Multifactorial Genetic Fuzzy Data Mining for Building Membership Functions

Ting-Chen Wang
Department of Computer Science
and Information Engineering
National Chung Cheng University
Chiayi County 62102, Taiwan
Email: wtc102p@cs.ccu.edu.tw

Rung-Tzuo Liaw
Department of Computer Science
and Information Engineering
Fu Jen Catholic University
New Taipei City 24205, Taiwan
Email: rtliaw@csie.fju.edu.tw

***Abstract*—Association mining is a famous data mining technology because its form is explainable by human beings. Innovating fuzzy set theory to associations mining provides a solution to quantitative database, where membership function plays an important role in mining fuzzy associations. Genetic algorithm (GA) has been successfully applied to the optimization of membership functions. Based on the spirit of divide-and-conquer, GA optimizes the membership functions for each item separately. Nevertheless, the cooperation among different items in the course of evolution was never considered. Evolutionary multitasking optimization (EMO) is an emerging searching paradigm which dedicates to solving multiple tasks simultaneously for improving the search efficiency. This study introduces the EMO into genetic fuzzy data mining to address the above issue. Specifically, this study incorporates a state-of-the-art genetic fuzzy data mining method, the structure-based representation genetic algorithm, with the well-known multifactorial evolutionary algorithm (MFEA). A series of experiments is conducted to validate the effectiveness and efficiency of the proposed method. The results indicate that the proposed method improves the structure-based representation genetic algorithm in terms of convergence speed and solution quality on all sizes of datasets. The results also show that the proposed method is about 20 times faster than the structure-based representation genetic algorithm with respect to the exploited number of evaluations.**

***Index Terms*—Evolutionary Multitasking, Multi-factorial, Genetic Fuzzy Data Mining, Structure-based Representation, Membership Function**

I. Introduction

The importance of data mining keeps increasing in recent decades [7], [8], [20]. Aiming at different targets, a variety of data mining technologies, for example, classification [4], clustering [25], and associations [13] have been proposed. Unlike black-box models such as neural networks, the associations extracted from database can be easily understood by persons due to their representation. One famous and successful application of associations is the analysis of customers' behavior in Walmart.

The most well-known algorithm for mining associations is the Apriori algorithm [1]. The basic idea of Apriori algorithm is to find the frequent itemsets from database. The limitation of the Apriori algorithm is that it only considers Boolean variables. By discretizing quantitative values Srikant and Agrawal [22] proposed a more general method based on Apriori algorithm. The other approach to tackle the problem of quantitative values in mining associations is the innovation of fuzzy set theory [3], [16], [17]. Specifically, the fuzzy transaction data mining algorithms (FTDAs) transform the quantitative values to fuzzy values by exploiting the concept of fuzzy sets and fuzzy logic [13], [14], [15]. The quantitative values are transformed into fuzzy values, which represent the degree of membership for a fuzzy set. This transformation fully relies on the membership functions, and the results form the fuzzy associations.

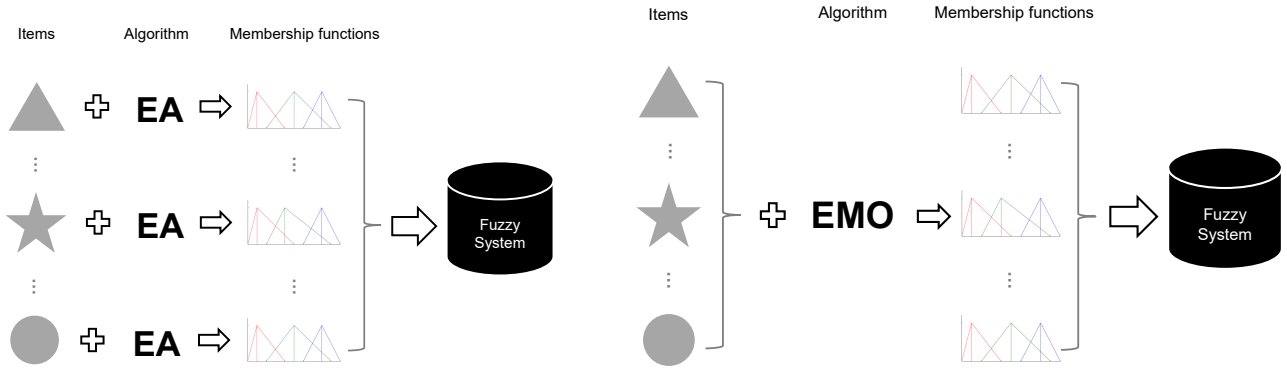
Membership function plays an important role in mining fuzzy associations. Each membership function depicts the degree of membership for a linguistic term, and it is used to map a quantitative value into a degrees value which is called fuzzy value. A mathematical formula of a membership function MF has the following form:

$$\mathrm{MF} : U \rightarrow [0, 1], \tag{1}$$

where U is the universal set, and the pair (MF, U) forms a fuzzy set. So far genetic algorithm (GA) has been successfully applied to the optimization of membership functions. Current GAs separate the optimization of membership functions by divide-and-conquer method such that each GA is responsible for an item. Nevertheless, these methods did not consider the cooperation among different items in the course of evolution.

In recent years, evolutionary multitasking optimization (EMO) becomes an emerging technique which intents on simultaneously tackling multiple tasks to leverage the similarity among tasks for improving the search efficiency [11], [26]. This study introduces the evolutionary multitasking into genetic fuzzy data mining to deal with above issue. Figure. 1 illustrates the difference between the traditional divide-and-conquer paradigm (1a) and the evolutionary multitasking paradigm (1b) for mining fuzzy associations. The EAs with divide-and-conquer paradigm optimize the membership functions of each item separately. Nonetheless, the evolutionary multitasking search paradigm optimizes the membership functions for all items concurrently. Specifically, this study integrates the structure-based representation genetic algorithm, a state-of-the-art genetic fuzzy data mining method, with the famous multi-factorial evolutionary algorithm (MFEA). This

978-1-7281-6929-3/20/$31.00 ©2020 IEEE



(a) Divide-and-conquer

(b) Multitasking

Figure 1: Two search paradigms for mining fuzzy associations: single-tasking with EAs and multitasking with evolutionary multitasking optimization (EMO)

study validates the effective and efficiency of the proposed structure representation based MFEA by conducting a series of experiments. Different sizes of datasets are considered to test the scalability from 10 thousand (10k) to 90 thousand (90k) transactions.

The contributions of this study are summarized as follows:

1) Introducing evolutionary multitasking into genetic fuzzy data mining to facilitate the cooperation among items
2) Incorporating the structure-based representation genetic algorithm with the well-known multi-factorial evolutionary algorithm
3) Examining the effectiveness and efficiency in terms of solution quality, convergence speed, and speedup through experiments

The remaining sections are organized as follows. Section II presents the acquaintance with fuzzy associations mining. Section III introduces the proposed method, and Section IV shows the experimental results. The concluding remarks are drawn in Section V.

II. Related Work

The most famous algorithm for mining the binary associations is the Apriori algorithm [1]. The Apriori algorithm finds large itemsets $L = \{L_1, \ldots, L_m\}$ by iteratively constructing candidate itemsets $C = \{C_1 \ldots, C_m\}$ and filtering out every candidate of which support is lower than the predefined minimum support θ_{supp}.

However, the associations depending upon binary transactional data concern the coexistence of items, but haven not taken the quantity of each item into account. Hong et al. [14], [15] proposed the utilization of fuzzy set theory for handling the issue of mining data with quantitative values. Introducing the fuzziness to the original associations forms the fuzzy associations. The counterparts of support in the fuzzy associations mining is the fuzzy support. Suppose there is a quantitative database D which has n transactions and m items. The fuzzy value of the j^{th} item I_j in the i^{th} transaction T_i, denoting as $f_{j,k}^{(i)}$, depends on the fuzzy region $R_{j,k}$, represented by the k^{th} membership function of item I_j.

Definition 1. Given a fuzzy region $R_{j,k}$, its fuzzy support is defined as the following equation:

$$\text{Fuzzy Support}\,(R_{j,k}) = \frac{1}{n}\sum_{i=1}^{n} f_{j,k}^{(i)}. \tag{2}$$

Similar to the Apriori algorithm, every fuzzy region $R_{j,k}$ of which fuzzy support is greater than or equal to the minimum support, i.e. $\text{Fuzzy Support}\,(R_{j,k}) \geq \theta_{\text{supp}}$, is joined to the large 1-itemset L_1.

Generalizing from one region to multiple regions, the fuzzy value of a set of regions $R = \{R_1, \ldots, R_p\}$ to the i^{th} transaction is defined by the intersection operator:

$$f_R^{(i)} = \bigcap_{k=1}^{p} f_{R_k}^{(i)}, \tag{3}$$

where a common intersection operator in fuzzy system is the minimum, and thus we have:

$$f_R^{(i)} = \min_{1 \leq k \leq p} f_{R_k}^{(i)}. \tag{4}$$

By taking the minimum as the intersection operator, it guarantees that the degree of membership for the joint region $R = \{R_1, \ldots, R_p\}$ will never exceed the marginal degree of membership for all regions $R_1, \ldots, R_p$ in R.

Consequently, we can derive the fuzzy support for a fuzzy regions R as follows.

Definition 2 (Fuzzy Support)**.** Given a set of fuzzy regions R, its fuzzy support is defined by following equation:

$$\text{Fuzzy Support}\,(R) = \frac{1}{n}\sum_{i=1}^{n} f_R^{(i)}. \tag{5}$$

If the fuzzy support of the set of fuzzy regions R is greater than or equal to the minimum support θ_{supp}, it becomes an elements of the large p-itemset L_p with $p = |R|$.

The transformation from quantitative value to fuzzy value and the basic elements of fuzzy associations are both dependent upon the membership functions. Therefore, the membership functions play a key role in the mining of fuzzy associations. The membership functions, serving as the mapping functions, have several shapes such as triangular, trapezoidal, and bell functions which are controlled by the parameters. The parameters of the membership functions also decide the fuzziness among membership functions. One of the most commonly adopted types is the triangular function with three points underdetermined. Optimizing the membership functions is essential in the mining of fuzzy associations. Many studies exploit evolutionary algorithms to optimize the membership functions due to their effectiveness. Hong et al. [14], [15] designed GAs and showed their ability to obtain proper membership functions. Hong et al. [13] further improved the efficiency of genetic fuzzy data mining algorithm through a divide-and-conquer paradigm. Chen et al. [6] imitated the fuzzy support of offspring by centers of clusters in population to reduce the calculation of fuzzy support. In addition, Chen et al. [5] proposed a method to mine the fuzzy coherent rules without the setting of minimum support. Rather than evolving the membership functions by the divide-and-conquer manner, Lee et al. [18], [19] devised a GA which encodes fuzzy association rule as its chromosome. In [24], Ting et al. proposed a genetic fuzzy data mining algorithm based on structure representation for ensuring the legality and suitability of membership functions.

Recently, a new class of evolutionary algorithm based on the multitasking paradigm, the multi-factorial evolutionary algorithm (MFEA), has been proposed for tackling different tasks at the same time [11], [12], [23]. Based on the ranking for each task, the individuals take the scalar fitness as their fitness value, which is the inversion of the minimum ranking over all tasks. On the aspect of fitness landscapes, the moving direction, which is benefit to one task, from a given point in the design space may also improve the fitness of the other task. Such phenomenon brings MFEA effectiveness and efficiency. In [10], the analysis of the synergy of fitness landscapes has been investigated through some numerical benchmark functions. Several variants of MFEA have been proposed for different applications. Gupta et al. [9] combined MFEA with a nested bi-level evolutionary algorithm to solve bi-level optimization problems in a multitasking paradigm. Sagarna and Ong [21] adopted MFEA to tackle the branch testing problem, which is one of the software testing problems. Chandra et al. [2] proposed an MFEA for optimizing several feed forward neural networks with different numbers of hidden layers at a time. Zhou et al. [26] adopted the MFEA on the capacitated vehicle routing problems. The MFEA has also been adopted on multi-objective optimization problem with 2 tasks [12]. Two performance metrics for multi-objective optimization problem, i.e., the nondominated front and crowding distance, are treated as different tasks. These studies show the possibility of evolutionary multitasking for handling multiple tasks at one time, which is also applicable to genetic fuzzy data mining.

Algorithm 1 The main procedure of MFEA

```
P ← Initialize()
for i ← 1 to m do
    Evaluate (P, τ_i)
end for
φ(P) ← Scalar Fitness(P)
τ(P) ← Skill Factor(P)
while Not terminated do
    P' ← Assotative Mating (P)
    Evaluate (P', τ(P'))
    S ← P ∪ P'
    φ(S) ← Scalar Fitness(S)
    τ(S) ← Skill Factor(S)
    P ← Survive (S)
end while
```

III. Method

This study introduces the idea of evolutionary multitasking to genetic fuzzy data mining fuzzy for improving the effectiveness and efficiency. Specifically, this study incorporates the structure-based representation genetic algorithm [24] with MFEA [11], forming the structure-based representation MFEA. There are two features in the structure-based representation MFEA: 1) Evolutionary multitasking search paradigm based on MFEA, and 2) structure-based representation for balancing the fuzzy support and suitability. The pseudocode of MFEA can be found in algorithm 1. The MFEA optimizes multiple tasks concurrently for efficiency. By treating the optimization of membership functions for each item as a single task, the MFEA can optimize membership functions for all items in one run. In addition, the MFEA considers a single representation to solve problems with different design variables such as satisfiability problem (SAT), traveling salesman problem (TSP) and numerical optimization problem, of which design variables are Boolean values, permutation, and floating points, respectively. This unified representation can be mapped to different design variables by variant mapping functions. In this study, we replace the unified representation in the original MFEA by the structure-based representation due to its effectiveness and efficiency in genetic fuzzy data mining. Therefore, the proposed structure-based representation MFEA does not need a mapping function for each task. More details about the proposed method are described below.

A. Representation

Traditionally, a membership function is represented by its parameters, which are put in genes to evolve in evolutionary algorithms. Without considering the structure of membership functions, such representation may generate inappropriate or even illegal membership functions after recombination. However, the structure is essential in the mining of fuzzy associations.

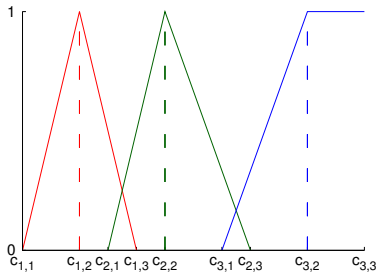


Figure 2: Example of three membership functions

Algorithm 2 Conversion into membership function

procedure CONVERT($\boldsymbol{x}$) ▷ $\boldsymbol{x} = (x_1, \ldots, x_{3\ell}, h)$
  $\boldsymbol{x}' \leftarrow \text{Sort}(\boldsymbol{x})$ ▷ Sort $x_1, \ldots, x_{3\ell}$
  $k \leftarrow 1$
  for $i \leftarrow 1$ **to** ℓ **do**
    for $j \leftarrow 1$ **to** 3 **do**
      $c_{i,j} \leftarrow x'_{ST(h,k)}$ ▷ $ST(h,k)$: k-th element of
      $k \leftarrow k + 1$ ▷ h-th structure
    end for
  end for
end procedure

In [24], Ting et al. proposed a structure based representation for genetic fuzzy data mining which takes both parameters and the structure of membership functions into account. That is, a chromosome consists of parameters and structure type. This study also considers the most commonly used triangular membership function as in [24]. There are 3ℓ real-encoded genes plus one gene for structure type for an item with ℓ linguistic terms. The structure type is encoded by an integer value which stands for the index of a structure. Every appropriate deployment of membership functions, recorded as a permutation, is labeled for indexing. Figure 3 illustrates the chromosome of the structure based representation. The first 3ℓ genes encode the parameters of ℓ membership functions and the last gene serves as the index of structure type. Algorithm 2 shows the transformation procedure of the structure based representation in detail, from chromosome to membership functions. The first 3ℓ real-valued genes are sorted and then deployed according to the permutation of the structure type $ST(h = 11) = (1, 2, 4, 3, 7, 5, 6, 8, 9)$. Consequently, the fourth lowest value is mapped to the third parameter, the fifth lowest value is mapped to the seventh parameter, and so forth.

Two concerns in the optimization of membership functions, i.e., the legality and suitability. The legality holds the shape and order of membership functions whereas the suitability keeps the expressiveness and interpretability of fuzzy regions.

a) Legality.: Suppose $c_{i,j}$ represents the j^{th} parameter of i^{th} triangular membership function, the legality is defined by the following two constraints:

$$c_{i,1} \leq c_{i,2} \leq c_{i,3}\,, \tag{6}$$

$$c_{1,2} \leq c_{2,2} \leq \cdots \leq c_{l,2}\,, \tag{7}$$

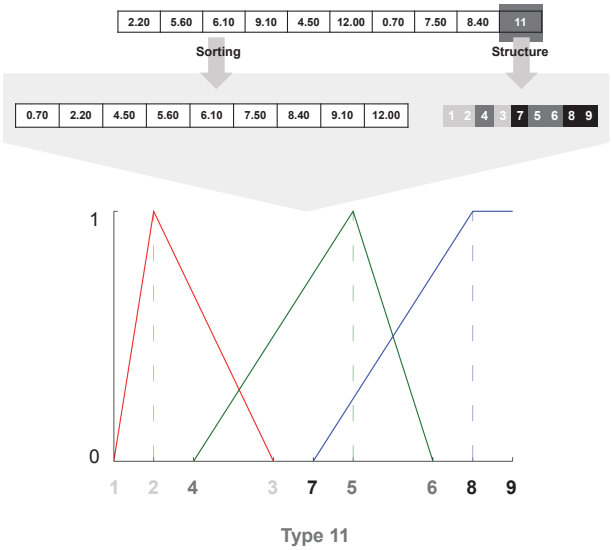


Figure 3: Example of chromosome representation

where inequality 6 and 7 retain the shape of triangle and the order of membership functions respectively. A proper setting of parameters which satisfies the two inequalities 6 and 7 can be found in Fig. 2. In traditional representation the variation operators may violate the above two constraints and illegal chromosomes should be fixed by rearranging the genes. Nonetheless, the structure based representation can avoid the illegality by the transformation procedure described in Algorithm. 2.

b) Suitability.: As aforementioned, suitability regulates the expressiveness and interpretability of fuzzy regions. According to the definition proposed by Hong et al. [13], the expressiveness and the interpretability are quantified by coverage and overlap, respectively. The coverage is defined by the covered area among all fuzzy regions while the overlap is evaluated by the overlapping area of two fuzzy regions. To ensure full coverage with proper overlap, Ting et al. [24] further proposed two constraints for coverage and overlap:

- Coverage:

$$c_{i-1,1} \leq c_{i,1} \leq c_{i-1,3} \tag{8a}$$

$$c_{i+1,1} \leq c_{i,3} \leq c_{i+1,3} \tag{8b}$$

- Overlap:

$$c_{i,3} \leq c_{i+2,1} \tag{9}$$

The use of structure representation can reduce the number of arrangements of membership functions and thereby reduce the searching space. Consider an item with three triangular membership functions, only 93 legal structure types over 9!=362880 permutations fulfill the legality constraints. Further taking the suitability constraints into account, there are 12 over 93 legal structure types, with full coverage and proper overlap.

B. Fitness Evaluation

Computing the fuzzy support of largest itemset is costly due to the exponential time complexity to the number of items in the dataset. Hong et al. [13] proposed an efficient evaluation based on the divide-and-conquer paradigm. For efficiency, the fuzzy support of large 1-itemset L_1 are evaluated rather than the largest itemset. This study also adopts such light evaluation of fuzzy support for efficiency.

The fitness function following the design in [13] is a nonlinear combination of fuzzy support and suitability. The computation of fuzzy support is described in Section II. The suitability is defined by two factors, to wit, overlap factor and coverage factor.

Definition 3 (Overlap Factor)**.** Given the membership functions after transformation procedure $\boldsymbol{c}$, the overlap factor is calculated by

$$\text{Overlap}\left(\boldsymbol{c}\right) = \sum_{i<j} \left(\max\left(\text{ovlratio}\left(R_i, R_j\right), 1\right) - 1\right) \quad (10)$$

with

$$\text{ovlratio}\left(R_i, R_j\right) = \frac{\text{The area covered by both } R_i \text{ and } R_j}{\min\left(c_{i,3} - c_{i,2}, c_{j,2} - c_{j,1}\right)}. \quad (11)$$

The ratio of overlap $\text{ovlratio}\left(R_i, R_j\right)$ computes the ratio of overlapped area over the smaller area between the right side of R_i and the left side of R_j with $i < j$. The ratio of overlap $\text{ovlratio}\left(R_i, R_j\right)$ is less than or equal to one for a moderate overlap, causing the best value of overlap factor being at 0.

Definition 4 (Coverage Factor)**.** Given the membership functions after transformation procedure $\boldsymbol{c}$, the coverage factor is calculated by

$$\text{Coverage}\left(\boldsymbol{c}\right) = \frac{\max\left(I\right)}{\text{range}\left(R_1, \ldots, R_l\right)}. \quad (12)$$

The coverage factor computes the ratio of range of item I over the range of all membership functions. A full coverage of membership functions has best coverage factor at 1.

Summing the overlap and coverage factors get the suitability. Then the fitness function of the membership functions $\boldsymbol{c}$ is defined as follows:

$$\text{Fitness}\left(\boldsymbol{c}\right) = \frac{\sum_{R \in L_1} \text{FuzzySupport}\left(R\right)}{\text{Overlap}\left(\boldsymbol{c}\right) + \text{Coverage}\left(\boldsymbol{c}\right)}. \quad (13)$$

The fitness function is the ratio of fuzzy support over suitability, aiming at maximizing the fuzzy support meanwhile minimizing the overlap and coverage factors. The consideration of large 1-itemset L_1 rather than the largest itemset abates the cost as mentioned above. The suitability forbids the endless pursuing of fuzzy support but regardless of shape of membership functions.

C. Scalar Fitness and Skill Factor

The MFEA handles different tasks by scalar fitness and skill factor. The scalar fitness is a rank based fitness. Given the ranking r_j^i of individual i on task j, the scalar fitness is defined as:

$$\varphi(i) = \frac{1}{\min\{r_1^i, ..., r_m^i\}},$$

where m denotes the number of tasks. Through the scalar fitness individuals for different tasks can be compared by a single scalar for survival. To know the best suitable tasks for the individuals, the skill factor of an individual records the task with best rank:

$$\tau(i) = \underset{j \in \{1,\ldots,m\}}{\arg\min}\ r_j^i.$$

The skill factor is the key to the genetic operators in MFEA, including the assortative mating, and fitness evaluation. The definitions of both scalar fitness and skill factor follow the definitions proposed in [11].

Algorithm 3 Assortative mating

if $(\tau_{prt_1} = \tau_{prt_2})$ **or** $(p < rmp)$ **then**
  $ofsp_1 \leftarrow \text{Crossover}(prt_1, prt_2)$
  $ofsp_2 \leftarrow \text{Crossover}(prt_1, prt_2)$
  $\tau_{ofsp_1} \leftarrow \text{Rand}(\tau_{prt_1}, \tau_{prt_2})$
  $\tau_{ofsp_2} \leftarrow \text{Rand}(\tau_{prt_1}, \tau_{prt_2})$
else
  $ofsp_1 \leftarrow \text{Mutation}(prt_1)$
  $ofsp_2 \leftarrow \text{Mutation}(prt_2)$
  $\tau_{ofsp_1} \leftarrow \tau_{prt_1}$
  $\tau_{ofsp_2} \leftarrow \tau_{prt_2}$
end if

D. Assortative Mating

Assortative mating generates offspring according to parents' skill factor. Algorithm 3 is the procedure of assortative mating. Given two randomly selected parents, the offspring can be sexual reproduction by both parents or asexual reproduction by one of the two parents, depending upon the parents' skill factor. For parents with the same skill factor or under the random mating probability (rmp) the sexual reproduction is taken by performing the crossover operator; otherwise, the offspring are generated by asexual reproduction, which mutates parents by mutation operator. The parameter rmp plays the role to balance the exploration of sexual reproduction and the exploitation of asexual reproduction. A suggested setting of rmp is 0.3 [11], which is also used in this study.

The sexual reproduction and asexual reproduction are performed by crossover and mutation operator, respectively. The crossover operator recombines the chromosome of the parents. This study adopts the uniform crossover for simplicity. The mutation operator alters the chromosome slightly. The creep mutation is adopted for perturbation of the parameters, while random resetting is used for reforming structure type.

Table I: Parameter setting

Parameter	GA_{93} and GA_{12}	$MFEA_{93}$ and $MFEA_{12}$
Representation	Parameter (real number) + Structure (integer)	
Parent selection	2-tournament	Random
Crossover	Uniform	Uniform
Crossover rate	0.8	-
rmp	-	0.3
Mutation	Creep ($\varepsilon = 3$)	Creep ($\varepsilon = 3$)
Mutation rate	0.01	-
Survival selection	$\mu + \lambda$	$\mu + \lambda$
Population size	50	128
#Evaluations	360000	360000

E. Evaluation and Survival Selection

After reproduction each offspring will be evaluated only on the task indicated by its skill factor. Hence, the number of evaluations used in each generation is equal to the population size. The skill factor of offspring is also determined by parents' skill factor. For each two offspring generated by sexual reproduction the skill factors are randomly chosen from their parents. On the contrary, each offspring obtained by asexual reproduction imitates the skill factor of its parent.

The survival selection decided the population of individuals for the next generation. The well-known $(\mu + \lambda)$ survival selection selects best μ individuals from the union of parents and offspring according to the fitness. Noteworthily, in MFEA the survival selection is based on the general scalar fitness rather than the fitness of each task (each item in this study).

IV. Experimental Results

This study validates the performance of the proposed structure-based representation MFEA through empirical studies. A series of experiments is conducted to investigate the effectiveness and efficiency of the proposed method. Six sizes of transactional data are experimented, composed of 10k, 30k, 50k, 70k, and 90k transactions, and there are 64 items in each dataset, forming 64 tasks for MFEA. The experiments consider four test algorithms: GA_{93} (GA using the structure-based representation), GA_{12} (GA_{93} refined by the suitability constraints), $MFEA_{93}$ and $MFEA_{12}$ (MFEA with structure-based representation and refined constraints.) The subscripts 93 and 12 stands for the numbers of structure types used. The parameter settings for the four algorithms are listed in table I, and the minimum support is set to 0.04. Most parameters follow the settings in [24] except the population size of MFEA, which is set to twice the number of tasks, i.e., $64 \times 2 = 128$. Every experiment takes 30 trials for significant analysis made by student's t-test with significant level $\alpha = 0.01$.

A. Effectiveness

First we show the effectiveness of the proposed $MFEA_{93}$ and $MFEA_{12}$, in comparison to GA_{93} and GA_{12} on datasets of five sizes from 10k to 90k. Figure 4 exhibits the progress of mean best fitness (MBF) in the course of evolution for the four test algorithms on datasets of 10k to 90k transactions. It is apparent that both $MFEA_{93}$ and $MFEA_{12}$ achieve better fitness and converge faster than GA_{93} and GA_{12} do. Comparing $MFEA_{93}$ and $MFEA_{12}$, there is no much difference between the two methods on datasets of size 10k, 70k and 90k in terms of MBF and convergence speed. For dataset of size 30k, $MFEA_{12}$ obtains better MBF than $MFEA_{93}$. This study also compares the obtained MBF and examines the statistical significance by t-test. Table II lists the MBF and p-values of the four test algorithms on datasets of 10k to 90k transactions. The $MFEA_{93}$ significantly excels GA_{93} on all datasets under the level of significance $\alpha = 0.01$. The $MFEA_{12}$ also outperforms GA_{12} on all five datasets. Nonetheless, there is no significant difference between $MFEA_{93}$ and $MFEA_{12}$. These fascinating outcomes point out the effectiveness of the proposed $MFEA_{93}$ and $MFEA_{12}$.

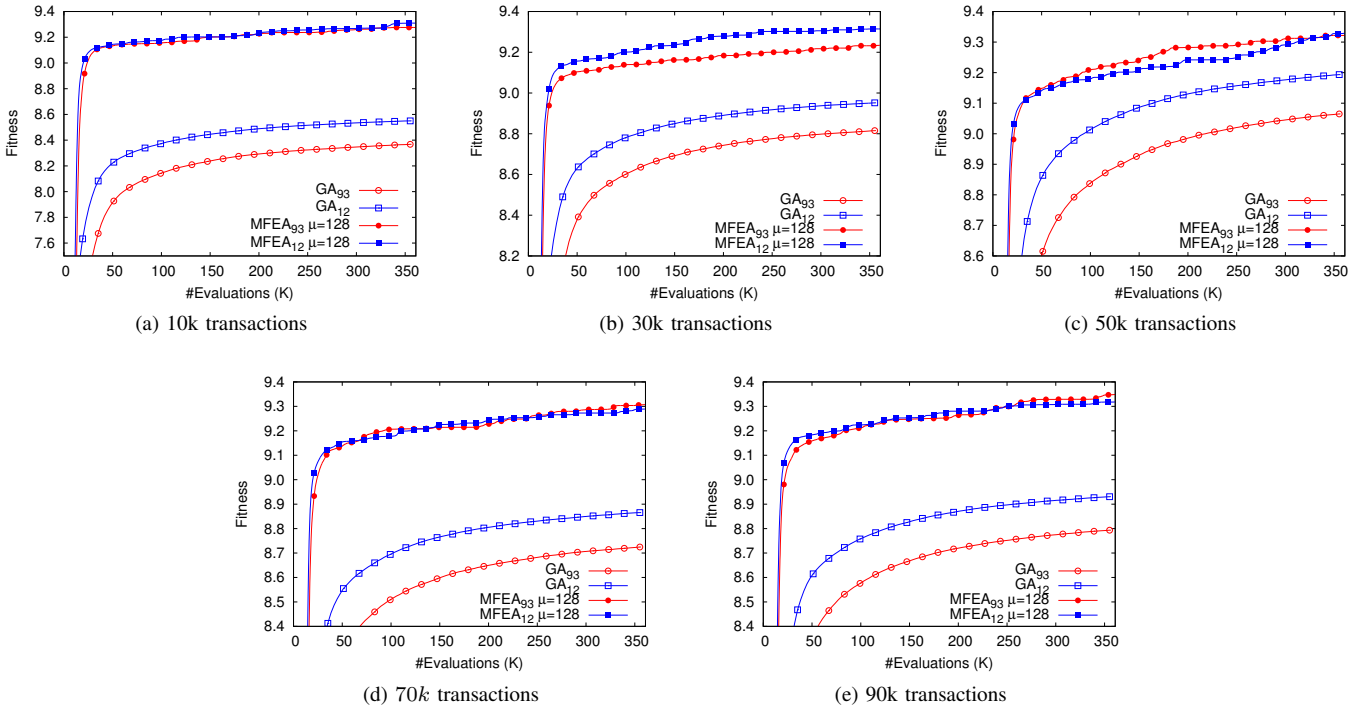
This study further makes comparison to the overlap, coverage, suitability, and fuzzy support for the GA_{93}, GA_{12}, $MFEA_{93}$, and $MFEA_{12}$. As shown in Table III the membership functions obtained from $MFEA_{12}$ achieve best overlap, coverage, and suitability on all datasets and acquire best fuzzy support on dataset of 30k transactions. On the contrary, the membership functions obtained from $MFEA_{93}$ get best fuzzy support on all datasets except the one with 30k transactions. The $MFEA_{93}$, and $MFEA_{12}$ both have better overlap, coverage, suitability, fuzzy support and fitness than GA_{93} and GA_{12} do aside from the dataset of 50k transactions. On the dataset having 50k transactions, GA_{93} performs best on fuzzy support but has poor performance in terms of suitability than $MFEA_{93}$, and $MFEA_{12}$; consequently, GA_{93} obtains ill performance of fitness.

B. Efficiency

This study also validates the efficiency of proposed $MFEA_{93}$ and $MFEA_{12}$. Table IV lists the number of evaluations and corresponding ratio and speedup used for $MFEA_{93}$ and $MFEA_{12}$ to exceed the finally obtained fitness of GA_{93} and GA_{12} after $360,000$ evaluations respectively on datasets of 10k to 90k transactions. For $MFEA_{93}$, only about 5% of overall evaluations used for GA_{93} is needed to achieve comparable fitness. The speedup of $MFEA_{93}$ over GA_{93} is 23.16 times in best case, and 13.71 times in worst case. In comparison to GA_{12}, the $MFEA_{12}$ also utilizes 4% to 5% of total number of evaluations except dataset of 50k transactions. The speedup of $MFEA_{12}$ over GA_{12} is 24.78 in best case and 20.04 in median case. In worst case $MFEA_{12}$ is still over three times faster than GA_{12}. These fruitful results express the efficiency of the proposed $MFEA_{93}$ and $MFEA_{12}$.

V. Conclusions

This study proposes a structure-based representation MFEA for mining fuzzy associations. The membership function of each item is treated as a single task, and all tasks are solved by MFEA at a time. The coalescence of MFEA and structure representation brings three advantages. First, the structure representation prevents the chromosome from illegality and knocks out membership functions with improper overlap and



(a) 10k transactions (b) 30k transactions (c) 50k transactions

(d) 70k transactions (e) 90k transactions

Figure 4: Anytime behavior against mean best fitness of the four test algorithms on datasets with 10k (a), 30k (b), 50k (c), 70k (d) and 90k (e) transactions

Table II: Mean best fitness and p-values from t-test statistical analysis for the GA_{93}, GA_{12},MFEA_{93} and MFEA_{12} on datasets of 10k to 90k transactions. The t-test compares the difference of two test algorithms A and B by p-value, where significant difference under level of significance $\alpha = 0.01$ is marked by boldface. The plus (+) and tilde (~) symbols stand for B is significantly better than or comparable to A, respectively.

	MBF				p-value		
#Tr.	GA_{93}	GA_{12}	MFEA_{93}	MFEA_{12}	GA_{93}:MFEA_{93}	GA_{12}:MFEA_{12}	MFEA_{93}:MFEA_{12}
10k	8.3678	8.5501	9.2760	**9.3098**	**9.68E-22** (+)	**1.59E-25** (+)	2.31E-01 (~)
30k	8.8152	8.9516	9.2340	**9.3143**	**1.05E-14** (+)	**1.17E-16** (+)	2.11E-02 (~)
50k	9.0649	9.1935	9.3216	**9.3293**	**7.40E-11** (+)	**1.55E-06** (+)	4.16E-01 (~)
70k	8.7247	8.8656	**9.3070**	9.2890	**5.28E-24** (+)	**1.43E-15** (+)	3.12E-01 (~)
90k	8.7935	8.9304	**9.3486**	9.3183	**4.57E-23** (+)	**8.35E-15** (+)	2.03E-01 (~)

coverage. Second, the MFEA improves the searching efficiency by evolving all tasks together. On the aspect of fitness landscapes, the moving direction, which is benefit to one task, from a given point in the design space may also improve the fitness of the other task. Third, the MFEA reduces a great amount of fitness evaluations in one generation since it exploits a single population; that is, $m\mu_{\mathrm{GA}} - \mu_{\mathrm{MFEA}}$ fitness evaluations are saved in one generation, where m is the number of items in the dataset. Considering the parameter setting used in this study that $\mu_{\mathrm{GA}} = 50$, $\mu_{\mathrm{MFEA}} = 128$, and $m = 64$, the structure-based MFEA takes only 4% of fitness evaluations of GA in one generation.

We examine the performance of the proposed MFEA in terms of effectiveness and efficiency through empirical analysis. The experimental results reveal that the structure-based representation MFEAs better the structure-based representation GAs significantly on datasets of five sizes (from 10k to 90k transactions). On the effect of efficiency, the results show that the structure-based representation MFEAs can exceed the best fitness obtained by structure-based representation GAs in about 5% of fitness evaluations, which means the structure-based representation MFEAs are about 20 times faster than structure-based representation GAs. These fruitful outcomes validate the effectiveness, efficiency, and scalability of the proposed method.

There are some possible future extensions. Memetic algorithms (MAs) have achieved great success in complex optimization problems. Incorporating local search to MFEA, forming the MFMA, can be a possible orientation. Moreover, this study considers MFEA as the evolutionary multitasking method. Adopting different evolutionary multitasking methods to genetic fuzzy data mining is another direction.

Table III: Behavior on different performance metrics, including overlap (Ovlp.), coverage (Cov.), suitability (Suit.), fuzzy support (Fzs.) and fitness for the four test algorithms on datasets of 10k to 90k transactions. The best result among the four algorithms is marked by boldface.

#Tr.	Algorithm	Ovlp.	Cov.	Suit.	Fzs.	Fitness
10k	GA_{93}	0.0505	1.0066	1.0571	8.8170	8.3678
	GA_{12}	0.0016	1.0056	1.0072	8.6117	8.5501
	$MFEA_{93}$	0.0177	1.0002	1.0179	**9.4317**	9.2760
	$MFEA_{12}$	**0.0000**	**1.0002**	**1.0002**	9.3113	**9.3098**
30k	GA_{93}	0.0363	1.0062	1.0424	9.1713	8.8152
	GA_{12}	0.0010	1.0054	1.0064	9.0088	8.9516
	MA_{93}	0.0062	1.0002	1.0064	9.2906	9.2340
	$MFEA_{12}$	**0.0000**	**1.0001**	**1.0001**	**9.3155**	**9.3143**
50k	GA_{93}	0.0361	1.0066	1.0427	9.4290	9.0649
	GA_{12}	0.0016	1.0052	1.0068	9.2561	9.1935
	$MFEA_{93}$	0.0086	1.0002	1.0088	**9.3992**	9.3216
	$MFEA_{12}$	**0.0000**	**1.0002**	**1.0002**	9.3315	**9.3293**
70k	GA_{93}	0.0385	1.0063	1.0449	9.1008	8.7247
	GA_{12}	0.0014	1.0052	1.0067	8.9246	8.8656
	$MFEA_{93}$	0.0102	1.0002	1.0103	**9.4008**	**9.3070**
	$MFEA_{12}$	**0.0000**	**1.0002**	**1.0002**	9.2909	9.2890
90k	GA_{93}	0.0410	1.0064	1.0475	9.1862	8.7935
	GA_{12}	0.0017	1.0054	1.0071	8.9936	8.9304
	$MFEA_{93}$	0.0016	1.0002	1.0018	**9.3653**	**9.3486**
	$MFEA_{12}$	**0.0001**	**1.0001**	**1.0003**	9.3207	9.3183

Table IV: The number of evaluations required for $MFEA_{93}$ and $MFEA_{12}$ to exceed the fitness of GA_{93} and GA_{12} respectively after 360000 evaluations, and the corresponding saving rates and speedups on datasets of 10k to 90k transactions

	$MFEA_{93}$			$MFEA_{12}$		
#Tr	#Eval	Rate(%)	Speedup	#Eval	Rate(%)	Speedup
10k	15547	4.32	23.16	14527	4.04	24.78
30k	18478	5.13	19.48	19115	5.31	18.83
50k	26251	7.29	13.71	116347	32.32	3.09
70k	17841	4.96	20.18	17076	4.74	21.08
90k	18096	5.03	19.89	17968	4.99	20.04

ACKNOWLEDGMENT

The authors would like to thank professor Chuan-Kang Ting, from department of power mechanical engineering at national Tsing Hua university in Taiwan, for giving comments about this study. This work was supported by the Ministry of Science and Technology of Taiwan, under contract MOST 109-2218-E-030-001-MY3.

REFERENCES

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Procddings of the International Conference on Very Large Data Bases*, pages 487–499, 1994.

[2] R. Chandra, A. Gupta, Y.-S. Ong, and C.-K. Goh. Evolutionary multitask learning for modular training of feedforward neural networks. In *Proceedings of International Conference on Neural Information Processing*, 2016.

[3] C.-C. Chang and W.-H. Au. Mining fuzzy association rules. In *Proceedings of the International Conference on Information and Knowledge Management*, pages 209–215, 1997.

[4] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(27):1–27, 2011.

[5] C.-H. Chen, A.-F. Li, and Y.-C. Lee. A fuzzy coherent rule mining algorithm. *Applied Soft Computing*, 13(7):3422–3428, 2013.

[6] C.-H. Chen, V.-S. Tseng, and T.-P. Hong. Cluster-based evaluation in fuzzy-genetic data mining. *IEEE Transactions on Fuzzy Systems*, 16(1):249–262, 2008.

[7] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. Advances in knowledge discovery and data mining. In *From Data Mining to Knowledge Discovery: An Overview*, pages 1–34. AAAI, 1996.

[8] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, 17:37–54, 1996.

[9] A. Gupta, J. Mańdziuk, and Y.-S. Ong. Evolutionary multitasking in bi-level optimization. *Complex & Intelligent Systems*, 1(1):83–95, 2015.

[10] A. Gupta, Y.-S. Ong, B. Da, L. Feng, and S. D. Handoko. Landscape synergy in evolutionary multitasking. In *Proceedings of IEEE Congress on Evolutionary Computation*, 2016.

[11] A. Gupta, Y.-S. Ong, and L. Feng. Multifactorial evolution: Toward evolutionary multitasking. *IEEE Transactions on Evolutionary Computation*, 20(3):343–357, 2016.

[12] A. Gupta, Y.-S. Ong, L. Feng, and K. C. Tan. Multiobjective multifactorial optimization in evolutionary multitasking. *IEEE Transactions on Cybernetics*, 2016.

[13] T.-P. Hong, C.-H. Chen, Y.-C. Lee, and Y.-L. Wu. Genetic-fuzzy data mining with divide-and-conquer strategy. *IEEE Transaction on Evolutionary Computation*, 12(2):252–265, 2008.

[14] 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*, 10(11):1091–1101, 2006.

[15] T.-P. Hong, C.-S. Kuo, and S.-C. Chi. Mining association rules from quantitative data. *Intelligent Data Analysis*, 3(5):363–376, 1999.

[16] T.-P. Hong and C.-Y. Lee. Induction of fuzzy rules and membership functions from training examples. *Fuzzy Sets and Systems*, 84(1):33–47, 1996.

[17] C.-M. Kuok, A. Fu, and M.-H. Wong. Mining fuzzy association rules in databases. *ACM SIGMOD Record*, 27(1):41–46, 1998.

[18] C. K.-H. Lee, K.-L. Choy, G. T.-S. Ho, and C. H.-Y. Lam. A slippery genetic algorithm-based process mining system for achieving better quality assurance in the garment industry. *Expert Systems with Applications*, 46:236–248, 2016.

[19] C. K.-H. Lee, G. T.-S. Ho, K.-L. Choy, and G. K.-H. Pang. A RFID-based recursive process mining system for quality assurance in the garment industry. *International Journal of Production Research*, 52(14):4216–4238, 2014.

[20] G. Piatesky-Shapiro, R. Brachman, W. Klösgen, and E. Simoudis. An overview of issues in developing industrial data mining and knowledge discovery applications. In *Proceedings of Knowledge Discovering and Data Mining*, pages 89–95, 1996.

[21] R. Sagarna and Y.-S. Ong. Concurrently searching branches in software tests generation through multitask evolution. In *Proceedings of IEEE Symposium Series on Computational Intelligence*, 2016.

[22] R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. *ACM SIGMOD Record*, 25(2):1–12, 1996.

[23] S. Strasser, J. Sheppard, N. Fortier, and R. Goodman. Factored evolutionary algorithms. *IEEE Transactions On Evolutionary Computation*, 2016.

[24] C.-K. Ting, T.-C. Wang, R.-T. Liaw, and T.-P. Hong. Genetic algorithm with a structure-based representation for genetic-fuzzy data mining. *Soft Computing*, pages 1–12, 2016.

[25] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained K-means clustering with background knowledge. In *Proceedings of the International Conference on Machine Learning*, pages 577–584, 2001.

[26] L. Zhou, L. Feng, J. Zhong, Y.-S. Ong, Z. Zhu, and E. Sha. Evolutionary multitasking in combinatorial search spaces: A case study in capacitated vehicle routing problem. In *Proceedings of IEEE Symposium Series on Computational Intelligence*, 2016.