International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-9 Issue-2, July 2020

A Fast Model Based on Genetic Algorithm to Construct Fuzzy Rules

Mohamed S.S. Basyoni

Abstract: Fuzzy rule has been used extensively in data mining. This paper presents a fast and flexible method based on genetic algorithm to construct fuzzy decision rule with considering criteria of accuracy. First, the algorithm determines the width that divides each attribute into "n" intervals according to the number of fuzzy sets, after that calculates the parameters width according to that width. Rough Sets Model Based on Database Systems technique used to reduce the number of attributes if there exists then we use the algorithm for extracting initial fuzzy rules from fuzzy table using SQL statements with a smaller number of rules than the other models without needing to use a genetic algorithm – Based Rule Selection approach to select a small number of significant rules, then it calculates their accuracy and the confidence.. Multiobjective evolutionary algorithms (EAs) that use nondominated sorting and sharing have been criticized mainly for computational complexity and needing for specifying a sharing parameter but in our genetic model each fuzzy set represented by "Real number" from 0 to 9 forming a gene on chromosome (individual). Our genetic model is used to improve the accuracy of the initial rules and calculates the accuracy of the new rules again which be higher than the old rules The proposed approach is applied on the Iris dataset and the results compared with other models: Preselection with niches, ENORA and NSGA to show its validity.

Keywords: Genetic algorithm, Fuzzy logic, Rough set, SQL statements, Accuracy.

I. INTRODUCTION

A real world data set always contains mixed types of data such as continuous valued, symbolic data, etc. Therefore all numerical or continuous data should be converted to discretized data, here "Fuzzy Logic" can solve this problem to reduce information overload. One drawback of traditional Fuzzy systems [1, 11, 13, 15] is that the fuzzy sets for numeric values of each attribute should determining by the membership functions of these linguistic terms that the user should define this parameters from his view which is different from one user to another. Therefore, we introduce a flexible Automated Fuzzy Model algorithm that can define those parameters automatically that a user determine only the number of fuzzy sets then the algorithm automatically determine the maximum and minimum values of each attribute and calculates the width (Δ) that divides the universe of discourse of each attribute into "n" intervals according to the number of fuzzy sets, after that calculates the parameters width (δi) according to width (Δ).

Rough Sets Model Based on Database Systems algorithm [6] is used to reduce the number of attributes if there exists through redefine some concepts of rough set theory such as core and reducts by using relational algebra so that the computation of core and reducts can be performed with very efficient set-oriented database operations.

our model used for extracting initial fuzzy rules from fuzzy table using SQL statements then calculates their accuracy and the confidence. This method can obtain an original rule set with a smaller number of rules than the other models [3, 4, 7] without needing to a genetic algorithm approach to select a small number of significant rules from those rules (Genetic Algorithm – Based Rule Selection) to reduce those large number of candidate fuzzy rules. Multiobjective evolutionary algorithms (EAs) that use nondominated sorting and sharing [4, 8, 14] have been criticized mainly for their: 1) *O(MN)* computational complexity (where *M* is the number of objectives and *N* is the population size); 2) the need for specifying a sharing parameter but in our genetic model each fuzzy set represented by "Real number" from 0 to 9 forming a gene on chromosome (individual). Our algorithm using Genetic algorithm technique to generate another new fuzzy rule from the initial rules then calculates their accuracy again which will be higher than the old rules before using genetic algorithm. The proposed model is applied on the Iris dataset and the results compared with other models: Preselection with niches, ENORA and NSGA [2, 4, 7, 9, 14] to show its validity.

This paper is organized as follows: We proposed our model and give an overview of the fuzzy logic theory and genetic algorithm and propose a new automated Fuzzy Based algorithm in section II. Also in the same section a Rough Sets Model Based on Database Systems technique is used to reduce the number of attributes if there exists. After that, we generate fuzzy rules using SQL statements. Finally, we use the genetic algorithm on the fuzzy rules and generate other efficient fuzzy rules in accuracy. In section III, shows the experiments and the results obtained for the problem of classification of the Iris data set and compare the result of our model with other models: Preselection with niches, ENORA and NSGA. In section IV, explain the comparison by performance curve of our model with the others. Finally, we conclude with some discussions in Section V.

Manuscript received on May 25, 2020.
Revised Manuscript received on June 29, 2020.
Manuscript published on July 30, 2020.
***** Correspondence Author
Mohamed S.S.Basyoni *, Department of Computer Sciences, Faculty of Graduate Studies for Statistical Research, Cairo University, Egypt. Email: mohamedssb@ yahoo.com

II. PROPOSED MODEL

A. Fuzzy Logic (FL)

Fuzzy logic has been applied very successfully in many areas where conventional model based approaches are difficult or not cost-effective to implement. However, as system complexity increases, reliable fuzzy rules and membership functions used to describe the system behavior are difficult to determine. The fuzzy rule set is included in the rule base. The antecedents of the fuzzy rules define local fuzzy regions, while the consequents describe the classification labels within those regions. The consequent of the rule is the class and the accuracy of the classification. The confidence and support of the association rules are used to choose the rules.

The fuzzy rule is considered to be the associate rule, which is described as follows:

$$Aq \Rightarrow Cq$$

Where A*q* is the antecedent of the rule, and C*q* is the consequent of the rule. These two conceptions are extended to the fuzzy association rule. The forms proposed by Ishibuchi and Nakashima [10] are used in this paper to get the confidence and support of the association rules as:

$$C\,(Aq \rightarrow Cq) = \frac{|D(Aq) \cap D(Cq)|}{|D(Aq)|} = \frac{\sum \mu Aq(x) \cap \mu Cq(x)}{\sum \mu Aq(x)} \qquad (1)$$

$$S\,(Aq \rightarrow Cq) = \frac{|D(Aq) \cap D(Cq)|}{|D|} = \frac{\sum \mu Aq(x) \cap \mu Cq(x)}{\text{No. of all tuples}} \qquad (2)$$

Where,

Aq: the antecedent part of the rule

Cq: the consequent part of the rule

$|D(Aq)| = \Sigma \mu Aq(x)$: the cardinality of a fuzzy set

|D|: no. of all patterns

Example .1

Suppose the eleven examinees whether they feel comfortable in a small car or not as shown in table 1

Table- I: the eleven examines on comfortable in a small car

Examinee (*p*)	*Weight* (*xp*)	Comfortable
1	45	yes
2	50	yes
3	55	yes
4	60	no
5	65	yes
6	70	no
7	75	no
8	80	no
9	85	no
10	90	no
11	95	no

Suppose the fuzzy set

$$D\,(middle) = \left\{ \frac{0.33}{55}, \frac{0.67}{60}, \frac{1}{65}, \frac{1}{70}, \frac{1}{75}, \frac{0.67}{80}, \frac{0.33}{85} \right\}$$

Now we will calculate *C* (*middle* => *uncomfortable*) and *C* (*middle* => *comfortable*) as following:

The total compatibility grade with the linguistic term *middle* is calculated as:

| *D* (*middle*) | = 0.33 + 0.67 + 1 + 1 + 1 + 0.67 + 0.33 = 5

1- From the table, | *D* (*middle*) ∩ *D (uncomfortable*) | = 0.67 + 1 + 1 + 0.67 + 0.33 = 3.67

The confidence and the support of the linguistic association rule **"*middle => uncomfortable*"** calculated as:

C (*middle* => *uncomfortable*) = 3.67 / 5 = 0.73

S (*middle* => *uncomfortable*) = 3.67 / 11 = 0.33

2- From the table, | *D* (*middle*) ∩ *D* (*comfortable*) | = 0.33 + 1 = 1.33

The confidence and the support of the linguistic association rule **"*middle => comfortable*"** calculated as:

C (*middle* => *comfortable*) = 1.33 / 5 = 0.27

S (*middle* => *comfortable*) = 1.33 / 11 = 0.12

Since *C* (*middle* => *uncomfortable*) is larger than *C* (*middle* => *comfortable*),

We choose the linguistic association rule: **"*middle* => *uncomfortable*"** rather than "*middle* => *comfortable*".

Each Linguistic Variable has a range of states called linguistic values, each of which is a fuzzy (linguistic) set defined over the same domain represented by its membership function this domain is called the *universe of discourse* of that linguistic variable

- **Triangular Membership Function**

$$\text{Triangular}\,(X\!: a, b, c) = \begin{cases} 0 & x \le a \\ (x - a) / (b - a) & a \le x \le b \\ (c - x) / (c - b) & b \le x \le c \\ 0 & x \ge c \end{cases}$$

For "n" linguistic values (fuzzy sets) then the universe of discourse "*U*" which is the attribute divided into "n" intervals with the width between center bi and bi+1 (Δ) where:

$$\Delta = \frac{Xmax - Xmin}{n - 1} \qquad \text{where,} \quad 1 \le x \le n+1 \qquad (3)$$

With: Xmin = Xmin – D1, Xmax = Xmax + D2,

and δi = (Δ+1) / 2

We can also calculate the parameters a, b and c by the following equations:

<u>Ai = (ai, bi, ci)</u>

bi = Xmin + (i - 1) * Δ (4)

With ai = bi – δi , ci = bi + δi

Where:

Δ: the width between center bi and bi+1

Xmax: the maximum value of the attribute

Xmin: the minimum value of the attribute

n: the number of linguistic values (fuzzy sets)

D1: A value subtracted from Xmin to make it integer value

D2: the value added to Xmax to make it integer value

δi: the parameters width between bi and ai, ci

B. Reduce the Number of Attributes using Rough Set Theory

After Fuzzifing original information system a New Rough Sets Model Based on Database Systems has been introduced [6] to redefine some concepts of rough set theory such as core attributes and reducts by using relational algebra so that the computation of core attributes and reducts can be performed with very efficient set-oriented database operations, such as the following relational algebra: *Cardinality* (*Card*) to denote the *Count*, and Π for *Projection* operation.

Example .2

Suppose we have a collection of 8 cars ($t_1, t_2, \ldots t_8$) as in table II, with information about the attributes *Weight*, *Door*, *Size*, *Cylinder* and *Mileage*, where *Weight*, *Door*, *Size* and *Cylinder* are the condition attributes and *Mileage* is the decision attribute

Table II: Cars table with attributes Weight, Door, Size, Cylinder and Mileage

Tuple-id	*Weight*	*Door*	*Size*	*Cylinder*	*Mileage*
t_1	low	medium	compact	medium	high
t_2	low	high	sub	high	low
t_3	medium	high	compact	medium	high
t_4	high	medium	compact	high	low
t_5	high	high	compact	medium	low
t_6	low	high	compact	medium	high
t_7	high	high	Sub	high	low
t_8	low	medium	sub	high	low

Definition I. An attribute $Cj \in C$ is a *core* attribute with respect to *D*, if it satisfies the following condition:

$$Card\ (\Pi\ (C - Cj + D)) \neq Card\ (\Pi\ (C - Cj)) \quad \text{Where:}$$

Card: The *cardinality* to denote the *count* of attribute
Π: *Projection* operation
C: Condition attributes *D:* Decision attribute

For example, in Table II, it can be shown that

Card (Π (*C* – *Cj* + *D*)) = *Card* (Π (*Door*, *Size*, *Cylinder*, *Mileage*) = 6,

Card (Π (*C* – *Cj*)) = *Card* (Π (*Door*, *Size*, *Cylinder*) = 5
Therefore, the attribute *Weight* is a *core* attribute in *C* with respect to attribute *Mileage*.

Definition II. An attribute $Cj \in C$ is a *dispensable* attribute in *C* with respect to *D*, if the classification result of each tuple is not affected without using *Cj*, that is,

$$Card\ (\Pi\ (C - Cj + D)) = Card\ (\Pi\ (C - Cj))$$

For example, in Table II, it can be shown that
Card (Π (*C* – *Cj* + *D*)) = *Card* (Π (*Weight*, *Size*, *Cylinder*, *Mileage*) = 6,

Card (Π (*C* – *Cj*)) = *Card* (Π (*Weight*, *Size*, *Cylinder*) = 6

Thus, *Door* is a dispensable attribute in *C* with respect to attribute *Mileage*.

Definition III. Let *RED* be a subset of the condition attributes set, $RED \subseteq C$. The degree of dependency between *RED* and the decision attribute set *D* in the decision table *T*(*C*, *D*), denoted K(*RED*, *D*), is defined as

$$K\ (RED, D) = Card\ (\Pi\ (RED + D)) \ /\ Card\ (C + D)$$

The value *K* (*RED*, *D*) is the proportion of those tuples in the decision table that can be classified. This value characterizes the ability to predict the class *D* and the complement ¬ *D* from tuples in the decision table.

Definition IV. The *merit* value of an attribute *Cj* in *C* is defined as:

$$Merit\ (Cj,C,D) = 1 - (Card\ (\Pi\ (C - Cj + D)) \ /\ Card\ (\Pi\ (C - Cj)))$$

Merit (*Cj*, *C*, *D*) reflects the degree of contribution made by the attribute *Cj* to the dependency only between *C* and *D*.
For example, in table II, it can be shown that
Card (Π (*C* – *Cj* + *D*)) = *Card* (Π (*Door*, *Size*, *Cylinder*, *Mileage*) = 6,

Card (Π (*C* + *D*)) = *Card* (Π (*Weight*, *Door*, *Size*, *Cylinder*, *Mileage*) = 8

$$Merit\ (Weight, Door, Weight, Size, Cylinder, Mileage) = 1 - 6/8 = 0.25$$

The algorithm for finding all core attributes based on the relational database system [6] without calculation of the lower and upper approximations is described in Algorithm .I as shown:

Algorithm I: *Core Attributes*

Input: a decision table T (*C*, *D*)
Output: Core - the core attribute of table T
Method:
1. Set Core = Φ
2. For each attribute $Cj \in C$
{
If Card (Π (*C* – *Cj* + *D*)) ≠ Card (Π (*C* – *Cj*))
Then Core = Core ∪ *Cj*
}

Algorithm .I consists of two steps. The first step is simply to initialize the core attributes set to be empty. The second step checks all condition attributes one by one to see if they are core attributes. If yes, they are added to the set of the core attributes.

The algorithm computes a minimal attribute subset (reduct) based on the relational database system [6] is described in Algorithm .II as shown:

Algorithm II: *Compute a minimal attribute subset (reduct)*

Input: A decision table T (*C*, *D*)
Output: A set of minimum attribute subset (RED, *D*)
Method:
1. Run Algorithm 1 to get a core attributes of the table T
2. RED = T
3. AR = *C* - RED
4. Compute the merit values for all attributes of AR
5. Sort attributes in AR based on merit values in decreasing order
6. Choose an attribute *Cj* with the largest merit values (if there are several attributes with the same merit value, choose the attribute which has the least number of combinations with those attributes in RED)
7. RED = RED ∪ {*Cj*}, AR = AR - {*Cj*}

8. If K (RED, *D*) = 1, then terminate, otherwise go back to Step 4

The output of this algorithm is a reduct *Weight*, *Door* and *Size* from the data in Table II. For each reduct, we can derive a reduct table from the original table.

C. Generating Fuzzy Rules

After reduce the number of attributes if there exists we can generate Fuzzy rules which help decision maker to take the proper decision. Most traditional methods [3, 4, 7] used a heuristic method to generate a linguistic rule for each cell of the pattern space which generates a large number of rules and need a genetic algorithm approach to select a small number of significant rules from them (Genetic Algorithm – Based Rule Selection) to reduce those large number of candidate fuzzy rules. This means that if we have K fuzzy sets for each of n attributes, then the number of linguistic rule = K^n rules.

For example, if we have 3 fuzzy sets and 4 attributes then the number of fuzzy rules will be (81) rules which will be a big number of rules and need a genetic algorithm to reduce them.

For this problem we use the algorithm for extracting fuzzy rules using SQL statements which generates efficient smaller number of fuzzy rules in terms of accuracy immediately without needing to run a genetic algorithm approach to do this step. In this algorithm all training patterns are fully compatible with the antecedent part of the fuzzy rules (i.e. The Training Classification Rate CR-training = 100%). Therefore, our model is faster than the other models. The algorithm for extracting fuzzy rules using SQL statements running by 3 steps:

1- Create temp table contains the selected reduct and decision attribute
2- Get all equivalence classes for the selected reduct
3- Get all decision rules for the current equivalence class

- **Create temp table**

```
SELECT      CURR_REDUCT, D
INTO        TMP_TBL
FROM        T
```

- **Get equivalence classes for current reduct**

```
SELECT      CURR_REDUCT
FROM        TMP_TBL
GROUP BY    CURR_REDUCT
```

- **Get decision rule for each equivalence classes**

```
SELECT    DISTINCT  D
FROM      TMP_TBL
WHERE     X1 = Y1  AND  X2 = Y2  AND …. Xn = Yn
```

D. Genetic Algorithm

A genetic algorithm is a way to perform a heuristic search in a solution space based on the evolutionary ideas of natural selection and genetics. It is an iterative procedure maintaining a population of structure of candidate solutions to specific domain, which each candidate solution is called a chromosome (or individual)

Genetic algorithms have been widely proposed to generate fuzzy if-then rules and tune the membership function of fuzzy sets in fuzzy rules [3, 5, 9, 10, 12].

In our genetic model each fuzzy set represented by "Real number" from 0 to 9 forming a gene on chromosome (individual) of fixed length that forms a population of fixed number of chromosomes. In addition to the linguistic terms, "don't care" is also used for each attribute as an additional antecedent fuzzy set.

For example we represent three fuzzy sets Low, Medium and High as:

1: Low, 2: Medium, 3: High, 0: Do not care

Therefore, if we have the following rule of five attributes and each attribute have the three previous fuzzy sets as:

Rule R_i: If x_1 is High and x_2 is Medium and x_3 is Low And x_5 is High then Class C_j

Therefore, the representation of this rule as chromosome will be as figure.1

3	2	1	0	3

Fig .1: Representation of rule as chromosome

The consequent class C_j of each fuzzy if-then rule is determined by the test patterns in the fuzzy subspace specified by the antecedent fuzzy sets.

After that, set parameters, max number of generations MAXGEN, population size and a fitness value condition

We set the max number of generations MAXGEN is 100, the accuracy value condition we need to reach is 98% and population size is 20. Also the Crossover probability p_c is 0.8; the Mutation probability p_m is 1/ (string length).

Next the fitness value of each linguistic rule in the current population is evaluated. Let S be the set of fuzzy rules in the current population.

The evaluation of fitness value of each fuzzy rule is performed by classifying all given test patterns by the rule set S using the single winner-based method. The rule receives a unit reward when it correctly classifies test patterns. After all the given test training patterns are classified by the rule set S, the fitness value $fitness(R_q)$ of each linguistic rule R_q is calculated as the following:

$$fitness(R_q) = NCP(R_q) \qquad (5)$$

Where, $NCP(R_q)$ is the number of correctly classified test patterns by R_q.

After that, the fitness value $NCP(S)$ of the rule set S is calculated as the following:

$$NCP(S) = \sum_{R_q \in S} NCP(R_q) \qquad (6)$$

We propose our constrained genetic model, with two objectives:

$$\text{Maximize} \quad NCP(S)$$
$$\text{Minimize} \quad |S|$$

Where,

$NCP(S)$ is the number of correctly classified test patterns by the rule set S, and $|S|$ is the number of fuzzy if-then rules in S.

The Algorithm is described as follows:

Step1: *Parameter Specification.* Population size, maximum generation MAXGEN, and a fitness value condition.

Step2: *Initialization.* Specify an initial population, the number of antecedent of fuzzy sets and their representation as a strings of length n, and set generation gen = 1.

International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-9 Issue-2, July 2020

Step3: *Genetic Operations*. Calculate the fitness value of each linguistic rule in the current population using the test patterns. Generate new linguistic rule using selection, crossover, and mutation from existing linguistic rule in the current population.
Step4: *Generation Update*. Remove the worst linguistic rules from the current population and add the newly generated rules to the current population gen = gen +1.
Step5: *Termination Test.* If gen = MXAGEN or get a fitness value then go to Step 6 else go to Step3.
Step6: *Best Rules*. Take the individual with maximum fitness value as the optimal fuzzy rule.

The Algorithm flow chart is as follows Figure 2:

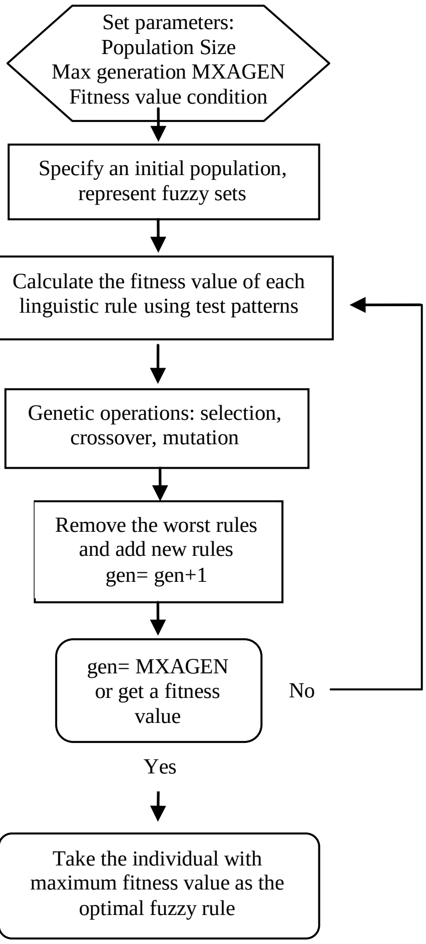

Fig. 2: Algorithm flow chart

III. EXPERIMENTS AND RESULTS

The Iris system is a common benchmark problem in classification and pattern recognition studies. It contains 150 measurements of four features (sepal length *(S_L)*, sepal width *(S_W)*, petal length *(P_L)*, petal width *(P_W)*) from each of three species Class *(RS)* (Setosa, Versicolor, Virginica). Therefore, there are four independent variables that form the antecedent of the rule, and three dependent variables that form the consequence of the rule. From each species, there are 50 observations regarding sepal length, sepal width, petal length and petal width (in cm).
We choose "Triangular Membership function", then define "3" fuzzy sets: Low, Medium and High. Therefore, to fuzzifing numerical data of attribute *Sepal length (S_L)*, then:

Xmin = 4.3 and Xmax = 7.9 So, D1 = 1 and D2 = 0.1
So, Xmin = 3.3 and Xmax = 8

The width between bi and bi+1 is: Δ = 8-3.3/ (3-1) = 2.35
Then, δi = (2.35+1) / 2 = 1.68, therefore:
1- Fuzzy set A1 ⟶ Low, δ1 =1.68
b1 = a1 = 3.3, c1 = 4.98
2- Fuzzy set A2 ⟶ Medium, δ2 =1.68
b2 = 5.65, a2 = 3.97, c2 = 7.33
3- Fuzzy set A3 ⟶ High, δ3 = 1.68
b3 = c3 = 8, a3 = 6.32

The result will be shown as Figure .3

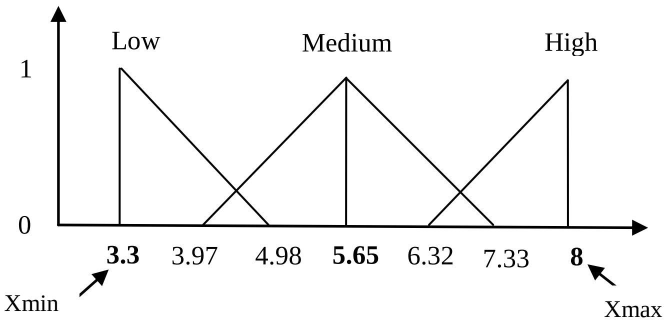

Fig .3: Sepal length (S_L) attribute

In the same manner to fuzzifing numerical data of attribute *Sepal width (S_W)*, then:

Xmin = 2.2 and Xmax = 4.4 So, D1 = 0.2 and D2 = 0.6
So, Xmin = 2 and Xmax = 5

The width between bi and bi+1 is: Δ = 5-2/ (3-1) = 1.5
Then, δi = (1.5+1) / 2 = 1.25, therefore:
1- Fuzzy set A1 ⟶ Low, δ1 =1.25
b1 = a1 = 2, c1 = 3.25
2- Fuzzy set A2 ⟶ Medium, δ2 =1.25
b2 = 3.5, a2 = 2.25, c2 = 4.75
3- Fuzzy set A3 ⟶ High, δ3 = 1.25
b3 = c3 = 5, a3 = 3.75

The result will be shown as Figure .4

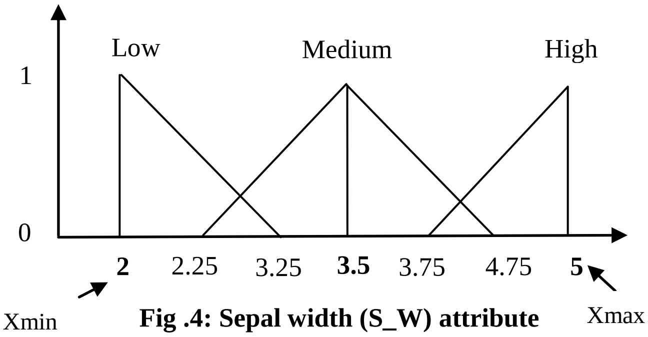

Fig .4: Sepal width (S_W) attribute

The algorithm fuzzifing numerical data of two attributes *Petal length (P_L), Petal width (P_W)* in the same manner.
Thus, after fuzzifing numerical data of the all four condition attributes, the result will be as shown in table III.

Table- III: Iris data table after fuzzifing four numerical condition attributes

Sno	*S_L*	*S_W*	*P_L*	*P_W*	*RS*
1	medium	medium	low	low	setosa
2	medium	medium	low	low	setosa
3	medium	medium	low	low	setosa
4	medium	medium	low	low	setosa
⋮					
51	high	medium	medium	medium	versicolor
52	medium	medium	medium	medium	versicolor
53	high	medium	medium	medium	versicolor
⋮					
101	medium	medium	high	high	verginica
102	medium	low	medium	medium	verginica
103	high	medium	high	medium	verginica
⋮					
150	medium	medium	medium	medium	verginica

After that we input a number of instances from table III that the algorithm *randomly* selects and generates the fuzzy rules from them. So, the remaining instances will be the test records. The algorithm calculates the accuracy of those fuzzy rules. We can easily select *Class (RS)* attribute as a decision attribute in table 2 then the remaining attributes *S_L*, *S_W*, *P_L* and *P_W* will be a four condition attributes.

We assumed that the number of fuzzy sets is (*L*), the number of fuzzy rules is (*M*) and the classification rate (*CR*) for evaluating the accuracy of training and test instances. We randomly choose 100 instances as the training instances and 50 instances as the testing instances. Then the algorithm generates 12 fuzzy rules from those 100 records using triangular membership functions as Figure .5.

The training classification rate (*CR*-training) of those rules will be 1.00000 and the number of fuzzy sets *L* is 3 fuzzy sets.

For the fuzzy rules in figure 5 that generated, the algorithm will automatically calculate the confidence and the accuracy of each fuzzy rough rule, the confidence calculated using the Average operator that takes the average confidence value of the fuzzy sets of each rule then it calculates the total accuracy of all rules as the following results in the table IV.

S_L is 'High' And S_W is 'Medium' And P_L is 'High' And P_W is 'High'
==> RS is virginica
S_L is 'High' And S_W is 'Medium' And P_L is 'High' And P_W is 'Medium'
==> RS is virginica
S_L is 'High' And S_W is 'Medium' And P_L is 'Medium' And P_W is 'Medium'
==> RS is versicolor
S_L is 'Low' And S_W is 'Medium' And P_L is 'Low' And P_W is 'Low'
==> RS is setosa
S_L is 'Medium' And S_W is 'High' And P_L is 'Low' And P_W is 'Low'
==> RS is setosa
S_L is 'Medium' And S_W is 'Low' And P_L is 'High' And P_W is 'Medium'
==> RS is virginica
S_L is 'Medium' And S_W is 'Low' And P_L is 'Medium' And P_W is 'Medium'
==> RS is versicolor OR virginica
S_L is 'Medium' And S_W is 'Medium' And P_L is 'High' And P_W is 'High'
==> RS is virginica
S_L is 'Medium' And S_W is 'Medium' And P_L is 'High' And P_W is 'Medium'
==> RS is virginica
S_L is 'Medium' And S_W is 'Medium' And P_L is 'Low' And P_W is 'Low'
==> RS is setosa
S_L is 'Medium' And S_W is 'Medium' And P_L is 'Medium' And P_W is 'High'
==> RS is virginica
S_L is 'Medium' And S_W is 'Medium' And P_L is 'Medium' And P_W is 'Medium'
==> RS is versicolor

Fig .5: The initial fuzzy rules

Table-IV: Confidence and Accuracy of each Fuzzy Rule before running Genetic algorithm

Fuzzy Rules No.	frequency of rows	Confidence	Accuracy
R1	1	0.82	2
R2	5	0.71	4
R3	2	0.66	0
R4	3	0.84	1
R5	2	0.83	0
R6	1	0.63	1
R7	19	0.58	2
R8	5	0.71	1
R9	4	0.6	5
R10	28	0.7	15
R11	1	0.66	2
R12	29	0.55	1
Total	***100***		***34***
***CR*-training**	***1***		
Accuracy % (*CR* – evaluation)			***0.68***

The last phase is running the Genetic algorithm on the pervious fuzzy rules. We represent the three Fuzzy sets in each rule as a one chromosome with a number from 0 to 9. Suppose we represent each fuzzy set of the pervious fuzzy rules by real number forming a gene on chromosome (individual) of fixed length that forms a population of fixed number of chromosomes as following:

1: Low, 2: Medium, 3: High, 0: Do not care

After that, set parameters, max number of generations MAXGEN, population size and a fitness value condition

We set the max number of generations MAXGEN is 100, the accuracy value condition we need to reach is 98% and population size is 20. Also the crossover probability is 0.8. Then the algorithm generates other new rules and adds them to the fuzzy rules. The algorithm adds (3) new rules that shown in table 3 in bolded font.

After that, the accuracy of those new 12 rules from randomly 100 rows will automatically calculated to be 0.96 instead of 0.68

The results of Iris Dataset after running Genetic will be as following in the table V.

We propose our constrained optimization model, with two objectives:

$$\textit{Maximize} \quad CR$$
$$\textit{Minimize} \quad M \text{ and } L$$

Table- V: Confidence and Accuracy of each Fuzzy Rule after running Genetic algorithm

Fuzzy Rules No.	frequency of rows	Confidence	Accuracy
R1	1	0.82	2
R2	**5**	**0.76**	**5**
R3	2	0.66	0
R4	3	0.84	1
R5	2	0.83	0
R6	1	0.63	1
R7	**19**	**0.85**	**10**
R8	5	0.71	1
R9	4	0.6	5
R10	28	0.7	15
R11	1	0.66	2
R12	**29**	**0.75**	**6**
Total	***100***		***48***
***CR*-training**	***1***		
Accuracy % (*CR* – evaluation)			***0.96***

Finally, we compare our model with Preselection with niches, NSGA-II (Non dominated sorting genetic algorithm II) and ENORA (Evolutionary Non dominated sorting with Radial slots) algorithms [2, 4, 7, 9, 14]. The result shows that our algorithm is better than the others in term of Accuracy (*CR* –evaluation) and *CR* –training and also in the number of fuzzy sets.
The chosen solution by the decision process is bolded as shown in table VI.

Table- VI: The best results obtained in this paper for Iris data

Models names	***M***	***L***	***CR*-training**	***CR*-evaluation**
Preselection w. niches	10	9	0.989899	0.94
NSGA-II	10	9	0.989899	0.94
ENORA	10	8	0.989899	0.92
	13	8	1	0.94
Model of This paper	**12**	**3**	**1**	**0.96**

IV. THE PERFORMANCE CURVE

The performance curve of values for maximum Accuracy % (*CR* –evaluation) for each generation of the our program comparing with the other models Preselection with niches, NSGA-II and ENORA algorithms [2, 4, 7, 9, 14] can be represented by Figure. 6. The result shows that our algorithm is better than the others in term of Accuracy (*CR* –evaluation) which is 96% rather than 92% and 94% in the other models.

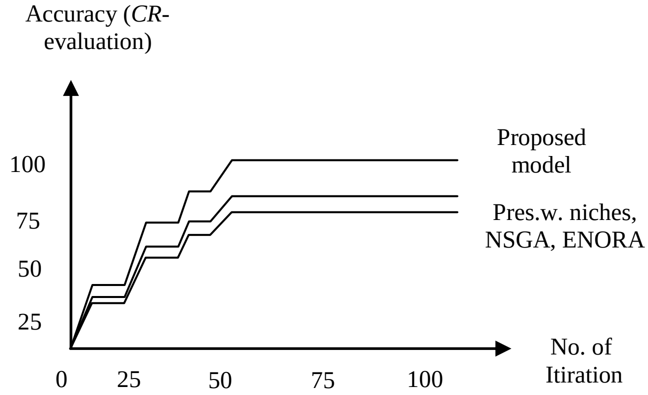

Figure .6: The performance curve for Iris Dataset

V. CONCLUSION

We have presented a flexible method for handling the Iris data classification problem.
A Genetic-Based Fuzzy Decision Model is proposed to be more flexible than the Traditional Genetic fuzzy models. First in traditional models the user should define the parameters of membership functions of each attribute from his view which is different from one user to another but our algorithm can define them automatically by determine the maximum and minimum values and the number of fuzzy sets of each attribute then it finds the width (Δ) that divides the universe of discourse "U" of each attribute into "n" intervals according to the number of fuzzy sets that we defined then it calculates the parameters width (δi) according to the width (Δ). After Fuzzifing original information system a New Rough Sets Model Based on Database Systems has been introduced to redefine some concepts such as core attributes and reducts by using relational algebra to reduce the number of attributes if there exists so that the computation of core attributes and reducts can be performed with very efficient set-oriented database operations. Most traditional methods used a heuristic method to generate a linguistic rule which generates a large number of rules and need a genetic algorithm approach to select a small number of significant rules from them (Genetic Algorithm – Based Rule Selection) to reduce those large number of candidate fuzzy rules. For this problem we use the algorithm for extracting fuzzy rules using SQL statements which generates efficient smaller number of fuzzy rules in terms of accuracy immediately without needing to run a genetic algorithm approach to do this step. After that, our model calculates automatically the confidence of each rule by using the Average operator that takes the average confidence value of the fuzzy sets of each rule. After that it calculates the accuracy of each fuzzy rule. The rules that have a high accuracy are called Strong rules. Multiobjective evolutionary algorithms (EAs) that use nondominated sorting and sharing have been criticized mainly for computational complexity and needing for specifying a sharing parameter but in our genetic model each fuzzy set represented by "Real number" from 0 to 9 forming a gene on chromosome (individual).

Our algorithm using Genetic algorithm technique to generate another new fuzzy rule from the initial rules then calculates their accuracy again which will be higher than the old rules before using genetic algorithm.
The proposed model is applied on the Iris dataset and the results compared with other models: Preselection with niches, NSGA-II (Non dominated sorting genetic algorithm II) and ENORA (Evolutionary Non dominated sorting with Radial slots) algorithms. The result shows that our algorithm is better than the other models in term of Accuracy (CR –evaluation) and in CR –training and also in the number of fuzzy sets.

REFERENCES

1. Yan-Qing Yao, Ju-Sheng Mi, Zhou-Jun Li. Attribute reduction based on Generalized fuzzy evidence theory in fuzzy sets and systems. Fuzzy Sets and Systems, (2011) 509-517.
2. Laumanns, M., Zitzler, E., Thiele, L.: On the Effects of Archiving, Elitism, and Density Based Selection in Evolutionary Multi-objective Optimization. In: Zitzler, E., Deb, K., Thiele, L., Coello Coello, C.A., Corne, D.W. (eds.) EMO 2001. LNCS, vol. 1993, pp. 181–196. Springer, Heidelberg (2001)
3. Cordon, O., Herrera, F., Hoffmann, F., Magdalena, L. (2001): Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases. World Scientific Publishers, Singapore.
4. Fernando Jiménez, Gracia Sánchez, Fuzzy Classification with Multi-objective Evolutionary Algorithms, Springer-Verlag Berlin Heidelberg 730–738, (2008).
5. I.H. Toroslu, Y. Arslanoglu, Genetic algorithm for the personnel assignment problem with multiple objectives, Information Sciences 177 (2007) 787–803.
6. Hu, X., Lin, T., Han, J, " A New Rough Sets Model Based on Database Systems" , Fundamenta Information 59 no. 2-3 pp.135-152 ., 2004
7. Jiménez, F., Gómez-Skarmeta, A.F., Sánchez, G., Deb, K.: An evolutionary algorithm for constrained multi-objective optimization. In: Proceedings IEEE World Congress on Evolutionary Computation (2002).
8. Yusliza Yusoff*, Mohd Salihin, Overview of NSGA-II for Optimizing Machining Process Parameters, Published by Elsevier Ltd (2007).
9. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. John Wiley and Sons, LTD, Chichester (2001)
10. Hisao Ishibuchi, Tomoharu Nakashima, Manabu Nii. Classification and Modeling with Linguistic Information Granules. Springer Berline Heidelberg New York (1998).
11. Gómez-Skarmeta, A.F., Jiménez, F.: Fuzzy modeling with hibrid systems. Fuzzy Sets and Systems 104, 199–208 (1999)
12. Gómez-Skarmeta, A.F., Jiménez, F., Sánchez, G.: Improving Interpretability in Approximative Fuzzy Models via Multiobjective Evolutionary Algorithms. International Journal of Intelligent Systems 22, 943–969 (2007)
13. Ishibuchi, H., Nakashima, T., Murata, T.: Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems. IEEE Transactions on Systems, Man, and Cubernetics - Part B: Cybernetics 29(5), 601–618 (1999)
14. Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, 6, 2 (2002), 182-197.
15. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Grunert da Fonseca, V.: Performance Assessment of Multiobjective Optimizers: An Analysis and Review. IEEE Transactions on Evolutionary Computation 7(2), 117–132 (2003).

AUTHORS PROFILE

Mohamed S.S.Basyoni
Department of Computer Sciences, Faculty of Graduate Studies for Statistical Research, Cairo University, Egypt. mohamedssb@ yahoo.com