International Journal of Computational Intelligence Systems, Vol. 5, No. 2 (April, 2012), 343-354

Finding Pareto-front Membership Functions in Fuzzy Data Mining

Chun-Hao Chen
Department of Computer Science and Information Engineering
Tamkang University
Taipei, 251, Taiwan, R.O.C.
chchen@mail.tku.edu.tw

Tzung-Pei Hong
(Corresponding Author)
Department of Computer Science and Information Engineering
National University of Kaohsiung
Kaohsiung, 811, Taiwan, R.O.C.
Department of Computer Science and Engineering
National Sun Yat-sen University
Kaohsiung, 80424, Taiwan, R.O.C.
tphong@nuk.edu.tw

Vincent S. Tseng
Department of Computer Science and Information Engineering
National Cheng Kung University
Tainan, 701, Taiwan, R.O.C.
tsengsm@mail.ncku.edu.tw

Received 15 December 2010

Accepted 1 June 2011

Abstract

Transactions with quantitative values are commonly seen in real-world applications. Fuzzy mining algorithms have thus been developed recently to induce linguistic knowledge from quantitative databases. In fuzzy data mining, the membership functions have a critical influence on the final mining results. How to effectively decide the membership functions in fuzzy data mining thus becomes very important. In the past, we proposed a fuzzy mining approach based on the Multi-Objective Genetic Algorithm (MOGA) to find the Pareto front of the desired membership functions. In this paper, we adopt a more sophisticated multi-objective approach, the SPEA2, to find the appropriate sets of membership functions for fuzzy data mining. Two objective functions are used to find the Pareto front. The first one is the suitability of membership functions and the second one is the total number of large 1-itemsets derived. Experimental comparisons of the proposed and the previous approaches are also made to show the effectiveness of the proposed approach in finding the Pareto-front membership functions.

Keywords: multi-objective optimization, genetic algorithm, fuzzy set, fuzzy association rules, data mining, Pareto front.

1. Introduction

Data mining is the process of extracting desirable knowledge or interesting patterns from existing databases for specific purposes [6]. Many types of knowledge and technology have been proposed for data mining. Among them, finding association rules from transaction data is most commonly seen. Most of the existing approaches handle items with binary values. Transactions with quantitative values are, however, commonly seen in real-world applications. Many

sophisticated data-mining approaches have thus been proposed in this research field [4, 31, 34].

As to fuzzy data mining, many approaches have also been proposed for mining fuzzy association rules [8, 20, 25, 27, 28, 33]. The fuzzy mining algorithms developed earlier were mainly based on the Apriori algorithm and some based on the FP trees was recently developed. Fuzzy mining algorithms with multiple minimum supports of different items were developed as well [28].

In fuzzy data mining, the membership functions have a critical influence on the final mining results. How to effectively decide the membership functions in fuzzy data mining thus becomes very important. Most of the previous fuzzy data mining algorithms assume the membership functions are already known. Pre-defined membership functions are not, however, actually suitable in usage. Mining algorithms that can automatically derive both the appropriate membership functions and the fuzzy rules are thus required. Many approaches have thus been proposed for deriving membership functions [10, 11, 18, 19, 23, 24].

Besides, several criteria may be considered in a real application. The multi-objective evolutionary algorithms, that are used to find a set of solutions with trade-offs among different criteria, are thus very suitable for solving such a task [13, 14]. In the fuzzy-control field, many approaches have been proposed for tuning parameters and learning membership functions [2, 3, 7, 17]. As to fuzzy mining, Kaya *et al.* proposed an approach that integrated the multi-objective genetic algorithm into clustering for fuzzy mining [5]. The number of large itemsets and the spent execution time were considered as two objective functions to derive appropriate membership functions for mining fuzzy association rules. Besides, Kaya also proposed an approach based on multi-objective genetic algorithms for mining optimized fuzzy association rules [26]. He defined three objectives, namely strongness, interestingness and comprehensibility, to derive appropriate membership functions for mining optimized fuzzy association rules. We also proposed a fuzzy mining approach based on the Multi-Objective Genetic Algorithm (MOGA) to find the Pareto front of the desired membership functions [9].

In this paper, we adopt a more sophisticated multi-objective approach, the SPEA2 [35], to find the appropriate sets of membership functions for fuzzy data mining. SPEA2 is usually regarded as having a better effect than MOGA. It adopts a fine-grained fitness assignment strategy, a density estimation technique, and an enhanced archive truncation method to derive better Pareto solutions [35]. Two objective functions are used here to find the Pareto front of membership functions. The first one is the suitability of membership functions and the second one is the total number of large 1-itemsets derived. The suitability measure is used to reduce the occurrence of bad types of membership functions. Besides, using the number of large 1-itemsets, instead of the number of rules, can achieve a trade-off between execution time and rule interestingness. Experimental results first show the effectiveness of the proposed algorithm in items of Parent fronts and number of derived rules. Then, comparison results between the proposed and the previous approaches are made to show the effectiveness of the proposed approach in finding the Pareto-front membership functions. There are two main contributions of this paper. The first one is that the proposed approach provides an enhanced approach for deriving more appropriate Pareto front. The second one is that it can provide different options in terms of number of rules to users for further analysis.

The remaining parts of this paper are organized as follows. The background knowledge of the multi-objective optimization problem is stated in Section 2. The details of the genetic process for membership functions and the two objective functions are explained in Section 3. The proposed algorithm for mining both membership functions and association rules are described in Section 4. An example to illustrate the proposed algorithm is given in Section 5. Experiments to demonstrate the performance of the proposed algorithm are stated in Section 6. Conclusions and future works are given in Section 7.

2. GA-Based Multi-Objective Optimization Problems

A multi-objective optimization problem can be defined as follows:

$$\text{Min/Max } y = g(x) = (g_1(x), g_2(x), \ldots, g_m(x)),$$

$$\text{subject to } x = (x_1, x_2, ..., x_n) \in X \text{ and}$$

$$y = (y_1, y_2, ..., y_m) \in Y,$$

where x is the decision vector, y is the objective vector, X represents the decision space, and Y represents the objective space. In the past, several GA-based approaches were proposed to get the solutions. For

example, Schaffer proposed the Vector Evaluated Genetic Algorithm (VEGA) to solve the multi-objective optimization problem [30]. The difference between VEGA and the simple genetic algorithm lay in the selection strategy. Then, Fonseca *et al.* proposed a modified approach called Multi-Objective Genetic Algorithm (MOGA) by using the extended rank-based fitness assignment [16]. They also defined three relationships among chromosomes, namely inferiority, superiority and non-inferiority, which are shown in Fig. 1 [16].

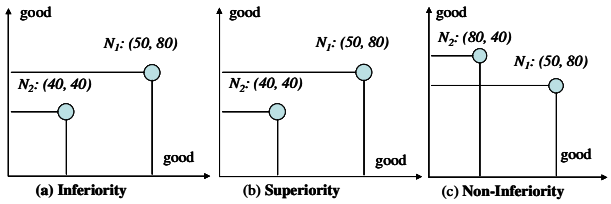

Fig. 1. Three relationships among chromosomes in MOGA

As shown in Fig. 1, the first relationship is inferiority. For example, since both the objective values of node N_1 are larger than those of node N_2, the latter is then said to be inferiority to the former (Fig. 1(a)). On the other hand, we can also say that N_1 is superiority to N_2 (Fig. 1(b)). The third relationship is non-inferiority. Take Fig. 1(c) as an example in which one objective value (x-axis) of node N_1 is larger than that of node N_3 and the other one of N_1 is smaller. In this case, N_1 is said to be non-inferiority to N_3. The MOGA strategy was thus proposed to find the set of non-inferiority solutions, also called Pareto optimal solutions or Pareto front. Fig. 2 explains the three relationships and the Pareto optimal solutions.

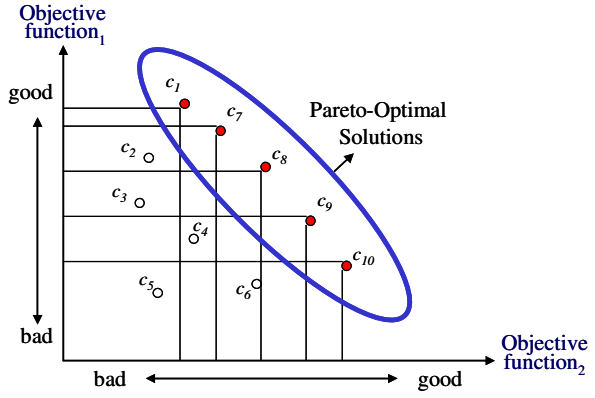

Fig. 2. An example for the Pareto optimal solutions

The goal of MOGA is to find the non-dominated points, also called Pareto optimal solutions. In this example, the chromosomes C_1, C_7, C_8, C_9 and C_{10} are non-dominated points. Besides, some variants of MOGA were also proposed. Two well-known approaches are NSGA-II [15] and SPEA2 [35]. Their main purpose was to get better Pareto fronts. NSGA-II used a fast non-dominated sorting procedure, an elitist strategy, and an approach without parameters to achieve this [14]. SPEA2 adopted a fine-grained fitness assignment strategy, a density estimation technique, and an enhanced archive truncation method to derive better Pareto solutions [35].

3. The SPEA2-based Multi-objective Genetic-Fuzzy Mining Approach

In this paper, we propose a SPEA2-based approach to derive the set of non-dominated solutions for fuzzy mining problems. The details of the proposed approach are described below.

3.1. *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 [1, 12, 29, 32]. In this paper, the set of membership functions for an item is encoded as shown in Fig. 3.

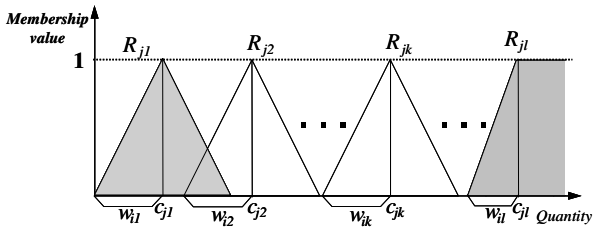

Fig. 3. The set of membership functions for an item I_j

In Fig. 3, each membership function is assumed to be isosceles-triangle and represented by a pair (c, w), with c indicating the center abscissa and w representing half the span. R_{jk} denotes the membership function of the k-th linguistic term of *item* I_j. All pairs of (c, w)'s for a certain item are concatenated to represent its membership functions. Since both c and w are numeric values, a chromosome is thus encoded as a fixed-length real-number string rather than a bit string.

Note that other types of membership functions (e.g. non-isosceles triangles and trapezes) can also be

adopted in our method. For coding non-isosceles triangles and trapezes, three and four points are needed instead of two for isosceles triangles. Besides, the numbers of membership functions for given items can be different.

3.2. *Initial Population*

A genetic algorithm requires a population of feasible solutions to be initialized and updated during the evolution process. As mentioned above, each individual within the population is a set of isosceles-triangular membership functions. Each membership function corresponds to a linguistic term in a certain item. The initial set of chromosomes is randomly generated with some constraints for forming feasible membership functions.

3.3. *The Two Objective Functions*

Kaya et al. proposed an approach to derive membership functions for mining problems [23]. It could get a maximum profit (maximum number of large itemsets) within an interval of user specified minimum support values. The derived membership functions were then used to mine fuzzy association rules. In our previous work, we have also proposed a genetic-fuzzy approach to learn an appropriate set of membership functions for mining problems [19]. In that paper, the fitness values were evaluated by the numbers of large 1-itemsets over the suitability of membership functions. The two factors (numbers of large 1-itemsets and suitability of membership functions) usually show a trade-off relationship. In this paper, we thus consider the mining of membership functions and fuzzy association rules as a multi-objective optimization problem, in which the above two factors are used as two objectives functions. A SPEA2-based mining algorithm is thus proposed to find the Pareto optimal solutions. The first objective function (Obj_1) for a chromosome C_q is defined as follows:

$$Obj_1(C_q) = suitability(C_q),$$

where $suitability(C_q)$ represents the shape suitability of the membership functions with C_q. $Suitability(C_q)$ is defined as:

$$\sum_{j=1}^{m}[\,overlap\_factor(C_{qj}) + coverage\_factor(C_{qj})],$$

where m is the number of items. $Overlap\_factor(C_{qj})$ represents the overlap factor of the membership functions for an item I_j in the chromosome C_q and is defined as:

$$overlap\_factor(C_{qj}) = \sum_{k \neq i}[\max((\frac{overlap(R_{jk}, R_{ji})}{\min(w_{jk}, w_{ji})}), 1) - 1],$$

where $overlap(R_{jk}, R_{ji})$ is the overlap length of R_{jk} and R_{ji}. $Coverage\_factor(C_{qj})$ represents the coverage ratio of a set of membership functions for an item I_j in the chromosome C_q and is defined as:

$$\text{coverage}\_factor(C_{qj}) = \frac{1}{\frac{range(R_{j1}, ..., R_{jl})}{\max(I_j)}},$$

where $range(R_{j1}, R_{j2}, \ldots, R_{jl})$ 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 suitability factor is used to reduce the occurrence of the two bad kinds of membership functions shown in Fig. 4, where the first one is too redundant and the second one is too separate.

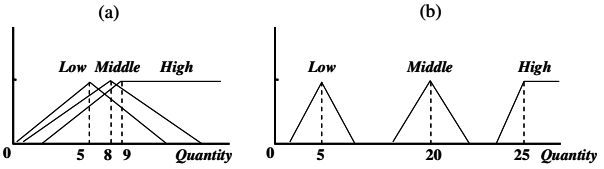

Fig. 4. Two bad membership functions

The second objective function is the total number of large 1-itemsets in a given set of minimum support values $\{ms_1, ms_2, \ldots, ms_h\}$. It is formally defined as follows:

$$Obj_2(C_q) = totalNumL1(C_q) = \sum_{g=1}^{h} | L_{1q}^{ms_g} |,$$

where $| L_{1q}^{ms_g} |$ is the number of large 1-itemsets obtained when the minimum support value is ms_g. Using the number of large 1-itemsets can achieve a trade-off between execution time and rule interestingness. Usually, a larger number of 1-itemsets will result in a larger number of all itemsets with a higher probability, which will thus usually imply more interesting association rules. In this paper, the proposed approach uses the above two objective functions to find appropriate Pareto solutions for the genetic-fuzzy mining problems.

3.4. *Fitness Assignment*

The fitness assignment is similar to that used in SPEA2 [35]. The fitness of a chromosome C_q is calculated by using the formula as follows:

$$f(C_q) = R(C_q) + D(C_q),$$

where $R(C_q)$ is the raw fitness of a chromosome, and $D(C_q)$ is the density information of a chromosome. The raw fitness is used to exhibit the strength of each chromosome, and is defined as follows:

$$R(C_q) = \sum_{j \in P + \overline{P},\, j \succ q} S(j),$$

where the strength value $S(j)$ is the number of solutions it dominates of chromosome C_j, and is calculated as follows:

$$S(j) = |\{j \mid j \in P + \overline{P} \wedge i \succ j\}|,$$

where $|\bullet|$ means the cardinality of a set, + represents multiset union and the symbol $\succ$ means the Pareto dominance relation. In other words, the raw fitness of a chromosome is determined by the strength of its dominators in both the population P and the archive $\overline{P}$. Thus, the lower the raw fitness is, the better the chromosome is. The density information of a chromosome C_q is defined as follows:

$$D(C_q) = \frac{1}{\sigma_q^k + 2},$$

where σ_q^k is the distance of C_q to its k-th nearest chromosome in both the population P and the archive $\overline{P}$, and k is calculated by $\sqrt{N + \overline{N}}$. The density information is used to distinguish the difference of chromosomes which have the same raw fitness.

3.5. *Genetic Operators*

Genetic operators are very important to the success of specific GA applications. Two genetic operators, the *max-min-arithmetical (MMA) crossover* proposed in [22] and the *one-point mutation*, are used in the proposed approach. Assume there are two parent chromosomes:

$C_u^t = (c_1, \ldots, c_h, \ldots, c_z)$, and $C_w^t = (c_1', \ldots, c_h', \ldots, c_z')$.

The *max-min-arithmetical (MMA)* crossover operator will generate the following four candidate chromosomes from the two parents:

1. $C_1^{t+1} = (c_{11}^{t+1}, \ldots, c_{1h}^{t+1}, \ldots, c_{1Z}^{t+1})$, where $c_{1h}^{t+1} = dc_h + (1-d)c_h'$,
2. $C_2^{t+1} = (c_{21}^{t+1}, \ldots, c_{2h}^{t+1}, \ldots, c_{2Z}^{t+1})$, where $c_{2h}^{t+1} = dc_h' + (1-d)c_h$,
3. $C_3^{t+1} = (c_{31}^{t+1}, \ldots, c_{3h}^{t+1}, \ldots, c_{3Z}^{t+1})$, where $c_{3h}^{t+1} = \min\{c_h, c_h'\}$,
4. $C_4^{t+1} = (c_{41}^{t+1}, \ldots, c_{4h}^{t+1}, \ldots, c_{4Z}^{t+1})$, where $c_{4h}^{t+1} = \max\{c_h, c_h'\}$.

The parameter d is either a constant or a variable whose value depends on the age of the population. The best two chromosomes among the 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 ε (between $-w_{jk}$ to $+ w_{jk}$) to the center or to the spread of an existing linguistic term, say R_{jk}. Assume that c and w represent the center and the spread of R_{jk}. The center or the spread of the newly derived membership function will be changed to $c + \varepsilon$ or $w + \varepsilon$ by the mutation operation. Mutation at the center 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 center values. Besides, the selection strategy used in the proposed approach can be the elitist or the roulette-wheel strategy.

4. The Proposed Mining Algorithm

According to the above description, the proposed SPEA2-based genetic-fuzzy mining algorithm for deriving both membership functions and fuzzy association rules is described below.

Notations used in this paper:

n: the total number of transaction data;
m: the total number of items;
I_j: the j-th item, $1 \le j \le m$;
$|I_j|$*:* the number of fuzzy regions for I_j;
$D^{(i)}$*:* the i-th transaction datum, $1 \le i \le n$;
R_{jk}*:* the k-th fuzzy region of I_j, $1 \le k \le |I_j|$;
$v_j^{(i)}$*:* the quantitative value of I_j for $D^{(i)}$;
$f_j^{(i)}$*:* the fuzzy set converted from $v_j^{(i)}$;
$f_{jk}^{(i)}$*: the membership value of* $v_j^{(i)}$ *in Region* R_{jk};
$count_{jk}$*:* the summation of $f_{jk}^{(i)}$ for $i = 1$ *to* n;
ms_g: the g-th minimum support;
G: the number of generations;
N: the population size;
$\overline{N}$: the archive size;
$\overline{P}$: the non-dominated (archive) set;
C_q: the q-th chromosome in the population;
$|L_{1q}^{ms_g}|$: the number of large 1-itemsets obtained by chromosome C_q with the minimum support ms_g;
$R(C_q)$: the raw fitness of chromosome C_q;
$D(C_q)$: the density information of chromosome C_q;

The SPEA-based Genetic-Fuzzy Mining Algorithm:

INPUT: A body of n quantitative transactions, a set of m items, each with a number of linguistic terms, a

set of minimum support values $\{ms_1, ms_2, ..., ms_h\}$, a population size N, an archive size $\overline{N}$, a crossover rate P_c, a mutation rate P_m, a number of generation G and a confidence threshold λ.

OUTPUT: A set of non-dominated solutions (sets of membership functions) with their fuzzy association rules.

STEP 1: Randomly generate a population P of N individuals, with each one being a set of membership functions for all the m items, encode each set of membership functions into a string representation according to the schema stated in Section 3, and initialize the non-dominated (archive) set $\overline{P}$ as empty.

STEP 2: For each chromosome C_q, calculate its two objective values, the suitability and the total number of large 1-itemsets according to the given set of minimum support values $\{ms_1, ms_2, ..., ms_h\}$, by the following substeps.

SUBSTEP 2.1: 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 $f_j^{(i)}$ by using the corresponding membership functions encoded in the chromosome and represented as:

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

where R_{jk} is the k-th fuzzy region (term) of item I_j, $f_{jk}^{(i)}$ 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.

SUBSTEP 2.2: For each item region R_{jk}, calculate its scalar cardinality on the transactions as follows:

$$count_{jk} = \sum_{i=1}^{n} f_{jk}^{(i)}.$$

SUBSTEP 2.3: For each R_{jk}, $1 \le j \le m$, $1 \le k \le |I_j|$, check whether $count_{jk}$ is larger than or equal to the set of minimum support values $\{ms_1, ms_2, ..., ms_h\}$. If R_{jk} satisfies the above condition, set $|L_{1q}^{ms_g}| = |L_{1q}^{ms_g}| + 1$, where $|L_{1q}^{ms_g}|$ is the number of large 1-itemsets obtained by using the set of membership functions in chromosome C_q and the minimum support value ms_g. The second objective value of C_q is shown as follows:

$$\text{totalNumL1}(C_q) = \sum_{g=1}^{h} |L_{1q}^{ms_g}|,$$

SUBSTEP 2.4: Calculate the suitability value $suitability(C_q)$ by using the formula defined in Section 3; set it as the first objective value of C_q.

STEP 3: Calculate the raw fitness $R(C_q)$ of each chromosome C_q by using following formula:

$$R(C_q) = \sum_{j \in P + \overline{P}, j \succ q} S(j),$$

where the strength value $S(j)$ is the number of solutions it dominates of chromosome C_j, and calculate as follows:

$$S(j) = |\{j \mid j \in P + \overline{P} \wedge i \succ j\}|,$$

where $|\bullet|$ means the cardinality of a set, $+$ represents multiset union and the symbol $\succ$ means the Pareto dominance relation.

STEP 4: Calculate the density information $D(C_q)$ of each chromosome C_q by using following formula:

$$D(C_q) = \frac{1}{\sigma_q^k + 2},$$

where σ_q^k is the distance of C_q to its k-th nearest chromosome in both population P and archive $\overline{P}$, and k is calculated by $\sqrt{N + \overline{N}}$.

STEP 5: Set fitness value of each chromosome as follows:

$$f(C_q) = R(C_q) + D(C_q).$$

STEP 6: Copy nondominated chromosomes to archive $\overline{P}$. In other words, chromosomes with their fitness values smaller than one will be copied to the archive.

STEP 7: Execute environmental selection according to the number of chromosomes in the archive. There are three cases. The first case is if the number of chromosomes in the archive $|\overline{P}|$ equals to $\overline{N}$, then go to next step. In second case, if the number of chromosome in the archive is smaller than $\overline{N}$, then the best $\overline{N}$ - $|\overline{P}|$ dominated chromosomes with fitness values larger than one are selected from previous population and archive. Otherwise, if number of chromosomes in the archive exceeds $\overline{N}$, truncation operator is used to reduce the size of archive. At each iteration, the chromosome C_q with the smallest σ_q^k, which is the distance of C_q to its k-th nearest chromosome in archive, is removed until $|\overline{P}| = \overline{N}$. In case of many chromosomes have the same minimum distance, and then the second

smallest distance is chosen for removal, and so on.

STEP 8: Use the selection operation to choose appropriate individuals from the archive $\overline{P}$ to form the next generation. Here, the binary tournament selection is performed.

STEP 9: Execute the crossover operation on the population.

STEP 10: Execute the mutation operation on the population.

STEP 11: If the termination criterion is not satisfied, go to Step 2; otherwise, do the next step.

STEP 12: Mine fuzzy association rules from the given database and based on the derived chromosomes in archive $\overline{P}$, where each chromosome represents a set of membership functions. The fuzzy mining algorithm proposed in [21] is then adopted to achieve this purpose for each set of membership functions.

STEP 13: Output the archive $\overline{P}$ and their corresponding fuzzy association rules.

5. An Example

In this section, a simple example is given to illustrate the proposed multi-objective genetic-fuzzy mining algorithm. Assume there are four items in a transaction database: milk, bread, cookies and beverage. The dataset includes the six transactions shown in Table 1.

Table 1. The six transactions in the example

TID	Items
T1	(milk, 5), (bread, 10), (cookies, 7), (beverage, 7).
T2	(milk, 7), (bread, 6), (cookies, 12).
T3	(bread, 8), (cookies, 12); (beverage, 3).
T4	(milk, 2); (bread, 5); (cookies, 5).
T5	(bread, 9).
T6	(milk, 10), (beverage, 6).

Assume each item has three fuzzy regions: *Low*, *Middle* and *High* for simplicity. Thus, three fuzzy membership functions must be derived for each item. Note that the numbers of fuzzy regions for the items are not necessarily the same for the proposed approach. For the data shown in Table 1, the proposed mining algorithm proceeds as follows.

STEP 1: *P* individuals are randomly generated to form the initial population. The non-dominated set *NDS* is also initialized as empty. In this example, *P* is set at 10. Each individual is thus a set of membership functions for all the four items including milk, bread, cookies, and beverage. Assume the following ten individuals are generated:

C_1: 5, 2, 6, 4, 10, 4, 1, 1, 3, 1, 4, 2, 2, 1, 4, 1, 7, 2, 6, 5, 7, 3, 9, 3,
C_2: 5, 1, 7, 3, 9, 3, 1, 1, 9, 1, 10, 1, 5, 2, 6, 5, 7, 5, 1, 1, 3, 1, 4, 1,
C_3: 5, 3, 7, 2, 8, 5, 4, 3, 6, 3, 8, 3, 2, 1, 3, 2, 8, 5, 1, 1, 6, 3, 10, 4,
C_4: 4, 1, 7, 5, 9, 1, 3, 1, 4, 3, 10, 3, 1, 1, 3, 2, 10, 1, 1, 1, 5, 1, 7, 4,
C_5: 3, 1, 6, 2, 9, 4, 7, 3, 8, 2, 10, 1, 4, 1, 5, 2, 7, 3, 3, 2, 5, 2, 7, 3,
C_6: 4, 3, 6, 4, 8, 3, 2, 1, 4, 1, 5, 1, 5, 1, 8, 3, 9, 2, 2, 1, 8, 1, 10, 4,
C_7: 4, 2, 5, 1, 10, 4, 3, 1, 4, 3, 10, 3, 1, 1, 3, 2, 6, 1, 6, 1, 7, 3, 10, 1,
C_8: 4, 1, 6, 1, 9, 4, 3, 1, 4, 3, 10, 2, 5, 1, 7, 4, 9, 4, 1, 1, 2, 1, 4, 1,
C_9: 2, 1, 8, 3, 9, 5, 4, 1, 6, 5, 9, 5, 2, 1, 3, 2, 5, 4, 2, 1, 7, 3, 10, 1,
C_{10}: 3, 1, 5, 1, 9, 4, 5, 1, 6, 5, 7, 1, 5, 1, 8, 1, 9, 2, 1, 1, 2, 1, 7, 3.

STEP 2: The suitability value and the total number of large 1-itemsets in the given set of minimum supports values of each chromosome are calculated by the following substeps:

SUBSTEP 2.1: The quantitative value of each transaction datum is transformed into a fuzzy set according the membership functions in each chromosome. Take the first item in transaction *T1* using the membership functions in chromosome C_1 as an example. The membership functions for milk in C_1 are represented as (5, 2, 6, 4, 10, 4). The amount "5" of item *milk* is then converted into the fuzzy set (1.0/Low + 0.75/Middle). The results for all the items are shown in Table 2, where the notation *item.term* is called a fuzzy region.

Table 2. The transformed fuzzy sets

TID	Fuzzy Set
T1	$(\frac{1.0}{milk.Low}+\frac{0.75}{milk.Middle})(\frac{1.0}{bread.High})(\frac{1.0}{cookies.High})$ $(\frac{0.8}{beverage.Low}+\frac{1.0}{beverage.Middle}+\frac{0.33}{beverage.High})$
T2	$(\frac{0.75}{milk.Middle}+\frac{0.25}{milk.High})(\frac{1.0}{bread.High})(\frac{1.0}{cookies.High})$
T3	$(\frac{1.0}{bread.High})(\frac{1.0}{cookies.High})(\frac{0.4}{beverage.Low})$
T4	$(\frac{0.0}{milk.Low})(\frac{1.0}{bread.High})(\frac{0.0}{cokkies.Middle})$
T5	$(\frac{1.0}{bread.High})$
T6	$(\frac{1.0}{milk.High})(\frac{1}{beverage.Low}+\frac{0.66}{beverage.Middle})$

SUBSTEP 2.2: The scalar cardinality of each fuzzy region in the transactions is calculated as the *count* value. Take the fuzzy region *milk.Middle* as an example. Its scalar cardinality = (0.75 + 0.75 + 0.0 + 0.0 + 0.0 + 0.0) = 1.5. The counts for all the fuzzy regions are shown in Table 3.

Table 3. The counts of all the fuzzy regions

Item	Count	Item	Count
milk.Low	1.00	*cookies.Low*	0.0
milk.Middle	1.50	*cookies.Middle*	0.0
milk.High	1.25	*cookies.High*	3.0
bread.Low	0.0	*beverage.Low*	2.2
bread.Middle	0.0	*beverage.Middle*	1.66
bread.High	5.0	*beverage.High*	0.33

SUBSTEPs 2.3: The suitability value of the chromosome C_1 can be calculated as 8.38 according to the formulas in Section 3.

SUBSTEP 2.4: The count of any fuzzy region is checked against the set of minimum support values. Assume the set of minimum support values is {0.08, 0.09, 0.1, …, 0.17}. Take the minimum support value set at 0.08 as an example. Since the count values of *milk.Low*, *milk.Middle, milk.High, bread.High, cookies.High, beverage.Low and beverage.Middle* are larger than 0.48 (= 0.08*6), the number of large 1-itemsets is thus 7. The number of large 1-itemsets for the other minimum support values can be similarly found. The total number of large 1-itemsets *totalL1*(C_1) is thus 69 (= 7+7+7+7+7+7+7+7+7+6). The two objective values of the chromosome C_1 are thus 8.38 and 69. The results of all the ten chromosomes are shown in Table 4.

Table 4. The *suitability value* and the *totalL1* of each chromosome

C_q	(suitability, totalL1)	C_q	(suitability, totalL1)
C_1	**(8.38, 69)**	C_6	**(8.51, 52)**
C_2	**(9.72, 85)**	C_7	**(7.79, 67)**
C_3	**(8.30, 78)**	C_8	**(8.36, 58)**
C_4	**(8.88, 48)**	C_9	**(9.98, 68)**
C_5	**(8.62, 87)**	C_{10}	**(8.09, 66)**

STEP 3: The raw fitness $R(C_q)$ of each chromosome C_q is calculated. Take chromosome C_1 as an example, we can know that the chromosome C_1 is dominated by chromosome C_3. Thus, before calculating the raw fitness, the strength value of chromosome C_3 is needed to be derived. In this example, the strength of chromosome C_3 is 5. The raw fitness of chromosome C_1 is thus 5. In the same way, the results of other chromosomes are shown in Table 5.

Table 5. The raw fitness of each chromosome

C_i	$R(C_q)$	C_i	$R(C_q)$
C_1	5	C_6	17
C_2	3	C_7	0
C_3	0	C_8	12
C_4	21	C_9	12
C_5	0	C_{10}	4

STEP 4: The density information $D(C_q)$ of each chromosome is then derived. Assume the archive size is set at 5, the parameter k is thus 3 ($=\lfloor(10 + 5)^{1/2}\rfloor$). Take chromosome C_1 as an example, the distance of third-nearest chromosome is calculated as 0.083 (= 1/12.0). In the same way, the density information of other chromosomes is shown in Table 6.

Table 6. The density information of each chromosome

C_i	$D(C_q)$	C_i	$D(C_q)$
C_1	0.833	C_6	0.055
C_2	0.047	C_7	0.058
C_3	0.071	C_8	0.045
C_4	0.045	C_9	0.055
C_5	0.045	C_{10}	0.049

STEP 5: The fitness of each chromosome is then set at the summation of its raw fitness and density information. The results are shown in Table 7.

Table 7. The fitness value of each chromosome

C_i	$f(C_q)$	C_i	$f(C_q)$
C_1	5.833	C_6	17.055
C_2	3.047	C_7	0.058
C_3	0.071	C_8	12.045
C_4	21.045	C_9	12.055
C_5	0.045	C_{10}	4.049

STEP 6: The chromosomes with their fitness values smaller than one are copied to the archive. In this example, the chromosomes C_3, C_5 and C_7 are copied to archive $\overline{P}$ as non-dominated chromosomes.

STEP 7: Since the number of chromosome in the archive is smaller than five, then first two dominated

chromosomes with fitness values larger than one are selected from population and archive. Here, chromosomes C_2 and C_{10} are selected.

STEP 8 to 11: The selection operation is then used to generate next population from archive. Here, the binary tournament selection is performed. Then, the crossover and mutation operations are used to produce new offspring. If the termination criterion is not satisfied, go to Step 2; otherwise, do the next step.

STEP 12 to 13: The derived chromosomes in archive, where each chromosome represents a set of membership functions, are used to mine fuzzy association rules from the given database and based on the fuzzy mining algorithm proposed in [21]. At last, the archive $\overline{P}$ and their corresponding fuzzy association rules are outputted.

6. Experimental Results

In this section, experiments made to show the performance of the proposed approach are described. They were implemented in Java on a personal computer with Intel Pentium IV 3.20 GHz and 512 MB RAM. 64 items and 10000 transactions were used in the experiments. The initial population size P is set at 50, the archive size is set at 25, the crossover rate p_c is set at 0.8, and the mutation rate p_m is set at 0.001. The parameter d of the crossover operator is set at 0.35 according to Herrera *et al.*'s paper [22] and the set of minimum support values is {3%, 4%, …, 13%}. In the following subsections, we first give a description of the experimental dataset. We then analyze the evolution of the Pareto fronts obtained by the proposed approach.

6.1. *Description of the Experimental Datasets*

Two simulated datasets with 64 items and with 10000 transactions were used in the experiments. One dataset followed exponential distribution and another one followed uniform distribution. The factors for the two 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 [1, 19] for both the two datasets. The purchased items in each transaction were then selected from the 64 items in a uniform distribution of the range [1, 64] for the uniform dataset and in an exponential distribution with the rate parameter set at 16 for the exponential dataset. Their quantities were then assigned from a uniform distribution of the range [1, 11] for the uniform dataset and from an exponential distribution with the rate parameter set at 5 for the exponential dataset. The simulation process was repeated until the dataset size was reached. An item could not be generated twice in a transaction.

6.2. *The Evolution of Pareto Fronts by the Proposed Approach*

The experiments were first made for demonstrating the evolution of the Pareto fronts by the proposed approach. The evolution of the Pareto fronts of chromosomes in the archive along with different generations by the proposed approach for two simulation datasets are shown in Fig. 5 and 6, respectively.

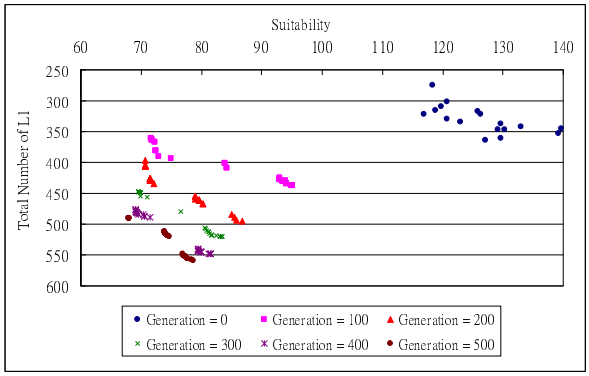

Fig. 5. The Pareto fronts derived by the proposed approach for the exponential dataset with different generations

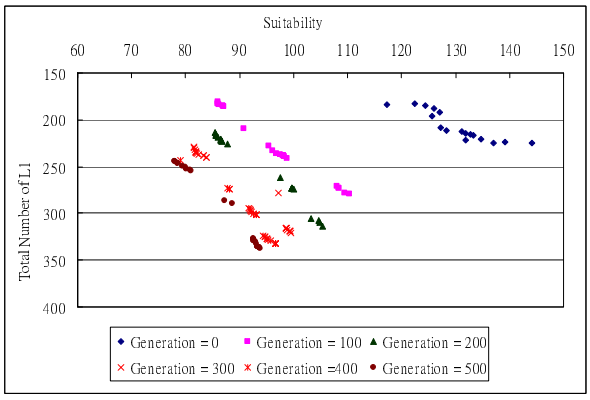

Fig. 6. The Pareto fronts derived by the proposed approach for the uniform dataset with different generations

From Fig. 5 and 6, we can observe that the solutions derived from the two datasets were distributed on the Pareto fronts and the final solutions after 500 generations were better than those in different generations. Additionally, we can also found that the

derived solutions on a Pareto front are trade-offs between the two objectives. It thus depends on the user preference to decide which solutions on a Pareto front are desired. In order to show the trade-off between the two objectives, experiments were then made to compare the number of rules derived by using the membership functions with the minimum suitability values (in short S) and with the maximum total number of large itemsets (in short L). The comparison results for the two datasets are shown in Fig. 7 and 8.

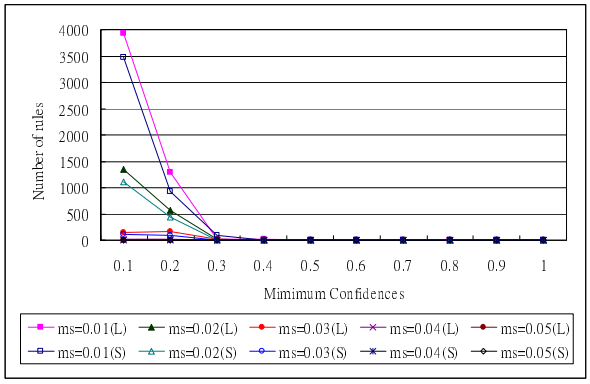

Fig. 7. The number of rules derived by the proposed approach for the exponential dataset

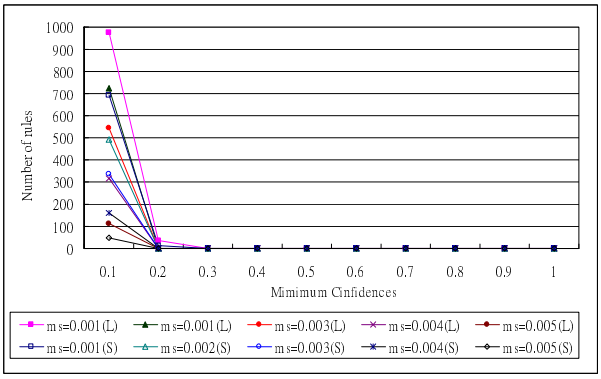

Fig. 8. The number of rules derived by the proposed approach for the uniform dataset

From Fig. 7 and 8, the following two common phenomena could be observed. The first one was that the number of rules always decreased along with the increase of the minimum confidence, no matter rules were derived by the membership functions with the minimum suitability values or with the maximum total number of large itemsets. The second one was that the number of rules derived by the membership functions with the minimum suitability value was smaller than that by the membership functions with the maximum total number of large itemsets. This phenomenon was reasonable from the properties of the two objective functions.

6.3. *The Comparison Results with Previous Approaches*

The experiment was then made for comparing the final Pareto front of chromosomes in the archive of the proposed approach with the previous approach [9], and is shown in Fig. 9.

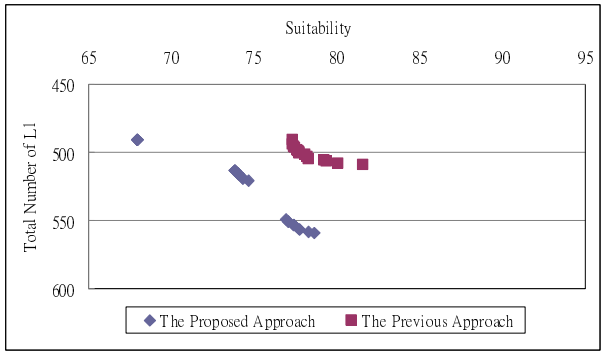

Fig. 9. Comparison results of the final Pareto fronts for the exponential dataset

From Fig. 9, it is easily to know that the Pareto front derived by using the proposed approach is better than the previous one.

At last, experiments were made for showing the comparison results in terms of number of rules of the proposed approach and the mono-objective approach [18]. The results are shown in Fig. 10.

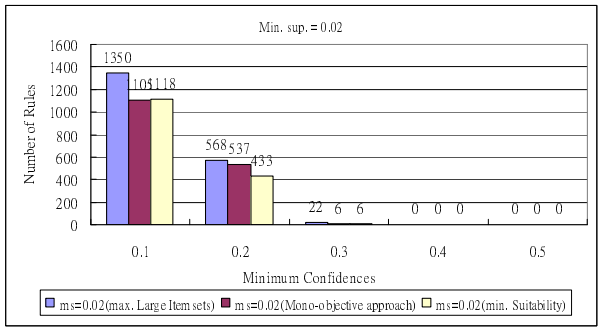

Fig. 10. The number of rules derived by the proposed approach for the uniform dataset

It could be observed from Fig. 10 that when using membership functions with the maximum total number of large itemsets to mine rules, the number of rules derived by the proposed approach was more than that by

the mono-objective approach. On the contrary, when using membership functions with the minimum suitability, the number of rules derived by the mono-objective approach was larger than that by the proposed approach. Thus, if users have different considerations for making decisions, the proposed approach could provide appropriate solutions.

From the experimental results, we thus can conclude that the proposed approach is not only effective in finding an appropriate set of solutions, but also can provide different options to users for further analysis.

7. Conclusion and Future Works

The SPEA2 adopted a fine-grained fitness assignment strategy, a density estimation technique, and an enhanced archive truncation method to derive better Pareto solutions [35]. In this paper, we have utilized it to propose a more sophisticated multi-objective approach to find the appropriate sets of membership functions for fuzzy data mining. Two objective functions are used to find the Pareto front. They are minimizing the suitability of membership functions and maximizing the total number of large 1-itemsets, respectively.

Experiments on two simulation datasets were also made to show the effectiveness of the proposed approach. The results show that the proposed approach is effective in finding an appropriate set of solutions. Further, the experiments also show that the proposed approach can derive better Pareto front than the previous one [9]. In the future, we will continuously enhance the multi-objective genetic-fuzzy approach for more complex problems.

Acknowledgements

This research was supported by the National Science Council of the Republic of China under contract NSC 99-2221-E-390-028.

References

1. J. Alcala-Fdez, R. Alcala, M. J. Gacto, F. Herrera, "Learning the membership function contexts for mining fuzzy association rules by using genetic algorithms," *Fuzzy Sets and Systems*, Vol. 160, No. 7, pp. 905-921, 2009.
2. R. Alcalá, Y. Nojima, F. Herrera and H. Ishibuchi, "Multiobjective genetic fuzzy rule selection of single granularity-based fuzzy classification rules and its interaction with the lateral tuning of membership functions," Soft Computing, DOI 10.1007/s00500-010-0671-2, 2010.
3. M. Antonelli, P. Ducange, B. Lazzerini and F. Marcelloni, "Learning knowledge bases of multi-objective evolutionary fuzzy systems by simultaneously optimizing accuracy, complexity and partition integrity," *Soft Computing*, DOI 10.1007/s00500-010-0671-2, 2010.
4. R. Agrawal, T. Imielinksi and A. Swami, "Database mining: a performance perspective," *IEEE Transactions on Knowledge and Data Engineering,* Vol. 5, No. 6, 1993, pp. 914-925.
5. R. Alhajj and M. Kaya, "Multi-objective genetic algorithms based automated clustering for fuzzy association rules mining," *Journal of Intelligent Information Systems*, Vol. 31, No. 3, pp. 243-264, 2008.
6. R. Agrawal and R. Srikant, "Fast algorithm for mining association rules," *The International Conference on Very Large Databases*, pp. 487-499, 1994.
7. A. Botta, B. Lazzerini, F. Marcelloni and D. Stefanescu, "Context adaptation of fuzzy systems through a multi-objective evolutionary approach based on a novel interpretability index," *Soft Computing*, Vol. 13, No 3, p. 437-449, 2009.
8. C. C. Chan and W. H. Au, "Mining fuzzy association rules," *The Conference on Information and Knowledge Management*, Las Vegas, pp. 209-215, 1997.
9. C. H. Chen, T. P. Hong, Vincent S. Tseng and L. C. Chen, "A multi-objective genetic-fuzzy mining algorithm," *The 2008 IEEE International Conference on Granular Computing,* pp. 115-120, 2008.
10. C. H. Chen, T. P. Hong, Vincent S. Tseng and C. S. Lee, "A genetic-fuzzy mining approach for items with multiple minimum supports," *Soft Computing,* Vol. 13, No. 5, pp. 521-533, 2009.
11. C. H. Chen, Vincent S. Tseng and T. P. Hong, "Cluster-based evaluation in fuzzy-genetic data mining," *IEEE Transactions on Fuzzy Systems*, Vol. 16, No. 1, pp. 249-262, 2008.
12. 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 Transactions on Fuzzy Systems*, Vol. 9, No. 4, pp. 667–674, 2001.
13. C. A. Coello, D. A. Van Veldhuizen and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-objective Problems*, Kluwer Academic Publishers, 2002.
14. K. Deb, Multi-objective Optimization Using Evolutionary Algorithms, John Wiley & Sons, 2001.
15. K. Deb, S. Agrawal, A. Pratab and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, pp. 681-695.
16. C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization," *The International Confidence on Genetic Algorithms*, pp. 416-423, 1993.
17. M. J. Gacto, R. Alcalá and F. Herrera, "Adaptation and application of multi-objective evolutionary algorithms for

rule reduction and parameter tuning of fuzzy rule-based systems," *Soft Computing*, Vol. 13, No 3, p. 419-436, 2009.
18. T. P. Hong, C. H. Chen, Y. L. Wu and Y. C. Lee, " Genetic-Fuzzy Data Mining with Divide-and-Conquer Strategy", *IEEE Transactions on Evolutionary Computation*, Vol. 12, No. 2, pp. 252-265, 2008.
19. 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 Computing*, Vol. 10, No. 11, pp. 1091-1101. 2006.
20. T. P. Hong, C. S. Kuo and S. C. Chi, "Mining association rules from quantitative data," *Intelligent Data Analysis*, Vol. 3, No. 5, pp. 363-376, 1999.
21. 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," *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems,* Vol. 9, No. 5, pp. 587-604, 2001.
22. F. Herrera, M. Lozano and J. L. Verdegay, "Fuzzy connectives based crossover operators to model genetic algorithms population diversity," *Fuzzy Sets and Systems,* Vol. 92, No. 1, pp. 21–30, 1997.
23. M. Kaya, R. Alhajj, "Genetic algorithm based framework for mining fuzzy association rules," *Fuzzy Sets and Systems*, Vol. 152, No. 3, pp. 587-601, 2005.
24. M. Kaya, R. Alhajj, "Utilizing genetic algorithms to optimize membership functions for fuzzy weighted association rules mining," *Applied Intelligence*, Vol. 24, No. 1, pp. 7-15, 2006.
25. M. Kaya, R. Alhajj, "Effective mining of fuzzy multi-cross-level weighted association rules," *Lecture Notes in Computer Science*, Vol. 4203, pp. 399-408, 2006.
26. M. Kaya, "Multi-objective genetic algorithm based approaches for mining optimized fuzzy association rules," Soft computing, Vol. 10, pp. 578-586, 2006.
27. C. Kuok, A. Fu and M. Wong, "Mining fuzzy association rules in databases," *SIGMOD Record*, Vol. 27, No. 1, pp. 41-46, 1998.
28. 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.
29. H. Roubos and M. Setnes, "Compact and transparent fuzzy models and classifiers through iterative complexity reduction," *IEEE Transactions on Fuzzy Systems*, Vol. 9, No. 4, pp. 516-524, 2001.
30. J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," *The International Conference on Genetic Algorithms*, pp. 93-100, 1985.
31. R. Srikant and R. Agrawal, "Mining quantitative association rules in large relational tables," *The 1996 ACM SIGMOD International Conference on Management of Data*, Monreal, Canada, June 1996, pp. 1-12.
32. C. H. Wang, T. P. Hong and S. S. Tseng, "Integrating membership functions and fuzzy rule sets from multiple knowledge sources," *Fuzzy Sets and Systems,* Vol. 112, pp. 141-154, 2000.
33. S. Yue, E. Tsang, D. Yeung and D. Shi, "Mining fuzzy association rules with weighted items," *The IEEE International Conference on Systems, Man and Cybernetics*, pp. 1906-1911, 2000.
34. Z. Zhang, Y. Lu and B. Zhang, "An effective partitioning-combining algorithm for discovering quantitative association rules," *The Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 261-270, 1997.
35. E. Zitzler, M. Laumanns and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," *Proc. Evolutionary Methods for Design, Optimization and Control with App. to Industrial Problems* (Barcelona, Spain, 2001) pp. 95-100.