A NEW METHOD FOR ESTIMATING NULL VALUES IN RELATIONAL DATABASE SYSTEMS BASED ON GENETIC ALGORITHMS

Shih-Wei Lee* and Shyi-Ming Chen**

*Department of Electronic Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, R. O. C.

**Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, R. O. C.

ABSTRACT

In this paper, we present a new method to estimate null values in relational database systems based on genetic algorithms. It can tune the membership functions of the linguistic values of the attributes in relational database systems for estimating null values. The proposed method can get a higher average estimated accuracy rate than the existing methods for estimating null values in relational database systems.

1. INTRODUCTION

In recent years, many methods have been proposed to estimate null values in relational database systems [2], [3], [7], [9], [10], [12]. In [2], Chen et al. presented a method to generate fuzzy rules from relational database systems for estimating null values. In [3], Chen et al. presented a method for estimating null values in the distributed relational databases environment. In [7], Hsiao et al. presented a method to estimate null values in relational database systems based on automatic clustering techniques. In [9], Huang et al. presented a method to estimate null values in relational database systems using genetic algorithms. In [10], Huang et al. presented a method to estimate null values in relational database systems with a negative dependency relationship between attributes. In [12], we extended the fuzzy concept learning system (FCLS) algorithm presented in [2] to present a new method to generate fuzzy rules from relational database systems for estimating null values, where the attributes appearing in the antecedent portions of the generated fuzzy rules have different weights, and the weights of the attributes are used to derive the certainty factor value of each generated fuzzy rule for estimating null values in relational database systems.

In this paper, we extend the work we presented in [12] to present a new method for estimating null values in relational database systems based on genetic algorithms. The proposed method can get a higher average estimated accuracy rate than the existing methods for estimating null values in relational database systems.

2. BASIC CONCEPTS OF FUZZY SETS

In 1965, Zadeh proposed the theory of fuzzy sets [16]. Let U be the universe of discourse, $U = \{u_1, u_2, ..., u_n\}$. A fuzzy set A of the universe of discourse U can be represented as follows:

$$A = \mu_A(u_1)/u_1 + \mu_A(u_2)/u_2 + ... + \mu_A(u_n)/u_n, \quad (1)$$

where $\mu_A(u_i)$ indicates the grade of membership of u_i in the fuzzy set A, $\mu_A(u_i) \in [0, 1]$, and $1 \le i \le n$.

The triangular fuzzy set A shown in Fig. 1 can be parametrized as (a, b, c), where "b" is called the "center" of the triangular fuzzy set A, and "a" and "c" are called the "left vertex" and the "right vertex" of the triangular fuzzy set A, respectively.

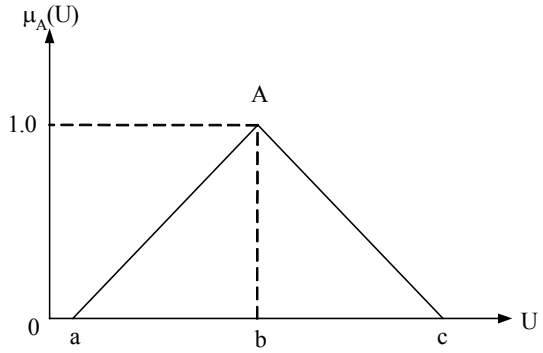

Fig. 1. A triangular fuzzy set.

The trapezoidal fuzzy set A shown in Fig. 2 can be parametrized as (a_1, a_2, a_3, a_4), where "a_1", "a_2", "a_3" and "a_4" are called the "left vertex", the "left corner", the "right corner" and the "right vertex" of the trapezoidal fuzzy set A, respectively.

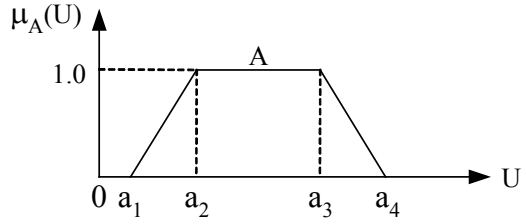

Fig. 2. A trapezoidal fuzzy set.

In [4], Chen have presented the defuzzification techniques of trapezoidal fuzzy sets based on [11]. Let A be a trapezoidal fuzzy set, where A = (a, b, c, d). Then, the defuzzified value DEF(A) of the trapezoidal fuzzy set A is as follows:

$$DEF(A) = \frac{a+b+c+d}{4}. \quad (2)$$

A triangular fuzzy set can be regarded as a special case of a trapezoidal fuzzy set. Let A be a triangular fuzzy set,

where A = (a, b, c). Based on [4] and [11], the defuzzified value DEF(A) of the triangular fuzzy set A is as follows:

$$DEF(A) = \frac{a + 2b + c}{4}. \quad (3)$$

3. A REVIEW OF LEE-AND-CHEN'S METHOD FOR ESTIMATING NULL VALUES IN RELATIONAL DATABASE SYSTEMS

In [12], we use the concept of "coefficient of determination" of the statistics [1], [13] to calculate the coefficients of determination of related attributes in relational database systems.

Definition 3.1: Assume that there are two variables X and Y, where X is an independent variable and Y is a dependent variable, then

$$\text{Coefficient of Determination from X to Y} = \left(\frac{\sum_{i=1}^{n} (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n} (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^{n} (Y_i - \bar{Y})^2}} \right)^2, \quad (4)$$

where X_i denotes the ith value of the variable X, Y_i denotes the ith vslue of the variable Y, $1 \le i \le n$, $\bar{X}$ denotes the mean value of the variable X, and $\bar{Y}$ denotes the mean value of the variable Y.

In [12], we consider the relation shown in Table 1 [2]. Firstly, we assign the ranking values to the values of the attribute "Degree" (i.e., "Bachelor", "Master", and "Ph.D."). For example, we set the ranking value of "Bachelor" to 1, set the ranking value of "Master" to 2, and set the ranking value of "Ph.D." to 3. Because the attribute "Salary" is determined by the attributes "Degree" and "Experience", we calculate the coefficient of determination from the attribute "Degree" to the attribute "Salary" and from the attribute "Experience" to the attribute "Salary" using formula (4), respectively, where the results are 0.5376 and 0.6204, respectively. Then, we assign the coefficient of determination from "Salary" to "Salary" to 1. Then, after normalizing the values 0.5376, 0.6204 and 1, we can get the weights of the attributes "Degree", "Experience" and "Salary", respectively, which are 0.25, 0.29 and 0.46, respectively.

Table 1. A Relation in A Relational Database System [2]

Emp-ID	Degree	Experience	Salary
S1	Ph.D.	7.2	63000
S2	Master	2	37000
S3	Bachelor	7	40000
S4	Ph.D.	1.2	47000
S5	Master	7.5	53000
S6	Bachelor	1.5	26000
S7	Bachelor	2.3	29000
S8	Ph.D.	2	50000
S9	Ph.D.	3.8	54000
S10	Bachelor	3.5	35000
S11	Master	3.5	40000
S12	Master	3.6	41000
S13	Master	10	68000
S14	Ph.D.	5	57000
S15	Bachelor	5	36000
S16	Master	6.2	50000
S17	Bachelor	0.5	23000
S18	Master	7.2	55000
S19	Master	6.5	51000
S20	Ph.D.	7.8	65000
S21	Master	8.1	64000
S22	Ph.D.	8.5	70000

In [12], we let the domain of the attribute "Degree" be {Ph.D. (P), Master (M), Bachelor (B)}, and let the linguistic terms of the attributes "Experience" and "Salary" be $\{L_{Experience}, SL_{Experience}, M_{Experience}, SH_{Experience}, H_{Experience}\}$ and $\{L_{Salary}, SL_{Salary}, M_{Salary}, SH_{Salary}, H_{Salary}\}$, respectively, where the membership functions of the linguistic values of the attributes "Experience" and "Salary" are shown in Fig. 3, respectively.

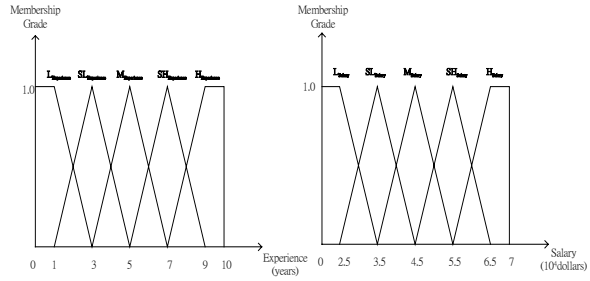

Fig. 3. Membership functions of the linguistic terms of the attributes "Experience" and "Salary".

Based on Fig. 3 and [15], we can fuzzify the relation shown in Table 1 into a fuzzy relation shown in Table 2.

Table 2. Fuzzified Relation of Table1 [2]

Emp-ID	Degree	Experience	Salary
S1	{Ph.D./1.0}	$\{SH_{Experience}/0.9\}$	$\{H_{Salary}/0.8\}$
S2	{Master/1.0}	$\{L_{Experience}/0.5\}$	$\{SL_{Salary}/0.8\}$
S3	{Bachelor/1.0}	$\{SH_{Experience}/1.0\}$	$\{SL_{Salary}/0.5\}$
S4	{Ph.D./1.0}	$\{L_{Experience}/0.9\}$	$\{M_{Salary}/0.8\}$
S5	{Master/1.0}	$\{SH_{Experience}/0.75\}$	$\{SH_{Salary}/0.8\}$
S6	{Bachelor/1.0}	$\{L_{Experience}/0.75\}$	$\{L_{Salary}/0.9\}$
S7	{Bachelor/1.0}	$\{SL_{Experience}/0.65\}$	$\{L_{Salary}/0.6\}$
S8	{Ph.D./1.0}	$\{L_{Experience}/0.5\}$	$\{M_{Salary}/0.5\}$
S9	{Ph.D./1.0}	$\{SL_{Experience}/0.6\}$	$\{SH_{Salary}/0.9\}$
S10	{Bachelor/1.0}	$\{SL_{Experience}/0.75\}$	$\{SL_{Salary}/1.0\}$
S11	{Master/1.0}	$\{SL_{Experience}/0.75\}$	$\{SL_{Salary}/0.5\}$
S12	{Master/1.0}	$\{SL_{Experience}/0.7\}$	$\{M_{Salary}/0.6\}$
S13	{Master/1.0}	$\{H_{Experience}/1.0\}$	$\{H_{Salary}/1.0\}$
S14	{Ph.D./1.0}	$\{M_{Experience}/1.0\}$	$\{SH_{Salary}/0.8\}$
S15	{Bachelor/1.0}	$\{M_{Experience}/1.0\}$	$\{SL_{Salary}/0.9\}$
S16	{Master/1.0}	$\{SH_{Experience}/0.6\}$	$\{M_{Salary}/0.5\}$
S17	{Bachelor/1.0}	$\{L_{Experience}/1.0\}$	$\{L_{Salary}/1.0\}$
S18	{Master/1.0}	$\{SH_{Experience}/0.9\}$	$\{SH_{Salary}/1.0\}$
S19	{Master/1.0}	$\{SH_{Experience}/0.75\}$	$\{SH_{Salary}/0.6\}$
S20	{Ph.D./1.0}	$\{SH_{Experience}/0.6\}$	$\{H_{Salary}/1.0\}$
S21	{Master/1.0}	$\{H_{Experience}/0.55\}$	$\{H_{Salary}/0.9\}$
S22	{Ph.D./1.0}	$\{H_{Experience}/0.75\}$	$\{H_{Salary}/1.0\}$

Then, we apply the fuzzy concept learning algorithm (FCLS) presented in [2] to sprout the fuzzy decision trees. The FCLS algorithm selects an attribute that has the smallest degree of fuzziness as a decision node. The definition of the degree of fuzziness of an attribute is reviewed from [2] as follows.

Definition 3.2: Let S be a set of antecedent attributes, S = {X, Y, …, W}, determining the consequent attribute Z. Let $t_j(X)$ be the value of the attribute X of the jth training instance i.e., (jth tuple of a relation), then the degree of fuzziness FA(X) of the attribute X is defined by

$$FA(X) = \frac{\sum_{j=1}^{c} (1 - \mu_{Xi}(t_j(X)))}{c}, \quad (5)$$

where c is the number of training instances.

The certainty factor value of each terminal node is calculated as follows.

Definition 3.3: Assume that there is a path in a fuzzy decision tree shown as follows.

$$\mathbf{D_1} \xrightarrow{F_1} \mathbf{D_2} \xrightarrow{F_2} \mathbf{D_3} \xrightarrow{F_3} (\mathrm{CF})$$

where D_1, D_2 and D_3 are attributes, and F_1, F_2 and F_3 are linguistic terms of the attributes D_1, D_2 and D_3, respectively. Then,

$$\mathrm{CF} = \mathrm{Avg}(F_1) \times \mathrm{weight1} + \mathrm{Avg}(F_2) \times \mathrm{weight2} + \mathrm{Avg}(F_3) \times \mathrm{weight3}, \quad (6)$$

where weight1, weight2 and weight3 are the weights of the attributes D_1, D_2 and D_3, respectively, $\mathrm{weight1} \in [0,1]$, $\mathrm{weight2} \in [0,1]$ and $\mathrm{weight3} \in [0,1]$; $\mathrm{Avg}(F_1)$, $\mathrm{Avg}(F_2)$ and $\mathrm{Avg}(F_3)$ are the average values of the linguistic terms F_1, F_2 and F_3 respectively, defined as follows:

$$\mathrm{Avg}(F_i) = \frac{\sum_{j=1}^{s} \mu_{Fi}(t_j(\mathbf{D}_i))}{s}, \quad (7)$$

where $t_j(D_i)$ denotes the values of the attribute D_i of the jth tuple of a relation, $\mu_{F_i}(t_j(D_i))$ denotes the grade of membership of the value of the attribute D_i of the jth tuple of the relation belonging to the linguistic term F_i, s is the number of training instances (i.e., the number of tuples in the relation), and $1 \le i \le 3$.

In [12], after applying the FCLS algorithm to generate fuzzy rules, we use the regression equations of the statistics [1], [13] to derive the values of the hypothetical certainty factor (HCF) nodes and to generate virtual fuzzy rules to make the rules complete.

Definition 3.4: Let

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2, \quad (8)$$

where Y is a dependent variable, X_1 and X_2 are independent variables, β_0, β_1 and β_2 are the regression coefficients, where β_0 denotes the distance between the intercept of the Y axis and the origin; β_1 and β_2 denote the average varying values of Y by varying the values of X_1 and X_2, respectively; the values of β_0, β_1 and β_2 are obtained by the following equations:

$$\begin{aligned} n\beta_0 + \beta_1 \sum_{i=1}^{n} X_{1i} + \beta_2 \sum_{i=1}^{n} X_{2i} &= \sum_{i=1}^{n} Y_i, \\ \beta_0 \sum_{i=1}^{n} X_{1i} + \beta_1 \sum_{i=1}^{n} X_{1i}^2 + \beta_2 \sum_{i=1}^{n} X_{1i}X_{2i} &= \sum_{i=1}^{n} X_{1i}Y_i, \\ \beta_0 \sum_{i=1}^{n} X_{2i} + \beta_1 \sum_{i=1}^{n} X_{1i}X_{21} + \beta_2 \sum_{i=1}^{n} X_{2i}^2 &= \sum_{i=1}^{n} X_{2i}Y_i, \end{aligned} \quad (9)$$

where X_{1i} denotes the ith data of the variable X_1, X_{2i} denotes the ith data of the variable X_2, Y_i denotes the ith data of the variable Y, $1 \le i \le n$, and n is the number of data.

Firstly, based on formula (8), formula (9) and Table 1, we can derive the regression equation of the attributes "Degree", "Experience" and "Salary" shown as follows:

$$\mathrm{Salary} = 10587.7 + 10147.06 \times \mathrm{Degree} + 3074.9 \times \mathrm{Experience}, \quad (10)$$

where $\beta_0 = 10587.7$, $\beta_1 = 10147.06$ and $\beta_2 = 3074.9$ are obtained by solving formula (9).

In [12], after generating fuzzy rules, we apply the generated fuzzy rules to estimate null values in relational database systems based on [2]. Let us consider the following fuzzy rules:

IF X is X_a and Y is Y_b THEN Z is Z_{M_1} (CF = C_1),
IF X is X_a and Y is Y_b THEN Z is Z_{M_2} (CF = C_2),
IF X is X_c and Y is Y_d THEN Z is Z_{N_1} (CF = D_1),
IF X is X_c and Y is Y_d THEN Z is Z_{N_2} (CF = D_2),

where X and Y are antecedent attributes of the fuzzy rules; Z is the consequent attribute of the fuzzy rules; X_a, X_c, Y_b, Y_d, Z_{M1}, Z_{M2}, Z_{N1} and Z_{N2} are linguistic terms represented by fuzzy sets; C_1, C_2, D_1 and D_2 are real values between zero and one. Assume that x and y are crisp domain values of the attributes X and Y in some tuples of a relational database, respectively, and assume that z is a null value of attribute Z. Let the fuzzified value of x be $\{X_a/\mu_{X_a}(x), X_b/\mu_{X_b}(x)\}$, and let the fuzzified value of y be $\{Y_a/\mu_{Y_a}(y), Y_b/\mu_{Y_b}(y)\}$. Then, the null value z can be calculated as follows:

$$z = \frac{\mu_{X_a}(x) \times \mu_{Y_c}(y) \times \frac{\sum_{i=1}^{2} C_i * DEF(Z_{M_i})}{\sum_{i=1}^{2} C_i} + \mu_{X_b}(x) \times \mu_{Y_d}(y) \times \frac{\sum_{i=1}^{2} D_i * DEF(Z_{N_i})}{\sum_{i=1}^{2} D_i}}{\mu_{X_a}(x) \times \mu_{Y_c}(y) + \mu_{X_b}(x) \times \mu_{Y_d}(y)}, \quad (11)$$

where $\mathrm{DEF}(Z_{M_i})$ and $\mathrm{DEF}(Z_{N_i})$ are the defuzzified values of the fuzzy sets Z_{M_i} and Z_{N_i}, respectively.

The estimated error is defended as follows:

$$\text{Estimated Error} = \frac{\text{Estimated Value} - \text{Original Value}}{\text{Original Value}}. \quad (12)$$

Assume that a relational database contains n tuples, and assume that the estimated error of tuple *i* is denoted as EER_i, where $1 \le i \le n$, then

$$\textit{Average Estimated Error} = \frac{EER_1 + EER_2 + \ldots + EER_n}{n}. \quad (13)$$

4. A NEW METHOD FOR ESTIMATING NULL VALUES IN RELATIONAL DATABASE SYSTEMS USING GENETIC ALGORITHMS

The theory of genetic algorithms was first proposed by Holland [8] in 1975. In a genetic algorithm, a chromosome consists of a lot of genes. A set of chromosomes in the same generation is called a population. A function calculating the compatible degree of each chromosome in the population is called the fitness function. A chromosome that has a higher fitness value is more adaptable to live. A genetic algorithm continuously executes the three operations (i.e., crossover, mutation and reproduction) to evolve a better population for the next generation. The genetic algorithm continues the evolution process until the result achieves the objective. Then, an optimal solution can be found to solve a problem.

In the following, we briefly review the three major operations of genetic algorithms from [5], [6] and [14] shown as follows:

(1) Crossover: The genetic algorithm randomly selects two chromosomes from a population. Then, the system exchanges the genes of two chromosomes after the crossover point. The system hopes to evolve better chromosomes which have higher fitness values by the crossover operations.

(2) Mutation: After performing the crossover operations, the genetic algorithm continues to perform the mutation operations. It randomly selects a chromosome from a population by using a parameter "MR" (i.e., Mutation Rate) to decide the mutation probability of a selected chromosome, where $\mathrm{MR} \in [0,1]$. The system randomly generates a value

between zero and one and compares it with the value of MR to decide whether the selected chromosome performs the mutation operation. If the value is smaller than or equal to the value of MR, then the system mutates the content of a randomly selected gene of a chromosome by assigning a random generated value to it.

(3) Reproduction: The genetic algorithm performs the reproduction operation to achieve the goal of "the survival of the fitter" according to the fitness value of each chromosome. The better chromosomes may have more chance to evolve the better next generation. In the reproduction process of a genetic algorithm, the system tries to select the better chromosomes into a mating pool. The genetic algorithm calculates the fitness values of each chromosome in a population by using a fitness function.

In the following, we introduce the format of a chromosome, the fitness function, the reproduction operations, the crossover operations, and the mutation operations of the genetic algorithm.

(1) The format of a chromosome: Based on Fig. 3, we define the format of a chromosome as shown in Fig. 4. There are ten genes in a chromosome, where the contents of the genes labeled "$L_{Experience}$", "$SL_{Experience}$", "$M_{Experience}$", "$SH_{Experience}$", and "$H_{Experience}$" are real values between 0 and 1.0; the content of the gene labeled "$L_{Experience}$" denotes the "right corner" of the membership function of the linguistic term "$L_{Experience}$" of the attribute "Experience"; the content of the gene labeled "$SL_{Experience}$" denotes the "center" of the membership function of the linguistic term "$SL_{Experience}$" of the attribute "Experience"; the content of the gene labeled "$M_{Experience}$" denotes the "center" of the membership function of the linguistic term "$M_{Experience}$" of the attribute "Experience"; the content of the gene labeled "$SH_{Experience}$" denotes the "center" of the membership function of the linguistic term "$SH_{Experience}$" of the attribute "Experience"; the content of the gene labeled "$H_{Experience}$" denotes the "left corner" of the membership function of the linguistic term "$SH_{Experience}$" of the attribute "Experience". The contents of the genes labeled "L_{Salary}", "SL_{Salary}", "M_{Salary}", "SH_{Salary}" and "H_{Salary}" are integer values between 0 and 70000; the content of the gene labeled "L_{Salary}" denotes the "right corner" of the membership function of the linguistic term "L_{Salary}" of the attribute "Salary"; the content of the gene labeled "SL_{Salary}" denotes the "center" of the membership function of the linguistic term "SL_{Salary}" of the attribute "Salary"; the content of the gene labeled "M_{Salary}" denotes the "center" of the membership function of the linguistic term "M_{Salary}" of the attribute "Salary"; the content of the gene labeled "SH_{Salary}" denotes the "center" of the membership function of the linguistic term "SH_{Salary}" of the attribute "Salary"; the content of the gene labeled "H_{Salary}" denotes the "left corner" of the membership function of the linguistic term "SH_{Salary}" of the attribute "Salary". For example, let us consider the membership functions of the linguistic terms of the attributes "Experience" and "Salary" shown in Fig. 3. The corresponding chromosome of the membership functions of the linguistic terms of the attributes "Experience" and "Salary" of Fig. 3 is shown in Fig. 5.

(2) Fitness Function: Because a chromosome represents the membership functions of the linguistic values of the attributes" Experience" and "Salary", according to [12], we can fuzzify the relation shown in Table 1 to generate fuzzy rules and estimate null values by the generated fuzzy rules. The fitness value of the fitness function is defined as follows:

Fitness Value = 1 – Average Estimated Error. (14)

The larger the fitness value, the better the chromosome.

(3) Reproduction: The tournament selection method [5], [14] is used during the reproduction process. In the tournament selection method, the system randomly selects N chromosomes from the current population. The system chooses the chromosome which has the largest fitness value and puts it into the mating pool. In this paper, we set N to 2 and the system randomly chooses two chromosomes from the current population to compare them. The chromosome with the largest fitness value is chosen. The process is repeated until the number of the chromosomes in a population is full.

(4) Crossover: We use the single-point crossover method [6] to perform the crossover operations. The system randomly generates a crossover point from a chromosome and randomly selects two chromosomes from a population. Then, the system considers whether to exchange the genes of these two selected chromosomes after the crossover point.

Case 1: The crossover point occurs at the left-half of the selected chromosome as shown in Fig. 6. If "SH1"≤"H2" and "SH2"≤"H1", then the system performs the crossover operations as shown in Fig. 6. Otherwise, the system does not perform the crossover operations.

Case 2: The crossover point occurs at the middle of the selected chromosome as shown in Fig. 7. In this case, the system performs the crossover operations as shown in Fig. 7.

Case 3: The crossover point occurs at the right-half of the selected chromosome as shown in Fig. 8. If "m1"≤"sh2" and "m2"≤"sh1", then the system performs the crossover operations as shown in Fig. 8. Otherwise, the system does not perform the crossover operations.

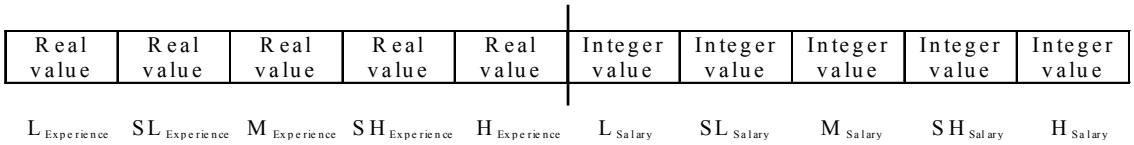

Fig. 4. The format of a chromosome.

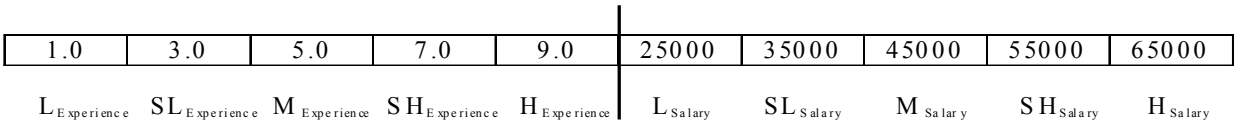

Fig. 5. A chromosome corresponding to the membership functions of the linguistic terms shown in Fig. 3.

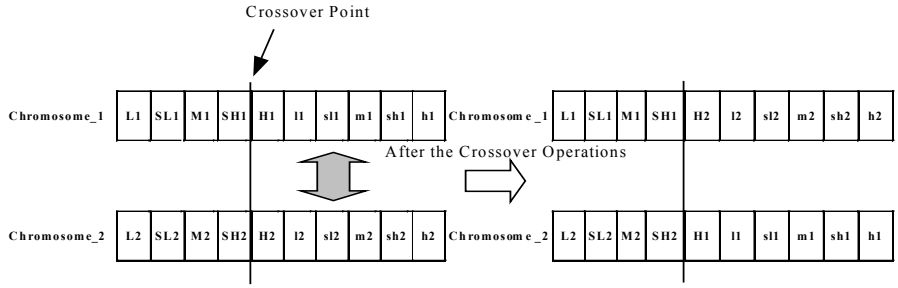

Fig. 6. The single-point crossover operations (The crossover point occurs at the left-half of the selected chromosomes).

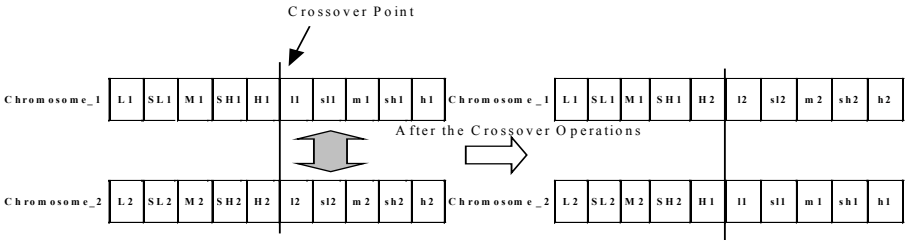

Fig. 7. The single-point crossover operations (The crossover point occurs at the middle of the selected chromosomes).

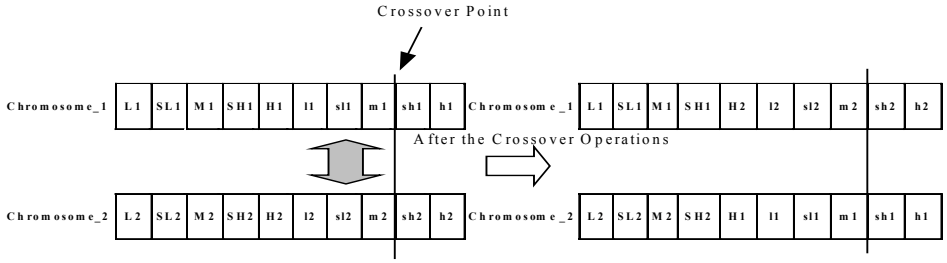

Fig. 8. The single-point crossover operations (The crossover point occurs at the right-half of the selected chromosomes).

(5) Mutation: The system randomly selects a chromosome from a population and randomly determines a mutation point. Then, it randomly generates a value between 0 and 1 and compares it with the value of MR (Mutation Rate) to decide whether the selected chromosome performs the mutation operation. In this paper, we set the mutation rate to 0.05. If the generated value is smaller than or equal to the value of MR (i.e., 0.05), then the system mutates the selected gene of the selected chromosome by randomly generating a value to replace the content of the selected gene of the selected chromosome as shown in Fig. 9.

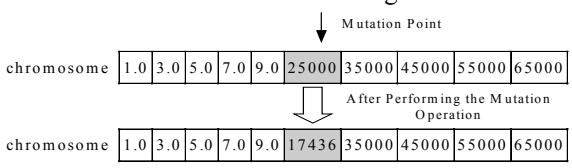

Fig. 9. The mutation operation of a chromosome.

Initially, we must decide the number of evolution times, the population size, and the mutation rate. In this paper, we set the number of evolution time = 300, the number of chromosomes in each population = 30, and the mutation rate = 0.05.The proposed algorithm for tuning the membership functions of the linguistic terms of the attributes for estimating null values in relational database systems is now presented as follows:

Step 1: Generate an initial population, where each chromosome in the initial population represents the membership functions of the linguistic terms of the attributes "Experience" and " Salary".

Step 2: **For** each chromosome in the population **do**
begin
based on the chromosome, apply the FCLS algorithm [2] to generate fuzzy rules from training instances and calculate the certainty factor values of the fuzzy rules using formula (5), formula (6) and formula (7);
based on formula (6), formula (7), formula (8), formula (9) and formula (10), derive the values of the hypothetical certainly factor nodes and

virtual fuzzy rules;
apply formula (11) to estimate the value of the attribute "Salary" of the tuple; get the average estimated error using formula (13);
calculate the fitness value of the chromosome using formula (14)

end.

Step 3: Perform the reproduction operations.

Step 4: Perform the crossover operations.

Step 5: Perform the mutation operation according to the mutation rate.

Step 6: If the evolution times reach the predefined number of evolutions, then go to **Step 7**. Otherwise, go to **Step 2**.

Step 7: Choose the chromosome which has the largest fitness value from the population. This chromosome is used to represent the membership functions of the linguistic terms of the attributes "Experience" and "Salary" for estimating null values in relational database systems.

5. EXPERIMENTAL RESULTS

We have implemented the proposed method on a Pentium III PC by using Borland JBuilder Version 5.0. A comparison of the average estimated error rate of the proposed method with the ones we presented in [2] and [12] is shown in Table 3.

Table 3. A Comparison of the Average Estimated Error Rate of the Proposed Method with the Existing Methods

Emp-ID	Degree	Experience	Salary	Chen-and-Yeh's Method [2]		Lee-and-Chen's Method [12]		The Proposed Method (Number of Evolutions = 300, Number of Chromosomes in Each Population = 30,and Mutation Rate = 0.05)	
				Salary (Estimated)	Estimated Error Rate	Salary (Estimated)	Estimated Error Rate	Salary (Estimated)	Estimated Error Rate
S1	Ph.D.	7.2	63000	65000	+3.17%	65000	+3.17%	65249	+3.57%
S2	Master	2	37000	30704	-7.02%	37587	+1.59%	38108	+2.99%
S3	Bachelor	7	40000	35000	-12.50%	35000	-12.50%	38010	-4.98%
S4	Ph.D.	1.2	47000	46000	-2.13%	46000	-2.13%	46945	-0.12%
S5	Master	7.5	53000	54500	+2.83%	54090	+2.06%	56863	+7.29%
S6	Bachelor	1.5	26000	26346	+1.33%	26387	+1.49%	26046	+0.18%
S7	Bachelor	2.3	29000	28500	-1.72%	28606	-1.36%	28760	-0.83%
S8	Ph.D.	2	50000	50000	+0.00%	50000	+0.00%	49330	-1.34%
S9	Ph.D.	3.8	54000	55000	+1.85%	55000	+1.85%	55009	+1.87%
S10	Bachelor	3.5	35000	31538	-9.89%	31661	-9.54%	32648	-6.72%
S11	Master	3.5	40000	41590	+3.98%	41381	+3.45%	41548	+3.87%
S12	Master	3.6	41000	45159	+10.14%	41622	+1.52%	41726	+1.77%
S13	Master	10	68000	65000	-4.41%	65000	-4.41%	65249	-4.05%
S14	Ph.D.	5	57000	55000	-3.51%	55000	-3.51%	57159	+0.28%
S15	Bachelor	5	36000	35000	-2.78%	35000	-2.78%	36130	+0.36%
S16	Master	6.2	50000	48600	-2.80%	48272	-3.46%	49304	-1.39%
S17	Bachelor	0.5	23000	25000	+8.70%	25000	+8.70%	22891	-0.47%
S18	Master	7.2	55000	52400	-4.73%	51909	-5.62%	55034	+0.06%
S19	Master	6.5	51000	49500	-2.94%	49090	-3.75%	50613	-0.76%
S20	Ph.D.	7.8	65000	65000	+0.00%	65000	+0.00%	65249	+0.38%
S21	Master	8.1	64000	58700	-8.28%	58454	-8.67%	58236	-9.01%
S22	Ph.D.	8.5	70000	65000	-7.14%	65000	-7.14%	65249	-6.79%
Average Estimated Error Rate				5.08%		4.03%		2.81%	

6. CONCLUSIONS

We have presented a new method to estimate null values in relational database systems based on genetic algorithms. The proposed method can tune membership functions of the linguistic terms of the attributes in relational database systems for estimating null values. The proposed method can get a higher average estimated accuracy rate than the existing methods for estimating null values in relational database systems.

REFERENCES

[1] M. L. Berenson, D. M. Levine, and M. Goldstein, *Intermediate Statistical Methods and Applications*. New Jersey: Prentice-Hall, 1983.

[2] S. M. Chen and M. S. Yeh, "Generating fuzzy rules from relational database systems for estimating null values," *Cybernetics and Systems: An International Journal*, vol. 28, no. 8, pp. 695-723, 1997.

[3] S. M. Chen and H. H. Chen, "Estimating null values in the distributed relational databases environment," *Cybernetics and Systems: An International Journal*, vol. 31, no. 8, pp. 851-871, 2000.

[4] S. M. Chen, "Using fuzzy reasoning techniques for fault diagnosis of the J-85 jet engines," *Proceedings of the Third National Conference on Science and Technology of National Defense*, Taoyuan, Taiwan, Republic of China, vol. 1, pp. 29-34, 1994.

[5] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Design*. New York: John Wiley & Sons, 1997.

[6] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. MA: Addison-Wesley, 1989.

[7] H. R. Hsiao and S. M. Chen, "A new method to estimate null values in relational database systems," *Proceedings of the 13th International Conference on Information Management*, Taipei, Taiwan, Republic of China, vol.2, pp. 413-420, 2002.

[8] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Cambridge, MA: MIT Press, 1975.

[9] C. M. Huang and S. M. Chen, "A new method to estimate null values in relational database systems using genetic algorithms," *Proceedings of the 6th Conference on Artificial Intelligence and Applications*, Kaohsiung, Taiwan, Republic of China, pp. 599-604, 2001.

[10] C. M. Huang and S. M. Chen, "Estimating null values in relational database systems with a negative dependency relationship between attributes," *Proceedings of the 13th International Conference on Information Management*, Taipei, Taiwan, Republic of China, vol. 1, pp. 151-158, 2002.

[11] A. Kandel, *Fuzzy Mathematical Techniques with Applications*. Massachusetts: Addison-Wesly, 1986.

[12] S. W. Lee and S. M. Chen, "A new method to generate fuzzy rules from relational database systems," *Proceedings of the 2001 Ninth National Conference on Fuzzy Theory and Applications*, Chungli, Taoyuan, Taiwan, Repulic of China, pp. 702-707, 2001.

[13] W. Mendenhall and R. J. Beaver, *Introduction to Probability and Statistics*. Belmont, CA: Wadsworth, 1994.

[14] Z. Michalewicz, *Genetic Algorithms + Data Structure = Evolution Programs*, Springer, Berlin, 1992.

[15] L. X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 6, pp. 1414-1427, 1992.

[16] L. A. Zadeh, " Fuzzy sets," *Information and Control*, vol. 8, pp. 338-353, 1965.