MERBIS - A Self-Adaptive Multi-Objective Evolutionary Rule Base Induction System

Christian Setzkorn, Ray C. Paton

Department of Computer Science, University of Liverpool, UK

Abstract. Supervised classification is a particular data-mining task that forms part of the knowledge discovery process. Its objectives are to extract accurate, comprehensible and interesting knowledge form data. However, many existing supervised classification approaches only focus on one of these objectives. This paper introduces a Multi-Objective Evolutionary Rule Base Induction System called MERBIS that is capable of producing trade-off solutions with regard to the accuracy and comprehensibility objectives. We utilise a superior accuracy measure, problem-tailored genetic operators and a self-adaptive mechanism that reduces the number of parameters. We compare MERBIS with several existing approaches for supervised classification on a number of benchmark data sets and show that it performs comparable while producing more comprehensible classifiers.

1 Introduction

This paper is concerned with supervised classification also known as classifier induction. It involves the determination of a classifier D from data that is capable of assigning objects to a class ω_j from a predefined class set $\Omega = \{\omega_1, \ldots, \omega_M\}$. Objects are often described by features and stored within the data set. Features can be categorical (e.g. colours, yes/no) or numerical (e.g. age).

Classifier induction has the objectives of producing comprehensible, accurate, and interesting knowledge [17, 20] from data and is therefore a multi-objective problem (MOP) that requires the deployment of a multi-objective optimiser (MOO). Usually there is no unique solution for a MOP but rather a set of solutions that represent trade-offs between the objectives [12, 19]. Indeed, there is no universally accepted definition of 'optimum' for multi-objective problems and the (human) decision maker has to decide what (s)he accepts as an optimum [5]. Thus an ideal MOO should be capable of finding as many trade-off solutions as possible [12].

This paper introduces MERBIS, which stands for **M**ulti-Objective **E**volutionary **R**ule **B**ase **I**nduction **S**ystem. As the name suggests, MERBIS applies a multi-objective evolutionary algorithm (MOEA) for the task of classifier induction because MOEAs can produce several trade-off solutions (classifiers) in a single run [5, 6, 12, 29]. To produce several trade-off solutions in a single run has the advantage, that if the preferences of the decision maker change, the search has not to be repeated. In addition, MOEAs can deal with incommensurable objectives [5], search large and complex search spaces [29], and are not susceptible to the distribution of the trade-off solutions [6].

The solutions produced by MERBIS take the form of fuzzy classification rule systems (FCRBs). These are a specific type of symbolic classifier and corresponds to an explicit knowledge representation [2, 37] that can exhibit high comprehensibility [35]. We chose to induce FCRBs, rather than other types of classifiers, because their potential comprehensibility has practical importance. In fact, some researchers argue that only comprehensible classifiers are actually adopted in practice [30, 31, 41]. One reason for this might be that domain experts are very wary and distrustful of incomprehensible results generated by a computer [55].

Although there already exist several single- and some multi-objective evolutionary approaches for the induction of FCRBs, we believe that our approach is novel in several respects. For example, most existing approaches utilise the misclassification (error) rate, or other measures derived from the contingency table, to measure the accuracy of the induced classifiers (e.g. [1, 9, 21, 22, 26, 39, 38, 43, 45, 54]. These measures, however, are inappropriate when the costs of misclassification and the class priors are unknown [16, 42]. This is almost always the case in practice. We therefore use a performance measure originally proposed by Hand et al. [23] that is based on the area under the receiver operating characteristic curve (AUC) (e.g., [4, 16, 24]). Hand et al.'s measure has many advantages because it does not exhibit the above-mentioned shortcomings, works with degrees of memberships (not necessarily probabilities) and can be deployed for multiple class problems. Only a few evolutionary approaches utilise an AUC based measure for estimating the accuracy of induced classifiers. The approach by Holmes et al. [28] is one example. However, as they induce classifier systems, this approach requires post-processing to reduce the number of rules. The approach proposed by Sebag et al. [46] also utilised Hand et al.'s AUC measure. Their approach is called EROL and has been tested on several benchmark data sets. We therefore used this approach for performance comparisons. MERBIS also deploys a self-adaptive scheme which reduces the number of free parameters and hence makes it much more practical. Furthermore we deploy several problem-specific genetic operators.

The remainder of this paper is organised as follows. Section 2 describes how FCRBs can be used to classify objects. Section 3 introduces the MERBIS system. Here we provide details of the chosen representation scheme, genetic operators, and objective functions. Section 4 provides results and we conclude in section 5.

2 Classification using Fuzzy Classification Rule Bases

This section describes how fuzzy rule bases can be used to classify objects. A rule base consists of rules of the following form:

$$IF\ cond_1\ AND\ cond_2\ \ldots\ AND\ cond_n\ THEN\ Class\ =\ \omega_i$$

The antecedent (IF-part) usually consists of conjunctions of conditions. Conditions (e.g. $cond_1$) are sets defined upon the domain of features (e.g. blood pressure = low, greater(income, expenditure)). Conjunctions correspond to the logical operator AND which combines different sets (e.g. blood pressure = low AND smoking = yes). The consequent (THEN-part) associates a class with the antecedents. A rule base can contain several antecedents with the same consequent. This makes it possible to describe multi-modal distributions within the feature space, which can cause problems to other inducers [13, 20, 44]. Antecedents with the same consequents are usually combined using a disjunction (logical OR operator). This type of rule base is also referred to as disjunctive normal form, the most common form of rule bases [13]. Normally an antecedent's conditions correspond to classical sets that use crisp decision thresholds. For example, the set of patients with low blood pressure could be defined as everyone whose blood pressure is exactly below/equal 70 mmHg. However, classical sets like this can lead to unstable systems that may produce very different responses (classifications) to similar inputs (objects) [14, 52]. For example, someone who has a blood pressure of 71 mmHg would not belong to the group of patients with low blood pressure even though her/his blood pressure is only slightly higher than 70 mmHg. One way to tackle this limitation is the use of fuzzy sets [56]. Fuzzy sets can be described by membership functions [33] that assign values between zero and one to each domain value, thus permitting smooth decision thresholds. As fuzzy sets produce values between zero and one, other conjunction and disjunction operators have to be defined to combine these sets. The Min and the $Prod$ operators are often used instead of the AND operator and the Max instead of the OR operator. Fuzzy sets combined via conjunctions define high-dimensional prototypical clusters within the feature space whose boundaries do not need to be axis parallel [40]. This is a further advantage of FCRBs.

3 The MERBIS System

Figure 1 summarises the structure of the MERBIS system. It corresponds to a general evolutionary algorithm whose processes are: genetic operators, fitness evaluation and selection.

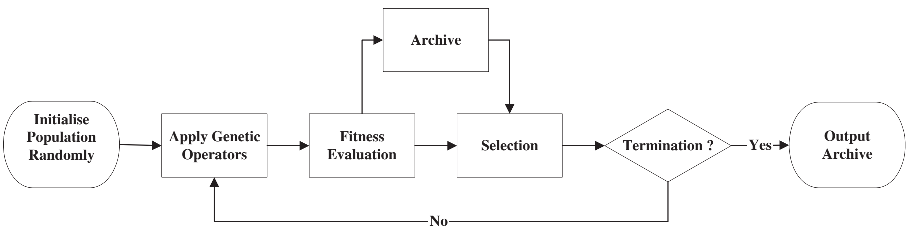


Fig. 1. The structure of MERBIS

In broad terms the system works as follows. Before the genetic operators are applied, a number of candidate solutions (i.e. a population of individuals) is randomly initialised. In our particular case, individuals take the form of fuzzy classification rule bases (FCRBs) (see section 3.1). After this step, genetic operators recombine and/or slightly change a certain number of individuals within the current population (see section 3.4). It follows the fitness evaluation during which each individual's performance (fitness) is determined. The fitness of an individual dependents on its performance on the training data set and its complexity (see section 3.2). The selection process generates a new population of individuals by sampling from the current population and the archive emulating Darwin's principle of the survival of the fittest [11] (see section 3.3). The archive contains the best (elite) individuals that have been found so far. To deploy an archive ensures that the best individuals are preserved, as they can otherwise get lost due to the randomness of the selection process [57]. The use of an archive is a form of elitism, which increases the likelihood of creating better individuals [12] and has long been considered a beneficial component of EAs [32]. The selection process is succeeded by the termination test, which either terminates the algorithm or transmits the current population (generation) to the genetic operators process. This repeats the above-described procedure and it is expected that better and better individuals will be produced over time. If the algorithm terminates (e.g. after a maximum number of generations) the individuals within the current population and the archive are evaluated on a test data. The final output of the system is the updated set of elite individuals within the archive. A detailed description of the system is now presented.

3.1 The Representation Scheme

Figure 2 depicts the structure of an individual. It consists of two parts labelled: *Self-Adaptation Components* and *Rules*. The former is utilised for the self-adaptation described in section 3.5 whereas the latter consists of a number of rules of the form described in section 2 and a confidence value ($CF_1 \ldots CF_n$) that measures a rule's past performance (see formula 4 in section 3.2).

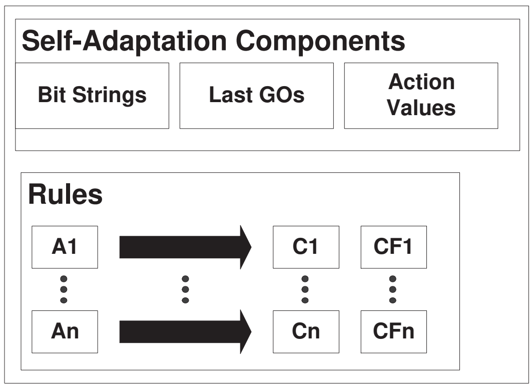


Fig. 2. The utilised representation scheme.

The number of rules is restricted by a maximum value but is not fixed. Each rule's antecedent ($A_1 \ldots A_n$) takes the form of a tree as shown in figure 3. The consequents ($C_1 \ldots C_n$) are numbers representing possible classes.

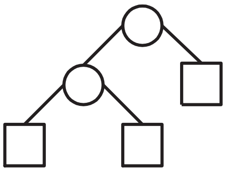


Fig. 3. Example of an antecedent tree.

To use a tree structure for the antecedents corresponds to the representation scheme of Genetic Programming (GP) $[3, 10, 34]$. However, as we do not combine the antecedents in one tree our representation scheme slightly differs from that of GP. We keep the antecedent trees apart to simplify the application of problem specific genetic operators (see section 3.4). The non-terminal nodes (depicted as circles in figure 3) can either be the Min or the $Prod$ operator. The terminal nodes (depicted as squares in figure 3) can be one of the fuzzy sets (membership functions) depicted in table 1.

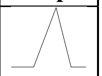
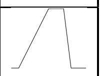
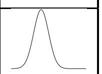
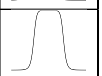
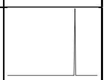
Formula	**Shape**
$MF1(x;a,b,c) = max(min(\frac{x-a}{b-a}, \frac{c-x}{c-b}), 0)$	
$MF2(x;a,b,c,d) = max(min(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}), 0)$	
$MF3(x;a,c) = e^{\frac{-(x-c)^2}{2a^2}}$	
$MF4(x;a,b,c) = \frac{1}{1+\lvert\frac{x-c}{a}\rvert^{2b}}$	
$MF5(x;a,b) = \begin{cases} a & \text{if } x = b \\ 0 & \text{otherwise} \end{cases}$	

Table 1. The membership functions utilised in MERBIS.

The membership functions are defined upon the domain of a feature. Hence, each antecedent tree can be a conjunction of fuzzy sets (see also section 2) and describes a cluster within the feature space. If a feature is categorical the singleton membership function ($MF5$) is used because no order of the attribute's domain values can be assumed. If a feature is numeric either the triangular ($MF1$), trapezoidal ($MF2$), gaussian ($MF3$) or the bell shaped membership function ($MF4$) can be deployed. There are further restrictions that make our representation scheme very problem specific and bias it towards comprehensibility. Each feature can only be used once within an antecedent tree. Furthermore, the size of a tree, although adaptable, is also limited by a maximum number of nodes. The trees are initialised in a top down manner (starting from the root node). A non-terminal node is created with a ninety-percent probability as long as the tree's number of nodes is less than the maximum of allowed nodes. Otherwise a terminal node is created.

Because an individual can contain several rules one could argue that our system resembles a Pittsburgh approach [50] where individuals are made of several rules rather than only one rule as in a Michigan approach [27]. However, because it is also possible that an individual only consist of one rule (or several rules that predict only one class), we would rather describe our system as a hybrid between these approaches.

When an object is presented to an individual each antecedent tree generates an output between zero and one. We have mentioned in section 3.1 that there may be several antecedents with the same associated class (consequent). If this is the case, the maximum value of those antecedents is chosen as the output value for this class. If an individual does not contain an antecedent for a particular class the output value for this class is zero. In summary, the response of an individual to an object is a vector whose dimensionality equals the number of classes to be predicted and each value within this vector indicates the membership degree that an individual associates with the corresponding class. The response vectors to a number of objects (data set) are then used to measure an individual's performance as described in section 3.2.

3.2 The Fitness Evaluation

As mentioned earlier, MERBIS is a multi-objective evolutionary approach capable of optimising several objectives. Unfortunately, there is no universally accepted definition of 'optimum' for multi-objective problems [5] when the relative importance of the objectives is unknown. Thus the definition of an individual's fitness is not as straight forward as in the single-objective case.

Our approach utilises the fitness assignment of SPEA2 [57] which makes use of the Pareto dominance relation and density information to prevent premature convergence of the algorithm. The Pareto dominance relation is the only basis on which an individual can be said to perform better than another in the total absence of information concerning the relative importance of the objectives [18] and it is defined as follows:

Definition 1. ***(Pareto Dominance Relation)*** *A solution x_1 is said to dominate a solution x_2, also expressed as $x_1 \succ x_2$, if x_1 is at least as good as x_2 in all objectives and better with respect to at least one objective.*

The Pareto dominance relation was introduced by Vilfredo Pareto in 1896 building upon the work of Francis Ysidro Edgeworth [7]. Dominating individuals are also called trade-off solutions and are incomparable to one another. This is summarised in the incomparability relation defined as follows:

Definition 2. ***(Incomparability Relation)*** *A solution x_1 is said to be incomparable to a solution x_2, if neither x_1 weakly dominates x_2 nor x_2 weakly dominates x_1.*

The incomparability relation will become important in section 3.5 and it is therefore necessary to define it at this point. Before we explain the actual fitness assignment we would like to introduce the different objectives that are optimised within the current version of MERBIS. There are three objectives, the performance objective and two comprehensibility objectives. They are described now.

The Performance Objective The performance measure (o_1), which has to be minimised, is computed according to formula 1.

$$o_1 = 1 - (\frac{2}{c(c-1)} \sum_{i<j} A(i,j)) \tag{1}$$

Here c denotes the number of classes to be predicted and $A(i,j)$ is computed as follows:

$$A(i,j) = \frac{A(i \mid j) + A(j \mid i)}{2} \tag{2}$$

Both values $A(i \mid j)$ and $A(j \mid i)$ estimate the AUC using the Mann-Whitney-Wilcoxon (MWW) two sample test statistic. This statistic compares two one-dimensional arrays. The first array contains the maximum responses of antecedents with the consequent (class) indicated by the first index (i in $A(i \mid j)$ or j in $A(j \mid i)$) to objects that belong to this class. It therefore contains the response values to signals indicated be the first index. Array two contains the maximum responses from antecedents with the consequent (class) that is indicated by the second index (j in $A(i \mid j)$ or i in $A(j \mid i)$). Thus this array contains responses to noise. If antecedents discriminate between objects from two different classes, the first array should contain (on the average) much higher values than the second array. This can be measured using the MWW statistic. To determine the MWW statistic, the two arrays are merged and arranged in ascending order (without loosing the information of whether an array value originates from array one or two). After this, equation 3 is applied.

$$A = \frac{S_0 - n_0(n_0+1)/2}{n_0 n_1} \tag{3}$$

Here S_0 denotes the sum of the ranks of response values from the first array. The values n_0 and n_1 denote the number of values in array one and two respectively.

As mentioned in section 2, we also determine a certainty degree CF for each rule. It measures the past performance of the rule [8] and is computed according to equation 4.

$$CF = \frac{Sj^k}{S^k} \tag{4}$$

Here Sj^k denotes to the sum of the response values from the k-th rule's antecedent to objects that belong to the class indicated by its consequent and S^k denotes the response values of the k-th rule's antecedent to any object.

The Comprehensibility Objectives According to Ishibuchi et al. [26] the number of rules (o_2) and conditions (o_3) within the FCRB can measure the comprehensibility of a FCRB. Thus both objectives have to be minimised as fewer rules and conditions improve the comprehensibility of a system.

The Fitness Assignment The fitness $F(i)$ of an individual i is computed according to equation 5.

$$F(i) = R(i) + D(i) \tag{5}$$

Here $R(i)$ captures dominance information (see equation 6 and 7) and $D(i)$ captures density information (see equation 8) associated with the i-th individual.

$$R(i) = \sum_{j \in P_t + \overline{P_t}, j \succ i} S(j) \tag{6}$$

$$S(i) = \mid \{ j \mid j \in P_t + \overline{P_t} \wedge i \succ j \} \mid \tag{7}$$

Here P_t and $\overline{P_t}$ refer to individuals from the population and the archive respectively. The expression $i \succ j$ denotes the dominance relation between individual i and j (see definition 1). Equation 6 determines how many individuals the i-th individual dominates within P_t and $\overline{P_t}$. Equation 6 determines the number of individuals which are dominated by the individuals that dominate the i-th individual. If the value of R_i is zero the individual i is non-dominated. The density information is computed according to equation 7 and is an adaptation of the k-th nearest neighbour method [48].

$$D(i) = \frac{1}{\sigma_i^k + 2} \tag{8}$$

The value σ_i^k measures the Euclidean distance between the objective values between the k-th and the i-th individual. The value for k is equal to the square root of the sample size: $k = \sqrt{N + \overline{N}}$ [48]. The value N and $\overline{N}$ denote the number of individuals in the population and archive respectively.

3.3 Selection

The selection process produces a new population of individuals from the current population and the archive. It utilises binary tournament selection [57] to generate a new population. During binary tournament selection, two individuals are picked randomly without replacement from either the population or the archive. The probability that an individual is picked from the archive is determined by the elitism degree (ED). The individual with the lowest fitness value (see equation 5) is declared as the winner and inserted into the new population. If a tie occurs one individual is chosen randomly and inserted into the new generation. This procedure is repeated until the new population has reached the size of the old one. Please note that the ED parameter is adaptable and correspond to the average of the current population's individuals elitism degree values (see section 3.5).

3.4 The Genetic Operators

Genetic operators (GOs) are responsible for the 'movement' of individuals through the search space. We distinguish between GOs that involve one individual (GO1s) or two individuals (GO2s). The GO1s can change the structure of an individual and mimics natural mutation. The GO2s can lead to an exchange of rules or their parts between two individuals and mimic sexual recombination or crossover. We have implemented several GOs of both types because it has been shown that an evolutionary algorithm, which deploys several GOs, can produce to superior results [51]. This was also confirmed empirically in an earlier study of the MERBIS approach [47]. The application of several GOs is now described. The described procedure is applied for the GO1s and GO2s separately.

To choose a genetic operator, we utilise the ϵ-greedy action selection method [53]. More sophisticated methods exist but their assumptions and complexities can make them impractical [53]. Each individual is equipped with an action value for each possible genetic operator (see *Action Values* in figure 2). The action values are adapted over time utilising a reinforcement learning approach that is described in section 3.5. An action value indicates how successful the corresponding genetic operator has been in the past to steer an individual to better parts of the search space. The genetic operator with the highest action value has the probability of $(1-\epsilon)$ to be selected. Each of the remaining n genetic operators can be selected with a probability of ϵ/n. The ϵ-greedy action selection method is one of the simplest reinforcement learning schemes and guarantees that the GO with the highest action value is most often applied while still leaving a low probability for other operators to be selected. This ensures the exploitation of the currently best GO and the exploration of other GOs and hence the search space.

Crossover involves two individuals, and therefore two sets of GO2 action values, we have decided to choose an action value set from either individual with an equal probability. Whether a particular operator is applied depends on the value of the mutation probability (MP) in the case of GO1s and the value of the crossover probability (CP) in the case of GO2s. These values are also adaptable (see section 3.5).

One Individual Genetic Operators The MERBIS approach deploys the following one individual genetic operators:

- The $GO1_1$ operator reinitialises each terminal (fuzzy set) within the FCRB antecedents with probability MP.
- The $GO1_2$ operator reinitialises each consequent of the FCRB with probability MP.
- The $GO1_3$ operator can reinitialise the whole individual with probability MP. If the individual is not reinitialises the antecedents trees and the consequents are reinitialised with the probability MP. Each node of the antecedent tree is examined with respect to MP. If a node is mutated, a new sub-tree is created at this point fulfilling the restrictions described in section 3.1. This operator resembles the standard genetic programming mutation operator as the individual is treated as one tree.
- The $GO1_4$ operator reinitialises the antecedents trees with probability MP after the same principle as described above. This operator cannot reinitialise the consequents and is therefore believed to be less destructive than $GO1_3$.
- The $GO1_5$ operator removes one rule from the FCRB with probability MP. This operator is only applicable if there is currently more than one rule within the FCRB.

- The $GO1_6$ operator adds one rule which is randomly drawn from the archive with probability MP. This operator is only applicable if there are currently less than the maximum number of allowed rules within the FCRB.
- The $GO1_7$ operator removes one rule that exhibits the lowest CF value with probability MP. This operator is only applicable if there is currently more than one rule within the FCRB.
- The $GO1_8$ operator slightly changes an antecedent's condition (fuzzy set) with probability MP. This means that the operator either changes the coverage[1] of a condition's fuzzy set or it moves the fuzzy set either to the left or to the right along the corresponding feature's domain.
- The $GO1_9$ operator mutates the antecedent tree that exhibits the lowest CF value with probability MP.
- The $GO1_{10}$ clones one rule of the FCRB, changes its antecedents fuzzy sets slightly (see $GO1_8$ operator), and reinserts it into the FCRB. This is done with the probability MP and only if there are less than the maximum number of rules within the FCRB.

Two Individuals Genetic Operators The MERBIS approach deploys the following two individuals operators:

- The $GO2_1$ operator performs an exchange (crossover) of sub-trees between two antecedent trees that where randomly chosen from two individuals. The exchange takes place with probability CP. Figure 4 illustrates this. The crossover point is chosen with a uniform probability. Please note that the exchange only takes place if the constraints, described in section 3.1, are not violated.
- The $GO2_2$ operator works after the same principle as the $GO2_1$ operator with the difference that both randomly chosen antecedent tree must have the same consequent.
- The $GO2_3$ operator randomly exchanges one rule between two FCRBs with probability CP.
- The $GO2_4$ operator removes rules from the first FCRB and inserts them into the second FCRB with probability CP. This is done as long as the first rule system contains more than one rules and the second individual's number of rules is less equal to the maximum number of allowed rules.
- The $GO2_5$ operator merges two FCRBs with probability MP. If the resulting FCRBs exceed the maximum number of allowed rules, rules are randomly removed until no constraint violation exists.

3.5 The Self-Adaptation Scheme

In an earlier study we have empirically shown that significant interactions exist between different parameters of our system and that each data set requires different parameter sets [47]. Due to the large number of parameters it is impractical to search for robust parameters for each new data set. We have therefore equipped the current version of MERBIS with a self-adaptive mechanism in order to reduce the number of parameters.

[1] The coverage refers to the interval of domain values to which the membership function of the fuzzy sets assigns values greater than zero.

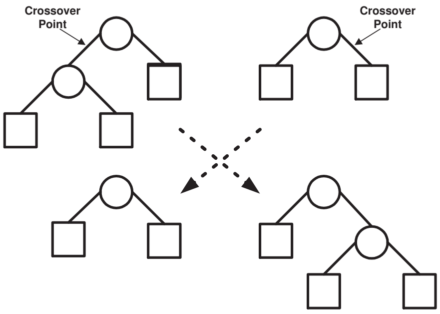


Fig. 4. Crossover between two antecedent trees.

This may also make MERBIS more effective and efficient as the parameters can adapt and are not static during the evolutionary process. To use static parameters may be disadvantageous since different stages of the evolutionary search may require other parameter values [15]. The deployment of a self-adaptive scheme is nothing new, see for example [15, 25, 49] for reviews. We now describe the utilised self-adaptive scheme.

As mentioned in section 3.1 the part labelled '*Self-Adaptation Components*' (see figure 2) is utilised to make most of the parameters adaptive. The part '*Bit Strings*' consists of five 7-bit binary strings that encode the parameters *elitism degree* (ED) (see section 3.6), *crossover probability* (CP), and *mutation probability* (MP) (see section 3.4) and two further parameters: *adaptive mutation probability* (AMP) and *adaptive crossover probability* (ACP). Each of the decoded binary strings can take a value between zero and one. In addition, they can undergo standard bit mutation and single-point crossover during the genetic operators process (see figure 1) whereby the mutation probability is determined by the current value of AMP. Since crossover involves two individuals the crossover probability is the averaged value of the individual's ACP values.

The '*Self-Adaptation Components*' part of figure 2 also contains the element '*Last GOs*'. It is a memory for which genetic operator has produced the individual. This information is exploited by a reinforcement learning mechanism that adapts the probabilities of a particular genetic operator to be applied (see section 3.4).

Reinforcement learning involves the discovery of the right actions that an agent has to take in order to maximise rewards that it receives from the environment over a time period (e.g. [53]). In our particular case, an agent is an individual and to take an action corresponds to applying a genetic operator. As mention in section 3.4, each individual is equipped with an action value for each genetic operator. It determines how likely it is that the corresponding genetic operator is applied to the individual. Since crossover involves two individuals we have decided to choose a set of action values from either individual with an equal probability.

Each time an genetic operator has been applied to an individual the corresponding action value of the individual is updated according to formula 9. This method is appropriated for non-stationary environments [53] and was therefore deployed.

$$Q_{k+1} = Q_k + \alpha[r - Q_k], \tag{9}$$

Here Q denotes an action value at time k or $k+1$, $alpha$ is a constant set to 0.1, and r is a reward. For single-objective problems the reward would be positive if the applied genetic operator has lead to an improvement in fitness. Unfortunately, as we are dealing with a multi-objective problem, the definition of improvement is not as straightforward as for the single-objective case as mentioned in section 3.2. We therefore utilise the Pareto dominance relation and the incomparability relation to determine whether or not the applied genetic operator has lead to an improvement. These relations where defined in section 3.2.

The incomparability relation is used in addition to the dominance relation because some genetic operators can never produce dominating individuals. This is believed to remove biases towards genetic operators that can produce dominating individuals. For example, the sixth 'one individual genetic operator' ($GO1_6$ in section 3.4) can increase the number of rules. Hence an individual produced by this operator could never dominate the original individual because one objective is deteriorated (the number of rules has to be minimised). Still the new individual could be incomparable in comparison to the old one. We assign a value of one to the reward if the newly produced individual is dominating or incomparable in comparison to the old one. Otherwise the reward is zero.

3.6 The Archive

As mentioned in section 3, an archive is an crucial component for a MOEA. However, the deployment of standard archives does not automatically guarantee convergence towards optimal solutions and diversity promotion [36]. Consequently, we utilise a new archive strategy that was originally proposed by Laumanns et al. [36]. It ensures diversity and convergence and in addition limits the size of the archive. Details of this archive strategy are beyond the scope of this paper. The interested reader is rather referred to [36].

4 Results and Discussion

In this section we compare MERBIS with two studies that evaluated different supervised classification approaches on several data sets from the UCI Machine Learning Repository[2]. These studies have been chosen because they deploy AUC based measures to evaluate the performance of the induced classifiers.

As MERBIS generates several trade-off solutions, we have decided to report the solution with the best performance value on the test data as the final output of the system. The system has two parameters, the size of the population and the number of generations. Both parameters where fixed to a value of 100 and 500 respectively. Table 2 compares MERBIS with two approaches reported in [46].

The last three columns contain the average AUC values together with their standard deviations, computed according to Hand et al. [23]. The EROL approach is an evolutionary approach proposed in [46] and SVM is a support vector machine approach. We applied ten-fold stratified cross-validation to obtain the values for MERBIS. It can be seen that MERBIS performs comparable to the other approaches.

[2] http://www.ics.uci.edu/%7Emlearn/MLRepository.html

Data Set	EROL	SVM	MERBIS
Bcw	67.39 ± 5.10	67.19 ± 5.30	97.87 ± 1.89
Crx	81.63 ± 5.60	83.92 ± 4.40	91.75 ± 5.12
German	71.20 ± 3.50	69.03 ± 2.30	74.93 ± 8.05
Promoters	86.26 ± 6.80	97.44 ± 1.60	85.81 ± 7.34
Vehicle	99.45 ± 0.53	99.33 ± 0.72	86.60 ± 2.13
Votes	99.29 ± 0.40	98.86 ± 0.50	98.12 ± 2.53
Waveform	97.07 ± 0.38	96.31 ± 7.80	90.20 ± 1.10

Table 2. A comparison of some existing classifier induction approaches with MERBIS. We report the average AUC values (computed according to Hand et al. [23]) together with their standard deviations.

Table 3 compares MERBIS with three other approaches reported by Fawcett in [16]. As the RL and the C4.5rules approach are capable of producing rules, the average number of generated rules is reported. Please note that all AUC values in table 3 were computed according to Fawcett in order to allow a comparison between MERBIS and the other approaches reported in [16]. The method proposed by Fawcett, for the computation of the AUC values for multiple classes, only slightly differs from that of Hand et al. [23].

	RL			C4.5rules WVote		MERBIS	
Data Set	Best	Rules	Naive Bayes	Results	Rules	Results	Rules
Bcw	97.6 ± 1.3	306.5	93.1 ± 5.5	97.4 ± 3.6	8.2	98.2 ± 2.2	3.9
Car	94.3 ± 1.4	107.6	92.3 ± 2.2	98.3 ± 0.7	78.6	91.7 ± 1.0	13.6
Cmc	63.9 ± 4.0	196.6	64.1 ± 5.8	66.5 ± 4.9	39.1	69.2 ± 2.5	7.9
Crx	90.2 ± 4.2	758.5	87.6 ± 4.3	90.1 ± 3.4	12.9	92.6 ± 2.4	3.5
German	71.9 ± 4.9	807.5	77.1 ± 4.5	67.9 ± 9.6	23.4	76.2 ± 3.2	3.9
Glass	74.4 ± 10.0	183.7	74.0 ± 8.7	75.7 ± 5.9	12.2	87.8 ± 6.7	9.2
Image	93.3 ± 1.4	811.4	95.6 ± 0.9	99.0 ± 0.5	28.6	88.8 ± 2.6	8.6
Kr-v-kp	92.6 ± 1.5	2328.3	95.1 ± 0.8	99.7 ± 0.2	26.3	95.5 ± 2.7	5.8
Mushroom	100.0 ± 0.0	2362.2	99.8 ± 0.1	100.0 ± 0.0	11.5	99.2 ± 0.8	7.8
Nursery	97.1 ± 0.2	606.6	98.0 ± 0.2	99.8 ± 0.1	336.8	94.5 ± 1.4	8.5
Promoters	83.5 ± 16.2	7432.2	97.7 ± 4.0	88.4 ± 12.8	8.0	77.5 ± 11.0	13.0
Sonar	65.8 ± 12.8	10075.7	76.1 ± 13.0	77.8 ± 13.7	9.1	73.0 ± 11.9	6.2
Splice	87.3 ± 1.6	8406.8	99.2 ± 0.6	97.2 ± 0.7	76.2	91.4 ± 2.8	13.1

Table 3. A comparison of some existing classifier induction approaches with MERBIS. We report the avearge number of rules and the average AUC values (computed according to Fawcett [16]) together with their standard deviations.

It can clearly be seen that, although MERBIS does not always produce better results, it generally produces much fewer rules and thus more comprehensible classifiers. This is believed to be much more important in practise [30, 31, 41, 55]. It is anticipated that better results could be achieved if the number of generations and the size of the population is increased.

5 Concluding Remarks

We have introduced a multi-objective evolutionary approach called MERBIS for the induction of fuzzy classification rule systems from data. We have shown that this approach performs comparable to other existing approaches while producing much fewer rules.

References

1. Cosimo Anglano, Attilio Giordana, Giuseppe Lo Bello, and Lorenza Saitta. An experimental evaluation of coevolutive concept learning. In *Proc. 15th International Conf. on Machine Learning*, pages 19–27. Morgan Kaufmann, San Francisco, CA, 1998.
2. C. Apte and S. Weiss. Data Mining with Decision Trees and Decision Rules. *Future Generation Computer Systems*, 13:197–210, 1997.
3. Wolfgang Banzhaf, Peter Nordin, and Frank D. Francone Robert E. Keller. *Genetic Programming : An Introduction - On the Automatic Evolution of Computer Programs and Its Applications*. Morgan Kaufmann Publishers, 1998.
4. A. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.
5. Carlos A. Coello Coello. A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. *Knowledge and Information Systems. An International Journal*, 1(3):269–308, 1999.
6. Carlos A. Coello Coello. An Updated Survey of Evolutionary Multiobjective Optimization Techniques: State of the Art and Future Trends. In *1999 Congress on Evolutionary Computation*, pages 3–13, Piscataway, NJ, 1999. IEEE Service Center.
7. Carlos A. Coello Coello, David A. Van Veldhuizen, and Gary B. Lamont. *Evolutionary algorithms for solving multi-objective problems*. New York ; London : Kluwer Academic, 2002.
8. O. Cordón, M. J. del Jesus, F. Herrera, and M. Lozano. A Proposal on Reasoning Methods in Fuzzy Rule-Based Classification Systems. *International Journal of Approximate Reasoning*, 20:21–45, 1999.
9. O. Cordón, M. J. del Jesus, F. Herrera, and M. Lozano. MOGUL: A methodology to obtain genetic fuzzy rule-based systems under the iterative rule learning. *Int. Journal of Intelligent Systems*, 14(11):1123–1153, 1999.
10. Nichael Lynn Cramer. A Representation for the Adaptive Generation of Simple Sequential Programs. In *Proceedings of an international conference on genetic algorithms and their applications*, 1985.
11. Charles Robert Darwin. *On the origin of species by means of natural selection*. Murray, London, 1860.
12. Kalyanmoy Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley Europe, 2001.
13. Vasant Dhar, Dashin Chou, and Foster J. Provost. Discovering Interesting Patterns for Investment Decision Making with GLOWER - A Genetic Learner Overlaid with Entropy Reduction. *Data Mining and Knowledge Discovery*, 4(4):251–280, 2000.
14. W. Duch, N. Jankowski, K. Grabczewski, and Rafał Adamczak. Optimization and interpretation of rule-based classifiers. In *Intelligent Information Systems*, Advances in Soft Computing, pages 1–14, 2000.
15. Ágoston Endre Eiben, Robert Hinterding, and Zbigniew Michalewicz. Parameter Control in Evolutionary Algorithms. *IEEE Trans. on Evolutionary Computation*, 3(2):124–141, 1999.

16. Tom Fawcett. Using Rule Sets to Maximize ROC Performance. In *Proceedings of the 2001 IEEE International Conference on DataMining*, pages 131–138. IEEE Computer Society, 2001.
17. U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery in databases. In *Advances in Knowledge Discovery and Data Mining*, pages 1–36. AAAI Press / The MIT Press, 1996.
18. Carlos M. Fonseca and Peter J. Fleming. Multiobjective Genetic Algorithms Made Easy: Selection, Sharing, and Mating Restriction. In *Proceedings of the First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, pages 42-52, Sheffield, UK, September 1995. IEE.*, 1995.
19. Carlos M. Fonseca and Peter J. Fleming. Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms—Part I: A Unified Formulation. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 28(1):26–37, 1998.
20. A. Freitas. *Data Mining and Knowledge Discovery With Evolutionary Algorithms*. Springer Verlag, 2002.
21. A. Giordana and F. Neri. Search-Intensive Concept Induction. *Evolutionary Computation Journal*, 3(4):375–416, 1996.
22. A. Gonzalez and R. Perez. SLAVE: A genetic learning system based on the iterative approach. *IEEE Transactions on Fuzzy Systems*, 7(2):176–191, 1999.
23. David J. Hand and Robert J. Till. A simple generalization of the area under the ROC curve for multiple class classification problems. *Machine Learning*, 45:171–186, 2001.
24. J. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic ROC curve. *Radiology*, 143:29–36, 1982.
25. Robert Hinterding, Ryszard S. Michalewicz, and Ágoston Endre Eiben. Adaptation in Evolutionary Computation: A Survey. In *Proceedings of The IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, 1997.
26. H.Ishibuchi, T.Nakashima, and T. Murata. Three-Objective Genetics-Based Machine Learning for Linguistic Rule Extraction. *Information Sciences*, 136:109–133, 2001.
27. John H. Holland. *Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence*. University of Michigan Press, Ann Arbor, 1975.
28. J.H. Holmes, D.R. Durbin, and F.K. Winston. The Learning Classifier System: An Evolutionary Computation Approach to Knowledge Discovery in Epidemiologic Surveillance. *Artificial Intelligence in Medicine*, 19(1):53–74, 2000.
29. J. Horn. Multicriteria Decision Making and Evolutionary Computation.
30. M. Humphrey, S. Cunningham, and I. Witten. Knowledge visualization techniques for machine learning. *Intelligent Data Analysis*, 2:333–347, 1998.
31. John F. Elder IV and Daryl Pregibon. A Statistical Perspective on Knowledge Discovery in Databases. In *Advances in Knowledge Discovery and Data Mining*, pages 83–113. AAAI Press / The MIT Press, 1996.
32. K. A. De Jong. An analysis of the behaviour of a class of genetic adaptive systems, 1975. PhD thesis, University of Michigan.
33. George J. Klir, Ute S. Clair, and Bo Yuan. *Fuzzy Set Theory: Foundations and Applications*. Prentice Hall, 1997.
34. John R. Koza. Genetic programming. In James G. Williams and Allen Kent, editors, *Encyclopedia of Computer Science and Technology*, volume 39, pages 29–43. Marcel-Dekker, 1998.
35. Ludmila I. Kuncheva. *Fuzzy classifier design*. Heidelberg : Physica-Verlag, 2000.
36. M. Laumanns, L. Thiele, K. Deb, and E. Zitzler. Archiving with Guaranteed Convergence And Diversity in Multi-objective Optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 439–447. Morgan Kaufmann Publishers, 2002.

37. Ryszard S. Michalski, Ivan Bratko, and Miroslav Kubat. *Machine learning and data mining : methods and applications*. Chichester : Wiley, 1998.
38. Carlos Andrés Pe na Reyes and Moshe Sipper. Fuzzy CoCo: A Cooperative Coevolutionary Approach to Fuzzy Modeling. *IEEE Transactions on Fuzzy Systems*, 9(5):727–737, 2001.
39. F. Neri and L. Saitta. Exploring the Power of Genetic Search in Learning Symbolic Classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:1135–1141, 1996.
40. Andreas Nürnberger, Aljoscha Klose, and Rudolf Kruse. Analyzing Borders Between Partially Contradicting Fuzzy Classification Rules. In *Proceedings of 19th International Conference of the North American Fuzzy Information Processing Society (NAFIPS 2000)*, pages 59–63, 2000.
41. M. Pazzani, S. Mani, and W. Shankle. Comprehensible knowledge discovery in databases. In *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society*, pages 596–601, 1997.
42. Foster Provost, Tom Fawcett, and Ron Kohavi. The case against accuracy estimation for comparing induction algorithms. In *Proc. 15th International Conf. on Machine Learning*, pages 445–453. Morgan Kaufmann, San Francisco, CA, 1998.
43. W. Romao, A.A. Freitas, and P.C.S. Pacheco. A genetic algorithm for discovering interesting fuzzy prediction rules: applications to science and technology data. In W.B. Langdon and E. Cantu-Paz et al., editors, *Proceedings of Genetic and Evolutionary Computation Conference (GECCO-2002)*, pages 1188–1195, San Francisco, CA, USA, July 2002. Morgan Kaufmann.
44. Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1994.
45. M. Russo. Genetic fuzzy learning. *IEEE Transactions on Evolutionary Computation*, 4:259–273, 2000.
46. Michele Sebag, Jerome Aze, and Noel Lucas. Supervised Learning and Optimizing the ROC Curve, 2003. Conference On Data mining and medical applications, Paris 2003.
47. Christian Setzkorn and Ray C. Paton. MERBIS - A Multi-Objective Evolutionary Rule Base Induction System. Technical Report ULCS-03-016, University of Liverpool, 2003.
48. Bernard W. Silverman. *Density estimation for statistics and data analysis*. Chapman and Hall, 1999.
49. J. Smith and T.C. Fogarty. Operator and parameter adaptation in genetic algorithms. *Soft Computing*, 1(2):81–87, 1997.
50. S.F. Smith. A Learning System Based on Genetic Algorithm, 1980. Ph.D. dissertation, University of Pittburgh.
51. William M. Spears. Adapting Crossover in Evolutionary Algorithms. In J. R. McDonnell, R. G. Reynolds, and D. B. Fogel, editors, *Proc. of the Fourth Annual Conference on Evolutionary Programming*, pages 367–384, Cambridge, MA, 1995. MIT Press.
52. Friedrich Steimann. Fuzzy set theory in medicine. *Artificial Intelligence in Medicine*, 11(1):1–7, 1997.
53. Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning - An Introduction*. MIT Press, 2000.
54. A. Weijters and J. Paredis. Rule induction with a genetic sequential covering algorithm (geseco). In C. Fyfe, editor, *roceedings of the second ICSC Symposium on Engineering of Intelligent Systems (EIS 2000)*, pages 245–251, 2000.
55. S. Weiss and C. Kulikowski. *Computer Systems That Learn*. Morgan Kaufmann, 1991.
56. L. A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.
57. Eckart Zitzler, Marco Laumanns, , and Lothar Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In *EUROGEN 2001 - Evolutionary Methods for Design, Optimisation and Control with Applications to Industrial Problems*, 2001.