

# Genetic-Fuzzy Data Mining With Divide-and-Conquer Strategy

Tzung-Pei Hong, Chun-Hao Chen, Yeong-Chyi Lee, and Yu-Lung Wu

**Abstract**—Data mining is most commonly used in attempts to induce association rules from transaction data. Most previous studies focused on binary-valued transaction data. Transaction data in real-world applications, however, usually consist of quantitative values. This paper, thus, proposes a fuzzy data-mining algorithm for extracting both association rules and membership functions from quantitative transactions. A genetic algorithm (GA)-based framework for finding membership functions suitable for mining problems is proposed. The fitness of each set of membership functions is evaluated by the fuzzy-supports of the linguistic terms in the large 1-itemsets and by the suitability of the derived membership functions. The evaluation by the fuzzy supports of large 1-itemsets is much faster than that when considering all itemsets or interesting association rules. It can also help divide-and-conquer the derivation process of the membership functions for different items. The proposed GA framework, thus, maintains multiple populations, each for one item's membership functions. The final best sets of membership functions in all the populations are then gathered together to be used for mining fuzzy association rules. Experiments are conducted to analyze different fitness functions and set different fitness functions and setting different supports and confidences. Experiments are also conducted to compare the proposed algorithm, the one with uniform fuzzy partition, and the existing one without divide-and-conquer, with results validating the performance of the proposed algorithm.

**Index Terms**—Association rule, data mining, fuzzy set, genetic algorithm (GA), membership function.

## I. INTRODUCTION

**D**ATA MINING IS most commonly used in attempts to induce association rules from transaction data. An association rule is an expression  $X \rightarrow Y$ , where  $X$  is a set of items and  $Y$  is a single item. It means in the set of transactions, if all the items in  $X$  exist in a transaction, then  $Y$  is also in the transaction with a high probability [1]. Most previous studies focused on binary valued transaction data. Transaction data in real-world applications, however, usually consist of quantitative

values. Designing a sophisticated data-mining algorithm able to deal with various types of data presents a challenge to workers in this research field.

Recently, fuzzy set theory has been used more and more frequently in intelligent systems because of its simplicity and similarity to human reasoning [17]. The theory has been applied in fields such as manufacturing, engineering, diagnosis, economics, among others [17]. Several fuzzy learning algorithms for inducing rules from given sets of data have been designed and used to good effect with specific domains [9], [11], [14]. As to fuzzy data mining, Hong *et al.* proposed an algorithm to mine fuzzy rules from quantitative data [12]. They transformed each quantitative item into a fuzzy set and used fuzzy operations to find fuzzy rules. Cai *et al.* proposed weighted mining to reflect different importance to different items [5]. Each item had a numerical weight that had been attached to it by users. Weighted supports and weighted confidences were then defined to determine interesting association rules. Yue *et al.* then extended their concepts to fuzzy item vectors [29]. Lee *et al.* proposed a mining algorithm which used multiple minimum supports of different items to mine fuzzy association rules [21]. In the above approaches, the membership functions were assumed to be known in advance. Although many approaches for learning membership functions were proposed [6], [23], [24], [26], [27], most of them were usually used for classification or control problems.

In mining problems, the given membership functions may, however, have a critical influence on the final mining results. Kaya *et al.* proposed a genetic algorithm (GA)-based clustering method to derive a predefined number of membership functions for getting a maximum profit within an interval of user specified minimum support values [19]. Kaya *et al.* also proposed a multiobjective GA to find a number of Pareto-optimal rules sets according to two objective functions, number of rules and execution time [18]. Hong *et al.* proposed a fuzzy data-mining algorithm for extracting both association rules and membership functions from quantitative transactions [10]. It maintained a population of sets of membership functions, and used the GA to automatically derive the resulting one. Since the membership functions of items were encoded into a long chromosome, it took lots of time to learn global solutions [10].

Recently, the divide-and-conquer strategy has been used in the evolutionary computation community to very good effect. Many algorithms based on it have also been proposed in different applications [2], [7], [20], [28]. In this paper, we thus propose an enhanced algorithm that can divide-and-conquer the derivation process of the membership functions for different items. The proposed algorithm maintains multiple populations, each for one item's membership functions. The fitness of each set of membership functions is evaluated by the fuzzy-supports

Manuscript received February 19, 2006; revised June 28, 2006, November 4, 2006, and January 28, 2007. This work was supported in part by the National Science Council of the Republic of China under Contract NSC 95-2221-E-390-025. This is a modified and expanded version of the paper "Using Divide-and-Conquer GA Strategy in Fuzzy Data Mining" presented at the Ninth IEEE Symposium on Computers and Communications, 2004, Egypt.

T.-P. Hong is with the Department of Electrical Engineering, National University of Kaohsiung, Kaohsiung 811, Taiwan, R.O.C. (e-mail: tphong@nuk.edu.tw).

C.-H. Chen is with the Department of Computer Science and Information Engineering, National Cheng-Kung University, Tainan 701, Taiwan, R.O.C. (e-mail: chchen@idb.csie.ncku.edu.tw).

Y.-C. Lee and Y.-L. Wu are with the Institute of Information Engineering, I-Shou University, Kaohsiung 840, Taiwan, R.O.C. (e-mail: d9003007@stmail.isu.edu.tw; wuyulung@isu.edu.tw).

Digital Object Identifier 10.1109/TEVC.2007.900992

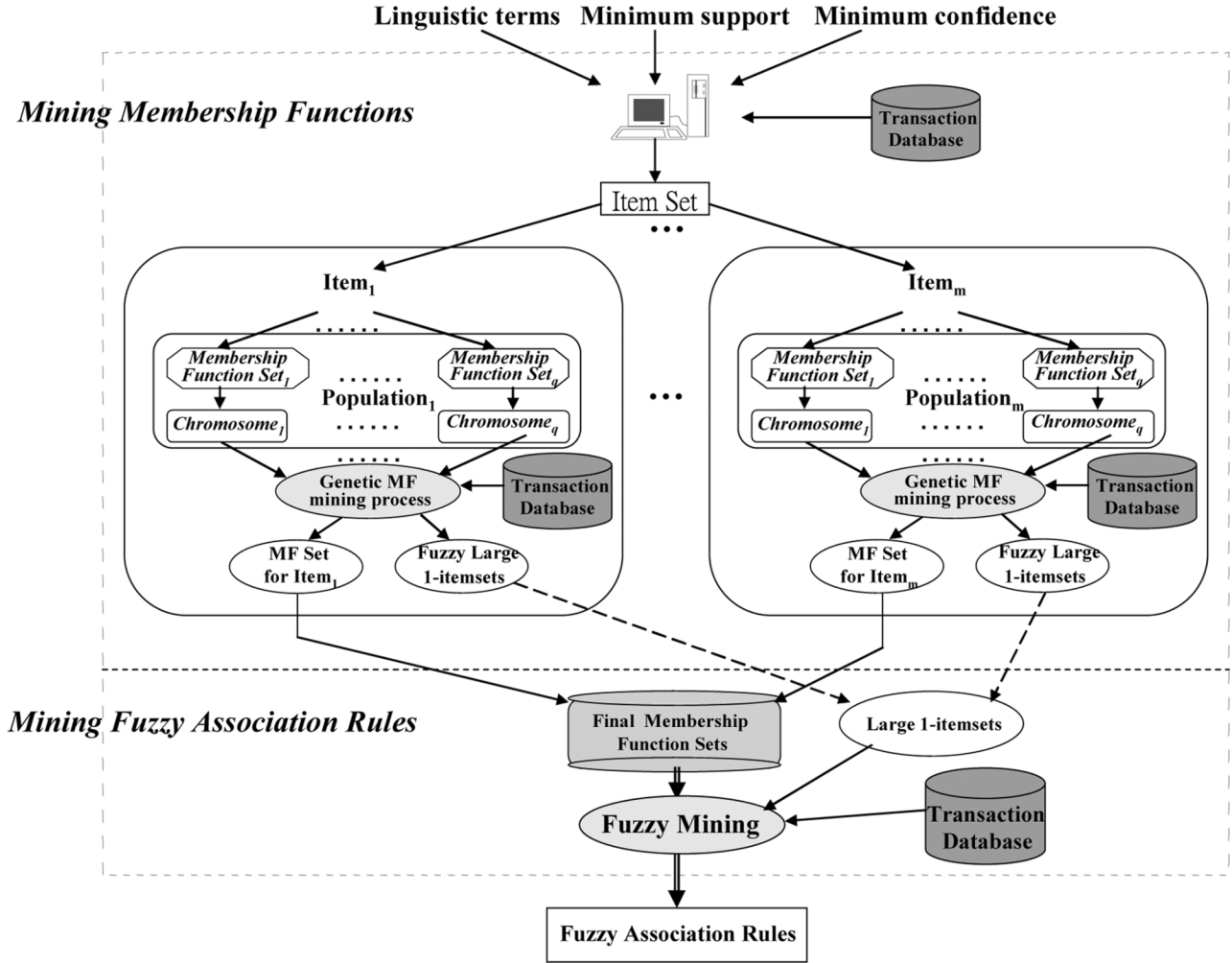


Fig. 1. The proposed GA-based framework for fuzzy data mining.

of the linguistic terms only in large 1-itemsets and by the suitability of the derived membership functions. A linguistic term of an item with a larger fuzzy-support value in 1-itemsets will result in its appearance in itemsets of more items with a higher probability, which will thus imply more interesting association rules. The evaluation by the fuzzy supports in 1-itemsets is, however, faster than that by considering all itemsets or interesting association rules. It can also help divide-and-conquer the derivation process of the membership functions for different items. The final best sets of membership functions in all the populations are then gathered together to be used for mining fuzzy association rules.

The remaining parts of this paper are organized as follows. A GA-based mining framework based on the divide-and-conquer strategy is proposed in Section II. The chromosome representation adopted is described in Section III. The adjustment process of membership functions is explained in Section IV. The details of the proposed algorithm for mining both association rules and membership functions are described in Section V. An example to illustrate the proposed algorithm is given in Section VI. Experiments to demonstrate the performance of the proposed algorithm are stated in Section VII. Conclusions and future works are given in Section VIII.

## II. A GA-BASED MINING FRAMEWORK BASED ON THE DIVIDE-AND-CONQUER STRATEGY

In this section, the fuzzy and GA concepts are used to discover both useful association rules and suitable membership functions from quantitative values. A GA-based framework with the divide-and-conquer strategy is proposed for searching for membership functions suitable for the mining problems. The final best sets of membership functions in all the populations are then gathered together to be used for mining fuzzy association rules. The proposed framework is shown in Fig. 1.

The proposed framework in Fig. 1 is divided into two phases: mining membership functions and mining fuzzy association rules. Assume the number of items is  $m$ . In the phase of mining membership functions, it maintains  $m$  populations of membership functions, with each population for an item  $I_j$  ( $1 \leq j \leq m$ ). Each chromosome in a population represents a possible set of membership functions for that item. The chromosomes in the same population are of the same length. The proposed mechanism then chooses appropriate strings for “mating,” gradually creating good offspring sets of membership functions. The offspring sets of membership functions undergo recursive “evolution” until a good set of membership functions

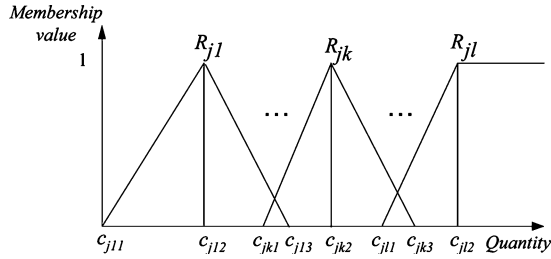


Fig. 2. Membership functions of item  $I_j$ .

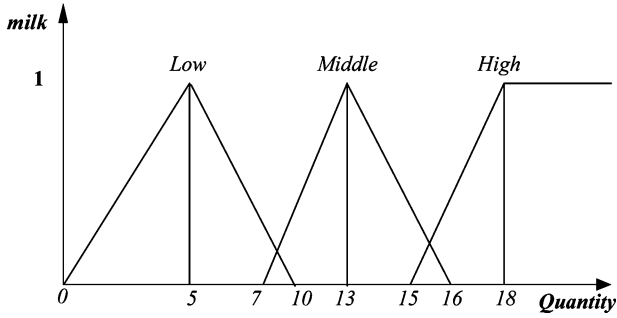


Fig. 3. An example of membership functions for item *milk*.

has been obtained. Next, in the phase of mining fuzzy association rules, the sets of membership function for all the items are gathered together and used to mine the fuzzy interesting association rules from the given quantitative database. Our previous fuzzy mining algorithm proposed in [13] is adopted to achieve this purpose.

### III. CHROMOSOME REPRESENTATION

It is important to encode membership functions as string representation for GAs to be applied. Several possible encoding approaches have been described in [6], [22], [26], and [27]. In this paper, each set of membership functions is encoded as a chromosome and handled as an individual with real-number schema.

Assume the membership functions are triangular. Three parameters are thus used to represent a membership function. Fig. 2 shows an example for item  $I_j$ , where  $R_{jk}$  denotes the membership function of the  $k$ th linguistic term for  $I_j$  and  $c_{jkp}$  indicates the  $p$ th parameter of fuzzy region  $R_{jk}$ . The inequality condition of the center values of membership functions is  $c_{j12} \leq c_{j22} \leq \dots \leq c_{jl2}$ . For each membership function, the inequality condition of the three parameters is  $c_{jk1} < c_{jk2} < c_{jk3}$ .

The membership functions of item  $I_j$  can be represented as a string  $c_{j11}c_{j12}c_{j13}c_{j21}c_{j22}c_{j23}, \dots, c_{j11}c_{j12}c_{j13}$ , where  $c_{jl3} = \infty$ . A chromosome is thus encoded as a real-number string rather than a bit string. All the chromosomes in the same population have the same string length. Below, an example is given to demonstrate the process of encoding membership functions.

*Example 1:* Assume there are four items in a transaction database: milk, bread, cookies, and beverage. Also, assume a possible set of membership functions for item *milk* is given, as shown in Fig. 3.

There are three linguistic terms, *Low*, *Middle*, and *High*, for this item. According to the proposed encoding scheme, the chro-

$$\overbrace{0, 5, 10, 7, 13, 16, 15, 18, \infty}^{MF_{milk}}$$

$$\underbrace{c_{111} c_{112} c_{111}}_{R_{11} \text{ Low}} \quad \underbrace{c_{113} c_{122} c_{121}}_{R_{12} \text{ Middle}} \quad \underbrace{c_{133} c_{132} c_{133}}_{R_{13} \text{ High}}$$

Fig. 4. The chromosome representation for the set of membership functions in Fig. 3.

mosome for representing the set of membership functions in Fig. 3 is encoded, as shown in Fig. 4.

The membership function of *Low* for *milk* is encoded as (0, 5, 10) according to Fig. 4. Similarly, the membership functions for *Middle* and *High* are, respectively, encoded as (7, 13, 16) and (15, 18,  $\infty$ ). The chromosome is then the catenation of the three tuples.

According to the proposed representation, each chromosome will consist of  $3 * |I_j|$  real numbers for item  $I_j$ , where  $|I_j|$  is the number of linguistic terms for  $I_j$ . Since the length is short when compared with the other approaches in which each chromosome consists of the membership functions for all items, the convergence of the solutions can easily be obtained. This representation, thus, allows genetic operators to search for appropriate solutions quickly. Other types of membership functions with more parameters can also be used as well, due to the short chromosome length. If computation time and memory requirement are critical issues, other simplified encoding methods can also be used in the approach. For example, adjacent membership functions may be set to always overlap with each other at the membership value 0.5 and the sum of adjacent membership functions may be always 1. In this case, the proposed approach can work with a shorter computation time and a smaller memory requirement, but the flexibility of setting the membership functions is decreased. Besides, the number of linguistic terms is not easily manually determined when the number of items is large. In this case, appropriate data preprocessing is thus needed. For example, clustering techniques may be used to find appropriate number of linguistic terms.

### IV. THE GENETIC PROCESS

#### A. Initial Population

A GA requires a population of feasible solutions to be initialized and updated during the evolution process. In the proposed mechanism, multiple populations are conceptually used, each for the membership functions of a certain item. They can be implemented in parallel or sequentially one by one. With parallel implementation, the phase of mining membership functions is accelerated since each population can be done at the same time. Each individual within a population is a possible set of triangular membership functions for an item. The initial set of chromosomes in a population is randomly generated within some constraints for forming feasible membership functions.

#### B. Fitness and Selection

In order to develop a good set of membership functions from an initial population, the GA selects parent sets of membership functions with high fitness values for mating. An evalua-

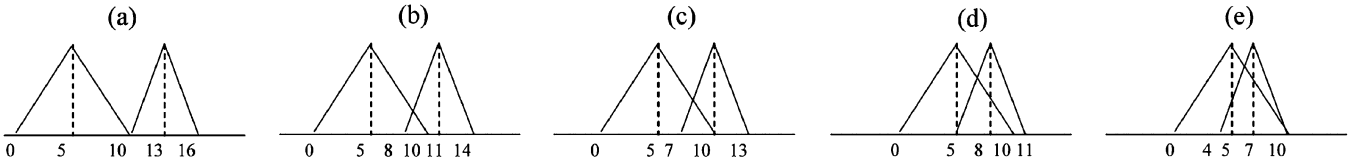


Fig. 5. The membership functions with different overlaps.

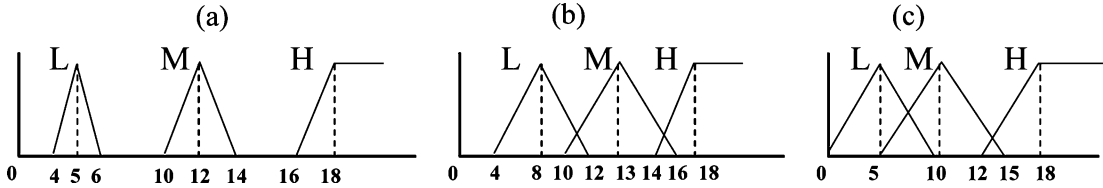


Fig. 6. The membership functions with different coverage factors.

tion function is defined to qualify the derived sets of membership functions. The evaluation values are then fed back to the GA to control how the solution space is searched, to promote the quality of the membership functions. The fitness value of each set of membership functions is determined according to two factors: suitability of membership functions and fuzzy supports of large 1-itemsets. The suitability of membership functions is composed of two terms, overlap and coverage, which are described below.

The overlap ratio of two membership functions  $R_{jk}$  and  $R_{ji}$  ( $k < i$ ) is defined as the overlap length divided by the minimum of the right span of  $R_{jk}$  and the left span of  $R_{ji}$ . That is

$$\text{overlap\_ratio}(R_{jk}, R_{ji}) = \frac{\text{overlap}(R_{jk}, R_{ji})}{\min(c_{jk3} - c_{jk2}, c_{ji2} - c_{ji1})}$$

where  $\text{overlap}(R_{jk}, R_{ji})$  is the overlap length of  $R_{jk}$  and  $R_{ji}$ .

If the overlap length is larger than the minimum of the above two half spans, then these two membership functions are thought of as a little redundant. In this case, appropriate punishment must be considered. Thus, the overlap factor of the membership functions for an item  $I_j$  in the chromosome  $C_q$  is defined as

$$\text{overlap\_factor}(C_q) = \sum_{k < i} \left[ \max \left( \left( \frac{\text{overlap}(R_{jk}, R_{ji})}{\min(c_{jk3} - c_{jk2}, c_{ji2} - c_{ji1})} \right), 1 \right) - 1 \right].$$

The overlap factor is designed for avoiding the membership functions overlapping too much. Fig. 5 shows five cases with different overlaps of membership functions.

The overlap\_factors of the three cases (a), (b), and (c) (Fig. 5) are all 0 since the overlapping range of the two membership functions are below a half span of any membership function. These membership functions are thus good for the overlapping criterion. On the contrary, the overlap\_factor of case (d) is 0.66 and of case (e) is 1 (Fig. 5). They are considered worse than the previous three cases. The overlap\_factor is thus designed for transforming the semantic meaning of overlaps between two membership functions into a numerical value and used for evaluating the fitness of membership functions.

The coverage ratio of membership functions for an item  $I_j$  is defined as the coverage range of the functions divided by the maximum quantity of that item in the transactions. The larger

the coverage ratio is, the better the derived membership functions are. Thus, the coverage factor of the membership functions for an item  $I_j$  in the chromosome  $C_q$  is defined as

$$\text{coverage\_factor}(C_q) = \frac{1}{\frac{\text{range}(R_{j1}, \dots, R_{jl})}{\max(I_j)}}$$

where  $\text{range}(R_{j1}, R_{j2}, \dots, R_{jl})$  is the coverage range of the membership functions,  $l$  is the number of membership functions for  $I_j$ , and  $\max(I_j)$  is the maximum quantity of  $I_j$  in the transactions.

The coverage\_factor is designed for avoiding membership functions separating too much. Fig. 6 shows three cases with different coverage of membership functions.

The coverage\_factor of case (c) is 1 (Fig. 6) since its coverage range of membership functions contains all the item's possible quantities in the transactions. These membership functions are thus good for the coverage criterion. By contrast, the coverage\_factor of case (a) is 2.27 and of case (b) is 1.20 when  $\max(I_j) = 18$  (Fig. 6). They are considered worse than the third case. The coverage\_factor is thus designed for transforming the semantic meaning of coverage of membership functions into a numerical value and used for evaluating the fitness of membership functions.

The suitability of the set of membership functions in a chromosome  $C_q$  is thus defined as

$$\text{suitability}(C_q) = \text{overlap\_factor}(C_q) + \text{coverage\_factor}(C_q).$$

The fitness value of a chromosome  $C_q$  is then defined as

$$f(C_q) = \frac{\sum_{X \in L_1} \text{fuzzy\_support}(X)}{\text{suitability}(C_q)}$$

where  $L_1$  is the set of large 1-itemsets obtained by using the set of membership functions in  $C_q$ , and  $\text{fuzzy\_support}(X)$  is the fuzzy support of the large 1-itemset  $X$  from the given transaction database. The  $\text{fuzzy\_support}(X)$  is defined as follows:

$$\text{fuzzy\_support}(X) = \frac{\sum_{i=1}^n f_{jk}^{(i)}}{n}$$

where  $X$  is a fuzzy region of an item (say  $R_{jk}$ ),  $f_{jk}^{(i)}$  represents the fuzzy value of  $X$  in the  $i$ th transaction, and  $n$  is the number of transactions.

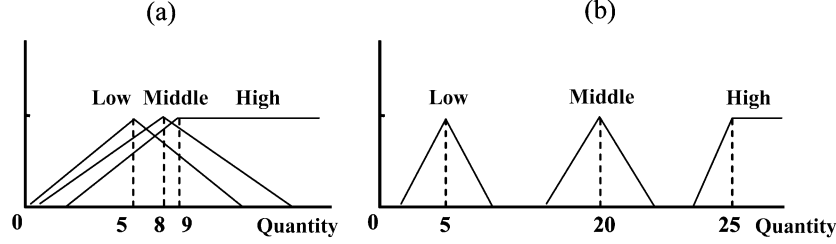


Fig. 7. Two bad sets of membership functions.

The suitability factor used in the fitness function can reduce the occurrence of the two bad kinds of membership functions shown in Fig. 7, where the first is too redundant, and the second is too separate.

The overlap factor is designed for avoiding the first bad case, and the coverage factor is for the second. Below, an example is given to illustrate the above idea.

*Example 2:* Continuing from Example 1, assume  $\max(I_1) = 13$ . The suitability of the chromosome  $C_1$  in Fig. 4 for item *milk* is calculated as follows:

$$\begin{aligned} \text{Suitability}(C_1) &= \sum_{k \neq i} \left[ \max \left( \left( \frac{\text{overlap}(R_{jk}, R_{ji})}{\min(c_{jk3} - c_{jk2}, c_{ji2} - c_{ji1})} \right), 1 \right) - 1 \right] \\ &\quad + \frac{1}{\frac{\text{range}(R_{j1}, R_{j2}, R_{j3})}{\max(I_j)}} \\ &= \left[ \max \left( \left( \frac{3}{\min(5, 6)} \right), 1 \right) - 1 \right] \\ &\quad + \left[ \max \left( \left( \frac{1}{\min(3, 3)} \right), 1 \right) - 1 \right] \\ &\quad + \left[ \max \left( \left( \frac{0}{\min(5, 3)} \right), 1 \right) - 1 \right] + \frac{1}{\frac{\text{range}(R_{11}, R_{12}, R_{13})}{13}} \\ &= 0 + 0 + 0 + 1 \\ &= 1. \end{aligned}$$

Besides, using the fuzzy-supports of large 1-itemsets can achieve a tradeoff between execution time and rule interestingness. Usually, a linguistic term of an item with a larger fuzzy-support in the 1-itemsets will result in its appearance in itemsets of more items with a higher probability, which will thus imply more association rules. The evaluation by the fuzzy supports in 1-itemsets is, however, much faster than that by considering all itemsets or interesting association rules.

### C. Genetic Operators

Genetic operators are very important to the success of specific GA applications. Two genetic operators, the *max-min-arithmetic (MMA) crossover* proposed in [8] and the *one-point mutation* are used in the genetic fuzzy mining framework. Assume there are two parent chromosomes shown below

$$\begin{aligned} C_u^t &= (g_{u1}, \dots, g_{uh}, \dots, g_{uZ}) \\ C_w^t &= (g'_{w1}, \dots, g'_{wh}, \dots, g'_{wZ}). \end{aligned}$$

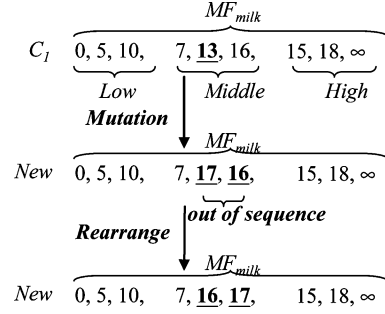


Fig. 8. A mutation operation.

$C_u^t$  represents the  $u$ th chromosome in generation  $t$ . The MMA crossover operator will generate the following four candidate chromosomes from them:

1.  $C_1^{t+1} = (g_{11}^{t+1}, \dots, g_{1h}^{t+1}, \dots, g_{1Z}^{t+1})$ , where  $g_{1h}^{t+1} = dg_{uh} + (1-d)g'_{wh}$ ,
2.  $C_2^{t+1} = (g_{21}^{t+1}, \dots, g_{2h}^{t+1}, \dots, g_{2Z}^{t+1})$ , where  $g_{2h}^{t+1} = dg'_{wh} + (1-d)g_{uh}$ ,
3.  $C_3^{t+1} = (g_{31}^{t+1}, \dots, g_{3h}^{t+1}, \dots, g_{3Z}^{t+1})$ , where  $g_{3h}^{t+1} = \min\{g_{uh}, g'_{wh}\}$ ,
4.  $C_4^{t+1} = (g_{41}^{t+1}, \dots, g_{4h}^{t+1}, \dots, g_{4Z}^{t+1})$ , where  $g_{4h}^{t+1} = \max\{g_{uh}, g'_{wh}\}$

where the parameter  $d$  is either a constant or a variable whose value depends on the age of the population. The best two chromosomes of the above four candidates are then chosen as the offspring.

The one-point mutation operator will create a new fuzzy membership function by adding a random value  $\varepsilon$  (may be negative) to one parameter of an existing linguistic term, say  $R_{jk}$ . Assume that  $c_{jkp}$  represents a parameter of  $R_{jk}$ . The parameter of the newly derived membership function may be changed to  $c_{jkp} + \varepsilon$  by the mutation operation. Mutation at a parameter of a fuzzy membership function may, however, disrupt the order of the resulting fuzzy membership functions. These fuzzy membership functions then need rearrangement according to their values. An example is given below to demonstrate the mutation operation.

*Example 3:* Continuing from Example 1, assume the mutation point is set at  $c_{122}$  and the random value  $\varepsilon$  is set at 4. The mutation process is shown in Fig. 8.

V. THE PROPOSED MINING ALGORITHM

According to the above description, the proposed algorithm for mining both fuzzy association rules and membership functions based on the divide-and-conquer strategy is described next.

A. The Proposed Mining Algorithm

INPUT: A body of  $n$  quantitative transaction data, a set of  $m$  items, each with a number of predefined linguistic terms, a support threshold  $\alpha$ , a confidence threshold  $\lambda$ , and a population size  $P$ .

OUTPUT: A set of fuzzy association rules with its associated set of membership functions.

STEP 1: Randomly generate  $m$  populations, each for an item; each individual in a population represents a possible set of membership functions for that items.

STEP 2: Encode each set of membership functions into a string representation.

STEP 3: Calculate the fitness value of each chromosome in each population by the following substeps.

STEP 3.1: For each transaction datum  $D_i, i = 1$  to  $n$ , and for each item  $I_j, j = 1$  to  $m$ , transfer the quantitative value  $v_j^{(i)}$  into a fuzzy set  $f_j^{(i)}$  represented as

$$\left( \frac{f_{j1}^{(i)}}{R_{j1}} + \frac{f_{j2}^{(i)}}{R_{j2}} + \dots + \frac{f_{jl}^{(i)}}{R_{jl}} \right)$$

using the corresponding membership functions represented by the chromosome, where  $R_{jk}$  is the  $k$ th fuzzy region (term) of item  $I_j, f_{jl}^{(i)}$  is  $v_j^{(i)}$ 's fuzzy membership value in region  $R_{jk}$ , and  $l(= |I_j|)$  is the number of linguistic terms for  $I_j$ .

STEP 3.2: For each item region  $R_{jk}$ , calculate its scalar cardinality on the transactions as follows:

$$\text{count}_{jk} = \sum_{i=1}^n f_{jk}^{(i)}.$$

STEP 3.3: For each  $R_{jk}, 1 \leq j \leq m$  and  $1 \leq k \leq |I_j|$ , check whether its  $\text{count}_{jk}$  is larger than or equal to the minimum support threshold  $\alpha$ . If  $R_{jk}$  satisfies the above condition, put it in the set of large 1-itemsets ( $L_1$ ). That is

$$L_1 = \{R_{jk} | \text{count}_{jk} \geq \alpha, 1 \leq j \leq m \text{ and } 1 \leq k \leq |I_j|\}.$$

STEP 3.4: Set the fitness value of the chromosome as the sum of the fuzzy supports (the sum of fuzzy values/ $n$ ) of the fuzzy regions in  $L_1$  divided by suitability( $C_q$ ).

That is

$$f(C_q) = \frac{\sum_{X \in L_1} \text{fuzzy\_support}(X)}{\text{suitability}(C_q)}.$$

STEP 4: Execute crossover operations on each population.

STEP 5: Execute mutation operations on each population.

STEP 6: Using the selection operation to choose individuals in each population for the next generation. Any selection operation, such as the elitism selection strategy or the roulette selection strategy may be used here.

TABLE I  
SIX TRANSACTIONS IN THIS EXAMPLE

TID	Items
T1	(milk, 5), (bread, 10), (cookies, 7), (beverage, 7).
T2	(milk, 7), (bread, 14), (cookies, 12).
T3	(bread, 15), (cookies, 12); (beverage, 10).
T4	(milk, 2); (bread, 5); (cookies, 5).
T5	(bread, 9).
T6	(milk, 13), (beverage, 12).

STEP 7: If the termination criterion is not satisfied, go to Step 3; otherwise, do the next step. The termination criterion may be number of iterations, allowed execution time or convergence of the fitness values.

STEP 8: Gather the sets of membership functions, each of which has the highest fitness value in its population.

The sets of best membership functions gathered from all the populations are then used to mine fuzzy association rules from the given quantitative database. Our previous fuzzy mining algorithm proposed in [13] is then adopted to achieve this purpose.

VI. AN EXAMPLE

In this section, an example is given to illustrate the proposed mining algorithm. This is a simple example to show how the proposed algorithm can be used to mine membership functions and fuzzy association rules from quantitative data. Assume there are four items in a transaction database: milk, bread, cookies, and beverage. The data set includes the six transactions shown in Table I.

Assume each item has three fuzzy regions: *Low*, *Middle*, and *High*. Thus, three fuzzy membership functions must be derived for each item. For the data shown in Table I, the proposed mining algorithm proceeds as follows.

STEP 1: Four populations are randomly generated, each for one item. Assume the population size is ten in this example. Each population then includes ten individuals. Each individual in the first population is a set of membership functions for item *milk*. Similarly, an individual in the other populations is a set of membership functions, respectively, for bread, cookies, and beverage.

STEP 2: Each set of membership functions for an item is encoded into a chromosome according to the proposed representation. Assume the ten individuals in the four populations are randomly generated, as shown in Table II.

STEP 3: The fitness value of each chromosome is then calculated by the following substeps. Take the chromosome  $C_1$  in *Population*<sub>1</sub> as an example. The membership functions in  $C_1$  for *milk* are represented as (0 5 10 7 13 16 15 18 18).

STEP 3.1: The quantitative value of each item in each transaction datum is transformed into a fuzzy set according to the membership functions represented by that chromosome. Take the first item in transaction  $T1$  as an example. The contents of  $T1$  include (milk, 5), (bread, 10), (cookies, 7), and (beverage, 7). The amount "5" of item *milk* is then converted into the fuzzy set ((1/milk.Low) + (0/milk.Middle) + (0/milk.High)) using the membership functions in  $C_1$  in *Population*<sub>1</sub>. The results for all the transactions by using chromosome  $C_1$  in *Population*<sub>1</sub>

TABLE II  
TEN CHROMOSOMES IN EACH OF THE FOUR POPULATIONS

Population <sub>1</sub> ( <i>milk</i> )		Population <sub>2</sub> ( <i>bread</i> )	
C <sub>1</sub>	0 5 10, 7 13 16, 15 18 18	C <sub>1</sub>	0 6 12, 6 12 18, 12 18 18
C <sub>2</sub>	3 5 10, 6 13 16, 15 20 20	C <sub>2</sub>	0 4 6, 4 6 10, 6 10 10
C <sub>3</sub>	3 6 10, 6 13 16, 12 20 20	C <sub>3</sub>	4 5 9, 8 10 15, 9 16 16
C <sub>4</sub>	0 4 8, 6 13 16, 12 20 20	C <sub>4</sub>	0 3 8, 10 15 16, 17 20 20
C <sub>5</sub>	4 5 6, 8 12 16, 15 20 20	C <sub>5</sub>	0 3 8, 10 12 16, 12 16 16
C <sub>6</sub>	0 10 20, 8 12 18, 9 20 20	C <sub>6</sub>	0 5 10, 5 10 15, 10 15 15
C <sub>7</sub>	3 6 10, 6 13 16, 12 15 15	C <sub>7</sub>	4 6 10, 8 10 15, 15 20 20
C <sub>8</sub>	0 5 10, 7 15 18, 15 20 20	C <sub>8</sub>	3 6 9, 6 9 12, 9 12 12
C <sub>9</sub>	3 8 12, 9 15 18, 16 20 20	C <sub>9</sub>	0 8 10, 10 15 20, 15 20 20
C <sub>10</sub>	3 5 8, 6 13 16, 12 20 20	C <sub>10</sub>	2 10 16, 8 15 20, 10 20 20
Population <sub>3</sub> ( <i>cookies</i> )		Population <sub>4</sub> ( <i>beverage</i> )	
C <sub>1</sub>	3 5 10, 6 13 16, 15 20 20	C <sub>1</sub>	4 5 6, 8 12 16, 15 20 20
C <sub>2</sub>	3 6 10, 6 13 16, 12 20 20	C <sub>2</sub>	0 10 20, 8 12 18, 9 20 20
C <sub>3</sub>	0 4 8, 6 13 16, 12 20 20	C <sub>3</sub>	3 6 10, 6 13 16, 12 15 15
C <sub>4</sub>	4 5 6, 8 12 16, 15 20 20	C <sub>4</sub>	3 6 9, 6 9 12, 9 12 12
C <sub>5</sub>	0 3 8, 10 12 16, 12 16 16	C <sub>5</sub>	0 8 10, 10 15 20, 15 20 20
C <sub>6</sub>	0 5 10, 5 10 15, 10 15 15	C <sub>6</sub>	2 10 16, 8 15 20, 10 20 20
C <sub>7</sub>	4 6 10, 8 10 15, 15 20 20	C <sub>7</sub>	3 5 10, 6 13 16, 15 20 20
C <sub>8</sub>	0 6 10, 6 10 15, 10 15 15	C <sub>8</sub>	3 6 10, 6 13 16, 12 20 20
C <sub>9</sub>	2 5 7, 8 10 12, 14 16 16	C <sub>9</sub>	0 4 8, 6 13 16, 12 20 20
C <sub>10</sub>	0 5 10, 5 10 15, 10 15 15	C <sub>10</sub>	0 5 10, 5 10 15, 10 15 15

are shown in Table III, where the notation *item.term* is called a fuzzy region.

STEP 3.2: The scalar cardinality of each fuzzy region in the transactions is calculated as the *count* value. Take the fuzzy region *milk.Low* as an example. Its scalar cardinality =  $(1.0 + 0.6 + 0.0 + 0.4 + 0.0 + 0.0) = 2.0$ . The counts of the fuzzy regions for item *milk* using  $C_1$  are shown in Table IV.

STEP 3.3: The count of any fuzzy region is checked against the predefined minimum support value  $\alpha$ . Assume in this example,  $\alpha$  is set at 0.25. Since only the count value of *milk.Low* is larger than  $0.25 * 6 (= 1.5)$ , *milk.Low* is then put in  $L_1$ .

STEP 3.4: Only one large 1-itemset, *milk.Low*, is derived from the membership functions of  $C_1$  in Population<sub>1</sub>. The fuzzy support of *milk.Low* is  $2/6 (= 0.33)$  and its suitability is calculated as 1. The fitness value of  $C_1$  is, thus,  $0.33/1 (= 0.33)$ . The fitness values of all the chromosomes in the four populations are calculated with their results shown in Table V.

STEP 4: The crossover operator is executed on the populations. Assume  $d$  is set at 0.35. Take  $C_1$  and  $C_3$  in Population<sub>1</sub> as an example. The following four candidate offspring chromosomes are generated:

- $C_1$  : 0 5 10, 7 13 16, 15 18 18;
- $C_3$  : 3 6 10, 6 13 16, 12 20 20;
- 1)  $C_1^{t+1}$  : 1.95 5.65 10, 6.35 13 16, 13.05 19.3 19.3;
- 2)  $C_2^{t+1}$  : 1.05 5.35 10, 6.65 13 16, 13.95 18.7 18.7;
- 3)  $C_3^{t+1}$  : 0 5 10, 6 13 16, 12 18 18;
- 4)  $C_4^{t+1}$  : 3 6 10, 7 13 16, 15 20 20.

The fitness value of the above four candidates are then evaluated, with results shown in Table VI.

In Table VI, the best two of the four candidate offspring chromosomes are chosen. Thus,  $C_2^{t+1}$  and  $C_3^{t+1}$  are chosen.

STEP 5: The *mutation operator* is executed to generate possible offspring. The operation is the same as the traditional one except that rearrangement may need to be done.

STEP 6: Assume the elitism selection strategy is used here. The best ten chromosomes in each population are, thus, selected as the next generation.

STEP 7: Assume the number of generations is used as the termination criterion. The same procedure is then executed until the predefined number of generations is achieved.

STEP 8: The best chromosome (with the highest fitness value) in each population is then output as the membership functions for deriving fuzzy association rules. Assume the final individuals in the four populations after the evolutionary process terminates are as shown in Table VII.

According to Table VII, the best individuals in the four populations are  $C_8$  in Population<sub>1</sub>,  $C_1$  in Population<sub>2</sub>,  $C_1$  in Population<sub>3</sub>, and  $C_8$  in Population<sub>4</sub>. The final set of membership functions for each item is shown in Fig. 9.

After the membership functions are derived, the fuzzy mining method proposed in [13] is then used to mine fuzzy association rules.

## VII. EXPERIMENTAL RESULTS

In this section, experiments conducted to show the performance of the proposed approach are described. They were implemented in Java on an Intel Pentium IV personal computer with 2.00 GHz and 256 MB RAM. The initial population size  $P$  was set at 50, the crossover rate  $p_c$  was set at 0.8, and the mutation rate  $p_m$  was set at 0.01 according to [25]. The parameter  $d$  of the crossover operator was set at 0.35 according to [8], the parameter  $\epsilon$  of the mutation operator was set at 3, and the minimum support  $\alpha$  is set at 0.04 (4%).

In the following subsections, we first give a description of the experimental datasets. We then analyze the membership functions according to the fitness function. Comparisons of uniform and nonuniform fuzzy partitions are then made to show the performance of the proposed algorithm. Finally, the analysis of the association rules via supports and confidences is made.

### A. Description of the Experimental Datasets

Simulated datasets with 64 items and with different dataset sizes from 10 to 90 k transactions were used in the experiments. The factors for datasets included the transaction length, the purchased items and their quantities. In the experiments, the number (transaction length) of purchased items in a transaction was randomly generated in a uniform distribution of the range of 1–19. The purchased items in each transaction were then selected from the 64 items in an exponential distribution with the rate parameter set at 16. Their quantities were then assigned from an exponential distribution with the rate parameter set at 5. The simulation process was terminated until the dataset size was reached. An item could not be generated twice in a transaction. The data generated for the experiments can be referred to in our website “<http://ailab.ee.nuk.edu.tw/data/index.htm>”.

TABLE III  
THE FUZZY SETS TRANSFORMED BY USING CHROMOSOME  $C_1$  IN Population<sub>1</sub>

	$T1$	$T2$	$T3$	$T4$	$T5$	$T6$
<b>Transformed Fuzzy Set</b>	$\left(\frac{1}{milk.Low}\right)$	$\left(\frac{0.6}{milk.Low}\right)$	Null	$\left(\frac{0.4}{milk.Low}\right)$	Null	$\left(\frac{1}{milk.Middle}\right)$

TABLE IV  
THE COUNTS OF THE FUZZY REGIONS FOR ITEM MILK WHEN USING  $C_1$

Population <sub>1</sub>	Count
<i>milk.Low</i>	2.00
<i>milk.Middle</i>	1.00
<i>milk.High</i>	0.00

TABLE V  
FITNESS VALUES OF ALL THE CHROMOSOMES IN THE FOUR INITIAL POPULATIONS

Population <sub>1</sub>	$f$	Population <sub>2</sub>	$f$
$C_1$	0.33	$C_1$	0.66
$C_2$	0.205	$C_2$	0.446
$C_3$	0	$C_3$	0.355
$C_4$	0.187	$C_4$	0.257
$C_5$	0	$C_5$	0
$C_6$	0.112	$C_6$	0.633
$C_7$	0	$C_7$	0.202
$C_8$	0.33	$C_8$	0.524
$C_9$	0	$C_9$	0.3
$C_{10}$	0	$C_{10}$	0.280
Population <sub>3</sub>	$f$	Population <sub>4</sub>	$f$
$C_1$	0.59	$C_1$	0.125
$C_2$	0.202	$C_2$	0.227
$C_3$	0.233	$C_3$	0.157
$C_4$	0.166	$C_4$	0
$C_5$	0.277	$C_5$	0
$C_6$	0.533	$C_6$	0.150
$C_7$	0	$C_7$	0.196
$C_8$	0.263	$C_8$	0.157
$C_9$	0	$C_9$	0.196
$C_{10}$	0.533	$C_{10}$	0.33

TABLE VI  
FITNESS VALUE OF THE FOUR CANDIDATE OFFSPRING CHROMOSOMES

chromosome	$f$	chromosome	$f$
$C_i^{t+1}$	0.216	$C_3^{t+1}$	0.250
$C_2^{t+1}$	0.239	$C_4^{t+1}$	0

*B. Analysis of the Membership Functions According to the Fitness Function*

After 500 generations, the final membership functions were apparently much better than the original ones. For example, the

initial membership functions of some four items among the 64 items are shown in Fig. 10.

In Fig. 10, the membership functions have the two bad types of shapes mentioned in the previous section. The membership functions for Item<sub>1</sub>, Item<sub>2</sub>, and Item<sub>4</sub> overlap too much. After 500 generations, the final membership functions for the same four items are shown in Fig. 11.

It is clear that the membership functions in Fig. 11 were better than those in Fig. 10. The two bad kinds of membership functions did not appear in the final results. Furthermore, it can also be observed that the center of a derived membership function is close to the upper limit of its left adjacent membership function and the lower limit of its right adjacent membership function. This is the effect of the suitability criterion for the membership functions. From the results, it may be found that using only one parameter for each membership function is another acceptable option for chromosome representation.

The average fitness values of the chromosomes for Item<sub>1</sub> along with different numbers of generations in the first population are shown in Fig. 12. As expected, the curve gradually went upward, finally converging to a fixed value. The average fitness values for the other items had similar shapes.

Next, to evaluate the proposed fitness function, experiments were made using suitability and fuzzy support separately as the fitness value. For the same experimental environments and data, the membership functions of the above four items after 500 generations, using only suitability as the fitness function are shown in Fig. 13, and using only the fuzzy\_support are shown in Fig. 14.

It can be easily seen from Fig. 13 that the derived membership functions considering only suitability were satisfactory because the suitability measure was designed for getting good shapes of membership functions. Its fuzzy\_support was, however, less than the original one (which will be shown later). By contrast, it was very natural for the derived membership functions by considering only fuzzy\_support to have a bad shape from Fig. 14. Their overlap degrees were quite high.

The sum of fuzzy\_support values in all large 1-itemsets by the original fitness function, by suitability alone, and by fuzzy support alone, were plotted against generation. The results are shown in Fig. 15.

It can easily be seen from Fig. 15 that the fuzzy\_support value obtained by considering only the suitability was the least among the three fitness functions. The suitability values by the three fitness functions along with different generations are then shown in Fig. 16.

From Fig. 16, the suitability value by the original fitness function is similar to that by only the suitability measure. Our proposed fitness function can, thus, achieve a good tradeoff between fuzzy\_support and suitability of membership functions.



TABLE VII  
FINAL INDIVIDUALS IN THE FOUR POPULATIONS

	Population <sub>1</sub> ( <i>milk</i> )	<i>f</i>		Population <sub>2</sub> ( <i>bread</i> )	<i>f</i>
C <sub>1</sub>	0 5 9, 7 13 16, 15 18 18	0.316	C <sub>1</sub>	<b>0 6 12, 6 12 18, 12 18 18</b>	<b>0.66</b>
C <sub>2</sub>	3 5 10, 6 13 16, 15 20 20	0.205	C <sub>2</sub>	0 4 6, 4 6 10, 6 10 10	0.446
C <sub>3</sub>	1.05 5.35 10, 6.65 13 16, 13.95 18.7 18.7	0.239	C <sub>3</sub>	4 5 9, 8 10 15, 9 16 16	0.355
C <sub>4</sub>	0 4 8, 6 13 16, 12 20 20	0.187	C <sub>4</sub>	0 3 8, 10 15 16, 17 20 20	0.257
C <sub>5</sub>	0 5 10, 6 13 16, 12 18 18	0.250	C <sub>5</sub>	0 3.65 6.7, 6.1 9.15 12.1, 9.85 13.5 13.5	0.625
C <sub>6</sub>	0 10 20, 8 12 18, 9 20 20	0.112	C <sub>6</sub>	0 5 10, 5 10 15, 10 15 15	0.633
C <sub>7</sub>	1.95 5.65 10, 6.35 13 16, 13.05 19.3 19.3	0.216	C <sub>7</sub>	0 4 8, 10 15 16, 17 20 20	0.235
C <sub>8</sub>	<b>0 5 10, 7 15 18, 15 20 20</b>	<b>0.33</b>	C <sub>8</sub>	3 6 9, 6 9 12, 9 12 12	0.524
C <sub>9</sub>	1 5 10, 6 13 16, 12 18 18	0.217	C <sub>9</sub>	0 8 10, 10 15 20, 15 20 20	0.3
C <sub>10</sub>	2 5 9, 6 13 16, 13 20 20	0.211	C <sub>10</sub>	2 10 16, 8 15 20, 10 20 20	0.280
	Population <sub>3</sub> ( <i>cookies</i> )	<i>f</i>		Population <sub>4</sub> ( <i>beverage</i> )	<i>f</i>
C <sub>1</sub>	<b>0 5 10, 6 13 16, 15 20 20</b>	<b>0.575</b>	C <sub>1</sub>	4 5 6, 8 12 16, 15 20 20	0.125
C <sub>2</sub>	3 6 10, 6 13 16, 12 20 20	0.202	C <sub>2</sub>	0 10 20, 8 12 18, 9 20 20	0.227
C <sub>3</sub>	0 4 8, 6 13 16, 12 20 20	0.233	C <sub>3</sub>	3 6 10, 6 13 16, 12 15 15	0.157
C <sub>4</sub>	1.05 5 10, 5.35 11.05 15.35, 11.05 16.75 16.75	0.5247	C <sub>4</sub>	1.95 6.75 13.5, 6.7 12.65 16.7, 12.9 20 20	0.255
C <sub>5</sub>	0 3 8, 10 12 16, 12 16 16	0.277	C <sub>5</sub>	1.05 8.25 16.5, 7.3 12.35 17.3, 11.1 20 20	0.270
C <sub>6</sub>	0 5 10, 5 10 15, 10 15 15	0.533	C <sub>6</sub>	0 5 10, 6 12 16, 9 20 20	0.174
C <sub>7</sub>	1.95 5 10, 5.65 12 16, 13 18 18	0.517	C <sub>7</sub>	3 5 10, 6 13 16, 15 20 20	0.196
C <sub>8</sub>	0 6 10, 6 10 15, 10 15 15	0.263	C <sub>8</sub>	<b>0 7 12.4 7.3 12 16 14 20 20</b>	<b>0.474</b>
C <sub>9</sub>	0 4 8, 10 15 16, 17 20 20	0.480	C <sub>9</sub>	0 4 8, 6 13 16, 12 20 20	0.196
C <sub>10</sub>	0 5 10, 5 10 15, 10 15 15	0.533	C <sub>10</sub>	0 5 10, 5 10 15, 10 15 15	0.33

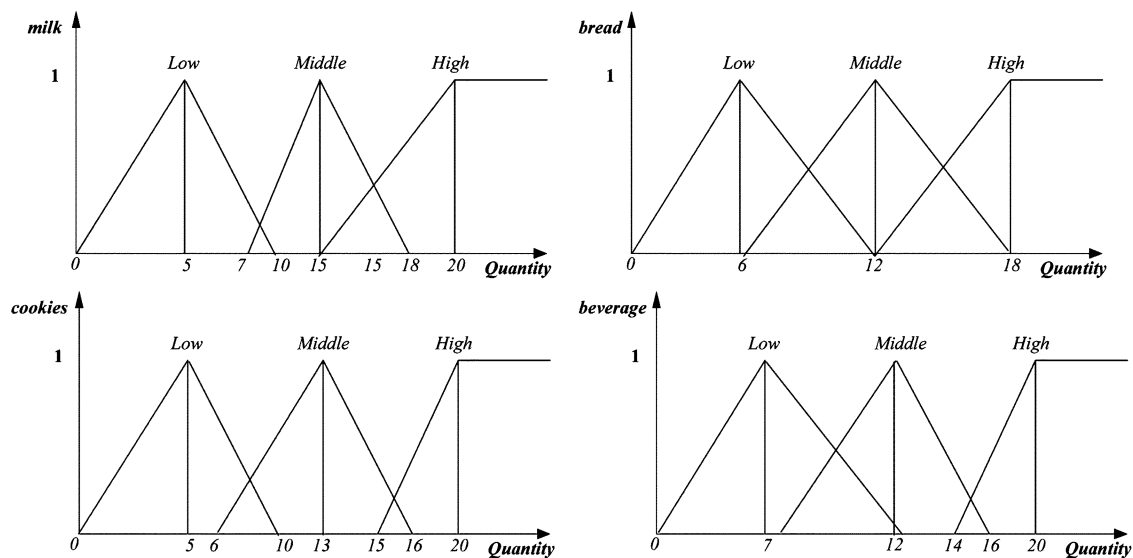


Fig. 9. The final set of membership functions for each item.

Experiments were then made to show the scalability of the proposed approach. The relationship between the execution time and the database size is shown in Fig. 17.

It is clear from Fig. 17 that the execution time of the proposed algorithm increased linearly with the increase of dataset size. The proposed approach thus has a good scalability property

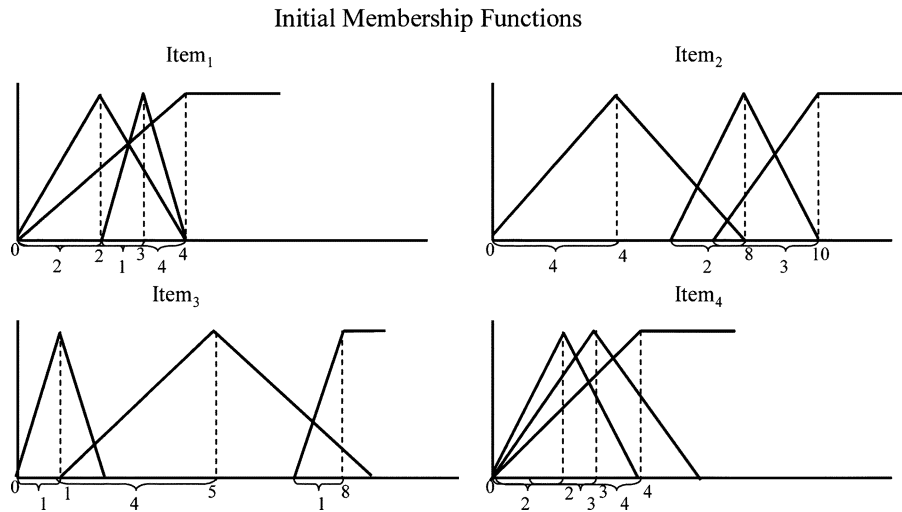


Fig. 10. The initial membership functions of some four items.

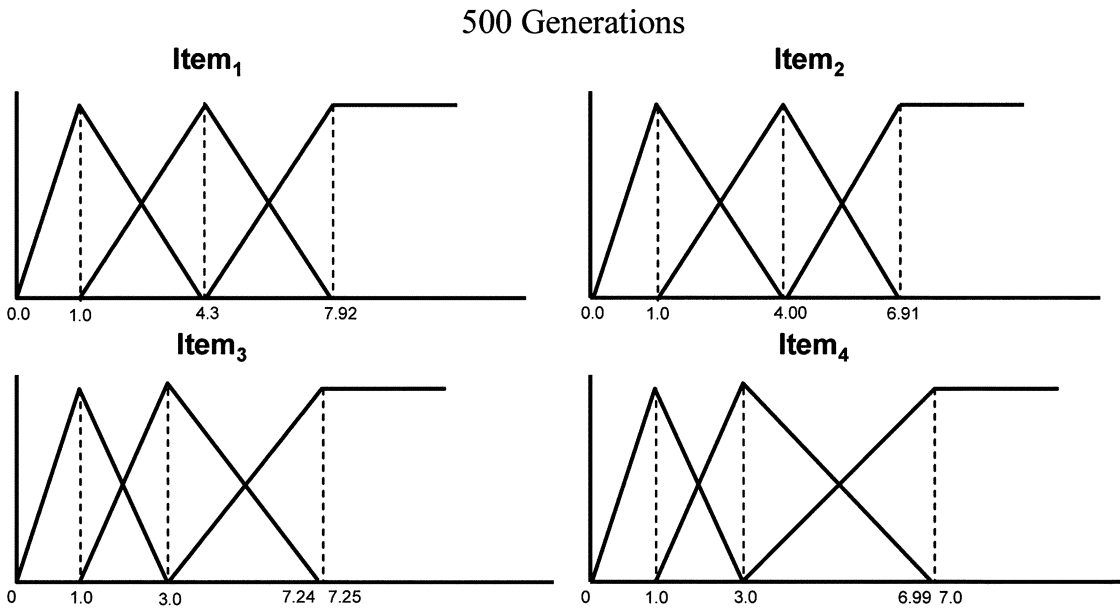


Fig. 11. The final membership functions of some four items after 500 generations.

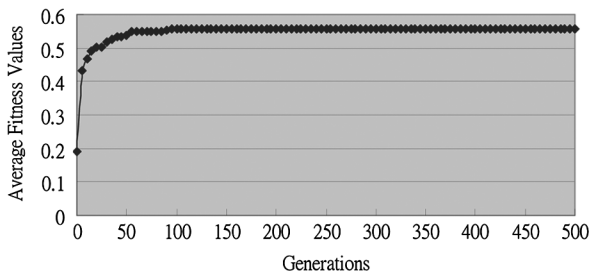


Fig. 12. The average fitness values along with different numbers of generations.

due to the linearity. Besides, the mining procedure is often performed offline and can even use sampling techniques to speed up execution when the data size is very large.

*C. Comparisons of Uniform and Nonuniform Fuzzy Partitions*

Next, experiments were conducted to provide a comparative analysis of the proposed approach with the fuzzy mining approach in [13] with a uniform fuzzy partition. Since the range of the dataset in our experiments fell in the interval 1–12, the membership functions shown in Fig. 18 were, thus, used for a uniform fuzzy partition.

The relationship between the numbers of large 1-itemsets and the minimum supports along with different minimum confidences by the proposed approach and the one with the uniform fuzzy partition is shown in Fig. 19.

It can be observed from Fig. 19 that the number of large 1-itemsets derived by the proposed algorithm was larger than the one with the uniform fuzzy partition. It was consistent with the previous discussion since the fitness function used could help the proposed approach generate more large 1-itemsets.

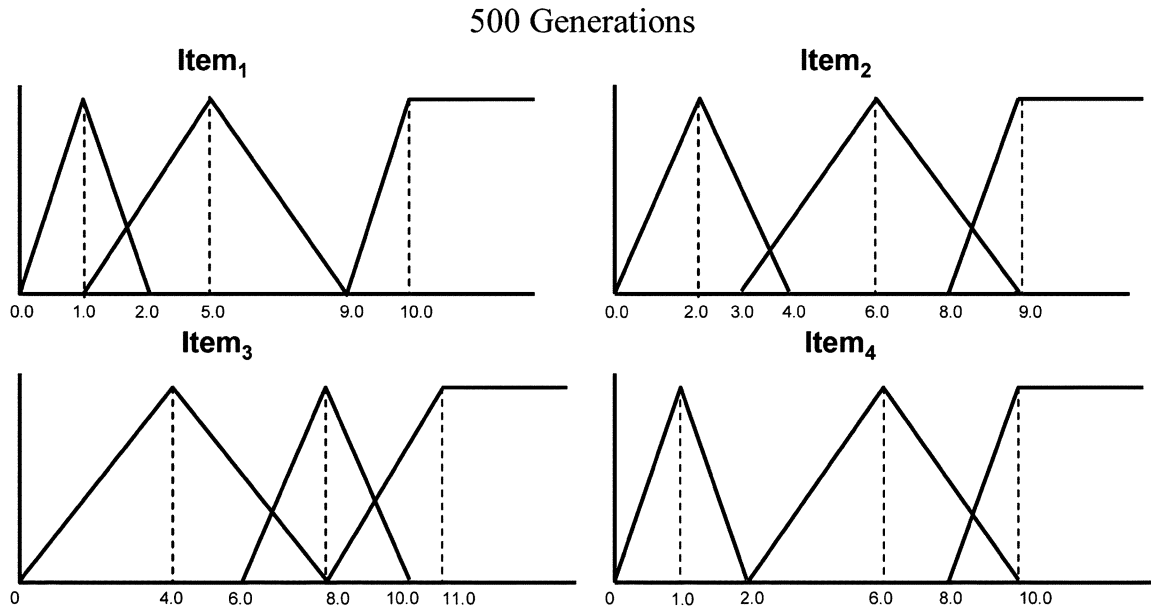


Fig. 13. The final membership functions when only suitability is considered.

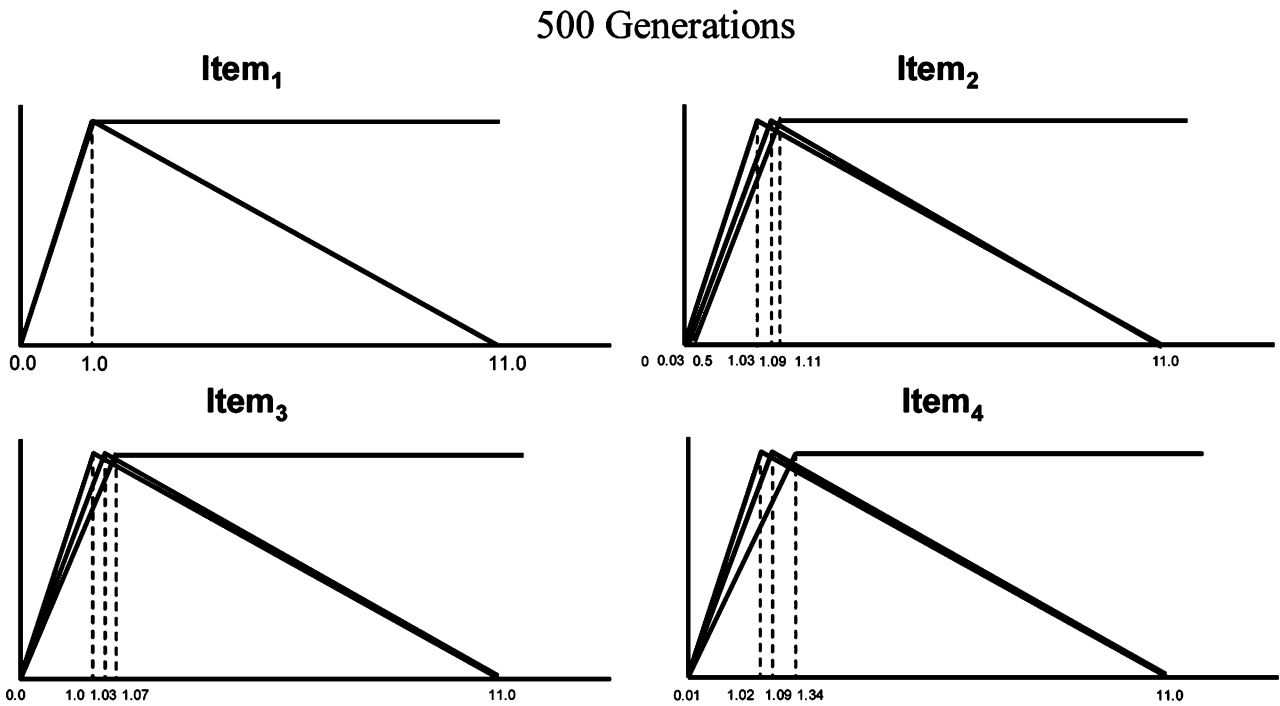


Fig. 14. The final membership functions when only fuzzy\_support is considered.

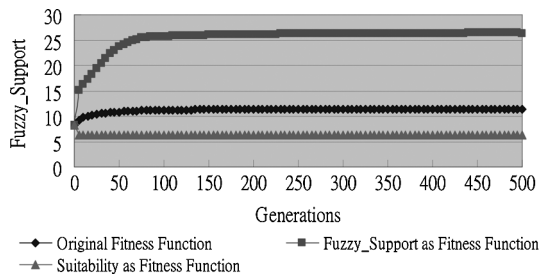


Fig. 15. The sum of fuzzy\_support values by the three different fitness functions.

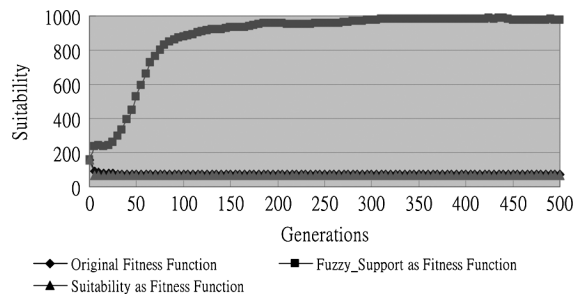


Fig. 16. The suitability value by the three different fitness functions.

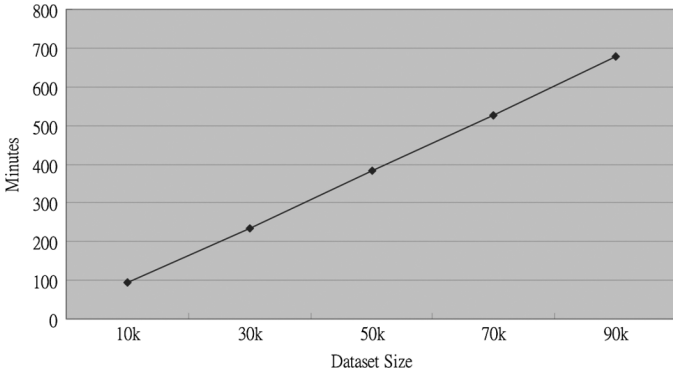


Fig. 17. The relationship between the execution time and the database size.

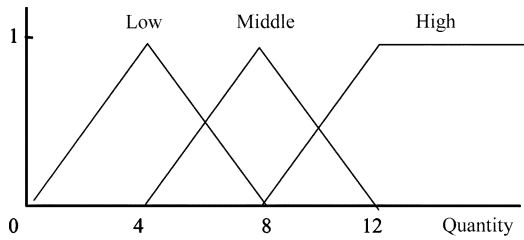


Fig. 18. The membership functions used for a uniform fuzzy partition.

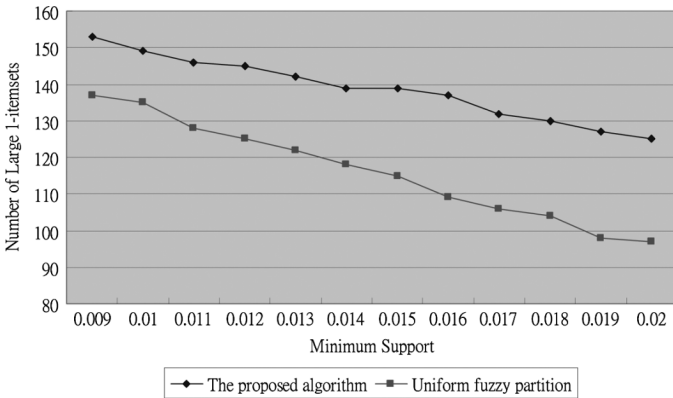


Fig. 19. The relationship between the numbers of large 1-itemsets and the minimum supports for the two approaches.

Finally, experiments were made for a comparison of the proposed algorithm and the existing one without divide-and-conquer strategy, which maintains a population of sets of membership functions (membership functions of all items are encoded into one chromosome) and uses the GA to automatically derive the resulting one [10]. The relationship between the numbers of large 1-itemsets and the minimum supports is shown in Fig. 20.

It can be observed from Fig. 20 that the proposed approach could generate more large 1-itemsets than the existing one without divide-and-conquer strategy [10]. The mean and the variance values of the number of large 1-itemsets and execution time in ten runs with different random seeds for the two approaches were shown in Table VIII.

The experimental results for convergence speeds are shown in Fig. 21.

It can be observed from Fig. 21 that the fitness value of the proposed algorithm converged after about 100 generations, but

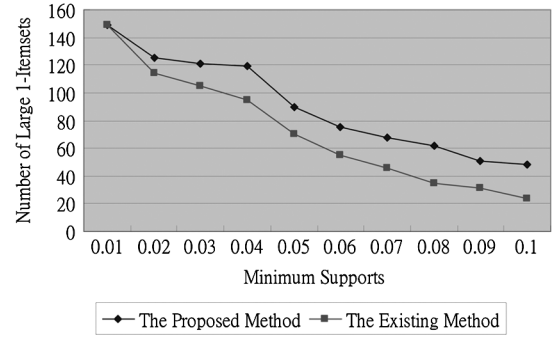


Fig. 20. The relationship between the numbers of large 1-itemsets and the minimum supports for the two approaches.

TABLE VIII  
THE MEAN AND THE VARIANCE VALUES OF THE NUMBER OF LARGE 1-ITEMSETS AND EXECUTION TIME (MINUTES) FOR THE TWO APPROACHES

	Mean	Variance
The proposed approach	114.5 (94.04)	2.27 (2.12)
The existing approach without divide-and-conquer strategy	88.1 (118.14)	3.21 (0.66)

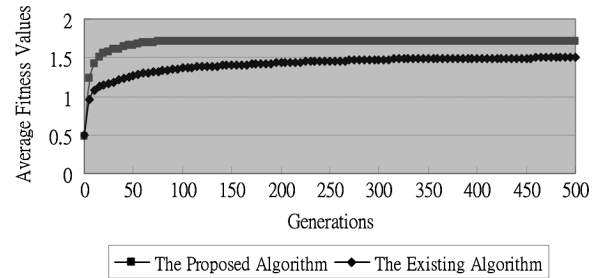


Fig. 21. The comparison of the convergence of the proposed algorithm and the existing one.

the existing one without divide-and-conquer strategy converged after about 350 generations. The proposed divide-and-conquer strategy can, thus, generate good effects and speed the convergence since each item can be independently processed under the adopted coding scheme and evaluation function.

#### D. Analysis of the Association Rules Via Supports and Confidences

Experiments were then made to provide an analysis of the association rules via supports and confidences. Fig. 22 shows the relationship between the numbers of association rules derived by the membership functions found in the first phase and the minimum supports along with different minimum confidences.

It can be observed from Fig. 22 that the number of rules decreased along with the increase of the minimum support values. Besides, the curve with a large minimum confidence value was smoother than those with a small value.

The relationship between the numbers of association rules derived by the membership functions found in the first phase and the minimum confidences along with different minimum supports is shown in Fig. 23.

It can be observed from Fig. 23 that the number of association rules decreased along with the increase of the minimum confi-

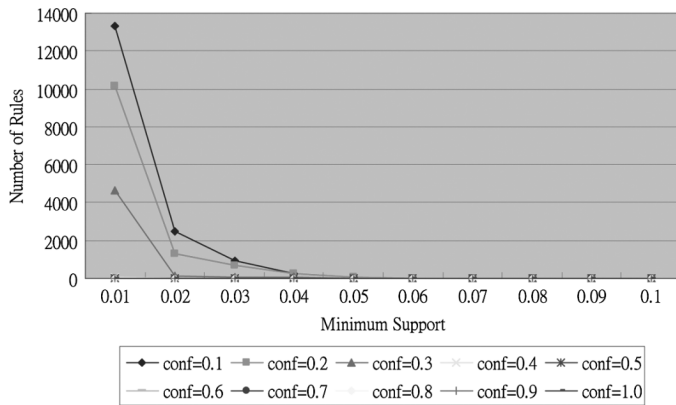


Fig. 22. The relationship between the numbers of association rules and the minimum supports along with different minimum confidences.

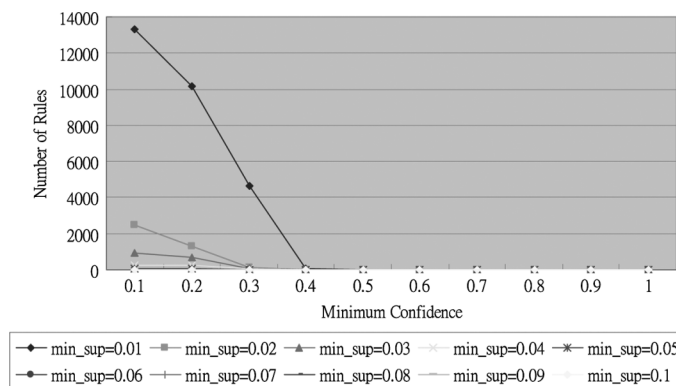


Fig. 23. The relationship between the numbers of association rules and the minimum confidences along with different minimum supports.

dence values. Besides, the curve with a large minimum support value was smoother than those with a small value, meaning that the minimum confidence value had a larger effect on the number of association rules when smaller minimum support values were used. Note that all of the curves approximated to 0 as the minimum confidence value approached 1. The results also showed that it was a tradeoff between the number of rules and the minimum supports (or confidences). When too many rules are derived, it may be difficult for humans to use the knowledge. In this case, it may be appropriate to consider a multiobjective approach to find a smaller number of Pareto-optimal rules [19], [15]. Some discussions of multiobjective rule selection after fuzzy data mining can also be found in [3], [4], [16].

### VIII. CONCLUSION AND FUTURE WORKS

In this paper, we have proposed a GA-based fuzzy data-mining algorithm for extracting both association rules and membership functions from quantitative transactions. Since the fitness of each set of membership functions is evaluated by fuzzy-support values in large 1-itemsets and by suitability of derived membership functions, the derivation process can easily be done using the divide-and-conquer strategy. We have thus proposed a mining framework, which maintains multiple populations, each for an item's membership functions. The final best sets of membership functions gathered from all the populations are used to mine fuzzy association rules. The experimental

results have also shown that the proposed genetic-fuzzy mining algorithm can get a good tradeoff between fuzzy supports of large 1-itemsets and suitability values of membership functions.

Note that if the density of values focuses on a certain interval, the proposed approach can still work. The generated membership functions will have their centers gathered around the interval. The leftmost and the rightmost membership functions may extend to the two end points of the possible range. The membership value, thus, becomes smaller when an item quantity is closer to the end points of the possible range.

Although the proposed divide-and-conquer method has fast convergence speed, its disadvantage is that only the supports of large 1-itemsets are considered. If itemsets with more than one item are considered, then the proposed approach may need to be greatly modified. In the future, we will continuously attempt to enhance the GA-based mining framework for more complex problems.

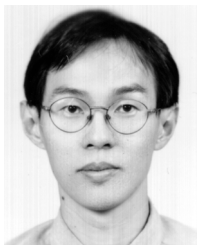
### ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their constructive comments and thank C.-S. Chen for his help in making some of the experiments.

### REFERENCES

- [1] R. Agrawal and R. Srikant, "Fast algorithm for mining association rules," in *Proc. Int. Conf. Very Large Databases*, 1994, pp. 487–499.
- [2] W. H. Au, K. C. C. Chan, and X. Yao, "A novel evolutionary data mining algorithm with applications to churn prediction," *IEEE Trans. Evol. Comput.*, vol. 7, no. 6, pp. 532–545, Dec. 2003.
- [3] J. Casillas, O. Cordon, F. Herrera, and L. Magdalena, *Interpretability Issues in Fuzzy Modeling*. New York: Springer, 2003.
- [4] J. Casillas, O. Cordon, F. Herrera, and L. Magdalena, *Accuracy Improvements in Linguistic Fuzzy Modeling*. New York: Springer, 2003.
- [5] C. H. Cai, W. C. Fu, C. H. Cheng, and W. W. Kwong, "Mining association rules with weighted items," in *Proc. Int. Database Eng. Appl. Symp.*, 1998, pp. 68–77.
- [6] O. Cordón, F. Herrera, and P. Villar, "Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the data base," *IEEE Tran. Fuzzy Systems*, vol. 9, no. 4, 2001.
- [7] P. J. Darwen and X. Yao, "Speciation as automatic categorical modularization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 2, pp. 101–108, Jul. 1997.
- [8] F. Herrera, M. Lozano, and J. L. Verdegay, "Fuzzy connectives based crossover operators to model genetic algorithms population diversity," *Fuzzy Sets Syst.*, vol. 92, no. 1, pp. 21–30, 1997.
- [9] T. P. Hong and J. B. Chen, "Processing individual fuzzy attributes for fuzzy rule induction," *Fuzzy Sets Syst.*, vol. 112, no. 1, pp. 127–140, 2000.
- [10] T. P. Hong, C. H. Chen, Y. L. Wu, and Y. C. Lee, "A GA-based fuzzy mining approach to achieve a trade-off between number of rules and suitability of membership functions," *Soft Comput.*, vol. 10, no. 11, pp. 1091–1101, 2006.
- [11] R. H. Hou, T. P. Hong, S. S. Tseng, and S. Y. Kuo, "A new probabilistic induction method," *J. Autom. Reasoning*, vol. 18, pp. 5–24, 1997.
- [12] T. P. Hong, C. S. Kuo, and S. C. Chi, "Mining association rules from quantitative data," *Intell. Data Anal.*, vol. 3, no. 5, pp. 363–376, 1999.
- [13] T. P. Hong, C. S. Kuo, and S. C. Chi, "Trade-off between time complexity and number of rules for fuzzy mining from quantitative data," *Int. J. Uncertainty, Fuzziness and Knowledge-Based Syst.*, vol. 9, no. 5, pp. 587–604, 2001.
- [14] T. P. Hong and C. Y. Lee, "Induction of fuzzy rules and membership functions from training examples," *Fuzzy Sets Syst.*, vol. 84, pp. 33–47, 1996.
- [15] H. Ishibuchi and T. Yamamoto, "Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining," *Fuzzy Sets Syst.*, vol. 141, pp. 59–88, 2004.
- [16] Y. Jin, *Multi-Objective Machine Learning*. New York: Springer, 2006.
- [17] A. Kandel, *Fuzzy Expert Systems*. Boca Raton, FL: CRC, 1992, pp. 8–19.

- [18] M. Kaya and R. Alhaji, "Integrating multi-objective genetic algorithms into clustering for fuzzy association rules mining," in *Proc. 4th IEEE Int. Conf. Data Mining*, 2004, pp. 431–434.
- [19] M. Kaya and R. Alhaji, "A clustering algorithm with genetically optimized membership functions for fuzzy association rules mining," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 2003, pp. 881–886.
- [20] V. R. Khare, X. Yao, B. Sendhoff, Y. Jin, and H. Wersing, "Co-evolutionary modular neural networks for automatic problem decomposition," *Proc. IEEE Congr. Evol. Comput.*, vol. 3, pp. 2691–2698, 2005.
- [21] Y. C. Lee, T. P. Hong, and W. Y. Lin, "Mining fuzzy association rules with multiple minimum supports using maximum constraints," *Lecture Notes in Computer Science*, vol. 3214, pp. 1283–1290, 2004.
- [22] A. Parodi and P. Bonelli, "A new approach of fuzzy classifier systems," in *Proc. 5th Int. Conf. Genetic Algorithms*, 1993, pp. 223–230.
- [23] H. Roubos and M. Setnes, "Compact and transparent fuzzy models and classifiers through iterative complexity reduction," *IEEE Trans. Fuzzy Syst.*, vol. 9, no. 4, pp. 516–524, Aug. 2001.
- [24] M. Setnes and H. Roubos, "GA-fuzzy modeling and classification: Complexity and performance," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 5, pp. 509–522, Oct. 2000.
- [25] M. Srinivas and L. M. Patnaik, "Genetic algorithms: A survey," *Computer*, vol. 27, no. 6, pp. 17–26, 1994.
- [26] C. H. Wang, T. P. Hong, and S. S. Tseng, "Integrating fuzzy knowledge by genetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 2, no. 4, pp. 138–149, 1998.
- [27] C. H. Wang, T. P. Hong, and S. S. Tseng, "Integrating membership functions and fuzzy rule sets from multiple knowledge sources," *Fuzzy Sets Syst.*, vol. 112, pp. 141–154, 2000.
- [28] X. Yao, "Adaptive divide-and-conquer using populations and ensembles," in *Proc. Int. Conf. Mach. Learn. Appl.*, 2003, pp. 13–20.
- [29] S. Yue, E. Tsang, D. Yeung, and D. Shi, "Mining fuzzy association rules with weighted items," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2000, pp. 1906–1911.



**Tsung-Pei Hong** received the B.S. degree in chemical engineering from the National Taiwan University, Taipei, in 1985, and the Ph.D. degree in computer science and information engineering from the National Chiao-Tung University, Hsinchu, Taiwan, in 1992.

From 1987 to 1994, he was with the Laboratory of Knowledge Engineering, National Chiao-Tung University, where he was involved in applying techniques of parallel processing to artificial intelligence. He was an Associate Professor with the

Department of Computer Science, Chung-Hua Polytechnic Institute from 1992 to 1994, and with the Department of Information Management, I-Shou University (originally Kaohsiung Polytechnic Institute) from 1994 to 1999. He was a Professor at I-Shou University from 1999 to 2001. He was in charge of the computerization and library planning for the National University of Kaohsiung in Preparation from 1997 to 2000, and served as the first Director of the Library and Computer Center, National University of Kaohsiung from 2000 to 2001, and as the Dean of Academic Affairs from 2003 to 2006. He is currently a Professor with the Department of Electrical Engineering and the Department of Computer Science and Information Engineering. He is also the Vice President of the National University of Kaohsiung since 2007. His current research interests include parallel processing, machine learning, data mining, soft computing, management information systems, and www applications.

Dr. Hong is a member of the Association for Computing Machinery, the Chinese Fuzzy Systems Association, the Taiwanese Association for Artificial Intelligence, and the Institute of Information and Computing Machinery.



**Chun-Hao Chen** received the B.S. and M.S. degrees from the Department of Information Management, I-Shou University, Kaohsiung, Taiwan, R.O.C., in 2002 and 2004, respectively. He is currently working towards the Ph.D. degree in computer science and information engineering at the National Cheng Kung University, Tainan, Taiwan.

His research interests include data mining, genetic algorithms, time series, and fuzzy theory.



**Yeong-Chyi Lee** received the B.S. and M.S. degrees from the Department of Information Management, I-Shou University, Kaohsiung, Taiwan, R.O.C., in 1998 and 2000, respectively. He is currently working towards the Ph.D. degree at the Department of Information Engineering, I-Shou University.

His research interests include data mining, genetic algorithms, and fuzzy theory.



**Yu-Lung Wu** received the B.S. degree in applied MIS from Takushoku University, Tokyo, Japan, in 1995 and the Ph.D. degree in applied MIS from Takushoku University in 2000.

He is now an Assistant Professor with the Department of Information Management, I-Shou University, Kaohsiung, Taiwan, R.O.C., since 2000, and Supervisor in the Extension Education Center of I-Shou University since 2003. His current research interests include electronic commerce, semiconductor industry, knowledge management, and IT outsourcing.

Dr. Wu is a member of the Japan Association Management System, the Chinese Information Management Association, and the Electronic Business Management Association.