# Towards Sparse Rule Base Generation for Fuzzy Rule Interpolation

Yao Tan, Jie Li, Martin Wonders, Fei Chao, Hubert P. H. Shum, Longzhi Yang

***Abstract*—Fuzzy inference systems have been successfully applied to many real-world applications. Traditional fuzzy inference systems are only applicable to problems with dense rule bases by which the entire input domain is fully covered, whilst fuzzy rule interpolation (FRI) is also able to work with sparse rule bases that may not cover certain observations. Thanks to their abilities to work with fewer rules, FRI approaches have also been utilised to reduce system complexity by removing those rules which can be approximated by their neighbouring ones for complex fuzzy models. A number of important fuzzy rule base generation approaches have been proposed in the literature, but the majority of these only target dense rule bases for traditional fuzzy inference systems. This paper proposes a novel sparse fuzzy rule base generation method to support FRI. The approach first identifies important rules that cannot be accurately approximated by their neighbouring ones to initialise the rule base. Then the raw rule base is optimised by fine-tuning the membership functions of the fuzzy sets. Experimentation is conducted to demonstrate the working principles of the proposed system, with results comparable to those of traditional methods.**

**Keywords–Sparse rule base generation, fuzzy rule interpolation, fuzzy rule base, fuzzy inference systems.**

## I. Introduction

Fuzzy sets and fuzzy logic theory offer a formal way of handling vague information that arises due to the lack of sharp distinctions or boundaries between pieces of information. With an inherent ability to effectively represent and reason on human natural language, fuzzy logic theory is considered as an advanced methodology in the field of control systems. The most common fuzzy models are rule-based fuzzy inference systems, each of which is composed of mainly two parts: an inference engine and a rule base (or knowledge base). The inference engines have been defined by different inference approaches, such as the Mamdani model [1] and the TSK model [2]. Although the TSK model is able to generate crisp output, the Mamdani model is more intuitive and suitable for dealing with human natural language inputs using max-min operators during the inference. Common to all these classical fuzzy inference systems is that they are only applicable to problems with dense rule bases by which the entire input domain is fully covered.

Fuzzy rule interpolation (FRI), initially proposed in [3], not only addresses this issue, but also helps in complexity reduction for complex fuzzy models. When observations do not overlap with any rule antecedent values, traditional fuzzy inference systems will not be applicable, as no rule can be fired. However, fuzzy rule interpolation can still generate a conclusion through a sparse rule base, thus improving the applicability of fuzzy models. FRI can also be employed to reduce the complexity of fuzzy models by excluding rules that can be approximated by their neighbouring ones. A number of important fuzzy rule interpolation methods have been proposed in the literature, such as [4], [5], [6], [7] and [8], which have been successfully applied to deal with real-world problems, including [9] and [10].

Although a dense fuzzy rule base is not required by FRI, a sparse rule base is still needed. Fuzzy rule base generation has been intensively studied in the literature and is usually implemented in one of two ways: data-driven (extracting rules from data) [11], [12] and knowledge-driven (generating rules from human expert knowledge) [13]. As expert knowledge may not always be available, knowledge-driven methods greatly limit the system modelling process. Data-driven rule base generation was proposed to minimise the involvement of human expertise. The success of data-driven approaches is built upon a large quantity of training data, and these approaches usually only target dense rule bases for traditional fuzzy inference approaches. In order to reduce the complexity of such rule bases, various rule base reduction approaches have been developed [14], [15], [16], [17].

This paper presents an initial investigation of a novel data-driven rule base generation approach for FRI, which is able to directly generate a compact sparse rule base from data. In particular, the proposed approach first partitions the problem domain into a number of sub-regions based on the given training data set and fuzzy partitions the problem domain accordingly, such that each sub-region is represented by a corresponding fuzzy rule. Then, the profile curvature of each sub-region is calculated to represent the extent to which the sub-region deviates from being 'flat' or 'straight'. Given a threshold, those sub-regions which have higher curvature values are then identified, and the corresponding rules will be selected to initialise the rule base. From this, the membership functions of the fuzzy sets involved in the initialised rule base are fine-tuned using a genetic algorithm (GA) optimisation method. The experiment shows that the proposed approach can directly generate a sparse rule base for FRI from a given data set, generating results that are comparable with those of [14].

The rest of the paper is structured as follows. Section II introduces the theoretical underpinnings of fuzzy rule interpolation (FRI), with a focus on the stability of the KH ap-

Yao Tan, Jie Li, Martin Wonders, Hubert P. H. Shum, and Longzhi Yang are with the Department of Computer Science and Digital Technologies, Faculty of Engineering and Environment, Northumbria University, Newcastle upon Tyne, UK (email: {yao.tan, jie2.li, martin.wonders, hubert.shum, longzhi.yang}@northumbria.ac.uk).

Fei Chao is with the Cognitive Science Department, School of Information Science and Engineering, Xiamen University, China (email: fchao@xmu.edu.cn).

proach upon which this work is built. Section III presents the proposed approach. Section IV details the experimentation for demonstration and validation. Section V concludes the paper and suggests probable future developments.

## II. Background

FRI not only makes fuzzy inference possible when only sparse rule bases are available, but also helps in complexity reduction when very complex rule bases are utilised. The current fuzzy rule interpolation approaches can be mainly categorised into two classes (with a few exceptions, such as type II fuzzy interpolation [18], [19]).

The first class of FRI approaches is based on the analogical reasoning mechanism [20] and is referred to as 'analogy-based fuzzy interpolation'. Methods of this type work by first creating an intermediate rule such that its antecedent is as 'close' (given a fuzzy distance metric) to the given observation as possible. Then, a conclusion is derived from the given observation by firing the generated intermediate rule through the analogical reasoning mechanism. That is, the shape differentiation between the resultant fuzzy set and the consequence of the intermediate rule is analogous to the shape differentiation between the observation and the antecedent of the generated intermediate rule. A number of ways to create an intermediate rule and then to infer a conclusion from the given observation by that rule have been developed in the literature, including [4], [5], [21], [22] and [23].

The second type of approach directly interpolates rules whose antecedent variables are identical to those observed. The most typical approach in this class was the very first FRI technique, proposed in [3], referred to as the KH approach, which was developed based on the decomposition and resolution principles [24]. According to these principles, each fuzzy set can be represented by a series of $\alpha$-cuts ($\alpha \in [0, 1]$). Given a certain $\alpha$, the $\alpha$-cut of the consequent fuzzy set is calculated from the $\alpha$-cuts of the observation and all the fuzzy sets involved in the rules used for interpolation. Knowing the $\alpha$-cuts of the consequent fuzzy set for all $\alpha \in [0, 1]$, the consequent fuzzy set can be assembled by applying the resolution principle. Approaches such as [8], [25] and [26] also belong to this group. In particular, the stabilised KH approach extends the original KH approach, which is based on a certain interpolation of a family of distances between fuzzy sets in the rules and in the observation [27]. Unlike the original KH approach, it does not consider the two closest neighbouring rules but takes all the rules and computes the conclusion based on the consequent parts weighted by the distances. This approach is outlined below.

Suppose that a sparse rule base is composed of $n$ rules $R_i, i \in \{1, 2, ..., n\}$ that are represented as follows:

$$R_i : \textbf{IF } x \text{ is } A_i \text{ and } y \text{ is } B_i \textbf{ THEN } z \text{ is } C_i. \quad (1)$$

In the current work, each variable value $A$ is represented as a triangular fuzzy set and conveniently denoted as $(a_1, a_2, a_3)$, where $a_2$ is the core and $(a_1, a_3)$ is the support. Given an observation $(A^*, B^*)$, the result $C^*$ can be calculated by

$$\begin{aligned} min\ C^*_\alpha = (&\frac{\sum_{j=1}^{n} inf\{C_{j\alpha}\}\ \frac{1}{d_L(A^*_\alpha,\ A_{i\alpha})}}{\sum_{j=1}^{n} \frac{1}{d_L(A^*_\alpha,\ A_{i\alpha})}} + \\ &\frac{\sum_{j=1}^{n} inf\{C_{j\alpha}\}\ \frac{1}{d_L(B^*_\alpha,\ B_{i\alpha})}}{\sum_{j=1}^{n} \frac{1}{d_L(B^*_\alpha,\ B_{i\alpha})}})/2, \\ max\ C^*_\alpha = (&\frac{\sum_{j=1}^{n} sup\{C_{j\alpha}\}\ \frac{1}{d_U(A^*_\alpha,\ A_{i\alpha})}}{\sum_{j=1}^{n} \frac{1}{d_U(A^*_\alpha,\ A_{i\alpha})}} + \\ &\frac{\sum_{j=1}^{n} sup\{C_{j\alpha}\}\ \frac{1}{d_U(B^*_\alpha,\ B_{i\alpha})}}{\sum_{j=1}^{n} \frac{1}{d_U(B^*_\alpha,\ B_{i\alpha})}})/2, \end{aligned} \quad (2)$$

where $A_{i\alpha}$ and $A^*_\alpha$ represent the $\alpha$-cut of $A_i$ and $A^*$ respectively; $d_L(A_{i\alpha}, A^*_\alpha)$ and $d_U(A_{i\alpha}, A^*_\alpha)$ represent the lower and upper Euclidean distance between $A^*$ and $A_i$, respectively. The values of $d_L(A_{i\alpha}, A^*_\alpha)$ and $d_U(A_{i\alpha}, A^*_\alpha)$ are illustrated in Fig. 1 and calculated as follows:

$$\begin{aligned} d_L(A_{i\alpha}, A^*_\alpha) &= d(inf\{A_{i\alpha}\}, inf\{A^*_\alpha\}), \\ d_U(A_{i\alpha}, A^*_\alpha) &= d(sup\{A_{i\alpha}\}, sup\{A^*_\alpha\}). \end{aligned} \quad (3)$$

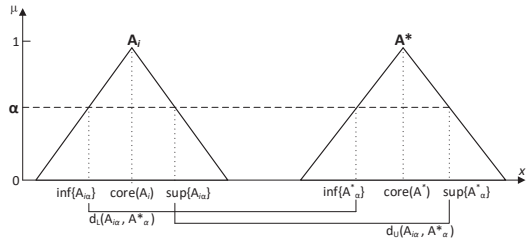


Fig. 1. Illustration of lower and upper distances

From this, the conclusion $C^*$ can be generated by assembling all $C_\alpha$, ($\alpha \in (0, 1]$) based on the resolution principle.

## III. Rule Base Generation

The proposed data-driven rule base generation approach for FRI is presented in this section, which is developed using the profile curvature values of different parts of the data pattern. Given that most of the existing FRI approaches are essentially fuzzy extensions of crisp linear interpolation, the 'flat' or 'straight' parts of a pattern can be easily approximated by its surroundings. Thus, the parts with higher curvature values need to be explicitly represented by fuzzy rules. The proposed system comprises two key parts: rule base initialisation and rule base optimisation. Firstly, the problem domain is partitioned into a number of sub-regions, each represented by a fuzzy rule using the common fuzzy partition practice. Then, the curvature of each sub-region is calculated. Given a curvature threshold $\theta$, if the curvature of a sub-region is greater than $\theta$, the corresponding rule will be selected to initialise the raw rule base. Finally, the general optimisation algorithm GA is employed to optimise the generated raw rule base to achieve the best possible performance.

### A. Rule Base Initialisation

The proposed approach first generates the raw rule base from a training data set. For simplicity, in the current work, only modelling problems with two inputs and one output are considered. Given a pre-processed training data set [28], the progress of rule base initialisation is summarised below.

*1) Problem Domain Determination:* The first step of the modelling is to determine the domain of each variable with respect to the given problem. Assume that the domain of the two inputs and single output are determined by the given data set as follows:

$$\begin{aligned} \textbf{Input: } x, &\quad x \in [\underline{x},\ \bar{x}], \\ y, &\quad y \in [\underline{y},\ \bar{y}], \\ \textbf{Output: } z, &\quad z \in [\underline{z},\ \bar{z}], \end{aligned} \tag{4}$$

where $\underline{x},\ \underline{y},\ \underline{z},\ \bar{x},\ \bar{y},\ and\ \bar{z}$ represent the lower and upper boundary of variables $x,\ y$, and $z$, respectively.

*2) Problem Domain Partition and Fuzzification:* To evaluate the 'flatness' or 'straightness' of different areas of the model, the input domain is equally partitioned into $a \times b$ $(a, b \in \mathbb{N})$ grid areas, where $a$ is the partition number of variable $x$ and $b$ is the partition number of variable $y$. The values of $a$ and $b$ are determined based on the specific problem, which are either arbitrarily given if only subjective training data are provided, or equal to the number of linguistic terms covering the entire variable domain. For instance, suppose that $x = [-2, 2]$ and $y = [0, 3]$. Variable $x$ can be described by four linguistic terms, 'very large', 'large', 'small', and 'very small', and variable $y$ can be described by three linguistic terms, 'tall', 'medium', and 'short'. The input domain therefore can be partitioned into $4 \times 3 = 12$ sub-regions, as shown in Fig. 2.

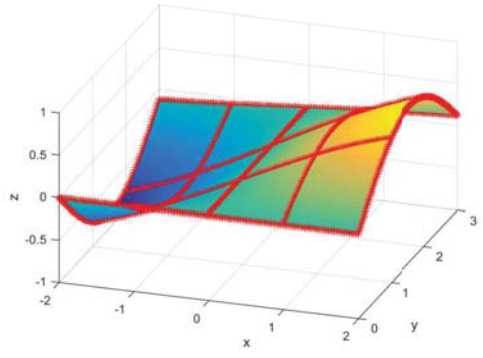


Fig. 2. An exemplar domain partition

Then, the input domain is fuzzy partitioned accordingly such that each region is represented by one fuzzy coordinate or a fuzzy rule. For simplicity, only triangular fuzzy sets are employed in this work, each of which can be precisely represented as $A = (a_1, a_2, a_3)$, where $a_2$ is the core and $(a_1, a_3)$ is the support. In order to preserve the interpretability, during the fuzzy partition process, the central point of a sub-region is guaranteed to be equal to the representative value [5] of the corresponding fuzzy set. In addition, the support of each fuzzy set is equal to twice the distance between the central points of two neighbouring sub-regions. Suppose that variable $x$ is divided into $n$ intervals. The fuzzy partition of variable $x$ is illustrated in Fig. 3, where $A_1, A_2, ..., A_{n-1}$ and $A_n$ $(n \in \mathbb{N})$ represent the partitioned fuzzy sets.

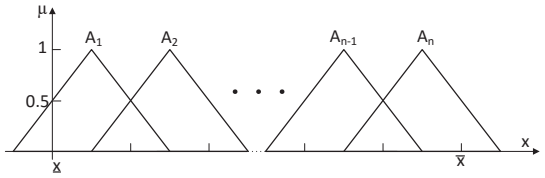


Fig. 3. Fuzzy partition of variable $x$

*3) Curvature Value Calculation:* Curvature values represent the extent to which a geometric object deviates from being 'flat' or 'straight'. By artificially viewing the pattern (hidden in the training data set) to be modelled as a geometry object, curvature values can be used to represent the linearity of the surface of the hidden pattern. The profile curvature approach [29], which represents the steepest downward gradient for a given direction, is employed in the current work to measure the 'flatness' or 'straightness' of each sub-region. The investigation of other types of curvature calculation and the comparison between them remain for further work. The profile curvature is the rate at which a surface slope changes whilst moving in the direction of $grad(f)$. Given a sub-region $f(x, y)$ and a certain direction, then the curvature value in this direction can be calculated by the directional derivative:

$$D_{(\widehat{n})}(F) = \bigtriangledown F \cdot \widehat{n}. \tag{5}$$

The directional derivative refers to the rate at which any given scalar field, F(x,y), is changing as it moves in the direction of some unit vector, $\widehat{n}$, such as $\widehat{n} = -(\bigtriangledown f/S)$ where S is the slope defined as the magnitude of the gradient vector and is a scalar field:

$$S(x, y) = |\bigtriangledown| = \sqrt{f_x^2 + f_y^2}. \tag{6}$$

In order to calculate the overall linearity of a sub-region, eight directions are defined that represent the directions from the centre of the sub-region to the four corners and the central points of the four edges, as illustrated in Fig. 4. Based on these directions, different directional derivative values $(K_{p_i}, i = (1, 2, ..., 8))$ can be calculated and the profile curvature is along the steepest downward gradient. That is, the final profile curvature takes the maximum value of the eight directional curvature values: $K_p = \max(K_{p_1}, K_{p_2}, K_{p_3}, K_{p_4}, K_{p_5}, K_{p_6}, K_{p_7}, K_{p_8})$. Note that the value of profile curvature can be either positive or negative. A negative value indicates that the surface is upwardly convex; a positive value indicates that the surface is upwardly concave; and a zero value indicates that the surface is linear. As the current work is only interested in the linearity of the surface, and convex and concave surfaces make no difference regarding this, the biggest absolute curvature value of the eight directional curvature values for each sub-region is utilised to represent the overall linearity of the concerned sub-region.

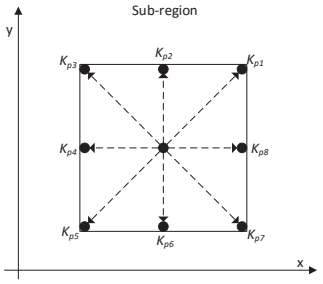


Fig. 4. Directions for profile curvature value approximation

Notice that data are often unevenly distributed in the problem domain. That is, the data in a given data set may be very dense for some parts but very sparse for the others. For each dense part, it is important to generate a representative data point to represent multiple original data points. For each sparse part, a data point may be approximated based on its neighbouring ones. Typical curve fitting approaches, such as linear or non-linear regression, can be used locally within a certain window for both situations. The window size for dense parts should be between a quarter and a full sub-region, whilst the size for sparse parts should be determined by the level of the sparsity. The use of such curve fitting approaches also helps in terms of noise reduction.

*4) Raw Rule Base Generation:* Based on the curvature values of sub-regions, the important rules for FRI can be identified. Given a threshold $\theta$, if the curvature value of a sub-region is greater than the threshold, the corresponding fuzzy rule will be selected to form the initial sparse rule base. This is because, if a sub-region is 'flat', it can be easily approximated by its neighbouring rules using FRI and thus the corresponding rule is not necessary for FRI. If a sub-region has a very high curvature value, multiple rules may be required. In this case, the above procedures can be used recursively in this sub-region to get more detailed rules for the rule base, which is a consideration for future work.

### B. Rule Base Optimisation

The above generated rule base can be optimised in two ways. First, the membership functions can be fine-tuned using a global optimisation approach. Second, the location of the rules may also be fine-tuned by another optimisation algorithm. In this initial investigation, only the first type of optimisation is considered, whilst the latter is a consideration for further work. In particular, GA has been adopted for membership function optimisation with the key steps summarised as follows.

*1) Problem Representation:* Suppose that the initialised raw rule base is composed of $n$ rules ($n \in \mathbb{N}$), which are

$$\begin{aligned} &R_1 : \textbf{IF } x \text{ is } A_1 \text{ and } y \text{ is } B_1, \textbf{ THEN } z \text{ is } C_1, \\ &R_2 : \textbf{IF } x \text{ is } A_2 \text{ and } y \text{ is } B_2, \textbf{ THEN } z \text{ is } C_2, \\ &\quad ..... \\ &R_n : \textbf{IF } x \text{ is } A_n \text{ and } y \text{ is } B_n, \textbf{ THEN } z \text{ is } C_n. \end{aligned} \tag{7}$$

The chromosome is designed in the current work to represent the whole rule base. As discussed above, the optimisation of the locations of rules may be considered for future work, and thus, in this work, the representative values of fuzzy sets are kept fixed and not affected during the membership function optimisation stage. For simplicity, this work also assumes that the membership functions are isosceles triangles. Given the fixed representative value and isosceles shape of a fuzzy set $A$, $A$ can be readily constructed from the support of $A$ (denoted as $supp(A)$). Then, the length of a chromosome $X$ is set to $3n$, as illustrated in Fig. 5, where $n$ represents the number of rules in the rule base.

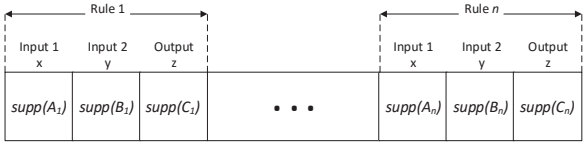


Fig. 5. Chromosome representation in GA

*2) Population Initialisation:* To enable the evolution process, the initial population $\mathbb{P} = \{I_1, I_2, ..., I_{|\mathbb{P}|}\}$ needs to be generated. Typically, the size of population ($|\mathbb{P}|$) is determined based on the given problem, and it may typically contain from several hundreds to several thousands of individuals, each of which is a potential solution to the problem. The first individual $I_1$ in this case represents exactly the raw rule base generated through the approach discussed in Section III, and the other individuals are modified versions of the first one. Intuitively, the evolved individuals should have a larger chance to be similar to the one in the raw rule base than others. Therefore, individuals $\{I_2, I_3..., I_{|\mathbb{P}|}\}$ are randomly generated such that the modified support values of a particular rule antecedent or consequence follow a normal distribution. In particular, suppose that the value at position $i$ of individual $I_1$ is $supp_i$, then the $i^{th}$ position for the rest of the individuals can be generated using a Gaussian distributed random number generation approach such as the classical Box-Muller-Wiener algorithm, with $supp_i$ being the expected value [30].

*3) Fitness Evaluation:* An objective function is used in GA to measure the fitness or quality of individuals. In this initial work, the objective function is defined as the root mean square of the error (RMSE). Given an individual $I_i$, $1 \leqslant i \leqslant |\mathbb{P}|$, the RMSE value regarding this individual can be calculated as follows:

$$RMSE_i = \sqrt{\frac{\sum_{j=1}^{m} (z_j - \hat{z}_j)^2}{m}}, \tag{8}$$

where $m$ is the size of the training data set; $z_j$ is the labeled (defuzzified) output value of the $j^{th}$ training data instance and $\hat{z}_j$ represents the (defuzzified) output that is generated by a particular FRI approach.

In this case, the fittest individuals will have the lowest numerical value of the associated objective function. As the 'roulette wheel' selection method is used in the current work to probabilistically select individuals for reproduction, a fitness function is used to transform the objective function

value into a measure of relative fitness [31], in an effort to prevent premature convergence by limiting the reproductive range so that no individuals generate an excessive number of offspring. The fitness of an individual $I_i$ in the current work is calculated as follows [31]:

$$f(I_i) = 2 - max + \frac{2(max-1)(r_i-1)}{|\mathbb{P}|}, \qquad (9)$$

where $r_i$ is the ranking position of individual $I_i$ in the ordered population $\mathbb{P}$, and $max$ is the bias or selective pressure, towards the fittest individuals in the population.

*4) Selection and Reproduction:* The population $\mathbb{P}$ are ranked based on their fitness values, and then $k$ pairs of individuals or elites are selected using a 'roulette wheel' mechanism to produce the next generation of individuals by two genetic operations: crossover and mutation. Crossover exchanges contiguous sections of the chromosomes, which takes two parent solutions and produces two children solutions from them. Each pair of individuals in the $k$ pairs of selected elites acts as the parents for reproduction. A single crossover point on each pair of selected parents, organism string is selected, and all data after the index point of the two parents are swapped. The resulting individuals (after mutation) will be part of the next generation, denoted as $\mathbb{P}'$.

Mutation is used to maintain genetic diversity from one generation of a population to the next, which simulates biological mutation. Mutation alters one gene value in a chromosome from its initial state, which helps the algorithm to avoid local minima by preventing the population of individuals from becoming too similar to each other. A certain percentage of offspring in $\mathbb{P}'$ are selected to take the mutation operation. In addition, a single or multiple points of a rule can take the mutation procedure. Particularly in the current work, the mutation procedure produces a random support value regarding a particular fuzzy set for the mutation point using another Gaussian random number generation approach.

Suppose that the second generation of population ($\mathbb{P}'$) has been generated. The fitness function, Equation 9, will be employed again to determine the quality of each individual in $\mathbb{P}'$. Finally, the best individuals in $\mathbb{P}'$, which are represented by the smallest fitness values, will be selected and used to replace the worst ranked individuals in $\mathbb{P}$, thus completing one iteration of the GA searching process by generating a new population of solution $\mathbb{P}''$. The entire process of selection and reproduction is shown in Fig. 6.

*5) Termination Conditions:* The reproduction procedure is repeated until the pre-specified maximum number of iterations is reached or the objective value of an individual is less than a predefined threshold. When the GA terminates, the fittest individual in the current population is the optimal solution.

Notice that GA has been used to optimise fuzzy rule bases to support FRI in [32]. In the current work, GA was used for clustering, which is the process of grouping similar interpolated rules to form clusters. These clusters were further used to form new, aggregated rules to update the existing rule base. Therefore, GA was used in the rule base updating stage after a number of inference iterations. In contrast, in this proposed approach, GA is used to optimise the raw rule base before it is used for inference. The introduction of GA to rule base generation indeed provokes high computational complexity, but this is acceptable, as the rule base generation is a one-time process that is carried out offline.

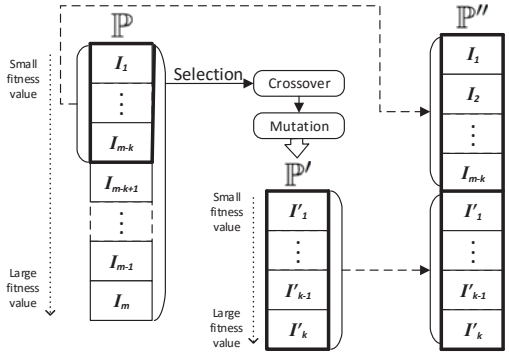


Fig. 6. Selection and reproduction process

## IV. Experimentation

The proposed system was evaluated in this section, by adopting the problem considered in [14]. The problem is to model the non-linear function given below:

$$f(x,y) = \sin\left(\frac{x}{\pi}\right)\sin\left(\frac{y}{\pi}\right).$$

The fuzzy model takes two inputs, $x$ ($x \in [-10, 10]$) and $y$ ($y \in [-10, 10]$), and produces one output $z$ ($z \in [-1, 1]$), as illustrated in Fig. 7. This experimentation is built upon the stabilised KH FRI approach [33], as introduced in Section II.

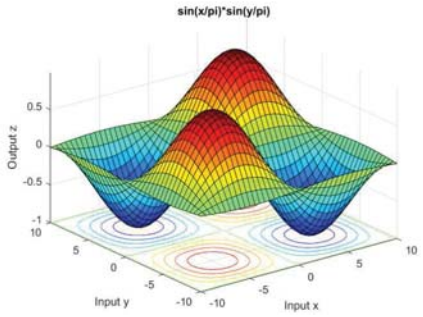


Fig. 7. Surface view of the model

### A. Rule Base Initialisation

The problem domain is equally partitioned into $20 \times 20$ grid areas, which results in a total of 400 sub-regions, as shown in Fig. 8. The input domain of variable $x$ has therefore been divided into 20 equal intervals, with each interval being represented as a fuzzy set, as shown in Fig. 9. This is also the case for variable $y$, thus facilitating the representation of each sub-region by a fuzzy rule.

The degree of flatness or sharpness of each such sub-region can be represented by its curvature value, calculated

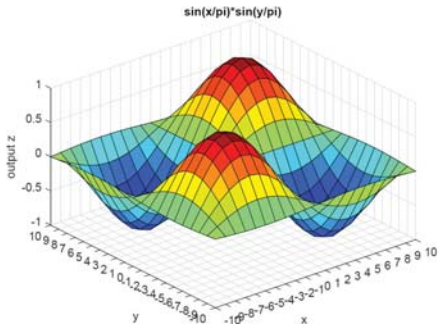


Fig. 8. Problem space partition

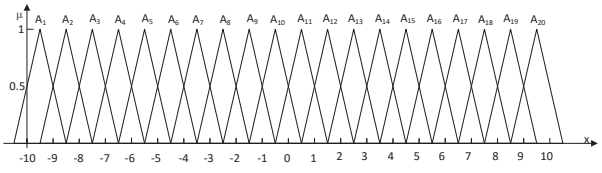


Fig. 9. Fuzzy partition of the domain of input $x$

using Equation 5, with the results listed in Table I. Note that, in this illustrative example, the data are noise-free and compact, thus data pre-processing is not required. However, pre-processing is necessary for most real-world applications. Real-world data are also often unevenly distributed, which may restrict the direct use of Equation 5. In this case, curve fitting approaches, such as linear regression, are then necessary to fill in missing data points for sparse areas and generate representative data points for dense parts, which also provides a means of noise reduction.

A threshold value is defined to identify those 'rugged' areas which indicate important rules that cannot be accurately approximated by their neighbouring ones through linear FRI. The larger is the threshold to be specified, whilst the smaller is the number of rules to be selected. The determination of the threshold value is therefore related to the specific problem to be modelled, which represents the balance between the accuracy performance and the complexity of the fuzzy model. For the illustrated example, if the threshold is set as $\theta = 0.06$, the curvature values of 100 sub-regions are greater than $\theta$, thus 100 rules are then selected to initialise the rule base, as illustrated in Fig. 10. If the threshold is defined as $\theta = 0.08$, there are only 52 sub-regions whose curvature values are higher than the threshold, as shown in Fig. 11. However, if given a threshold of $\theta = 0.09$, there are only 20 sub-regions whose curvature values are higher than the threshold, as illustrated in Fig. 12.

### B. Rule Base Optimisation

The initialised rule base includes all the most important rules that cannot be accurately represented by their neighbouring ones, but these rules are not optimal in terms of their membership functions. After generating the raw rule base in the previous section, the optimisation algorithm GA is employed to fine-tune the membership functions involved in the rules. In this experimentation, the population size

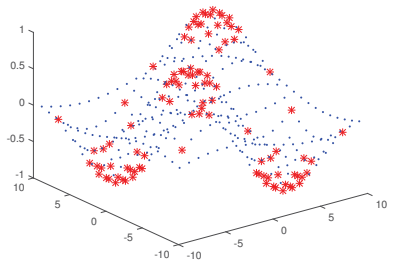

Fig. 10. The sub-regions with curvature values greater than 0.06

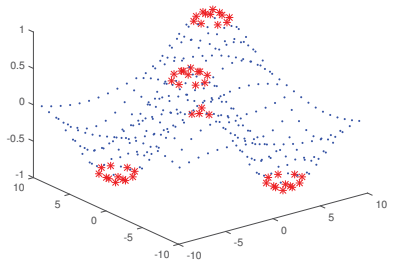

Fig. 11. The sub-regions with curvature values greater than 0.08

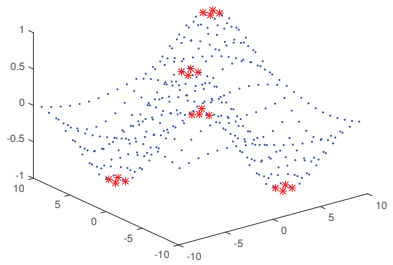

Fig. 12. The sub-regions with curvature values greater than 0.09

was set as 100, with the first individual in the population configured to represent exactly the generated raw rule base. All other individuals were randomly generated using the approach introduced in Section III-B. The $max$ value parameter of the fitness function (Equation 9) was set to 2, the maximum number of generations was set to 1000, and the probabilities of crossover and mutation were set to 0.8 and 0.01, respectively.

The optimisation generally leads to a decrease of the average error by 5% to 20% compared to the error resulting from the employment of the raw rule base. When the rule base contains a relatively large number of rules, a significant improvement in accuracy was recorded, as shown in Fig. 14, where the rule base with 12 rules has the smallest error; conversely, when the rule base consists of fewer rules, only a very small improvement was recorded, as demonstrated in Fig. 13, where rule bases with 8 and 4 rules have smaller errors.

TABLE I. CURVATURE VALUES OF THE SUB-REGIONS

| no. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.0985 | 0.0385 | 0.0100 | -0.0043 | -0.0109 | -0.0113 | -0.0055 | 0.0076 | 0.0333 | 0.0930 | -0.0930 | -0.0333 | -0.0076 | 0.0055 | 0.0113 | 0.0109 | 0.0043 | -0.0100 | -0.0385 | -0.0985 |
| 2 | 0.0385 | 0.0652 | 0.0240 | -0.0167 | -0.0393 | -0.0407 | -0.0208 | 0.0180 | 0.0615 | 0.0491 | -0.0491 | -0.0615 | -0.0180 | 0.0208 | 0.0407 | 0.0393 | 0.0167 | -0.0240 | -0.0652 | -0.0385 |
| 3 | 0.0100 | 0.0240 | 0.0063 | -0.0323 | -0.0639 | -0.0661 | -0.0375 | 0.0019 | 0.0235 | 0.0132 | -0.0132 | -0.0235 | -0.0019 | 0.0375 | 0.0661 | 0.0639 | 0.0323 | -0.0063 | -0.0240 | -0.0100 |
| 4 | -0.0043 | -0.0167 | -0.0323 | -0.0550 | -0.0824 | -0.0849 | -0.0586 | -0.0348 | -0.0185 | -0.0058 | 0.0058 | 0.0185 | 0.0348 | 0.0586 | 0.0849 | 0.0824 | 0.0550 | 0.0323 | 0.0167 | 0.0043 |
| 5 | -0.0109 | -0.0393 | -0.0639 | -0.0824 | -0.0948 | -0.0963 | -0.0843 | -0.0667 | -0.0428 | -0.0148 | 0.0148 | 0.0428 | 0.0667 | 0.0843 | 0.0963 | 0.0948 | 0.0824 | 0.0639 | 0.0393 | 0.0109 |
| 6 | -0.0113 | -0.0407 | -0.0661 | -0.0849 | -0.0963 | -0.0975 | -0.0868 | -0.0690 | -0.0443 | -0.0153 | 0.0153 | 0.0443 | 0.0690 | 0.0868 | 0.0975 | 0.0963 | 0.0849 | 0.0661 | 0.0407 | 0.0113 |
| 7 | -0.0055 | -0.0208 | -0.0375 | -0.0586 | -0.0843 | -0.0868 | -0.0619 | -0.0399 | -0.0228 | -0.0075 | 0.0075 | 0.0228 | 0.0399 | 0.0619 | 0.0868 | 0.0843 | 0.0586 | 0.0375 | 0.0208 | 0.0055 |
| 8 | 0.0076 | 0.0180 | 0.0019 | -0.0348 | -0.0667 | -0.0690 | -0.0399 | -0.0021 | 0.0175 | 0.0100 | -0.0100 | -0.0175 | 0.0021 | 0.0399 | 0.0690 | 0.0667 | 0.0348 | -0.0019 | -0.0180 | -0.0076 |
| 9 | 0.0333 | 0.0615 | 0.0235 | -0.0185 | -0.0428 | -0.0443 | -0.0228 | 0.0175 | 0.0585 | 0.0429 | -0.0429 | -0.0585 | -0.0175 | 0.0228 | 0.0443 | 0.0428 | 0.0185 | -0.0235 | -0.0615 | -0.0333 |
| 10 | 0.0930 | 0.0491 | 0.0132 | -0.0058 | -0.0148 | -0.0153 | -0.0075 | 0.0100 | 0.0429 | 0.0962 | -0.0962 | -0.0429 | -0.0100 | 0.0075 | 0.0153 | 0.0148 | 0.0058 | -0.0132 | -0.0491 | -0.0930 |
| 11 | -0.0930 | -0.0491 | -0.0132 | 0.0058 | 0.0148 | 0.0153 | 0.0075 | -0.0100 | -0.0429 | -0.0962 | 0.0962 | 0.0429 | 0.0100 | -0.0075 | -0.0153 | -0.0148 | -0.0058 | 0.0132 | 0.0491 | 0.0930 |
| 12 | -0.0333 | -0.0615 | -0.0235 | 0.0185 | 0.0428 | 0.0443 | 0.0228 | -0.0175 | -0.0585 | -0.0429 | 0.0429 | 0.0585 | 0.0175 | -0.0228 | -0.0443 | -0.0428 | -0.0185 | 0.0235 | 0.0615 | 0.0333 |
| 13 | -0.0076 | -0.0180 | -0.0019 | 0.0348 | 0.0667 | 0.0690 | 0.0399 | 0.0021 | -0.0175 | -0.0100 | 0.0100 | 0.0175 | -0.0021 | -0.0399 | -0.0690 | -0.0667 | -0.0348 | 0.0019 | 0.0180 | 0.0076 |
| 14 | 0.0055 | 0.0208 | 0.0375 | 0.0586 | 0.0843 | 0.0868 | 0.0619 | 0.0399 | 0.0228 | 0.0075 | -0.0075 | -0.0228 | -0.0399 | -0.0619 | -0.0868 | -0.0843 | -0.0586 | -0.0375 | -0.0208 | -0.0055 |
| 15 | 0.0113 | 0.0407 | 0.0661 | 0.0849 | 0.0963 | 0.0975 | 0.0868 | 0.0690 | 0.0443 | 0.0153 | -0.0153 | -0.0443 | -0.0690 | -0.0868 | -0.0975 | -0.0963 | -0.0849 | -0.0661 | -0.0407 | -0.0113 |
| 16 | 0.0109 | 0.0393 | 0.0639 | 0.0824 | 0.0948 | 0.0963 | 0.0843 | 0.0667 | 0.0428 | 0.0148 | -0.0148 | -0.0428 | -0.0667 | -0.0843 | -0.0963 | -0.0948 | -0.0824 | -0.0639 | -0.0393 | -0.0109 |
| 17 | 0.0043 | 0.0167 | 0.0323 | 0.0550 | 0.0824 | 0.0849 | 0.0586 | 0.0348 | 0.0185 | 0.0058 | -0.0058 | -0.0185 | -0.0348 | -0.0586 | -0.0849 | -0.0824 | -0.0550 | -0.0323 | -0.0167 | -0.0043 |
| 18 | -0.0100 | -0.0240 | -0.0063 | 0.0323 | 0.0639 | 0.0661 | 0.0375 | -0.0019 | -0.0235 | -0.0132 | 0.0132 | 0.0235 | 0.0019 | -0.0375 | -0.0661 | -0.0639 | -0.0323 | 0.0063 | 0.0240 | 0.0100 |
| 19 | -0.0385 | -0.0652 | -0.0240 | 0.0167 | 0.0393 | 0.0407 | 0.0208 | -0.0180 | -0.0615 | -0.0491 | 0.0491 | 0.0615 | 0.0180 | -0.0208 | -0.0407 | -0.0393 | -0.0167 | 0.0240 | 0.0652 | 0.0385 |
| 20 | -0.0985 | -0.0385 | -0.0100 | 0.0043 | 0.0109 | 0.0113 | 0.0055 | -0.0076 | -0.0333 | -0.0930 | 0.0930 | 0.0333 | 0.0076 | -0.0055 | -0.0113 | -0.0109 | -0.0043 | 0.0100 | 0.0385 | 0.0985 |

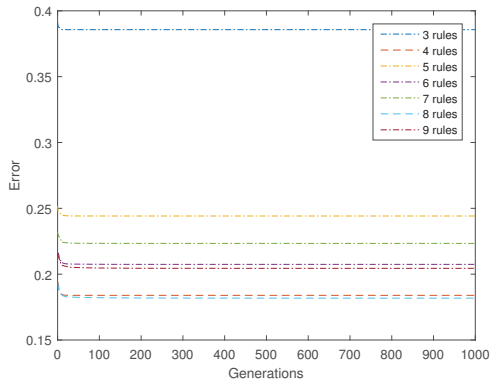


Fig. 13. The average error values decrease over time during membership function optimisation for rule bases with 3 to 9 rules

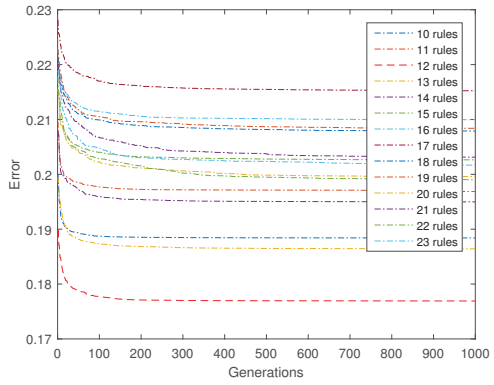


Fig. 14. The average error values decrease over time during membership function optimisation for rule bases with 10 to 23 rules

### C. Results and Discussion

To facilitate the comparison between the proposed approach and the approaches presented in [14], the sum of errors for 36 random testing points produced by different approaches with different sizes of rule bases are summarised in Fig. 15. The black and purple lines represent the results generated by the approach proposed in [14] with nearest neighbour interpolation and piecewise polynomial cubic spline interpolation approaches, respectively, whilst the blue line represents the results generated by the proposed approach. From this figure, it is clear that the employment of rule bases with fewer rules generally leads to large errors, whilst the employment of rule bases with more rules generally results in small errors. However, it should be noted that this is not always the case. For instance, the summed error produced by the rule base with 12 rules is smaller than that led by 17 rules. From preliminary investigation, this is partly caused by the fixed locations of the rules and partly due to the employed FRI approach. Further investigation on this is left for future work.

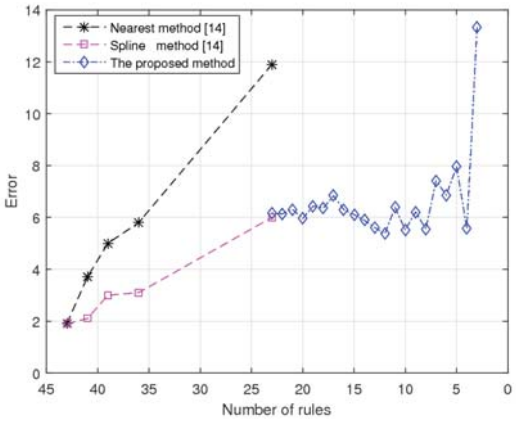


Fig. 15. Errors produced by rule bases with different numbers of rules

This experimentation is based on the KH-stabilised FRI approach, and thus the experimental results may be KH-stabilised specific. Note that a number of important FRI approaches have been proposed, as such, it may be worthwhile to further validate the proposed sparse rule base generation approach using other FRI methods. In addition, the curvature thresholds in this experiment are arbitrarily selected, therefore, a more intelligent approach may be considered. Given that the ultimate goal of the proposed approach is to generate accurate results using a minimum number of rules, the curvature value can also be included in the objective function or fitness function of the GA as a parameter. This will certainly help in balancing the number of rules and the accuracy of the inference results, and both of these suggestions may be considered in further research.

## V. Conclusion

A novel rule base generation method particularly for FRI has been proposed. The approach firstly partitions the input domain into sub-regions, with each sub-region being represented as a fuzzy rule. Those sub-regions with curvature values greater than a predefined threshold are then identified and the corresponding rules are selected to initialise the rule base. This initialised rule base is then optimised by fine-tuning the membership functions of the fuzzy sets involved in the rules. The proposed work was demonstrated using an example based on previous work, with comparable results generated.

The proposed method shows promising and verifiable results and can be further extended by fine-tuning the location of fuzzy rules within the sub-regions. The experimentation presented here is limited to rules with two inputs and one output, and further work may be conducted to include rules with multiple antecedents, thus enhancing the real-world applicability of the method. Given that the proposed method is built upon the stabilised KH approach, it may be worthwhile, in further research, to investigate how this method could support other FRI approaches.

## References


[1] Ebrahim H Mamdani. Application of fuzzy logic to approximate reasoning using linguistic synthesis. *Computers, IEEE Transactions on*, 100(12):1182–1191, 1977.

[2] Tomohiro Takagi and Michio Sugeno. Fuzzy identification of systems and its applications to modeling and control. *Systems, Man and Cybernetics, IEEE Transactions on*, 1:116–132, 1985.

[3] LászlóT Kóczy and Kaoru Hirota. Approximate reasoning by linear rule interpolation and general approximation. *International Journal of Approximate Reasoning*, 9(3):197–225, 1993.

[4] Shyi-Ming Chen and Yu-Chuan Chang. Weighted fuzzy interpolative reasoning for sparse fuzzy rule-based systems. *Expert Systems with Applications*, 38(8):9564–9572, 2011.

[5] Zhiheng Huang and Qiang Shen. Fuzzy interpolative reasoning via scale and move transformations. *Fuzzy Systems, IEEE Transactions on*, 14(2):340–359, 2006.

[6] Longzhi Yang and Qiang Shen. Adaptive fuzzy interpolation and extrapolation with multiple-antecedent rules. In *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*, pages 1–8. IEEE, 2010.

[7] Longzhi Yang and Qiang Shen. Adaptive fuzzy interpolation. *Fuzzy Systems, IEEE Transactions on*, 19(6):1107–1126, 2011.

[8] Longzhi Yang and Qiang Shen. Closed form fuzzy interpolation. *Fuzzy Sets and Systems*, 225:1–22, 2013.

[9] Jie Li, Longzhi Yang, Hubert P. H. Shum, and Yao Tan. Intelligent home heating controller using fuzzy rule interpolation. In *The 15th UK Workshop on Computational Intelligence (UKCI'2015)*, 2015.

[10] Gergely I Molnárka, Szilveszter Kovács, and László T Kóczy. Fuzzy rule interpolation based fuzzy signature structure in building condition evaluation. In *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*, pages 2214–2221. IEEE, 2014.

[11] Min-You Chen and Derek A Linkens. Rule-base self-generation and simplification for data-driven fuzzy models. In *Fuzzy Systems, 2001. The 10th IEEE International Conference on*, volume 1, pages 424–427. IEEE, 2001.

[12] Li-Xin Wang and Jerry M Mendel. Generating fuzzy rules by learning from examples. *Systems, Man and Cybernetics, IEEE Transactions on*, 22(6):1414–1427, 1992.

[13] Zsolt Csaba Johanyák and Szilveszter Kovács. Sparse fuzzy system generation by rule base extension. In *Proceedings of 11th IEEE International Conference of Intelligent Engineering Systems (INES 2007), Budapest, Hungary*, pages 99–104. Citeseer, 2007.

[14] Hatem Bellaaj, Rouf Ketata, and Mohamed Chtourou. A new method for fuzzy rule base reduction. *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, 25(3):605–613, 2013.

[15] Laszlo T Koczy and Kaoru Hirota. Size reduction by interpolation in fuzzy rule bases. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 27(1):14–25, 1997.

[16] Pintu Chandra Shill, MAH Akhand, MD Asaduzzaman, and Kazuyuki Murase. Optimization of fuzzy logic controllers with rule base size reduction using genetic algorithms. *International Journal of Information Technology & Decision Making*, 14(05):1063–1092, 2015.

[17] Chin-Wang Tao. A reduction approach for fuzzy rule bases of fuzzy controllers. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 32(5):668–675, 2002.

[18] Shyi-Ming Chen and Yu-Chuan Chang. A new method for fuzzy rule interpolation based on the ratio of fuzziness of interval type-2 fuzzy sets. In *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*, pages 1–6. IEEE, 2010.

[19] Longzhi Yang, Chengyuan Chen, Nanlin Jin, Xin Fu, and Qiang Shen. Closed form fuzzy interpolation with interval type-2 fuzzy sets. In *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*, pages 2184–2191. IEEE, 2014.

[20] Bernadette Bouchon-Meunier and Llorenç Valverde. A fuzzy approach to analogical reasoning. *Soft Computing*, 3(3):141–147, 1999.

[21] Zhiheng Huang and Qiang Shen. Fuzzy interpolation and extrapolation: A practical approach. *Fuzzy Systems, IEEE Transactions on*, 16(1):13–28, 2008.

[22] Péter Baranyi, László T Kóczy, and Tamás Tom D Gedeon. A generalized concept for fuzzy rule interpolation. *Fuzzy Systems, IEEE Transactions on*, 12(6):820–837, 2004.

[23] Qiang Shen and Longzhi Yang. Generalisation of scale and move transformation-based fuzzy interpolation. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 15(3):288–298, 2011.

[24] Lotfi A Zadeh. Quantitative fuzzy semantics. *Information Sciences*, 3(2):159–176, 1971.

[25] Yu-Chuan Chang, Shyi-Ming Chen, and Churn-Jung Liau. Fuzzy interpolative reasoning for sparse fuzzy-rule-based systems based on the areas of fuzzy sets. *Fuzzy Systems, IEEE Transactions on*, 16(5):1285–1301, 2008.

[26] Wen-Hoar Hsiao, Shyi-Ming Chen, and Chia-Hoang Lee. A new interpolative reasoning method in sparse rule-based systems. *Fuzzy Sets and Systems*, 93(1):17–22, 1998.

[27] Domonkos Tikk, István Joó, László Kóczy, Péter Várlaki, Bernhard Moser, and Tamás D Gedeon. Stability of interpolative fuzzy kh controllers. *Fuzzy Sets and Systems*, 125(1):105–119, 2002.

[28] Pei-Chann Chang, Chen-Hao Liu, and Yen-Wen Wang. A hybrid model by clustering and evolving fuzzy rules for sales decision supports in printed circuit board industry. *Decision Support Systems*, 42(3):1254–1269, 2006.

[29] Scott D Peckham. Profile, plan and streamline curvature: A simple derivation and applications. *Proceedings of Geomorphometry*, pages 27–30, 2011.

[30] David B Thomas, Wayne Luk, Philip HW Leong, and John D Villasenor. Gaussian random number generators. *ACM Computing Surveys (CSUR)*, 39(4):11, 2007.

[31] James Edward Baker. Adaptive selection methods for genetic algorithms. In *Proceedings of an International Conference on Genetic Algorithms and Their Applications*, pages 101–111, 1985.

[32] Naren Naik, Ren Diao, and Qiang Shen. Genetic algorithm-aided dynamic fuzzy rule interpolation. In *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*, pages 2198–2205. IEEE, 2014.

[33] Zsolt Csaba Johanyák, Domonkos Tikk, Szilveszter Kovács, and Kok Wai Wong. Fuzzy rule interpolation matlab toolbox-fri toolbox. In *Fuzzy Systems, 2006 IEEE International Conference on*, pages 351–357. IEEE, 2006.