International Journal of Innovative Computing, Information and Control
Volume **8**, Number **10(A)**, October **2012**

ICIC International ©2012 ISSN 1349-4198
pp. **6551–6568**

MULTI-OBJECTIVE GENETIC-FUZZY DATA MINING**

Chun-Hao Chen[1], Tzung-Pei Hong[2,3,*], Vincent S. Tseng[2] and Lien-Chin Chen[4]

[1]Department of Computer Science and Information Engineering
Tamkang University
No. 151, Yingzhuan Rd., Tamsui Dist., New Taipei City 25137, Taiwan
chchen@mail.tku.edu.tw

[2]Department of Computer Science and Information Engineering
National Cheng Kung University
No. 1, University Rd., Tainan City 701, Taiwan
tsengsm@mail.ncku.edu.tw

[3]Department of Computer Science and Engineering
National Sun Yat-sen University
No. 70, Lienhai Rd., Kaohsiung 80424, Taiwan
*Corresponding author: tphong@nuk.edu.tw

[4]Institute of Information Science, Academia Sinica
No. 128, Academia Rd., Section 2, Nankang, Taipei 115, Taiwan
lcchen@iis.sinica.edu.tw

Received June 2011; revised October 2011

Abstract. *Many approaches have been proposed for mining fuzzy association rules. The membership functions, which critically influence the final mining results, are difficult to define. In general, multiple criteria are considered when defining membership functions. In this paper, a multi-objective genetic-fuzzy mining algorithm is proposed for extracting membership functions and association rules from quantitative transactions. Two objective functions are used to find the Pareto front. The first one is the suitability of membership functions. It consists of the coverage factor and the overlap factor and is used to avoid two unsuitable types of membership function. The second one is the total number of large 1-itemsets from a given set of minimum support values. Experimental results 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

1. **Introduction.** Data mining is commonly used to derive 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 [2]. It means that 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. Most previous studies focused on binary-valued transaction data. Transaction data in real-world applications, however, usually consist of quantitative values. Many sophisticated data mining approaches have thus been proposed [1,26,30].

Fuzzy set theory is increasingly used in intelligent systems due to its simplicity and similarity to human reasoning. The theory has been applied in fields such as manufacturing, engineering, and economics [11]. Many approaches have been proposed for mining

**This is a modified and expanded version of the paper "A multi-objective genetic-fuzzy data mining algorithm", presented at The IEEE International Conference on Granular Computing, 2008, China.

fuzzy association rules [3,14,21,22,29]. They can be divided into two types, namely those that solve single-minimum-support fuzzy-mining (SSFM) and multiple-minimum-support fuzzy-mining (MSFM) problems, respectively. Most approaches have been proposed for the SSFM problem [3,14,21,29], in which a single minimum support threshold is set for all the items or itemsets. In real applications, different criteria may be used to judge the importance of different items and quantitative data may exist. Lee et al. thus proposed a mining algorithm which uses multiple minimum support values of different items to mine fuzzy association rules for the MSFM problem [22].

In fuzzy data mining, the membership functions critically influence the final mining results. However, it is difficult to define an appropriate set of membership functions for items. Most existing fuzzy data mining algorithms thus assume that the membership functions are already known. Pre-defined membership functions are not, however, suitable in practice. Mining algorithms that can automatically derive both the appropriate membership functions and the fuzzy rules are thus required. Many approaches have been proposed for deriving membership functions for both SSFM [5,12,13,17,18] and MSFM problems [4].

Several criteria may be considered in a real application. Multi-objective evolutionary algorithms, which are used to find a set of solutions with trade-offs among the criteria, are very suitable for solving such cases [7,8]. Kaya et al. proposed an approach that integrates a multi-objective genetic algorithm (GA) into clustering for fuzzy mining [19]. The number of large itemsets and the execution time were considered as two objective functions to derive appropriate membership functions for mining fuzzy association rules. Kaya also proposed an approach based on multi-objective GAs for mining optimized fuzzy association rules [20]. He defined three objectives, namely strongness, interestingness, and comprehensibility, to derive appropriate membership functions for mining optimized fuzzy association rules.

We previously proposed a genetic-fuzzy data mining algorithm for extracting association rules and membership functions from quantitative transactions [13]. Its fitness function was evaluated using the number of large 1-itemsets over the suitability of membership functions. The suitability measure was used to reduce the occurrence of unsuitable types of membership function. Using the number of large 1-itemsets instead of the number of rules provides a trade-off between execution time and rule interestingness. The two criteria, the number of large 1-itemsets and the suitability of membership functions, also have a trade-off relationship. In real-world applications, decision makers need diverse information to make good marketing strategies. Only a solution derived by using a genetic-fuzzy data mining algorithm [13] may be insufficient. An approach which can yield a spectrum of solutions under different criteria is thus needed.

The study proposes a multi-objective genetic-fuzzy mining approach for finding the Pareto solutions based on the two objective functions for deriving membership functions for the SSFM problem. Experimental results both on simulated data and on a real application show the effectiveness of the proposed algorithm. Especially, the two main contributions are listed as follows.

1. The proposed approach takes the number of large 1-itemsets and the suitability of membership functions into consideration for mining appropriate sets of membership functions.

2. The derived set of non-dominated solutions can provide multiple points of view for analysis. For example, if an analyst needs more knowledge mined from the transactions, then the solution with the largest total number of large 1-itemsets can be used to mine fuzzy association rules; on the contrary, if the shapes of membership functions

are emphasized, then the solution with the best suitability value can be used for further analysis.

The rest of this paper is organized as follows. Multi-objective optimization problems are introduced in Section 2. The details of the adjustment process for membership functions are explained in Section 3. The proposed algorithm for mining membership functions and association rules is described in Section 4. An example to illustrate the proposed algorithm is given in Section 5. Experiments that demonstrate the performance of the proposed algorithm are described in Section 6. Conclusions and future work are given in Section 7.

2. **GA-Based Multi-objective Optimization Problems.** In traditional optimization problems, the goals to be achieved are usually transformed into fitness functions for maximization or minimization. Unfortunately, it is difficult to find the best fitness function for a problem. Several criteria may be considered in a real application, such that multi-objective optimization problems become more and more important. Formally, a multi-objective optimization problem can be defined as follows:

$$\text{Min/Max } y = g(x) = (g_1(x), g_2(x), \ldots, g_m(x))$$
$$\text{subject to } x = (x_1, x_2, \ldots, x_n) \in X \text{ and } y = (y_1, y_2, \ldots, y_m) \in Y$$

where x is the decision vector, y is the objective vector, X represents the decision space, and Y represents the objective space. Several GA-based approaches have been proposed to solve this problem. Schaffer proposed the Vector Evaluated Genetic Algorithm (VEGA) for solving multi-objective optimization problems [25]. The difference between VEGA and a simple genetic algorithm is the selection strategy. For a problem with m objective functions, VEGA first divides the population into m sub-populations, one for each objective. Assume that the population size is P. P/m chromosomes are then selected from each sub-population and all the selected chromosomes were gathered to form the next population. Fonseca et al. pointed out that VEGA has two problems [10]. The first one is that two non-dominated individuals are sampled at different rates. The second one is that the population tends to split into different species. They thus proposed a modified approach called the Multi-Objective Genetic Algorithm (MOGA) that uses the extended rank-based fitness assignment [10] for solving the above two problems. They also defined three relationships among chromosomes, namely inferiority, superiority and non-inferiority, which are shown in Figure 1 [10].

As shown in Figure 1, the first relationship is inferiority. Since the two objective values of node N_1 are larger than the corresponding values of node N_2, the latter is said to be inferior to the former (Figure 1(a)), or that N_1 is superior to N_2 (Figure 1(b)). The third relationship is non-inferiority. In Figure 1(c), one objective value (x-axis) of node

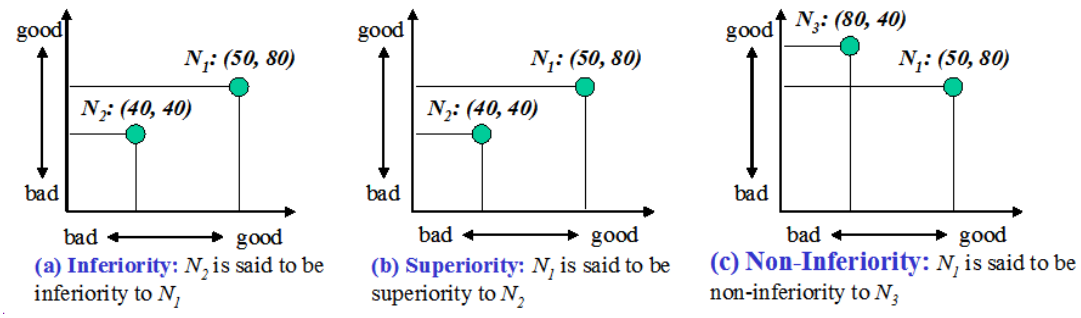

FIGURE 1. Three relationships among chromosomes in MOGA

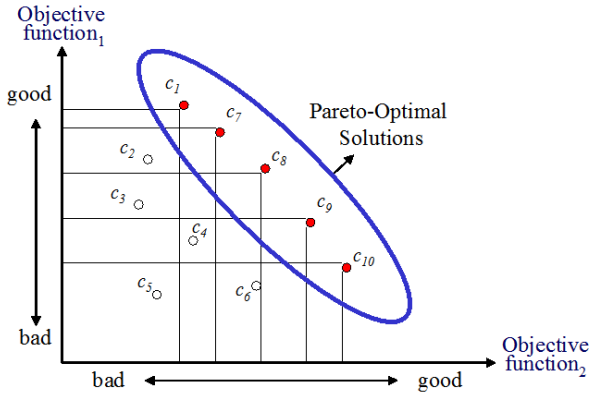

FIGURE 2. Example of Pareto-optimal solutions

N_1 is larger than the corresponding value of node N_3, and the other is smaller than the corresponding value. In this case, N_1 is said to be non-inferior to N_3.The MOGA strategy is used to find the set of non-inferior solutions, also called the Pareto-optimal solutions or the Pareto front. Figure 2 explains the three relationships and the Pareto-optimal solutions.

In Figure 2, there are ten chromosomes and two objectives. The two objective values of a chromosome are represented by a data point in the figure. Take chromosomes C_1 and C_2 as an example. Chromosome C_2 is said to be inferior to C_1 since the two objective values of C_2 are worse than the corresponding values of C_1. In this case, C_2 is dominated by C_1. Chromosome C_1 is said to be superior to C_2 and that it dominates C_2. Chromosome C_1 is said to be non-inferior to C_7 or vice versa. In this case, C_1 and C_7 are non-dominated points. The goal of MOGA is thus to find the non-dominated points, also called the Pareto-optimal solutions. In this example, chromosomes C_1, C_7, C_8, C_9 and C_{10} are non-dominated points. Some variants of MOGA have been proposed. Two well-known approaches are NSGA-II [9] and SPEA2 [31], whose goal is to obtain better Pareto fronts. NSGA-II uses a fast non-dominated sorting procedure, an elitist strategy, and an approach without parameters [9]. SPEA2 adopts a fine-grained fitness assignment strategy, a density estimation technique, and an enhanced archive truncation method [31].

3. Multi-objective Genetic-Fuzzy Mining Approach.

This study proposes a MOG A-based approach to derive a set of non-dominated solutions for mining problems. The details of the proposed approach are described below.

3.1. Chromosome representation.

It is important to encode membership functions as a string representation for the application of GAs. Several encoding approaches are described in [6,24,27,28]. In this study, the set of membership functions for an item is encoded as shown in Figure 3.

In Figure 3, each membership function is assumed to be an isosceles triangle represented by (c, w), where c is the center abscissa and w is half the span. R_{jk} denotes the membership function of the k-th linguistic term of item I_j. All (c, w) pairs 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 function (e.g., non-isosceles triangles and trapezes) can also be adopted. For coding non-isosceles triangles and trapezes, three and four points

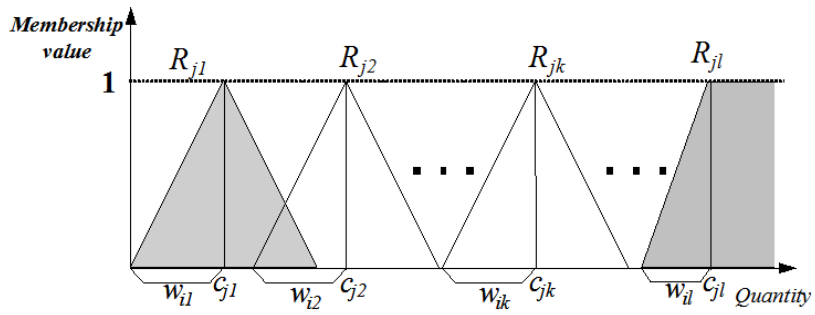

FIGURE 3. Set of membership functions for item I_j

are needed, respectively. The numbers of membership functions for the given items can vary.

3.2. **Initial population.** A GA 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. **Two objective functions.** Kaya et al. proposed an approach to derive membership functions for mining problems [17]. It finds the 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 proposed a genetic-fuzzy approach to learn an appropriate set of membership functions for mining problems [13]. In that paper, the fitness values were evaluated using 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 the present study, the mining of membership functions and fuzzy association rules is considered as a multi-objective optimization problem, in which the above two factors are used as two objective functions. A MOGA-based mining algorithm is proposed to find the Pareto-optimal solutions. The first objective function (Obj_1) for 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 item I_j in chromosome C_q and is defined as:

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

where $overlap(R_{jk}, R_{ji})$ denotes the overlap lengths of R_{jk} and R_{ji}. $coverage\_factor(C_{qj})$ represents the coverage ratio of a set of membership functions for item I_j in chromosome

C_q and is defined as:

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

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 unsuitable kinds of membership function, as shown in Figure 4, where the first one is too redundant and the second one is too separated.

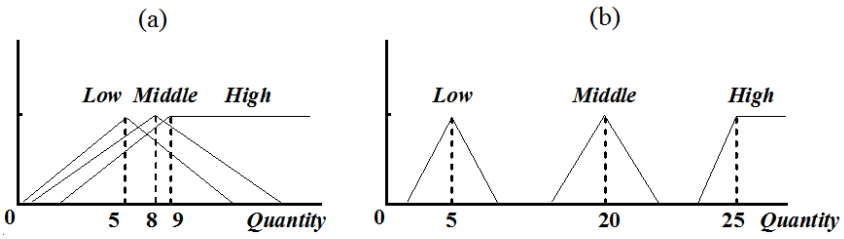

FIGURE 4. Two unsuitable membership functions

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:

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

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 provides a trade-off between execution time and rule interestingness. Usually, a larger number of 1-itemsets results in a larger number of all itemsets with a higher probability, which usually yields more interesting association rules. The proposed approach uses the two objective functions to find the appropriate Pareto solutions for the SSFM problem.

3.4. **Fitness assignment.** The fitness assignment is similar to that used in MOGA [10]. There are three steps: ranking chromosomes, assigning fitness, and averaging fitness values of the individuals with the same rank. The ten chromosomes in Figure 1 are ranked in Figure 5 according to their two objective values.

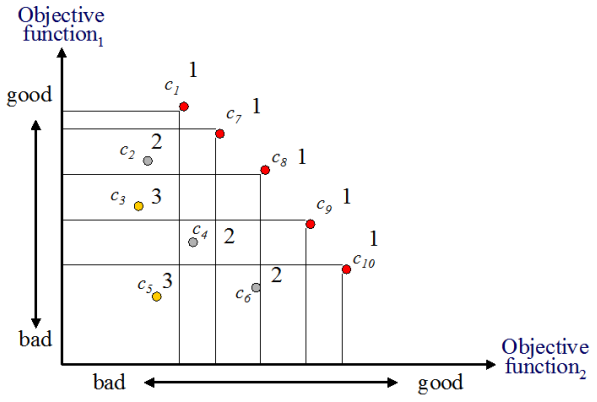

FIGURE 5. Ranking results of the ten chromosomes in Figure 1

In Figure 5, a chromosome with a lower ranking value has better quality. chromosomes with ranking values of 1 are non-dominated solutions. The fitness value of a chromosome is then assigned according to its rank value. For chromosome C_q with a ranking value of 1, its fitness value is assigned as:

$$f(C_q) = DomonatedBy(C_q)/(P+1),$$

where $DominatedBy(C_q)$ is the number of chromosomes dominated by chromosome C_q and P is the population size. For a chromosome with a ranking value larger than 1, its fitness value is assigned as:

$$f(C_q) = 1 + \sum_{C_p \in P \text{ and } C_p \text{ dominates } C_q} f(C_p),$$

where $f(C_p)$ is the fitness value of chromosome C_p, which dominates chromosome C_q. The constant value 1 is used here to ensure that the fitness value of a dominated chromosome is larger than that of a non-dominated chromosome. Therefore, a chromosome with a smaller fitness value is considered better. For instance, chromosome C_1 in Figure 5 dominates three chromosomes. Its fitness value is thus 3/11 (0.27). The fitness values of chromosomes C_7, C_8, C_9 and C_{10} are 0.36, 0.36, 0.27 and 0.18, respectively. Chromosome C_2 is dominated by C_1 and C_7. Its fitness value is thus calculated as $1+0.27+0.36 = 1.63$. The results of the other chromosomes are shown in Figure 6.

There are five non-dominated chromosomes in Figure 6; their fitness values are not al the same. Since they are all non-dominated, they are assumed to have equal importance

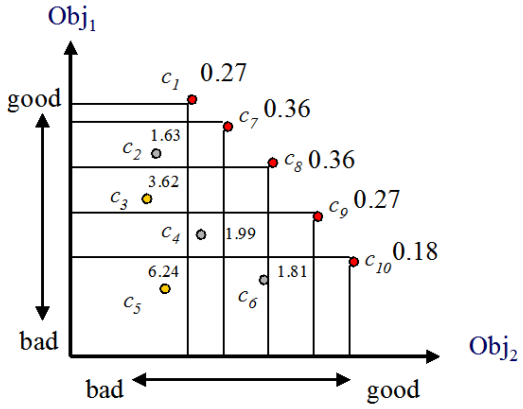

FIGURE 6. Fitness values of the ten chromosomes in Figure 1

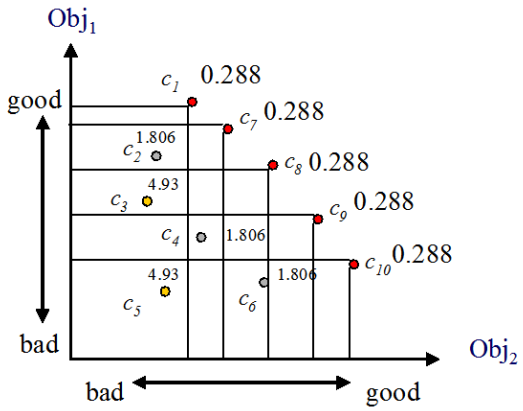

FIGURE 7. Average fitness values of the ten chromosomes in Figure 1

to be reproduced in the selection procedure. Therefore, instead of the original fitness values, the average fitness value of the non-dominated chromosomes is calculated and assigned to each of them. In this example, the average fitness value of the non-dominated chromosomes is $0.288(= (0.27 + 0.36 + 0.36 + 0.27 + 0.18)/5)$. The fitness values for the chromosomes with the same ranks are also calculated in this way. The results for all chromosomes are shown in Figure 7.

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 [16] and the one-point mutation, are used in the proposed approach. Assume that there are two parent chromosomes:

$$C_u^t = (c_1, \ldots, c_h, \ldots, c_Z),$$
$$C_w^t = (c_1^{'}, \ldots, c_h^{'}, \ldots, c_Z^{'}).$$

The MMA crossover operator generates 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 creates 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}, respetively. The center or the spread of the newly derived membership function changes 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 need to be rearranged according to their center values. The proposed approach can use either the elitist or the roulette-wheel selection strategy.

4. **Proposed Mining Algorithm.** The proposed multi-objective genetic-fuzzy algorithm for mining membership functions and fuzzy association rules is described below.

Multi-Objective Genetic-Fuzzy Mining Algorithm:

INPUT: A body of n quantitative transactions, a set of m items, each with a number of linguistic terms, a population size P, a crossover rate P_c, a mutation rate P_m, a set of h minimum support values, 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 of P individuals, with each one being a set of membership functions for all 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 set *NDS* as empty.

STEP 2: For each chromosome C_q, calculate its two objective values, the suitability ($sutiability(C_q)$) and the total number of large 1-itemsets in the given set of minimum support values ($totalNumL1(C_q)$), as follows:

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 $f_j^{(i)}$ represented as:

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

using the corresponding membership functions represented by the chromosome, where R_{jk} is the k-th fuzzy region (term) of item I_j, $f_{jk}^{(i)}$ 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: Calculate the suitability value $suitability(C_q)$ using the formula defined in Section 3; set it as the first objective value of C_q.

SUBSTEP 2.4: For each R_{jk}, $1 \le j \le m$, $1 \le k \le |I_j|$, and for each minimum support value ms_g, $1 \le g \le h$, check whether $count_{jk}$ is larger than or equal to the minimum support value ms_g. 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 using the set of membership functions in chromosome C_q and the minimum support value ms_g; let $totalNumL1(C_q) = \sum_{g=1}^{h} |L_{1q}^{ms_g}|$ as the second objective value of C_q.

STEP 3: Rank the chromosomes according to the two objectives, $suitability(C_q)$ and $totalNumL1(C_q)$, as follows:

SUBSTEP 3.1: Set the variable r for representing the current rank, which is initially at 0.

SUBSTEP 3.2: Find the non-dominated chromosomes among the unranked ones in the population, set $r = r+1$, and set the ranking values of the non-dominated chromosomes as r.

SUBSTEP 3.3: If there are still unranked chromosomes in the population, go to SUBSTEP 3.2; otherwise, go to the next step.

STEP 4: Calculate the fitness value of each chromosome based on the ranking value as follows:

SUBSTEP 4.1: Calculate the fitness values of the chromosomes with their ranking values equal to one as follows:

$$f(C_q) = DomonatedBy(C_q)/(P+1),$$

where $DominatedBy(C_q)$ is the number of chromosomes dominated by chromosome C_q and P is the population size.

SUBSTEP 4.2: Calculate the fitness values of the chromosomes with their ranking values larger than one as follows:

$$f(C_q) = 1 + \sum_{C_p \in P \text{ and } C_p \text{ dominates } C_q} f(C_p),$$

where $f(C_p)$ is the fitness value of chromosome C_p which dominates chromosome C_q and the constant value 1 is used to ensure that the fitness

values of dominated chromosomes are larger than those of non-dominated ones.

STEP 5: Calculate the average fitness values of the chromosomes with the same ranking values such that each of them can be selected equally by the selection strategy.

STEP 6: Copy the chromosomes with ranking values equal to one into the non-dominated set *NDS* and remove the chromosomes which are dominated by other chromosomes in *NDS*.

STEP 7: Execute the crossover operation on the population.

STEP 8: Execute the mutation operation on the population.

STEP 9: Calculate the fitness values of the new chromosomes using STEPS 2 to 8.

STEP 10: Use the selection operation to choose appropriate individuals from the set of *NDS* to form the next generation. Here, the selection strategy can be elitist or roulette wheel. If the size of *NDS*, called *NDSSize*, is less than the population size, *PSize*, all the chromosomes in *NDS* are copied into the next population and the number (*PSize* – *NDSSize*) of chromosomes are selected from the difference set of the offspring chromosomes and the current *NDS*.

STEP 11: If the termination criterion is not satisfied, go to STEP 6; otherwise, go to the next step.

STEP 12: Execute the truncation operator proposed in [31] on the non-dominated set *NDS* to find the best k solutions. Since there may be more than one chromosome kept in *NDS*, the goal of this step is to keep the k representative solutions at the Pareto front. Note that this step is optional.

STEP 13: Mine fuzzy association rules from the given database and based on the derived chromosomes in *NDS* (or the k representative chromosomes if STEP 12 is applied), where each chromosome represents a set of membership functions. The fuzzy mining algorithm proposed in [15] is adopted for this purpose for each set of membership functions.

STEP 14: Output the non-dominated set *NDS* and the corresponding fuzzy association rules.

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

Assume that each item has three fuzzy regions, namely *Low*, *Middle* and *High*, for simplicity. Thus, three fuzzy membership functions must be derived for each item. Note

Table 1. 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).

TABLE 2. Fuzzy sets transformed from the data in Table 1

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

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 initialized as empty. In this example, P is set to 10. Each individual is thus a set of membership functions for the four items. Assume that 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 support values of each chromosome are calculated as follows:

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.

SUBSTEP 2.2: The scalar cardinality of each fuzzy region in the transactions is calculated as the *count* value. Take the fuzzy region $milk.Midd\text{-}le$ 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.

SUBSTEP 2.3: The suitability value of 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 that the set of minimum support values is {0.08, 0.09, 0.1, ..., 0.17}. Take the minimum support value set to 0.08 as an

TABLE 3. Counts for 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

TABLE 4. Suitability value and total $|L_1|$ of each chromosome

C_q	(Suitability, total $\|L_1\|$)	C_q	(Suitability, total $\|L_1\|$)
C_1	**(8.38, 69)**	C_6	**(8.51, 62)**
C_2	**(9.72, 95)**	C_7	**(7.79, 68)**
C_3	**(8.30, 78)**	C_8	**(8.36, 68)**
C_4	**(8.88, 58)**	C_9	**(9.98, 68)**
C_5	**(8.62, 87)**	C_{10}	**(8.09, 77)**

example. Since the count values of *milk.Low*, *milk.Middle*, *milk.High*, *bread.High*, *cookies.Hi-gh*, *beverage.Low* and *beverage.Middle* are larger than $0.48(= 0.08 * 6)$, the number of large 1-itemsets is 7. The number of large 1-itemsets for the other minimum support values can be similarly found. The total number of large 1-itemsets $totalNumL1$ (C_1) is thus $69(= 7 + 7 + 7 + 7 + 7 + 7 + 7 + 7 + 7 + 6)$. The two objective values of chromosome C_1 are thus 8.38 and 69. The results of all ten chromosomes are shown in Table 4.

STEP 3: The ranking procedure is executed to rank the ten chromosomes according to the two objectives, $suitability(C_q)$ and *totalNumL1*, as follows:

SUBSTEP 3.1: Set the variable r for the current rank initially at 0.

SUBSTEP 3.2: The non-dominated chromosomes are found according to the two objectives. In this example, the non-dominated chromosomes are C_2, C_3, C_5, C_7 and C_{10}. r is thus set to $0 + 1(= 1)$. The ranking values of the chromosomes are set to 1.

SUBSTEP 3.3: Since there are unranked chromosomes in the population, SUBSTEP 3.2 is repeated to rank the other chromosomes. The next non-dominated chromosomes are then found from the remaining (unranked) chromosomes in the initial population. They are C_1 and C_8. Their ranking values are then 2. The ranking results of all ten chromosomes are shown in Table 5.

TABLE 5. Ranking results of all ten chromosomes

Ranking	Chromosomes
1	$C_2, C_3, C_5, C_7, C_{10}$
2	C_1, C_8
3	C_6, C_9
4	C_4

STEP 4: The fitness value of each chromosome is calculated based on its ranking value as follows:

SUBSTEP 4.1: The chromosomes with ranking values of 1 are first evaluated. In this example, there are five chromosomes that satisfy the condition, as shown in Table 5. Take chromosome C_3 as an example. Chromosomes C_1, C_4, C_6, C_8 and C_9 are dominated by C_3. The number of chromosomes dominated by C_3 is thus 5. The population size is 10. The fitness value of C_3 is thus $5/(10+1) = 0.45$. The fitness values of C_2, C_5, C_7 and C_{10} can be similarly calculated as 0.09, 0.18, 0.36 and 0.45, respectively.

SUBSTEP 4.2: The fitness values of the chromosomes with ranking values larger than one are then calculated. Two chromosomes, C_1 and C_8, have a ranking value of 2. Take C_1 as an example. It is dominated by C_3 and C_{10}, whose fitness values are both 0.45. The fitness value of C_1 is thus $1+0.45+0.45 = 1.90$. The fitness values of all ten chromosomes are shown in Table 6.

TABLE 6. Fitness values of all ten chromosomes

C_q	Fitness	C_q	Fitness
C_1	1.90	C_6	6.45
C_2	0.09	C_7	0.36
C_3	0.45	C_8	2.27
C_4	13.09	C_9	6.72
C_5	0.18	C_{10}	0.45

STEP 5: The average fitness values of the chromosomes with the same ranking values are calculated. Take chromosomes C_1, C_4, C_6, C_8 and C_9, which have the same ranking value of 1, as an example. The average value of these chromosomes is $(0.09 + 0.45 + 0.18 + 0.36 + 0.45)/5 = 0.309$. The resulting fitness values of all ten chromosomes are shown in Table 7.

TABLE 7. Resulting fitness values of ten chromosomes

C_q	Fitness	C_q	Fitness
C_1	2.090	C_6	6.59
C_2	0.309	C_7	0.309
C_3	0.309	C_8	2.09
C_4	13.09	C_9	6.59
C_5	0.309	C_{10}	0.309

STEP 6: The chromosomes with ranking values equal to 1 are copied into the non-dominated set *NDS*. That is, $NDS = \{C_2, C_3, C_5, C_7, C_{10}\}$.

STEPS 7 to 10: The crossover and the mutation operations are executed on the population. Here, the MMA crossover operator and one-point mutation operator are used for generating new offspring. The fitness values of the new chromosomes are calculated using STEPS 3 to 7. The selection operation is then used on the non-dominated set *NDS* to choose appropriate individuals for the next generation. In this example, chromosomes C_2, C_3, C_5, C_7 and C_{10} in *NDS* are first kept in the next generation. The other five chromosomes are selected from the current population according to

their fitness values. These ten selected chromosomes are then used as the next population.

STEPS 11 to 14: If the termination criterion is not satisfied, go to STEP 6; otherwise, the chromosomes in the non-dominated set *NDS* are output as the sets of membership functions for deriving fuzzy association rules. The fuzzy mining method proposed in [15] is then used to mine fuzzy association rules for each set of membership functions.

6. **Experimental Results.** In this section, experiments conducted to show the performance of the proposed approach are described. They were implemented in Java on a personal computer with an Intel Pentium IV 3.20-GHz CPU and 512MB of RAM. Two simulated datasets and a real foormart database are used in the experiments. In the simulated datasets, 64 items and 10000 transactions were used in the experiments. The initial population size P was set to 50, the crossover rate p_c was set to 0.8, and the mutation rate p_m was set to 0.001. The parameter d of the crossover operator was set to 0.35, following to Herrera et al. [16] and the set of minimum support values was $\{3\%, 4\%, \ldots, 13\%\}$. In the following subsections, the experimental datasets are described. The evolution of the Pareto fronts obtained using the proposed approach is then analyzed. The effects of the minimum support and minimum confidence values are then analyzed.

6.1. **Description of experimental datasets.** Two simulated datasets with 64 items and 10,000 transactions were used in the experiments. One dataset followed a uniform distribution and the other followed an exponential distribution. The parameters of 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 in the range [1,19] for both the datasets. The purchased items in each transaction were then selected from the 64 items in a uniform distribution in the range [1,64] and in an exponential distribution with the rate parameter set to 16, respectively. Their quantities were then assigned from a uniform distribution in the range [1,11] and from an exponential distribution with the rate parameter set to 5, respectively. The simulation process was terminated when the desired dataset size was reached. An item could not be generated twice in a transaction.

In addition, the foodmart database [23] was also used to evaluate the performance of the algorithms under various thresholds. The Microsoft SQL Server 2000 was used for keeping the foodmart database. The database has 21, 556 transactions and 1,600 different items.

6.2. **Evolution of Pareto fronts obtained using proposed approach.** The first experiment was conducted to demonstrate the evolution of the Pareto fronts obtained using the proposed approach. The evolution of the Pareto fronts for the uniform-distribution and exponential-distribution datasets are shown in Figures 8 and 9, respectively.

Figure 8 shows that the solutions were distributed widely on the Pareto fronts in different generations. The final solutions (after 500 generations) were the best. Figure 9 shows that the solutions were distributed on the Pareto fronts, but a little narrow. The final solutions (after 500 generations) were the best. The evolution of the Pareto fronts obtained using the proposed approach for the foodmart dataset is shown in Figure 10.

Figure 10 shows that the non-dominated solutions were distributed widely on the Pareto fronts in different generations. The final non-dominated solutions (after 500 generations) were the best. From the derived set of non-dominated solutions, if an analyst needs more knowledge amount from the transactions, then the non-dominated solution with the largest total number of large 1-itemsets can be used to mine the fuzzy association rules.

If an analyst emphasizes on the shapes of membership functions, then the non-dominated solution with the the best suitability value can be used to derive the fuzzy association rules. Usually, an analyst can get a solution between the above two. The experimental

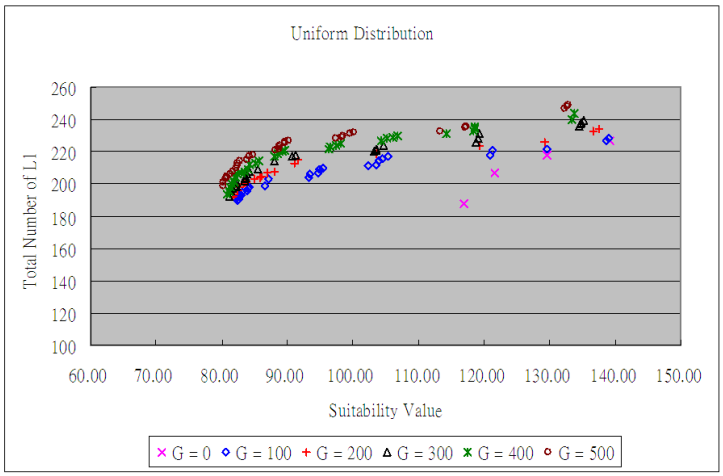

FIGURE 8. Evolution of Pareto fronts for uniform-distribution dataset

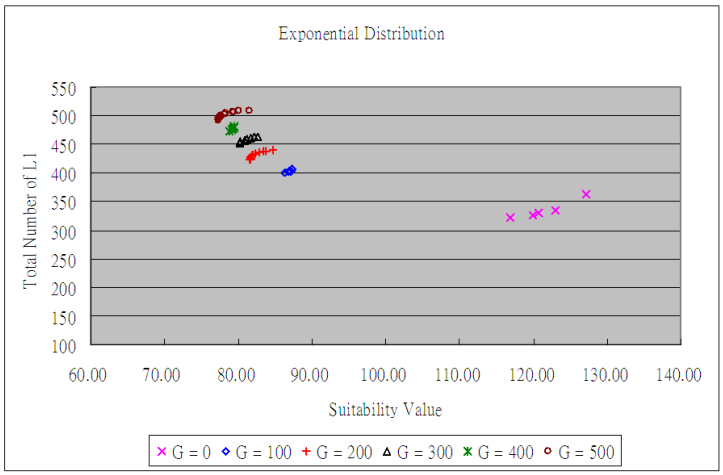

FIGURE 9. Evolution of Pareto fronts for exponential-distribution dataset

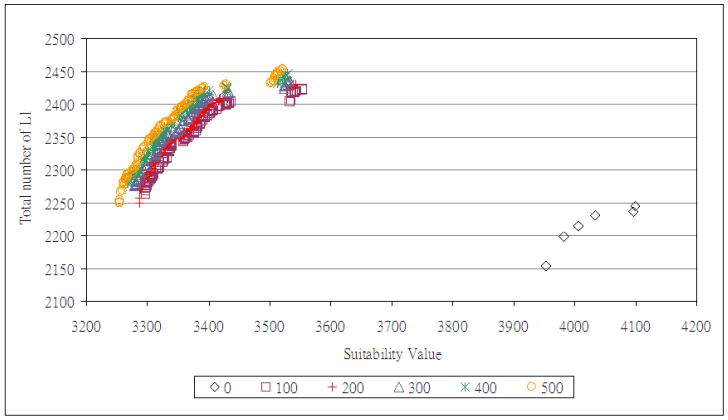

FIGURE 10. Evolution of Pareto fronts for the foodmart dataset

results confirmed that the proposed approach is effective in finding an appropriate set of solutions for further analysis.

6.3. **Effects of minimum support and minimum confidence values.** Experiments were conducted to analyze the effects of minimum support and minimum confidence values on the results obtained using the proposed approach. The truncation operator proposed in [31] was first applied to the final non-dominated set *NDS* for finding k solutions. Here, the parameter k was set to 10. To demonstrate the distribution of the solutions, the two extreme solutions were picked from the ten chromosomes for comparison. The first one had the highest total number of large 1-itemsets and the second one had the best suitability value. The relationship between the number of rules and the minimum confidence with the minimum support set to 0.01, 0.03 and 0.05, respectively, for the two extreme solutions for the uniform-distribution dataset is shown in Figure 11, and that with the minimum support set to 0.01, 0.02 and 0.03, respectively, for the two extreme solutions for the exponential-distribution dataset is shown in Figure 12.

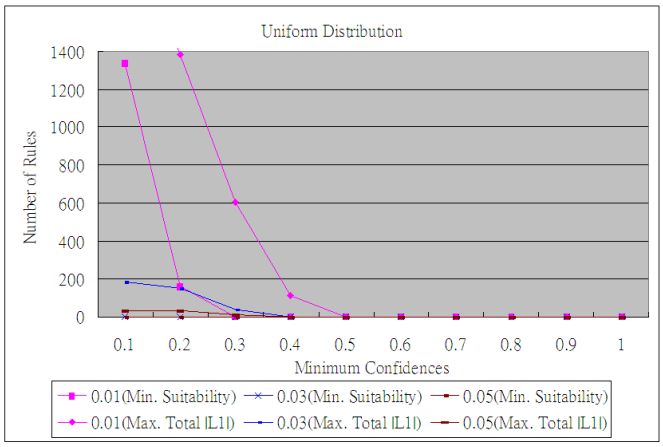

FIGURE 11. Relationship between the number of rules and the minimum confidence value for the two extreme solutions for the uniform-distribution dataset

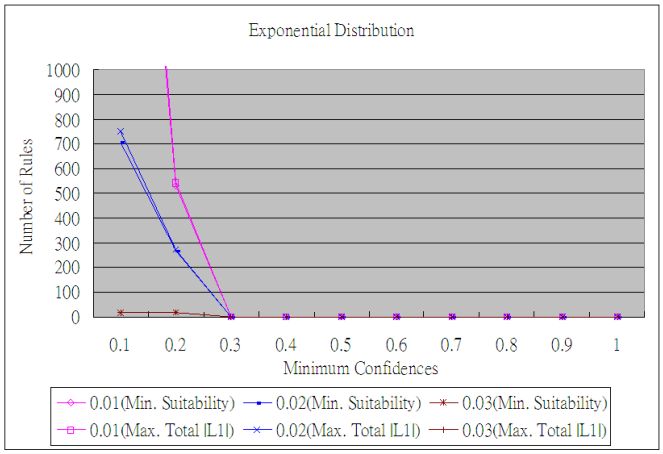

FIGURE 12. Relationship between the number of rules and the minimum confidence value for the two extreme solutions for the exponential-distribution dataset

Figure 11 shows that the number of rules derived using the proposed approach decrease with increasing minimum confidence value. The results obtained using various minimum support values show similar behavior. The number of rules derived from the first extreme solution (with the highest total number of large 1-itemsets) was larger than that derived from the second one (with the best suitability value). This was due to the two objective functions focusing on different goals. The solutions on a Pareto front are a trade-off between the two objectives. The user can decide which solutions are desired. In Figure 12, similar results were obtained although the difference between the two extreme solutions is small. The proposed approach can thus provide different options to users for further analysis.

7. **Conclusion and Future Works.** A multi-objective genetic-fuzzy mining algorithm for extracting membership functions from quantitative transactions for the SSFM problem was proposed. Two objective functions, namely $suitability(C_q)$ and $totalNumL1$, are used to find the Pareto front solutions. $suitability(C_q)$ is used to reduce the occurrence of two unsuitable kinds of membership function and *totalNumL1* is used to derive more information. The proposed approach has a trade-off between the quality of membership functions and the number of interesting rules. Experimental results show that the proposed approach is effective in finding the Pareto front solutions. In the future, we will enhance the multi-objective genetic-fuzzy approach for more complex problems, such as solving the MSFM problem. The deficiency of the proposed approach is when the item number is large, the convergence may need much execution time. Thus, another future research is to efficiently and effectively handle the high-dimensional mining problem. We may consider designing appropriate pre-processing techniques or using clustering approaches for reducing evaluation time.

Acknowledgment. This research was supported by the National Science Council of Taiwan under grant 100-2221-E-032-065-.

REFERENCES

[1] R. Agrawal, T. Imielinksi and A. Swami, Database mining: A performance perspective, *IEEE Transactions on Knowledge and Data Engineering*, vol.5, no.6, pp.914-925, 1993.

[2] R. Agrawal and R. Srikant, Fast algorithm for mining association rules, *Proc. of the International Conference on Very Large Databases*, pp.487-499, 1994.

[3] C. C. Chan and W. H. Au, Mining fuzzy association rules, *Proc. of the Conference on Information and Knowledge Management*, Las Vegas, NV, USA, pp.209-215, 1997.

[4] C.-H. Chen, T.-P. Hong, V. 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.

[5] C.-H. Chen, V. 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.

[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 Transactions on Fuzzy Systems*, vol.9, no.4, pp.667-674, 2001.

[7] C. A. Coello, D. A. Van Veldhuizen and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-objective Problems*, Kluwer Academic Publishers, 2002.

[8] K. Deb, *Multi-objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, 2001.

[9] 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, 2002.

[10] 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.

[11] W. Gu, G. Li and M. Yin, Extending fuzzy soft sets with fuzzy description logics, *ICIC Express Letters, Part B: Applications*, vol.2, no.5, pp.1001-1008, 2011.

[12] 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.
[13] 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.
[14] 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.
[15] 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.
[16] 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.
[17] M. Kaya and R. Alhajj, A clustering algorithm with genetically optimized membership functions for fuzzy association rules mining, *Proc. of the IEEE International Conference on Fuzzy Systems*, pp.881-886, 2003.
[18] M. Kaya and R. Alhaji, Genetic algorithms based optimization of membership functions for fuzzy weighted association rules mining, *The International Symposium on Computers and Communications*, vol.1, pp.110-115, 2004.
[19] M. Kaya and R. Alhajj, Integrating mutlti-objective genetic algorithms into clustering for fuzzy association rules mining, *Proc. of the IEEE International Conference on Data Mining*, pp.431-434, 2004.
[20] M. Kaya, Multi-objective genetic algorithm based approaches for mining optimized fuzzy association rules, *Soft Computing*, vol.10, pp.578-586, 2006.
[21] C. Kuok, A. Fu and M. Wong, Mining fuzzy association rules in databases, *SIGMOD Record*, vol.27, no.1, pp.41-46, 1998.
[22] 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.
[23] Microsoft Corporation, *Example Database FoodMart of Microsoft Analysis Services.*
[24] 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.
[25] J. D. Schaffer, Multiple objective optimization with vector evaluated genetic algorithms, *Proc. of the International Conference on Genetic Algorithms*, pp.93-100, 1985.
[26] R. Srikant and R. Agrawal, Mining quantitative association rules in large relational tables, *Proc. of the 1996 ACM SIGMOD International Conference on Management of Data*, Monreal, Canada, pp.1-12, 1996.
[27] C.-H. Wang, T.-P. Hong and S.-S. Tseng, Integrating fuzzy knowledge by genetic algorithms, *IEEE Transactions on Evolutionary Computation*, vol.2, no.4, pp.138-149, 1998.
[28] 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.
[29] S. Yue, E. Tsang, D. Yeung and D. Shi, Mining fuzzy association rules with weighted items, *Proc. of the IEEE International Conference on Systems, Man and Cybernetics*, pp.1906-1911, 2000.
[30] Z. Zhang, Y. Lu and B. Zhang, An effective partitioning-combining algorithm for discovering quantitative association rules, *Proc. of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp.261-270, 1997.
[31] E. Zitzler, M. Laumanns and L. Thiele, SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization, *Proc. of Evolutionary Methods for Design, Optimization and Control with App. to Industrial Problems*, Barcelona, Spain, pp.95-100, 2001.