https://doi.org/10.1016/j.asoc.2019.105518

This is the Pre-Published Version.

Mining significant fuzzy association rules with differential evolution algorithm

Anshu Zhang[a] and Wenzhong Shi[a,*]

[a] Department of Land Surveying and Geo-Informatics, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, P.R. China

* Corresponding author. Tel: +852 2766 5975, Fax: +852 2330 2994.

Email addresses: lswzshi@polyu.edu.hk (W. Shi), anshu.zhang@connect.polyu.hk (A. Zhang)

Abstract: This article presents a new differential evolution (DE) algorithm for mining optimized statistically significant fuzzy association rules that are abundant in number and high in rule interestingness measure (RIM) values, with strict control over the risk of spurious rules. The risk control over spurious rules, as the most distinctive feature of the proposed DE compared with existing evolutionary algorithms (EAs) for association rule mining (ARM), is realized via two new statistically sound significance tests on the rules. The two tests, in the experimentwise and generationwise adjustment approach, can respectively limit the familywise error rate (the probability that any spurious rules occur in the ARM result) and percentage of spurious rules upon the user specified level. Experiments on variously sized data show that the proposed DE can keep the risk of spurious rules well below the user specified level, which is beyond the ability of existing EA-based ARM. The new method also carries forward the advantages of EA-based ARM and distinctive merits of DE in optimizing the rules: it can obtain several times as many rules and as high RIM values as conventional non-evolutionary ARM, and even more informative rules and better RIM values as genetic-algorithm-based ARM. Case studies on hotel room price determinants and wildfire risk factors demonstrate the practical usefulness of the proposed DE.

Keywords: association rule mining; evolutionary computation; differential evolution; statistical evaluation; quality control

© 2019. This manuscript version is made available under the CC-BY-NC-ND 4.0 license https://creativecommons.org/licenses/by-nc-nd/4.0/

1. Introduction

Association rule mining (ARM) has been an important subfield in data mining and a powerful tool for practical decision support. ARM seeks for implicit ‘antecedent $\rightarrow$ consequence’ patterns called *association rules* in data that meet specified constraints on *rule interestingness measures* (RIMs) and other criteria. The quality of ARM results concerns:

- Abundance of authentic rules, which is the basic value of resultant rules;
- Control over spurious rules, that is, rules not meeting specified constraints but falsely admitted into ARM results;
- Accuracy and fitness of RIM values. The accuracy measures the closeness of RIM values observed in data to their true values. The fitness is with respect to specific user needs; for example, in a business profit study, rules of high fitness can be those with high values of a RIM indicating profit gains.

Fuzzy ARM with evolutionary algorithms (EAs) [1–5] is a powerful approach for enhancing the quality of resultant rules. In ARM, domains of numerical data attributes are normally first discretized into intervals. Then these intervals are explored for rules and usually assigned linguistic concepts, for example, ‘high’ and ‘near’, for interpreting the rules. In ordinary ARM, numerical data is discretized into crisp value intervals, which is inaccurate for the commonly gradual or vague linguistic concepts [6–7] and can greatly distort resultant rules and RIM values. Fuzzy ARM [8] may alleviate this problem by discretizing the data into fuzzy intervals, thereby improving the accuracy of RIM values. Also, experts often lack the knowledge of appropriate data discretization schemes, including the number of concepts and original data value interval for each concept. This issue can be addressed by EAs that mimic natural selection. EA-based fuzzy ARM can therefore generate optimized data discretization schemes and rules for specific user demands [9] with boosted number of rules discovered and/or fitness of their RIM values, as well as more accurate RIM values due to the fuzzy approach for semantic representations in the rules.

A critical barrier in EA-based ARM remains on the control over spurious rules. Due to the enormous number of candidate rules, spurious rules can take up significant percentages or even become the majority in ARM results, mislead users into poor decisions, and make the results unusable [10–11]. Statistical hypothesis testing plays a key role in controlling spurious rules [12–15]. Data are finite representations of associations in the real world which can potentially repeat for infinite times, thus rules may fulfil specified interestingness constraints in data by pure chance when they do not meet the constraints in reality. The statistical tests aim at filtering out such spurious rules and admit only statistically significant ones. *Statistically sound evaluation* [10] is a particularly effective technique and can control the *familywise error rate* (FWER), the chance that any spurious rules exist in entire ARM results, upon a low user specified level, for example 5%. This technique adjusts significance levels of statistical tests by the *search space size*, or the number of all candidate rules that can be constituted by the data, when large numbers of rules are evaluated concurrently. Albeit successful in conventional ARM with predefined data discretization schemes, current statistically sound evaluation is inapplicable to EA-based ARM, since the latter holds completely different searching methodology and search space size from conventional ARM. Also, little research has been done on controlling spurious rules in EAs with other statistical testing techniques.

This article presents a differential evolution (DE) algorithm for mining significant fuzzy association rules (DESigFAR). The most distinctive feature of DESigFAR against existing EA-based ARM is its ability to strictly control the risk of spurious rules via newly developed statistically sound tests. Also, as the first fuzzy ARM algorithm based on DE, one of the latest and best performing EA techniques, the proposed DE can produce optimized ARM results with abundant rules and RIM values of high fitness and accuracy, thus achieves an overall improvement on the quality of ARM results.

DESigFAR contains two options of statistical tests on rules: the *experimentwise* and *generationwise adjustment approach*, which can control the FWER and percentage of spurious rules under the user specified level, respectively. These approaches maintain the key idea of significance level adjustment based on search space sizes in the statistically sound evaluation. A new evolutionary model is also designed for feasible and computationally efficient DE with these two approaches. The proposed method is experimentally proven to produce several times as many rules and high RIM values as conventional non-EA statistically sound ARM, and performs better than genetic algorithm, the dominating technique in current EA-based ARM. While existing EA-based ARM without proper statistical tests cannot effectively control spurious rules, DESigFAR can keep the FWER or percentage of spurious rules well below user required level. In the case studies on hotel room price and wildfire risk factors, the new algorithm has helped deepen the understanding on interactions of the factors and their influences on the room prices and fire risks.

This article is organized as follows. Section 2 reviews existing methods for avoiding spurious association rules and EA-based fuzzy ARM. Section 3 describes the methodology of DESigFAR. Section 4 experiments DESigFAR with data in various conditions, analyzes the results against existing ARM methods, and discusses practical implications of the hotel room pricing and wildfire risk case studies. Section 5 makes the concluding remarks.

2. **Prior works**

2.1 ARM and avoidance of spurious rules

This article focuses on ARM with numerical data that usually takes an attribute-value form, that is, each record *R* in dataset *D* contains an *item* like 'attribute = value' for each attribute in *D*. An association rule is an implication $X \rightarrow Y$, where the *antecedent X* and *consequent Y* are sets of items in *D*. This study is described using single-item consequent *y*, and the method it presents equally applies to multi-item consequents.

ARM seeks for association rules that meet specified constraints, mostly minimum values of certain RIMs. The most basic RIMs are *support* and confidence [16]:

$$supp\left(X \to y\right) = supp\left(X \cup \{y\}\right) = \left|R \in D : X \cup \{y\} \subseteq R\right|, \tag{1}$$

$$conf\left(X \to y\right) = supp\left(X \to y\right) / supp\left(X\right). \tag{2}$$

Numerous other RIMs have also been proposed, among which 61 well-known ones are reviewed by Tew *et al.* [17]. Other quantitative criteria proposed for pruning uninteresting rules are usually also relevant to RIMs. For instance, the productive rule criterion [10] requires a rule $X \to y$ to have a positive *improvement* [18]:

$$imp(X \to y) = conf(X \to y) - \max_{Z \subset X}(conf(Z \to y)) > 0. \tag{3}$$

That is, each item in *X* must make the rule have a higher confidence. Productive rules are highly desirable if the ARM aims to find positive data associations. The non-redundant rule [19] and actionable rule [20] criteria also require rules to have higher confidences than their specializations or generalizations. Specializations of a rule are obtained by adding extra items, and generalizations are obtained by removing some items. Although RIMs and other relevant criteria are very useful in selecting interesting rules, they are prone to accept spurious rules that fulfil them in data due to pure chance instead of real data associations. Spurious rules normally take up over 10%, and sometimes even the majority, of discovered rules [10, 11].

Statistical hypothesis testing is a key solution to spurious rules [12–15]. For each rule $X \to y$, a test results in a probability *p* that $X \to y$ has the observed RIM value when the null hypothesis "$X \to y$ does not meet the specified constraint in reality" is true, or that the rule is spurious. Rules with *p* values above significance level *α*, say 0.05, are considered too risky to be spurious and pruned. Hereafter the tests are exemplified by the test for productive rules, and the same approach can be right applied to other tests. To test the productivity of a rule $X \to y$, or whether $imp(X \to y) > 0$, is to test

$$\forall Z \subseteq X,\ \Pr\left(y \mid X\right) > \Pr\left(y \mid X \setminus Z\right), \tag{4}$$

where \ denotes set difference. Alternatively, a simplified test with similar result and lower computational cost can be conducted [10]:

$$\begin{aligned} \text{Null hypothesis } H_0 &: \exists x_m \in X,\ \Pr\left(y \mid X\right) \le \Pr\left(y \mid X \setminus \{x_m\}\right) \\ \text{Alternative hypothesis } H_1 &: \forall x_m \in X,\ \Pr\left(y \mid X\right) > \Pr\left(y \mid X \setminus \{x_m\}\right) \end{aligned}. \tag{5}$$

Chi-square is a common statistic for testing H_1 in Eq. (5): for each $x_m \in X$,

$$\chi_m^2 = \frac{\left(ad - bc\right)\left(a + b + c + d\right)}{\left(a + b\right)\left(c + d\right)\left(a + c\right)\left(b + d\right)}\ , \tag{6}$$

where

$$\begin{aligned} a &= supp\left(X \cup \{y\}\right) \\ b &= supp\left(X \cup \neg\{y\}\right) \\ c &= supp\left(\left(X \setminus \{x_m\}\right) \cup \neg\{x_m\} \cup \{y\}\right) \\ d &= supp\left(\left(X \setminus \{x_m\}\right) \cup \neg\{x_m\} \cup \neg\{y\}\right) \end{aligned}\ , \tag{7}$$

and ¬ refers to that the record does not contain the item. The p_m value for x_m can be looked up from the χ^2 table with one degree of freedom. $X \rightarrow y$ is accepted if the p value for every $x \in X$ is below the significance level.

When many rules are evaluated concurrently, the tests face the multiple comparisons problem [21]: testing rules at a significance level α, say 0.05, only guarantees that each accepted rule has less than 0.05 probability to be spurious. Then the number of spurious rules might be nearly 5% of rules that should be rejected. If only small parts of the evaluated rules, probably less than 5%, are authentic, the tests may accept more spurious rules than authentic ones. This problem may be addressed by a Bonferroni correction to adjust the significance level to $\kappa = \alpha/n$, where n is the number of hypothesis tests applied [22]. Yet it is often ineffective to take the number of tested rules as n, since the tested rules are typically pre-filtered by other constraints such as the minimum confidence and are more likely to pass the tests than arbitrary rules.

Webb [10] proposed the statistically sound evaluation on rules which sets $\kappa = \alpha/s$, where s is the search space size, or the total number of potential rules that can be constituted by data items. That is, κ is adjusted by the numbers of all potential rules instead of only pre-filtered and tested ones. The computation of s is detailed in [10]. With α=0.05, statistically sound tests can achieve an FWER below 1% and less than 0.1% spurious rules. This highly effective technique, on the other hand, is conservative and can also reject many authentic rules, making the number of rules discovered and fitness of RIM values even more sensitive to data discretization schemes. This greatly motivates the development of statistically sound tests for DE-based ARM where the data discretization schemes can be optimized.

2.2 Fuzzy ARM

As said in Section 1, fuzzy ARM can better model gradual or vague concepts than ordinary ARM by discretizing numerical data domains into fuzzy intervals, thereby improving the accuracy of RIM values. For numerical attribute x and a concept l for x, a fuzzy *membership function* μ_l is defined to map each value in x to a *membership degree* $\mu_l(x) \in [0, 1]$ that x belongs to l. $core(\mu_l) = \{x \in U | \mu_l(x) = 1\}$ and $supp(\mu_l) = \{x \in U | \mu_l(x) > 0\}$ are called *core* and *support* of μ_l [23]. Fig. 1 illustrates several common forms of membership functions.

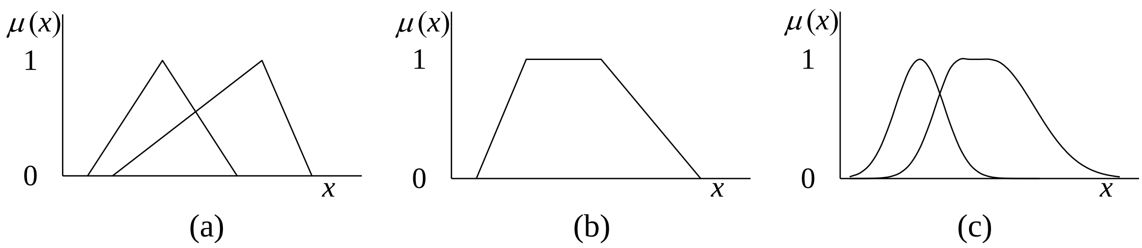

Fig. 1. Common fuzzy membership functions for ARM. a. Triangular [2, 24–26] b. Trapezoidal [27] c. Gaussian-curve-based [28, 29]. Reproduced from [30].

Conjunctive membership degrees to multiple concepts can be computed by t-norm, an associative, commutative and monotone function denoted as $\otimes$. The commonest t-norms include minimum t-norm: $\alpha \otimes_{\min} \beta = \min(\alpha, \beta)$ and product t-norm: $\alpha \otimes_{\text{prod}} \beta = \alpha\beta$. The *fuzzy support* of an itemset $V = \{'x_1 = v_1' \ldots 'x_m = v_m'\}$ is

$$supp(V) = \sum_{R \in D} \mu_{v_1}(r_1) \otimes \ldots \otimes \mu_{v_m}(r_m), \quad (8)$$

where $r_1 \ldots r_m$ are original numerical values for $x_1 \ldots x_m$ in R. Fuzzy ARM can then run using fuzzy instead of crisp supports for all itemsets in computations of RIM values.

2.3 DE

EAs are metaheuristics that mimic Darwinian evolution for solving optimization problems, and DE is one of the best performing EA techniques due to its convergence characteristics and small number of model parameters [31]. In DE, each *individual* is a vector of variables representing a candidate of entire or part of solution to an optimization problem. Each individual has a *fitness value*, denoted as *fval*, computed from one or multiple objective functions for measuring the goodness of the solution it represents. DE starts with an initial *population* of N individuals and continues for G generations. In each generation, three key operators are applied to evolve the population toward better solutions:

- *Mutation*: to create *mutant vectors V* by perturbing an individual with the difference of other individuals. A classical and popular approach of mutation utilizes three different randomly selected individuals: for generation t,

  $$V_i^t = X_a^t + F\left(X_b^t - X_c^t\right),\ i = 1 \ldots N, \quad (9)$$

  where F is the mutation scale, X represents individuals, $a, b, c \in \{1 \ldots N\}$ are distinct random indices.
- *Crossover*: to recombine individuals and mutant vectors into trial vectors U. The most popular approach is binomial crossover:

  $$u_{j,i}^t = \begin{cases} v_{j,i}^t & \text{if } rand_i[0,\ 1] \le Cr \text{ or } j = j_{rand} \\ x_{j,i}^t & \text{otherwise} \end{cases}, \quad (10)$$

Where *Cr is the crossover rate*, $x_{j,i}^{t}, u_{j,i}^{t}$ and $v_{j,i}^{t}$ are *j*-th variables in X_i^t, U_i^t and V_i^t, and j_{rand} is a random index of variables in an individual to ensure that the trial vector includes at least one variable from the mutant vector.

- *Selection*: to determine which one in each pair of parent individual and trial vector will survive to the next generation *t*+1, according to which vector has a better fitness. If the objective function(s) is to be maximized, then

$$X_i^{t+1} = \begin{cases} U_i^t & \text{if } fval\left(U_i^t\right) \geq fval\left(X_i^t\right) \\ X_i^t & \text{otherwise} \end{cases} \text{[32].} \tag{11}$$

2.4 EAs for fuzzy ARM

Facing the uncertainty in data discretization, ARM has employed techniques such as clustering to optimize data discretization schemes for individual attributes [33, 34]. However, such optimization is based on data distribution of individual attributes, and the result can be quite different from optimal combination of discretization on multiple attributes that leads to good rules. This problem is promisingly to be resolved by EAs which have the power to address more complicated optimization problems.

EAs have been used with ordinary and fuzzy ARM and achieved notable enhancement on the number of rules and fitness of RIM values. In EA-based ARM, individuals can be either entire data discretization schemes or individual rules. Membership functions for items in the rules may be either predefined or encoded and optimized, and may have known shapes and other constraints. Most objective functions are about numbers, RIM values and diversity of resultant rules. To date, almost all EA-based fuzzy ARM studies, such as [1-5], have taken GA approaches. MODENAR [31], the only DE algorithm for numerical ARM, is for ordinary rule mining. Experiment results of this study, however, reveal that DE has certain merits over GA for mining statistically significant association rules.

3. DE for mining significant fuzzy association rules (DESigFAR)

This section presents the proposed DESigFAR algorithm. Section 3.1 describes the individual encoding. Section 3.2 illustrates the fitness assignment on individuals with two new statistical testing approaches, the experimentwise and generationwise adjustment, for controlling the risk of spurious rules. Section 3.3 gives the algorithm structure and designs of key evolutionary operators.

3.1 Individual encoding

The algorithm uses each individual (parameter vector) to encode a *main rule* [1] as a part of candidate resultant rules. A main rule is a collection of rules with the same attributes in the antecedents and same in the consequents. All rules like $a_1 = l_{a_1 i_1} \wedge \ldots \wedge a_q = l_{a_q i_q} \rightarrow b = l_{bj}$ is under the main rule *M*: $a_1 \wedge \ldots \wedge a_q \rightarrow b$, where $a_1 \ldots a_p b$ are attributes with corresponding concepts $l_{a_1 i_1} \ldots l_{a_q i_q} l_{bj}$.

While the proposed method applies to ARM with all kinds of fuzzy data discretization models, we suggest a specific model proposed in [30] (Fig. 2). This model is Gaussian-curve-based, meaning that the concept *transitions* (intervals where $0 < \mu_l(x) < 1$) in the model are Gaussian curves. The standard deviation of the Gaussian transition curve between interval (*a*, *c*) is (*c*-*a*)/2.473, so that $\int_a^c \mu_l(x)\,dx$ = (*c*-*a*)/2, which appears unbiased for modelling *l*. The model has been justified as well representing the fuzzy membership of numerical data to linguistic concepts and more robust against data noises than triangular and trapezoidal models (see Fig. 1). For each attribute *a*, three groups of variables are encoded to define a main rule:

- k_a: the number of concepts $l_{a1} \ldots l_{ak_a}$ for *a*: $k_a \le k_{\max}$, $k_{\max}$ is the predefined maximum number of concepts for any attribute;
- cr_{ai_L}, cr_{ai_R}: left and right endpoints of $core(\mu_{l_{ai}}), i = 1 \ldots k_{\max}$. These variables are specific to the recommended data discretization model. For other models, variables that can fully define the concepts can be used instead without affecting the application of DESigFAR;

- loc_a : the location of items involving *a* in rules; $loc_a = 1, 2, 0$ if the items are in the rule antecedent, rule consequent and neither.

The encoding of *a* is

$$k_a\ cr_{a1\_\mathrm{R}}\ cr_{a2\_\mathrm{L}}\ cr_{a2\_\mathrm{R}} \dots cr_{a(k_{\max}-1)\_\mathrm{L}}\ cr_{a(k_{\max}-1)\_\mathrm{R}}\ cr_{ak_{\max}\_\mathrm{L}}\ loc_a, \tag{12}$$

$cr_{ak_a\_\mathrm{R}} \dots cr_{ak_{\max}\_\mathrm{L}}$ are assigned empty values. The entire vector for all *n* attributes in data, with a length of $n(k_{\max}+2)$, is

$$k_1\ cr_{11\_\mathrm{R}}\ cr_{12\_\mathrm{L}}\ cr_{12\_\mathrm{R}} \dots cr_{1k_{\max}\_\mathrm{L}}\ loc_1 \dots k_n\ cr_{n1\_\mathrm{R}}\ cr_{n2\_\mathrm{L}}\ cr_{n2\_\mathrm{R}} \dots cr_{nk_{\max}\_\mathrm{L}} \dots loc_n. \tag{13}$$

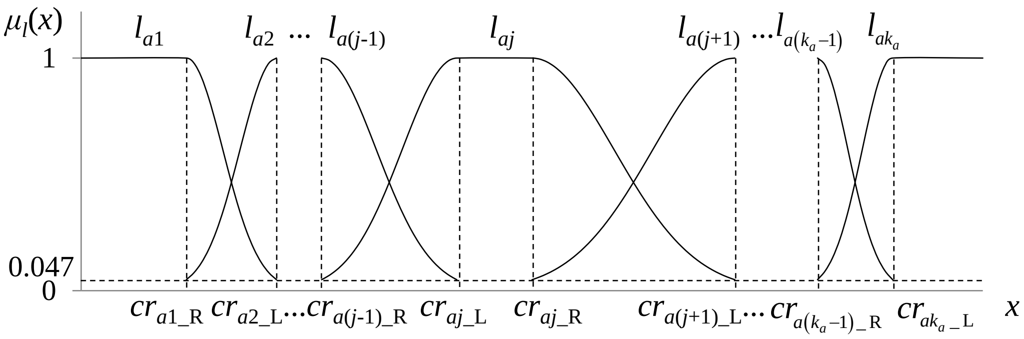

Fig. 2. Recommended data discretization model of attribute *a*.

As DESigFAR calls for RIM evaluation and statistical testing on every candidate rule, encoding individuals as main rules is much more efficient than as entire data discretization schemes. With $k_{\max}$ = 5 and up to 4 items in rule antecedents, data with modest numbers of attributes typically constitute at least 10^5–10^{10} potential rules [10, 11], while a main rule contains only 2^2–5^5 = 4–3125 rules. Thus, encoding entire data discretization schemes may require hundreds to millions of times more rules evaluated and time consumed than encoding main rules.

The main rule encoding also enables more flexible resultant rules. All rules under a main rule concerning the same attribute groups in antecedents and in consequents follow the same discretization scheme, thereby avoiding the confusion due to inconsistent concept definitions and maintaining reasonable interpretability of these rules. Meanwhile, different main rules may follow different data discretization schemes. Thus, when interacting with different groups of other attributes, an attribute may have variant optimal intervals of original numerical data values for concepts such as 'high' and 'low'. This is also reasonable and can lead to better RIM values than encoding entire data discretization schemes and using one scheme for all resultant rules.

3.2 Fitness assignment with statistically sound tests

The proposed DE may be used to optimize various RIMs and work with different statistical tests on rules. The objective function *fval* for computing the fitness value for each individual (main rule) *M* depends on the objective RIM:

- For RIMs based on extra support of a rule, compared with that if items in the rule are independent with part of or all other items, such as leverage [35]:

  $$lev(X \to y) = supp(X \to y) - supp(X)supp(y)/|D|, \tag{14}$$

  $fval(M)$ is equal to summed RIM value of all significant rules under *M* that pass the statistical test and meet other user specified constraint(s) φ. We call such rules *eligible rules*;
- For RIMs evaluating higher occurrence probabilities of a rule, compared with that if items in the rule are independent, such as confidence and improvement: $fval(M)$ is equal to the average RIM value of all eligible rules.

It is optional but common for ARM to include φ in addition to the target RIM for preliminary filtering of uninteresting rules. The commonest φ is the minimum rule support. It is also usual to consider only rules whose target RIM values suggest positive associations, for example, to specify φ as leverage > 0 when $fval(M)$ is summed leverage. Since all rules under a certain main rule shall not repeat, if multiple individuals encode the same main rule (have the same $k_1 \dots k_n$ and $loc_1 \dots loc_n$), only the one with the highest *fval* value remains unchanged. Other individuals are reset to *fval* = 0, and rules under them will not enter ARM results.

To answer different user needs for balancing the abundance of significant rules and risk of spurious rules, we propose two approaches to adjust significance levels of statistical tests on rules, resolve the multiple comparisons problem, and strictly control spurious rules in the DE. Suppose that rules under main rule M: $a_1 \wedge \ldots \wedge a_q \rightarrow b$ with numbers of concepts $k_{a_1} \ldots k_{a_q}\ k_b$ are tested.

(1) The *experimentwise adjustment* approach aims at limiting the FWER in entire DE to no more than user specified level α, say 0.05. The significance level is adjusted to

$$\kappa = \alpha \Big/ \left(2GN \times \prod_{i=1}^{q} k_{ai} \times k_b \right). \tag{15}$$

In each generation, rules contained in $2N$ individuals, including N parents and N trial vectors, are evaluated. Eq. (15) applies three-level Bonferroni corrections to α: first to limit the risk of having any spurious rules in each generation to at most α/G, then to limit such risk for each individual in a generation to no more than $\alpha/2GN$, and finally to share the risk for each individual among all rules under it. Alternatively, slightly more rules may be discovered by using Holm procedure [36] to replace the last Bonferroni correction. That is, to rank p values of the tests on all rules ascendingly from p_1, and accept such rules corresponding to $p_1 \ldots p_i$ that

$$\forall 1 \leq j \leq i, p_j \leq \alpha \Big/ \left(2GN \times \left(\prod_{i=1}^{q} k_{ai} \times k_b - j + 1 \right) \right). \tag{16}$$

Eqs. (15) and (16) are multi-level extensions to statistically sound tests in conventional ARM [10] and hold the same logic as the latter to adjust significance levels of the tests by the search space size instead of the number of pre-filtered and tested rules. Thus, Eqs. (15) and (16) should be able to strictly control the FWER upon α like the existing statistically sound evaluation.

(2) The *generationwise adjustment* approach aims at limiting the percentage of spurious rules to no more than α. Using purely Bonferroni corrections, the adjusted significance level is

$$\kappa = \alpha \Big/ \left(2N \times \prod_{i=1}^{q} k_{ai} \times k_b \right). \tag{17}$$

If the Holm procedure is adopted, j eligible rules with the smallest $p_1 \leq \ldots \leq p_i$ values in the tests will be accepted, if

$$\forall 1 \leq j \leq i, p_j \leq \alpha \Big/ \left(2N \times \left(\prod_{i=1}^{q} k_{ai} \times k_b - j + 1 \right) \right). \tag{18}$$

Eqs. (17) and (18) restrict the probability of accepting any spurious rules in each generation to at most α. Thus, no more than $\alpha \times 100\%$ generations are expected to generate spurious rules. As spurious rules occur purely by chance, the expected number of new rules discovered in a generation is independent of occurrences of spurious rules in the generation. Therefore, even all newly accepted rules in a generation are spurious if any of them are, the expected percentage of spurious rules in ARM results is still no more than α, and should be often far below α, since the above worst case is unusual.

The generationwise adjusted test has a much higher significance level, about G times of that under the experimentwise approach, and thus may accept considerably more rules than the latter. The generationwise approach cannot maintain a minimum FWER, but its control over the percentage of spurious rules is much more effective than unadjusted tests with raw significance level α, since the latter usually results in much more than $\alpha \times 100\%$ spurious rules [10, 11]. Users may choose the appropriate approach considering the benefit of discovering more rules and acceptable hazard of spurious rules for specific ARM tasks.

DESigFAR uses the *crisp-fuzzy* strategy [30] for mining significant rules: the statistical tests use crisp supports of involved patterns, while RIM and *fval* evaluation uses their fuzzy supports. The acceptance or rejection of rules by statistical tests is qualitative and does not concern accurate depictions of fuzzy transitions between concepts, thus the tests can control spurious rules by using the crisp supports as effectively as using fuzzy supports. Further, testing with crisp supports usually results in larger numbers of significant rules.

The crisp supports should hold the same concepts of maximum membership degrees as in the fuzzy data discretization model for computing RIM values. For the recommended Gaussian-curve-based model, each record adds 1 to $supp\left(a = l_{aj}\right)$ if the original a value is in $\left[\left(cr_{a(j-1)\_R} + cr_{aj\_L}\right)/2,\ \left(cr_{aj\_R} + cr_{a(j+1)\_L}\right)/2\right)$, and 0 otherwise.

3.3 Evolutionary model

The DESigFAR algorithm is overviewed in Fig. 3. Considerations on common DE operators are detailed below, and specific techniques in the algorithm are presented in Sections 3.3.1–3.3.3.

- Population initialization: values of variables for each individual can be generated as random numbers within their valid ranges. Alternatively, the core endpoints can be generated based on classification methods such as equisize classification plus random numbers.
- Mutation: variable values in the produced mutant variables may be invalid in that, for example, the values fall outside their ranges, or $cr_{\mathrm{L}} > cr_{\mathrm{R}}$ for some concepts. Thus, the mutation includes a repair to adjust those invalid values to valid ones, using the same method as the DE-based ARM algorithm MODENAR [31].
- Crossover: binomial crossover is used by regarding all variables encoding an attribute, that is, a chromosome section in Eq. (12) as a single crossover unit u in Eq. (10). Pilot experiment has shown that this approach produces better results than using smaller crossover units, as smaller crossover units can make trial vectors contain many invalid variables and require repair again, which will disturb the evolution.
- Selection: in the competition between each parent and trial vector pair, apart from the selection rule in Eq. (11), if the pair of vectors both have positive fitness values, the algorithm first tries to find a trial vector with $fval$ = 0 (mostly because there is another individual for the same main rule with better RIM values) and let the parent individual under concern replace it and survive. The one in the pair with lower fitness will be

discarded only if such a trial vector is unavailable. This strategy utilizes the 'empty places' of individuals with zero fitness and better preserves good main rule encoding.

```
Input:   population size N, No. of generations G, mutation scale F,
         crossover fraction Cr, generation jumping rate Jr
Output: eligible rules from optimized individuals

Initialize population P0
For t = 0, 1…G − 1
  Generate a random number num between 0 and 1
  If num < Jr
    Perform opposition based generation jumping
  Else
    Perform mutation, get mutant vectors V1V2…VN
    Perform ftmin repair on V1V2…VN
    Perform crossover on individuals M1M2…MN with V1V2…VN, get trial vectors U1U2…UN
    For i = 1…N
      Test all rules in Ui (generationwise/experimentwise), get eligible rules
      fval(Ui) = ∑lev(r), mean(imp(r)), etc. of all eligible rules r in Ui
    End For
    For i = 1…N
      If ∃ Mj or Uj, j = 1…N represents the same main rule as Mi or Ui and has a larger fval value
        fval(Mi) = 0 or fval(Ui) = 0
      End If
    End For
    For i = 1…N
      If fval(Mi) > fval(Ui)
        Add Mi into Pt+1
      Else
        Add Ui into Pt+1
      End If
    End For
  End If
End For
Return eligible rules in all individuals with fval > 0
```

Fig. 3. Overall procedures of DESigFAR.

3.3.1 Generation jumping

DESigFAR also incorporates opposition based generation jumping [37] to avoid being trapped in local optima. Each generation in the DE has a probability *Jr* (*Jr* ≤ 0.04) to generate an opposite population *OP* from current population *P*, and *N* individuals with the best *fval* values in *OP* ∪ *P* are selected. In existing literature, *OP* is generated by replacing each variable *x* within range [a, b] in each individual by its opposite number $\breve{x}:\breve{x}=a+b-x$. To accommodate highly skewed data, in this study *OP* is generated based on ranks of data values instead:

$$\breve{x}=rank^{-1}\left(a+b-rank\left(x\right)\right), \tag{19}$$

where *rank*(*x*) is the rank of *x* among all data values of the attribute it is in, and $rank^{-1}(r)$ is the data value with rank *r*.

3.3.2 Maintaining fuzziness of concepts

Crisp data discretization generates binary membership degrees that are more contrasting than fuzzy ones and thus usually overestimates RIMs on the strength of data associations. By continuously searching for fuzzy membership functions that lead to higher RIM values, the DE tends to end up with near-crisp concepts with very narrow transitions and suffer from inaccurate RIM values like ordinary ARM. To avoid this situation, the *fraction of transition*, *ft* is defined to measure the fuzziness of concepts with core [cr_{L}, cr_{R}] and base [*a*, *b*]:

$$ft=1-\left(cr_{\mathrm{R}}-cr_{\mathrm{L}}\right)/\left(b-a\right). \tag{20}$$

The use of *ft* is in line with the widely used fuzziness measure of fuzzy sets [38]:

$$fuzziness=1-\frac{1}{\left(b-a\right)^{1/p}}\left[\int_a^b\left|2\mu\left(x\right)-1\right|^p dx\right]^{1/p}, \tag{21}$$

where μ is the membership function. $\int_a^b\left|2\mu\left(x\right)-1\right|dx$ is equal to the area under $\left|2\mu\left(x\right)-1\right|$ curve (Fig. 4). When *p*=1, *fuzziness* is equal to 0.4095*ft* for the suggested Gaussian-curve-based data discretization model and 0.5*ft* for trapezoidal model. Instead of using *fuzziness*, DESigFAR uses *ft* which is simpler for users to interpret and set a minimum threshold for.

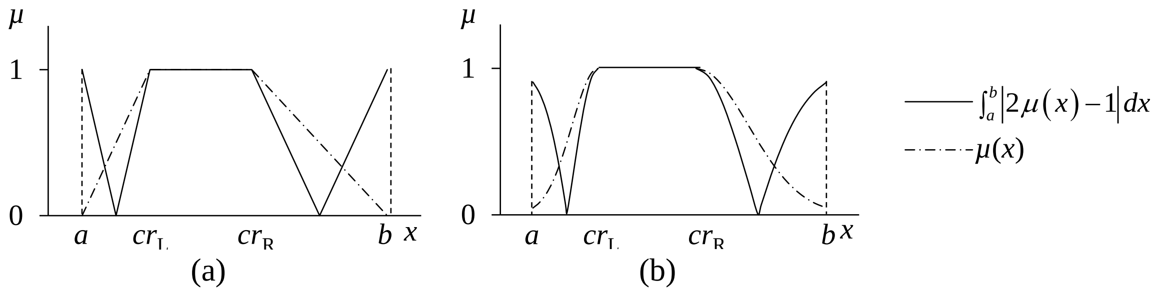

Fig. 4. Relation between *fuzziness* and (a) trapezoidal and (b) Gaussian-curve-based membership functions.

After mutation, a mutant vector survives only if all concepts in it fulfil the user specified minimum *ft*, $ft_{\min}$. To avoid losing favorable mutant vectors, DESigFAR first tries to repair rather than discard unqualified mutant vectors. For, say, the left transition of a problematic concept, *a* and cr_L are respectively decreased and increased by equal magnitude to make $ft = ft_{\min}$. The repair succeeds if it does not conflict other concept cores or the attribute value range. Over 95% repairs succeeded in the experiments of this study.

3.3.3 Sampling strategy for speeding up algorithm

For large datasets, the main computation overhead of fuzzy ARM, including DESigFAR, usually lies in fuzzy data discretization. To improve the algorithm scalability, for data containing tens of thousands or more records, the proposed DE uses randomly sampled data records for fuzzy data discretization during RIM evaluation. The exact RIM values are recomputed once using the full data by the end of the DE. The necessary sample size mainly depends on the number of data attributes and data distributions and should increase much slower than that of datasize. Fuzzy data discretization and the sampling strategy are applied in and affect only the RIM evaluation: due to its crisp-fuzzy approach, DESigFAR performs the much faster crisp discretization on compressed data in the statistical test stage. Experimental results in Section 4.4 show that the sampling has minimal effect on the goodness of RIM values obtained by the algorithm.

4. Experiments: Hotel room pricing and wildfire risk factors

This section presents two experiments for DESigFAR: Hotel experiment on smaller-sized data for investigating impacts of hotel accessibilities (nearness) to tourism resources on hotel room prices in Hong Kong, and Fire experiment on larger-sized data for studying relations between topographical variables and wildfire risks in Colorado, US. Sections 4.1–4.2 describe the experiment data and specifications. Sections 4.3–4.4 evaluate the efficacy of DESigFAR in controlling spurious rules and discovering true rules, respectively, as compared with existing ARM methods. Section 4.5 presents the computational performance of the algorithms, and Sections 4.6–4.7 discuss the practical implications of Hotel and Fire experiment results improved by the proposed DE.

4.1 Data and preprocessing

4.1.1 Hotel experiment

The study area, metropolis Hong Kong in southern China, is a world's leading financial center and tourism destination. Landmark scenic spots and luxury hotels concentrate in the city downtown around the Victoria Harbour. Midweek prices of the cheapest double rooms were acquired on 1 April 2015 from Agoda, the online hotel agency including the largest number of Hong Kong hotels. Prices three and seven weeks before check-in date were collected and averaged to balance the effects of offering discounted room rates. Accessibilities to various tourism resources from hotels, represented by walkable road network distances, are summarized in Table 1. The distances were measured from Google Maps using JavaScript codes and manual interventions for quality control. Multiple economic hotels in the same building were merged into one record with average price of their rooms weighted by numbers of rooms. These hotels are of homogeneous resource accessibilities, conditions and room prices; such highly correlated subjects should be merged and treated as one for statistical tests in ARM and conventional regressive price modelling, which both assume mutual independence between studied subjects. The preprocessed data contained 290 records covering around 68,000 rooms (83% of total rooms in Hong Kong by December 2014 [40]). The hotels and selected resources are mapped in Fig. 5.

Table 1 Accessibility attributes in Hotel experiment.

	Name	Description
1–5	dist_topspot1–dist_topspot5	Distance to 1st–5th nearest ‘top 10 attractions’ receiving most visitors [39], major city parks and theme parks
6	dist_museum	Distance to nearest museums[a]
7	dist_worship	Distance to nearest worship places, e.g. temples, churches
8	dist_beach	Distance to nearest beaches[a]
9–13	dist_shop1–dist_shop5	Distance to 1st–5th nearest multi-storey shopping places
14	dist_subway	Distance to nearest subway station entrances
15–19	dist_bus1–dist_bus5	Distance to 1st–5th nearest bus stops

[a] The most significant 30 museums, 30 worship places and 10 beaches highlighted by HKTB

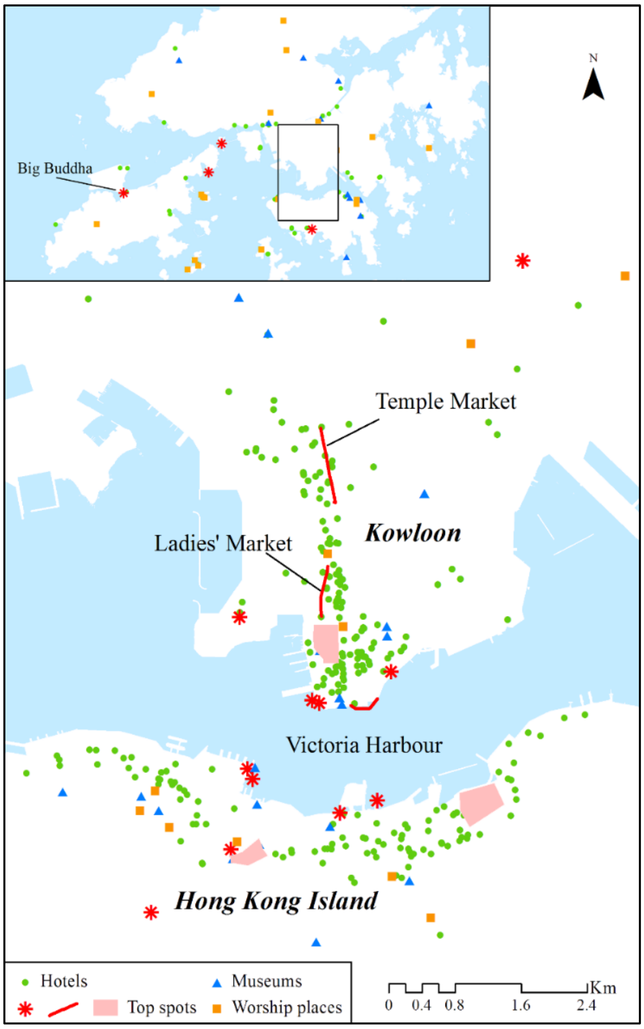

Fig. 5. Hong Kong map with experimented hotels and selective resources.

4.1.2 Fire experiment

This experiment utilizes the Covertype dataset from the UCI Machine Learning Repository [41]. The dataset covered four wilderness areas in the wildfire-prone Colorado Front Range, US. The experiment used the data for Rawah area containing 260,796 records, with each record representing a 30×30m cell of land. The data contained eight topographical attributes serving as wildfire risk factors, as listed in Table 2, and attribute h_dist_fire for horizontal distance to the nearest past wildfire ignition point. To evaluate the robustness of DESigFAR against datasize variations, the experiment was performed on two random samples of Fire data containing 2500 and 20,000 records as well as the full data, later referred to as FireS, FireM and FireL datasets, respectively.

Table 2 Wildfire risk factor attributes in Fire experiment.

	Factor	Description	Values suggesting high fire risk
1	elevation	-	Lower elevation in areas with elevation > 1600m (case of study area) [42, 45, 46]
2	slope	-	Steep slopes [43, 44, 46]
3–4	h_dist_water, v_dist_water	Horizontal/vertical distance to nearest surface water	Proximity to water can reduce fire risks, but might also increase forest density and thus fire risk [47]
5	h_dist_road	Horizontal distance to nearest road	Proximity to roads [42–45]
6–8	hillshade_9am/ 12nn/3pm	Summer hillshade index at 9am/noon/ 3pm	High index (radiation) especially at pm increases fire risk, but high am/noon values may imply east slopes/flat terrain with lower risks [43, 44, 46]

4.2 Experiment specifications

DESigFAR was implemented and run on all datasets using leverage as the optimization objective, leverage > 0 as the user constraint φ and a minimum support of 0.02 times the datasize. The statistical test applied was chi-square test for productive rules, with user specified maximum risk of spurious rules α = 0.05 and significance levels adjusted by Eqs. (16) and (18). The maximum number of concepts in an attribute was set at $k_{\max}$ = 5, as rules with more concepts had small supports and hardly survived early generations of the DE. Other specifications are listed in Table 3. N, F and Cr values were such determined as to achieve more efficient evolutions (smaller $G \times N$ values). As datasize increased and rules enriched, the DE favored larger F values to more actively search for alternative rules, smaller Cr values to maintain combinations of good discretization for different attributes, and smaller Jr values as the generation jumping became less useful. Following this principle, users may conduct fast pilots on samples of data to determine appropriate parameter values. The G value was large enough for the DE to converge to a certain extent: the increase in total leverage of all significant rules slowed down to only around 3% during the last 1/4 generations, and was much slower (typically by one half for the next 1/4 generations) if the algorithm continued running. Each experiment group of the same specifications (called a *treatment*) was applied for 25 times (*runs*) to produce average results, unless stated otherwise.

Table 3 Experiment specifications.

	Hotel	Fire
Form of rules (Max. 4 items in antecedent)	resource accessibility(ies) $\rightarrow$ room price	fire risk factor(s) $\rightarrow$ h_dist_fire
Population size N	80	175
No. of generations G	2000	1000/700/300 (FireS/M/L)
Crossover fraction Cr	0.5	0.2
Mutation scale F	0.5	0.7/0.8/1.0 (FireS/M/L)
Generation jumping rate Jr	0.04	0.02/0/0 (FireS/M/L)
Population initialization	Based on equisize classification (see Sect. 3.3)	

4.3 Assessing control over spurious rules

DESigFAR was first examined for its ability in controlling spurious rules in comparison to conventional statistical tests without adjustments to the significance levels. Because authentic and spurious rules are unknown in real-world data, known spurious rules needed to be artificially introduced. In each run, six out of the 19 accessibility attributes in Hotel data, and three out of the eight topographical attributes in Fire data were randomly selected, and values in them randomly reordered, making these attributes ‘irrelevant’ to any data associations. Rules involving these attributes, termed *irrelevant rules*, should be spurious.

For Hotel data, both generationwise and experimentwise adjusted tests were experimented with $ft_{\min}$ = 0.3, 0.5 and 0.7, triangular, trapezoidal and Gaussian-curve-based fuzzy membership functions (see Fig. 1). Each treatment was paired with two control treatments taking traditional unadjusted statistical test (with $\kappa = \alpha = 0.05$) in crisp-fuzzy and conventional fuzzy approach, respectively. For the conventional fuzzy treatments, p values of the rules were computed using fuzzy pattern supports. Fire data were experimented with $ft_{\min}$ = 0.5 only, as the Hotel experiment result turned out to show robust efficacy of DESigFAR in controlling spurious rules with various $ft_{\min}$ values.

Table 4 lists the results for DESigFAR with generationwise adjusted test and for unadjusted test in conventional fuzzy approach. ‘Significant’ and ‘irrelevant’ respectively refer to numbers of significant and irrelevant rules. The unadjusted test in crisp-fuzzy approach accepted much larger numbers of irrelevant rules than that in conventional fuzzy approach and obviously failed to control spurious rules, which agreed to past study results on unadjusted tests for non-EA ordinary ARM [10, 11]. DESigFAR in generationwise approach well controlled the percentage of spurious rules below the user specified level. This approach resulted in fewer than 1.5% irrelevant rules for all datasets and data discretization models, far below the 5% upper limit as user specified by setting $\alpha = 0.05$. The percentages of spurious rules became even lower as datasize increased, from over 1% for Hotel data to 0.2% for FireL. Because the generationwise approach ensures that spurious rules arise from no more than $\alpha \times 100\%$ of generations, with richer data and more rules discovered, spurious rules are likely a smaller part of rules accepted in these generations, and this approach becomes more powerful in controlling spurious rules.

Table 4 Result on control over spurious rules, with generationwise approach for DESigFAR.

Data	Discretization model	$ft_{\min}$	DESigFAR (crisp-fuzzy), generationwise		Conv. fuzzy [a], unadjusted test	
			Significant	Irrelevant	Significant	Irrelevant
Hotel	Tri. [a]	0.3	36.0	0.4	146.9	9.4
		0.5	34.2	0.5	143.6	10.9
		0.7	31.2	0.3	140.9	4.4
		Average	33.8	0.4 **(1.1%)**	143.8	8.2 **(5.7%)**
	Trapez.	0.3	36.3	0.6	176.6	13.8
		0.5	35.3	0.6	166.4	10.3
		0.7	32.8	0.4	151.0	13.2
		Average	34.8	0.5 **(1.4%)**	164.7	12.5 **(7.6%)**
	Gaus.	0.3	38.8	0.7	180.9	16.0
		0.5	33.5	0.7	176.2	12.0
		0.7	32.8	0.1	158.4	7.9
		Average	35.0	0.5 **(1.4%)**	171.9	12.0 **(7.0%)**
FireS	Tri.	0.5	35.6	0.2 **(0.7%)**	137.4	63.7 **(46.4%)**
	Trapez.		40.4	0.6 **(1.4%)**	182.3	95.6 **(52.5%)**
	Gaus.		38.1	0.3 **(0.8%)**	190.6	102.2 **(53.6%)**
FireM	Tri.	0.5	84.0	0.2 **(0.2%)**	187.4	61.2 **(32.7%)**
	Trapez.		87.3	0.6 **(0.6%)**	236.2	98.2 **(41.6%)**
	Gaus.		89.8	0.3 **(0.4%)**	247.8	06.7 **(43.1%)**
FireL-sampled	Tri.	0.5	154.8	0.2 **(0.2%)**	-[b]	-
	Trapez.		161.6	0.4 **(0.2%)**	-	-
	Gaus.		157.0	0.2 **(0.2%)**	-	-

[a] Conv. fuzzy: conventional fuzzy, Tri.: triangular, trapez.: trapezoidal, Gaus.: Gaussian-curve-based; same in later tables and figures

[b] Pilot runs produced >>5% false rules like on FireS and FireM; stopped due to long run time

To evaluate the control over FWER by DESigFAR in the experimentwise approach, the p value of each irrelevant rule generated in the generationwise experiment were compared with the experimentwise adjusted significance level κ computed by Eq. (16). If the p value was smaller than κ, the rule would have been a spurious rule if an experimentwise test had been used. Out of the results for all datasets, only the FireS result contained one rule with $p/\kappa < 1$. Even if we consider any irrelevant rules with $p/\kappa < 50$ might have a risk to be accepted by the end of the evolution, only 2 runs (0.9% of all runs) for Hotel data, 3 runs (3.9%) for FireS, and 0 run for other datasets contained rules with $p/\kappa < 50$. Thus, the experimentwise approach should be able to control the FWER well below 5% as user specified.

The unadjusted test, even under the more conservative conventional fuzzy approach, failed to control spurious rules at 5%. For Fire data, 1/3–1/2 rules accepted by the unadjusted test were irrelevant ones (Table 4). The rule mining results containing so many spurious rules indistinguishable with authentic ones can be considered useless. For Hotel data, the percentages of spurious rules still exceeded the 5% user tolerance, even though they were smaller than those for Fire data. The test actually generated far more than 10% irrelevant rules at early generations of the evolution. As these irrelevant rules arose from random data and were expected to have small leverages, many of them were phased out later when competing with rules without irrelevant attributes. For Fire data, with a larger population size and more individuals to accommodate a larger number of significant rules, irrelevant rules had lower chance to be phased out. If a much larger population size is used for Hotel data, the results will also contain extremely high percentages of spurious rules.

To sum up, experiments show that the proposed statistical tests for DESigFAR are necessary and capable in controlling spurious rules. While the unadjusted test cannot control spurious rules in DE, the generationwise and experimentwise approaches can strictly control the percentage of spurious rules and FWER below user specified level α, respectively, and are more effective for larger datasets.

4.4 Evaluating ability of discovering significant rules

The ability of DESigFAR in discovering significant rules was evaluated on original data without artificial irrelevant attributes. For FireL data, DESigFAR was run with both the full dataset and a random sample of 40,000 records for the fitness evaluation, the latter for evaluating the sampling strategy for speeding up the algorithm. Fig. 6 and Fig. 7 show the results of Hotel and Fire experiment, respectively. FireL results obtained with and without the sampling strategy are compared in Table 5. The control treatments using the full data for fitness evaluation were conducted for only five runs per treatment, due to their relatively long run time and the fact that they were experimented mainly to show the similarity of their results to the result of DESigFAR (with crisp-fuzzy and sampling strategies).

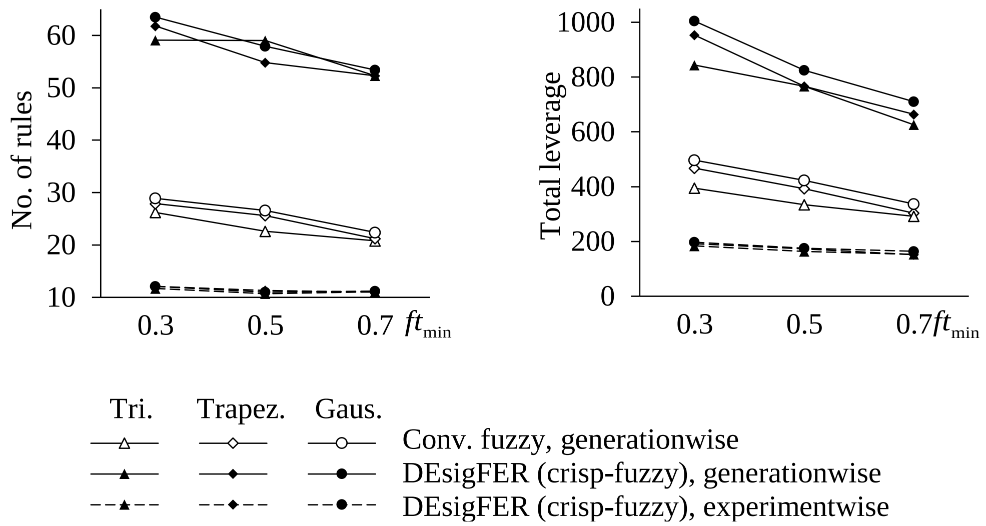

Fig. 6. Result in discovering significant rules: Hotel experiment.

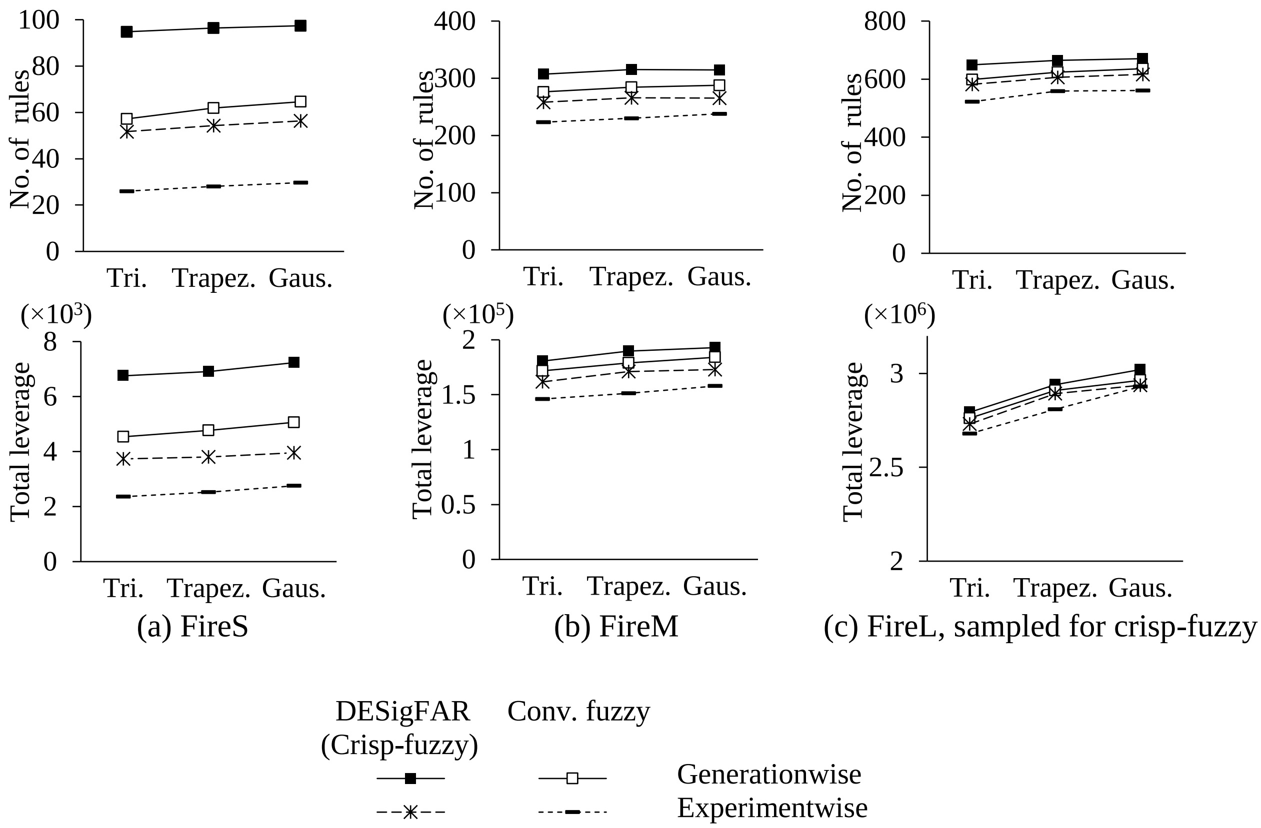

Fig. 7. Result in discovering significant rules: Fire experiment, $ft_{\min} = 0.5$.

Table 5 FireL results obtained with and without the sampling strategy.

	Discretization model	No. of rules		Total leverage ($\times 10^6$)	
		Full data	Sampled	Full data	Sampled
Generationwise	Tri.	649.2	638.0 (-1.7%)	2.82	2.79 (-0.9%)
	Trapez.	664.2	675.4 (+1.7%)	2.93	2.94 (+0.2%)
	Gaus.	670.4	662.9 (-1.1%)	3.04	3.02 (-0.5%)
Experimentwise	Tri.	583.8	582.1 (-0.3%)	2.74	2.73 (-0.3%)
	Trapez.	610	606.6 (-0.6%)	2.90	2.89 (-0.2%)
	Gaus.	601.2	616.3 (+2.5%)	2.95	2.93 (-0.5%)

Used with various forms of membership functions and $ft_{\min}$ values, the proposed DESigFAR incorporating crisp-fuzzy ARM consistently obtained more rules and larger total leverages of these rules than the conventional fuzzy approach (Fig. 6, Fig. 7), which reconfirmed the merit of the crisp-fuzzy strategy in finding more abundant rules revealed in [30]. As the datasize increased, such superiority of DESigFAR lessened (Fig. 7a-c), but its advantage in computational efficiency over the conventional fuzzy approach amplified, as will be shown in Section 4.6. FireL results obtained with full and sampled data for RIM evaluation were quite similar (Table 5), suggesting that the sampling strategy is unlikely to compromise the quality of results obtained by the proposed DE.

It should be acknowledged that the experimentwise test can be overconservative for small datasets: experimentwise DESigFAR only discovered 11–12 significant rules on average from Hotel data, and its conventional fuzzy control treatments only resulted in 1–2 rules, which were too few to be plotted on Fig. 6. Yet the experimentwise approach seems not very meaningful for such small data with only dozens of significant rules, as the generationwise approach can already limit the expected number of spurious rules to very few. The difference between the experimentwise and generationwise tests quickly diminished with increasing datasize. The total leverage resulted from the two approaches differed by only 2.3% for FireL data. Thus, for datasets with hundreds or more rules expected, the experimentwise approach is appropriate and can give good results if the specific ARM applications requires a very strict control over spurious rules.

Gaussian-curve-based or trapezoidal membership functions resulted in more rules and better RIM values than alternative experiment settings (Figs. 6, 7). Their advantage over the triangular function should be attributed to their cores of arbitrary sizes which enable more flexible search for optimal numerical data intervals of the concepts. Smaller $ft_{\min}$ values were also found beneficial for discovering more rules (Fig. 6), as they were less likely to cause substantial reduction in rule leverage values in the $ft_{\min}$ repair operation or make the repair fail and the individual discarded. As DESigFAR uses crisp supports in statistical tests on the rules which are independent of the $ft_{\min}$ value , the discovery of smaller numbers of rules with larger $ft_{\min}$ values should be a delay instead of a defect in the evolution, and can be made up by running the algorithm for more generations. The decrease of total leverages with increasing $ft_{\min}$ values should be due to both fewer rules discovered and fuzzier concepts.

4.4.1 Comparison with non-evolutionary and GA-based ARM

As baselines for evaluating DESigFAR, traditional non-evolutionary ARM with prespecified data discretization and GA-based ARM were also run on Hotel and Fire data, with statistical tests matching the proposed experimentwise and generationwise tests to filter out spurious rules. Comparisons between the results of DESigFAR, non-evolutionary ARM and GA-based ARM show that DESigFAR has marked advantages over the other two methods in terms of discovering larger number or more informative rules and obtaining better RIM values.

(1) Non-evolutionary ARM

The Hotel and Fire data were first divided into 2–5 concepts for each attribute by classical data discretization techniques for ARM: equisize classification, K-means clustering and agglomerative clustering, using the scikit-learn toolkit [48]. Then the discretized datasets with 2–5 concepts in all attributes were explored for rules by the KORD algorithm [49]. For each clustering algorithm, KORD were run on two more discretized datasets where each attribute had the number of concepts (clusters) that gave the clustering results the smallest

Davies-Bouldin Index [50] and largest Silhouette Coefficient [51], both suggesting better separation of the clusters and supposedly better data discretization. Existing statistically sound test with Holm procedure [10] for limiting the FWER upon 5%, and the Benjamini-Hochberg-Yekutieli procedure [52] for limiting the percentage of spurious rules upon 5% were applied for productive rules. These two statistical procedures, though inapplicable to EAs, have equal effects on controlling spurious rules to Eqs. (16) and (18) for the experimentwise and generationwise tests in DESigFAR, respectively.

Table 6 compares the non-evolutionary ARM and DESigFAR results. DESigFAR exhibited striking superiority, obtaining 2–10 times as many rules and 3–10 times as high leverages as the best non-evolutionary ARM result for each dataset (bolded in Table 6). The non-evolutionary ARM computed rule leverages with crisp supports, and since ordinary ARM mostly overestimates RIM values [30], the leverages would be even smaller if they were computed with fuzzy supports like DESigFAR. The superiority of DESigFAR appears to be mainly due to its strength in optimizing data discretization schemes. For ARM with prespecified data discretization, it is hardly feasible to find even the optimal numbers of concepts for each attribute by trials: to try out 2–5 concepts for *n* attributes, the algorithm needs to run for 4^n times, that is, $4^{20}=1.1\times10^{12}$ times for Hotel data and $4^9 = 2.6\times10^5$ times for Fire data. The clustering algorithms and metrics should to some extent optimize the number of concepts in each attribute and data intervals of the concepts, but they did not help much in this experiment; in fact, the best results for most datasets were obtained by the equisize scheme (Table 6). As stated in Section 2.4, data discretization by clustering are based on the distribution of data in individual attributes, and the thereby determined scheme may not also optimize the associations between the attributes, or resultantly optimize the rules or RIM values. If experts can prespecify an appropriate and practically meaningful data discretization scheme, the scheme may be favored over an optimized one, even it results in lower RIM values. However, such a occasion falls beyond the scope of this article and optimized ARM in general.

Table 6. Comparison between results of DESigFAR and non-evolutionary ARM.

Class division by	No. of concepts	Hotel				FireS				FireM				FireL			
		Stat. sound + Holm[a] (FWER<5%)		B-H-Y[b] (FDR <5%)		Stat. sound + Holm (FWER<5%)		B-H-Y (FDR <5%)		Stat. sound + Holm (FWER<5%)		B-H-Y (FDR <5%)		Stat. sound + Holm (FWER<5%)		B-H-Y (FDR <5%)	
		No. of rules	Total leverage	No. of rules	Total leverage	No. of rules	Total leverage	No. of rules	Total leverage	No. of rules	Total leverage ($\times 10^4$)	No. of rules	Total leverage ($\times 10^4$)	No. of rules	Total leverage ($\times 10^5$)	No. of rules	Total leverage ($\times 10^5$)
Equisize	2	4	54.5	4	54.5	14	1312.7	18	1546.2	94	4.03	111	4.37	182	7.77	188	7.95
	3	**7**	**105.9**	0	0.0	24	1971.5	**30**	**2296.6**	**116**	**4.21**($\times 10^4$)	**150**	**4.85**($\times 10^4$)	**307**	**9.77**($\times 10^5$)	**320**	**9.94**($\times 10^5$)
	4	0	0.0	0	0.0	**25**	**1720.5**	27	1804.8	106	3.39	120	3.66	244	6.89	257	7.07
	5	1	13.4	0	0.0	20	1307.6	20	1307.6	72	2.11	80	2.24	132	3.71	137	3.81
K-means clustering	2	2	19.6	2	19.6	14	1050.2	16	1186.6	59	2.15	72	2.48	139	5.38	144	5.56
	3	1	10.1	0	0.0	16	1045.9	19	1213.5	83	2.47	98	2.72	233	6.60	258	6.99
	4	0	0.0	0	0.0	21	1251.7	24	1389.3	78	2.04	88	2.17	242	5.25	265	5.47
	5	1	14.2	1	14.2	12	609.4	12	609.4	58	1.41	70	1.57	167	3.25	172	3.35
	Min D-B[c]	0	0.0	0	0.0	11	864.1	13	952.6	67	1.87	72	1.96	189	4.69	202	4.82
	Max Silh.[de]	2	19.6	2	19.6	20	1526.2	23	1821.4	75	3.02	96	3.50	179	7.57	186	7.77
Agglom-erative clustering[e]	2	6	72.5	**6**	**72.5**	12	902.9	15	958.8	85	3.85	98	4.21	151	7.24	157	7.42
	3	2	24.7	2	24.7	13	1019.8	14	1073.1	93	2.89	117	3.33	274	7.80	291	8.06
	4	0	0.0	0	0.0	12	751.8	12	751.8	75	1.90	88	2.10	217	4.81	232	5.03
	5	0	0.0	0	0.0	12	662.2	12	662.2	64	1.47	67	1.53	172	3.61	180	3.72
	Min D-B	1	11.9	1	11.9	13	924.8	13	924.8	78	2.00	91	2.41	240	5.91	255	6.11
	Max Silh.	6	72.5	6	72.5	16	1218.7	19	1315.5	86	3.27	104	3.68	180	7.03	186	7.19
DESigFAR, Gaus. fuzzy set, ft_{min}=0.5		**11** (experimentwise)	**175.6**	**57.9** (generationwise)	**824.7**	**56.3** (experimentwise)	**5150.9**	**97.4** (generationwise)	**7232.1**	**265.2** (experimentwise)	**1.73×10^5**	**314.5** (generationwise)	**1.93×10^5**	**616.32** (experimentwise)	**2.93×10^6**	**662.9** (generationwise)	**3.03×10^6**

[a] Statistically sound test + Holm procedure
[b] Benjamini-Hochberg-Yekutieli procedure
[cd] Number of concepts for each attribute with minimal Davies-Bouldin Index and maximal Silhouette Coefficient, among 2-5 concepts
[e] Results based on FireM were used for FireL, since the computations on FireL exceeded the memory of a 192GB-memory server

(2) GA-Based ARM

The baseline GA-based ARM was set to be identical to DESigFAR in as many aspects as possible. The GA adopted the same individual encoding and fitness assignment as DESigFAR, but a standard GA evolutionary model with elitism. The GA parameters were decided by a preliminary tuning aiming to speed up the evolution. The population size and number of elites were 120 and 40 for Hotel data, and 335 and 160 for Fire data. These left 80 and 175 non-elite individuals, which were equal to the entire populations in DESigFAR for the datasets, so that the two algorithms had the same number of individuals that could evolve per generation. The GA adopted two-point crossover with a crossover fraction of 0.8. The mutation rate was 0.025, that is, each gene in a mutated individual had a probability of 0.025 to be mutated, by adding a random value with mean = 0 and standard deviation = 0.03 timed the attribute value range. The generationwise and experimentwise tests on rules were also conducted, with κ values determined by replacing $2N$ with N in Eqs. (16) and (18), since the risk of spurious rules in GA was shared by N individuals instead of $2N$ in DE. Other settings, such as number of generations, forms of rules, and crisp-fuzzy and sampling strategies, were the same as DESigFAR.

Table 7 compares the GA and DESigFAR results, with each value representing the statistic of 25 runs, and lists the results of student's *t*-tests on whether the total leverages obtained by DESigFAR were larger than those by the GA. DESigFAR obtained significantly larger total leverages than the GA in most treatments, showing its superiority in optimizing the rules. Note that the experimented GA was unavailable in past studies, since there were no statistical tests for strictly controlling the spurious rules in EAs prior to the proposed generationwise and experimentwise tests.

Table 7 Comparison between results of DESigFAR and GA-based ARM.

(a) Hotel experiment.

Discretization model	ft_{min}	GA								DE (DESigFAR)								t-test: DE > GA in leverage	
		No. of rules				Total leverage				No. of rules				Total leverage					
		Mean	SD	Max	Min	Mean	SD	Max	Min	Mean	SD	Max	Min	Mean	SD	Max	Min	t	p
Tri.	0.3	39.4	5.1	50	29	523.7	61.3	644.5	398.1	59.1	4.6	66	52	843.9	53.6	939.8	753.3	19.7	0.0000
	0.5	38.8	4.9	47	24	507.6	59.7	586.3	328.3	59.0	5.9	69	49	765.9	58.7	896.4	669.5	15.4	0.0000
	0.7	32.6	5.5	43	22	397.0	56.3	505.7	271.3	52.3	5.2	61	41	625.7	49.3	723.9	521.5	15.3	0.0000
Trapez.	0.3	40.0	3.8	48	34	605.7	52.2	716.6	535.9	61.8	5.2	75	54	953.2	68.1	1120.8	875.5	20.3	0.0000
	0.5	40.3	4.5	47	30	589.4	43.6	651.0	494.1	54.8	5.0	65	44	766.1	46.7	846.9	661.0	13.8	0.0000
	0.7	27.1	3.2	33	22	378.0	41.9	465.6	287.5	52.3	5.2	61	42	664.1	44.0	743.5	577.1	23.5	0.0000
Gaus.	0.3	43.1	4.2	52	37	664.9	64.0	799.4	556.7	63.5	6.3	76	51	1004.4	77.5	1143.0	873.9	16.9	0.0000
	0.5	43.3	3.6	50	36	642.7	49.9	739.7	557.4	57.9	6.7	70	46	824.7	70.9	967.0	700.1	10.5	0.0000
	0.7	29.0	3.8	38	21	412.7	43.6	508.7	291.2	53.4	5.8	64	39	710.2	57.2	806.7	579.7	20.7	0.0000

(b) Fire experiment, $ft_{min} = 0.5$.

Data		Discretization model	GA								DE (DESigFAR)								t-test: DE > GA in leverage	
			No. of rules				Total leverage ($\times 10^5/10^6$ for FireM/FireL)				No. of rules				Total leverage ($\times 10^5/10^6$ for FireM/FireL)					
			Mean	SD	Max	Min	Mean	SD	Max	Min	Mean	SD	Max	Min	Mean	SD	Max	Min	*t*	*p*
FireS	Generati-onwise	Tri.	89.6	4.8	100	80	5843	376.4	6698	5416	94.8	3.4	103	89	6754	219.2	7206	6360	10.5	0.0000
		Trapez.	93.7	4.3	103	85	6887	403.1	7501	6007	96.4	4.3	106	86	6913	275.9	7663	6367	0.3	0.3945
		Gaus.	94.9	4.0	102	88	6910	421.5	7927	6317	97.4	3.5	103	90	7232	240.4	7643	6835	3.3	0.0015
	Experim-entwise	Tri.	43.8	2.7	48	37	3332	98.94	3491	3110	51.7	2.4	56	47	4804	159.2	5135	4486	39.3	0.0000
		Trapez.	48.2	3.8	57	41	3952	183.8	4354	3631	54.3	3.2	61	47	4967	186.7	5429	4646	19.4	0.0000
		Gaus.	49.0	3.2	56	44	4056	214.2	4483	3764	56.3	3.2	62	50	5151	215.8	5516	4669	18.0	0.0000
FireM	Generati-onwise	Tri.	358.6	11.1	386	341	1.573	0.048	1.690	1.488	307.1	5.4	319	297	1.807	0.017	1.841	1.763	22.9	0.0000
		Trapez.	351.2	8.3	368	335	1.758	0.038	1.831	1.686	315.2	8.8	329	299	1.896	0.022	1.924	1.852	15.8	0.0000
		Gaus.	348.9	8.0	366	332	1.778	0.044	1.856	1.693	314.5	7.8	329	300	1.929	0.024	1.974	1.883	15.1	0.0000
	Experim-entwise	Tri.	268.3	7.2	291	259	1.313	0.035	1.398	1.242	258.1	4.2	265	251	1.617	0.020	1.651	1.575	37.6	0.0000
		Trapez.	264.7	7.0	278	255	1.475	0.031	1.536	1.407	265.8	7.6	279	243	1.712	0.066	1.990	1.650	16.3	0.0000
		Gaus.	266.5	7.2	281	256	1.530	0.047	1.612	1.462	265.2	5.0	273	255	1.729	0.031	1.795	1.674	17.6	0.0000
FireL (sampled)	Generati-onwise	Tri.	806.5	20.3	849	767	2.440	0.082	2.617	2.319	638.0	22.5	672	588	2.794	0.041	2.864	2.714	19.4	0.0000
		Trapez.	789.6	25.4	841	739	2.733	0.058	2.837	2.628	675.4	22.1	715	630	2.940	0.043	3.028	2.850	14.3	0.0000
		Gaus.	780.3	25.7	823	708	2.793	0.065	2.920	2.672	662.9	27.0	703	613	3.021	0.047	3.095	2.912	14.2	0.0000
	Experim-entwise	Tri.	749.4	18.8	789	712	2.336	0.055	2.456	2.226	582.1	19.0	619	548	2.731	0.036	2.809	2.667	30.0	0.0000
		Trapez.	713.8	26.8	759	637	2.685	0.104	3.089	2.543	611.9	31.3	663	561	2.888	0.071	3.123	2.764	8.1	0.0000
		Gaus.	719.3	24.0	755	669	2.726	0.072	2.823	2.558	616.3	18.3	662	579	2.938	0.035	3.007	2.861	13.3	0.0000

A larger number of resultant rules was not an objective of the optimization, though it is desirable when the rules are relatively scarce, as was the case with Hotel and FireS data, from which DESigFAR indeed discovered more rules than non-evolutionary and GA-based ARM (Tables 6, 7). For FireM and FireL data, both DESigFAR and the GA discovered rules from at least 160 main rules. With eight attributes for the antecedent of up to four items and fixed attribute for the consequent, Fire data could constitute $C_8^1 + C_8^2 + C_8^3 + C_8^4 = 162$ main rules. Thus, the data was so rich that rules rising from almost any attribute combinations, even subtle rules from weakly associated attributes, could be statistically significant. Then the focus of EAs should shift from discovering more rules to finding better data discretization schemes, or concept definitions. For FireM and FireL data, DESigFAR resulted in fewer rules but still notably larger total leverages than the GA. A look into the rule contents revealed that the main reason should be DESigFAR often found more concise rules. For example, when the GA resulted in two rules “elevation = low → h_dist_fire = near” and “elevation = mid-low → h_dist_fire = near”, DESigFAR tended to discover a single rule “elevation = low → h_dist_fire = near”, and the value range of low elevation roughly covered the ranges of low and mid-low elevations in GA. This shows the higher ability of DESigFAR to optimize the concept definitions over GA.

Investigated by altering each part of the algorithms, the advantage of DESigFAR on small data over GA were found to mainly came from the opposition-based generation jump which did not fit GA. The advantage for large data was found to be attributed to the DE mutation: if the raw data value intervals for the same concept in different individuals had large disparities from each other, the magnitude of mutation would be automatically enlarged to actively search for better intervals. If the interval values in different individuals were close to each other, the magnitude of mutation became smaller to maintain already good intervals [32]. In addition, as a single crossover or mutation is much more likely to worsen rather than improve the genes, GA must keep a weirdly large number of elites free from crossover or mutation to preserve good rules, like 40 and 160 elites in this study. Otherwise, the GA result would be much worse, which had been confirmed in the preliminary tuning. DESigFAR avoids this problem, as in DE the individuals containing good rules will survive unaltered if its fitness is higher than its offspring.

4.5 Computational performance

Fig. 8 illustrates the average run times for different treatments and datasets programmed by MATLAB® 2012a. Fire data was experimented on a Windows Server with Intel Xeon E5 2.00GHz, 8-core parallel processing and about 3.5x speedup. The small-sized Hotel data did not benefit from parallel processing due to relatively small workload per generation, and since the server was not designed for efficient single-core processing, the data was experimented on a Windows laptop with Intel i7 2.10GHz to produce a more realistic evaluation on the computational performance.

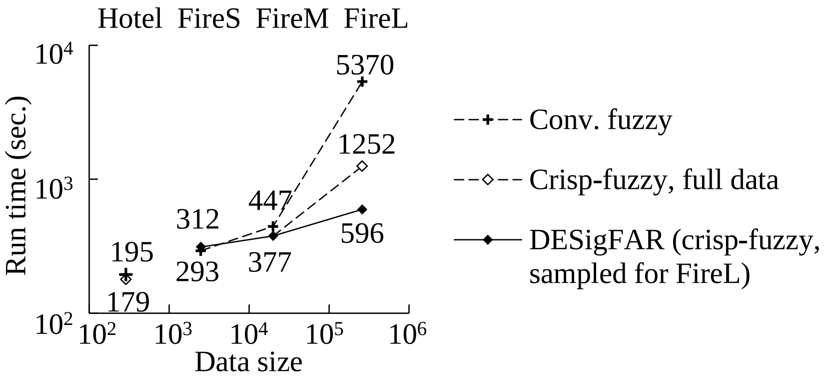

Fig. 8. Run time for treatments with ft_{min} = 0.5, average of generationwise and experimentwise approaches.

As datasize increased, DESigFAR, thanks to its crisp-fuzzy strategy, started to gain marked efficiency advantage over the conventional fuzzy method. Further, the proposed sampling strategy substantially speeded up the algorithm, making the run time less than double for the datasize increase of over 100 times from FireS to FireL (Fig. 8). Like EA-based ARM in general, the run time of DESigFAR is roughly proportional to population size and number of generations. Larger-sized data usually requires fewer generations for the DE to converge, as more rules become significant in early generations with richer data. DESigFAR was mostly faster than its conventional fuzzy counterpart due to its much faster crisp discretization on compressed data in the statistical test stage. Still, as fuzzy discretization for RIM evaluation must be performed on each record rather than compressed data, in the worst case, this operation can be of linear time complexity against the datasize. Therefore, the sampling strategy is critical for keeping the DE highly scalable.

4.6 Practical implications of Hotel experiment

4.6.1 Results: resource accessibility, hotel room price premium and scale effect

In Hotel experiment, DESigFAR found 67 rules in the best run with $ft_{\min} = 0.5$ and the generationwise approach. Table 8 listed these rules, with each item like

$$\underbrace{\text{attribute } a = \text{concept } l_i}_{\substack{\textit{numerical data interval where} \\ m_{l_i} \textit{ is the largest among } m_{l_1} \ldots m_{l_k}}}.$$

Concepts for accessibilities containing 2–5 values were ‘near, far’, ‘near, mid, far’, ‘near, mid-near, mid-far, far’ and ‘near, mid-near, mid, mid-far, far’. Concepts for hotel room prices were similar but with ‘low/high’ instead. The price level concepts optimized by DESigFAR did not simply reflect star ratings, but rather implied hotel profitability. ‘Mid’ and ‘high’ prices in most rules were divided at HK$1200–1300 around median prices of 4-star hotels and differentiated underpriced and well-sold ones among them. 4-star hotels constituted the largest star rating group with 114 of the 290 hotels in data.

Table 8 Resultant rules of Hotel experiment.

	Antecedent (m)	Consequent: room_price = (HK$)
1	dist_topspot1 = near ∧ dist_topspot4 = near ∧ dist_worship = far *<1227* *<2453* *>799*	high *>854*
2	dist_topspot1 = near ∧ dist_bus4 = far *<966* *>369*	high *>1265*
3	dist_topspot2 = near *<1599*	high *>1261*
4	dist_topspot2 = near ∧ dist_topspot4 = near ∧ dist_worship = far *<1493* *<2453* *>799*	high *>854*
5	dist_topspot2 = near ∧ dist_worship = mid *<1130* *620–1155*	high *>1167*
6	dist_topspot2 = near ∧ dist_bus4 = mid *<2097* *>369*	high *>1278*
7	dist_topspot3 = near *<1822*	high >1245
8	dist_topspot3 = near ∧ dist_worship = far *<1822* *>632*	high *>1164*
9	dist_topspot3 = near ∧ dist_worship = far ∧ dist_topspot4 = near *<1974* *>799* *<2453*	high *>854*
10	dist_topspot3 = near ∧ dist_subway = far *<1974* *>212*	high *>854*
11	dist_topspot3 = near ∧ dist_bus4 = far *<1851* *>369*	high *>1265*
12	dist_topspot4 = near *<1937*	high *>1208*
13	dist_topspot4 = near ∧ dist_topspot2 = far *<1810* *>775*	high *>1050*
14	dist_topspot4 = near ∧ dist_topspot2 = far ∧ dist_shop2 = near *<1810* *>775* *<405*	high *>1069*
15	dist_topspot4 = near ∧ dist_worship = mid *<2481* *659–1799*	high *>1125*
16	dist_topspot4 = near ∧ dist_subway = far *<1809* *>215*	high *>1048*
17	dist_topspot4 = near ∧ dist_bus4 = far *<2952* *>369*	high *>1269*
18	dist_topspot5 = near *<3048*	high *>1265*
19	dist_topspot5 = near ∧ dist_worship = mid *<2321* *653–1556*	high *>1113*
20	dist_topspot5 = near ∧ dist_subway = far *<2213* *>212*	high >854
21	dist_topspot5 = near ∧ dist_bus4 = far *<2213* *>369*	high >1267
22	dist_museum = near *<929*	high *>1221*
23	dist_museum = near ∧ dist_worship = mid *<1199* *668–1605*	high *>1040*

24	dist_museum = near ∧ dist_shop1 = near *<929* *<63*	high *>1221*
25	dist_museum = near ∧ dist_subway = far *<929* *>205*	high *>740*
26	dist_shop1 = near *<37*	high *>1339*
27	dist_shop1 = near ∧ dist_worship = mid *<80* *781–1250*	high *>1401*
28	dist_shop3 = near ∧ dist_bus4 = far *<650* *>369*	high *>1265*
29	dist_shop4 = near ∧ dist_bus4 = far *<647* *>369*	high *>1267*
30	dist_topspot1 = near *<395*	low *<567*
31	dist_topspot1 = near ∧ dist_worship = near *<523* *<648*	low *<699*
32	dist_topspot1 = near ∧ dist_shop5 = near *<465* *<575*	low *<563*
33	dist_topspot1 = near ∧ dist_subway = near *<456* *<244*	low *<563*
34	dist_topspot2 = near ∧ dist_worship = near *<1130* *<620*	low *<1167*
35	dist_topspot2 = near ∧ dist_worship = near *<1130* *<620*	low *<1167*
36	dist_topspot5 = near ∧ dist_worship = near *<2321* *<653*	low <642
37	dist_shop1 = mid 37–184	low *<556*
38	dist_shop1 = near ∧ dist_bus4 = near *<122* *<227*	low *<556*
39	dist_shop2 = near <298	low <553
40	dist_shop3 = near *<357*	low *<561*
41	dist_shop3 = near ∧ dist_subway = near *<350* *<270*	low *<570*
42	dist_shop4 = near *<424*	low *<561*
43	dist_shop4 = near ∧ dist_subway = near *<519* *<272*	low *<561*
44	dist_shop5 = near *<479*	low *<563*
45	dist_shop5 = near ∧ dist_subway = near *<567* *<214*	low *<559*
46	dist_topspot1 = far *>395*	mid *567–1510*
47	dist_topspot1 = far ∧ dist_topspot4 = far *>360* *>1815*	mid *510–1239*
48	dist_topspot1 = far ∧ dist_topspot5 = far *>360* *>3082*	mid *510–1239*

49	dist_topspot2 = far *>1599*	mid *536–1250*
50	dist_topspot2 = far ∧ dist_topspot5 = far *>1129* *>2975*	low *<1303*
51	dist_topspot3 = far *>1822*	mid *516–1245*
52	dist_topspot4 = far *>1937*	low *<1531*
53	dist_topspot4 = far ∧ dist_topspot5 = far *>1937* *>2250*	mid *538–1208*
54	dist_topspot4 = far ∧ dist_shop1 = far *>1725* *>121*	mid *510–1239*
55	dist_topspot4 = far ∧ dist_shop2 = far *>1937* *>239*	mid *510–1239*
56	dist_topspot5 = far *>3048*	mid *524–1265*
57	dist_museum = far *>929*	low *<1221*
58	dist_shop1 = far *>185*	mid *556–1339*
59	dist_shop2 = far *>298*	mid *553–1329*
60	dist_shop3 = far *>357*	mid *561–1380*
61	dist_shop4 = far *>424*	mid *561–1380*
62	dist_shop5 = far *>472*	mid *563–1301*
63	dist_worship = mid *654–1451*	high *>1103*
64	dist_subway = near *<232*	low *<556*
65	dist_subway = far *>232*	mid *556–1881*
66	dist_bus3 = near *<145*	low *<545*
67	dist_bus5 = mid *194–209*	low *<629*

Resultant rules suggest direct associations between high room prices of hotels and their proximity to the nearest and clusters of top attractions (<1km–<3km for the nearest to fifth nearest, rule 1–21, Table 8), museums (<1km, rule 22–25) and shopping places (<650m for the third and fourth nearest, rule 28–29). Hotels relatively far to these resources tend to have low to medium room prices (rule 46–62), which also implies the importance of high accessibility to these resources to room price premium. Meanwhile, hotels nearest to top attractions (<400m–<2300m for the nearest to fifth nearest) and shops (<100–<500m for the nearest to fifth nearest) can have low prices (rule 30–45). These distance ranges, however, do not suggest much more convenient walking accesses than the distances for high–price hotels in rule 1–29. Hence, rule 30–45 seem not to suggest adverse effects of high accessibility to resources on room prices, but instead smaller scales of architectures and nearness concepts for areas clustered with cheap hotels than with expensive ones. Most expensive hotels locate in upscale commercial areas with large and widely spaced buildings. In terms of distances for winding walks between entrances of large buildings, distances in rule 30–45 seem a bit too short for these upscale areas, except for hotels immediately adjacent to these resources. Cheap hotels concentrate in old districts with dense and smaller buildings and have larger chances to locate very close to the resources. For example, dozens of cheap hotel buildings are within 300m to top attractions Ladies' Market and Temple Street featured for night markets (Fig. 5). The exceptional rule 26 for high-price hotels within 37m to shopping places reflects luxury hotels built directly over malls.

Proximity to worship attractions, subway stations and bus stop clusters alone appear not contributive to room price premium (rule 63–67). Looking into the data, most religious spots locate in either old and crowded districts or remote places with few nearby hotels, such as the Big Buddha (Fig. 5). Hong Kong has a dense subway network, making most hotels within 600m to subway entrances, and its bus network is even much denser. Thus, except for some remote hotels, accessibility to subway and buses are indiscriminate among the studied hotels, and "near" concepts for these facilities are more likely to reflect small architecture scales and old crowded districts. This can explain why the hotels near top attractions and shops have low prices especially when they are also very close to subway stations (rule 33, 41, 43, 45), bus stop clusters (rule 38) or worship places (rule 31, 34–36), but have high prices when they are not very close to these resources (rule 1, 2, 4–6, 8–11, 13–17, 19-21, 27–29).

In sum, nearness to top attractions, museums and shopping places are generally favorable for hotel room price premium. Nearness to worship places is unhelpful; accessibilities to subway and buses are largely indiscriminate among the studied hotels, and very close proximity to them suggests old crowded areas with small architecture scales and low hotel room prices. Closest proximity to top attractions and shopping places, if accompanied with nearness of worship places, subway and buses, can behave opposite to their general positive associations with room prices and also suggest old crowded areas and low room prices.

4.6.2 Comparison with hedonic price modelling and practical recommendations

Existing studies on determinants of hotel room prices primarily take a regressive hedonic price modelling approach, with price the as dependent variable and various hotel attributes as independent variables. The regression models are typically linear, semilog, loglinear or in other forms monotonic with respect to distances [53–59]. Hotel accessibilities to attractions and transport facilities are generally found positively correlated with room prices [53–59], but sometimes they exhibit insignificant or even slightly negative correlations with the prices for certain room types [53], hotel types [56] or geographical locations [55]. It is difficult to confirm whether such inconsistent findings are due to heterogonous impacts of the accessibilities under various conditions, or simply inability to draw statistically significant correlations from limited numbers of hotels, usually up to hundreds for city-level studies. Besides, most prior studies measured hotel accessibilities to either any available or only one type of attractions or transport facilities. Few studies have compared the accessibilities to different attractions or transport subtypes.

In Hotel experiment, DESigFAR partially overcame the above difficulty of limited datasize and discovered relatively rich DE-optimized rules with low risk to be spurious, thereby enabling the analysis on accessibilities to resources in more detailed subtypes than past hedonic studies. Besides, the algorithm utilized correlations among accessibility factors, which could rather deteriorate regressive price modelling results, to generate multi-factor rules for reasoning the effect of and interaction between individual factors, as shown in Section 4.5.1.

The experiment result has three implications on hotel room price modeling studies. First, effects of accessibilities to tourism resources on room prices can vary a lot across urban regions with different scales of streets and architectures. Monotonic regression models tend to overlook this difference, which might have hindered conclusive and consistent findings in past hedonic hotel pricing studies. DESigFAR may help investigate such different scales and effects of accessibilities. Second, accessibilities to different resource subtypes can have heterogeneous effects on room prices. Unhelpful or indiscriminate subtypes, like worship places, subway entrances and bus stops in this study, can hide real influences of other subtypes and degrade the modelling result. Such unfavorable subtypes are difficult to identify once included in accessibilities for more general resource types, and may also have contributed to inconsistent findings in past hedonic pricing studies. It is recommended that future studies try to sub-classify the resource types and pilot the effects of these subtypes on room prices before constructing the pricing model. Third, the distance intervals for accessibility levels optimized by DESigFAR can help users learn effective distances to various resource types that contributes to the price premium. The resultant rule leverages can help estimate the room sale premium by locating the hotels in favorable sites.

4.7 Practical implications of Fire result

Focusing on evaluating DESigFAR with various datasizes rather than discovering new practical insights, Fire experiment employed a relatively small number of factors whose fire-inducing effects were mostly known through past empirical studies. Table 9 lists the top-20 rules in terms of leverage with h_dist_fire = near (high fire risks) as the consequent from the best run taking $ft_{\min}$ = 0.5 and Gaussian membership functions. Most rules agree to findings in past empirical studies (Table 2). Optimized concept boundaries in these rules can help identify risky value ranges of these factors in fire risk monitoring. Interestingly, proximity to surface waters shows two-way effects: distance beyond around 150m (rule 4, 17) and within 330m (rule 8, 10, 11) to waters are both linked to higher risks. It appears that fire-mitigating effect of waters is effective within around 150m, while their effect on increasing forest density and thus fire risks in the water-stressed study area [47] is effective within around 330m, leaving the distance range in between the most prone to fires. Rules with h_dist_water alone as the antecedent (rule 372, 478, Table 7) agree to this speculation, but they ranked low by leverage among the 696 resultant rules, showing that water is not among the most influential risk factors. Thus, it can be difficult to deduce the above detailed fire-inducing effect of waters through empirical regressive studies.

Table 9 Resultant rules for Fire experiment, top-20 and selective ones, ranked by leverage.

	Antecedent (degree for slope, m for others)	Consequent: h_dist_fire = near (m)
1	elevation = low ∧ h_dist_road = near *<2756 <2837*	*<1302*
2	h_dist_road = near *<1208*	*<1430*
3	elevation = low *<2700*	*<1126*
4	h_dist_water = far ∧ h_dist_road = near *>121 <1526*	*<2044*
5	elevation = low ∧ slope = large ∧ v_dist_water = far ∧ h_dist_road = near *<3061 >16 >12 <2308*	*<2112*
6	elevation = low ∧ v_dist_water = far ∧ h_dist_road = near *<3090 >27 <2323*	*<2306*
7	elevation = low ∧ slope = large ∧ h_dist_road = near *<2756 >14 <3026*	*<1111*
8	elevation = low ∧ h_dist_water = near *<2683 <327*	*<1529*
9	elevation = low ∧ slope = large *<2748 >14*	*<1111*
10	elevation = low ∧ h_dist_water = near ∧ h_dist_road = near *<2683 <327 <1983*	*<1529*
11	elevation = low ∧ h_dist_water = near ∧ v_dist_water = far ∧ h_dist_road = near *<3061 <342 >12 <1897*	*<2379*
12	v_dist_water = far ∧ h_dist_road = near ∧ hillshade_12nn = low *>13 <2067 <236*	*<1676*
13	slope = large *>17*	*<1201*
14	slope = large ∧ h_dist_water = near *>19 <373*	*<1676*
15	elevation = low ∧ slope = large ∧ v_dist_water = far *<3121 >17 >22*	*<2080*
16	elevation = low ∧ v_dist_water = far *<2752 >14*	*<1118*
17	elevation = low ∧ slope = large ∧ h_dist_water = far ∧ h_dist_road = near *<3098 >12 >168 <2336*	*<2359*
18	elevation = low ∧ hillshade_12nn = low *<2739 <223*	*<1034*

19	slope = large ∧ v_dist_water = far ∧ h_dist_road = near ∧ hillshade_12nn = low *>15* *>13* *<2117* *<236*	 *<1286*
20	v_dist_water = far ∧ h_dist_road = near *>30* *<1207*	 *<1282*
372	h_dist_road = far *>366*	(mid-far) *2583–4126*
478	h_dist_road = near *<101*	 *<1261*

5. Conclusions

This article puts forward DESigFAR, the DE-based statistically sound fuzzy ARM, for mining fuzzy association rules with overall quality improvement regarding abundance of rules, low risk of spurious rules, and goodness of RIM values. For the first time in EA-based ARM, DESigFAR realizes strict control over the risk of spurious rules via statistically sound significance tests on the rules, meaning that the tests have their significance levels corrected for the multiple comparisons problem with respect to numbers of all potential rules rather than pre-filtered and tested rules. The proposed DE can also markedly increase the number of resultant rules and fitness of their RIM values via genetic optimizations, compared with conventional ARM with predetermined data discretization schemes.

As the existing statistically sound test for conventional ARM does not apply to EAs, new tests are developed for DE with two options: the experimentwise adjustment approach for controlling the FWER, and the generationwise adjustment approach for controlling the percentage of spurious rules under the user specified level. Specific individual encoding, evolutionary model and speedup strategy for the DE are also developed. The proposed DE may be used with various RIMs as optimization objectives and criteria on interesting rules.

Experiments with variously sized data show that DESigFAR can obtain 2–10 times as many rules and 3–10 times as high RIM values as non-EA ARM, while keeping the FWER and percentage of spurious rules well below the user specified level. The algorithm is highly scalable to large datasets. In case studies on hotel room price and wildfire risk modeling, DESigFAR revealed correlated influences on room prices of more detailed tourism resource subtypes than prior studies and variations in scales of architecture and nearness measurement, as well as detailed fire-inducing behaviors of certain fire risk factors.

The current study is planned to be extended into multiobjective DE-based fuzzy ARM, which can find rules that achieve compromised near-optimum for multiple objectives, so as to satisfy the need for multicriteria rule selection in real applications. Compared with evaluating the individuals by weighted fitness values of all objectives, the strategy of prioritizing non-dominated individuals [1, 2, 4] is more objective and suitable for non-comparable objectives, for example, rule confidence versus the number of rules. An individual is non-dominated if no other individuals have higher fitness than it for all objectives. While the new features in DESigFAR are compatible with multiobjective algorithm with the prioritization of non-dominated individuals, further investigations are needed for unforeseen issues in integration of the two techniques.

Acknowledgements

This study is funded by Minister of Science and Technology of P.R. China (2017YFB0503604), the National Natural Science Foundation of China (41331175）and the Hong Kong Polytechnic University (1-ZE24, 4-ZZFZ).

References

[1] Kaya, M., 2006. Multi-objective genetic algorithm based approaches for mining optimized fuzzy association rules. *Soft Computing*, 10, 578–586.

[2] Chen, C., Hong, T., Tseng, S., and Chen, L., 2008. A Multi-objective genetic-fuzzy mining algorithm. In: *2008 IEEE International Conference on Granular Computing*, 26-28 August 2008 Hangzhou, 115–120.

[3] Alcalá-Fdez, J. Alcalá, R., Gacto, M.J., and Francisco Herrera, F., 2009. Learning the membership function contexts for mining fuzzy association rules by using genetic algorithms. Fuzzy Sets and Systems, 160, 905–921.

[4] Casillas, J. and Martinez-Lopez, F.J., 2009. Mining uncertain data with multiobjective genetic fuzzy systems to be applied in consumer behaviour modelling. *Expert Systems with Applications*, 36(2), 645–1659.

[5] Chen, C., Hong, T., Lee, Y., and Tseng, V, 2015. Finding active membership functions for genetic-fuzzy data mining. International Journal of Information Technology & Decision Making, 14(6), 1215–1242.

[6] Hüllermeier, E., 2009. Fuzzy methods in data mining. In: John Wang (ed.) *Encyclopedia of Data Warehousing and Mining, 2nd Edition*. IGI Global: Hershey, USA, 907–912.

[7] Farzanyar, Z. and Kangavari, M., 2012. Efficient mining of fuzzy association rules from the pre-processed dataset. *Computing and Informatics,* 31, 331–347.

[8] Kuok, C. M., Fu, A., and Wong, M. H., 1998. Mining Fuzzy Association Rules in Databases. *SIGMOD Record*, 27(1), 41–46.

[9] Fazzolari, M., Alcalá, R., Nojima, Y., shibuchi, H., and Herrera, F., 2013. A review of the application of multiobjective evolutionary fuzzy systems: Current status and further directions. *IEEE Transactions on Fuzzy Systems*, 21(1), 45–64.

[10] Webb, G.I., 2007. Discovering significant patterns. *Machine Learning*, 68, 1–33.

[11] Zhang, A., Shi, W., and Webb, G.I., 2016. Mining significant association rules from uncertain data. *Data Mining and Knowledge Discovery,* 30(4), 928–963.

[12] Megiddo, N. and Srikant, R., 1998. Discovering predictive association rules. In: *The fourth international conference on knowledge discovery and data mining*. Menlo Park: AAAI, 27–78.

[13] Liu, B., Hsu, W., and Ma, Y., 1999. Pruning and summarizing the discovered associations. In: *The* fifth *ACM SIGKDD international conference on knowledge discovery and data mining*. New York: AAAI, 125–134.

[14] Bay, S.D. and Pazzani, M.J., 2001. Detecting group differences: Mining contrast sets. *Data Mining and Knowledge Discovery*, *5*(3), 213–246.

[15] Zhang, H., Padmanabhan, B., and Tuzhilin, A., 2004. On the discovery of significant statistical quantitative rules. In: *The tenth international conference on knowledge discovery and data mining*, New York: ACM, 374–383.

[16] Agrawal, R., Imielinski, T., and Swami, A., 1993. Mining associations between sets of items in massive databases. In: *1993 ACM-SIGMOD International Conference on Management of Data,* Washington, DC, 207–216.

[17] Tew, C., Giraud-Carrier, C., Tanner, K., and Burton, S., 2014. Behavior-based clustering and analysis of interestingness measures for association rule mining. *Data Mining and Knowledge Discovery,* 28(4),1004–1045.

[18] Bayardo, R. J., Jr., Agrawal, R., and Gunopulos, D., 2000. Constraint-based rule mining in large, dense databases. *Data Mining and Knowledge Discovery*, *4*(2/3), 217–240.

[19] Zaki, M.J., 2000. Generating non-redundant association rules. In: *The 6th ACM SIGKDD International* Conference *on Knowledge Discovery and Data Mining (KDD-2000)*, 34–43.

[20] Liu, B., Hsu, W., and Ma, Y., 2001. Identifying non-actionable association rules. In: *The 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'01)*, 329–334.

[21] Jensen, D. D. and Cohen, P. R., 2000. Multiple comparisons in induction algorithms. *Machine Learning*, *38*(3), 309–338.

[22] Shaffer, J. P., 1995. Multiple hypothesis testing. *Annual Review of Psychology*, 46, 561–584.

[23] Bosc, P., Dubois, D., HadjAli, A., Pivert, O., and Prade, H., 2007. Adjusting the core and/or the support of a fuzzy set - A new approach to fuzzy modifiers. In: *IEEE International Fuzzy Systems Conference 2007*, 23-26 July 2007 London, 1–6.

[24] Herrera, F. and Martinez, L., 2000. A 2-tuple fuzzy linguistic representation model for computing with words. *IEEE Transactions on Fuzzy Systems*, 8 (6), 746–752.

[25] Carmona, C.J., Gonzalez, P., del Jesus, M.J., and Herrera, F., 2010. NMEEF-SD: Non-dominated multiobjective evolutionary algorithm for extracting fuzzy rules in subgroup discovery. *IEEE Transactions on Fuzzy Systems*, 18(5), 958–970.

[26] Alhajj, R. and Kaya, M., 2008. Multi-objective genetic algorithms based automated clustering for fuzzy association rules mining. *Journal of Intelligent Information Systems*, 31, 243–264.

[27] Ladner, R., Petry, F.E., and Cobb, M.A., 2003. Fuzzy set approaches to spatial data mining of association rules. *Transactions in GIS*, 7(1), 123–138.

[28] Bordogna, G. and Pasi, G, 1993. A fuzzy linguistic approach generalizing Boolean information retrieval: A model and its evaluation. *Journal of the American Society for Information Science*, 44(2), 70–82.

[29] Burda, M., Pavliska. V., and Valasek, R., 2014. Parallel mining of fuzzy association rules on dense data sets. In: *2014 IEEE International Conference on Fuzzy Systems*, 6–11 July 2014 Beijing.

[30] Shi, W., Zhang, A., and Webb, G.I., 2018. Mining significant crisp-fuzzy spatial association rules. *International Journal of Geographical Information Science*. 32(6), 1247–1270.

[31] Alatas, B., Akin, E., and Karci, A., 2007. MODENAR: Multi-objective differential evolution algorithms for mining numeric association rules. *Applied Soft Computing*, 8, 646–656.

[32] Das, S. and Suganthan, P.N., 2011. Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, 15(1), 4–31.

[33] Kaya M. and Alhajj, R., 2004. Integrating multi-objective genetic algorithms into clustering for fuzzy association rules mining. In: *The IEEE International Conference on Data Mining*, 1–4 November 2004 Brighton, UK.

[34] Thilagam, P.S. and Ananthanarayana, V.S., 2008. Extraction and optimization of fuzzy association rules using multi-objective genetic algorithm. *Pattern Analysis and Applications*, 11, 159–168.

[35] Piatetsky-Shapiro, G., 1991. Discovery, analysis, and presentation of strong rules. In: Piatetsky-Shapiro, G. and Frawley, J. (Eds.), *Knowledge Discovery in Databases*, 229–248. Menlo Park: AAAI/MIT Press.

[36] Holm, S., 1979. A simple sequentially rejective multiple test procedure. *Scandinavian Journal* of *Statistics*, 6, 65–70.

[37] Rahnamayan, S., Tizhoosh, H.R., and Salama, M.M.A., 2008. Opposition-based differential evolution. *IEEE Transactions on Evolutionary Computation*, 12(1), 64–79.

[38] Yager, R.R., 1979. On the measure of fuzziness and negation, Part I: Membership in the Unit Interval. *International Journal of General Systems*, 5, 221–229.

[39] Hong Kong Tourism Board, 2015. *Things to Do*. Available at: https://www.discoverhongkong.com/us/see-do/index.jsp [accessed 2 April 2015].

[40] Census and Statistics Department, HKSAR, 2015. *Hong Kong Annual Digest of Statistics 2015* [online]. Available at: http://www.statistics.gov.hk/pub /B10100032015AN15B0100.pdf [accessed 12 Dec 2016].

[41] [Dataset] Jock A. Blackard, J.A., Dr. Dean, D.J. and Anderson, C.W, Forest CoverType Dataset. UCI Machine Learning Repository, 1998. Available at: https://kdd.ics.uci.edu/databases/covertype/covertype.data.html [accessed 28 April 2015].

[42] Anchor Point Group, 2010. *Anchor Point national wildfire hazard/risk rating model* [online]. Available from: http://www.anchorpointgroup.com/images/APG%20National%20Fire%20Model%20-%20Public.pdf [Accessed 3 May 2015].

[43] Gerdzheva, A.A., 2014. A comparative analysis of different wildfire risk assessment models (a case study for Smolyan district, Bulgaria). *European Journal of Geography*, 5 (3): 22–36.

[44] The Virginia Department of Forestry, 2003. *GIS FAQs: Statewide wildfire risk assessment* [online]. Available from: http://www.dof.virginia.gov/gis/dwnload/Statewide-faq.htm [Accessed 11 May 2015].

[45] Thompson, W.A., Vertinsky, I., Schreier, H., and Blackwell, B.A., 2000. Using forest fire hazard modelling in multiple use forest management planning. *Forest Ecology and Management*, 134(1–3), 63–176.

[46] Ghobadi, G.J., Gholizadeh, B., and Dashliburun, O.M., 2012. Forest fire risk zone mapping from geographic information system in northern forests of Iran (case study,

Golestan province). *International Journal of Agriculture and Crop Sciences*, 4(12), 818–824.

[47] Krasnow, K., Schoennagel, T., and Veblen, T.T., 2009. Forest fuel mapping and evaluation of LANDFIRE fuel maps in Boulder County, Colorado, USA. *Forest Ecology and Management*, 257, 1603–1612.

[48] Pedregosa, F., Varoquaux, G., and Gramfort, A. et al., 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

[49] Webb, G.I. and Zhang, S., 2005. *K*-optimal rule discovery. *Data Mining and Knowledge Discovery*, 10(1), 39–79.

[50] Davies, D.L. and Bouldin, D.W., 1979. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2), 224–227.

[51] Rousseeuw, P.J., 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Computational and Applied Mathematics*, 20, 53–65.

[52] Benjamini, Y. and Yekutieli, D., 2001. The control of the false discovery rate in multiple testing under dependency. *Annals of Statistics*, 29, 1165–1188.

[53] Thrane, C., 2007. Examining the determinants of room rates for hotels in capital cities: The Oslo experience. *Journal of Revenue and Pricing Management*, 5(4), 315–323.

[54] Andersson, D.E., 2010. Hotel attributes and hedonic prices: an analysis of internet-based transactions in Singapore's market for hotel rooms. *The Annals of Regional Science*, 44, 229–240.

[55] Zhang, H., Zhang, J., Lu, S., Cheng, S., and Zhang J., 2011. Modelling hotel room price with geographically weighted regression. *International Journal of Hospitality Management*, 30, 1036–1043.

[56] Zhang, Z., Ye, Q., and Law, R., 2011. Determinants of hotel room price: An exploration of travelers' hierarchy of accommodation needs. *International Journal of Contemporary Hospitality Management*, 23(7), 972–981.

[57] Park, E. and Kim, Y., 2012. An analysis of urban hotel location focusing on market segment and local & foreign guest preference. In: *The Eighth International Space Syntax Symposium*, 3–6 January 2012 Santiago, Chile.

[58] Balaguer, J. and Pernías, J.C, 2013. Relationship between spatial agglomeration and hotel prices. Evidence from business and tourism consumers. *Tourism Management*, 36, 391–400.

[59] Napierala, T. and Lesniewska, K., 2014. Location as a determinant of accommodation prices: Managerial approach. In: *The 7th World Conference for Graduate Research in Tourism, Hospitality and Leisure*, 3–7 June 2014, Istanbul, Turkey, 687–692.

Biographies

Anshu Zhang received her PhD degree from and is currently a Postdoctoral Fellow in Department of Land Surveying and Geo-Informatics, The Hong Kong Polytechnic University. Her research interest is in spatial data mining, particularly spatial association rule mining and uncertainty issues regarding fuzzy computing, evolutionary computing and data noises. She was the Secretary of Working Group II/1, the International Society for Photogrammetry and Remote Sensing (ISPRS, 2012–2016).

Prof. Wenzhong Shi is the Head and a Chair Professor in Department of Land Surveying and Geo-Informatics, The Hong Kong Polytechnic University. His research interests are in GISci and remote sensing, with focusing on spatial data uncertainty and quality control, spatial data mining, and remote sensing data analytics. He has published over 400 research articles and 11 monographs, and received over 20 international and national academic awards. He is serving editorship in six international journals and was the president of Technical Commission II, ISPRS (2008–2012).