Generating Weighted Fuzzy Rules From Relational Database Systems for Estimating Null Values Using Genetic Algorithms

Shyi-Ming Chen, *Senior Member, IEEE,* and Chung-Ming Huang

***Abstract*—In recent years, some methods have been proposed to estimate null values in relational database systems. However, the estimated accuracy of the existing methods are not good enough. In this paper, we present a new method to generate weighted fuzzy rules from relational database systems for estimating null values using genetic algorithms (GAs), where the attributes appearing in the antecedent part of generated fuzzy rules have different weights. After a predefined number of evolutions of the GA, the best chromosome contains the optimal weights of the attributes, and they can be translated into a set of rules to be used for estimating null values. The proposed method can get a higher average estimated accuracy rate than the methods we presented in two previous papers.**

***Index Terms*—Fuzzy sets, genetic algorithms (GAs), membership functions, null values, relational database systems, weighted fuzzy rules.**

I. Introduction

It IS obvious that data mining is one of the important research topics of artificial intelligence. With data mining techniques, we can extract useful knowledge from training data to solve problems. Many methods have been proposed to generate fuzzy rules from training instances [3], [5], [6], [15], [17], [18] based on the fuzzy set theory [19]. In [6], we have presented a method for generating fuzzy rules from relational database systems for estimating null values. However, the average estimated accuracy rate of the method we presented in [6] is not good enough. It is necessary to develop a more powerful method to estimate null values in relational database systems.

In [4], we have presented a method to estimate null values in the distributed relational databases environment. However, there is a drawback in the method we presented in [4], i.e., the rules used in [4] are given by experts and are not automatically generated by the system. Furthermore, the average estimated accuracy rate of the method we presented in [4] is also not good enough.

Manuscript received September 5, 2002; revised November 19, 2002 and December 16, 2002. This work was supported in part by the National Science Council, R.O.C., under Grant NSC 90-2213-E-011-053.

S.-M. Chen is with the Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei 106, Taiwan, R.O.C. (e-mail: smchen@et.ntust.edu.tw).

C.-M. Huang is with the Department of Electronic Engineering, National Taiwan University of Science and Technology, Taipei 106, Taiwan, R.O.C.

Digital Object Identifier 10.1109/TFUZZ.2003.814837

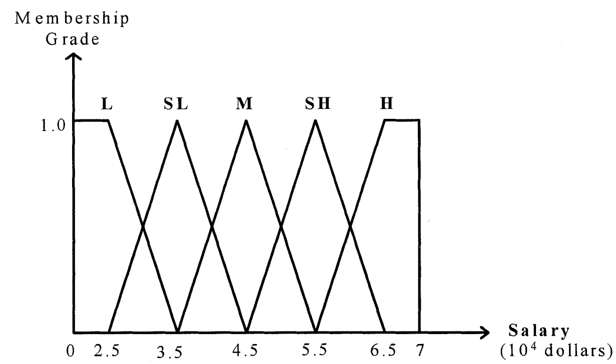

Fig. 1. Membership functions of the linguistic terms of the attribute “Salary.”

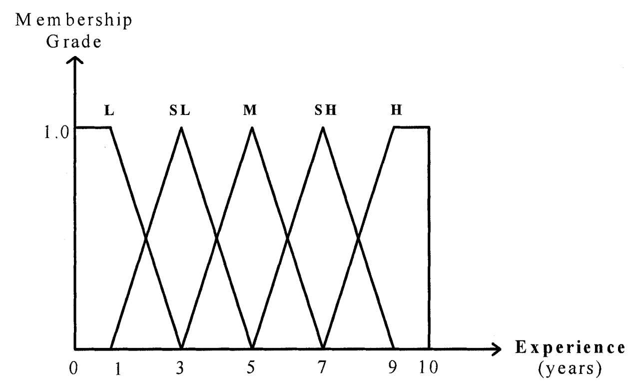

Fig. 2. Membership functions of the linguistic terms of the attribute “Experience.”

	v_1	v_2	$\cdots$	v_n
v_1	1	u_{12}	$\cdots$	u_{1n}
v_2	u_{21}	1	$\cdots$	u_{2n}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
v_n	u_{n1}	u_{n2}	$\cdots$	1

Fig. 3. Fuzzy relation matrix of the linguistic variable V.

In this paper, in order to overcome the drawback of [4] and [6], we improve the methods we presented in [4] and [6] and propose a new method to estimate null values in relational database systems using genetic algorithms (GAs) [8], [10], [16], where the attributes appearing in the antecedent parts of the generated fuzzy rules have different weights. The proposed method uses GAs to adjust the weights of the attributes for estimating null

TABLE I
RULE BASE CONTAINS WEIGHTED FUZZY RULES

Rule 1:	IF $A_1 = a_{11}$ ($W = w_{11}$)	AND $A_2 = a_{12}$ ($W = w_{12}$)	AND …	AND $A_n = a_{1n}$ ($W = w_{1n}$)	THEN $B_1 = t_1$
Rule 2:	IF $A_1 = a_{21}$ ($W = w_{21}$)	AND $A_2 = a_{22}$ ($W = w_{22}$)	AND …	AND $A_n = a_{2n}$ ($W = w_{2n}$)	THEN $B_2 = t_2$
⋮				⋮	
Rule m:	IF $A_1 = a_{m1}$ ($W = w_{m1}$)	AND $A_2 = a_{m2}$ ($W = w_{m2}$)	AND …	AND $A_n = a_{mn}$ ($W = w_{mn}$)	THEN $B_m = t_m$

values in relational database systems. It can get a higher average estimated accuracy rate than the methods we presented in [4] and [6].

The rest of this paper is organized as follows. In Section II, we briefly review the basic concepts of fuzzy sets from [19]. In Section III, we briefly review a method for estimating null values in database systems from [4]. In Section IV, we present a method to tune the weights of the attributes appearing in the antecedent parts of the generated fuzzy rules using GAs for estimating null values in relational database systems. In Section V, we present an example to illustrate the process of estimating null values in relational database systems. The conclusions are discussed in Section VI.

II. Basic Concepts of Fuzzy Sets

In 1965, Zadeh proposed the theory of fuzzy sets [19]. In a fuzzy set, each element in the set is associated with a membership value between 0 and 1 described by a membership function to indicate the grade of membership of the element in the fuzzy set. There are two types of membership functions to represent fuzzy sets. One is the discrete type membership function, and the other is the continuous type membership function. Let U be the universe of discourse, $U = \{u_1, u_2, \ldots, u_n\}$. A fuzzy subset A of the universe of discourse U can be represented as follows:

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

where μ_A is the membership function of the fuzzy subset A, μ_A: $U \rightarrow [0, 1]$, and $\mu_A(u_i)$ indicates the grade of membership of u_i in the fuzzy subset A. If the universe of discourse U is a continuous set, then the fuzzy subset A can be represented as follows:

$$A = \int_U \mu_A(u) \Big/ u, \qquad u \in U. \quad (2)$$

A linguistic term can be represented by a fuzzy set represented by a membership function. In this paper, the membership functions of the linguistic terms "L," "SL," "M." "SH," and "H" of the attributes "Salary" and "Experience" in a relational database system are adopted from [6] as shown in Fig. 1 and Fig. 2, respectively, where "L" denotes "Low," "SL" denotes "Somewhat Low," "M" denotes "Medium," "SH" denotes "Somewhat High," and "H" denotes "High." Assume that there is an employee whose salary is 48 000 dollars per month, and assume that his working experience is 4 y, then according to Fig. 1, we can see that the degree of membership that his salary (i.e., 48 000 dollars) belonging to the linguistic terms "Medium" (M) and "Somewhat High" (SH) are 0.7 and 0.3, respectively. According to Fig. 2, we can see that the degrees of membership that his experience (i.e., 4 y) belonging to the linguistic terms of "Somewhat Low" (SL) and "Medium" (M) are 0.5 and 0.5, respectively. Let x be a real value, and let X and Y be two linguistic terms. Assume that the degrees of membership of x belonging to X and Y are a and b, respectively, where $a \in [0, 1]$ and $b \in [0, 1]$. Based on [6] and [18], if $a \geq b$, then the real value x is fuzzified into X/a. Thus, in the previous example, the employee's salary (i.e., 48 000 dollars) is fuzzified into M/0.7, and his experience (i.e., 4 years) is fuzzified into SL/0.5. If p is a nonnumeric datum, where $p \in \{\text{Bachelor, Master, Ph.D.}\}$, then according to [6], p is fuzzified into p/1.0, where $p \in \{\text{Bachelor, Master, Ph.D.}\}$.

III. Review of Chen-and-Chen's Method for Estimating Null Values in Relational Database Systems

In this section, we briefly review the method we presented in [4] for estimating null values in relational database systems. First, we can use fuzzy similarity matrices to represent fuzzy relations. Assume that a linguistic variable V has linguistic terms $v_1, v_2, \ldots$, and v_n, and assume that its fuzzy similarity matrix is shown in Fig. 3, where u_{ij} denotes the degree of similarity between v_i and v_j. The closeness degree between v_i and v_j of the linguistic variable V is denoted as $CD_v(v_i, v_j)$, where $CD_v(v_i, v_j) = u_{ij}$, $1 \leq i \leq n$, and $1 \leq j \leq n$. It is obvious that $CD_v(v_i, v_i) = 1$, where $1 \leq i \leq n$. It should be noted that each element in a fuzzy relation matrix is defined by a domain expert.

Furthermore, the expert also defines a ranking function for linguistic terms in order to rank linguistic terms. Assume that a linguistic variable V contains linguistic terms v_i and v_j, and assume that the rank of linguistic term v_i is prior to the rank of linguistic term v_j, then the ranking order between v_i and v_j can be defined as follows:

$$\text{Rank}(v_i) > \text{Rank}(v_j)$$

where $1 \leq i \leq n$, $1 \leq j \leq n$, and $i \neq j$.

A. Rule Base

A rule base is used to indicate relationships in which some attributes determine other attributes. For example, Table I shows a set of rules including the weights of the attributes, where all rules in the rule base are given by experts, w_{ij} denotes the weight of attribute A_j of the ith rule in the rule base, $w_{ij} \in [0, 1]$, $1 \leq i \leq n$, and $1 \leq j \leq n$.

B. Estimating Null Values

The basic idea of the method we presented in [4] is to see which rule appearing in the rule base shown in Table I is closest to the tuple having a null value. The null value can be estimated

Gene Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Real value	Real value	Real value	Real value	Real value	Real value	Real value	Real value	Real value	Real value	Real value	Real value	Real value	Real value	Real value
	B-H	**M-H**	**P-H**	**B-SH**	**M-SH**	**P-SH**	**B-M**	**M-M**	**P-M**	**B-SL**	**M-SL**	**P-SL**	**B-L**	**M-L**	**P-L**

Fig. 4. Format of a chromosome.

TABLE II
RELATION IN A RELATIONAL DATABASE [4], [6]

EMP-ID	Degree	Experience	Salary
S1	Ph.D.	7.2	63,000
S2	Master	2.0	37,000
S3	Bachelor	7.0	40,000
S4	Ph.D.	1.2	47,000
S5	Master	7.5	53,000
S6	Bachelor	1.5	26,000
S7	Bachelor	2.3	29,000
S8	Ph.D.	2.0	50,000
S9	Ph.D.	3.8	54,000
S10	Bachelor	3.5	35,000
S11	Master	3.5	40,000
S12	Master	3.6	41,000
S13	Master	10.0	68,000
S14	Ph.D.	5.0	57,000
S15	Bachelor	5.0	36,000
S16	Master	6.2	50,000
S17	Bachelor	0.5	23,000
S18	Master	7.2	55,000
S19	Master	6.5	51,000
S20	Ph.D.	7.8	65,000
S21	Master	8.1	64,000
S22	Ph.D.	8.5	70,000

by the closeness degree of the tuple with respect to the closest rule.

Assume that there is a relation in a relational database system having attributes A_1, A_2, …, A_n and B, and assume that the attributes A_1, A_2, …, and A_n determine the attribute B. Let "$r_j.A_k$" denote the value of attribute A_k appearing in the antecedent portion of rule r_j and let "$T_i.A_k$" denote the value of attribute A_k of tuple T_i. If there is a null value in the attribute B, then the null value can be estimated as follows.

Case 1: If the attribute B is defined in a numerical domain, and there is a rule r_j, where $1 \le j \le m$, in the rule base shown as follows:

$$\begin{aligned}&\textbf{IF}\ A_1 = a_{j1}(W = w_{j1})\ \textbf{AND}\ A_2 = a_{j2}(W = w_{j2})\\ &\quad \textbf{AND}\ \cdots\ \textbf{AND}\ A_n = a_{jn}(W = w_{jn})\\ &\textbf{THEN}\ B = N_j\end{aligned}$$

where $1 \le j \le m$. Since attribute A_k can be defined either in a numerical or a nonnumerical domain, where $1 \le k \le n$, there are two cases to be considered to calculate the closeness degree between tuple T_i and rule r_j.

1) A_k is defined in a numerical domain: The closeness degree $CDv_{ji}(r_j.A_k, T_i.A_k)$ between $r_j.A_k$ and $T_i.A_k$ can be calculated as follows:

$$CDv_{ji}(r_j.A_k,\, T_i.A_k) = \frac{T_i.A_k}{r_j.A_k}. \tag{3}$$

2) A_k is defined in a nonnumerical domain: First, check the ranking order between $T_i.A_k$ and $r_j.A_k$. Then, the closeness degree $CDv_{ji}(r_j.A_k, T_i.A_k)$ between $r_j.A_k$ and $T_i.A_k$ can be calculated as follows:

$$CDv_{ji}(r_j.A_k,\, T_i.A_k) = \begin{cases}\dfrac{1}{R[r_j.A_k, T_i.A_k]}, & \text{if } \mathrm{Rank}(T_i.A_k) > \mathrm{Rank}(r_j.A_k)\\ R[r_j.A_k,\, T_i.A_k], & \text{if } \mathrm{Rank}(T_i.A_k) \le \mathrm{Rank}(r_j.A_k)\end{cases} \tag{4}$$

where R is a fuzzy similarity matrix and $R[r_j.A_k,\, T_i.A_k]$ denotes the degree of similarity

TABLE III
FUZZIFIED RESULTS OF THE VALUES OF THE ATTRIBUTES "DEGREE" AND "EXPERIENCE"

EMP-ID	Degree	Experience	Salary
S1	Ph.D./1.0	SH/0.9	63,000
S2	Master/1.0	L/0.5	37,000
S3	Bachelor/1.0	SH/1.0	40,000
S4	Ph.D./1.0	L/0.9	47,000
S5	Master/1.0	SH/0.75	53,000
S6	Bachelor/1.0	L/0.75	26,000
S7	Bachelor/1.0	SL/0.65	29,000
S8	Ph.D./1.0	L/0.5	50,000
S9	Ph.D./1.0	SL/0.6	54,000
S10	Bachelor/1.0	SL/0.75	35,000
S11	Master/1.0	SL/0.75	40,000
S12	Master/1.0	SL/0.7	41,000
S13	Master/1.0	H/1.0	68,000
S14	Ph.D./1.0	M/1.0	57,000
S15	Bachelor/1.0	M/1.0	36,000
S16	Master/1.0	SH/0.6	50,000
S17	Bachelor/1.0	L/1.0	23,000
S18	Master/1.0	SH/0.9	55,000
S19	Master/1.0	SH/0.75	51,000
S20	Ph.D./1.0	SH/0.6	65,000
S21	Master/1.0	H/0.55	64,000
S22	Ph.D./1.0	H/0.75	70,000

between $\mathrm{r_j.A_k}$ and $\mathrm{T_i.A_k}$. The closeness $\mathrm{CD(r_j, T_i)}$ degree between tuple $\mathrm{T_i}$ and rule $\mathrm{r_j}$ can be calculated as follows:

$$CD(r_j, T_i) = \sum_{k=1}^{n} CDv_{ji}(r_j.A_k, T_i.A_k) \times w_{jk} \tag{5}$$

where $\mathrm{w_{jk}}$ denotes the weight of attribute $\mathrm{A_k}$ of rule $\mathrm{r_j}$, $1 \leq \mathrm{j} \leq \mathrm{m}$, and $1 \leq \mathrm{k} \leq \mathrm{n}$. After the closeness degree between tuple $\mathrm{T_i}$ and each rule $\mathrm{r_j}$ in the rule base has been calculated, where $1 \leq \mathrm{j} \leq \mathrm{m}$, the system will choose the rule whose closeness degree $\mathrm{CDv_{ji}(r_j.A_k, T_i.A_k)}$ with respect to tuple $\mathrm{T_i}$ is closest to 1.0. Assume that rule $\mathrm{r_j}$ whose closeness degree with respect to tuple $\mathrm{T_i}$ is closest to 1.0, then the null value of attribute B of tuple $\mathrm{T_i}$ can be estimated as follows:

$$T_i.B = CD(r_j, T_i) \times N_j. \tag{6}$$

Case 2: If the attribute B is defined in a nonnumerical domain, and there is a rule $\mathrm{r_j}$ in the knowledge base shown as follows:

$$\begin{aligned}&\textbf{IF}\, \mathrm{A_1} = a_{j1}(\mathrm{W} = \mathrm{w_{j1}})\, \textbf{AND}\, \mathrm{A_2} = \mathrm{a_{j2}}(\mathrm{W} = \mathrm{w_{j2}})\\ &\quad \textbf{AND} \ldots \textbf{AND}\, \mathrm{A_n} = \mathrm{a_{jn}}(\mathrm{W} = \mathrm{w_{jn}})\\ &\textbf{THEN}\, \mathrm{B} = \mathrm{W_j}\end{aligned}$$

where $1 \leq \mathrm{j} \leq \mathrm{m}$. Because attribute $\mathrm{A_k}$ can be defined either in a numerical or a nonnumerical domain, where $1 \leq \mathrm{k} \leq \mathrm{n}$, there are two cases to be considered to calculate the closeness degree between tuple $\mathrm{T_i}$ and rule $\mathrm{r_j}$.

1) $\mathrm{A_k}$ is defined in a numerical domain: The closeness degree $\mathrm{CDv_{ji}(r_j.A_k, T_i.A_k)}$ between $\mathrm{r_j.A_k}$ and $\mathrm{T_i.A_k}$ can be calculated by formula (3).
2) $\mathrm{A_k}$ is defined in a nonnumerical domain.
   First, check the ranking order between $\mathrm{T_i.A_k}$ and $\mathrm{r_j.A_k}$, respectively. Then, the closeness degree $\mathrm{CDv_{ji}(r_j.A_k, T_i.A_k)}$ between $\mathrm{r_j.A_k}$ and $\mathrm{T_i.A_k}$ can be calculated by formula (4). The closeness degree $\mathrm{CD(r_j, T_i)}$ between tuple $\mathrm{T_i}$ and rule $\mathrm{r_j}$ can be calculated by formula (5), where $\mathrm{w_{jk}}$ denotes the weight of attribute $\mathrm{A_k}$ of rule $\mathrm{r_j}$, $1 \leq \mathrm{j} \leq \mathrm{m}$, and $1 \leq \mathrm{k} \leq \mathrm{n}$. After the closeness degree between tuple $\mathrm{T_i}$ and each rule $\mathrm{r_j}$ in the rule base has been calculated, where $1 \leq \mathrm{j} \leq \mathrm{m}$, the system will choose the rule whose closeness degree with respect to tuple $\mathrm{T_i}$ is closest to 1.0. Assume that rule $\mathrm{r_j}$ whose closeness degree with respect to tuple $\mathrm{T_i}$ is closest to 1.0, then the null value of attribute B of tuple $\mathrm{T_i}$ can be estimated as follows:

$$T_i.B = W_j. \tag{7}$$

For more details, please refer to [4].

Gene Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0.01	0.071	0.343	0.465	0.505	0.303	0.495	0.081	0.778	0.717	0.303	0.869	0.869	0.828	0.434
	B-H	**M-H**	**P-H**	**B-SH**	**M-SH**	**P-SH**	**B-M**	**M-M**	**P-M**	**B-SL**	**M-SL**	**P-SL**	**B-L**	**M-L**	**P-L**

Fig. 5. Example of a chromosome.

IV. Tuning the Weights of the Attributes Using GAs

In this section, we present a method to tune the weights of attributes using GAs for estimating null values in relational database systems.

A. Format of a Chromosome

Let us consider a relation of a relational database shown in Table II [4], [6]. Based on Figs. 1 and 2, the values of attributes "Degree" and "Experience" shown in Table II can be fuzzified into in Table III. First, we define the format of a chromosome as shown in Fig. 4, where the value of each gene in a chromosome is a real value between zero and one, and the 13th gene labeled "B-L" shown in Fig. 4 denotes the fuzzified values of the attributes "Degree" and "Experience" are "Bachelor" (B) and "Low" (L), respectively (e.g., the tuples whose EMP-ID are S6 and S17 as shown in Table III).

From Fig. 4, we can see that each chromosome represents a combination of the weights of attributes, and it is a string of the weights of the attributes which will be used to estimate null values in relational database systems. A population contains a set of chromosomes, and we can arbitrary set the number of chromosomes in a population. In this paper, we let a chromosome consist of 15 genes. Because the total weights of attributes must be equal to one, the weight of attribute "Experience" must equal to one minus the weight of attribute "Degree." For example, assume that there is a chromosome as shown in Fig. 5. Assume that we want to estimate the null value of the attribute "Salary" of a tuple whose fuzzified values of the attributes "Degree" and "Experience" are "Ph.D."(P) and "Somewhat Low"(SL), then from Fig. 5 we can see that the value of the gene labeled "P-SL" is 0.869. It means that if we want to estimate the null value of the attribute "Salary" of this tuple, where the values of the attributes "Degree" and "Experience" of the tuple are "Ph.D."(P) and "Somewhat Low"(SL), we will let the weights of the attributes "Degree" and "Experience" of the tuple be equal to 0.869 and 0.131 (i.e., $1 - 0.869 = 0.131$), respectively, to calculate the degrees of closeness between the tuple which contains the null value and other tuples in the database, respectively. Therefore, the contents of the chromosome shown in Fig. 5 can be translated into the following 15 rules.

Rule 1: **IF** Degree = Bachelor **AND** Experience = High, **THEN** the Weight of Degree = 0.010 **AND** the Weight of Experience = 0.99.

Rule 2: **IF** Degree = Master **AND** Experience = High, **THEN** the Weight of Degree = 0.071 **AND** the Weight of Experience = 0.929.

Rule 3: **IF** Degree = Ph.D. **AND** Experience = High, **THEN** the Weight of Degree = 0.343 **AND** the Weight of Experience = 0.657.

Rule 4: **IF** Degree = Bachelor **AND** Experience = Somewhat High, **THEN** the Weight of Degree = 0.465 **AND** the Weight of Experience = 0.535.

Rule 5: **IF** Degree = Master **AND** Experience = Somewhat High, **THEN** the Weight of Degree = 0.505 **AND** the Weight of Experience = 0.495.

Rule 6: **IF** Degree = Ph.D. **AND** Experience = Somewhat High, **THEN** the Weight of Degree = 0.303 **AND** the Weight of Experience = 0.697.

Rule 7: **IF** Degree = Bachelor **AND** Experience = Medium, **THEN** the Weight of Degree = 0.495 **AND** the Weight of Experience = 0.505.

Rule 8: **IF** Degree = Master **AND** Experience = Medium, **THEN** the Weight of Degree = 0.081 **AND** the Weight of Experience = 0.919.

Rule 9: **IF** Degree = Ph.D. **AND** Experience = Medium, **THEN** the Weight of Degree = 0.778 **AND** the Weight of Experience = 0.222.

Rule 10: **IF** Degree = Bachelor **AND** Experience = Somewhat Low, **THEN** the Weight of Degree = 0.717 **AND** the Weight of Experience = 0.283.

Rule 11: **IF** Degree = Master **AND** Experience = Somewhat Low, **THEN** the Weight of Degree = 0.303 **AND** the Weight of Experience = 0.697.

Rule 12: **IF** Degree = Ph.D. **AND** Experience = Somewhat Low, **THEN** the Weight of Degree = 0.869 **AND** the Weight of Experience = 0.131.

Rule 13: **IF** Degree = Bachelor **AND** Experience = Low, **THEN** the Weight of Degree = 0.869 **AND** the Weight of Experience = 0.131.

Rule 14: **IF** Degree = Master **AND** Experience = Low, **THEN** the Weight of Degree = 0.828 **AND** the Weight of Experience = 0.172.

Rule 15: **IF** Degree = Ph.D. **AND** Experience = Low, **THEN** the Weight of Degree = 0.434 **AND** the Weight of Experience = 0.566.

TABLE IV
DEGREE OF SIMILARITY BETWEEN THE VALUES OF THE ATTRIBUTE "DEGREE"

	Bachelor	**Master**	**Ph.D.**
Bachelor	1	0.6	0.4
Master	0.6	1	0.6
Ph.D.	0.4	0.6	1

B. Calculation of the Fitness Degree

Assume that there are n tuples $\mathrm{T}_1, \mathrm{T}_2, \ldots, \mathrm{T}_n$ in a relation R of a relational database system, where the value of the attribute "Salary" of tuple $\mathrm{T_i}$ is denoted as "$\mathrm{T_i}$.Salary." Let "$\mathrm{ET_i}$.Salary" denote the estimated value of $\mathrm{T_i}$.Salary. In order to derive the value of $\mathrm{ET_i}$.Salary, we must find a tuple $\mathrm{T_j}$ which is closest to the tuple $\mathrm{T_i}$ regarding the values of the attributes "Degree" and "Experience." In the following, we present a method to calculate the degree of closeness between two tuples.

Based on a fuzzy similarity matrix, we can obtain the degree of similarity between two nonnumeric values. For example, from Table IV, we can see that the degree of similarity between the degrees "Bachelor" and "Ph.D." is 0.4. From [4], we can see that the ranks of the terms "Bachelor," "Master," and "Ph.D." are 1, 2, and 3, respectively. That is

$$\begin{aligned} \mathrm{Rank(Bachelor)} &= 1 \\ \mathrm{Rank(Master)} &= 2 \\ \mathrm{Rank(Ph.D.)} &= 3. \end{aligned}$$

Let X be a nonnumeric attribute. Based on the value $\mathrm{T_i}$.X of the attribute X of tuple $\mathrm{T_i}$ and the value $\mathrm{T_j}$.X of the attribute X of tuple $\mathrm{T_j}$, where i $\neq$ j, the degree of closeness Closeness($\mathrm{T_i}$, $\mathrm{T_j}$) between tuples $\mathrm{T_i}$ and $\mathrm{T_j}$ can be calculated by (8) or (9), where Weight($\mathrm{T_j}$.Degree) and Weight($\mathrm{T_j}$.Experience) denote the weights of the attributes "Degree" and "Experience," respectively, obtained from the fuzzified values of the attributes "Degree" and "Experience" of tuple $\mathrm{T_j}$, derived from a chromosome

$$\begin{aligned} &\text{If}\,\mathrm{Rank}(T_i.X) \geq \mathrm{Rank}(T_j.X)\, \textit{then} \\ &\quad \mathrm{Closeness}(T_i, T_j) = \mathrm{Similarity}(T_i.X, T_j.X) \\ &\qquad \times \mathrm{Weight}(T_j.\mathrm{Degree}) + \frac{T_i.\mathrm{Experience}}{T_j.\mathrm{Experience}} \\ &\qquad \times \mathrm{Weight}(T_j.\mathrm{Experience}) \end{aligned} \tag{8}$$

$$\begin{aligned} &\text{If}\,\mathrm{Rank}(T_i.X) < \mathrm{Rank}(T_j.X)\, \text{then} \\ &\quad \mathrm{Closeness}(T_i, T_j) = 1\,/\mathrm{Similarity}(T_i.X, T_j.X) \\ &\qquad \times \mathrm{Weight}(T_j.\mathrm{Degree}) + \frac{T_i.\mathrm{Experience}}{T_j.\mathrm{Experience}} \\ &\qquad \times \mathrm{Weight}(T_j.\mathrm{Experience}) \end{aligned} \tag{9}$$

where Similarity($\mathrm{T_i}$.X, $\mathrm{T_j}$.X) denotes the degree of similarity between $\mathrm{T_i}$.X and $\mathrm{T_j}$.X, and its value is obtained from a fuzzy similarity matrix of the linguistic terms of the attribute X defined by a domain expert.

Let $\mathrm{T_i}$, $\mathrm{T_j}$ and T_k be three tuples in a relational database. Assume that the degree of closeness between tuple $\mathrm{T_i}$ and $\mathrm{T_j}$ is denoted as Closeness($\mathrm{T_i}$, $\mathrm{T_j}$), and the degree of closeness between tuples $\mathrm{T_i}$ and $\mathrm{T_k}$ is denoted as Closeness($\mathrm{T_i}$, T_k). Let x = | Closeness($\mathrm{T_i}$, $\mathrm{T_j}$) − 1| and let y = | Closeness($\mathrm{T_i}$, $\mathrm{T_k}$) − 1|. If x < y, then tuple $\mathrm{T_j}$ is more close to tuple $\mathrm{T_i}$ than tuple $\mathrm{T_k}$. After the closeness degree between tuple $\mathrm{T_i}$ and all the other tuples in the relational database have been calculated, the tuple whose closeness degree with respect to tuple $\mathrm{T_i}$ is closest to 1.0 is regarded as closest to tuple $\mathrm{T_i}$, where 1 $\leq$ i $\leq$ n.

After calculating the degrees of closeness of the other tuples in the database with respect to tuple $\mathrm{T_i}$, the system will pick a tuple which is closest to tuple $\mathrm{T_i}$. Assume that tuple $\mathrm{T_j}$ is closest to tuple $\mathrm{T_i}$, then we can calculate the estimated value "$\mathrm{ET_i}$.Salary" of the attribute "Salary" of tuple $\mathrm{T_i}$ as follows:

$$ET_i.\mathrm{Salary} = T_i.\mathrm{Salary} \times \mathrm{Closeness}(T_i, \mathrm{T_j}) \tag{10}$$

where $\mathrm{T_i}$.Salary denotes the value of the attribute "Salary" of tuple $\mathrm{T_j}$.

In the same way, we can calculate the estimated values of the attribute "Salary" of all the tuples in a relational database. After we get the estimated values of the attribute "Salary" of all the tuples in a relational database, we can calculate the estimated error of each tuple by (11), where $\mathrm{Error_i}$ denotes the estimated error between the estimated value $\mathrm{ET_i}$.Salary of the attribute "Salary" of tuple $\mathrm{T_i}$ and the actual value $\mathrm{T_i}$.Salary of the attribute "Salary" of tuple $\mathrm{T_i}$

$$\mathrm{Error}_i = \frac{ET_i.\mathrm{Salary} - T_i.\mathrm{Salary}}{T_i.\mathrm{Salary}}. \tag{11}$$

Let Avg_Error denote the average estimated error of the tuples based on the combination of weights of the attributes derived from the chromosome, where

$$\mathrm{Avg\_Error} = \frac{\sum_{i=1}^{n} \mathrm{Error}_i}{n}. \tag{12}$$

Then, we can obtain the fitness degree of this chromosome as follows:

$$\text{Fitness Degree} = 1 - \mathrm{Avg\_Error}. \tag{13}$$

C. Selection Operations

After a new population is generated, we can calculate the average fitness value of this population. Assume that Chromosome 1, Chromosome 2, and Chromosome3 are three chromosomes of a population, and assume that their fitness values are 0.975, 0.744, and 0.480, respectively. Furthermore, assume that the average fitness value of this population is 0.627, then

i) for Chromosome 1: $0.975/0.627 = 1.55 \cong 2$;
ii) for Chromosome 2: $0.744/0.627 = 1.18 \cong 1$;
iii) for Chromosome 3: $0.480/0.627 = 0.77 \cong 1$.

Then, Chromosome 1 will get two positions, and Chromosome 2 and Chromosome 3 will get one position, respectively, to continue the crossover operations. After we check all the chromosomes in the population, if the number of selected chromosomes is less than the number of chromosomes in a population, we will randomly pick up chromosomes to fill it up.

The First Chromosome — Crossover Point

Gene Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0.222	0.798	0.848	0.030	0.141	0.596	0.828	0.636	0.939	0.192	0.495	0.152	0.899	0.000	0.727
	B-H	**M-H**	**P-H**	**B-SH**	**M-SH**	**P-SH**	**B-M**	**M-M**	**P-M**	**B-SL**	**M-SL**	**P-SL**	**B-L**	**M-L**	**P-L**

The Second Chromosome — Crossover Point

Gene Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0.323	0.465	0.424	0.000	0.354	0.677	0.707	0.061	0.475	0.889	0.535	0.010	0.758	0.030	0.030
	B-H	**M-H**	**P-H**	**B-SH**	**M-SH**	**P-SH**	**B-M**	**M-M**	**P-M**	**B-SL**	**M-SL**	**P-SL**	**B-L**	**M-L**	**P-L**

Fig. 6. Chromosomes before crossover operations.

The First Chromosome

Gene Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0.222	0.798	0.848	0.030	0.141	0.596	0.828	0.636	0.939	0.192	0.495	0.152	0.899	0.030	0.030
	B-H	**M-H**	**P-H**	**B-SH**	**M-SH**	**P-SH**	**B-M**	**M-M**	**P-M**	**B-SL**	**M-SL**	**P-SL**	**B-L**	**M-L**	**P-L**

The Second Chromosome

Gene Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0.323	0.465	0.424	0.000	0.354	0.677	0.707	0.061	0.475	0.889	0.535	0.010	0.758	0.000	0.727
	B-H	**M-H**	**P-H**	**B-SH**	**M-SH**	**P-SH**	**B-M**	**M-M**	**P-M**	**B-SL**	**M-SL**	**P-SL**	**B-L**	**M-L**	**P-L**

Fig. 7. Chromosomes after crossover operations.

D. Crossover Operations

In this paper, we set the crossover rate α to 1.0. Therefore, after the selection operations, the number of chromosomes in a population will continue to perform the crossover operations, where the system randomly picks up two chromosomes as the parents and randomly picks a crossover point. Then, the system performs the crossover operations on these two chromosomes at this crossover point to generate their two children. For example, after the crossover operations, the chromosomes shown in Fig. 6 will become the chromosomes shown in Fig. 7, where the crossover point is at the 13th gene labeled "B-L."

E. Mutation Operations

After the crossover operations, the chromosomes in a population will continue to perform the mutation operations, where the system randomly picks chromosomes to perform the mutation operations, and it also randomly determines which gene in a chromosome will perform the mutation operations. It should be noted that not every generation performs the mutation operations. We can choose a mutation rate β, where $\beta \in [0, 1]$, for performing the mutation operations. When the crossover operations are finished, the system generates a random number in [0, 1]. If the random number is less than β, then the system will perform the mutation operations with respect to the chromosomes in this population. For example, assume that there is a chromosome as shown in Fig. 8(a), and assume that the system randomly chooses the seventh gene of the chromosome to perform the mutation operation, then the system randomly gives the selected gene a new real value (e.g., 0.624), and the chromosome shown in Fig. 8(a) will become the chromosome as shown in Fig. 8(b).

Gene Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0.01	0.071	0.343	0.465	0.505	0.303	0.495	0.081	0.778	0.717	0.303	0.869	0.869	0.828	0.434
	B-H	**M-H**	**P-H**	**B-SH**	**M-SH**	**P-SH**	**B-M**	**M-M**	**P-M**	**B-SL**	**M-SL**	**P-SL**	**B-L**	**M-L**	**P-L**

Gene Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0.01	0.071	0.343	0.465	0.505	0.303	0.624	0.081	0.778	0.717	0.303	0.869	0.869	0.828	0.434
	B-H	**M-H**	**P-H**	**B-SH**	**M-SH**	**P-SH**	**B-M**	**M-M**	**P-M**	**B-SL**	**M-SL**	**P-SL**	**B-L**	**M-L**	**P-L**

Fig. 8. Mutation operation of a chromosome. (Top) Before the mutation operation. (Bottom) After the mutation operation.

For each generation, the GA will continue to perform the selection operations, the crossover operations, and the mutation operations. After the GA converged, we can get a best combination of the weights of the attributes. If there is a tuple in the database which has a null value, then we can estimate the null value by using the weights of attributes generated by the GA.

V. Estimating Null Values in Relational Database Systems

Assume that there is a relation in a relational database containing a null value as shown in Table V, where Table V is derived from Table II by letting the value of the attribute "Salary" of the tuple T_{22} be a null value. It should be noted that we assume that the value of the attribute "Salary" of the tuple T_{22} is unknown, and then we are able to check how close the estimation comes to the actual value.

Based on Table V, after applying the GA under the parameters: the size of population $= 60$, the number of generations $= 300$, the crossover rate $= 1.0$, and the mutation rate $= 0.2$, we can get the best chromosome which contains the combination of the weights of the attributes "Degree" and "Experience" as shown in Fig. 9.

In order to estimate the null value of the attribute "Salary" of tuple T_{22} whose EMP-ID is S22, we must find a tuple which is closest to the tuple T_{22}. The process for computing the degree of closeness between two tuples is illustrated as follows. Let us consider the tuple T_1 shown in Table V whose EMP-ID is S1 as an example. We can see that the values of the attributes "Degree" and "Experience" of the tuple T_1 are "Ph.D." (P) and "7.2," respectively. We also can see that the degree of similarity between the values of the attribute "Degree" of the tuple T_1 and the tuple T_{22} is 1 (i.e., Similarity(T_{22}.Degree, T_1.Degree) = 1). Then, based on the values of the attribute "Experience" of the tuple T_1 and the tuple T_{22}, we can get

$$\frac{\mathrm{T}_{22}.\mathrm{Experience}}{\mathrm{T}_1.\mathrm{Experience}} = \frac{8.5}{7.2} = 1.180.$$

From Table III, we can see that the fuzzified value of the attribute "Experience" of the tuple T_1 whose EMP-ID is S1 is "SH" (Somewhat High). Therefore, we pick the value of the 6th gene labeled "P-SH" of Fig. 9 (i.e., 0.303). It means that the weights of the attributes "Degree" and "Experience" are 0.303 and 0.697 (i.e., $1 - 0.303 = 0.697$), respectively. Based on (9), the degree of closeness between the tuples T_{22} and T_1 can be calculated as follows:

$$\begin{aligned}\mathrm{Closeness}(T_{22}, T_1) &= 1 \times 0.303 + 1.180 \times (1 - 0.303)\\ &= 0.303 + 0.822\\ &= 1.125.\end{aligned}$$

After the degrees of closeness of all other tuples with respect to the tuple T_{22} are calculated, we can see that the tuple T_{20} is closest to the tuple T_{22} (i.e., the closeness degree Closeness(T_{22}, T_{20}) between the tuples T_{22} and T_{20} is closest to 1.0), where the degree of closeness Closeness(T_{22}, T_{20}) between the tuples T_{22} and T_{20} is calculated as follows:

$$\begin{aligned}\mathrm{Closeness}(T_{22}, T_{20}) &= 1 \times 0.303 + \frac{8.5}{7.8} \times (1 - 0.303)\\ &= 0.303 + 1.0897 \times 0.697\\ &\cong 1.0625.\end{aligned}$$

Based on (10), by multiplying the value of the attribute "Salary" of the tuple T_{20} and the degree of closeness Closeness(T_{22}, T_{20}) between the tuple T_{20} and T_{22}, the estimated value "ET_{22}.Salary" of the attribute "Salary" of the tuple T_{22} can be calculated as follows:

$$\begin{aligned}\mathrm{ET}_{22}.\mathrm{Salary} &= 65\,000 \times 1.0625\\ &= 69\,065.83.\end{aligned}$$

From Table II, we can see that the actual value of the attribute "Salary" of the tuple T_{22} is 70 000. Based on (11), we can calculate the estimated error of the estimated value of the attribute "Salary" of the tuple T_{22} as follows:

$$\begin{aligned}\mathrm{Error}_{22} &= \frac{69\,065.83 - 70\,000}{70\,000}\\ &= -\frac{934.17}{70\,000}\\ &\cong -0.01.\end{aligned}$$

TABLE V
RELATION IN A RELATIONAL DATABASE CONTAINING A NULL VALUE

EMP-ID	Degree	Experience	Salary
S1	Ph.D.	7.2	63,000
S2	Master	2.0	37,000
S3	Bachelor	7.0	40,000
S4	Ph.D.	1.2	47,000
S5	Master	7.5	53,000
S6	Bachelor	1.5	26,000
S7	Bachelor	2.3	29,000
S8	Ph.D.	2.0	50,000
S9	Ph.D.	3.8	54,000
S10	Bachelor	3.5	35,000
S11	Master	3.5	40,000
S12	Master	3.6	41,000
S13	Master	10.0	68,000
S14	Ph.D.	5.0	57,000
S15	Bachelor	5.0	36,000
S16	Master	6.2	50,000
S17	Bachelor	0.5	23,000
S18	Master	7.2	55,000
S19	Master	6.5	51,000
S20	Ph.D.	7.8	65,000
S21	Master	8.1	64,000
S22	Ph.D.	8.5	Null

Gene Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0.010	0.071	0.343	0.465	0.505	0.303	0.495	0.081	0.778	0.717	0.303	0.869	0.869	0.828	0.434
	B-H	**M-H**	**P-H**	**B-SH**	**M-SH**	**P-SH**	**B-M**	**M-M**	**P-M**	**B-SL**	**M-SL**	**P-SL**	**B-L**	**M-L**	**P-L**

Fig. 9. Best chromosome obtained by applying the GA under the parameters: the size of population $= 60$, the number of generations $= 300$, the crossover rate $= 1.0$, and the mutation rate $= 0.2$.

By repeating the same process, the estimated salary and the estimated error of each tuple in the database shown in Table II can be obtained as shown in Table VI. From Table VI, we can see that the average estimated error of the proposed method is equal to 0.018.

It is obvious that there are many parameters in a GA, such as the size of population, the crossover rate, the mutation rate, the number of generations, etc. In this paper, we estimate the value of the attribute "Salary" of each tuple shown in Table II by considering four situations with four parameters (i.e., the size of population, the number of generations, the mutation rate, and the crossover rate), where the average estimated errors for different situations are shown in Table VII.

From Table VII, we can see that the best chromosome which has the largest fitness value occurs at the fourth situation of Table VII (i.e., under the parameters: the size of population $= 60$, the number of generations $= 300$, the crossover rate $= 1.0$, and the mutation rate $= 0.2$), where the best chromosome indicating the best combination of the weights of the attributes "Degree" and "Experience" is shown in Fig. 9. From Fig. 9 we can get the following 15 rules:

Rule 1: **IF** Degree = Bachelor **AND** Experience = High, **THEN** the Weight of Degree $= 0.010$ **AND** the Weight of Experience $= 0.990$

Rule 2: **IF** Degree = Master **AND** Experience = High, **THEN** the Weight of Degree $= 0.071$ **AND** the Weight of Experience $= 0.929$

Rule 3: **IF** Degree = Ph.D. **AND** Experience = High, **THEN** the Weight of Degree $= 0.343$ **AND** the Weight of Experience $= 0.657$

TABLE VI
ESTIMATED SALARY AND ESTIMATED ERROR OF EACH TUPLE

EMP-ID	Degree	Experience	Salary	Salary (Estimated)	Estimated Error
1	Ph.D.	7.2	63,000	61515.00	- 0.024
2	Master	2	37,000	36967.44	- 0.001
3	Bachelor	7	40,000	40634.14	+ 0.016
4	Ph.D.	1.2	47,000	46873.66	- 0.003
5	Master	7.5	53,000	56134.37	+ 0.059
6	Bachelor	1.5	26,000	26146.40	+ 0.006
7	Bachelor	2.3	29,000	27822.08	- 0.041
8	Ph.D.	2	50,000	50067.20	+ 0.001
9	Ph.D.	3.8	54,000	53958.94	- 0.001
10	Bachelor	3.5	35,000	35152.00	+ 0.004
11	Master	3.5	40,000	40206.19	+ 0.005
12	Master	3.6	41,000	40796.57	- 0.005
13	Master	10	68,000	68495.74	+ 0.007
14	Ph.D.	5	57,000	56240.72	- 0.013
15	Bachelor	5	36,000	34277.54	- 0.048
16	Master	6.2	50,000	49834.85	- 0.003
17	Bachelor	0.5	23,000	23722.40	+ 0.031
18	Master	7.2	55,000	51950.6	- 0.055
19	Master	6.5	51,000	51197.58	+ 0.004
20	Ph.D.	7.8	65,000	64813.75	- 0.003
21	Master	8.1	64,000	60853.28	- 0.049
22	Ph.D.	8.5	70,000	69065.83	- 0.013
Average Estimated Error					0.018

TABLE VII
AVERAGE ESTIMATED ERRORS FOR DIFFERENT PARAMETERS OF THE GA

Size of Population	Number of Generations	Crossover Rate	Mutation Rate	Average Estimated Error
30	100	1.0	0.1	0.036
40	150	1.0	0.1	0.032
50	200	1.0	0.2	0.027
60	300	1.0	0.2	0.018

Rule 4: **IF** Degree = Bachelor **AND** Experience = Somewhat High, **THEN** the Weight of Degree = 0.465 **AND** the Weight of Experience = 0.535

Rule 5: **IF** Degree = Master **AND** Experience = Somewhat High, **THEN** the Weight of Degree = 0.505 **AND** the Weight of Experience = 0.495

Rule 6: **IF** Degree = Ph.D. **AND** Experience = Somewhat High, **THEN** the Weight of Degree = 0.303 **AND** the Weight of Experience = 0.697

Rule 7: **IF** Degree = Bachelor **AND** Experience = Medium, **THEN** the Weight of Degree = 0.495 **AND** the Weight of Experience = 0.505

Rule 8: **IF** Degree = Master **AND** Experience = Medium, **THEN** the Weight of Degree = 0.081 **AND** the Weight of Experience = 0.919

Rule 9: **IF** Degree = Ph.D. **AND** Experience = Medium, **THEN** the Weight of Degree = 0.778 **AND** the Weight of Experience = 0.222

Rule 10: **IF** Degree = Bachelor **AND** Experience = Somewhat Low, **THEN** the Weight of Degree = 0.717 **AND** the Weight of Experience = 0.283

Rule 11: **IF** Degree = Master **AND** Experience = Somewhat Low, **THEN** the Weight of Degree = 0.303 **AND** the Weight of Experience = 0.697

Rule 12: **IF** Degree = Ph.D. **AND** Experience = Somewhat Low, **THEN** the Weight of Degree = 0.869 **AND** the Weight of Experience = 0.131

Rule 13: **IF** Degree = Bachelor **AND** Experience = Low, **THEN** the Weight of Degree = 0.869 **AND** the Weight of Experience = 0.131

Rule 14: **IF** Degree = Master **AND** Experience = Low, **THEN** the Weight of Degree = 0.828 **AND** the Weight of Experience = 0.172

Rule 15: **IF** Degree = Ph.D. **AND** Experience = Low, **THEN** the Weight of Degree = 0.434 **AND** the Weight of Experience = 0.566.

Based on the best chromosome shown in Fig. 9, we can estimate the values of the attribute "Salary" of the tuples shown in Table II. A comparison of the average estimated error of the proposed method with that of the methods we presented in [4] and [6] is shown in Table VIII. From Table VIII, we can see that the average estimated error of the proposed method is smaller than that of the methods we presented in [4] and [6].

TABLE VIII
COMPARISON OF THE ESTIMATED RESULTS OF THE PROPOSED METHOD WITH THE EXISTING METHODS

EMP-ID	Degree	Experience	Salary	Chen-and-Chen's Method [4]		Chen-and-Yeh's Method [6]		The Proposed Method	
				Salary (Estimated)	Estimated Error	Salary (Estimated)	Estimated Error	Salary (Estimated)	Estimated Error
1	Ph.D.	7.2	63,000	63000	+ 0.000	65000	+ 0.032	61515.00	- 0.024
2	Master	2	37,000	33711	- 0.089	30704	- 0.170	36967.44	- 0.001
3	Bachelor	7	40,000	46648	+ 0.166	35000	- 0.125	40634.14	+ 0.016
4	Ph.D.	1.2	47,000	36216	- 0.229	46000	- 0.021	46873.66	- 0.003
5	Master	7.5	53,000	56200	+ 0.060	54500	+ 0.028	56134.37	+ 0.059
6	Bachelor	1.5	26,000	27179	+ 0.045	26346	+ 0.013	26146.40	+ 0.006
7	Bachelor	2.3	29,000	29195	+ 0.007	28500	- 0.017	27822.08	- 0.041
8	Ph.D.	2	50,000	39861	- 0.203	50000	+ 0.000	50067.20	+ 0.001
9	Ph.D.	3.8	54,000	48061	- 0.110	55000	+ 0.019	53958.94	- 0.001
10	Bachelor	3.5	35,000	32219	- 0.079	31538	- 0.099	35152.00	+ 0.004
11	Master	3.5	40,000	40544	+ 0.014	41590	+ 0.040	40206.19	+ 0.005
12	Master	3.6	41,000	41000	+ 0.000	45159	+ 0.101	40796.57	- 0.005
13	Master	10	68,000	64533	- 0.051	65000	- 0.044	68495.74	+ 0.007
14	Ph.D.	5	57,000	55666	- 0.023	55000	- 0.035	56240.72	- 0.013
15	Bachelor	5	36,000	35999	- 0.000	35000	- 0.028	34277.54	- 0.048
16	Master	6.2	50,000	51866	+ 0.037	48600	- 0.028	49834.85	- 0.003
17	Bachelor	0.5	23,000	24659	+ 0.072	25000	+ 0.087	23722.40	+ 0.031
18	Master	7.2	55,000	55200	+ 0.004	52400	- 0.047	51950.6	- 0.055
19	Master	6.5	51,000	52866	+ 0.037	49500	- 0.029	51197.58	+ 0.004
20	Ph.D.	7.8	65,000	65000	+ 0.000	65000	+ 0.000	64813.75	- 0.003
21	Master	8.1	64,000	58200	- 0.091	58700	- 0.083	60853.28	- 0.049
22	Ph.D.	8.5	70,000	67333	- 0.038	65000	- 0.071	69065.83	- 0.013
Average Estimated Error				0.062		0.051		0.018	

VI. CONCLUSION

In this paper, we have presented a new method to estimate null values in relational database systems using GAs, where the attributes appearing in the antecedent parts of the generated fuzzy rules have different weights. We use GAs to tune the weights of attributes for estimating null values in relational database systems. After a predefined number of evolutions of the GA, the best chromosome contains the optimal weights of the attributes, and they can be translated into a set of rules to be used for estimating null values. The proposed method can get a higher average estimated accuracy rate than the methods we presented in [4] and [6].

REFERENCES

[1] P. Bosc, D. Dubois, and H. Prade, "Fuzzy functional dependencies and redundancy elimination," *J. Amer. Soc. Inform. Sci.*, vol. 49, pp. 217–235, 1998.

[2] P. Bosc and L. Liétard, "Functional dependencies and the design of relational databases extended to imprecise data," in *Proc. Conf. Information Processing and Management Uncertainty Knowledge-Based System (IPMU'98)*, Paris, France, 1998, pp. 412–419.

[3] D. G. Burkhardt and P. P. Bonissone, "Automated fuzzy knowledge base Generation and tuning," in *Proc. 1992 IEEE Int. Conf. Fuzzy Systems*, San Diego, CA, 1992, pp. 179–188.

[4] S. M. Chen and H. H. Chen, "Estimating null values in the distributed relational databases environment," *Cybern. Syst.*, vol. 31, no. 8, pp. 851–871, 2000.

[5] S. M. Chen, S. H. Lee, and C. H. Lee, "A new method for generating fuzzy rules from numerical data for handling classification problems," *Appl. Art. Intell.*, vol. 15, no. 7, pp. 645–664, 2001.

[6] S. M. Chen and M. S. Yeh, "Generating fuzzy rules from relational database systems for estimating null values," *Cybern. Syst.*, vol. 28, no. 8, pp. 695–723, 1997.

[7] J. C. Cubero and M. A. Vila, "A new definition of fuzzy functional dependency in fuzzy relational databases," *Int. J. Intell. Syst.*, vol. 9, no. 5, pp. 441–448, 1994.

[8] L. David, *Handbook of Genetic Algorithms*. New York: Van Nostrand Reinhold, 1991.

[9] J. T. Dockey and E. Murray, "Fuzzy linguistic database: An application," *Inform. Syst.*, vol. 14, no. 6, pp. 443–453, 1989.

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

[11] J. Grant, "Null values in a relational data base," *Inform. Processing Lett.*, vol. 6, no. 5, pp. 156–157, 1977.

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

[13] C. M. Huang and S. M. Chen, "A new method to estimate null values in relational database systems using genetic algorithms," in *Proc. 6th Conf. Artificial Intelligence Applications*, Kauhsiung, Taiwan, R.O.C., 2001, pp. 599–604.

[14] B. Jeng and T. P. Liang, "Fuzzy indexing and retrieval in case-based system," *Expert Syst. Applicat.*, vol. 8, no. 1, pp. 135–142, 1995.

[15] H. L. Lin and S. M. Chen, "Generating weighted fuzzy rules from training data for handling fuzzy classification problems," in *Proc. 2000 Int. Computer Symp.: Workshop Artificial Intelligence*, Chiayi, Taiwan, R.O.C., 2000, pp. 11–18.

[16] M. Srinivas and L. M. Patnaik, "Genetic algorithms: A survey," *IEEE Comput.*, vol. 27, pp. 17–26, June 1994.

[17] H. Tsukimoto, "Extracting rules from trained neural networks," *IEEE Trans. Neural Networks*, vol. 11, pp. 377–389, Apr. 2000.

[18] L. X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, pp. 1414–1427, Dec. 1992.

[19] L. A. Zadeh, "Fuzzy sets," *Inform. Control*, vol. 8, pp. 338–353, 1965.

[20] ——, "Fuzzy logic," *IEEE Comput.*, vol. 21, pp. 89–91, Apr. 1988.

[21] ——, "The concept of a linguistic variable and its application to approximate reasoning—I," *Inform. Sci.*, vol. 8, no. 3, pp. 199–249, 1975.

Shyi-Ming Chen (M'88–SM'96) was born in Taipei, Taiwan, R.O.C., in 1960. He received the B.S. degree in electronic engineering from the National Taiwan University of Science and Technology, Taipei, Taiwan, in 1982 and the M.S. and Ph.D. degrees in electrical engineering from the National Taiwan University, Taipei, Taiwan, in 1986 and 1991, respectively.

From August 1987 to July 1989 and from August 1990 to July 1991, he was with the Department of Electronic Engineering, Fu-Jen University, Taipei, Taiwan. From August 1991 to July 1996, he was an Associate Professor in the Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan. From August 1996 to July 1998, he was a Professor in the Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan. From August 1998 to July 2001, he was a Professor in the Department of Electronic Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan. Since August 2001, he has been a Professor in the Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology. He was a Visiting Scholar in the Department of Electrical Engineering and Computer Science, University of California, Berkeley. His current research interests include fuzzy systems, database systems, knowledge-based systems, artificial intelligence, data mining, and genetic algorithms. He has published more than 170 papers in refereed journals, book chapters, and conference proceedings.

Dr. Chen has won numerous awards, including the 1994 Outstanding Paper Award of the *Journal of Information and Education*, the 1995 Outstanding Paper Award of the Computer Society of the Republic of China, the 1997 Outstanding Youth Electrical Engineer Award of the Chinese Institute of Electrical Engineering, Republic of China, the Best Paper Award of the 1999 National Computer Symposium, Republic of China, the the 1999 Outstanding Paper Award of the Computer Society of the Republic of China, the 2001 Outstanding Talented Person Award, Republic of China, for the contributions in Information Technology, and the Outstanding Electrical Engineering Professor Award granted by the Chinese Institute of Electrical Engineering (CIEE), Republic of China, in 2002. He is a Member of the ACM, the International Fuzzy Systems Association (IFSA), and Phi Tau Phi. He is an Executive Committee Member of the Taiwanese Association for Artificial Intelligence (TAAI). He is also an Executive Committee Member of the Chinese Fuzzy Systems Association (CFSA). He is an Editor of the *Journal of the Chinese Grey System Association* and an Associate Editor of the *International Journal of Fuzzy Systems*. He is listed in *International Who's Who of Professionals*, *Marquis Who's Who in the World*, and *Marquis Who's Who in Science and Engineering*.

Chung-Ming Huang was born in 1978, in Taipei, Taiwan, R.O.C. He received the B.S. degree in computer and information science from Soochow University, Taipei, Taiwan, and the M.S. degree in electronic engineering from National Taiwan University of Science and Technology, Taipei, in 2000 and 2002, respectively.

His current research interests include fuzzy systems, database systems, and artificial intelligence.