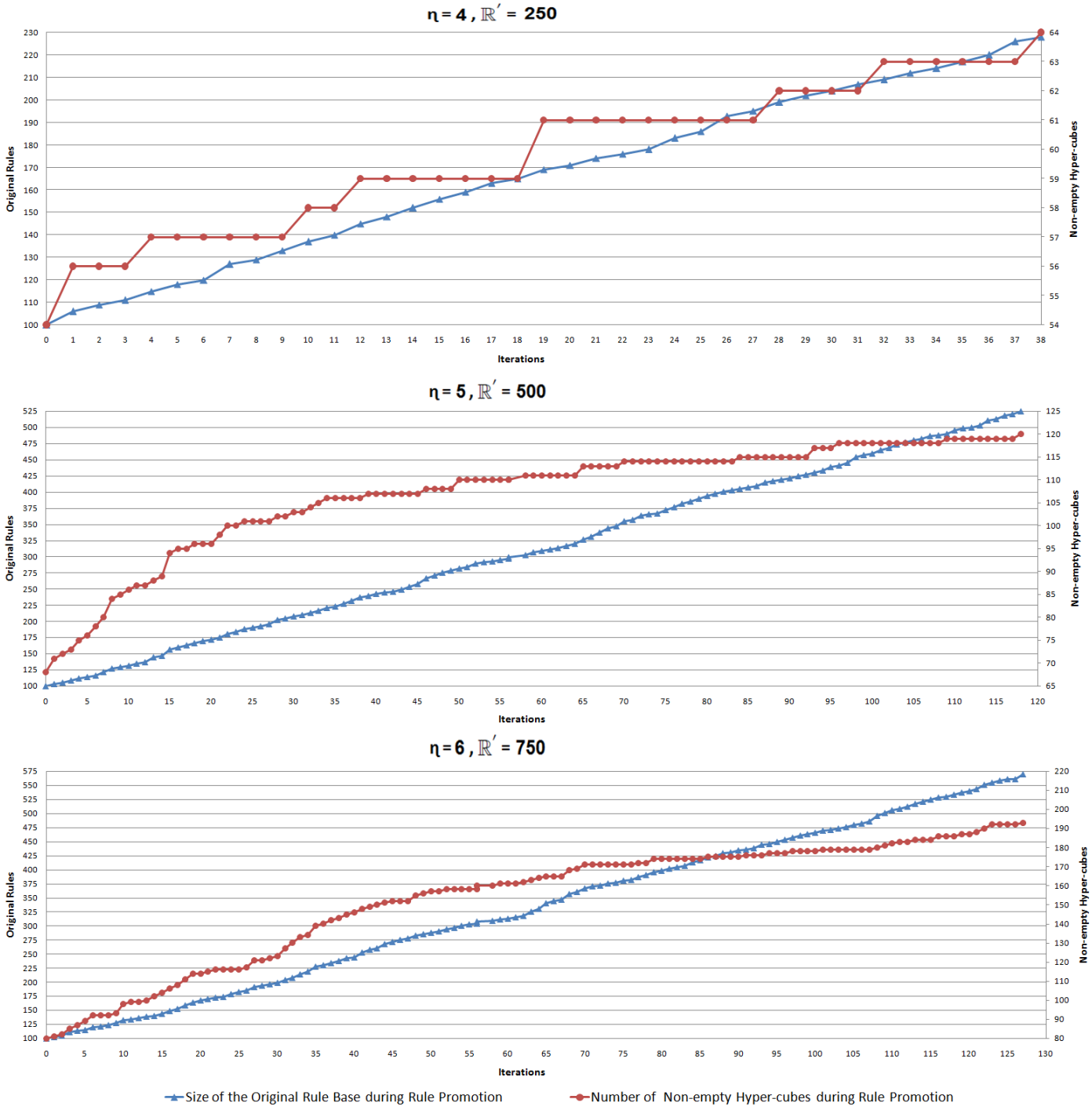

Fig. 3. Iterative GA Based Dynamic FRI Results

V. Conclusion

This paper has presented a GA-based, dynamic FRI approach. While running the dynamic FRI process, the interpolated rules are analysed, selected, aggregated, and promoted when appropriate into the original sparse rule base. According to experimental simulations, the accuracy of using the rule base containing both the original and promoted rules outperform that of using just the original when the conventional T-FRI is employed. It is interesting to note that the resultant system may gradually relax the need of FRI while maintaining an efficient yet accurate reasoning system. This is because the rule base is enriched gradually such that it is no longer sparse and the compositional rule of inference can be applied directly.

An intelligent method for configuring the rule base partitioning remains a vital part of future development. Additionally, the use of state-of-the-art aggregation methods [25], [26] may further improve the quality of the promoted rules. Ideas developed for dynamic rule learning [10], [11], [12] and nature-inspired clustering algorithms [15], [16], [17] may also provide useful insights. While the T-FRI approach is employed in the paper to perform interpolation, the flexibility of the proposed approach may allow the use of more general, similarity-based calculations [27], [28], which would support different choices of similarity measures. Although the current focus of the work is on rule promotion (addition), it is also necessary to examine the scenario of dynamic rule base consolidation including the removal of redundant and inconsistent rules, which is an integral component of a truly intelligent and dynamic approach.

References

[1] D. Dubois and H. Prade, "On fuzzy interpolation," *International Journal Of General System*, vol. 28, no. 2-3, pp. 103–114, 1999.

[2] L. Koczy and K. Hirota, "Approximate reasoning by linear rule interpolation and general approximation," *International Journal of Approximate Reasoning*, vol. 9, no. 3, pp. 197–225, 1993.

[3] ——, "Interpolative reasoning with insufficient evidence in sparse fuzzy rule bases," *Information Sciences*, vol. 71, no. 1-2, pp. 169–201, 1993.

[4] S. Saga, H. Makino, and J. I. Sasaki, "A method for modelling freehand curves- The fuzzy spline interpolation," *Systems and Computers in Japan*, vol. 26, pp. 77–87, 1995.

[5] K. J. Astrom and B. Wittenmark, *Adaptive Control*, 2nd ed. Addison-Wesley, 1995.

[6] B. D. Liu, C. Y. Chen, and J. Y. Tsao, "Design of adaptive fuzzy logic controller based on linguistic-hedge concepts and genetic algorithms," *IEEE Trans. Syst., Man, Cybern. B*, vol. 31, no. 1, pp. 32–53, 2001.

[7] S. Mohan and S. Bhanot, "Comparative study of some adaptive fuzzy algorithms for manipulator control," *International Journal of Computational Intelligence*, vol. 3, pp. 303–311, 2006.

[8] Y. Wang, H. Deng, and Z. Chen, "Adaptive fuzzy logic controller with rule-based changeable universe of discourse for a nonlinear MIMO system," in *Intelligent Systems Design and Applications*, 2005, pp. 8–13.

[9] H. Zhang and Z. Bien, "Adaptive fuzzy control of MIMO nonlinear systems," *Fuzzy Sets and Systems*, vol. 115, no. 1, pp. 191–204, 2000.

[10] S. Wu and M. Joo, "A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks," *IEEE TRANSACTIONS ON FUZZY SYSTEMS*, vol. 9, no. 4, pp. 578–594, 2001.

[11] P. Angelov and R. Buswell, "Automatic generation of fuzzy rule-based models from data by genetic algorithms," *Information Sciences*, vol. 150, no. 1/2, pp. 17–31, 2003.

[12] P. Angelov, "An evolutionary approach to fuzzy rule-based model synthesis using rules indices," *Fuzzy Sets and Systems*, vol. 137, no. 3, pp. 325–338, 2003.

[13] N. Naik, R. Diao, C. Quek, and Q. Shen, "Towards dynamic fuzzy rule interpolation," in *IEEE International Conference on Fuzzy Systems*, 2013, pp. 1–7.

[14] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *5th Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297.

[15] B. B. Chaudhuri and G. Garai, "Grid clustering with genetic algorithm and tabu search process," *Journal Of Pattern Recognition Research*, vol. 4, no. 1, pp. 152–168, 2009.

[16] H. L. Lin, F. W. Yang, and Y. T. Kao, "An efficient ga-based clustering technique," *Tamkang Journal of Science and Engineering*, vol. 8, no. 2, pp. 113–122, 2005.

[17] B. Gabrys and A. Bargiela, "General fuzzy min-max neural network for clustering and classification," *IEEE Trans. on Neural Network*, vol. 11, no. 3, pp. 769–783, 2000.

[18] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, 1973.

[19] Z. Huang and Q. Shen, "Fuzzy interpolative reasoning via scale and move transformations," *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 2, pp. 340–359, 2006.

[20] ——, "Fuzzy interpolation and extrapolation: A practical approach," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 1, pp. 13–28, 2008.

[21] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Univ Michigan Press, 1975.

[22] M. Younes, M. Rahli, and L. A. Koridak, "Economic power dispatch using evolutionary algorithm," *Journal of ELECTRICAL ENGINEERING*, vol. 57, no. 4, pp. 211–217, 2006.

[23] D. Goldberg, *Genetic algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

[24] R. Kumar and Jyotishree, "Economic power dispatch using evolutionary algorithm," *International Journal of Machine Learning and Computing*, vol. 2, no. 4, pp. 365–370, 2006.

[25] T. Boongoen and Q. Shen, "Nearest-neighbor guided evaluation of data reliability and its applications," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 40, no. 6, pp. 1622–1633, 2010.

[26] P. Su, C. Shang, and Q. Shen, "Owa aggregation of fuzzy similarity relations for journal ranking," in *IEEE International Conference on Fuzzy Systems*, 2013, pp. 1–7.

[27] B. Bouchon-Meunier, F. Esteva, L. Godo, M. Rifqi, and S. Sandri, "A principled approach to fuzzy rule based interpolation using similarity relations," in *Conference of the European Society for Fuzzy Logic and Technology*, 2005, pp. 757–763.

[28] F. Esteva, M. Rifqi, B. Bouchon-Meunier, and M. Detyniecki, "Similarity-based fuzzy interpolation method," in *Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 2004, pp. 1443–1449.