Artif Intell Rev (2011) 36:205–248
DOI 10.1007/s10462-011-9212-3

Evolutionary multi objective optimization for rule mining: a review

Sujatha Srinivasan · Sivakumar Ramakrishnan

Published online: 23 March 2011

Abstract Evolutionary multi objective optimization (EMOO) systems are evolutionary systems which are used for optimizing various measures of the evolving system. Rule mining has gained attention in the knowledge discovery literature. The problem of discovering rules with specific properties is treated as a multi objective optimization problem. The objectives to be optimized being the metrics like accuracy, comprehensibility, surprisingness, novelty to name a few. There are a variety of EMOO algorithms in the literature. The performance of these EMOO algorithms is influenced by various characteristics including evolutionary technique used, chromosome representation, parameters like population size, number of generations, crossover rate, mutation rate, stopping criteria, Reproduction operators used, objectives taken for optimization, the fitness function used, optimization strategy, the type of data, number of class attributes and the area of application. This study reviews EMOO systems taking the above criteria into consideration. There are other hybridization strategies like use of intelligent agents, fuzzification, meta data and meta heuristics, parallelization, interactiveness with the user, visualization, etc., which further enhance the performance and usability of the system. Genetic Algorithms (GAs) and Genetic Programming (GPs) are two widely used evolutionary strategies for rule knowledge discovery in Data mining. Thus the proposed study aims at studying the various characteristics of the EMOO systems taking into consideration the two evolutionary strategies of Genetic Algorithm and Genetic programming.

Keywords Rule mining · Evolutionary systems · Multi Objective Optimization · Genetic algorithms · Genetic programming

1 Introduction

Evolutionary algorithms are a class of systems which are inspired by natural evolution. They work on a set of individuals which are the representation of the solution space and tend to

S. Srinivasan (✉) · S. Ramakrishnan
Department of Computer Science, AVVM Sri Pushpam College, Poondi, Tamil Nadu, India
e-mail: ashoksuja03@yahoo.co.in

improve their quality in each generation. Evolutionary Algorithms (EAs) use the reproduction strategies of selection, crossover, and mutation to produce new individuals. The "survival of the fittest" strategy is used to retain only the best individuals in the population and discard worst ones. Evolutionary strategies discover new areas of the solution space which they then exploit to improve the solutions in the future generations. Thus the system is expected to converge to better solutions in each generation.

Association rule induction is an important class of problems for discovering knowledge from data bases. Association rules are used to discover associations between various attributes that describe the data and a special attribute known as the class attribute which is to be predicted for future data sets. Classification rule induction is a sub class of the association rule induction problem and falls under the category of descriptive knowledge discovery. They are comprised of a learning system known as supervised learning. It is a predictive data mining problem where a categorical dependent variable needs to be predicted based on a set of independent variables (Zhao 2007). The rule or set of rules forms a classifier and are used for classification and prediction of unknown records in the future.

The metrics used to evaluate classification rules can be either objective or subjective. The objective measures of rules include support, confidence, coverage Jaccard coefficient, sensitivity, specificity, precision, recall, etc., while rule interestingness, comprehensibility, surprisingness, novelty are some of the subjective measures. The performance of the classification system is evaluated based on one or a subset of the above metrics. In most cases a threshold is specified by the user and rules which satisfy this minimum threshold are chosen for classifying a dataset. Here the system of rule induction and classification becomes one of maximizing or minimizing the measure specified by the user. When one metric is used the problem becomes a single objective one. When more than one measure is used the problem becomes a multi objective problem where multiple objectives are to be optimized. Thus the problem of classification rule discovery becomes a multi-objective problem. Since evolutionary systems are better at dealing with multi-objective optimization problems, they are widely used for classification rule discovery where rules with specific properties are to be discovered, the properties being the objectives to be optimized. The performance of evolutionary multi-objective optimization systems is influenced by many features which include evolutionary features, rule features, and also on the features of the application domain. Evolutionary features that influence the rule induction system are the type of evolutionary strategy used, reproduction operators namely selection, crossover and mutation operators, parameters including crossover rate, mutation rate and population size, chromosome representation, stopping criteria, the fitness function, selection strategy used for choosing individuals for reproduction and the optimization strategy used to select best individuals for the next generation. The rule features include the metrics or objectives used for evaluating and choosing rules, and the rule representation also known as phenotype. The features of the data set include the type of data, size of the data base, number of attributes, type of the class attribute, the number of class attribute values and the domain of application.

The EMOO knowledge discovery system can be further made to perform better by incorporating techniques like fuzzification, parallelization, user interaction, visualization, intelligent agents, and Meta data for Meta heuristics with data mining. The process of integrating two or more techniques for enhancing the performance of the system is known as hybridization. Hybridization not only improves the system performance but also increases the usability of the system. Evolutionary strategies along with Multi objective optimization technique have been integrated with Data mining in particular to discover knowledge in the form of classification rules. The study intends to review the discovery of rule knowledge as an evolutionary multi-objective optimization (EMOO) problem and studies the various parameters which

influence the system. Widely used evolutionary strategies for rule induction are the Genetic algorithms and Genetic programming. Thus, Sect. 2 discusses about EMOO systems for solving multi-objective optimization problems and Genetic algorithms and Genetic programming for EMO rule mining systems. Section 3 discusses the features of the rule including rule types, rule representation and rule metrics. Section 4 gives an overview of various attributes that influence the EMOO rule mining systems. Section 5 reviews Evolutionary Multi Objective Optimization systems for classification rule mining. Section 6 discusses the hybridization of other techniques with evolutionary data mining for enhancing the performance and usability of the system. Section 7 summarizes the study with some future enhancements that can be incorporated into EMOO systems for rule induction.

2 Evolutionary computing for solving multi objective optimization problems

A multi-objective optimization problem is one where two or more quality criteria must be simultaneously optimized. Evolutionary computing is an area of computer science which uses Darwin's principle of evolution to solve problems. This spans a wide area of algorithms including evolutionary algorithms, evolutionary programming, genetic algorithms, genetic programming, swarm intelligence, cultural algorithms, etc. Evolutionary algorithms perform a global search and are convenient for parallelization (Baykasoglu and Ozbakir 2007). They are robust search methods that adapt to the environment and can discover interesting knowledge that will be missed by greedy algorithms (Freitas 2007). Also they allow the user to interactively select interesting properties to be incorporated into the objective function providing the user with a variety of choices (De la Iglesia et al. 2003). Thus Evolutionary algorithms are very suitable for multi-objective optimization since they allow various objectives to be simultaneously incorporated into the solution.

2.1 EMOO systems for rule knowledge discovery

Classification rule mining is a class of mining which comes under association rule mining where association rule mining is applied to a classification problem. The rules produced by the rule mining approach are evaluated using various metrics which are called the properties of the rule. The classification rules have to satisfy various properties to be used as a good classifier. The metrics often used are support and confidence. However there are other properties like comprehensibility and interestingness of the rule that make the classifiers more usable. But the objectives used for evaluation of rules are sometimes conflicting. For example a user may wish to have rules which are both novel and are accurate. These two objectives are conflicting since as accuracy increases the novelty of the rule decreases. Thus the problem of constructing rules with specific properties should be faced as a multi-objective optimization problem where the maximization or minimization of each property is one single objective (Giusti et al. 2008).

The use of an EMO algorithm was proposed to search for Pareto-optimal classification rules with respect to support and confidence for partial classification initially by De la Iglesia et al. (2003), followed by Ishibuchi and Namba (2004). In these systems accuracy and complexity of classification rules were considered for optimization. Genetic algorithm and Genetic programming are two evolutionary strategies that are widely used for evolving the solution space for the multi-objective optimization problems. Both GA and GP use the same evolutionary approaches of reproduction of population to find better solutions. The difference lies in the chromosome representation and the output. In GAs, in general a candidate

solution consists mainly of values of variables i.e., data. By contrast, in GP the candidate solution usually consists of both data and functions (Freitas 2007). Both GA and GP start with a population of individuals and apply the reproduction operators of selection, crossover and mutation to produce new individuals. The fitness of the individuals is then evaluated. The new population is filled in with better individuals using an optimization strategy like elitism to retain the best individuals through generations. Since the worst individuals are discarded in each generation, the solution gets better through iterations. This process of reproduction, fitness evaluation and selection is repeated until a stopping criterion is reached or a good solution is obtained. The output of a GA for rule mining is a set of rules while that from a GP could be an algorithm or program construct for inducing rules.

2.1.1 Genetic algorithm for rule knowledge discovery

Genetic algorithms have been applied to optimization problems since they can span large search space to find diverse sets of solutions. The search space is encoded as chromosomes. The encoding includes binary encoding and value encoding. Binary encoding encodes the data in the search space as a string of binary bits while value encoding is represented as a string of values from the search space which includes real numbers and other data formats. GA selects two parents for crossover, a crossover point is chosen at random and the parts of the string starting at the crossover point are swapped to produce two individuals. Mutation is done just by flipping a bit or by changing the value of an attribute. In the case of rule induction systems the search space is the dataset of records. The chromosome is a binary string of length equal to the product of the number of attributes and the values they take. Generally real valued attributes are discretized before encoding. The output of a GA as a rule induction system is a simple "If…Then" rule for each individual if the Michigan style approach is used, each rule representing a class. In the case of Pittsburg style approach the output is a complex "If…Then…ElseIf" rule which encodes the entire system of knowledge base.

2.1.2 Genetic programming for rule knowledge discovery

GPs work on a more complex programming structure like trees, lists and graphs. The genome of GP individuals is often represented by a variable-size tree genome where the internal nodes contain functions and the leaf nodes contain terminals (Freitas 2007). GP uses the crossover operator on two trees called the parents by swapping a selected sub-tree of one parent with a selected sub-tree of the other. Mutation randomly selects a point in a single tree and replaces the sub-tree starting at that point with a new randomly generated sub-tree. The output of a GP is a rule induction algorithm which can be applied to datasets to induce rules. GPs are said to be more flexible in their representation.

3 Rule features

"If-then rules" which are produced by rule induction algorithms, are considered as one of the highly usable and readable outputs of data mining (Abe and Tsumoto 2008). The two features of the rules that influence the performance of the EMOO system are rule representation and the rule evaluation criteria used. Rules can be applied to data sets for classifying the whole data set or can be applied for partial classification. Partial classification, also known as

nugget discovery, involves the production of accurate yet simple rules (nuggets) that describe subsets of interest within a database. Both classification and partial classification involves mining "If… Then" rules.

3.1 Rule representation

A classification rule is a symbolic representation of knowledge (Giusti et al. 2008). The general format of the rules induced by rule induction algorithm is: "Antecedent → Consequent", the antecedent and the consequent are constructed from attribute tests (ATs). Rule antecedents are conjunctions of ATs and are different from the attributes in the consequent. A record is classified as belonging to a class of interest if it matches the antecedent of any one of the rules selected. The attribute takes either numerical or categorical values. Depending upon the type of the attribute, the attribute test may be any of the following: *Binary partition:* Here the AT is given by "Attribute $\leq$ a numerical value" or "Attribute $\geq$ a numerical value". For example "Salary $\geq$ 50K"; *Range:* Here the AT is of the form Lower bound $\leq$ Attribute $\leq$ Upper bound. For example "20 $\leq$ Age $\leq$ 40"; *Value:* where the AT is given by "Attribute=value". For example "Color = Red"; *Inequality:* where the AT is given by "Attribute $\neq$ value". For example "Color $\neq$ Blue"; and *Subset:* where the AT is of the form Attribute $\in$ any subset of Domain(attribute). For example "Objective $\in$ {Support, Confidence, Precision, Recall}"

Michigan style or Pittsburg style approaches are the most frequently used approaches for representing rules. In the *Michigan style approach* each individual in the population corresponds to classification rules that evolve as a whole and represents an individual and is of the form "Chromosome=Rule". The rule antecedent is a conjunction of attribute tests. For example

$$IF(\text{“}Age = 30\text{”})AND(\text{“}Salary >= 50k\text{”})THEN(Credit\_Status = Good\text{”}).$$

In *Pittsburg Approach*, each individual encodes a complete set of classification rules i.e. "Chromosome = Set of rules" describing the whole data base. In the Pittsburg style approach the rule antecedent is in conjunctive normal form. For example, a Pittsburg style rule may be of the form:

$$IF(Quality = MediumORHigh)AND(Advertisement = YesOR$$
$$Telemarketing = Yes)AND(Gifts = Yes)AND(SalesProfile = GoodOR$$
$$Medium)THEN(Profit = Good)ELSE(Profit = Low)$$

3.2 Rule metrics

Various metrics are used to evaluate the rules. Rule evaluation metrics are classified as subjective and objective measures. Objective metrics can be calculated using definite mathematical formula, while the subjective measure requires the involvement of human experts. The objective measures include support, confidence, coverage, Jaccard coefficient, recall, precision, Jmeasure, etc. to name a few. The subjective measures include surprisingness of a rule, interestingness, novelty, peculiarity and comprehensibility or complexity of a rule. In most cases subjective measures have often been converted into objective measures by using mathematical formula. Researchers have developed more than forty objective indices based on number of instances, probability, statistics values, information quantity, distance or attributes of rules, and complexity of rules. A list of these measures can be found in Abe and Tsumoto (2008), where correlation analysis of the objective metrics are carried out.

4 Issues in EMOO for rule mining

The rule knowledge discovery system is influenced by various factors including the initialization of the rules population, chromosome representation (Genotype, Phenotype), parameters including population size, number of generations, number of rule evaluations per generation, stopping criteria, rule width, cross over probability, mutation probability, operators used for reproduction including selection for reproduction, crossover, mutation, selection of population for next generation i.e. MOO criteria for optimization, objectives optimized, fitness function, type of data used, number of class attributes, and area of application.

Del Jesus et al. (2005) state that the application of a GA to solve a problem must determine: (a) A genetic representation of the search space (called genotype) and of the solutions of the problem (called phenotype); (b) A way to create an initial population of solutions; (c) An evaluation or fitness function which provides a quality value to each chromosome; (d) Operators which modify the genetic composition of the descendants during reproduction; and (e) Values for the parameters used (population size, probabilities of application of genetic operators, etc.).

4.1 Chromosome representation (Genotype/ Phenotype)

The gene representation has immense influence on the performance of the algorithm. It is said to improve the predictive accuracy of classification rules (Baykasoglu and Ozbakir 2007).

In GAs, the chromosomes are encoded as binary strings or value strings. While most of the systems use fixed length chromosome other variable length representations are also found to be useful. GP's chromosomes are represented as trees, expression trees, graphs and lists. The length of the chromosome is variable.

4.2 Parameters

The parameters that influence the performance of the algorithm are, *Population size,* i.e. the number of individuals in the mating pool, *number of generations* represents the number of iterations of the rule induction or rule selection system, *Crossover probability* which is the probability of creating a new individual via crossover of selected individuals, *Mutation probability* which is the probability of creating a new individual via mutation based on a selected individual and the *Stopping criteria* which may be specified by the user if a satisfactory solution is found or it can be the number of generations.

4.3 Reproduction operators

Selection for reproduction or mating requires that two chromosomes be selected from the population and crossover or mutation is applied. Three types of selection mechanisms used in EMOO for rule mining are *Roulette wheel selection* where each individual is given a chance to become a parent in proportion to its fitness evaluation; *Tournament selection* in which a group (2 in most cases) of parents is selected and a tournament is held to decide which of the individuals will be the parent; *Fitness ranking* where individuals are sorted in order of raw fitness and given ranks. Good ranked individuals are chosen as parents**.***Crossover* is applied to the selected parents by choosing crossover points. Depending upon the crossover points, there are *one point* crossover, *two point* crossovers and *uniform* crossover. *Mutation* makes change to the gene of the chromosome by changing the value of an attribute if value encoding is used or flipping a bit if binary representation is used. The mutation point in

the chromosome is chosen randomly or through heuristics. In GP sub-trees are swapped or replaced by crossover or mutation.

4.4 Objectives (rule metrics) taken for optimization

Classification accuracy is by far the criterion most used in fitness functions for evolving classification rules (Freitas 2007). However the user would like to have more than one objective for a rule. The objectives are the rule metrics used for evaluating the rule which may be objective or subjective measures. An objective metric can be defined by a specific mathematical formula with less or no user involvement, while subjective measures are more user-dependent and requires an expert in evaluation. In most cases the objectives are represented as a vector of fitness or rank values of the metrics for each rule. These vectors are compared with each other for choosing the best solution using an optimization strategy.

4.5 The fitness function and the Optimization strategy

In a multi-objective evolutionary algorithm, the fitness function accounts for all objectives simultaneously and is related to the relative non-dominance of a candidate solution (Zhao 2007). Three approaches are discussed by Freitas (2004) to handle multi objective optimization problems. These include (a) transforming the original multi-objective problem into a single-objective problem by using a weighted formula; (b) the lexicographical approach, where the objectives are ranked in order of priority; and (c) the Pareto approach, which consists of finding as many non-dominated solutions as possible and returning the set of non-dominated solutions to the user. A critical review of these three approaches is provided (Freitas 2004), and it is concluded that the weighted formula approach is an ad-hoc approach while the lexicographic and Pareto approach are principled approaches. There are other non-pareto approaches which use methods like ranking composition. The individuals are given a set of ranks for various metrics and the ranks are combined by different ranking composition methods like median, arithmetic mean, harmonic mean, Condorcet, and weighted mean (Giusti et al. 2008).

In fitness assignment, most EMOO algorithms fall into two categories, non-Pareto and Pareto-based. Non-Pareto methods use the objective values as the fitness value to decide an individual's survival. The individuals in the population with high fitness values are regarded as fittest regardless of their single objective values (Dehuri and Mall 2006). Most of the EMOO algorithms for rule induction are extensions of either Non-dominated Sorting Genetic Algorithm (NSGA II) (Deb et al. 2002) or Strength Pareto Evolutionary Algorithm (SPEA2) (Zitzler et al. 2001) both of which use Pareto dominance for finding a set of good solutions.

NSGA II

NSGA-II was proposed for multi objective optimization by Deb et al. (2002) and uses non-dominated sorting as a mechanism for introducing elitism in the search along with a crowding operator to ensure diversity of solutions within the Pareto front. A population of individuals is created. This initial population of individuals is sorted into fronts according to non domination with respect to the multiple objective functions. The multiple objectives are represented as a vector. Solutions within the same front are then sorted according to crowding distance. Solutions non-dominated by others are known as the Pareto-front and are given priority for reproduction. The cycle of selection and reproduction using crossover and mutation called a generation creates a new population which is merged with the initial population.

SPEA 2

SPEA 2 by Zitzler et al. (2001) uses an initial population and an empty archive. Starting with these, all non-dominated population members are first copied to the archive; any dominated individuals or duplicates are removed from the archive during this update operation. A clustering technique which preserves the characteristics of the non-dominated front is used. Fitness values are assigned to both archive and population members. The fitness of an individual in the population is calculated by summing the strength values. In the mating selection phase individuals from the union of population and archive are selected by means of binary tournaments. Finally, after recombination and mutation the old population is replaced by the resulting offspring population. A comparative study of the evolutionary strategies of NSGA II and SPEA 2 for multi objective optimization using GP can be found in Zhang and Rockett (2007).

4.6 Data sets and area of application

Classification can be done on any type of data sets belonging to any area of application. The data sets may include information collected from various disciplines including physics, chemistry, biology, medicine, finance, human decision making, pattern recognition, space research, electricity problems, etc. Most of the algorithms for classification rule mining have been applied on bench mark data sets available in the University of California at Irwin (UCI) machine learning repository (Newman et al. 1998). Other data sets from various local repositories have also been used.

5 Evolutionary multi objective optimization systems for rule mining

Although there are a variety of evolutionary strategies used in solving various problems, the most frequently used strategies in rule mining are GAs and GPs. A review of Evolutionary algorithms in Data mining is presented in Freitas (2007). The review gives a brief overview of EAs, focusing mainly on two kinds of EAs, viz. Genetic Algorithms (GAs) and Genetic Programming (GP). Also the main concepts and principles used by EAs designed for solving several data mining tasks of discovery of classification rules, clustering, attribute selection and attribute construction is discussed. Multi-Objective EAs, based on the concept of Pareto dominance, and their use in several data mining tasks is given special attention. In the following section Multi Objective Optimization systems are reviewed under the two evolutionary strategies of Genetic Algorithm and Genetic Programming taking into consideration the rule mining task.

5.1 EMOO systems which use GA

In data mining, nugget discovery is the discovery of interesting classification rules that apply to a target class. De la Iglesia et al. (2003), propose the use of multi-objective optimization evolutionary algorithms to allow the user to interactively select a number of interest measures and deliver the best nuggets.

Chromosome representation: The solution is represented as a conjunctive rule called a nugget. A binary string is used for this as follows. The first part of the string is used to represent the numeric fields or attributes. Each numeric attribute is represented by a

set of Gray-coded lower and upper limits, where each limit is allocated a user-defined number of bits, p (p = 10 is the default). There is a scaling procedure that transforms any number in the range of possible values using p bits [0, $2^p - 1$] to a number in the range of values that the attribute can take. The second part of the string represents categorical attributes, with each attribute having v number of bits, where v is the number of distinct values that the categorical attribute can take. If a bit assigned to a categorical attribute is set to 0 then the corresponding label is included as an inequality in one of the conjuncts.

Parameters: The *initialization* procedure uses mutated forms of the default rule as initial solutions. The default rule is the rule in which all limits are maximally spaced and all labels are included. In other words, it predicts the class without any pre-conditions. For the initialization, all solutions in the population pool are initialized to be copies of the default rule and then some of the bits mutated according to a parameter representing the probability of mutation. The *Population size* has been varied from 100 to 160 in steps of 10. The *number of generations* can be set by the user and varied from 50 to 100 in steps of 10. *Crossover rate* ranges from 60% to 90% in steps of 10% while two types of *mutation rates* are used. Initial mutation rate affects the mutation of individual bits when solutions are created initially. Mutation also occurs when solutions are reproduced to form new solutions. It has been varied between 0% and 4% in steps of 1%. Number of generations is used as the *Stopping criteria*.

Reproduction operators: Binary tournament *selection* and one point *crossover* are used while *mutation* involves flipping of randomly selected bit.

Objectives, fitness function and the Optimization strategy: The proposed system is said to present to the user a number of measures of interest from which some measures can be selected. Pareto-based MOEA is used to deliver nuggets that are in the Pareto optimal set according to some measures chosen by the user. Accuracy and coverage are taken as measures of interest. In the main function, a child population is created at each stage using binary tournament selection, single-point crossover and mutation. Both populations are combined and the resulting population of size 2N (where N is the size of the initial population) is then sorted according to non-domination into different fronts. Hence solutions that belong to the first front are non-dominated solutions. Once all the data has been examined the measures of interest, namely the coverage and accuracy are calculated for each nugget. A fitness measure has also been introduced based on a parameter λ. This can order rules according to accuracy and coverage measures so that rules with differing degrees of coverage and accuracy can be obtained. The NSGA II Pareto-based MOEA is used as optimization strategy.

Data sets and area of application: The databases used for these experiments were extracted from the UCI repository (Newman et al. 1998) and includes Adult, Mushroom, and Contraception data sets which fall under biology domain.

However, the drawbacks of the algorithm as stated by the authors are that, only accuracy and coverage can be taken into consideration with the present approach. To optimize nuggets in terms of simplicity, for example, a post-processing algorithm has to be run which removes conjuncts in a greedy manner choosing the one that produces no deterioration in accuracy, one at a time.

As an extension of De la Iglesia et al. (2003), a combination of innovative approaches to rule induction to encourage the production of interesting sets of classification rules is

studied in De la Iglesia et al. (2005). These include multi-objective metaheuristic to induce the rules, measures of rule dissimilarity to encourage the production of dissimilar rules and rule clustering algorithms to evaluate the results obtained. Their previous implementation of NSGA-II for rule induction produces a set of coverage-confidence (cc) optimal rules. Among the set of rules produced, there may be rules that are very similar. Therefore the concept of rule similarity is explored. Taking this into consideration, experiments with a number of modifications of the crowding distance to increase the diversity of the partial classification rules produced by the multi-objective algorithm have been carried out. Novelty of the rules in relation to other rules within the set are studied. As defined in their earlier work cc-optimal rules are selected. However it is stated that the problem with cc-optimality as a criterion to choose rules is that, if two rules have the same confidence and coverage, only one of them may be kept in the cc-optimal set. But the two rules could be very different either in attribute space or they could describe a different subset of records. In such cases it is stated that the cc-optimal rule set may not be suitably representative of the most interesting rules underlying the database. Therefore the authors investigate this claim. In order to assess these factors, the set of rules produced by the algorithm are analyzed in terms of similarity or rather dissimilarity of rules within the set. Dissimilarity in the sets of records that 'match' different rules is considered. Jaccard coefficient is used for calculating rule dissimilarity.

In order to understand the results of applying distance metrics to the rules obtained by the NSGA-II algorithm, a clustering algorithm is also applied to cluster similar rules together which help in the presentation of results. Two of the algorithms: Partitioning Around Medoids (PAM) and AGlomerative NESting (AGNES) are used for this purpose. Both algorithms work on a pre-prepared dissimilarity matrix which contains the distance between each pair of rules calculated using the Jaccard coefficient. The crowding measure is modified to be a count of the number of rules within a certain threshold distance, T, from the rule being examined, according to the Jaccard dissimilarity measure.

As further extension of De la Iglesia et al. (2003, 2005), Reynolds and De la Iglesia (2006) present two enhancements to the algorithm, describing how the use of modified dominance relations may increase the diversity of rules presented to the user and how clustering techniques may be used to aid in the presentation of the potentially large sets of rules generated. They have used an enhanced chromosome representation as follows.

Chromosome representation: The algorithms described in this paper produce partial classification rules of the form *antecedent* $\rightarrow$ *consequent;* where both the antecedent and the consequent are constructed from *attribute tests* (ATs). Each rule created has the same consequent, consisting of just one AT. This defines the class of interest. Rule antecedents are conjunctions of ATs, none of which may share the same attribute as the consequent. In the new representation used in this work, a rule is stored as an array of ATs. Values that occur in both the numeric and categorical fields of the training set are stored in reference arrays. Indices into these arrays are then used in the representation of the chromosomes. As it is possible to have more than one AT for a given field, a variable length representation is used.

Parameters: The *Initial Population* is constructed at random using at most three ATs, though rules produced in later generations were permitted six ATs. The *Population* consisted of 200 rules and the *Number of generations* was fixed at 150. A *Crossover rate* of 0.2 and a m*utation rate* of 0.2 were used. High mutation rate is justified by the fact that mutation occurs on ATs rather than binary strings. Number of generations was used as s*topping criteria.*

Reproduction operators: Binary tournament *selection* has been used for selecting parents for reproduction. Both mutation and crossover is said to occur on an AT by AT basis with*uniform crossover*. Here, each AT is equally likely to be assigned to either child. The randomly generated bits indicate which child should take the associated AT. After the application of crossover and mutation, rules are simplified if possible, for example by removing redundant ATs. Finally, if a simplified rule exceeds the AT limit, further ATs are removed at random.

Mutation is performed on each type of AT as follows: *Categorical value and Categorical inequality:* Represented by the categorical field number and the category index. A mutation changes the category index to a randomly selected value. *Categorical subset:* Represented by the categorical field number and a bit-string indicating which categories are permitted. A mutation either adds a category to the permitted list or removes one by flipping a bit. *Numeric binary partition:* Represented by the numeric field number, the index of the bound value and a flag indicating the type of bound. A mutation changes the index of the bound by up to 20% of the number of values that occur in the database, while ensuring that the AT does not become trivial or impossible to satisfy. The type of the bound is not changed. *Numeric range:* Represented by the numeric field number, the index of both the lower and upper bound and two Booleans indicating whether each bound is present. A mutation changes a valid bound in the same way as for a binary partition AT. Mutations may also remove an AT entirely or add a new one.

Objectives, fitness function and the Optimization strategy: Support, confidence and coverage are used as objectives to be optimized. Novelty is also taken as one of the measures. Simple Optimality Criteria and dominance using pc-dominance and cc-dominance are used as Meta heuristic. Further the algorithm discussed in this paper finds rules that satisfy certain constraints: ATs may be constrained to be of particular types and there may be a limit on the number of ATs permitted. A pc-optimal (or cc-optimal) rule is one that is not dominated by any other rule that satisfies the constraints. It is argued that the use of cc-dominance may result in interesting rules being dominated, while using pc-dominance results in the generation of very large sets of non-dominated rules.

Three methods based on the support sets of the rule have been developed for evaluating the 'novelty' of rule q, with respect to rule r. The first of these is referred to as the *absolute novelty,* the second, *relative novelty* and a third measure known as the a*pparent rule novelty* is introduced. To measure apparent or syntactic novelty, a model of the user's knowledge of the data may be used to produce his estimate of either absolute or relative novelty. A modified dominance relation using dominance margin is introduced which allows the user to be presented with a more diverse set of rules, but without allowing the size of the rule set to grow excessively. This dominance margin is used to modify the dominance relation to encourage novel rules.

The multi-objective genetic algorithm, NSGA II, used in this paper requires only a change in the form of crowding measure to define a multi objective metaheuristic. The standard crowding method in NSGA II is applied to one front at a time, using only the objective values of the solutions. The overall crowding distance is given by the sum of the objective distances. If a solution ever appears at an end of the list, it is assigned a large crowding distance to indicate no crowding. pc-dominance and cc-dominance are used to remove uninteresting rules.

Data sets and area of application: The algorithm is applied on Adult data set and the Contraceptive Method Choice Database (CMC), from UCI machine learning repository (Newman et al. 1998).

Reynolds and De la Iglesia (2009) discuss how to adapt a MO algorithm for the task of partial classification. Additionally, they introduce a new MO algorithm for this task based on a metaheuristic known as greedy randomized adaptive search procedure (GRASP). The resulting algorithm is guided solely by the concepts of dominance and Pareto-optimality.

Chromosome representation: While previous work by the same authors used a fixed length bit-string representation for rules, the work described in this paper uses an alternative representation. A rule is stored as an array of antecedent ATs. Since it is possible to have more than one AT for a given field, a variable length representation is used. A reference array is created for each field, containing the values that occur within the dataset. Rather than directly encoding the field value in the AT representation, the index of the value in the reference array is used instead.

Parameters: The algorithm starts with the default rule in the *initial population*, that is, the rule with no ATs in the antecedent. The greedy component of the MOG is used to iteratively add 'restrictions' to a rule. Restrictions that may be added depend on the types of ATs being used, as follows: For n*umeric* ATs a restriction may be the tightening of either a lower or an upper bound. If there are no ATs that specify a lower bound on the field in question in the original antecedent, tightening the lower bound implies the addition of a new AT. Otherwise, the AT with the lower bound is merely modified. For *value categorical ATs* a restriction is the selection of a category, resulting in the addition of a new AT. For *inequality categorical ATs* a restriction is the elimination of a category and adding a new AT. For *subset categorical ATs* both the selection and the elimination of categories are potential restrictions. This leads to the addition of an AT if the value of the categorical field is unconstrained in the original antecedent or the modification of an AT otherwise.

Population sizes of 10, 20, 50, 100, 200, and 500 individuals have been used while *Crossover rate* is varied from 0 to 100% in steps of 20. The *Mutation rate* is varied from 0 to 20%. 60 runs have been performed for each combination of settings forming a *generation* where *number of rule evaluations per generation* was set to a fixed number of 50,000 evaluations per generation. The number of generations was used as *stopping criteria*.

Reproduction operators: Uniform crossover has been used while *Mutation* is carried out by the neighborhood operator used by the local search phase. This may perform any of the following three tasks: remove a randomly selected AT, adds a new AT generated at random, or mutate a randomly selected AT.

Objectives, fitness function and the Optimization strategy: Support, confidence and coverage are used as objectives to be optimized. Multi objective GRASP which can be described as a multi-start algorithm is used for optimization. Each iteration of the algorithm consists of two phases: (1) the application of a randomized greedy algorithm, usually constructive in nature, and (2) the subsequent application of local search to improve the solution constructed. The first phase is used to construct solutions one element at a time. At each step, candidate elements are evaluated and ranked according to a greedy evaluation function. A restricted candidate list (RCL) is created, consisting either of the best n elements, where n is fixed, or of those elements that meet a quality threshold. The element to be added to the solution is then

selected at random from the RCL. In an alternative approach used in this work, an initial front of solutions is generated before the application of local search. Local search is then applied to the front as a whole, eliminating the need for weighted utility functions in the local search. Repeating this procedure of front generation and optimization produces the algorithm.

Data sets and area of application: Adult database and the Forest cover type database from the UCI machine learning repository (Newman et al. 1998), both of which come under biology are used for testing.

As an extension to the previous algorithm by Reynolds and De la Iglesia (2009), Reynolds et al. (2009) describe the application of a multi-objective GRASP to rule selection, where previously generated simple rules are combined to give rule sets that minimize complexity and misclassification cost. This paper also investigates a range of multi-objective approaches for creating this initial rule set and the effect on the quality of the resulting classifier. In this work the authors adapt the GRASP for the problem of rule selection. The required diversity in the initial rule set is produced by using a multi-objective rule induction algorithm, provided the dominance relation is modified in a way that encourages such diversity. For example by utilizing novelty measures and dominance margins as explained in the previous work.

The rule selection algorithm does not need access to the data or to the details of the rules. It only requires access to the following information: which rules match which records; which records are in the class of interest; and the complexity of each of the rules. Once the data and rules have been read in, a match table containing this Meta data is created. The algorithm then need only refer to the match table. The (correct) match count of a rule is defined as the number of remaining records that (correctly) match the rule. To significantly reduce the number of updates required, match counts are updated lazily, i.e. only when the rule in question is being considered for addition. Rules are grouped in buckets for each level of complexity. Within each bucket, rules are ordered according to the count of correct matches. When using cost thresholds, the simplest rules are scanned first, with equally complex rules scanned in decreasing order of the count of correct matches. In the complexity limit approach, rules are scanned in order of decreasing match count. Rules with the same counts are scanned in order of increasing complexity. Once it becomes clear that further scanning cannot result in a better rule, scanning is halted. Rules that result in too many false positives to improve the rule set, regardless of how many false negatives are eliminated and are removed from consideration.

Narukawa et al. (2005) have examined three methods for improving the search ability of the NSGA-II algorithm to find a variety of non-dominated rule sets of a three-objective fuzzy rule selection problem. Three methods are examined to achieve this goal. One is the *removal of overlapping rule sets* in the objective space, another is the *selection of similar rule sets* as parents for crossover, and the other is the *selection bias* toward rule sets with high accuracy. It is experimentally proved that the performance of the NSGA-II algorithm was improved by removing overlapping solutions in terms of the variety of obtained non-dominated rule sets. Also the choice of extreme and similar parents is said to have driven the population toward rule sets with high classification accuracy. Two problem specific heuristics are used to efficiently decrease the number of fuzzy rules in each rule set Sduring the execution of the NSGA-II algorithm. One is biased mutation probabilities where a larger probability is assigned to the mutation from 1 to 0 than that from 0 to 1. The other is the removal of unnecessary fuzzy rules.

Chromosome representation: Candidate rules are represented by a binary string of length N as $S = s1s2...sN$ where $sj = 1$ means that the jth candidate rule is included in S and $sj = 0$ *means* jth candidate rule is excluded from S.

Parameters: The *initial population* is constructed using different solutions in the objective space. 300 fuzzy rules are chosen for each class as candidate rules for multi-objective fuzzy rule selection. A pre-specified number of promising candidate fuzzy rules are selected from those short fuzzy rules using a heuristic rule evaluation measure known as the SLAVE measure. The p*opulation size* consists of 200 strings, the *number of generations* being 5000 with a *Crossover rate* of 0.8 and *biased mutation probabilities* of $p\text{m}(0\rightarrow 1) = 1/300M$ and $p\text{m}(1\rightarrow 0) = 0.1$, the number of generations is used as the s*topping criteria*.

Reproduction operators: A *similarity-based selection scheme* is used to select similar parents. In this mating scheme, the first parent, the candidate with the highest classification accuracy is selected by the *binary tournament selection* scheme in the same manner as in the NSGA-II algorithm. On the other hand, its mate is chosen in the following manner. First β candidates are selected by iterating the binary tournament selection scheme β times. Then the most similar candidate to Parent A is selected as Parent B from the β candidates. The similarity is calculated by the Euclidean distance between the parents in the objective space. The selection bias toward similar parents is adjustable by the value of β in this mating scheme. *Uniform crossover* and *biased mutation,* where a larger probability is assigned to the mutation from 1 to 0 than that from 0 to 1 is used.

Objectives, fitness function and the Optimization strategy: Accuracy and complexity are taken as objectives to be optimized. The accuracy of each fuzzy rule-based classification system is measured by the number of correctly classified training patterns while its complexity is measured by the number of fuzzy rules and the total number of antecedent conditions. NSGA II is used as the optimization strategy.

Data sets and area of application: Wisconsin Breast cancer, Diabetes, Glass, Heart C, Sonar, and Wine data sets from UCI machine learning repository (Newman et al. 1998) are used for experimentation.

Ishibuchi and Nojima (2005), compare fuzzy rules with interval rules through computational experiments on benchmark data sets from the UCI database (Newman et al. 1998) using an evolutionary multi-objective rule selection method. In the design of fuzzy and interval rule based systems for classification problems, they have used three types of partitions: homogeneous fuzzy partitions, inhomogeneous entropy-based interval partitions, and inhomogeneous fuzzy partitions derived from the interval partitions. Using each type of partition, a pre-specified number of candidate rules are generated based on a heuristic rule evaluation measure. Then an evolutionary multi-objective optimization algorithm is used to find a large number of non-dominated rule sets with respect to the three objectives from the candidate rules. By examining the classification performance of obtained non-dominated rule sets, they compare the three types of partitions.

Chromosome representation: If-then rules are used for n-dimensional pattern classification problem where the Rule is given by $Rq : If x1 is Aq1 and \ldots and xn is Aqn then Class Cq$ *with CFq*, where Rq is the label of the q-th rule, **x** € (x1, …, xn) is an n-dimensional pattern vector, Aqi is an antecedent fuzzy set or interval, Cq is a consequent class, CFq is a rule weight, and N_{rule} is the number of fuzzy rules. The consequent class Cq and the rule weight CFq of each rule Rq are specified from compatible training patterns with its antecedent part **Aq** € (Aq1, …, Aqn) in a heuristic manner. The rule weight CFq is used as the strength of Rq when new patterns are to be classified by the rule-based system with the N_{rule} rules.

Parameters: The evolutionary multi-objective rule selection method consists of two stages: candidate rule generation and genetic rule selection. In the *Candidate Rule Generation* stage, a pre-specified number of promising rules are chosen from possible rules based on a heuristic rule evaluation measure called the SLAVE measure to form the *initial population*. The algorithm is applied on a *population size* of 200 strings, with 5,000 generations, a *crossover rate* of 0.8 and biased *mutation probabilities*, pm $(0 \rightarrow 1) = 1/300M$ and $p\text{m}(1 \rightarrow 0) = 0.1$, with number of generations being used as *stopping criteria*.

Reproduction operators: Uniform crossover and *biased mutation* where a larger probability is assigned to the mutation from 1 to 0 than that from 0 to 1 are used.

Objectives, fitness function and the Optimization strategy: The objectives consist of maximizing the number of correctly classified training patterns; minimizing the number of fuzzy rules; and minimizing the total number of antecedent conditions of fuzzy rules. Candidate rules are extracted from training patterns (90% of the whole data set) for each class using one of the three types of partitions. Then the NSGA-II algorithm is applied to the candidate rules to find a number of non-dominated rule sets.

Data sets and area of application: The algorithm is tested on the following data sets from UCI machine learning repository (Newman et al. 1998): Wisconsin Breast cancer, Diabetes, Glass, Heart C, Sonar, and Wine data sets.

Ishibuchi (2007) as an extension of their previous work (Ishibuchi and Nojima 2005); explain two approaches to evolutionary multi-objective classification rule mining. One is to search for Pareto-optimal rules and the other is to search for Pareto-optimal rule sets. The authors also demonstrate the usefulness of evolutionary rule selection as a post-processing procedure in the second phase of classification rule mining where classification rule mining is considered as a two step process. The first step is inducing rules and the second step where best rules are selected. They discuss the application of evolutionary multi-objective optimization (EMOO) to association rule mining. Especially, attention is focused on classification rule mining in a continuous feature space where the antecedent and consequent parts of each rule are an interval vector and a class label, respectively. The relation between Pareto optimal rules and Pareto-optimal rule sets in the classifier design are also examined. The same *Chromosome representation* as in their previous work is used.

Parameters: The *initial population* consisted of candidate classification rules with three or fewer antecedent conditions using pre-specified values of the minimum support and confidence extracted using an association rule mining technique. The *minimum support* threshold being 1, 2, 5, and 10%, and *minimum confidence* threshold being 60, 70, 80, and 90%. All the extracted classification rules for each combination of the two threshold values are used in evolutionary rule selection as candidate rules. NSGA-II was executed with a *population size* of 200 strings, 1,000 *generations*, with *crossover probability* of 0.9, *biased mutation probability* of 0.05 (for $0 \rightarrow 1$) and $1/N$(for $1 \rightarrow 0$) where $N$is the string length, and number of generations being the *stopping criteria.*

Reproduction operators: Uniform crossover and a *biased mutation* where a larger probability is assigned to the mutation from 1 to 0 than that from 0 to 1 have been used.

Objectives, fitness function and the Optimization strategy: The following three objectives have been considered for optimization; the number of correctly classified training patterns

by S, the number of selected rules in S, and the total number of antecedent conditions over selected rules in S. In NSGA-II and SOGA (Single Objective GA), a problem-specific heuristic procedure for decreasing the number of rules in each string have been used. The classification of each training pattern is based on a single-winner scheme, i.e., some rules in a string are used for the classification of many patterns while others are used for the classification of no patterns. Such useless rules are removed without degrading the classification accuracy which at the same time improves the second and third objectives.

Data sets and area of application: The following data sets from the UCI machine learning repository (Newman et al. 1998), were used in computational experiments: Wisconsin Breast cancer, Car, Glass, Heart C, Iris, Letter, Nursery, Sonar, Soybean L, Tic-tac-toe, Vote, and Wine.

As an extension of Ishibuchi et al. (2007), Ishibuchi (2007) proposes an evolutionary multi-objective approach to the design of accurate and interpretable fuzzy rule-based systems and have applied multi-objective genetic fuzzy rule selection to some problems in the UCI machine learning repository (Newman et al. 1998). Each data set is divided into two subsets of the same size: training data and test data. First fuzzy rules are extracted using training data satisfying the minimum confidence 0.6 and the minimum support of 0.01. The maximum rule length is specified as three in the rule extraction phase. All the extracted fuzzy rules have been used as candidate rules. Then NSGA-II has been applied to the extracted candidate rules to search for Pareto-optimal rule sets i.e., Pareto-optimal subsets of the candidate rules with respect to the three objectives to Maximize Accuracy and minimize Complexity1and Complexity2. String length $N$is the same as the number of the candidate rules because their subsets are represented by binary strings of length N.They use a hill-climbing procedure to remove unnecessary rules from each string.

An agent-based evolutionary approach is proposed to extract interpretable rule-based knowledge by Wang et al. (2005). Fuzzy set agent autonomously determines its own fuzzy sets information such as the number and distribution of the fuzzy sets. It can further consider the interpretability of fuzzy systems with the aid of hierarchical chromosome formulation and interpretability-based regulation method. Based on the obtained fuzzy sets, the Pittsburgh-style approach is applied to extract fuzzy rules that take both accuracy and interpretability of fuzzy systems into consideration. In addition, the fuzzy set agents can cooperate with each other to exchange their fuzzy sets information and generate offspring agents. The parent agents and their offspring compete with each other through the arbitrator agent. They compete with each other based on the criteria associated with the accuracy and interpretability to allow them to remain competitive enough to move into the next population.

Chromosome representation: A hierarchical chromosome formulation for GA is used where the genes of the chromosome are classified into two different types: control genes and parameter genes. To indicate the activation of the control genes, an integer 1 is assigned for each control gene that is ignited, whereas 0 is for turning off. This chromosome formulation is said to enable the number as well as the distribution of fuzzy sets to be optimized. Each fuzzy rule is coded as a string of length N.The string is an N-length array and the *ith* element of the array indicates which fuzzy set of the *ith* fuzzy variable is fired. The *ith* element is denoted as *ci* and initially set to an integer between 0 and Mi^a, which is the number of fuzzy variables. The rule consequents are not involved in the chromosome encoding. *Pittsburgh-style* rules are extracted.

Parameters: The *initial population* consists of randomly generated fuzzy rule sets. The Fuzzy Set Agent initializes its own control genes and parameter genes randomly. The *population size* is 40 with *number of generations* being 100. Mating restriction is not incorporated i.e., number of Fuzzy Set Agents in the current population are different with one another and selected randomly with the same probability. *Stopping criteria* is taken as number of generations.

Reproduction operators: Fuzzy Set Agents are different with one another and *selection* is random with the same probability for reproduction, while *one-point crossover* is used. Crossover operation randomly selects a different cutoff point for each parent to generate offspring rule sets. *Mutation* operation randomly replaces each element of the rule set's string with another linguistic value if a probability test is satisfied. Elimination of existing rules and addition of new rules are also used as mutation operations.

Objectives, fitness function and the Optimization strategy : The Fuzzy Set Agent uses the following three criteria to evaluate fuzzy rule set candidates: Accuracy measured in terms of Mean-Square-Error (MSE which are classification error rates for classification problems), the number of fuzzy rules, and the total length of fuzzy rules, i.e., the total number of the rule antecedents displayed in the rule base. In order to compare the fuzzy rule base candidates, the preference for the three criteria are predefined. The accuracy is given the first priority and the other two criteria of interpretability are given the second priority. If one rule base candidate is better than the other based on the accuracy preference, then it is not required to compare the other two criteria. The difference of the accuracy value of fuzzy rule base candidates is used to design the preference. If the difference is less than or equal to a predefined value, then it is considered that the candidates have the same accuracy level.

The Fuzzy Set Agent selects *Npop* best candidates from the mixed populations using an elitism strategy. The agents apply NSGA-II multi-objective decision making method to evaluate fuzzy rule sets candidates. The agents interact with each other by switching fuzzy sets information and also give birth to new agents. Based on the multiple criteria about the accuracy and interpretability of fuzzy systems, the elite agents are retained in the multi-agent system, whereas the obsolete agents are destroyed by the arbitrator agent.

Data sets and area of application: Iris Data in the biology domain from UCI repository (Newman et al. 1998), and simulated data using Matlab have been used for experiments.

The above Multi objective genetic fuzzy system is applied to the problem of anomaly intrusion detection by Tsang et al. (2005, 2007) and tested on KDD cup bench mark data from UCI machine learning repository (Newman et al. 1998). The system extracts accurate and interpretable fuzzy rule based knowledge from network data with high detection rate and low false positives. Confusion matrix is used to calculate various measures namely Precision, F-measure, and Overall accuracy to evaluate the classifier. Agents were found to continuously improve the average accuracy using the elitism strategy in each generation. The agent cooperation and competition enables effective exchange of fuzzy set information for constructing accurate and compact fuzzy systems. The experimental results are said to have demonstrated that the agent based genetic fuzzy system is both accurate and interpretable.
Dehuri and Mall (2006) present a multi-objective genetic algorithm for mining highly predictive and comprehensible classification rules from large databases. They have proposed a multi-objective evolutionary algorithm called Improved Niched Pareto genetic algorithm (INPGA) for this purpose. The INPGA rule generation is to associate each individual of the population with the same predicted class, which is never modified during the running of the algorithm.

Chromosome representation: Each individual in the population represents a candidate rule 'R' of the form "if A then C". The antecedent of this rule can be formed by a conjunction of at most $n - 1$ attributes, where n is the number of attributes being mined. Each condition is of the form $Ai = V_{ij}$, where Ai is the ith attribute and V_{ij} is the jth value of the ith attributes's domain. The consequent consists of a single condition of the form Gk = V_{kl}, where Gk is the kth goal attribute and V_{kl} is the lth value of the kth goal attributes's domain. The user specifies the goal attribute that is of interest to him. A string of fixed size encodes an individual with n genes representing the values that each attribute can assume in the rule. If an attribute is not present in the rule antecedent, the corresponding value in gene is "-1". This value is a flag to indicate that the attribute does not occur in the rule antecedent. This encoding is said to effectively represent a variable-length individual (rule).

Parameters: The data-mining algorithm needs to discover rules by accessing the training set. The algorithm has access to the values of both predicting attributes and the goal attribute of each example (record) in the training set. Thus an *initial population* is created directly from the dataset. A *population size* of 100 for Zoo dataset and 500 for Nursery dataset are used with n*umber of generations* being 500 and used as the stopping criteria. A *crossover rate* of 0.8 for Zoo dataset 0.75 for Nursery dataset and *mutation rates* of 0.03 for Zoo dataset 0.002 for Nursery dataset have been used with the *Tournament size* for selection being 15 for Zoo dataset and 50 for Nursery dataset. The *Niche radius* is 11 for Zoo dataset and 20 for Nursery dataset.

Reproduction operators: Pareto domination tournaments are held for *selection* and *uniform crossover* is used. There is a probability for applying crossover to a pair of individuals and another probability for swapping each gene (attribute)'s value in the genome (rule antecedent) of two individuals. After crossover is complete, the algorithm analyses if any invalid individual was created. If so, a repair operator is used to produce valid-genotype individuals. Besides crossover and mutation, the insert and remove operators directly try to control the size of the rules being evolved, thus influencing the comprehensibility of the rules. The *mutation* operator randomly transforms the value of an attribute into another value belonging to the same domain of the attribute.

Objectives, fitness function and the Optimization strategy: The discovered rules should have high predictive accuracy and high comprehensibility. The fitness function is computed as the arithmetic weighted mean of comprehensibility and predictive accuracy. Finally, the fitness function is given by: *f(x) = (w1 * Comprehensibility+w2 * Predictive accuracy) / (w1+w2)*, where w1 and w2 are user-defined weights. Further, the tournament selection is altered in two ways. First, Pareto domination tournament is introduced. Second, a non-dominant tournament sharing is implemented to determine the winner. Pareto domination tournaments are modified in order to give more domination pressure than binary tournament, and more control of that pressure. A sampling scheme is implemented, as follows. Two candidates for selection are picked at random from the population. A comparison set of individuals is also picked randomly from the population. Each of the candidates is then compared against each individual in the comparison set. If one candidate is dominated by the comparison set, and the other is not, the latter is selected for reproduction. If neither or both are dominated by the comparison set, then sharing is used to choose a winner. Equivalence sharing on the non-dominated frontier is incorporated by using continuously updated sharing and niche count sampling techniques in the Niched Pareto GA. In order to maintain diversity along the phenotypic Pareto optimal front, sharing is done in attribute space. The "best fit" candidate

is determined to be that candidate which has the least number of individuals in its niche and thus the smallest niche count. In order to consider the measure that can maintain useful diversity in the Pareto set the following approach called Improved Niched Pareto Genetic Algorithm (INPGA) is used. It first finds the center of gravity of both niche radii, and then the standard deviation (SD) of each point of both radii is calculated, the candidate having larger SD being chosen.

Data sets and area of application: The simulations have been performed using the zoo and nursery datasets obtained from the UCI repository (Newman et al. 1998).

Berlanga et al. (2006) and Del Jesus et al. (2007) present a multi objective genetic algorithm for obtaining fuzzy rules for subgroup discovery. The multi-objective algorithm proposed in this paper defines three objectives. One of them is used as a restriction on the rules in order to obtain a Pareto front composed of a set of quite different rules with a high degree of coverage over the examples. The other two objectives taken into account are the support and the confidence of the rules.

Chromosome representation: All the information relating to a rule is contained in a fixed-length chromosome with a binary representation in which, for each feature it is stored a bit for each of the possible values of the feature indicating the inclusion or non-inclusion of the corresponding linguistic label or discrete value of the variable. A rule containing all the bits corresponding to a feature with the value 1 indicates that this feature has no relevance for the information contributed in the rule. In the proposal presented in this paper, fuzzy rules in disjunctive normal form (DNF fuzzy rules) are induced. Each individual codifies a single rule, and a set of rules is codified by a subset of the complete population. The fuzzy sets corresponding to the linguistic labels for a linguistic variable are specified by means of the corresponding membership functions which can be defined by the user or defined by means of a uniform partition if expert knowledge is not available. In this algorithm, uniform partitions with triangular membership functions are used.

Parameters: The *Population* consists of 100 chromosomes with an *elite population size* of 5. The maximum *numbers of evaluations* of individuals in each GA run being 10,000 with a *crossover probability* of 0.7 and *mutation probability* of 0.01. The experiments are carried out with 5 runs for each class of the categorical target variable, *low*, *medium* and *high* efficiency with 3 linguistic labels for the continuous variables.

Reproduction operators: Binary tournament selection with replacement, *two point crossover* and *biased uniform mutation* operator are used. Half of the mutations carried out have the effect of eliminating the corresponding variable in order to increase the generality of the rules.

Objectives, fitness function and the Optimization strategy: Confidence, support, and original-support are taken for optimization as objective quality measures. Subjective measures for the descriptive induction process namely *Significance for a rule* indicating how significant is a finding, measured by the likelihood ratio of a rule and *Unusualness for a rule* defined as the *weighted relative accuracy* of a rule described as the balance between the coverage of the rule and its accuracy gain are proposed. The multi-objective GA is based on the SPEA2 approach, and so applies the concepts of elitism in the rule selection using a secondary or elite population and search of optimal solutions in the Pareto front. In order to preserve the diversity at a phenotypic level the algorithm uses a niches technique that considers the

proximity in values of the objectives and an additional objective based on the novelty to promote rules which give information on examples not described by other rules of the population.

Data sets and area of application: The algorithm is applied to the problem over the extraction of useful information on trade fairs. For this real problem, the data mining algorithm extracts information of interest about each efficiency group. The rules generated can be used to determine the influence which the different "fair planning variables" have over the results obtained by the "exhibitor fair planning" policies to be improved.

Type and size of data set and number of class attributes: A questionnaire was designed to reflect the variables that better allow explaining the trade fair success containing 104 variables. 7 of them are continuous and the rest are categorical features based on expert discretization. The stand's global efficiency is rated as *high*, *medium* or *low*, in terms of the level of achievement of objectives set for the trade fair. The data contained in this dataset were collected in the Machinery and Tools biennial held in Bilbao in March 2002 and contains information on 228 exhibitors.

Khabzaoui et al. (2008) state that mining frequent rules is not always interesting as it may reveal already known associations. On the contrary, mining non frequent rules may reveal associations that may occur in a subset of experiments, or for a subset of individuals, and may explain some specific cases such as a specific disease which may be to the interest of Biologists. With this as motivation, the authors have studied different quality criteria for association rules. A multi-objective model has also been proposed for rules mining taking it as a multi-objective combinatorial optimization problem. The aim of the MOO algorithm is to find both non frequent and interesting rules. As the search space may be very large, a discussion about different approaches is proposed and a hybrid approach that combines a metaheuristic and an exact operator is presented. Their applications deal with rule mining in micro-array data. In such genomic data, expression levels of thousands of genes are measured according to several experimental conditions and for several individuals. A hybrid approach, combining a dedicated genetic algorithm and an enumerative procedure is also proposed. The frequency of application of the enumerative procedure is studied through experiments. It appears that the procedure is time consuming and a choice has to be made between quality of solutions and time allowed.

Chromosome representation: Attributes are coded as genes. For example on a public microarray database "MIPS Yeast Genome Database" containing 2,467 genes (attributes) and 79 chips (rules). In this problem, 2,467 attributes are candidate to form rules and 79 relations between those attributes are given to evaluate the rules.

Parameters: An Enumerative Procedure (EP) which is an adopted extension of Apriori is used to construct the *initial population* of rules by adding one attribute at a time to the rule. The *population* consists of 150 individuals in an iteration. The probability of s*election in population* is 1/3 with *Global Mutation rate* being 0.5, *Crossover rate* being 0.8 and the s*election in Pareto archive (elitism)* being 0.5. The minimum *number of generations* is taken to be 200 while the maximal *number of attributes for the enumeration procedure* (Max*Nb*) is10.

Reproduction operators: The classical *roulette selection* is used which is based on the ranking notion where, the probability of selection of a solution is proportional to its rank. Pareto ranking where the rank of a solution corresponds to the number of solutions, in

the current population, by which it is dominated is used. The *crossover* mixes the features of two rules by the combination of their attributes. The proposed crossover operator has two versions, to take into account the fact that the parents may share a common attribute: *Crossover by value exchange:* If two rules X and Y have one or several common attributes in their *C*parts, one common attribute is randomly selected. The value of the selected attribute in X is exchanged with its counterpart in Y.*Crossover by insertion:* Conversely, if X and Y have no common attribute, one term is randomly selected in the *C*part of X and inserted in Y with a probability inversely proportional to the length of Y. Similar operation is performed to insert one term of Y in X. The Enumerative Procedure (*EP*) is used as a crossover operator when the number of distinct attributes composing the two rules is not too large.

Four *mutation* operators are implemented. The *Value mutation* replaces an attribute value by a randomly chosen one. The *Attribute mutation* replaces a term by another. The *Insertion operator* adds a term i.e. a randomly chosen attribute with a randomly chosen value in the rule, and the *Delete operator* removes a term of the rule if the number of terms is greater or equal to 3. The choice of the mutation operator is not made on advance, but the probability of appliance of a mutation operator is made in an adaptive manner. At the beginning of the algorithm, all the mutation operators have the same probability to be selected. An adaptive strategy for calculating the rate of application of each mutation operator which favors operators that often improve solutions is proposed. It calculates the "improvement ratio" of each operator and determines the probability of appliance of each operator, using this indication. The progress of an operator is evaluated by comparing the Pareto ranking of the solution obtained after the application of the operator with the rank of the initial solution. Then the new selection probabilities of the mutation operators are computed proportionally to the progress calculated.

Objectives, fitness function and the Optimization strategy: Chosen criteria are support, confidence, Jmeasure, interest and surprise. The elitist non dominated sorting replacement where the worst ranked solutions are replaced by dominating solutions generated by crossover and mutation operators is used. The size of the population remains unchanged. Non dominated association rules are archived into a secondary population called the "Pareto Archive" in order to keep track of them. It consists in archiving all the Pareto association rules encountered over generations. This archive has to be updated each time a solution is added. The Pareto solutions are not only stored permanently, they also take part in the selection and participate in the reproduction. Therefore a probability of selecting a parent from the archive is set.

Data sets and area of application: The algorithm is applied to a public micro-array database "MIPS Yeast Genome Database" containing 2,467 genes. The area of application being biology.

Giusti et al. (2008) report research that combines evolutionary algorithms and ranking composition methods for multi-objective optimization. In this approach, candidate solutions are built, evaluated and ranked according to their performance in each individual objective. Then rankings are composed into a single ranking which reflects the candidate solutions' ability to solve the multi-objective problem considering all objectives simultaneously. The behaviors of 5 ranking composition methods are discussed. These methods are compared and it is concluded that all of the studied ranking composition methods provide good balance of objectives. The rule's contingency matrix is used to estimate rule quality levels according to different quality criteria.

Ranking composition is a non-Pareto technique. A ranking is a collection of items arranged in order according to some quality which they all possess. The position of each item in the

ranking called a rank is usually a numerical value and indicates the items position compared to others. In the work discussed ranking composition is performed as a two-step process, the ranking step and the composition step. The ranking step consists in ranking each item of the collection according to one single objective and is independent of the composition method. The result of this step is a number of rankings that equals the number of optimization objectives, and each item will have the same number of ranks. The next step consists in composing those ranks into a single value using a composition method. The result of this composition is a final ranking that reveals which item or items are best in providing good balance of all the objectives (Giusti et al. 2008). The five composition methods considered here are mean, median, inverse, harmonic and Condorcet composition.

Chromosome representation: The search space consists of rules induced by *C4.5, C4.5 rules* and *CN2*.

Parameters: The *initial population* consists of a set of rules which are manually constructed by the user and which present the specific properties desired by the user, but sub-optimally. For each training set, a rule set with all rules induced by *C*4.5, *C4.5rules* and *CN2* are given as input to the multi-objective evolutionary algorithm. 100 train-and-test experiments have been performed for each combination of dataset and ranking composition method. A *crossover rate* of 60% and m*utation rate* of 5% have been used while the *Stopping criteria* using standard deviation is followed. After constructing each population, the algorithm would evaluate the mean of all quality measures. If that statistic's standard deviation becomes smaller than a threshold value, the GA process will be terminated.

Reproduction operators: The evolutionary algorithm is a genetic algorithm that combines knowledge by means of two mutation operators and three crossover operators but no explanation is given.

Objectives, fitness function and the Optimization strategy: In the case of this work, there is never a single rule that maximizes all desired measures. Instead, it is stated that the interest is in finding a rule that provides a good balance of the measures even if none of the measures is maximized. The task of optimization considered in this study consists of simultaneously maximizing the rules in terms of three measures of rule quality namely novelty, Laplace and support. These three measures are said to represent some of the most desirable characteristics of knowledge rules discovered by inducers. Rules with high support are applicable to a large number of examples, rules with high Laplace are precise in classifying new examples, and rules with high novelty represent knowledge that is potentially novel to the user or domain expert.

In the case of the work reported in this paper ranking composition methods are used to evaluate the rules according to the specific properties and select the rules which would be used to construct the next generation. The convergence criterion is standard deviation convergence. Once the convergence condition is satisfied, all optimized rules present in the last generation are evaluated in the training set, and the rule with the best evaluation, i.e., positioned first in the ranking, is selected and evaluated in the test set.

Data sets and area of application: Datasets from the UCI dataset repository (Newman et al. 1998) including Breast cancer, Bupa, E.Coli, German, Glass, Haberman, New-thyroid, Post-operative from biological domain and two other datasets namely Sonar, and Vehicle have been used in experiments.

However a drawback of the ranking composition method is that the composition does not reveal how much better the items are when compared to the others.

Casillas et al. (2008, 2009) propose Pitts-DNF-C, a multi-objective Pittsburgh-style Learning Classifier System that evolves a set of DNF-type fuzzy rules for classification tasks. The fuzzy rules have antecedent in *conjunctive normal form*. The system is explicitly designed to create consistent i.e., each input subspace has only one possible class, complete i.e., every training example fires at least one fuzzy classification rule, compact i.e., without redundant rules and without over-general rules i.e., avoiding covering input areas without data. For this purpose, new genetic operators are designed to guarantee that all the individuals in the population satisfy these four conditions. Incompleteness and inconsistency are used as penalty in the rule's fitness.

Chromosome representation: Pitts-DNF-C consists of a population of individuals. Each chromosome consists of the concatenation of a number of rules. The rules have condition in *conjunctive normal form*. Each rule which is a part of the chromosome is encoded by a binary string for the antecedent part and an integer coding scheme for the consequent part. The consequent part has a size equal to the number of output variables where each gene contains the index of the linguistic term used for the corresponding output variable. The number of rules is not fixed a priori so, the chromosome size is variable-length. A one-valued allele indicates that the corresponding linguistic term is used in the variable. The antecedent part has a size equal to the sum of the number of linguistic terms used in each input variable. The allele '1' means that the corresponding linguistic term is used in the corresponding variable.

Parameters: In the *initial population* all chromosomes start with the same number of rules. In order for the initialization procedure to guarantee that the initial individuals cover all the input examples, first, a rule is generated for each training example. The linguistic term that maximizes the matching with the input value is assigned to each variable and the class of the rule is set to the class of the input example. Then, redundant and inconsistent rules are removed. Instead of them, a new rule with the same antecedent and the majority class among this group of rules is introduced into the first individual. The remaining individuals are initialized similarly. For each new individual, the first individual is copied, and the class of each rule is randomly chosen among all the classes of the training examples with which the rule has a matching degree greater than zero. The length of all the individuals in the initial population is the same. *The population size* is 60 with 300 *generations* and *crossover rate* being 0.7 while a high *mutation rate* of 0.2 is considered. Number of generations is taken as the *stopping criteria*.

Reproduction operators: Binary tournament selection is used. *Crossover* interchanges rules between the two parents, but it does not modify them. Two types of *mutation* operators are used as follows: *Antecedent mutation operator* acts on input variables to create new rules and explores only feasible solutions. Two types of the antecedent mutation operators are used namely a contraction operator which converts the mutated rule into a more specific one by choosing a gene of the selected variable with a '1' and flipping to '0'and an expansion operator which carries out the opposite process to contraction operator, making the rule more general. It chooses a gene with allele '0' and flips it to '1'. The *Consequent mutation operator* is applied with a given probability rate to each individual and creates new rules by changing the consequent. In addition to these a *Completeness operator* or a reparation operator is used for adding rules to patch the uncovered input subspaces.

Objectives, fitness function and the Optimization strategy: Two objective functions are used to assess the quality of the generated fuzzy systems. The approximation error or mean squared error is used to improve the accuracy and the complexity based on number of DNF-type fuzzy rules. A generational approach with the multi-objective elitist replacement strategy of NSGA-II is used. Crowding distance in the objective function space is considered. Binary tournament selection based on the non domination rank (or the crowding distance when both solutions belong to the same front) is applied. The crowding distance is normalized for each objective according to the extreme values of the solutions contained in the analyzed front. Covering hyper matrix is used to store the label combinations of the antecedent that cover all the examples in the training data set when generating new rules. This is said to efficiently avoid over-generality or generating rules in regions without training data. The structure of this hyper matrix is an array, whose dimension is equal to the number of input variables, containing '1' in a cell if the corresponding input combination covers at least a training example and containing '0' in other case.

Data sets and area of application: In (Casillas et al. 2008), a collection of six data sets from the UCI repository (Newman et al. 1998) including Bupa, Glass, Iris, Tao, Thyroid, Wisconsin Breast-cancer are selected, and a data set known as Tao, which has been selected from a local repository are used. In Casillas et al. (2009), Diabetes problem whose data sets are obtained from L. Torgo's website are used. The Ele1 problem whose samples are obtained from real measurements from 495 towns in Spain and the Ele2 problem concerning the estimation of electrical network maintenance costs of medium voltage line are also considered. Moreover the algorithm is also applied to the Laser problem which is a set of laser data from the Santa Fe Institute (SFI) time series prediction and DEE problem involving predicting the daily average price of TkWhe electricity energy in Spain. The data sets for laser problem and DEE problem are obtained from KEEL website.

5.1.1 Discussion: EMOO system characteristics for rule mining using GA

Table 1 summarizes the EMOO system characteristics for rule mining using GA.

Chromosome Representation: The genetic representation of the solutions is the most determining aspect of the characteristics of any GA proposal (Del Jesus et al. 2007). Various data types are used in representing the chromosome on which the GA operates. The binary string representation uses bits to indicate the presence and absence of attribute values. The binary strings can be of fixed length or variable length. Fixed binary representation is used by De la Iglesia et al. (2003, 2005), Berlanga et al. (2006) and Del Jesus et al. (2007). However this representation has the limitation that the binary string chromosome has to be re-converted to "IF-Then" rules to be presented to the user. Another type of representation is the string representation where the attribute values are coded as genes and there are as many genes as there are attributes. This representation is more expressive than binary representation. This type of string representation is used in Khabzaoui et al. (2008), and Dehuri and Mall (2006). However representing chromosomes as string makes the processing slower and hence a string representation with indices to represent the genes and attribute values stored in arrays is used by Reynolds and De la Iglesia (2006, 2009), and Reynolds et al. (2009). There are other representations which use Hierarchical chromosome formulation with control gene and parameter gene (Wang et al. 2005; Tsang et al. 2005, 2007), and mixed representations which uses Concatenation of rules encoded as binary string for the antecedent part,

and an integer coding scheme for the consequent (Casillas et al. 2008, 2009). There are other EMOO algorithms which consider rule discovery as a post processing phase where they take rules created by other algorithms and optimize these rules using various objectives as in Narukawa et al. (2005), Ishibuchi and Nojima (2005), Ishibuchi et al. (2007), Ishibuchi (2007) and Giusti et al. (2008). These are rule selection systems rather than rule induction systems.

Initialization of population: Evolutionary systems work on a population of individuals. The population is initialized with individuals using an initialization process. Either the user can specify the initial set of individuals as in Giusti et al. (2008), or an automated procedure can be used for initialization. Casillas et al. (2008) state that the initialization procedure has to guarantee that the initial individuals cover all the input examples from the training data set. The procedure can be a random one as in Reynolds and De la Iglesia (2006), Wang et al. (2005) and Tsang et al. (2005, 2007). Mutated default rules are used by De la Iglesia et al. (2003, 2005). Rules with 3 ATs is proposed by Reynolds and De la Iglesia (2006) while Dehuri and Mall (2006) use rules constructed from training data. An enumerative procedure is used by Khabzaoui et al. (2008) to construct the initial rules. Default rule with no ATs have been used by Reynolds and De la Iglesia (2009) and Reynolds et al. (2009). In Casillas et al. (2008, 2009), all chromosomes start with the same number of rules and cover all the input examples. The post processing or rule selection algorithms create the initial population by using rules with minimum threshold on certain rule metrics like minimum support and confidence as in Ishibuchi et al. (2007) and Ishibuchi (2007), while SLAVE measure is used to choose the initial rules in Narukawa et al. (2005) and Ishibuchi and Nojima (2005).

Parameters: The basic set of parameters used in most evolutionary systems include the size of the population, number of generations, crossover rate and mutation rate which are discussed below.

Population size: Population size is the number of individuals in a generation. The population size ranges from 10 to 500 in the algorithms discussed. Wang et al. (2005), Tsang et al. (2005, 2007) and Casillas et al. (2008, 2009) use a population of less than 100, whereas most others use more than 100 individuals in their experiments like Reynolds and De la Iglesia (2006), Khabzaoui et al. (2008), Narukawa et al. (2005), Ishibuchi and Nojima (2005), Ishibuchi et al. (2007), Ishibuchi (2007), Berlanga et al. (2006), Del Jesus et al. (2007) and Dehuri and Mall (2006). De la Iglesia et al. (2003, 2005), Reynolds and De la Iglesia (2009), Reynolds et al. (2009) have experimented with varying population sizes ranging from 10 to 500 individuals.

Number of generations: This is the parameter which decides the number of iterations of the algorithm. In most cases the number of generations is used as the stopping criteria. The number of generations range from 50 to 5,000.

Cross over rate: Crossover rate is the probability of reproduction using crossover. Most of the algorithms use a crossover rate which is greater than 0.5 as in De la Iglesia et al. (2003, 2005), Khabzaoui et al. (2008), Narukawa et al. (2005), Ishibuchi and Nojima (2005), Ishibuchi et al. (2007), Ishibuchi (2007), Casillas et al. (2008, 2009), Berlanga et al. (2006), Del Jesus et al. (2007); Dehuri and Mall (2006) and Giusti et al. (2008). A low cross over rate of 0.2 is used by Reynolds and De la Iglesia (2006). Reynolds and De la Iglesia (2009) have experimented with a range of crossover rates starting from 0. The systems by Wang et al. (2005) and Tsang et al. (2005, 2007), which use agents does not impose mating restrictions on their chromosomes.

Table 1 EMOO system characteristics for rule mining using GA

EMOO system by	Chromosome representation	Initialization of the rules population	Pop. size: No. of generations Cross over rate (CR) Mutation rate (MR)	Objectives optimized	Reproduction operators			MOO strategy for optimization
					Selection	Crossover	Mutation	
De la Iglesia et al. (2003, 2005)	Binary string	Mutated default rule	100–160 in steps of 10 50–100 in steps of 10. CR: 60–90% steps of 10 MR: 0–4% in steps of 1	Accuracy and coverage	Binary tournament	One point	Random flipping of bits	NSGA II
Reynolds and De la Iglesia (2006)	Array of ATs. The value index used for gene encoding	Initial rules constructed at random using at most three ATs	200 rules, 150 CR: 0.2, MR: 0.2	Support, coverage confidence	Binary tournament	Uniform	Mutation depends on type of AT	NSGA II
Reynolds and De la Iglesia (2009), Reynolds et al. (2009)		Default rule with no AT in the antecedent	10, 20, 50, 100, 200, 500 rules 60 for each combination CR: 0, 20, 40, 60, 80, & 100% MR: 0, 5, 10, & 20%	Support, confidence, and coverage	Not specified	Uniform	Mutation depends on type of AT	Multi objective GRASP
Khabzaoui et al. (2008)	Attributes are coded as genes	Enumerative procedure to construct rules	150, 200 CR: 0.8 MR: 0.5	Support, Jmeasure interest, surprise confidence	Roulette Wheel	Value exchange, insertion	Value & attribute mutation, insertion deletion operators	Elitism using Pareto dominance
Narukawa et al. (2005), Ishibuchi and Nojima (2005)	If-then rules	Short fuzzy rules for all classes chosen using the SLAVE measure	200 5000 CR: 0.8 MR: p(0→1) = 1/300 M and pm(1→0) = 0.1	Accuracy and complexity	Similarity based tournament selection	Uniform	Biased mutation	NSGA II

Table 1 continued

EMOO system by	Chromosome representation	Initialization of the rules population	Pop. size: No. of generations Cross over rate (CR) Mutation rate (MR)	Objectives optimized	Reproduction operators			MOO strategy for optimization
					Selection	Crossover	Mutation	
Ishibuchi et al. (2007), Ishibuchi (2007)	If-then rules	Short fuzzy rules with minimum support,confidence thresholds	200 1000 CR: 0.9 MR: p(0→1) = 1/300 M and pm(1→0) = 0.1	Accuracy, number of rules and number of antecedent conditions	Not specified	Uniform	Biased mutation	NSGA II
Wang et al. (2005), (Tsang et al. 2005, 2007)	Hierarchical formulation with control, parameter gene	Random initialization	40 100 Mating restriction is not incorporated	Accuracy, number of rules, number of antecedent ATs	Random selection	One point	Random	NSGA II
Casillas et al. (2008, 2009)	Binary string for the antecedent, integer coding for the consequent	All chromosomes with same no. of rules covering all examples	60 300 CR: 0.7 MR: 0.2	Accuracy and Complexity	Binary tournament selection	One point	Antecedent and Consequent mutation operators	NSGA II
Berlanga et al. (2006) and Del Jesus et al. (2007)	Fixed length binary string	Not specified	100 5× No. Of classes CR: 0.7 MR: 0.01	Confidence, support, coverage, significance, unusualness	Binary tournament	Two point	Biased uniform mutation operator	SPEA 2

Table 1 continued

EMOO system by	Chromosome representation	Initialization of the rules population	Pop. size: No. of generations Cross over rate (CR) Mutation rate (MR)	Objectives optimized	Reproduction operators			MOO strategy for optimization
					Selection	Crossover	Mutation	
Dehuri and Mall (2006)	Fixed size string	Rules generated from training data	100, 500 500 CR: 0.8, 0.75 MR: 0.03, 0.002	Predictive accuracy, comprehen-sibility	Pareto domination tournament	Uniform crossover	Simple mutation, insert and delete operators	INPGA
Giusti et al. (2008)	Rules induced by *C*4.5, *C4.5rules* and *CN2*	Rules manually constructed by the user	Not specified 100 CR: 0.6, MR: 0.05	Novelty, laplace, support	Not specified			Ranking composition

Mutation rate: Mutation rate is the probability of an individual being produced by mutation. Mutation rate is usually less than crossover rates. In most systems it ranges from 0.0 to 0.3. However, Khabzaoui et al. (2008) use a high mutation rate of 0.5 which they call the global mutation rate and they adopt an adaptive mutation rate calculation procedure for their four types of mutation operators using the progression of the results obtained after applying them.

Reproduction Operators: Evolutionary systems operate on a set of individuals by using special types of operators known as reproduction operators. These include the selection, crossover and mutation operators.

Selection: Selection is the process of choosing the parent individuals for creating new individuals. The types of selection include random, roulette wheel and tournament selection. Wang et al. (2005) and Tsang et al. (2005, 2007) use random selection. Binary tournament is used by De la Iglesia et al. (2003, 2005), Reynolds and De la Iglesia (2006), Berlanga et al. (2006), Del Jesus et al. (2007) and Casillas et al. (2008, 2009), whereas similarity based tournament is used by Narukawa et al. (2005), and Ishibuchi and Nojima (2005). Pareto domination tournaments are used by Dehuri and Mall (2006). Roulette wheel selection is used by Khabzaoui et al. (2008).

Crossover: Crossover exchanges the genetic material of the parents to create new individuals. One point crossover is used in De la Iglesia et al. (2003, 2005), Wang et al. (2005) and Tsang et al. (2005, 2007), whereas two point crossover is used by Berlanga et al. (2006) and Del Jesus et al. (2007). Uniform crossover is carried out in Narukawa et al. (2005), Ishibuchi and Nojima (2005), Ishibuchi et al. (2007), Ishibuchi (2007), Dehuri and Mall (2006), Reynolds and De la Iglesia (2006, 2009) and Reynolds et al. (2009). While others like Casillas et al. (2008, 2009) and Khabzaoui et al. (2008) do not mention the type of crossover used.

Mutation: Mutation chooses a point in the individual and changes the value occurring at that point. Wang et al. (2005) and Tsang et al. (2005, 2007) use random mutation where the mutation point is chosen at random. Random flipping of bits is used by De la Iglesia et al. (2003, 2005) in their binary string representation whereas biased mutation is used by Narukawa et al. (2005), Ishibuchi and Nojima (2005), Berlanga et al. (2006), Del Jesus et al. (2007), Ishibuchi et al. (2007) and Ishibuchi (2007). Some have introduced new insertion and deletion operators along with usual mutation as in Khabzaoui et al. (2008), and Dehuri and Mall (2006). Casillas et al. (2008, 2009) use separate mutation for their antecedent and consequent parts while different mutation operators are used for different types of attribute tests in Reynolds and De la Iglesia (2006, 2009) and Reynolds et al. (2009)

Objectives optimized: The rule measures used as objectives for optimization include both objective and subjective measures. The various objective measures used include support, confidence, coverage, accuracy, predictive accuracy, misclassification cost, precision and specificity. Although called by various names they all aim at maximizing the predictive accuracy of the rules. The subjective measures most of which are derived from the basic objective measures include novelty, surprisingness, interestingness, Laplace measure and the Jmeasure.

MOO strategy for optimization: Most of the EMOO systems use NSGA II as their optimization strategy including De la Iglesia et al. (2003, 2005), Narukawa et al. (2005), Ishibuchi and Nojima (2005), Wang et al. (2005), and Reynolds and De la Iglesia (2006). While Tsang et al. (2005, 2007), Berlanga et al. (2006), Del Jesus et al. (2007), Ishibuchi et al. (2007),

Ishibuchi (2007), and Casillas et al. (2008, 2009) use SPEA 2, Khabzaoui et al. (2008) use elitism with pareto dominance without discussing crowding distance or separate elite population. Reynolds and De la Iglesia (2009) and Reynolds et al. (2009) introduce a Multi objective GRASP which is a greedy randomized search procedure for searching optimal rules. Five types of ranking composition methods are discussed in Giusti et al. (2008) which are non-pareto based methods.

Data sets and area of application: Most of the algorithms have been tested on benchmark data sets from the UCI machine learning repository which is a collection of datasets spanning a wide area of applications including medicine, biology, finance, transportation, computer security, data sets for pattern matching, etc. The data sets and descriptions of these data sets can be obtained from the UCI machine learning repository website (Newman et al. 1998). Berlanga et al. (2006) and Del Jesus et al. (2007) have used data sets from Machinery and Tools biennial held in Bilbao in March 2002 and contains information on 228 exhibitors. Trade fair decisions regarding the position of the stall can be taken based on the information. Casillas et al. (2009) have used Diabetes data set from L. Torgo's website. The data sets for Ele1 consists of real electricity measurements from 495 towns in Spain, while the Ele2 problem deals with the estimation of electrical N/W maintenance costs and DEE problem for daily average price of electricity energy all of which describe the electricity problems in Spain. A data set for Laser problem from Santa Fe Institute has also been used. The Laser and DEE data sets have been obtained from KEEL website, while data sets for the electricity problems of Ele1, and Ele2 have been obtained from a local website.

5.2 MOO systems which use GP

There are EMOO systems for rule mining which use genetic programming as their evolutionary strategy. Zhao (2007) proposes a multi-objective genetic programming (MOGP) approach to developing Pareto optimal decision trees for the classification problem. It allows the decision maker to specify partial preferences on the conflicting objectives, such as false negative vs. false positive, sensitivity vs. specificity, and recall vs. precision. This paper makes a unique contribution by formulating cost sensitive classification as a multi-objective optimization problem and providing an evolutionary computation approach. Although the MOO algorithm is for developing decision trees, since decision trees can easily be converted to classification rules, it is considered for discussion.

Chromosome representation: A decision tree can be naturally represented with a tree structure. There are two kinds of tree nodes, terminals and functions. Terminals are attributes (integer), values (real), or classes (binary.) A binary classification problem is described by a binary class y and a vector of attributes $x = (x1, x2...xm)$. An attribute terminal is an integer number in the range $[1, m]$, representing an attribute number. A value terminal is a real number in the range [0, 1), representing a threshold value (after a linear transformation) for a numeric attribute or the index (after a conversion into integer) of a possible value for a nominal attribute. A class terminal is binary, representing a leaf node in a decision tree. The node function represents an intermediate node of a decision tree. It takes four arguments and returns a binary result; its signature is N: integer $\times$ real $\times$ binary $\times$ binary $\rightarrow$ binary. It is represented as a four-tuple, $N = (a, v, L, R)$, where a is an attribute terminal, v is a value terminal, and L and R are class terminals or node functions. Since both the type of a class terminal and the output type of a node function are binary, L and R can be either class terminals or node functions.

Parameters: Population size, No. of generations, Crossover rate and mutation rate are chosen by the user using a Java interface. The decision maker can decide to stop the procedure when satisfactory solutions have been found or when the solutions on the front appear to have stabilized.

Reproduction operators: The *tournament selection* method has been adopted in MOGP. When a tournament is held to select a parent, a small number of participants are randomly drawn from the current population and the winner, the fittest individual in the tournament, is selected. The selection mechanism of MOGP also takes the size of a candidate tree into account where the smaller tree is preferred. This helps in tackling the over fitting problem. Elitist selection is also incorporated in MOGP. *Crossover* operates on two individuals. It combines the characteristics of two parents by swapping a selected sub-tree of one parent with a selected sub-tree of the other. *Mutation* operates on one individual. It randomly selects a point in the tree and replaces the sub-tree starting at that point with a new randomly generated sub-tree.

Objectives, fitness function and the Optimization strategy: Two conflicting objectives, of minimizing false negative rate and minimizing false positive rate are taken into consideration. The performance of a classifier is assessed based on a confusion matrix summarizing the numbers of different prediction outcomes. In the MOGP system, the fitness of an individual is assigned on the basis of its relative non-dominance. Since the tournament selection method is used in the evolution procedure, the rank rather than absolute value is used in the fitness evaluation. Thus, all non dominated individuals in the current population are assigned rank one, while individuals dominated by one or more others are assigned rank two or higher.

Data sets and area of application: The system has been applied on several binary classification datasets publicly available from the UCI machine learning repository (Newman et al. 1998) including breast cancer, Wisconsin breast cancer, hepatitis, horse colic, heart disease (Statlog project), and Pima Indians diabetes which comes under the area of biology, while credit card application approval and German credit (Statlog project) come under financial decision making. Other data sets include the congressional voting records, labor relations, sonar, ionosphere and chess king– rook-vs.-king–pawn.

Reynolds and De la Iglesia (2007), describe how, by using a more complex representation of the rules, it is possible to produce effective classifiers for two class problems. Furthermore, through the use of multi-objective genetic programming, the user can be provided with a selection of classifiers providing different trade-offs between the misclassification costs and the overall model complexity.

Chromosome representation: In this paper an alternative approach of using a more expressive rule representation, specifically by using expression trees is proposed. The algorithm described in this paper manipulates rules of the form *antecedent* $\rightarrow$ *consequent,* where both antecedent and consequent are constructed from *attribute tests*. Three different types of attribute test (AT) namely value, inequality and binary partition are used. Value and inequality tests are used exclusively on categorical fields, while binary partition tests are used with a numeric field. Values occurring in each field are stored in reference arrays. The index values, rather than the values from the database, are used in the representation of the ATs. ATs are combined in expression trees that represent the rule antecedent to form *Rule Trees*. Leaf nodes contain ATs, while internal nodes contain a Boolean operator. These operators have been restricted to be either 'OR' or 'AND'

Parameters: The *initial population* is initialized with randomly generated balanced trees of depth two, where the root node is considered to be at depth zero. Experiments have been performed with six *population sizes* 10, 20, 50, 100, 200 and 500 individuals. Each experiment consists of 30 runs or *generations* of the algorithm, with 200,000 rule evaluations per run. Experiments have been conducted with a focus on finding the best values for crossover rate and population size. Six *Crossover rates*, 0, 20, 40, 60, 80 and 100% have been tested. Variable *Mutation rates* with 50% probability that a random AT is mutated; a 25% probability that an AT and its parent node are removed and a 25% probability that a random AT with a new internal node is added are used. Number of generations is used as *stopping criteria*.

Reproduction operators: The client or user *selects* a rule for reproduction. *Sub tree crossover* proceeds by selecting a node at random in each tree and swapping the sub trees headed by these nodes. A choice between crossover and mutation is made when creating new solutions, rather than both being applied probabilistically. Solutions generated during genetic programming tend to suffer from *bloat*, i.e. they grow excessively. In this paper, bloat is counteracted in three ways. The simplicity of a rule is considered as an objective of the problem counteracting bloat. Secondly, rule simplification is performed, removing redundant sections from rules. Finally, a simple limit on rule size is imposed. If, after simplification, a rule exceeds this AT limit, ATs and their parent nodes are removed until the constraint is satisfied. In this paper, this limit has been set to 20 ATs, in order to demonstrate the effect of rule size on misclassification costs.

Objectives, fitness function and the Optimization strategy: Misclassification costs on the training data and rule complexity are taken as objectives to be minimized.

Data sets and area of application: Five datasets from the UCI machine learning repository (Newman et al. 1998) including the Adult data set, Forest Cover Type, Contraception method choice, Breast Cancer (Wisconsin) and the Pima Indians Diabetes datasets have been used for experimentation all of which come under the area of biology.

Baykasoglu and Ozbakir (2007), in their work use a new chromosome representation and propose a solution technique based on Multi-Expression Programming (MEP) which they call MEPAR-miner (Multi-Expression Programming for Association Rule Mining).

Chromosome representation: MEP is similar to classical genetic programming but it uses fixed length linear strings of chromosomes to represent programs in the form of expression trees. The MEP genes are represented by substrings of variable length. The number of genes in a chromosome is constant and it represents the chromosome length. Each gene encodes a terminal (an element in the terminal set T) or a function symbol (an element in the function set F). T $\rightarrow$ {a, b, c, d} and F$\rightarrow$ { +, _,*, /}. A gene encoding a function includes pointers towards the function arguments. Function parameters always have indices of lower values than the position of that function itself in the chromosome. According to the MEP representation scheme, the first symbol in a chromosome must be a terminal symbol. This ensures that only syntactically correct programs are obtained. Each MEP chromosome encodes a number of expressions equal to the chromosome length.

Parameters: Initial population is generated according to predefined population size parameter which determines the number of chromosomes in the population. *Population size* parameter determines the number of individuals evaluated in each generation and is equal to 250. The algorithm is repeatedly executed "number of classes X number of generations" times,

where the number of generations is 250. A*Crossover rate* of 0.9, *Mutation rate* of 0.2 have been used while number of generations is used as the *stopping criteria.*

Reproduction operators: Binary tournament selection procedure is used to fill the mating pool. Two individuals are selected randomly from the current population. The best individual is copied to the mating pool. Two parent chromosomes are randomly selected from the mating pool for *Crossover*. A crossover point is randomly determined to perform the recombination process. Each symbol (terminal pointer, function, function pointer) in the chromosome may be the target of the *mutation* operator. By mutation some symbols in the chromosome are changed according to the predefined mutation probability. Random mutation points within the chromosome are determined. If it is a terminal gene then the terminal pointers are replaced by another relational operator and the attribute value is modified accordingly to be within the domain range. If the mutation point is a function gene then logical function is replaced by another logical function. The pointers of mutated logical function which point to the preceding genes are reassigned.

Objectives, fitness function and the Optimization strategy: The values of true positive, true negative, false positive, and false negative are taken into consideration for defining the sensitivity and specificity which are taken as objectives for optimization. The fitness function is defined as the product of specificity and sensitivity. The value of the fitness function is in the range of 0–1. The fitness value is 1 when all of the instances are correctly classified by the rule. Before the application of genetic operators, the chromosome with the best logical expression in population is copied to the next generation without change.

Data sets and area of application: Nine data sets from the UCI Machine Learning Repository (Newman et al. 1998) are used for application of the algorithm. These include Wisconsin breast cancer data set (WBCD), Ljubljana breast cancer data set, Nursery data set, Adult data set, Hepatitis data set, Dermatology data set, and Cleveland heart disease data set from the area of medicine while Tic-Tac-Toe comes from the game domain, and Credit application approval data set from financial decision making have been used.

Pappa and Freitas (2009), present a Multi-Objective grammar-based genetic programming (MOGGP) system that automatically evolves complete rule induction algorithms following the sequential-covering approach, which produces both accurate and compact rule models.

Chromosome representation: Individuals are represented by a linear genome which is generated independently from the grammar. When evaluating the individuals, a genotype/phenotype mapping is made, and the genetic material is used to select appropriate production rules from the grammar.

Parameters: The individuals in the *initial population* are built through a set of derivation steps, and production rules are applied to the tree until all the leaf nodes are represented by terminals. The *Population size* is 100, with 30 *generations*. A *crossover rate* of 0.7 and a *mutation rate* of 0.25 have been used. Number of generations is used as *stopping criteria.*

Reproduction operators: The individuals are selected using a *tournament selection* where the chromosomes are simply compared among themselves using the two objectives to be optimized and the best is chosen. Crossover and mutation operations are restricted to non-terminals, and different non-terminals might be assigned different crossover/mutation rates. In the case of *crossover,* a non-terminal *Nx* is randomly selected from the tree of the first

individual $I1$. After that, the system searches for the same non-terminal *Nx* in the tree of individual $I2$. If *Nx* is present in $I2$, the sub-trees rooted at *Nx* in individuals $I1$ and $I2$ are swapped (respecting the maximum individual size parameter). If *Nx* is not present in $I2$, the operation is not performed. During *mutation* a random non-terminal *Nx* is selected from the derivation tree of the individual, the sub-tree rooted at *Nx* is deleted, and a new sub-tree is created by following the productions of the grammar starting from *Nx*.

Objectives, fitness function and the Optimization strategy: The predictive accuracy obtained when classifying a set of test examples, and the size (complexity) of the model (rule set) used to classify new examples are taken as objectives to be maximized and minimized respectively. Pareto-multi-objective optimization concept is used to find the set of optimal solutions. The individuals have to be selected according to a relationship of Pareto dominance instead of a simple fitness value. In the case of the elitism scheme used by the MOGGP algorithm, all the solutions in the current estimated Pareto front are passed to the next generation by elitism, as long as their number does not exceed half of the size of the population. If they do, then the best individuals are given priority. The best individuals are returned to the user, in the last generation, and then tested in a meta-test set.

Data sets and area of application: Data sets used by the MOGGP in the meta-training set are Monks-2, Monks-3, Balance scale, Lymph, Zoo, Glass, Pima Indians diabetes, Hepatitis, Vehicle, and Vowel. Data sets used by the MOGGP in the meta-test set are Credit application approval, Segment Sonar, Heart-C, Ionosphere, Monks, Mushroom, Wisconsin Breast Cancer, Promoters and Splice. All data sets have been taken from UCI repository (Newman et al. 1998) and span a variety of application areas.

5.2.1 Discussion: EMOO system characteristics for rule mining using GP

Genetic programming systems are evolutionary systems whose outputs are program constructs that can be used under various environments. There are a few systems for multi objective rule knowledge discovery that have been modeled using GP as evolutionary strategy. Table 2 summarizes the EMOO system characteristics for rule mining using GP.

Chromosome representation: The chromosome representations in GPs are usually decision trees, graphs, grammar based constructs and the like. Zhao (2007) use trees with function and terminal nodes, while Expression trees whose Leaf nodes contain ATs and internal nodes contain a Boolean operator is used by Reynolds and De la Iglesia (2007). Baykasoglu and Ozbakir (2007) use fixed length linear strings of chromosomes to represent programs in the form of expression trees. The latest work by Pappa and Freitas (2009) use linear genome directly encoded from solution space for their grammar based GP.

Initialization of population: The initialization of population is done randomly or based on a-priori knowledge from earlier results in Zhao (2007), while randomly generated balanced trees of depth two are used by Reynolds and De la Iglesia (2007). Baykasoglu and Ozbakir (2007) use a Generative procedure to generates initial rules and Pappa and Freitas (2009) derive their initial population using a set of generative steps.

Parameters: In Zhao (2007), all the parameters are specified by the user through an interface. In the other systems the following parameters are used.

Table 2 EMOO system characteristics for rule mining using GP

EMOO system by	Chromosome representation	Initialization of the rules population	Pop. size: No. of generations Cross over rate (CR) Mutation rate (MR)	Objectives optimized	Reproduction operators			MOO strategy for optimization
					Selection	Crossover	Mutation	
Zhao (2007)	Tree with terminal and function node	A-priori knowledge from earlier results is used or randomly	Specified by user through Java interface	Specificity and sensitivity	Tournament	Single point	Replace sub-tree with random sub-tree	Pareto optimality
Reynolds and De la Iglesia (2007)	Expression trees. Leaf nodes contain ATs, while internal nodes contain a Boolean operator	Randomly generated balanced trees of depth two	10, 20, 50, 100, 200, 500 30 CR: 0.0, 0.2, 0.4, 0. 6, 0.8, 1.0 MR: 0.5—AT is mutated 0.25—AT and parent removed 0.25—node is added	Misclassification cost and complexity	Random	Single point	Random	NSGA II
Baykasoglu and Ozbakir (2007)	Fixed length linear strings chromosomes to represent programs as expression trees	Generative procedure generates initial rules	250 250 CR: 0.9 MR: 0.2	Sensitivity and Specificity	Binary tournament	Single point	Random	Fitness function as product of Specificity and sensitivity
Pappa and Freitas (2009)	Linear genome directly encoded from solution space	Through a set of derivation steps	100 30 CR: 0.7 MR: 0.25	Accuracy and Complexity	Binary tournament	Random	Random	Pareto domination

Population size: Reynolds and De la Iglesia (2007) have experimented with a range of population sizes from 10 to 500 individuals. Baykasoglu and Ozbakir (2007) use 250 individuals while Pappa and Freitas (2009) use a population size of 100.

No. of generations: Baykasoglu and Ozbakir (2007) and Pappa and Freitas (2009) have used 30 generations whereas Reynolds and De la Iglesia (2007) have used 250 generations or iterations.

Cross over rate: Reynolds and De la Iglesia (2007) have experimented with a set of cross over rates ranging from 0.0 to 1.0 while Baykasoglu and Ozbakir (2007) and Pappa and Freitas (2009) use crossover rates of 0.9 and 0.7 respectively.

Mutation rate: Reynolds and De la Iglesia (2007) use different mutation rates where it is 0.5 when a random AT is mutated, 0.25 when AT and parent are removed and 0.25when a random node is added. The mutation rate is 0.2 in Baykasoglu and Ozbakir (2007) and 0.25 in Pappa and Freitas (2009). As can be observed high mutation rates are used by GPs rather than Gas.

Objectives optimized: Specificity and sensitivity are used as objectives for optimization in Zhao (2007) and Baykasoglu and Ozbakir (2007). Reynolds and De la Iglesia (2007) use misclassification cost and complexity, whereas Pappa and Freitas (2009) use accuracy and complexity as objectives.

Reproduction Operators: Selection: Random selection is used by Reynolds and De la Iglesia (2007), Binary tournament selection is used by Baykasoglu and Ozbakir (2007) and Pappa and Freitas (2009) and tournament selection is used by Zhao (2007).

Crossover: Random crossover is performed in Pappa and Freitas (2009) whereas single point crossover is preferred in the other three systems.

Mutation: Random mutation is used by all the systems.

MOO strategy for optimization: Zhao (2007) and Pappa and Freitas (2009) use Pareto optimality as optimization strategy while Reynolds and De la Iglesia (2007) use NSGA II. A fitness function which is the product of specificity and sensitivity is used by Baykasoglu and Ozbakir (2007).

Data sets and area of application: Most of the algorithms have been tested on benchmark data sets from the UCI machine learning repository (Newman et al. 1998). The area of application includes biology, finance, vehicles, alphabets, etc.

6 Hybridization techniques

Integrating techniques from different disciplines has been shown to improve the performance of a system. Table 3 provides a summary of the techniques that have been and can be integrated with EMOO systems for rule knowledge discovery to improve the performance as well as usability of the system.

Table 3 Hybridization techniques for rule mining with EMOO systems

EMOO system by	Intelligent agents	Data Pre-processing-fuzzification/ discretization	Meta data and Meta heuristics	Parallelism	User interaction	Interface for rule visualization
De la Iglesia et al. (2003, 2005)	x	Pre-screening	√	Partial classification	√	x
Reynolds and De la Iglesia (2006)	x	x	√	Partial classification	√	x
Reynolds and De la Iglesia (2009) Reynolds et al. (2009)	x	x	√	Partial classification	√	x
Narukawa et al. (2005), Ishibuchi and Nojima (2005),	x	Fuzzification	√	Parallelism	x	x
Ishibuchi et al. (2007), Ishibuchi (2007)	x	Fuzzification	√	Parallelism	x	x
Wang et al. (2005), Tsang et al. (2005, 2007)	√	Fuzzification	√	Parallelism	x	x
Berlanga et al. (2006), Del Jesus et al. (2007)	x	Fuzzification	√	Partial classification	x	x
Dehuri and Mall (2006)	x	x	√	Separate run for each class	√	x
Giusti et al. (2008)	x	x	√	Parallelism	√	x
Khabzaoui et al. (2008)	x	x	√	Parallelism	x	x
Casillas et al. (2008, 2009)	x	Fuzzification	√	Parallelism	x	x
Zhao (2007)	x	Normalization	x	Parallelism	√	√
Reynolds and De la Iglesia (2007)	x	Pre-screening	√	Parallelism	√	x
Baykasoglu and Ozbakir (2007)	x	Discretization	x	Separate run for each class	x	x
Pappa and Freitas (2009)	x	Pre-screening	√	Sequential covering	√	x

6.1 Use of intelligent agents

The emergence of intelligence in agent-mining interaction may massively strengthen the problem-solving capability of an intelligent system and DM techniques such as association rule extraction have no equivalent in agent systems (Cao 2009). In the EMOO systems so far discussed only three systems use intelligent agents. Many of the other techniques like parallelism, user interaction and Meta heuristics can be incorporated into the system by using agent technology.

The EMOO systems by Wang et al. (2005) and Tsang et al. (2005, 2007) use Arbitrator agent and Fuzzy set agents. The Fuzzy set agents initializes its own control genes and parameter genes randomly. They are different with one another and selected randomly with the same probability for reproduction. They autonomously determine their own fuzzy sets information such as the number and distribution of the fuzzy sets. They can further consider the interpretability of fuzzy systems with the aid of hierarchical chromosome formulation and interpretability-based regulation method. In addition, the fuzzy set agents can cooperate with each other to exchange their fuzzy sets information and generate offspring agents. The parent agents and their offspring compete with each other through the arbitrator agent based on the criteria associated with the accuracy and interpretability to allow them to remain competitive enough to move into the next population. Based on the multiple criteria about the accuracy and interpretability of fuzzy systems, the elite agents are retained in the multi-agent system, whereas the obsolete agents are destroyed by the arbitrator agent. Arbitrator agent and the Fuzzy set agents are distributed independently. Fuzzy set agents obtain information from the arbitrator agent. Arbitrator agent uses the NSGA-II algorithm to evaluate the fuzzy set agents. If Crossover and mutation operations introduce the same rules, the fuzzy set agent will check the offspring fuzzy rule base to delete the redundant rules. Incorporation of agents with DM is said to have improved the performance of the rule mining system.

6.2 Data pre-processing: fuzzification Vs data discretization

Data discretization using methods like normalization or linear transformations can be used to create a homogeneous chromosome structure. But the use of fuzzification not only allows a homogeneous structure but also allows for greater understandability of the system making it more usable. The systems discussed use pre processing through transformation or normalization and fuzzification before applying the mining algorithm. Any records with missing values were removed from the database prior to the application of the algorithm by De la Iglesia et al. (2003), and Reynolds and De la Iglesia (2007) whereas a scaling procedure is used by De la Iglesia et al. (2005) that transforms any number in the range of possible values using $p$bits to a number in the range of values that the attribute can take. The continuous attributes in data sets are discretized in Baykasoglu and Ozbakir (2007) after removing records with missing values. Zhao (2007) use normalization using a linear transformation. In Pappa and Freitas (2009), the data sets which compose the meta-training set have been selected based on the execution time of the rule induction algorithms, so that the data sets leading to faster runs of the rule induction algorithms were included in the meta-training set.

Genetic fuzzy systems are learning techniques that use genetic algorithms to optimize different components of fuzzy rule-based systems (Casillas et al. 2008). Moreover Fuzzy rules can be usually interpreted in a linguistic manner because they are described by linguistic values such as *low* and *high*. Fuzzy rule-based systems have high accuracy as well as high interpretability. Ishibuchi (2007) explain three types of partitions: homogeneous fuzzy partitions, inhomogeneous entropy-based interval partitions, and inhomogeneous fuzzy partitions

derived from the interval partitions. Experimental results have shown that the fuzzification of interval rules improves their generalization ability for many data sets. Class entropy measure is used to divide a continuous attribute into K intervals by Ishibuchi and Nojima (2005) while Wang et al. (2005) propose an agent based evolutionary approach to extract interpretable fuzzy rule-based knowledge. Casillas et al. (2008) propose Pitts-DNF-C, a multi-objective Pittsburgh-style Learning Classifier System that evolves a set of DNF-type fuzzy rules for classification tasks. These types of learning methodologies aim at obtaining highly accurate and understandable models. The two objectives being contradictory since more accurate models tend to be less interpretable. Fuzzy rule allows representing knowledge about patterns of interest in an explanatory and understandable form which can be used by the expert as stated by Berlanga et al. (2006) and Del Jesus et al. (2007). In their proposal for discovering fuzzy rules in disjunctive normal form, the fuzzy sets corresponding to the linguistic labels for a linguistic variable are specified by means of the corresponding membership functions which can be defined by the user or defined by means of a uniform partition if expert knowledge is not available. In this algorithm, uniform partitions with triangular membership functions are used.

6.3 Meta data and Meta heuristics

Hybridizing metaheuristic approaches becomes a common way to improve the efficiency of optimization methods especially in DM for rule mining. Incorporating knowledge into operators using Meta data will improve exploiting as well as exploring interesting areas in the search space (Jourdan et al. 2006) and hence enhance the performance. Also Meta heuristics can enhance mining a diverse set of rules. De la Iglesia et al. (2005) have created a modification of NSGA-II by introducing the concept of rule dissimilarity in the crowding measure. This is reported to have increased the diversity of rules in some areas of the Pareto front in terms of support sets. Again cc-optimality is a new heuristic used by them for choosing diverse and novel rules. Two simple approaches using the concepts of pc-dominance and cc-dominance have been used to remove uninteresting rules in Reynolds and De la Iglesia (2006). The approach considered involves modifying the dominance relation without explicitly adding a third objective by introducing three novelty measures namely absolute novelty, relative novelty and apparent rule novelty to present to the user novel rules. This dominance relation allows the multi-objective metaheuristic to find novel rules that would otherwise be dominated (Reynolds and De la Iglesia 2006). Further in the extended work reported by Reynolds and De la Iglesia (2009), a new MO algorithm for the task of partial classification based on a metaheuristic known as greedy randomized adaptive search procedure (GRASP) is introduced. An initial front of solutions is generated before the application of local search. Local search is then applied to the front as a whole which eliminates the need for weighted utility functions in the local search. While Meta data stored in the form of a match table is used by Reynolds et al. (2009) in their MO algorithm for rule selection which uses GRASP. The rule selection algorithm does not need access to the data or to the details of the rules. It only requires access to which rules match which records; which records are in the class of interest; and the complexity of each of the rules. Once the data and rules have been read in, a match table containing this information is created. The algorithm then need only refer to the match table thus reducing the memory overhead.

Specific operators have been proposed by Khabzaoui et al. (2008), for the association rules problem like, Crossover by value exchange, Crossover by insertion, and four mutation operators like Value and Attribute mutation, Insert and Delete operators. An adaptive mutation rate which changes according to the improvement in the solution is also proposed. Choosing

similar rule sets as parents for crossover operations and using biased selection probability of parents toward rule sets with high accuracy are used as meta-heuristics by Narukawa et al. (2005). While Ishibuchi and Nojima (2005), Ishibuchi et al. (2007) and Ishibuchi (2007), use two problem-specific heuristic tricks. One is biased mutation probabilities where a larger probability is assigned to the mutation from 1 to 0 than that from 0 to 1. This heuristic trick is used to efficiently decrease the number of rules in each rule set by the mutation operation. The other is the removal of unnecessary rules. Restricting candidates to Pareto-optimal and near Pareto-optimal rules using a measure known as ε-dominance have been proposed as Meta heuristic in Ishibuchi et al. (2007).

The crossover operator considered by Dehuri and Mall (2006) is based on uniform crossover in their Improved Niched Pareto Genetic Algorithm (INPGA). There is a probability for applying crossover to a pair of individuals and another probability for swapping each gene's value in the genome (rule antecedent) of two individuals. After crossover is complete, the algorithm analyses if any invalid individual was created. If so, a repair operator is used to produce valid-genotype individuals. Besides crossover and mutation, the insert and remove operators directly try to control the size of the rules being evolved, thereby influencing the comprehensibility of the rules. Also the tournament selection is altered in two ways. First, Pareto domination tournament is introduced. Second, when a non-dominant tournament (i.e., a tie), sharing is implemented to determine the winner. The algorithm uses the following strategy: Finds out the center of gravity of both niche radius and calculates the standard deviation of each point of both radii, finally the candidate having larger SD is chosen.

Narukawa et al. (2005) use biased selection probability of parents toward rule sets with high accuracy. A heuristic measure is used in an iterative fuzzy genetics-based machine learning algorithm called SLAVE. A pre-specified number of promising short fuzzy rules for each class are chosen using the SLAVE measure. They also use two problem specific heuristics to efficiently decrease the number of fuzzy rules in each rule set during the execution of the NSGA-II algorithm. One is biased mutation where a larger probability is assigned to the mutation from 1 to 0 than that from 0 to 1. The other is the removal of unnecessary fuzzy rules. Biased uniform mutation operator is also proposed in Berlanga et al. (2006) and Del Jesus et al. (2007).

Mutation in Reynolds and De la Iglesia (2007) changes the index of the bound by up to 20% of the number of values that occur in the database. Giusti et al. (2008) use the rule's contingency matrix as Meta data, which is used to estimate rule quality levels according to different quality criteria. In order to generate a generic classifier, the individuals evolved by the GGP by Pappa and Freitas (2009) are evaluated using a fitness function based on their accuracy on a set of data sets named meta-training set. At the end of the evolutionary process, the individuals are then validated in a new set of data sets named meta-test set.

Confusion matrix is used as Meta data to calculate various measures like precision, F-measure, and overall accuracy by Tsang et al. (2007) whereas a Covering Hyper matrix is used in Casillas et al. (2008, 2009). This matrix is used when generating new rules to efficiently avoid over-generality or generating rules in regions without training data. This structure stores the label combinations of the antecedent that cover all the examples in the training data set. It is also responsible for avoiding over generality in the rule sets. Antecedent Mutation Operators expand the variable, i.e., add a new linguistic term to the variable, or contract the variable, i.e., remove a linguistic term form the variable. The operator analyzes all the possible mutations that ensure consistency and non-over generality of the resulting rule set. Consistency after mutation is checked by analyzing the collision of the new rule with the remaining rules of the individual.

However incorporation of a separate Meta data structure to store the good rules (even if they are dominated by others) encountered during each generation along with its performance measure could still be useful. This will enable to use them for reproduction in future generations which may generate a diverse as well as a good set of rules which would otherwise be lost.

6.4 Parallelism

The EMOO algorithms use different strategies for rule induction/selection. Parallel rule induction or selection where rules for all the classes are created simultaneously is used by Narukawa et al. (2005), Ishibuchi and Nojima (2005), Ishibuchi et al. (2007), Ishibuchi (2007), Wang et al. (2005), Tsang et al. (2005, 2007), Khabzaoui et al. (2008), Casillas et al. (2008, 2009), Zhao (2007), and Reynolds and De la Iglesia (2007). Whereas the algorithm has to be executed separately for each class in the systems by Dehuri and Mall (2006) and Baykasoglu and Ozbakir (2007). Partial classification rules or rule for a particular class of interest is generated by the algorithms in De la Iglesia et al. (2003, 2005), Reynolds and De la Iglesia (2006), Berlanga et al. (2006), Del Jesus et al. (2007), Reynolds and De la Iglesia (2009), and Reynolds et al. (2009). Sequential covering approach which removes the records that are covered by a rule at the end of each iteration is used by Pappa and Freitas (2009).

6.5 User interaction

The main advantage of Interactiveness of the system with the user is that rules that are comprehensible and considered good by the user can be discovered, however this may slow down the system (Freitas 2007). However participation of the user in the process is essential to improve the chance that discovered knowledge will be actually useful for the user (Freitas 2004). The ultimate objective of multi-objective algorithms is to guide the user's decision making, through the provision of a set of solutions that have differing trade-offs between the various objectives (Reynolds and De la Iglesia 2006). Some systems allow the user to specify the metrics for optimization and/or the threshold values for rule selection while a very few systems allow the user to interact with the system during execution.

De la Iglesia et al. (2005) propose to use Pareto-based MOEA to deliver nuggets that are in the Pareto optimal set according to some measures of interest which can be chosen by the user. A strong rule is defined as one that meets certain confidence and coverage thresholds. Those thresholds are normally set by the user and are based on domain or expert knowledge about the data. Also a binary string represented by a set of Gray-coded lower and upper limits, where each limit is allocated a user-defined number of bits $p$is used. In Reynolds and De la Iglesia (2006), the user is presented with a set of descriptions of the class of interest from which he may select a subset whereas in Reynolds and De la Iglesia (2009), the mutation rate is provided by the user. The user specifies the goal attribute that is of interest to him in Dehuri and Mall (2006). In the case of the work reported in Giusti et al. (2008), the population which is a set of rules with specific properties desired by the user, is used in each generation. The MOO algorithm proposed by Zhao (2007) allows the decision maker to specify partial preferences on the conflicting objectives, such as false negative vs. false positive, sensitivity vs. specificity, and recall vs. precision to reduce the number of alternative solutions. The system visualizes the progress of the evolution of solutions such that the decision maker can decide to stop the procedure when satisfactory solutions have been found or when the solutions on the front appear to have stabilized.

The EMOO algorithm by Reynolds and De la Iglesia (2007) produces a range of rules from the training data with differing trade-offs between misclassification cost and rule complexity. In order to give the client some idea as to how well the rules produced generalize, the rules are reevaluated on new validation data. At this point, the client selects a rule. The client can be provided with a range of models with different trade-offs between rule complexity and misclassification costs. This allows the client to select a rule that is accurate enough while also being comprehensible. In Pappa and Freitas (2009), the best estimated Pareto front found by the GGP is the set of solutions returned to the user, who can then select the best one according to his/her preference in the system.

6.6 User interface for visualization

ROC graphs are an increasingly popular way of analyzing the performance of a classifier. In a ROC graph, the ideal performance corresponds to the upper-left point (0, 1) (Freitas 2004).

But they can just be used for evaluating the performance of the algorithms by using visualization. Moreover this seems intuitive only when there are just two dimensions to visualize. As the true Pareto front is not known, it is not possible to compare to it in order to evaluate performance (De la Iglesia et al. 2003). The area under the curve (AUC) is sometimes used as an aggregate performance measure where a recall–precision curve or a false negative–false positive curve is generated and visualized. In order to understand the results of applying distance metrics to the rules obtained by the NSGA-II algorithm, De la Iglesia et al. (2005) apply a clustering algorithm to cluster similar rules together which help in the presentation of results. The system by Zhao (2007) supports three ways for specifying objective preferences and three typical visualization methods. The visualization method used is ROC. The Pareto front can be visualized periodically. The system visualizes the progress of the evolution of solutions such that the decision maker can decide to stop the procedure when satisfactory solutions have been found or when the solutions on the front appear to have stabilized.

But visualization for presenting the rules to the user in terms of an interface is seldom discussed in any of the works discussed so far. A good interface for rule mining systems must be able to present the actual "If…Then" rules to the user. The user must be able to experiment with various rules, for example by changing the attribute values interactively and visualizing the effect of the changes he has made in the form of various metrics.

7 Summary

Association and classification rules are highly understandable representations in data mining. The rules have various properties called metrics which fall into two categories namely objective and subjective measures. In order for the system to be useful, the rules presented to the user should be compact, understandable and most importantly usable. Therefore the rules should have certain properties as desired by the user. Thus rule knowledge discovery becomes a multi objective optimization problem. Since evolutionary systems are best at solving multi objective optimization problems, they are extensively used by researchers for rule mining. But these knowledge discovery systems can further be improved to make the knowledge into actionable knowledge by integrating techniques like intelligent agents, parallelism, and a good interactive user interface for visualization as well as experimenting with the presented rules. Further the performance of the systems can be improved by careful experimentation about the representation of the solution and the search space, incorporation of Meta data and

Meta heuristics into the reproduction operators and fine tuning of various parameters that influence the evolutionary rule mining algorithm.

Integration of intelligent agent technology will make the system more interactive, and will allow for better use of Meta data and Meta heuristic. Moreover different types of agents can be used for discovering rules for different classes in parallel instead of executing the algorithm once for each class. Agents can also be used for converting the discovered knowledge into actionable knowledge by embedding the various rules into agents to solve a specific problem. Further if the user is allowed to experiment with the rules presented to him, he will tend to understand the system well and thus use it.

References

Abe H, Tsumoto S (2008) Analyzing correlation coefficients of objective rule evaluation indices on classification rules. In: Wang G, et al (eds) RSKT 2008, LNAI 5009. Springer, Berlin, pp 467–474

Baykasoglu A, Ozbakir L (2007) MEPAR-miner: multi-expression programming for classification rule mining. Eur J Oper Res 183:767–784

Berlanga F, del Jesus MJ, Gonzalez P, Herrera F, Mesonero M (2006) Multi-objective evolutionary induction of subgroup discovery fuzzy rules: a case study in marketing. In: Perner P (ed) ICDM 2006, LNAI 4065. Springer, Berlin, pp 337–349

Cao L (2009) Introduction to agent mining interaction and integration. In: Cao L (ed) Data mining and multiagent integration, LLC 2009. Springer, Berlin pp 3–36

Casillas J, Orriols-Puig A, Bernad/o-Mansilla E (2008) Toward evolving consistent, complete, and compact fuzzy rule sets for classification problems. In: proceedings of 3rd international workshop on genetic and evolving fuzzy systems, Witten-Bommerholz, Germany, pp 89–94

Casillas J, Pedro Martinez AE, Benitez Alicia D (2009) Learning consistent, complete and compact sets of fuzzy rules in conjunctive normal form for regression problems. Soft Comput 13:419–465

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197

Dehuri S, Mall R (2006) Predictive and comprehensible rule discovery using a multi-objective genetic algorithm. Knowl Syst 19:413–421

De la Iglesia B, Philpott MS, Bagnall AJ, Rayward-Smith VJ (2003) Data mining rules using multi-objective evolutionary algorithms. In: proceedings of 2003 IEEE congress on evolutionary computation, pp 1552–1559

De la Iglesia B, Reynolds Alan, Rayward-Smith Vic J (2005) Developments on a multi-objective metaheuristic (MOMH) algorithm for finding interesting sets of classification rules. In: proceedings of third international conference on evolutionary multi-criterion optimization, EMO 2005, LNCS 3410. Springer, Berlin, pp 826–840

Del Jesus MJ, Gonzalez P, Herrera F, Mesonero M (2005) Evolutionary induction of descriptive rules in a market problem. Stud Comput Intell (SCI) 5:267–292

Del Jesus MJ, Gonzalez P, Herrera F (2007) Multi-objective genetic algorithm for extracting subgroup discovery fuzzy rules. In: proceedings of the 2007 IEEE symposium on computational intelligence in multi-criteria decision making (MCDM 2007), pp 50–57

Freitas AA (2004) A critical review of multi-objective optimization in data mining: a position paper. SIGKDD Explor 6(2):77–86

Freitas AA (2007) A review of evolutionary algorithms for data minin, soft computing for knowledge discovery and data mining. Springer, USA 79–111

Giusti R, Gustavo EA, Batista PA, Prati Ronaldo C (2008) Evaluating ranking composition methods for multi-objective optimization of knowledge rules. In: proceedings of eighth international conference on hybrid intelligent systems, pp 537–542

Ishibuchi H (2007) Evolutionary multi-objective design of fuzzy rule-based systems. In: proceedings of the 2007 IEEE symposium on foundations of computational intelligence (FOCI 2007), pp 9–16

Ishibuchi H, Namba S (2004) Evolutionary multiobjective knowledge extraction for high-dimensional pattern classification problems, parallel problem solving from nature—PPSN VIII, LNCS 3242. Springer, Berlin 1123–1132

Ishibuchi H, Nojima Y (2005) Comparison between fuzzy and interval partitions in evolutionary multi-objective design of rule-based classification systems. In: proceedings of the 2005 IEEE international conference on fuzzy systems, pp 430–435

Ishibuchi H, Kuwajima I, Nojima Y (2007) Multi-objective classification rule mining, natural computing series. Springer, Berlin 219–240

Jourdan L, Dhaenens C, Talbi E-G (2006) Using data mining techniques to help metaheuristics: a short survey, Hybrid Metaheuristics, LNCS 4030. Springer, Berlin 57–69

Khabzaoui M, Dhaenens C, Talbi EG (2008) Combining evolutionary algorithms and exact approaches for multi-objective knowledge discovery. RAIRO Oper Res 42: 69–83. doi:10.1051/ro:2008004

Narukawa K, Nojima Y, Ishibuchi H (2005) Modification of evolutionary multi-objective optimization algorithms for multi-objective design of fuzzy rule-based classification systems. In: proceedings of the 2005 IEEE international conference on fuzzy systems, pp 809–814

Newman D, Hettich S, Blake C, Merz C (1998) UCI repository of machine learning databases, Department of Information and Computer Science, University of California at Irvine http://www.ics.%20uci.edu/?mlearn/MLRepository.html

Pappa GL, Freitas AA (2009) Evolving rule induction algorithms with multi-objective grammar-based genetic programming. Knowl Inf Syst 19:283–309

Reynolds AP, de la Iglesia B (2006) Rule induction using multi-objective metaheuristic: encouraging rule diversity, In: proceedings of IJCNN 2006, pp 6375–6382

Reynolds AP, de la Iglesia B (2007) Rule induction for classification using multi-objective genetic programming. In: proceedings of 4th international conference on evolutionary multi-criterion optimization, LNCS 4403. Springer, Berlin, pp 516–530

Reynolds AP, de la Iglesia B (2009) A multi-objective GRASP for partial classification. Soft Comput 13(3):227–243

Reynolds AP, Corne David W, de la Iglesia B (2009) A multi-objective grasp for rule selection. In: proceedings of the 11th annual conference on genetic and evolutionary computation, GECCO'09, Montréal Québec, Canada, pp 643–650

Tsang C-H, Kwong S, Wang H (2005) Anomaly intrusion detection using multi-objective genetic fuzzy system and agent-based evolutionary computation framework. In: proceedings of the fifth IEEE international conference on data mining (ICDM'05), pp 789–792

Tsang C-H, Kwong S, Wang H (2007) Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection. Pattern Recogn 40:2373–2391

Wang H, Kwong S, Jin Y, Wei W, Man KF (2005) Agent based evolutionary approach for interpretable rule-based knowledge extraction. IEEE Trans Syst Man Cybern 35(2):143–155

Zhang Y, Rockett P (2007) A comparison of three evolutionary strategies for multi-objective genetic programming. Artif Intell Rev 27:149–163

Zhao H (2007) A multi-objective genetic programming approach to developing Pareto optimal decision trees. Decis Supp Syst 43:809–826

Zitzler E, Laumanns M, Thiele L (2001) SPEA2: improving the strength Pareto evolutionary algorithm for multi-objective optimization. In: proceedings of evolutionary methods for design, optimization and control with applications to industrial problems (EUROGEN2001), Barcelona, pp 95–100