**Noname manuscript No.**
(will be inserted by the editor)

# Mining Fuzzy Association Rules Using a Memetic Algorithm Based on Structure Representation

**Chuan-Kang Ting · Rung-Tzuo Liaw · Ting-Chen Wang · Tzung-Pei Hong**

the date of receipt and acceptance should be inserted later

**Abstract** The association rules render the relationship among items and have become an important target of data mining. The fuzzy association rules introduce fuzzy set theory to deal with the quantity of items in the association rules. The membership functions play a key role in the fuzzification process and, therefore, significantly affect the results of fuzzy association rule mining. This study proposes a memetic algorithm (MA) for optimizing the membership functions in fuzzy association rule mining. The MA adopts a novel chromosome representation that considers the structures of membership functions. Based on the structure representation, we develop a local search operator to improve the efficiency of the MA in exploring good membership functions. Two local search strategies for the MA are further investigated. This study conducts a series of experiments to examine the proposed MA on different amounts of transactions. The experimental results show that the MA outperforms state-of-the-art evolutionary algorithms in terms of solution quality and convergence speed. These preferable results show the advantages of the structure-based representation and the local search in improving the performance. They also validate the high capability of the proposed MA in mining fuzzy association rules.

**Keywords** Memetic computing · Genetic algorithm · Structure representation · Local search · Fuzzy association rules · Membership function

## 1 Introduction

Data mining explores the information hidden in the data and transforms it into explicit valuable knowledge [10]. Several data mining technologies have been proposed to discover knowledge for different purposes, such as classification, clustering, and association rules [9]. Mining association rules is an important data mining technology aiming to find the relationship among items, also known as a homogeneous group or an affinity group, from the database and has achieved many successful stories, e.g., the prediction of customers' behavior in Walmart.

The Apriori algorithm is well-known for mining association rules [1]. This method establishes association rules based on the frequent itemsets according to a user-defined minimum confidence. Srikant and Agrawal [31] extended the rules into quantitative association rules, which can tackle the data of quantitative values or categories. They presented a method similar to the Apriori algorithm but requiring an additional preprocess of data discretization. The fuzzy set theory is introduced to different aspects of association rules [6, 17, 19]. In particular, Hong et al. [15, 16] proposed the fuzzy transaction data mining algorithm (FTDA) by adopting fuzzy sets to analogue the values of original data. In the FTDA, quantitative values of data are transformed into fuzzy values according to the membership functions. The results are known as the fuzzy association rules. The FTDA holds the advantages in suitability, tolerance, and extension for nonlinear systems.

C.-K. Ting · R.-T. Liaw · T.-C. Wang
Department of Computer Science and Information Engineering
National Chung Cheng University, Taiwan
E-mail: {ckting, lrt101p, wtc102p}@cs.ccu.edu.tw

T.-P. Hong
Department of Computer Science and Information Engineering
National University of Kaohsiung, Taiwan
Department of Computer Science and Engineering
National Sun Yat-sen University, Taiwan
E-mail: tphong@nuk.edu.tw

The membership functions establish the mapping from a quantity to a fuzzy value and thus exert a significant influence on fuzzy association rule mining. Recently, some studies adopt genetic algorithm (GA) to optimize the membership functions and receive promising results. However, two important issues remain open in the genetic-fuzzy association mining. First, the current variants of GA ordinarily use integers or real values as the chromosome representation for membership functions. The structure of membership functions plays a crucial role but is not considered in the chromosome representation. Second, GA is well known for good global search but performs poorly at local search. This weakness makes it harder for GA to find the best membership functions for fuzzy data mining, especially in a large number of transactions.

This paper proposes a memetic algorithm (MA) to address the above issues. Beyond *Darwinian* genetics, MA considers the *Lamarckian* or *Baldwinian* effect by integrating local search into evolutionary algorithms. This integration has been widely shown as to be effective in improving the performance of evolutionary algorithms [12,18,26,33]. The proposed MA considers the structure of membership functions in the chromosome representation. The novel representation facilitates local search in the various structures of membership functions. This study accordingly designs a local search operator and incorporates it into a GA as an MA for enhancement of its search capability, which is helpful for the genetic-fuzzy system to mine association rules among a large amount of transactions. In addition, two strategies for the local search are investigated. A series of experiments, including 10,000 to 1,000,000 transactions, is conducted to examine the performance of the proposed MA in different data scales.

The remainder of this paper is organized as follows. Section 2 introduces fuzzy association rule mining. Section 3 sheds light on the proposed MA. Section 4 presents and discusses the experimental results. Finally, we draw conclusions and recommend the directions for future work in Section 5.

## 2 Mining Fuzzy Association Rules

The association rules reveal the coincidence of items among transactions. This information is important and commonly used in the analysis of customers' behavior and business intelligence. An association rule has a general form:

$$X \rightarrow Y, \tag{1}$$

where $X$ and $Y$ are sets of items. For example, the association rule $\{bread, cheese\} \rightarrow \{milk\}$ mined from the transactions in Walmart infers "If buying *bread* and *cheese*, then buying *milk*".

Given an itemset $I = \{I_1, \ldots, I_m\}$ with $m$ items and a database $D = \{T_1, \ldots, T_n\}$ with $n$ transactions $T_k \subseteq I$, an association rule $X \rightarrow Y$ represents the possibility: if $X \subseteq T_i$ then $Y \subseteq T_i$. Two metrics, namely support and confidence, are adopted to measure an association rule.

**Definition 1 (Support)** The support of association rule $X \rightarrow Y$ is defined by the probability that $X$ and $Y$ coexist, i.e.,

$$\text{Support}\,(X \rightarrow Y) = \mathcal{P}\,(X \cup Y)\,. \tag{2}$$

**Definition 2 (Confidence)** The confidence of association rule $X \rightarrow Y$ is defined by the probability that $Y$ exists given $X$ exists, i.e.,

$$\text{Confidence}\,(X \rightarrow Y) = \mathcal{P}\,(Y|X) = \frac{\mathcal{P}\,(X \cup Y)}{\mathcal{P}\,(X)}. \tag{3}$$

Agrawal and Srikant [1] proposed the Apriori algorithm to mine the association rules. Given a minimum support $\theta_{\text{supp}}$, the algorithm selects large itemsets $L = \{L_1, \ldots, L_p\}$ from the candidate itemsets $C = \{C_1 \ldots, C_p\}$ progressively. The large itemsets obtained are used to generate the candidate association rules. An associate rule is valid if its confidence is greater than or equal to a predetermined minimum confidence $\theta_{\text{conf}}$.

The original association rules consider the existence of items but ignore their quantity. To address this issue, Hong et al. [16] introduced the notion of fuzzy sets to association rules, thereby considering the quantity of items. The fuzzy association rules involve the fuzzy support and fuzzy confidence based on the membership functions.

Given a database $D$ with $n$ transactions among $m$ items. Let $R_{j,k}$ denote the fuzzy region of the $k$-th membership function for item $I_j$. The fuzzy membership value $f_{j,k}^{(i)}$ of region $R_{j,k}$ is determined by the quantity $v_j^{(i)}$ of the $j$-th item in the $i$-th transaction $T_i$.

**Definition 3** The fuzzy support of region $R_{j,k}$ is calculated by

$$\text{FuzzySupport}\,(R_{j,k}) = \frac{1}{n}\sum_{i=1}^{n} f_{j,k}^{(i)}. \tag{4}$$

If a fuzzy region $R_{j,k}$ has $\text{FuzzySupport}\,(R_{j,k}) \geq \theta_{\text{supp}}$, it is included to the large 1-itemset $L_1$ like the Apriori algorithm.

For a set of fuzzy regions $R = \{R_1, \ldots, R_p\}$, its fuzzy value in transaction $T_i$ is computed by the intersection of the membership values $f_{R_k}^{(i)}$, i.e.,

$$f_R^{(i)} = \bigcap_{k=1}^{p} f_{R_k}^{(i)}. \tag{5}$$

The minimum function is widely used as the intersection operator in fuzzy systems. The fuzzy value of $R$ is therefore

$$f_R^{(i)} = \min_{1 \leq k \leq p} f_{R_k}^{(i)}. \tag{6}$$

Based on the above equation, the fuzzy support of $R$ is defined as follows.

**Definition 4 (Fuzzy Support)** The fuzzy support of $R$ is defined by

$$\text{FuzzySupport}\,(R) = \frac{1}{n} \sum_{i=1}^{n} f_R^{(i)}. \tag{7}$$

If the fuzzy support of $R$ is greater than the minimum support $\theta_{\text{supp}}$, $R$ is added to the large $p$-itemset $L_p$. The collection of large itemsets continues until $L_p = \emptyset$.

The fuzzy association rules are built upon the large itemsets and their fuzzy confidence. Given a large $p$-itemset $L_p = \{R_1, \ldots, R_p\}$, a candidate rule has the form:

$$X \rightarrow Y, \quad X, Y \subset L_p \text{ and } X \cap Y = \emptyset, \tag{8}$$

where $X$ and $Y$ are two disjoint subsets of $L_p$, which represent the antecedent and consequent, respectively. The candidate rules need to be further examined on their confidence values.

**Definition 5 (Fuzzy Confidence)** The fuzzy confidence of a candidate rule $R : X \rightarrow Y$ associated with $L_p$ is defined by

$$\text{FuzzyConfidence}\,(R) = \frac{\text{FuzzySupport}(X \cup Y)}{\text{FuzzySupport}(X)}. \tag{9}$$

If a candidate rule has fuzzy confidence greater than, or equal to, the minimum confidence $\theta_{\text{conf}}$, then it is qualified as a fuzzy association rule. All the candidate rules generated from the large itemset $L_p$ must be examined.

The membership functions determine the mapping of fuzzy values, and thus, exert a significant influence on the fuzzy association rules. The common membership functions include the triangular, trapezoidal, Gaussian, and bell functions. Each membership function has certain parameters to tune concerning the fuzziness. For example, in the most widely used type of membership function, namely the triangular function, there exist three parameters (vertices) to be set. The optimization of membership functions, therefore, becomes a key task in fuzzy association rule mining. To address this task, several studies utilize evolutionary algorithms in view of their high capability in optimization. Hong et al. [16] developed GAs for fuzzy data mining and showed that GAs can gain the appropriate membership functions for the fuzzy association rules. They further proposed using a divide-and-conquer strategy to increase the efficiency of genetic-fuzzy data mining [15]. Chen et al. [8] devised a cluster-based method to reduce the computational cost at evaluation. The proposed method divides the population into several clusters and adopts the support value of a representative chromosome in each cluster for evaluation. Alcalá-Fdez et al. [2] used GA to select the fuzzy association rules rather than optimize the membership functions. Lee et al. [20] proposed using GA to directly evolve a population of fuzzy rules instead of membership functions in mining of the process parameters and product quality. Moreover, Chen et al. [7] presented a fuzzy coherent rule mining algorithm to deal with the issue of setting the minimum support. In addition to GA, particle swarm optimization (PSO) and other evolutionary algorithms are adopted to optimize the parameters of membership functions. Cai et al. [5] devised a nonlinear PSO method for mining fuzzy association rules, where a particle represents all the parameters of membership functions. Mishra et al. [24] applied PSO to mine gene expression data for fuzzy frequent patterns. The PSO generates the initial population using the frequent pattern growth method to improve its performance.

Beyond single objective, multi-objective optimization is also considered in fuzzy association rule mining [11]. These studies take more than one objective into account at the optimization of membership functions. Qodmanan et al. [28] considered both the support and confidence, and Meng and Pei [22] included a linguistic quantifier and truth in the fitness function. Minaei-Bidgoli et al. [23] considered more objectives, including support, confidence, comprehensibility, and interestingness. They used the Michigan approach for the multi-objective evolutionary algorithm, where each chromosome represents a single association rule. By contrast, Rudziński [29] adopted the Pittsburgh approach for representation of the association rules considering two objectives, i.e., the root mean square error and interpretability. Antonelli et al. [3] presented a multi-objective evolutionary learning scheme to simultaneously optimize the association rules and the membership function parameters.

Memetic algorithms have achieved considerable success in data mining, such as classification [30], clustering [27], feature selection [25], and data processing [21]. Some research applies MA to fuzzy data mining. Gál et al. [13] proposed an MA based on bacterial evolutionary algorithm (BEA) for extracting the fuzzy rules of trapezoidal membership functions. The local search operator relies on bacterial mutation and the Levenberg-Marquardt method. Balázs and Kóczy [4] devised a bac-

**Algorithm 1** Memetic Algorithm

```
$P \leftarrow$ Initialize()
Evaluate(all $\boldsymbol{x} \in P$)
while not terminated do
    $P' \leftarrow \emptyset$
    repeat
        $\boldsymbol{x}_1, \boldsymbol{x}_2 \leftarrow$ Select($P$)
        $\boldsymbol{y}_1, \boldsymbol{y}_2 \leftarrow$ Crossover($\boldsymbol{x}_1, \boldsymbol{x}_2$)
        $\boldsymbol{y}_i \leftarrow$ Mutate($\boldsymbol{y}_i$), $i \in \{1, 2\}$
        Evaluate($\boldsymbol{y}_i$), $i \in \{1, 2\}$
        $P' \leftarrow P' \cup \{\boldsymbol{y}_1, \boldsymbol{y}_2\}$
    until $P'$ is full
    $P' \leftarrow$ LocalSearch($P'$)
    $P \leftarrow$ Survive($P, P'$)
end while
```

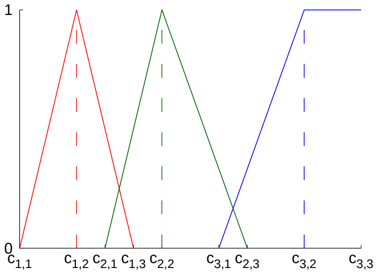


Fig. 1: Example of three membership functions

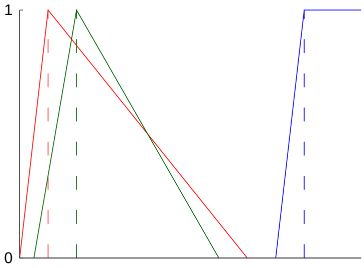


Fig. 2: Example of inappropriate membership functions

terial memetic programming for building a hierarchical-interpolative fuzzy system. The bacterial memetic programming combines bacterial genetic programming with a gradient based local search operator. Ho and Garibaldi [14] mined Takagi-Sugeno-Kang (TSK) fuzzy models by a memetic parallel GA, which integrates a parallel GA for the premise of fuzzy rules with a QR Householder least square method for the consequence of fuzzy rules. The premises and consequences of fuzzy rules are evolved by different search mechanisms. Tsakonas [32] considered memetic genetic programming (GP) for constructing a Takagi-Sugeno fuzzy system. The MA hybridizes GP with the least square and the gradient descent methods. The above studies show that local search can improve the convergence of evolutionary algorithms. Therefore, this paper considers an MA based on GA with local search on the structure types for improving the convergence speed and solution quality in fuzzy association rules mining.

## 3 The Proposed Memetic Algorithm

This study designs a memetic algorithm (MA) to optimize the membership functions. The proposed MA is based on the evolutionary scheme of GA and a local search operator. The MA adopts a novel chromosome representation considering the structures, in addition to the parameters, of the membership functions. The genetic operators are further developed to adapt to the new representation. Moreover, we devise a local search operator that utilizes the structure information for the MA. Algorithm 1 presents the framework of the proposed MA. More details are described below.

### 3.1 Representation

A membership function is composed of parameters to be optimized in fuzzy association rule mining. These parameters are encoded as genes of a chromosome in evolutionary algorithms. The structure of membership functions, however, is essential yet neglected in the conventional chromosome representation.

This study proposes a novel chromosome representation considering the parameters plus the structure of membership functions. Restated, a chromosome comprises two parts: 1) parameters, and 2) structure type. Given an item using $\ell$ linguistic terms to describe its quantity, the first part of a chromosome is represented by $3\ell$ real-valued genes to determine the parameters of $\ell$ membership functions. This part is conventionally used for chromosome representation. In addition, this study introduces the structure information to chromosomes. The structure type is indexed according to the deployment of membership functions. For example, the first $3\ell$ real-valued genes in Figure 3 represent the parameters of $\ell$ membership functions, and the last integer-valued gene indicates the structure type. When forming the membership functions, the $3\ell$ parameters are first sorted and then arranged according to the structure index. Algorithm 2 presents the pseudocode of converting a chromosome into membership functions. As illustrated in Fig. 3, according to the structure type $ST(\tau = 11) = (1, 2, 4, 3, 7, 5, 6, 8, 9)$, the lowest value 0.7 corresponds to the first parameter, the third lowest value 4.5 corresponds to the fourth parameter, and so forth.

Legality and suitability are two key factors needed to be considered in the parameter setting of member-

**Algorithm 2** Conversion

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

ship functions. The former regulates that the parameters should comply with the form of membership function, while the latter is concerned with the expressiveness and interpretability of fuzzy sets.

*Legality.* Given a triangular membership function[1] and let $c_{i,j}$ denote the $j$-th parameter of $i$-th membership function for a given item. A chromosome must satisfy the following two constraints:

$$c_{i,1} \leq c_{i,2} \leq c_{i,3} \tag{10}$$

$$c_{1,2} \leq c_{2,2} \leq \cdots \leq c_{l,2} \tag{11}$$

The first constraint maintains the triangular shape and the second constraint regulates the order of linguistic terms. Figure 1 illustrates a legal membership function that satisfies the above constraints.

*Suitability.* In addition to legality, the suitability must be considered in the optimization of membership functions. Hong et al. [15] defined the suitability by coverage and overlap of membership functions. The coverage represents the range covered by all the membership functions for an item; the overlap is measured by the area covered by two membership functions. Figure 2 illustrates inappropriate membership functions, where the overlap of the left two membership functions is too high, and the coverage is incomplete, due to the gap between the right two membership functions.

The following inequalities secure the full coverage and moderate overlap:

- Coverage:

$$c_{i-1,1} \leq c_{i,1} \leq c_{i-1,3} \tag{12a}$$

$$c_{i+1,1} \leq c_{i,3} \leq c_{i+1,3} \tag{12b}$$

- Overlap:

$$c_{i,3} \leq c_{i+2,1} \tag{13}$$

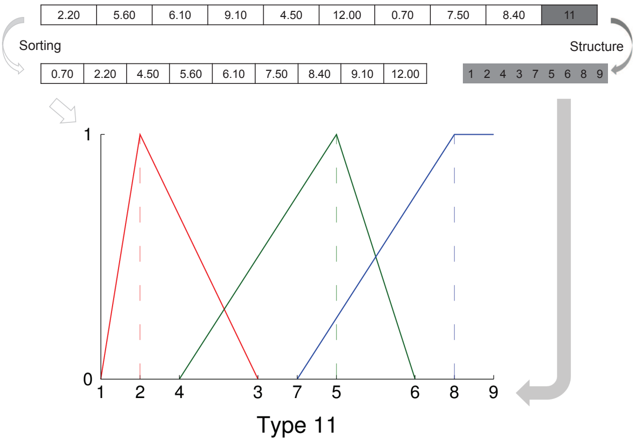


Fig. 3: Example of chromosome representation

The proposed representation, consisting of parameters and structure, can help refine the chromosomes for the appropriate membership functions. For fuzzy association rules using three triangular membership functions, the nine vertices account for $9! = 362880$ permutations. Among them, the proposed chromosome representation considers only 93 legal structure types that satisfy constraints (10) and (11) on shape and order, which substantially filters out the illegal membership functions and reduces the search space. Furthermore, in the light of suitability, 12 structures types are selected as per (12) and (13). These 12 structure types have an appropriate arrangement of membership functions, i.e., full coverage and moderate overlap.

### 3.2 Fitness Evaluation

In mining fuzzy association rules, the fuzzy support of the largest itemset is usually used to evaluate the fitness of chromosomes. This evaluation is time-consuming due to the time complexity $O(2^m)$ for garnering the largest itemset. To address this issue, Hong et al. [15] presented the divide-and-conquer strategy for fitness evaluation in fuzzy data mining. More specifically, the fitness function adopts the fuzzy support of the large 1-itemset $L_1$ in place of the largest itemset to decrease the computational cost. The fitness function further considers the suitability based on the coverage and overlap of membership functions. In light of these advantages, we use this fitness function for the MA.

The fitness function evaluates a chromosome according to the fuzzy support and suitability. The fuzzy support is calculated using (7); the suitability is composed of the coverage and overlap of the membership functions defined in the chromosome.

---

[1] This study uses triangular membership functions; however, other shapes such as trapezoidal and bell functions are also applicable.

**Definition 6 (Overlap Factor)** The overlap factor of a chromosome $\boldsymbol{x}$ is defined by

$$\text{Overlap}\left(\boldsymbol{x}\right)=\sum_{i<j}\left(\max\left(\text{ovlratio}\left(R_i,R_j\right),1\right)-1\right) \tag{14}$$

with

$$\text{ovlratio}\left(R_i,R_j\right)=\frac{\text{The area covered by both } R_i \text{ and } R_j}{\min\left(c_{i,3}-c_{i,2},c_{j,2}-c_{j,1}\right)}. \tag{15}$$

The ratio of overlap $\text{ovlratio}\left(R_i,R_j\right)$ accounts for the proportion of the area covered by both the membership functions to the smaller of $(c_{i,3}-c_{i,2})$ and $(c_{j,2}-c_{j,1})$, which are the right half of the left membership function and the left half of the right membership function, respectively. The overlap factor is nonnegative with a best value of 0. Note that the zero overlap factor implies a moderate overlap, rather than non-overlap, of all pairs of the membership functions.

**Definition 7 (Coverage Factor)** The coverage factor of a chromosome $\boldsymbol{x}$ is defined by

$$\text{Coverage}\left(\boldsymbol{x}\right)=\frac{\max\left(I\right)}{\text{range}\left(R_1,\ldots,R_l\right)}. \tag{16}$$

The coverage factor is in inverse proportion to the range covered by all membership functions. The factor has a minimum (best) value of 1, indicating full coverage of the item's quantity.

The suitability is defined as the sum of the overlap and coverage factors. Accordingly, the fitness value of a chromosome $\boldsymbol{x}$ is computed by

$$f\left(\boldsymbol{x}\right)=\frac{\sum_{R\in L_1}\text{FuzzySupport}\left(R\right)}{\text{Overlap}\left(\boldsymbol{x}\right)+\text{Coverage}\left(\boldsymbol{x}\right)}. \tag{17}$$

The fitness function considers large 1-itemset $L_1$ obtained from the membership functions converted from $\boldsymbol{x}$. The use of $L_1$ instead of the largest itemset reduces the computational cost at the calculation of fuzzy support. In addition, the suitability prevents the MA from pursuing the membership functions with high fuzzy support but serious overlap or narrow coverage.

### 3.3 Genetic Operators

The proposed MA is based on the scheme of GA, in which the genetic operators include parent selection, crossover, mutation, and survival selection. The parent selection operator selects two chromosomes out of the population to serve as the parents for subsequent reproduction. The probability for a chromosome to be selected is ordinarily in proportion to its fitness. For example, the $k$-tournament selection chooses the best of the $k$ chromosomes as a parent.

The reproduction process includes two operators, i.e., crossover and mutation. The crossover operator generates the offspring by recombining the genetic information of the parents. This study adopts the uniform crossover that is widely used for GA. The uniform crossover assigns each gene randomly from either parent to produce offspring. Next, the MA performs mutation for small changes on the offspring. Here we use the creep mutation and random resetting to randomly alter the parameters and structure type, respectively.

Noteworthily, the crossover and mutation operations may result in illegal offspring violating constraints (10) and (11). In the conventional chromosome representation without structure information, these illegal offspring need to be fixed by rearranging the sequence of genes. By contrast, the proposed structure-based representation is capable of preventing illegal offspring, in that the shape and order have been considered and stipulated in the structure type.

The survival selection implements the principle "survival of the fittest" to pick the chromosomes and form the population for the next generation. In the well-known $(\mu+\lambda)$ survival selection, parents and offspring all compete for survival.

### 3.4 Local Search

This study proposes a local search operator to improve the search capability of the MA. The local search acts upon the structure types in that the chromosome representation facilitates exploring the structures of membership functions. As Algorithm 3 shows, the local search operator scans the neighborhood of a given chromosome for a better solution, where the neighbors are defined as the chromosomes having the same parameters but different structure types. For example, there are 92 neighbors for a chromosome using three triangular membership functions. Compared to the neighborhood defined in the parameter domain, the neighborhood based on

**Algorithm 3** Local Search

```
procedure LocalSearch($\boldsymbol{x}, h$)    ▷ chromosome $\boldsymbol{x}$, size $h$
    for $i \leftarrow 1$ to $h$ do
        $\boldsymbol{y}_i \leftarrow (x_1,\ldots,x_{3\ell},\text{rand}(|ST|))$    ▷ randomize type
    end for
    $\boldsymbol{y}^* \leftarrow \text{BestOf}(\boldsymbol{y}_{i\in\{1,\ldots,h\}})$
    if $f(\boldsymbol{y}^*) > f(\boldsymbol{x})$ then
        $\boldsymbol{x} \leftarrow \boldsymbol{y}^*$
    end if
end procedure
```

structures has a small size and can enhance the efficiency of local search.

Two strategies are developed for choosing the target chromosomes of local search: *best* and *random*. The former chooses the best $k$ chromosomes, whereas the latter randomly picks $k$ chromosomes from the population. Additionally, we limit the local search to check $h$ neighbors to control the number of extra fitness evaluations. Hence, the local search requires $kh$ additional fitness evaluations at each generation.

Table 1: Parameter setting

| Parameter | Value |
|---|---|
| Representation | Parameter (real number) +<br>Structure (integer) |
| Parent selection | 2-tournament |
| Crossover | Uniform |
| Crossover rate | $p_c = 0.8$ |
| Mutation | Creep ($\varepsilon = 3$) |
| Mutation rate | $p_m = 0.01$ |
| Survival selection | $\mu + \lambda$ |
| Population size | 50 |
| #Evaluations | 25000 |

## 4 Experimental Results

This study carries out a series of experiments to examine the performance of the proposed MA on the optimization of membership functions for mining fuzzy association rules. In addition, we investigate the influences of the proposed chromosome representation and the local search operator upon the fitness as well as its components, to wit, overlap, coverage, and fuzzy support. The experiments include five algorithms: GA [15], $\text{GA}_{93}$ (GA using the novel representation), $\text{GA}_{12}$ ($\text{GA}_{93}$ refined by the suitability constraints), $\text{MA}_{93}$ and $\text{MA}_{12}$ ($\text{GA}_{93}$ and $\text{GA}_{12}$ applying the local search operator, respectively). The subscripts 93 and 12 account for the numbers of structure types encoded in the chromosome representation. Table 1 summarizes the parameter setting for the five test algorithms. The minimum support is set to 0.04. Different data sizes are tested in the experiments, including 10, 30, 50, 70, and 90k transactions, each consisting of 64 items [15]. Moreover, we examine the performance of test algorithms on a dataset with one million transactions (1000k) to simulate big data. Each experiment includes 30 independent runs of each algorithm.

### 4.1 Local Search Strategy and Size

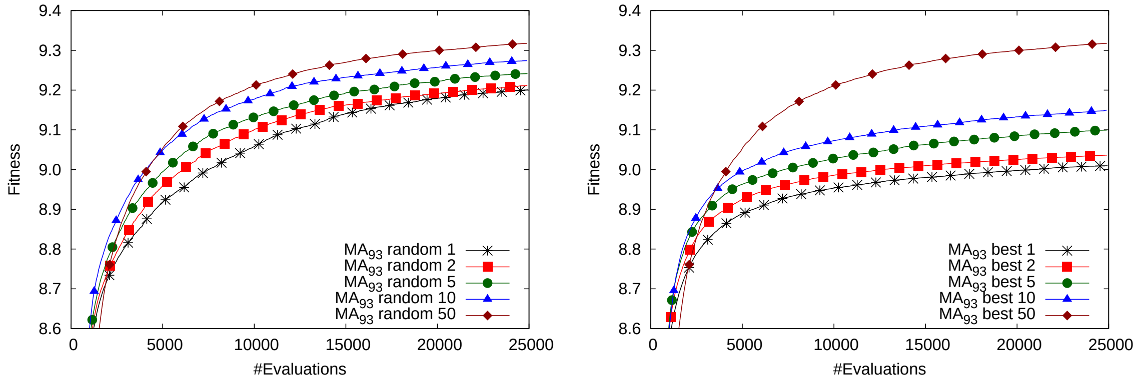
First, we investigate the effects of the best and random strategies on the MA. Figure 4 plots the progress of the mean best fitness (MBF) for $\text{MA}_{93}$ and $\text{MA}_{12}$ using the best and random strategies performed on different number of chromosomes at each generation. Due to space limitation, only the experimental results on the 90k transactions are presented. The figures show that $\text{MA}_{93}$ benefits from the increase of individuals for both the best and random strategies. In particular, $\text{MA}_{93}$ using either the best or random strategy on the whole population, i.e., 50 chromosomes, outperforms that with other sizes in terms of fitness. On the other hand, the enhancement on $\text{MA}_{12}$ depends upon the strategy adopted for the local search: The increase of individuals leads to better fitness for the best strategy, but worse for the random strategy. Furthermore, we look into the influences over the overlap, coverage, suitability, and fuzzy support of $\text{MA}_{93}$ and $\text{MA}_{12}$. According to Table 2, $\text{MA}_{12}$ gains better overlap, coverage, and therefore better suitability than $\text{MA}_{93}$ does. Nonetheless, the constraints on the structure types of $\text{MA}_{12}$ hinder its pursuit of fuzzy support and result in the fitness inferior to $\text{MA}_{93}$.

Second, we compare the performance of MA using different local search sizes. The results in Fig. 5 show that $\text{MA}_{93}$ and $\text{MA}_{12}$ with $h = 1$ perform best and their solution quality and convergence speed deteriorate as the local search size increases. Table 3 lists the overlap, coverage, suitability, fuzzy support, and fitness values of the membership functions obtained from $\text{MA}_{93}$ and $\text{MA}_{12}$ with different sizes of $h$. The performance trends with respect to $h$ are similar among the two datasets and two MAs: The increase in size $h$ worsens the overlap, coverage, suitability, fuzzy support, and fitness, suggesting the advantage of using $h = 1$, i.e., local search checking only one neighbor.

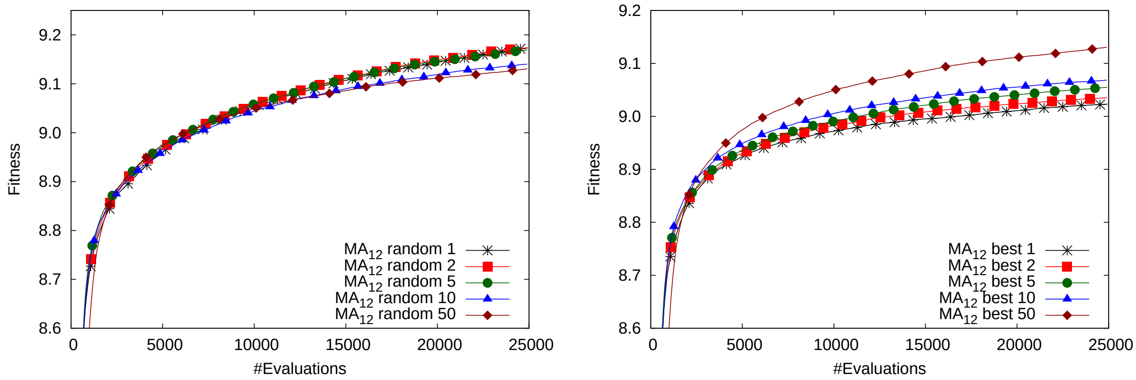
In view of the generally good performance, the following experiments use the strategy of local search on all chromosomes ($k = 50$) with only one neighbor ($h = 1$) for $\text{MA}_{93}$ and $\text{MA}_{12}$.

### 4.2 Performance Comparison

This study examines the performances of $\text{MA}_{93}$ and $\text{MA}_{12}$ in comparison to GA, $\text{GA}_{93}$, and $\text{GA}_{12}$. First, we investigate the performances on the medium size of datasets (10–90k transactions). As Figure 6 shows, GA using the structure-based representation, i.e., $\text{GA}_{93}$ and $\text{GA}_{12}$, achieves higher fitness than the original GA does, which verifies the advantages of using structure types in the representation. The results further indicate the effectiveness of the proposed local search operator on improving the solution quality: $\text{MA}_{93}$ and



(a) $MA_{93}$-random (b) $MA_{93}$-best

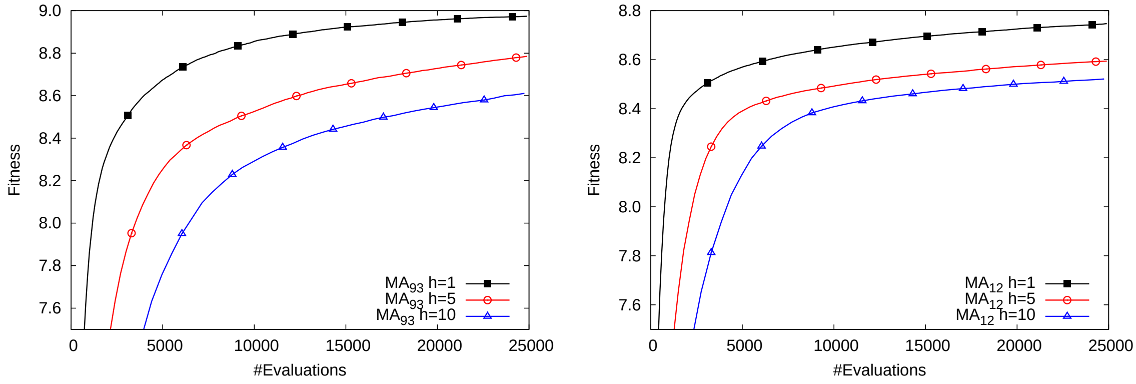


(c) $MA_{12}$-random (d) $MA_{12}$-best

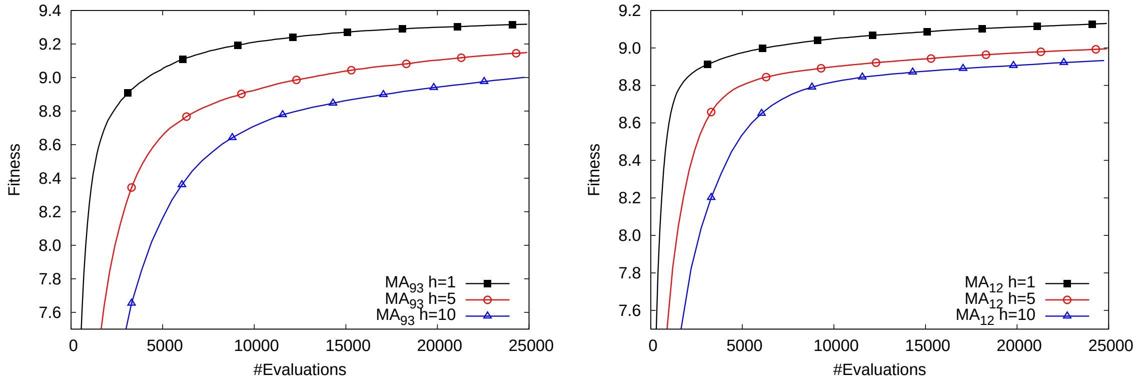
Fig. 4: Progress of MBF for $MA_{93}$ and $MA_{12}$ using local search on different numbers of random (left) and best (right) chromosomes on the dataset of 90k transactions

Table 2: Fitness, overlap, coverage, suitability, and fuzzy support of the membership functions obtained from $MA_{93}$ and $MA_{12}$ using local search on different numbers of random and best chromosomes on the dataset of 90k transactions.

| MA | Strategy | #Chroms | Overlap | Coverage | Suitability | Fuzzy Support | Fitness |
|---|---|---|---|---|---|---|---|
| $MA_{93}$ | Random | 1 | 0.0086 | 1.0020 | 1.0106 | 9.2933 | 9.2009 |
| | | 2 | 0.0085 | 1.0022 | 1.0107 | 9.3042 | 9.2114 |
| | | 5 | 0.0103 | 1.0024 | 1.0127 | 9.3519 | 9.2415 |
| | | 10 | 0.0059 | 1.0028 | 1.0087 | 9.3505 | 9.2741 |
| | | 50 | 0.0032 | 1.0038 | 1.0070 | 9.3802 | 9.3176 |
| | Best | 1 | 0.0135 | 1.0016 | 1.0152 | 9.1396 | 9.0098 |
| | | 2 | 0.0140 | 1.0017 | 1.0156 | 9.1728 | 9.0362 |
| | | 5 | 0.0098 | 1.0018 | 1.0116 | 9.2006 | 9.0994 |
| | | 10 | 0.0100 | 1.0021 | 1.0121 | 9.2551 | 9.1495 |
| | | 50 | 0.0032 | 1.0038 | 1.0070 | 9.3802 | 9.3176 |
| $MA_{12}$ | Random | 1 | 0.0000 | 1.0016 | 1.0016 | 9.1882 | 9.1731 |
| | | 2 | 0.0000 | 1.0016 | 1.0016 | 9.1887 | 9.1737 |
| | | 5 | 0.0000 | 1.0018 | 1.0018 | 9.1857 | 9.1693 |
| | | 10 | 0.0000 | 1.0019 | 1.0019 | 9.1574 | 9.1402 |
| | | 50 | 0.0000 | 1.0026 | 1.0026 | 9.1546 | 9.1306 |
| | Best | 1 | 0.0000 | 1.0016 | 1.0016 | 9.0373 | 9.0231 |
| | | 2 | 0.0000 | 1.0016 | 1.0016 | 9.0501 | 9.0353 |
| | | 5 | 0.0000 | 1.0017 | 1.0017 | 9.0705 | 9.0550 |
| | | 10 | 0.0000 | 1.0019 | 1.0019 | 9.0856 | 9.0684 |
| | | 50 | 0.0000 | 1.0026 | 1.0026 | 9.1546 | 9.1306 |



(a) $MA_{93}$ (10k) (b) $MA_{12}$ (10k)

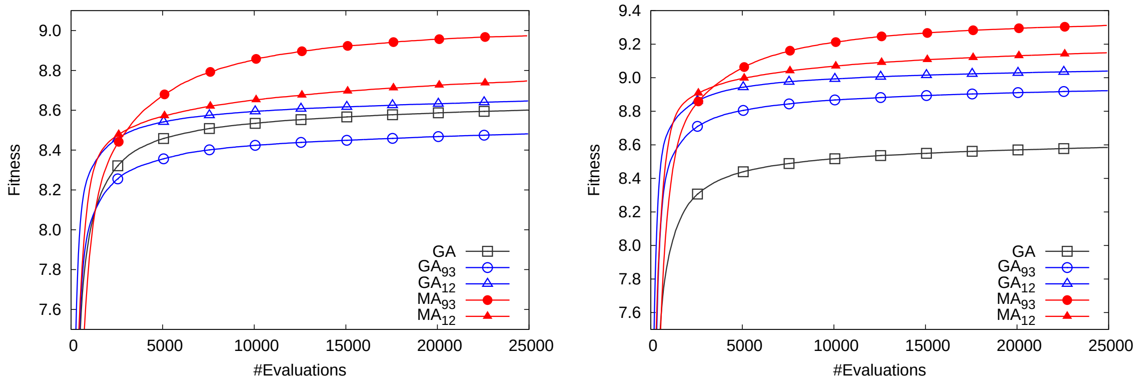


(c) $MA_{93}$ (90k) (d) $MA_{12}$ (90k)

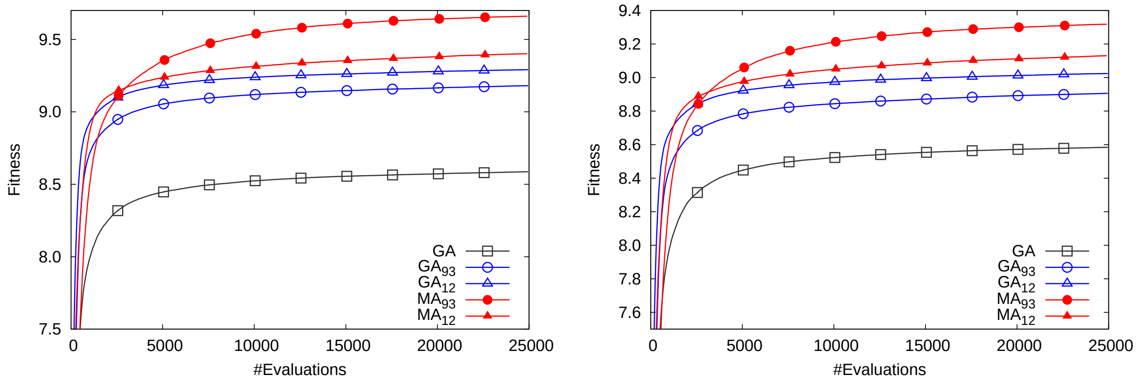
Fig. 5: Progress of MBF for $MA_{93}$ and $MA_{12}$ using local search with different size $h$ on the datasets of 10k and 90k transactions

Table 3: Fitness, overlap, coverage, suitability, and fuzzy support of the membership functions obtained from $MA_{93}$ and $MA_{12}$ using local search with different size $h$ on the datasets of 10k and 90k transactions

| Data | MA | Size | Overlap | Coverage | Suitability | Fuzzy Support | Fitness |
|---|---|---|---|---|---|---|---|
| 10k | $MA_{93}$ | 1 | 0.0050 | 1.0039 | 1.0089 | 9.0501 | 8.9738 |
| | | 5 | 0.0066 | 1.0110 | 1.0176 | 8.9349 | 8.7852 |
| | | 10 | 0.0082 | 1.0167 | 1.0249 | 8.8199 | 8.6107 |
| | $MA_{12}$ | 1 | 0.0000 | 1.0027 | 1.0027 | 8.7709 | 8.7473 |
| | | 5 | 0.0000 | 1.0075 | 1.0075 | 8.6588 | 8.5943 |
| | | 10 | 0.0000 | 1.0120 | 1.0121 | 8.6235 | 8.5214 |
| 90k | $MA_{93}$ | 1 | 0.0032 | 1.0038 | 1.0070 | 9.3802 | 9.3176 |
| | | 5 | 0.0060 | 1.0111 | 1.0170 | 9.3008 | 9.1493 |
| | | 10 | 0.0053 | 1.0167 | 1.0219 | 9.1951 | 9.0013 |
| | $MA_{12}$ | 1 | 0.0000 | 1.0026 | 1.0026 | 9.1546 | 9.1306 |
| | | 5 | 0.0000 | 1.0078 | 1.0078 | 9.0649 | 8.9951 |
| | | 10 | 0.0001 | 1.0118 | 1.0119 | 9.0381 | 8.9326 |



(a) $10k$ transactions (b) $30k$ transactions



(c) $50k$ transactions (d) $90k$ transactions

Fig. 6: Progress of MBF against the number of fitness evaluations for GA, $GA_{93}$, $GA_{12}$, $MA_{93}$, and $MA_{12}$ on different datasets

$MA_{12}$ gain higher fitness than $GA_{93}$ and $GA_{12}$, respectively. When comparing the two MAs, $MA_{93}$ outperforms $MA_{12}$ in terms of fitness. As aforementioned, the two constraints on $MA_{12}$ help to guarantee the suitability but impairs the fuzzy support; consequently, $MA_{12}$ gains lower fitness than $MA_{93}$ does. Table 4 presents the MBF obtained and the $p$-values of one-tailed $t$-test on the MBF. With confidence level $\alpha = 0.01$, the $t$-test results indicate that $MA_{93}$ and $MA_{12}$ achieve significantly better fitness than $GA_{93}$ and $GA_{12}$, respectively. In addition, $MA_{93}$ significantly outperforms $MA_{12}$ on all datasets. The superior outcomes validate the benefits of the structure representation and local search.

Table 5 compares the overlap, coverage, suitability, and fuzzy support of the membership functions obtained from the five test algorithms. The results indicate that $GA_{12}$ and $MA_{12}$ benefit from the two constraints in coverage and overlap; nevertheless; their fuzzy support values are relatively low due to the limitation on the structure types. By contrast, GA attains the highest fuzzy support at the cost of suitability; especially, it suffers from excessive overlap of membership functions. Compared to the above methods, $MA_{93}$ maintains a balance between fuzzy support and suitability: By virtue of the structure representation and local search, $MA_{93}$ achieves the best fitness with suitability better than GA and fuzzy support higher than $GA_{12}$ and $MA_{12}$.

This study further investigates the performance of the proposed MA on a large dataset, consisting of $1000k$ transactions, to simulate mining fuzzy association rules on big data. The experimental results on Table 6 show that the proposed $MA_{93}$ and $MA_{12}$ achieve a satisfactory performance on optimizing the membership functions for large datasets. With confidence level $\alpha = 0.01$, the $t$-test results indicate that both $MA_{93}$ and $MA_{12}$ obtain significantly higher fitness than GA, $GA_{93}$, and $GA_{12}$. Although $MA_{93}$ and $MA_{12}$ have a slightly worse coverage, they gain better overlap and fuzzy support than $GA_{93}$ and $GA_{12}$, respectively. Figure 7 plots the anytime behaviors of the five test algorithms in fitness, overlap, coverage, and fuzzy support. The figure shows the advantage of local search: The two MAs progress beyond the GAs in overlap, fuzzy support, and fitness.

Table 4: MBF and $p$-values for the five test algorithms on the datasets of different sizes. The $p$-values account for the results of $t$-test on the MBF obtained from $X$ and $Y$ algorithms (denoted by $X$:$Y$), where positive $p$-values indicate that $Y$ is superior to $X$. Boldfaced MBF marks the best result among the five algorithms; boldfaced $p$-values denote the statistical significance with confidence level $\alpha = 0.01$

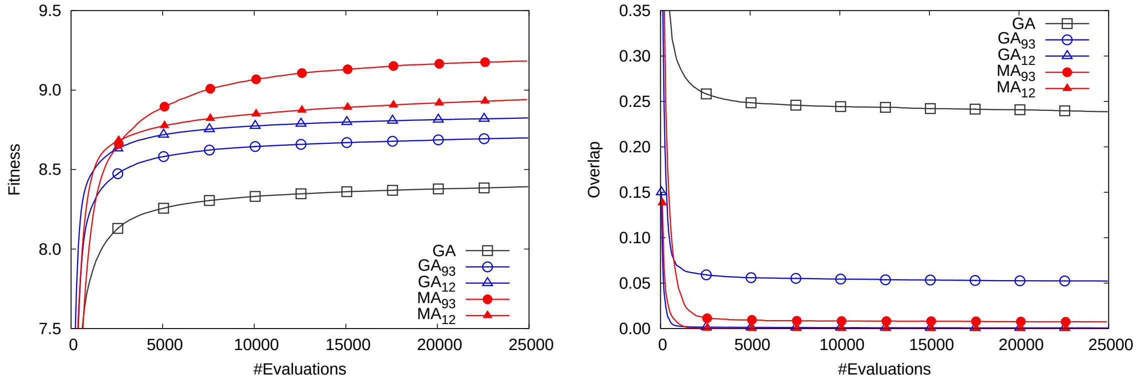
| | MBF | | | | | $p$-value | | |
|---|---|---|---|---|---|---|---|---|
| #Tr. | GA | GA$_{93}$ | GA$_{12}$ | MA$_{93}$ | MA$_{12}$ | GA$_{93}$:MA$_{93}$ | GA$_{12}$:MA$_{12}$ | MA$_{12}$:MA$_{93}$ |
| 10$k$ | 8.6010 | 8.4815 | 8.6467 | **8.9738** | 8.7473 | **2.60E-45** | **1.25E-21** | **1.19E-37** |
| 30$k$ | 8.5845 | 8.9229 | 9.0398 | **9.3120** | 9.1491 | **2.45E-41** | **1.75E-25** | **8.50E-30** |
| 50$k$ | 8.5865 | 9.1803 | 9.2911 | **9.6599** | 9.4012 | **2.46E-42** | **5.96E-21** | **7.74E-41** |
| 70$k$ | 8.5970 | 8.8305 | 8.9550 | **9.2304** | 9.0579 | **6.87E-46** | **1.38E-21** | **2.62E-29** |
| 90$k$ | 8.5842 | 8.9057 | 9.0240 | **9.3176** | 9.1306 | **3.45E-45** | **4.34E-24** | **2.64E-34** |

Table 5: Overlap, coverage, suitability, and fuzzy support of the membership functions obtained from the five test algorithms on the data of different sizes. Boldface marks the best result among the five algorithms

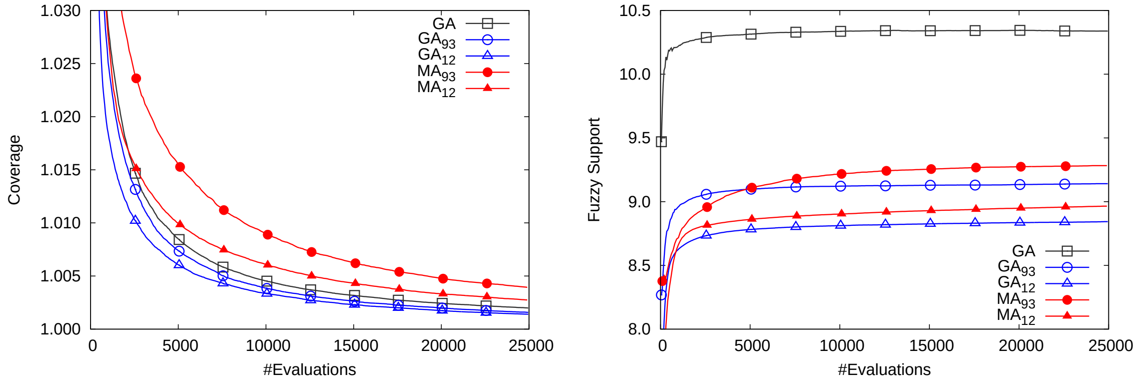
| #Tr. | Algorithm | Overlap | Coverage | Suitability | Fuzzy Support | Fitness |
|---|---|---|---|---|---|---|
| 10$k$ | GA | 0.1928 | 1.0019 | 1.1946 | **10.1821** | 8.6010 |
| | GA$_{93}$ | 0.0470 | 1.0016 | 1.0485 | 8.8656 | 8.4815 |
| | GA$_{12}$ | 0.0010 | **1.0014** | **1.0024** | 8.6673 | 8.6467 |
| | MA$_{93}$ | 0.0050 | 1.0039 | 1.0089 | 9.0501 | **8.9738** |
| | MA$_{12}$ | **0.0000** | 1.0027 | 1.0027 | 8.7709 | 8.7473 |
| 30$k$ | GA | 0.1886 | 1.0018 | 1.1904 | **10.1259** | 8.5845 |
| | GA$_{93}$ | 0.0331 | 1.0016 | 1.0346 | 9.2157 | 8.9229 |
| | GA$_{12}$ | 0.0007 | **1.0014** | **1.0021** | 9.0596 | 9.0398 |
| | MA$_{93}$ | 0.0046 | 1.0040 | 1.0086 | 9.3887 | **9.3120** |
| | MA$_{12}$ | **0.0000** | 1.0026 | 1.0026 | 9.1726 | 9.1491 |
| 50$k$ | GA | 0.2012 | 1.0019 | 1.2030 | **10.2289** | 8.5865 |
| | GA$_{93}$ | 0.0337 | 1.0015 | 1.0352 | 9.4814 | 9.1803 |
| | GA$_{12}$ | 0.0010 | **1.0014** | **1.0024** | 9.3134 | 9.2911 |
| | MA$_{93}$ | 0.0038 | 1.0041 | 1.0079 | 9.7329 | **9.6599** |
| | MA$_{12}$ | **0.0000** | 1.0028 | 1.0028 | 9.4271 | 9.4012 |
| 70$k$ | GA | 0.1992 | 1.0019 | 1.2011 | **10.2227** | 8.5970 |
| | GA$_{93}$ | 0.0362 | 1.0015 | 1.0377 | 9.1479 | 8.8305 |
| | GA$_{12}$ | 0.0009 | **1.0014** | **1.0023** | 8.9754 | 8.9550 |
| | MA$_{93}$ | 0.0037 | 1.0039 | 1.0076 | 9.2983 | **9.2304** |
| | MA$_{12}$ | **0.0000** | 1.0028 | 1.0028 | 9.0829 | 9.0579 |
| 90$k$ | GA | 0.2019 | 1.0019 | 1.2038 | **10.2386** | 8.5842 |
| | GA$_{93}$ | 0.0378 | 1.0016 | 1.0393 | 9.2322 | 8.9057 |
| | GA$_{12}$ | 0.0010 | **1.0014** | **1.0024** | 9.0457 | 9.0240 |
| | MA$_{93}$ | 0.0032 | 1.0038 | 1.0070 | 9.3802 | **9.3176** |
| | MA$_{12}$ | **0.0000** | 1.0026 | 1.0026 | 9.1546 | 9.1306 |

Table 6: Fitness, overlap, coverage, suitability, and fuzzy support of the membership functions obtained from the five test algorithms on the dataset of 1000$k$ transactions. Boldface marks the best result among the five algorithms

| #Tr. | Algorithm | Overlap | Coverage | Suitability | Fuzzy Support | Fitness |
|---|---|---|---|---|---|---|
| 1000$k$ | GA | 0.2387 | 1.0020 | 1.2407 | **10.338** | 8.3919 |
| | GA$_{93}$ | 0.0523 | 1.0016 | 1.0539 | 9.1410 | 8.6991 |
| | GA$_{12}$ | 0.0006 | **1.0014** | **1.0020** | 8.8431 | 8.8251 |
| | MA$_{93}$ | 0.0074 | 1.0039 | 1.0114 | 9.2827 | **9.1831** |
| | MA$_{12}$ | **0.0000** | 1.0027 | 1.0027 | 8.9648 | 8.9403 |



(a) Fitness (b) Overlap



(c) Coverage (d) Fuzzy support

Fig. 7: Variation of fitness, overlap, coverage, and fuzzy support against the number of evaluations for the five test algorithms on $1000k$ transactions

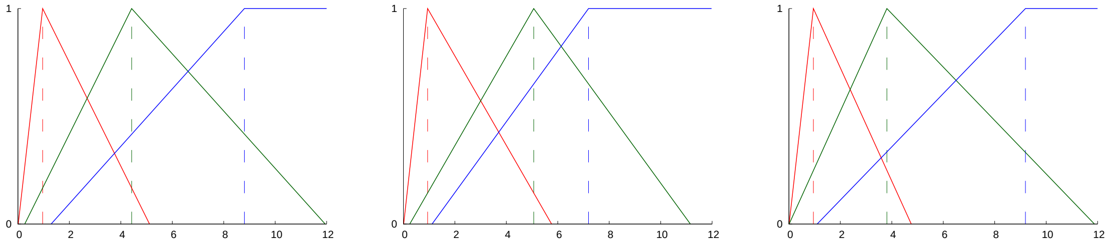
They also address the issue that GA tends to sacrifice overlap for fuzzy support. These outcomes confirm the proposed MA is capable of improving GA in fuzzy data mining on large datasets. They also validate the benefits of the chromosome representation and the local search operator for the MA.

### 4.3 Comparison of Membership Functions

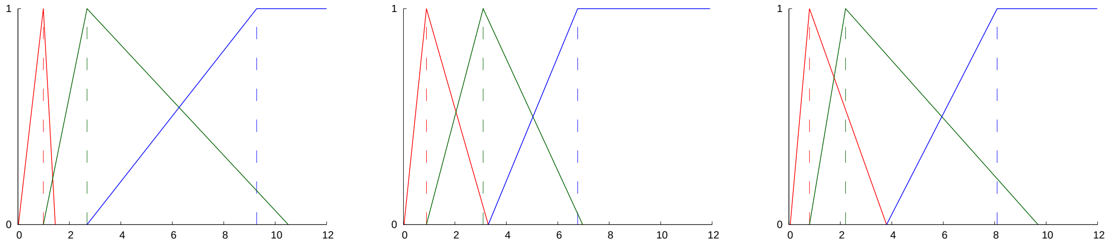
The shape of membership functions is key to the interpretability of fuzzy sets and exerts significant influence on coverage, overlap, and fuzzy support. This section compares the membership functions obtained from GA, $\text{MA}_{93}$ and $\text{MA}_{12}$ on the data of 10k, 90k, and 1000k transactions. According to Fig. 8, GA achieves membership functions with highest fuzzy support but lowest suitability. In particular, the membership functions generated by GA have serious overlap, making the fuzzy regions turn out to be trivial. That is, GA tends to sacrifice suitability for fuzzy support. On the other hand, the results show that $\text{MA}_{93}$ and $\text{MA}_{12}$ can maintain proper overlap and coverage while pursuing fuzzy support. Hence, the membership functions obtained from $\text{MA}_{93}$ and $\text{MA}_{12}$ have higher fitness values and reasonable shapes than that from GA. In addition, $\text{MA}_{12}$ imposes constraints (12) and (13) to secure suitable coverage and overlap; nevertheless, these constraints also hinder $\text{MA}_{12}$ attaining high fuzzy support because the limited fuzzy regions bring about lower fuzzy support, which is reflected in the resultant membership functions in Fig. 8.

## 5 Conclusions

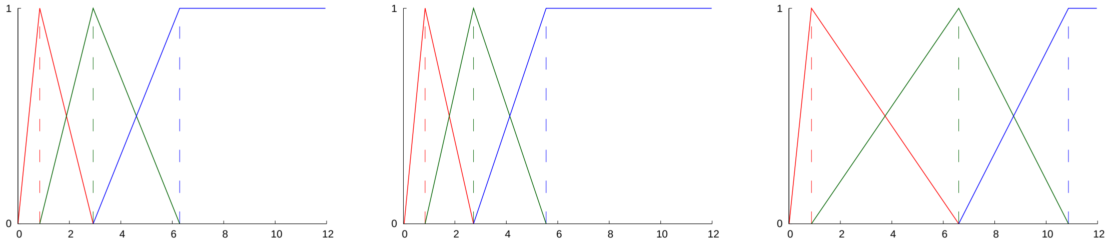
This study proposes an MA to optimize the membership functions for fuzzy association rule mining. The MA uses a chromosome representation consisting of the parameters and structure type of membership functions. The consideration of structure in the representation gains three advantages. First, the structure type helps to filter out the illegal membership functions, thereby reducing the search space. Second, it facilitates the design of heuristics and constraints for appropriate cover-



(a) GA (10$k$ transactions) (b) GA (90$k$ transactions) (c) GA (1000$k$ transactions)



(d) $\mathrm{MA}_{93}$ (10$k$ transactions) (e) $\mathrm{MA}_{93}$ (90$k$ transactions) (f) $\mathrm{MA}_{93}$ (1000$k$ transactions)



(g) $\mathrm{MA}_{12}$ (10$k$ transactions) (h) $\mathrm{MA}_{12}$ (90$k$ transactions) (i) $\mathrm{MA}_{12}$ (1000$k$ transactions)

Fig. 8: Comparison of membership functions obtained from GA, $\mathrm{MA}_{93}$ and $\mathrm{MA}_{12}$ on the datasets of 10k (left), 90k (center), and 1000k (right) transactions

age and overlap. Third, local search can be performed on the structure types; accordingly; we develop a local search operator to improve the search efficiency of the MA.

This study conducts a series of experiments to examine the performance of the proposed MA. Two instances of the MA, namely $\mathrm{MA}_{93}$ and $\mathrm{MA}_{12}$, are investigated to consider different constraints on the structures. The experimental results show that the MAs achieve significantly higher fitness than GA and GA using the structure representation. In exploring the influences on the membership functions, the results show that the MA can attain a high fuzzy support and maintain suitability, i.e., appropriate overlap and coverage. The improvement of the proposed MAs upon the GAs becomes even greater as the dataset grows to one million transactions. These outcomes indicate the advantages of the chromosome representation and local search operator for the MA. In addition, they validate the effectiveness of the proposed MAs on the optimization of membership functions for fuzzy association rule mining.

Future work includes some directions: Advanced design of heuristics and constraints based on structure types can be considered for enhancing the MA. Additionally, applying the proposed MA to optimize the membership functions for other fuzzy systems is promising. Extending the performance comparison with more evolutionary algorithms would also help to validate the advantages of the proposed MA.

## Acknowledgment

The authors would like to thank the editor and reviewers for their valuable comments and suggestions. This work was supported by the Ministry of Science and Technology of Taiwan, under contract MOST 104-2221-E-194-015-MY3.

## Compliance with Ethical Standards

**Conflict of interest:** The authors declare that they have no conflict of interest.
**Ethical approval:** This article does not contain any studies with human participants or animals performed by any of the authors.

## 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. J. Alcalá-Fdez, R. Alcalá, and F. Herrera. A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning. *IEEE Transactions on Fuzzy Systems*, 19(5):857–872, 2011.
3. M. Antonelli, P. Ducange, and F. Marcelloni. A fast and efficient multi-objective evolutionary learning scheme for fuzzy rule-based classifiers. *Information Sciences*, 283(1):36–54, 2014.
4. K. Balázs and L. T. Kóczy. Genetic and bacterial memetic programming approaches in hierarchical-interpolative fuzzy system construction. In *Proceedings of IEEE International Conference on Fuzzy Systems*, pages 1–8, 2012.
5. G.-R. Cai, S.-Z. Li, and S.-L. Chen. Mining fuzzy association rules by using nonlinear particle swarm optimization. *Quantitative Logic and Soft Computing*, 82:621–630, 2010.
6. K. Chan and W. H. Au. Mining fuzzy association rules. In *Proceedings of the International Conference on Information and Knowledge Management*, pages 209–215, 1997.
7. C.-H. Chen, A.-F. Li, and Y.-C. Lee. A fuzzy coherent rule mining algorithm. *Applied Soft Computing*, 13(7):3422–3428, 2013.
8. 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.
9. M.-S. Chen, J. Han, and P. S. Yu. Data mining: An overview from a database perspective. *IEEE Transactions on Knowledge and data Engineering*, 8(6):866–883, 1996.
10. U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, 17:37–54, 1996.
11. M. Fazzolari, R. Alcala, Y. Nojima, H. Ishibuchi, and F. Herrera. A review of the application of multiobjective evolutionary fuzzy systems: Current status and further directions. *IEEE Transactions on Fuzzy Systems*, 21(1):45–65, 2013.
12. L. Feng, Y. S. Ong, A. H. Tan, and I. W. Tsang. Memes as building blocks: a case study on evolutionary optimization + transfer learning for routing problems. *Memetic Computing*, 7(3):159–180, 2015.
13. L. Gál, J. Botzheim, L. T. Kóczy, and A. E. Ruano. Fuzzy rule base extraction by the improved bacterial memetic algorithm. In *Proceedings of International Symposium on Applied Machine Intelligence and Informatics*, pages 49–53, 2008.
14. D.-T. Ho and J. M. Garibaldi. An improved optimisation framework for fuzzy time-series prediction. In *Proceedings of IEEE International Conference on Fuzzy Systems*, pages 1–8, 2013.
15. 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.
16. T.-P. Hong, C.-S. Kuo, and S.-C. Chi. Mining association rules from quantitative data. *Intelligent Data Analysis*, 3(5):363–376, 1999.
17. 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.
18. X.-P. Ji, X.-B. Cao, and K. Tang. Sequence searching and evaluation: A unified approach for aircraft arrival sequencing and scheduling problems. *Memetic Computing*, 8(2):109–123, 2016.
19. C. M. Kuok, A. Fu, and M. H. Wong. Mining fuzzy association rules in databases. *ACM SIGMOD Record*, 27(1):41–46, 1998.
20. 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.
21. M. A. El Majdouli, I. Rbouh, S. Bougrine, B. El Benani, and A. A. El Imrani. Fireworks algorithm framework for big data optimization. *Memetic Computing*, pages 1–15, 2016.
22. D. Meng and Z. Pei. Extracting linguistic rules from data sets using fuzzy logic and genetic algorithms. *Neurocomputing*, 78(1):45–54, 2012.
23. B. Minaei-Bidgoli, R. Barmaki, and M. Nasiri. Mining numerical association rules via multi-objective genetic algorithms. *Information Sciences*, 233(1):15–24, 2013.
24. S. Mishra, D. Mishra, and S. K. Satapathy. Particle swarm optimization based fuzzy frequent pattern mining from gene expression data. In *Proceedings of the International Conference on Computer & Communication Technology*, pages 15–20, 2011.
25. M. Nekkaa and D. Boughaci. A memetic algorithm with support vector machine for feature selection and classification. *Memetic Computing*, 7(1):59–73, 2015.
26. Y. S. Ong, M. H. Lim, and X. S. Chen. Research frontier: Memetic computation — past, present & future. *IEEE Computational Intelligence Magazine*, 5(2):24–36, 2010.
27. J. Prakash and P. K. Singh. An effective multiobjective approach for hard partitional clustering. *Memetic Computing*, 7(2):93–104, 2015.
28. H. R. Qodmanan, M. Nasiri, and B. Minaei-Bidgoli. Multi objective association rule mining with genetic algorithm without specifying minimum support and minimum confidence. *Expert Systems with Applications*, 38(1):288–298, 2011.
29. F. Rudziǹski. A multi-objective genetic optimization of interpretability-oriented fuzzy rule-based classifiers. *Applied Soft Computing*, 38:118–133, 2016.
30. H. Samma, C. P. Lim, J. M. Saleh, and S. A. Suandi. A memetic-based fuzzy support vector machine model and its application to license plate recognition. *Memetic Computing*, 8(3):235–251, 2016.
31. R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. *ACM SIGMOD Record*, 25(2):1–12, 1996.
32. A. Tsakonas. Local and global optimization for takagi-sugeno fuzzy system by memetic genetic programming. *Expert Systems with Applications*, 40(8):3282–3298, 2013.
33. Y. Zhang, J. Liu, M. X. Zhou, and Z. Z. Jiang. A multi-objective memetic algorithm based on decomposition for big optimization problems. *Memetic Computing*, 8(1):45–61, 2016.