Two New Heuristic Methods Based on Crisp and Fuzzy Partitions for Training Data Reduction

Tri Minh Huynh

***Abstract*—This paper is to introduce two heuristic methods based on crisp and fuzzy partitions for selecting the subset of instances from the training data set in high dimensional problems. This subset is called the representative training data set (RTR). A proposed genetic algorithm (GA) is used to learn a compact fuzzy rule-based system (FRBS) with the instances of RTR. RTR size is rather smaller than the initial training data set, thus time cost for learning FRBS decreases significantly. Therein the number of fuzzy rules is not only reduced but rule lengths are also shorter. The smaller size of the rule base is closely related to the interpretability of the FRBS. As a result, the final FBRS gets a suitable and acceptable balance between interpretability and accuracy.**

***Index Terms*—Crisp partition, fuzzy partition, fuzzy rule set reduction, data reduction techniques, genetic algorithm, interpretability.**

I. Introduction

One of the most important applications of Fuzzy Set Theory is Fuzzy Rule-Based Systems (FRBSs). FRBSs have been applied successfully in many areas, such as classification, diagnosis, signal processing, and decision support [1], [2]. In recent years, there have been many papers referring to the applications of Fuzzy Systems to solve many problems in various domains, such as Agriculture and Biological Science, Biochemistry, Genetics and Molecular Biology, Chemistry, Earth and Planetary Sciences, Environmental Sciences, Material Science, Medicine, Neuroscience, Physics and Astronomy, Social Sciences [3].

Genetic Algorithms (GAs) have been employed as robust tools to search optimal solutions in complex spaces. They usually give effective and efficient solutions to the complicated real-world problems. So a combination of GAs with Fuzzy Logic has been led to the birth of Genetic Fuzzy Rule-Based Systems (GFRBSs). These are hybrid fuzzy systems in which a learning process is based on GAs [4].

An interested research direction of FRBSs is to study FRBSs for high dimensional problems. It deals with big databases having high number of features and/or instances. When the number of instances increases, the size of rule set will grow exponentially. Similarly, when the number of features rises, rule lengths grow correlative in number. In linguistic FRBSs, the interpretability is capable of expressing real-world systems in such a way that humans can understand and use them easily. The interpretability of linguistic FRBSs depends closely not only on the size of fuzzy rule set and rule lengths but also on the aspect that fuzzy rules are purely linguistic [5], [6].

Therefore there are several research directions for data reduction techniques, as follows:

- The first research direction focuses on the compact fuzzy rule base so that it retains fewer and length-shorter rules [7], [8].
- The second research direction focuses on selecting features to reduce the number of columns in a data set, so that only the most relevant variables still remain before or during carrying out an inductive learning FRBS process [9].
- The third research direction focuses on selecting relevant instances for reducing the number of rows in a data set before carrying out a learning FRBS process [10], [11].

Until now, the last direction has only been used for creating decision trees and classification problems but almost not been employed for learning and tuning FRBSs [12], [13].

In the section II, thus we propose a new heuristic method used for selecting a subset of instances from the initial training data set (TR). The subset is called the representative training data set (RTR). Our proposed GA in [1], is used to learn a FRBS with the instances of RTR. It involves learning the parameters one by one while other methods simultaneously learn these. This let us easier to find better parameters. GA is also used to learn adding language hedges in each rule. The size of RTR is rather smaller than the one of TR, thus computational cost decreases. The obtained FRBS has a compact rule base reducing the number of rules. It retains good balance between interpretability and accuracy.

In this paper, we consider a MISO FRBS (Multi-Input, Single Output Fuzzy Rule Based System). Let's assume that the input-output data set used as training data is TR={ $(x^{(k)},y^{(k)}) \mid x^{(k)} \in R^m, y^{(k)} \in R, k=1, 2,.., n$ }, where $x^{(k)} = (x_1^{(k)}, .., x_m^{(k)})$ is the input vector of the kth input pair and $y^{(k)}$ is the corresponding output, m is the dimension of the input vector $x^{(k)}$.

The RB may initially be created from TR by the method proposed by us in [1] or by any other simple methods such as Wang and Mendel's [14]. Cost in time for creating the initial KB is negligible.

Let ℓ be the number of fuzzy rules. Suppose we did have an initial rule set, as following:

$$R_1: if\ x_1\ is\ A_1^{(1)}\ and..and\ x_m\ is\ A_m^{(1)}\ then\ y\ is\ B^{(1)}$$

$$R_2: if\ x_1\ is\ A_1^{(2)}\ and..and\ x_m\ is\ A_m^{(2)}\ then\ y\ is\ B^{(2)} \quad ($$

$$...$$

$$R_\ell: if\ x_1\ is\ A_1^{(\ell)}\ and..and\ x_m\ is\ A_m^{(\ell)}\ then\ y\ is\ B^{(\ell)}$$

Manuscript received October 16, 2011; revised October 29, 2011.
Tri Minh Huynh is with Department of Information Technology Sai Gon University Ho Chi Minh City, Viet Nam (e-mail: tri.hm@cb.sgu.edu.vn).

For each fuzzy rule R_p: If x_1 is $A_1^{(p_1)}$ and… and x_m is $A_m^{(p_m)}$ then y is $B^{(p)}$ and each instance $e_r = (x^{(r)}, y^{(r)}) = (x_1^{(r)}, .. ,x_m^{(r)}, y^{(r)}) \in TR$, the covering value CV_T of rule R_p bases on data e_r is computed as [15]:

$$CV_T(R_p, e_r) = T(\mu_{A_1^{(p_1)}}(x_1^{(r)}),..,\mu_{A_m^{(p_m)}}(x_1^{(r)}), \mu_{B^{(p)}}(y^{(r)}))$$

with T being t-norm function. In this paper, T is the product function.

II. Selecting the Representative Training Data Set (RTR) by Using Crisp Partition

A. *The Crisp Partition of the Training Data Set TR.*

- For each instance $e_r \in TR$, set $D_{er} = \max_{1<=p<=l}(CV_T(R_p, e_r))$, and $I_{e_r} = \min_{1<=k<=l}\{k \mid CV_T(R_k, e_r) = D_{er}\}$. It means for each instance $e_r \in TR$, I_{e_r} is the smallest index of the rules which have the covering value CV_T based on e_r, equal to D_{er}. Therefore, each $e_r \in TR$ is determined by only one rule $R_{I_{e_r}}$.

  From previous comment, we can deduce that each rule R_k corresponds to a subset of some instances e_r that has $I_{e_r} = k$. In other words, each rule R_k corresponds to an instance subset of TR, denoted by L_k where $k \in \{1,2,.., \ell\}$.
- Set $L = \{ L_k \mid k \in \{1,2,..\ell\} \}$

  It is easy to see that L forms a crisp partition of the training data set TR, ie:

  1) $L_i \cap L_j = \phi, \ \forall i \neq j, \ i, j \in \{1,2,.., \ell\}$
  2) $\bigcup_{i=1}^{\ell} L_i = TR$

  Consequently L defines an equivalence relation ~ over TR as following: $\forall e_r, e_s \in TR, e_r \sim e_s$ if and only if $\exists i \in \{1,2,.., \ell\}, e_r \in L_i$ and $e_s \in L_i$.

B. *Selecting the Representative Training Data Set (RTR).*

Based on the crisp partition L described in the previous subsection A, we will select RTR from the initial TR. RTR is a relatively small subset compared to TR, thus the space complexity is reduced and computational cost is also decreased. Moreover the final result of rule learning with RTR is as good as the one with TR.

First, we need to select an instance ratio of RTR to TR. This ratio is denoted by p with p being a real number between 0 and 1.

Assuming that n, n_i and m are respectively the instance numbers of TR, L_i and RTR. Because L is a crisp partition of TR, $n = \sum_{i=1}^{\ell} n_i$. So let us set $m = \sum_{i=1}^{\ell} \lceil p.n_i \rceil$ with $\lceil . \rceil$ being ceiling function. From the comments above, for each $i \in \{1,2,.., \ell\}$, we need to choose a subset of L_i consisting of $m_i = \lceil p.n_i \rceil$ instances, denoted by RTR_i. Then the relationship between the number of instances of RTR_i and RTR is calculated as follows: $m = \sum_{i=1}^{\ell} m_i$. For each L_i, set $y_{max}^{(i)} = \max\{ y \mid e = (x,y) \in L_i \}$, $y_{min}^{(i)} = \min\{ y \mid e = (x,y) \in L_i \}$.

We will consider the following cases of m_i:

1) If m_i=1 then select $e = (x, y) \in L_i$ with y closest to $(y_{max}^{(i)} + y_{min}^{(i)})/2$ to put in RTR_i.
2) If m_i=2 then select $e_u^{'} = (x_u, y_u) \in L_i$ with $y_u = y_{max}^{(i)}$, and $e_v^{'} = (x_v, y_v) \in L_i$ with $y_v = y_{min}^{(i)}$ to put in RTR_i.
3) If m_i >=3 then calculate $\varepsilon_i = (y_{max}^{(i)} - y_{min}^{(i)})/(2.(m_i - 2))$ and select $e_u^{'} = (x_u, y_u) \in L_i$ with $y_u = y_{max}^{(i)}$, $e_v^{'} = (x_v, y_v) \in L_i$ with $y_v = y_{min}^{(i)}$ to put in RTR_i. Finally, we need to choose other m_i - 2 instances $e^{'} = (x, y) \in L_i$. To do this, for each $k \in \{1,2,.., m_i - 2\}$, set $y_k = y_{min}^{(i)} + 2.k.\varepsilon_i$. Next we will,in turn, choose one instance $e_k^{'} = (x, y) \in L_i$ to put in the subset RTR_i with $y \in [y_k - \varepsilon_i, y_k + \varepsilon_i]$ so that y is closest to $y_k - \varepsilon_i$.

Finally,we set $RTR = \bigcup_{i=1}^{\ell} RTR_i$.

III. Selecting the Representative Training Data Set (RTR) by Using Fuzzy Partition

A. *The Fuzzy Partition of The Training Data Set TR.*

- For each instance $e_r \in TR$, for each R_k, $CV_T(R_k, e_r)$ is calculated as in section I.

  For each instance $e_r \in TR$, let us set $CV_T(e_r) = \sum_{k=1}^{\ell} CV_T(R_k, e_r)$. Note that $CV_T(e_r) > 0$ because there always exists k such that $CV_T(R_k, e_r) > 0$ according to the method of generating KB in [1].
- Each R_k corresponds to fuzzy set $\tilde{L}_k$ in the training data set TR. The fuzzy set $\tilde{L}_k$ is presented by with $\mu_{\tilde{L}_k}(.)$ being the membership function of $\tilde{L}_k$. $\mu_{\tilde{L}_k} : TR \to [0, 1]$ is defined as follows:

  $$\forall e_r \in TR, \ \mu_{\tilde{L}_k}(e_r) = CV_T(R_k, e_r)/CV_T(e_r)$$
- Set $\tilde{L} = \{\tilde{L}_k \mid k \in \{1,2,..\ell\}\}$.

  When two fuzzy sets $\tilde{L}_i$ and $\tilde{L}_j$ are given, we can obtain the union $\tilde{L}_i \cup \tilde{L}_j$ and the intersection $\tilde{L}_i \cap \tilde{L}_j$. The set $\tilde{L}_i \cup \tilde{L}_j$ and $\tilde{L}_i \cap \tilde{L}_j$ are defined by membership functions [26]

  $$\mu_{\tilde{L}_i \cap \tilde{L}_j}(e_r) = \min(\mu_{\tilde{L}_i}(e_r), \mu_{\tilde{L}_j}(e_r))$$

  $$\mu_{\tilde{L}_i \cup \tilde{L}_j}(e_r) = \max(\mu_{\tilde{L}_i}(e_r), \mu_{\tilde{L}_j}(e_r))$$

  For each fuzzy set $\tilde{L}_k$, the support of $\tilde{L}_k$, $supp(\tilde{L}_k)$, is defined as [26]

  $$supp(\tilde{L}_k) = \{ e \in TR \mid \mu_{\tilde{L}_k}(e) > 0 \}$$

  It is easy to see that $\tilde{L} = \{\tilde{L}_k \mid k \in \{1,2,..\ell\}\}$ a fuzzy partition of the training set TR because it satisfies the following conditions such as [26]:

1) $\tilde{L}_i \neq \varnothing \quad ,\tilde{L}_i \neq TR, \forall i = 1,..,\ell$

2) $\bigcup_{k=1}^{\ell} supp(\tilde{L}_k) = TR$

3) $\sum_{k=1}^{\ell} \mu_{\tilde{L}_k}(e_r) = 1, \forall e_r \in TR$

- For each instance $e_r \in TR$, set $F_{er} = \max_{1<=k<=l}(\mu_{\tilde{L}_k}(e_r))$, $J_{e_r} = \min_{1<=k<=l}\left\{k \mid \mu_{\tilde{L}_k}(e_r) = F_{er}\right\}$. It means for each instance $e_r \in TR$, J_{e_r} is the smallest index of the rules which have $\mu_{\tilde{L}_k}(e_r)$, equal to F_{er}. Therefore, each $e_r \in TR$ is determined by only one rule $R_{J_{e_r}}$.

  In other words, each rule R_k corresponds to an instance subset of TR, denoted by H_k where $k \in \{1,2,..,\ell\}$. H_k is all crisp set.
- Set $H = \{H_k \mid k \in \{1,2,..\ell\}\}$

It is easy to see that H forms a crisp partition of the training data set TR, i.e.

1) $H_i \cap H_j = \phi, \ \forall i \neq j, i, j \in \{1,2,.., \ell\}$

2) $\bigcup_{i=1}^{\ell} H_i = TR$

Consequently H defines an equivalence relation $\approx$ over TR as following: $\forall e_r, e_s \in TR, e_r \approx e_s$ if and only if

$$\exists i \in \{1,2,.., \ell\}, e_r \in H_i \text{ and } e_s \in H_i.$$

B. Selecting The Representative Training Data Set (RTR).

Based on the crisp partition H described in the previous subsection A, section III, we will select RTR from the initial TR. RTR is a relatively small subset compared to TR, thus the space complexity is reduced and computational cost is also decreased. Moreover the final result of rule learning with RTR is as good as the one with TR.

First, we need to select an instance ratio of RTR to TR. This ratio is denoted by q with q being a real number between 0 and 1.

Assuming that n, h_i and m are respectively the instance numbers of TR, H_i and RTR. Because H is a crisp partition of TR, $n = \sum_{i=1}^{\ell} h_i$. So let us set $m = \sum_{i=1}^{\ell} \lceil q.h_i \rceil$ with $\lceil . \rceil$ being ceiling function. From the comments above, for each $i \in \{1,2,.., \ell\}$, we need to choose a subset of H_i consisting of $m_i = \lceil q.h_i \rceil$ instances, denoted by RTR_i. Then the relationship between the number of instances of RTR_i and RTR is calculated as follows $m = \sum_{i=1}^{\ell} m_i$. For each H_i, set $y_{max}^{(i)} = \max\{y \mid e = (x, y) \in H_i\}, y_{min}^{(i)} = \min\{y \mid e = (x, y) \in H_i\}$. We will consider the following cases of m_i:

In case $m_i = 1$ or $m_i = 2$, select e', e'_u and $e'_v \in H_i$ to be included in RTR_i as in subsection B, section II.

If $m_i >= 3$ then calculate $\varepsilon_i = (y_{max}^{(i)} - y_{min}^{(i)})/(2.(m_i-2))$ and select $e'_u = (x_u, y_u) \in H_i$ with $y_u = y_{max}^{(i)}$, $e'_v = (x_v, y_v) \in L_i$ with $y_v = y_{min}^{(i)}$ to put in RTR_i.

Finally, we need to choose other $m_i - 2$ instances $e' = (x, y) \in H_i$. To do this, for each $k \in \{1,2,.., m_i - 2\}$, set $y_k = y_{min}^{(i)} + (2.k - 1).\varepsilon_i$. Next we will, in turn, choose one instance $e'_k = (x, y) \in H_i$ with $y \in [y_k - \varepsilon_i, y_k + \varepsilon_i]$ so that *y is closest to* y_k to put in the subset RTR_i. Note that y_k is the middle point of the interval $y \in [y_k - \varepsilon_i, y_k + \varepsilon_i]$.

Finally, we set $RTR = \bigcup_{i=1}^{\ell} RTR_i$.

IV. BGenerating the Initial KB

The initial KB will be generated by our method that has been discussed in [1]. The four principal components of the FRBS are a fuzzification interface, a knowledge base (KB), a decision-making logic and a defuzzification interface.

A. BFuzzification Interface

The fuzzification interface performs a mapping that converts crisp values of input variables into fuzzy sets.

B. BKnowledge Base (KB)

The KB consists of two main components that are a data base (DB) and a rule base (RB). The DB is composed of the linguistic term sets and the membership functions specifying their meanings. The RB includes set of fuzzy linguistic IF-THEN rules and joined by "also" operator. That means those rules will be activated simultaneously with the same input data.

In this paper, we assume that the domain interval of the ith input variable x_i, is evenly divided into N_i fuzzy sets labeled as $A_i^{(1)}, A_i^{(2)},\ldots, A_i^{(Ni)}$, for i = 1,2,...,m. Similarly, the domain interval of the output variable y, is evenly divided into N fuzzy sets labeled as $B^{(1)}, B^{(2)},\ldots, B^{(N)}$. Any type of membership functions, such as triangle-shaped, trapezoid-shaped and bell-shaped, can be used to specify fuzzy sets. For each $A_i^{(j)}$, we employ an exponential Gaussian membership function as following:

$$\mu_{A_i^{(j)}}(x) = \exp(-(x - m_i^{(j)})^2/(2\sigma_i^{(j)2})).$$

The graph of a Gaussian function is a characteristic symmetric "bell curve" shape in which is the center of the peak and is the spread of the "bell" determined by the method proposed in [1].

C. Decision Making Logic and Deffuzifying Interface

For each rule R_p in section I and for each input $x=(x_1,..,x_m)$, w_p is the firing strength: *(2)*

$$w_p(x) = \mu_{A_1^{(p)}}(x_1).\mu_{A_2^{(p)}}(x_2)...\mu_{A_m^{(p)}}(x_m)$$

The output fuzzy set is computed by $w_p.\mu_{B^{(p)}}(y)$.

Defuzzifying the output fuzzy sets by using the simulated center-of-area method of Lin and Lee [15]:

$$\hat{y} = \frac{\sum_{p=1}^{l} w_p . m^{(p)} . \sigma^{(p)}}{\sum_{p=1}^{l} w_p . \sigma^{(p)}}$$

where is the output of the system, w_p is given by formula above and $m^{(p)}$, $\sigma^{(p)}$ are the centers and the standard deviations of Gaussian functions $\mu_{B^{(p)}}(.)$.

V. Tuning the Initial KB with RTR

KB consists of two components such as DB and RB. Thus there are several different approaches in tuning the initial KB depending on that DB and RB adjusted separately or both simultaneously [16], [17], [18], [19]. We have proposed a new genetic method for tuning both of DB and RB with training data sets [1]. In this paper, this method will be applied but with RTR described in section II and section III.

A. Tuning KB by Using Linguishtic Hedges

In a fuzzy logic-based system, the information is described linguistically. The linguistic hedges are operators used to modify the shape of membership functions. According to their effects to the meanings of membership functions, linguistic hedge operations can be classified into three categories: concentration, dilation, and contrast intensification [20]. In this paper, we only focus on the concentration-type and the dilation-type hedge operations

The concentration-type hedge operations have a behavior of reinforcement such as "very", "strong", etc. The dilation-type hedge operations have a behavior of weakening such as “more or less,” “relatively”, etc.

The concentration-type hedge operations have general formulas [20]:

$$CON(x) = x^{\alpha} \quad ; \alpha > 1$$

The dilation-type hedge operations have general formulas [20]:

$$DIL(x) = x^{\alpha} \quad ; 0 < \alpha < 1$$

B. BCoding of KB

Similar as in [1] but without the component C, each chromosome is presented as three components P+L+R. The P part encodes the basic parameters of the membership functions. The L part expresses the language hedges added in the antecedent and consequent part of the rules in the initial RB. The R part expresses which rules chosen in the initial RB.

The P part includes pairs of real values $m_i^{(j)}$ and $\sigma_i^{(j)}$ being parameters of the exponential Gaussian membership functions

$$\mu_{A_i^{(j)}}(x) = \exp(-(x - m_i^{(j)})^2 / (2\sigma_i^{(j)^2}).$$

Each parameter $m_i^{(j)}$ or $\sigma_i^{(j)}$ will vary in its variation interval. The variation interval of each parameter is already mentioned in the section II above.

The L part is encoded into an integer string with length $\ell.(m+1)$ where ℓ is rule number, m+1 is the number of input variables and one output variable. $L_{k,i}$ is the gene corresponding to the linguistic hedge that modifies the membership function associated to the linguistic term of *ith* variable in k*th* rule. $L_{k,i}$ can take values in {0,..,9} corresponding to the linguistic hedges as Table I follows[21].The R part is encoded into a ℓ-*bit* string in which ℓ is the number of fuzzy rules in the RB. Value 1 at position i in the sequence means that the i*th* rule is used, otherwise the value 0 at position i means that the i*th* rule is not used.

The method tuning the KB by genetic algorithm is described as follows.

TABLE I: Linguistic Hedges and Corresponding Functions

$L_{k,i}$	**Linguistic hedges**	**Corresponding functions**
0	“absolutely”	$(\mu_{A_i^{(k)}}(x))^4$
1	No hedge used	$\mu_{A_i^{(k)}}(x)$
2	“extremely”	$(\mu_{A_i^{(k)}}(x))^3$
3	“very”	$(\mu_{A_i^{(k)}}(x))^2$
4	“much more”	$(\mu_{A_i^{(k)}}(x))^{1.75}$
5	“more”	$(\mu_{A_i^{(k)}}(x))^{1.5}$
6	“plus”	$(\mu_{A_i^{(k)}}(x))^{1.25}$
7	“minus”	$(\mu_{A_i^{(k)}}(x))^{0.75}$
8	“more or less”	$(\mu_{A_i^{(k)}}(x))^{0.5}$
9	“slightly”	$(\mu_{A_i^{(k)}}(x))^{0.25}$

C. The Genetic Algorithm Components

4) The objective function that needs to be minimized is the following one:

$$MSE = \frac{1}{2.n}\sum_{i=1}^{n}(\hat{y}^{(i)} - y^{(i)})^2$$

where n is size of data set, $\hat{y}^{(i)}$ is the output of the FBRS corresponding with given inputs $x^{(i)}$, and the known desired output $y^{(i)}$.

5) The generation of the initial gene pool consists of two steps:
- A chromosome, representing the initial KB, is included. That means, its genes in the P part receives the values from the initial parameters of membership functions of the system (Section III) and in the L, R, alleles 1 will be used.
- The remaining chromosomes of the population are generated with the P part at random within the variation intervals for each gene. Meanwhile, in the L, R, alleles 1 will be still used.

6) Crossover operator

The crossover operator is applied as follows:

In the P part, the max-min-arithmetical operator is used as the crossover operator:

If $P_u^t = (c_1,.., c_k,.., c_h)$ and $P_v^t = (c'_1,.., c'_k,.., c'_h)$ are the P parts of the chromosome u and v in t th generation. If u and v are selected to mate, the four offspring with the below P part will be generated:

$$P_1^{t+1} = a\ P_u^t + (1-a)P_v^t$$
$$P_2^{t+1} = a\ P_v^t + (1-a)P_u^t$$
$$P_3^{t+1} \text{ with } c_{3,k}^{t+1} = \min\{c_k\ ,\ c_k^{'}\}$$
$$P_4^{t+1} \text{ with } c_{4,k}^{t+1} = \max\{c_k\ ,\ c_k^{'}\}$$

with a ∈[0; 0.5] being a given parameter by the designer. In this paper, a is chosen equal to 0.35. In the L, R parts, the standard two-point crossover is used. Then recombine each part, we choose the two best offspring among the 16 children to replace their parents. Note this is 16 offspring are generated from the combination of four different P parts, two different L parts, two different R parts.

7) Mutation operator

The mutation operators are applied differently on each part of the chromosomes chosen for mutation.

In the P part, the uniform mutation operator is applied. Each selected allele of the genes will be replaced by a randomly generated allele on the variation interval of the gene.

In the L part, if value of selected gene is 1, it is changed to a random value in set {0, 2,.., 9} otherwise, it is changed to 1.

In the R, if value of selected gene is 1, it is changed to 0 otherwise, it is changed to 1.

If an individual is selected to be mutated, a randomly selected gene from each its part will be applied a corresponding mutation operator.

Baker‘s stochastic universal sampling (SUS) [19] together with elitism is considered in the paper. Elitism first copies the best chromosome (or some best chromosomes) to new population so it prevents losing the best found solution.

VI. BEXPERIMENTAL STUDY

The two methods based on crisp partition in section II and on fuzzy partition in section III are respectively called method 1 and method 2. In this section, we present tests performed on RTR created by method 1 and method 2.

Our testing consists of the following three steps:

- Step 1: Using the method mentioned in section IV are called as TMH (as the method has been discussed in [1]) or any other simple methods such as Wang and Medel (WM) in [14] to create the KB from numerical data. This process only takes negligible time, much less time than steps 2 and 3.
- Step 2: Using the method 1 or 2 to select the representative training data set (RTR) from the initial numerical data set.
- Step 3: Simplifying RB and tuning DB with RTR by the GA mentioned in section IV via one of six following models.

The operators of GA can be applied on the R part, the P part, the L part with different ways, it will give different models. In this paper, we will mention six models , due to good results that these models offer, extra reference to [1].

A. Model 1:

Implementing Step 1, then the operators of GA are applied on the R first, then applied only on m, a gene of the P part, finally applied only on σ, a gene of the P part. The two above processes are done sequentially but always go together with the GA operators applied on the L, also known as model TMH+R+ML+SL.

It means simplifying RB before, then tuning individually m, afterward tuning σ individually. After the phase R, the set of rules is optimal with smaller size and shorter-length rules. Throughout the two above ML and SL processes, the antecedent parts of rules in RB are also simultaneously modified by adding the appropriate linguistic hedges (section V). In the phase ML, the parameters m of the membership functions have been tuned, learned. In the phase SL, the parameters σ of the membership functions have been tuned, learned. In both of two phases above, the model structure has been extended by using linguistic modifiers.

B. Model 2:

This model is similar to model 1 but the phase R is executed first, then the phase SL is executed before the phase ML is done, also known as model TMH+R+SL+ML.

C. Model 3:

This model is similar to model 1 but the phase SL is executed first, then the phase ML is done and finally the phase R is done, also known as model TMH+SL+ML+R.

D. Model 4:

This model is similar to model 1 but the phase ML is executed first, then the phase SL is done and finally the phase R is done, also known as model TMH+ML+SL+R.

E. Model 5:

This model is similar to model 1. First implementing Step 1, then the operators of GA are applied on m, a gene of the P part, applied on σ, a gene of the P part and applied on the L. Finally, the operators of GA are applied on the R part. The three above phase M, S, L are done simultaneously, also known as model TMH+MSL+R.

In the phase MSL, the parameters m and σ of membership functions have been tuned, learned and the model structure has been extended by using linguistic modifiers. After the phase R, the set of rules is optimal with smaller size and shorter-length rules.

F. Model 6:

This model is similar to model 1. First implementing Step 1, then the operators of GA are applied on the R part. Finally, the operators of GA are applied on m, a gene of the P part, applied on σ, a gene of the P part and applied on the L. The three above phase M, S, L are done simultaneously, also known as model TMH+R+MSL.

After the phase R, the set of rules is optimal with smaller size and shorter-length rules. In the phase MSL, the parameters m and σ of membership functions have been tuned, learned and the model structure has been extended by using linguistic modifiers.

As result of the six models above, the obtained FRBS gets a fuzzy compact and simple rule base with high accuracy and good generalization capacity.

The parameters were used in the tests such as: a population size of 50 individuals, 0.6 as crosser probability, 0.2 as mutation probability per chromosome used. The results such as number of rules and MSE were calculated by taking average for values of all of the tests.

The tests were done on the PC with Pentium Dual core 2.4 GHz processor, 4 GB RAM, Windows 7.0 operating system and development tool Delphi 7.0.

G. The Experimental Study: Estimation of Electrical Network Maintenance Costs Problem

The objective of the problem is to deal with the estimations of minimum maintenance costs of medium-voltage power lines of towns with the four following variables: total length of all streets in town, the total area of the town, the area occupied by buildings, and energy supply to the town. A sample of 1056 simulated towns has been provided in [22]. The data in Table II, Table III, Table IV includes the average value of #R, MSEtra, MSEtst of the tests and are rounded to integers.

TABLE II: THE RESULTS OF OTHER AUTHORS IN [23], [24], [25] FOR THE ESTIMATION OF ELECTRICAL NETWORK MAINTENANCE COSTS PROBLEM

Models	$\bar{x}$			
	#R	MSE_{tra}	MSE_{tst}	h:m:s
Nozaki [23]	532	26,705	27,710	0:00:00
Thirft [24]	565	31,228	37,579	3:13:25
Liska [25]	625	49,263	56,089	7:13:34

TABLE III: THE RESULTS FOR THE ESTIMATION OF ELECTRICAL NETWORK MAINTENANCE COSTS PROBLEM TRAINED WITH RTR *CREATED BY GRISP PARTITION* - THE RATIO OF RTR TO TR: *P*=0.3

Models	$\bar{x}$				
	#R	MSE_{RTR}	MSE_{tra}	MSE_{tst}	h:m:s
TMH	65	77,765	50,191	50,448	-
Model 1	46	13,122	13,367	14,429	0:23:21
Model 2	46	12,837	12,890	13,244	0:22:12
Model 3	55	11,643	11,613	12,741	0:25:04
Model 4	55	12,516	12,253	13,984	0:24:13
Model 5	54	12,916	13,703	16,637	0:27:19
Model 6	46	13,086	13,412	13,725	0:23:26

In Table III, the phase R, ML and SL of the tests were done an average of 100 generations, the phase MSL was done were done an average of 200 generations.

In Table IV, the phase R of the tests was done an average of 100 generations, the phase ML and SL were done an average of 150 generations, the phase MSL was done were done an average of 300 generations. Therefore time cost in Table IV is greater than time cost in Table III.

TABLE IV: THE RESULTS FOR THE ESTIMATION OF ELECTRICAL NETWORK MAINTENANCE COSTS PROBLEM TRAINED WITH RTR *CREATED BY FUZZY PARTITION* - THE RATIO OF RTR TO TR: *P*=0.3

Models	$\bar{x}$				
	#R	MSE_{RTR}	MSE_{tra}	MSE_{tst}	h:m:s
TMH	65	77,765	50,191	50,448	-
Model 1	46	11,060	11,995	13,936	0:30:42
Model 2	46	11,238	12,839	13,662	0:29:54
Model 3	63	10,604	12,170	13,004	0:34:21
Model 4	61	11,215	12,011	12,228	0:33:35
Model 5	60	11,470	12,599	14,666	0:35:51
Model 6	46	10,160	12,358	14,841	0:31:09

VII. CONCLUSIONS

We propose two new heuristic method used for selecting RTR - a subset of instances- from the initial training data set (TR). One method is based on the crisp partition of TR, meanwhile the other is based on the fuzzy partition of TR. The RTRs created by the two proposed methods, give similar results in experimental studies. A proposed tune is performed on the parameters of the membership functions by a new genetic GA. Tuning is done on the RTRs to compare the cost of time and accuracy. Tuning the parameters sequentially one by one give better results than tuning the parameters simultaneously. Reducing the number of rules first leads to the smallest rule base, however sometimes does not lead to rule base with the best accuracy. Finally, an issue arises as to choose ratio p or q for the RTR to the TR. It needs to be chosen how much to be appropriate for each problem. According to our experience, p or q should be selected in the range from 0.3 to 0.5 are appropriate, depending on each problem and its data training set.

ACKNOWLEDGMENT

We sincerely thank the colleagues, Faculty of Information Technology, Saigon University and in particular the professors of the Computer Science Department, Faculty of Information Technology, HCMC University of Natural Sciences for comments and supports for our research.

REFERENCES

[1] M. H Tri.., "A Method to Buid a Fuzzy Streamlined and Effective Rule-based System", *The 2011 3rd International Conference on Machine Learning and Computing*, vol. 4, pp 319-326, Feb, 2011

[2] H. J. Zimmermann, "Fuzzy Set Theory-and Its Applications. 2nd ed. Norwell", MA: Kluwer, 1991.

[3] F. Herrera, "Genetic Fuzzy Systems: Status, Critical Considerations and Future Directions", *International Journal of Computational Intelligence Research,* vol.1, no.1, pp. 59-67, 2005.

[4] O. Cordón, F. Herrera, F. Hoffmann, L. Magdalena, Genetic Fuzzy Systems Evolutionary Tuning and Learning of Fuzzy Knowledge Bases.World Scientific, 2001.

[5] Guillaume, "Designing fuzzy inference systems from dadt: an interpretability-oriented review." *IEEE Trans, Fuzzy Syst.*, vol. 9, pp 426-443, June, 2001.

[6] M. Sugeno and T. Yasukawa, "A fuzzy-logic-based approach to qualitative modeling" *IEEE Trans, Fuzzy Syst.*, vol. 1, pp 7-31, Feb, 1993.

[7] Kweku-Muata, Osey-Bryson, "Evaluation of decision trees: a multicriteria approach," *Computers and Operations Research*, vol. 31, pp. 1933-1945, 2004.

[8] R. Mikut, J. Jäkel, L. Gröll, "Interpretability issues in data-based learning of fuzzy systems," *Fuzzy Sets and Systems*, vol. 150, pp. 179-197, 2005.

[9] H. Liu and H. Motoda, Feature Selection for Knowledge Discovery and Data Mining. Norwell, MA: Kluwer, 1998.

[10] H. Brighton and C. Mellish, "Advances in instance selection for instance-based learning algorithms", *Data Mining and Knowl. Dis*, vol.6, pp. 153–172, 2002.

[11] H. Liu and H.Motoda, "On issues of instance selection," *Data Mining and Knowl.Dis*, vol. 6, pp. 115–130, 2002.

[12] J.R.Cano, F.Herrera, M.Lozano, "On the combination of evolutionary algorithms and stratified strategies for training set selection in data mining," Applied Soft Computing, 2006, in press.

[13] M. Sebban, R. Nock, J. H. Cahuchat, R. Rakotomalala. "Impact of learning set quality and size on decision tree performance," *Int. J. of Computers, Syst. and Signals*, vol. 1, pp. 85-105, 2000.

[14] L. X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from example", *IEEE Trans, Syst., Man, Cybern*., vol. 22, no. 6, pp 1414-1427, Dec, 1992.

[15] C. T. Lin and C. S. G. Lee, "Neural-network-based fuzzy logic control and decision ystem," *IEEE Trans. Comput*., vol. 40, pp. 1320–1336, Oct. 1991.

[16] M. Setnet and J. A. Roubos, "GA-fuzzy modeling and classification: Complexity aand performance." *IEEE Trans, Fuzzy Syst.*, vol. 8, no. 5, pp 509-522, Oct, 2000.

[17] C. L. Karr, "Genetic algorigthms for fuzzy controlers." *Al Expert*, vol. 6, no.2, pp. 26-33, 1991.

[18] O. Cordón, M. J. Del Jesus, and F. Herrera, "Genetic learning of fuzzy rule-based classification systems cooperating with fuzzy reasoning methods." *Int. J. Intel.Syst*, vol 13, pp. 1025-1053, 1998.

[19] J. Casillas, O. Cordón, M. J. Del Jesus, and F. Herrera, "Genetic tuning of fuzzy rule deep structures preserving interpretability and its interaction with fuzzy rule set reduction." *IEEE Trans, Fuzzy Syst.*, vol. 13, pp 14-29, Feb, 2005.

[20] A. Zadeh, Outline of a new approach to the analysis of complex systems and decision processes, I*EEE Transactions on Systems, Man, and Cybernetics SMC*-3 (1973), 28–44.

[21] B-D Liu, C-Y Chen, J-Y Tsao, Design of Adaptive Fuzzy Logic Controller Based on Linguistic-Hedge Concepts and Genetic Algorithms, IEEE Trans on Sys, Man, and Cybernetics—Part b: Cybernetics, vol. 31, no. 1, Feb, 2001.

[22] O. Cordón, F.Herrera, and L.Sánchez, "Solving electrical distribution problems using hybrid evolutionary data analysis techniques," *Appl. Intell.*, vol. 10, no. 1, pp. 5–24, 1999.

[23] K. Nozaki, H. Ishibuchi, and H. Tanaka, "A simple but powerful heuristic mothod for generating fuzzy rules for numerical data." *Fuzzy Sets Syst*., vol. 86, no. 13, pp 251-270, 1997.

[24] P. Thrift, "Fuzzy logic symthesys with genetic algorigthms." *In Proc, 4 th Int. Conf. Genetic Algorigthms, R. K. Belew, and L. B. Booker, Eds., SanMateo, CA*, 1991, , pp 509-513.

[25] J. Liska and S.S.Melsheimer, "Complete design of Fuzzy logic systems using genetic algorigthms", *In Proc, 3 rd IEEE Int. Conf. Fuzzy Systems, Orlando, FL*, 1994, pp. 1377-1382.

[26] Hyung Lee-Kwang and Keon-Myung Lee, "Fuzzy Hypergraph and Fuzzy Partition", *IEEE Trans, Syst., Man, Cybern*., vol. 25, no. 1, pp 196-201, Jan, 1995.

Tri Minh Huynh received the M.Sc. degree in Mathematics in 1997 from the HCMC University of Pedagogy, Viet Nam, the M.Sc. degree in Computer Science in 2000 from the University of Science, Ho Chi Minh City, Viet Nam. He is currently a lecturer in the Department of Information Technology at Sai Gon University, Ho Chi Minh City, Viet Nam. His current research interests include fuzzy systems, genetic fuzzy systems, neural networks, and their applications.