We are IntechOpen, the world's leading publisher of Open Access books
Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

MO-Miner: A Data Mining Tool Based on Multi-Objective Genetic Algorithms

Gina M. B. de Oliveira[1], Luiz G. A. Martins[1] and Maria C. S. Takiguti[2]
[1]Universidade Federal de Uberlândia (UFU)
[2]Universidade Presbiteriana Mackenzie
Brazil

1. Introduction

Data mining (DM) is the process to extract previously unknown and implicit information from large databases. Several techniques have been used to discover such kind of knowledge; most of them derived from machine learning and statistics. The majority of these approaches focus on the discovery of accurate knowledge. However, this knowledge should be useless if it does not offer some kind of surprisingness to the final user. Based on this idea, some investigations were started recently with the aim of extracting accurate and interesting information from datasets. In this sense, data mining can be faced as a multi-objective problem.

Different machine learning approaches have been employed to perform data mining tasks. Most of them are based on evolutionary methods, like genetic algorithms (Goldberg, 1989). Another advantage of genetic algorithms is that they can be adapted to treat multi-objective problems in a Pareto sense. Various multi-objective genetic algorithms have been proposed in the literature, like the method known as *Non-dominated Sorting Genetic Algorithms* (NSGA) (Srinivas & Deb, 1994).

The task performed in a data mining process depending on what kind of knowledge someone needs to extract. The main types of tasks performed by DM algorithms are classification, association, clustering, regression, sequences analysis, summarizing and dependence modelling. Classification task searches for the knowledge able to predict the value of a previously defined goal attribute based on other attributes. This knowledge is often represented by IF-THEN rules and it is the most investigated data mining task. Dependence modelling task can be seen as a generalization of classification. It also aims to discover rules able to predict the goal attribute value, from values of the prediction attributes. However, in dependence modelling, there are more than one goal attribute. Initially, a small set of goal attributes is specified, whose prediction is considered interesting. These attributes may occur in the consequent or in the antecedent parts of the rule. The others attributes occur only in the antecedent.

A multi-objective evolutionary data mining environment named *MO-miner* was implemented based on the family of algorithms called non-dominated sorting genetic algorithms (NSGA). The two desirable properties of the rules being mined - accuracy and interestingness - are simultaneously manipulated. *MO-miner* keeps the metrics related to

these properties separated during the evolution, as different objectives used in the fitness calculus in a Pareto-based approach. This chapter describes the investigation performed using *MO-miner* in the mining of accurate and interesting rules for the dependence modelling task. *MO-miner* was tested in two public domain databases named *Zoo* and *Nursery*. The results obtained by *MO-miner* had been compared with those generated by a standard GA in order to identify the benefits related to the multi-objective approach.

Section 2 presents a brief review on multi-objective genetic algorithms with emphasis on the method used here, the NSGA proposed by Srinivas and Deb (1994). The major data mining tasks are presented in Section 3. This section also presents a review of recent works which have also investigated the application of multi-objective approaches to data mining tasks. Section 4 describes the multi-objective environment named *MO-miner*, which was implemented to perform the dependence modelling task. The two data sets used to evaluate the environment are characterized in Section 5. Section 6 presents the results and their analysis regarding the application of *MO-miner* and a comparative single-objective environment to the data sets. Finally, Section 7 presents the major conclusions of this work.

2. Evolutionary algorithms and multi-objective problems

Evolutionary algorithms (EAs) are computational search methods gleaned from biological evolution. In these methods a population of candidate solutions is generated by random means, and then evaluated in terms of how close they are to the desired solution of the problem at issue. According to the score derived from that evaluation, a selection procedure then determines which subpopulation will be used as the basis for the creation of another population of candidate solutions. The new population is obtained through the application of genetic operators over the selected subpopulation, the expectation being that the new population be, in average, better than its predecessor. This process is repeated over generations, until a population emerges with a satisfactory candidate solution, or until some predefined computational constraint (such as a certain number of generations) is achieved. Evolutionary methods enable us to obtain solutions to problems for which there are no known exact algorithms or, for those where an algorithm exists, but the solution requires a long time of processing to be found (Mitchell, 1997). The genetic algorithm (GA) proposed by Holland is the most well-known EA (Goldberg, 1989). The success of GA search is due to the symbiosis of three operators: selection, crossover and mutation.

2.1 Pareto optimum and multi-objective genetic algorithms

Many real world problems involve simultaneous optimization of multiple objectives, so that it is not always possible to achieve an optimum solution in respect to all of them individually considered. In this kind of problem, there is a set of solutions better than all the other solutions in the search space (Srinivas & Deb, 1994), which is named the Pareto optimum or set of non-dominated solutions.

Suppose that there are N objectives $f_1, f_2, \ldots f_N$ to be simultaneously optimized. A solution A is said to be dominated by another solution B, or B dominates A, if B is better than A in relation to at least one of the objectives f_i, and is better than or equal to A in relation to all other objectives $f_1, \ldots f_{i-1}, f_{i+1} \ldots f_N$. Two solutions A and B are non-dominated in relation to each other if A does not dominate B and B does not dominate A. For example, suppose that functions f1 and f2 in Figure 1 must be simultaneously maximized. One can affirm that solution A is better than solutions C and D; that is, C and D are dominated by A. However,

in the case of solutions A and B, it is not possible to affirm which one is the best. Therefore, one can say that solutions *A* and *B* are non-dominated and they dominate solutions *C* and *D*.
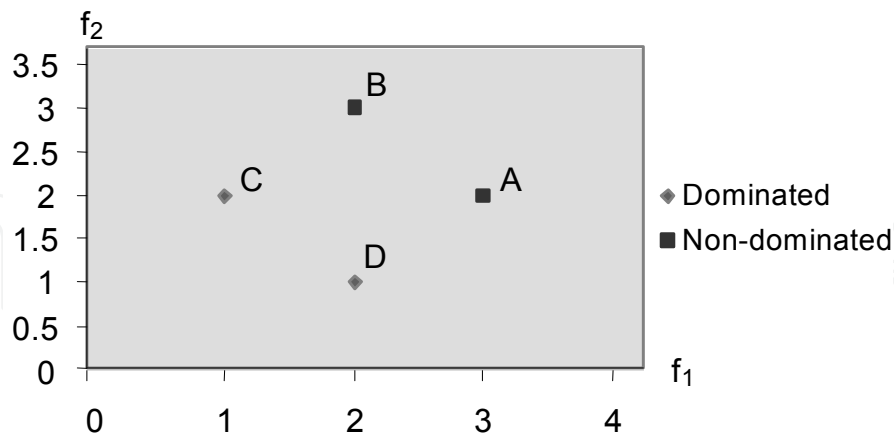

Fig. 1. Example of dominated and non-dominated solutions when maximizing f1 and f2.

The Pareto optimum is the set of non-dominated solutions considering the entire search space, that is, any candidate solution that is better than one of the solutions from the Pareto optimum in respect to one of the objectives, is guaranteed to be worse with respect to another objective.

Multi-objective evolutionary methods try to find this solution set by using each objective separately, without aggregating them as an unique objective (Coello, 1999). Some of the most traditional multi-objective evolutionary methods are VEGA (Vector Evaluated Genetic Algorithm; Shaffer, 1985), NSGA (Non-dominated Sorting Genetic Algorithms; Srinivas & Deb, 1994), MOGA (Multiple Objective Genetic Algorithm; Fonseca & Fleming, 1993), and NPGA (Niched Pareto Genetic Algorithm; Horn & Nafpliotis, 1993); the NSGA is briefly described below, insofar as it provided the procedures used in the present work.

More recently, new evolutionary methods have been developed, characterised by the fact that – unlike the methods mentioned above – an elitist strategy was incorporated in them, leading to methods such as NSGAII (Non-dominated Sorting Genetic Algorithm II; Deb *et al.*, 2000), SPEA (Strength Pareto Evolutionary Algorithm; Zitzler & Thiele, 1999), SPEA2 (Strength Pareto Evolutionary Algorithm; Zitzler *et al.*, 2001), PAES (Pareto Archieved Evolution Strategy; Knowles & Cornes, 1999) and PESA (Pareto Enveloped-based Selection Algorithm; Knowles *et al.*, 2000).

2.2 Non-dominated sorting genetic algorithm

The Non-dominated Sorting Genetic Algorithm (NSGA) was proposed by Srinivas and Deb (1994) and is based on the concept of non-domination in a Pareto sense as explained previously.

The basic difference of NSGA in relation to a standard GA is the way in which the individuals are evaluated and selected. Basically, in order to obtain the fitness value of an individual, instead of using the fitness components associated with each objective involved in the problem, these components are used to rank the individuals according to their degree of domination over the others in the population; this measure of domination is used to define the fitness value that guides the action of the genetic operators and the selection process.

In order to work out that ranking, the population is organised in several layers of non-domination, the outermost layer representing the non-dominated individuals, and the innermost containing the most dominated ones. Initially, all individuals in the population that are non-dominated are separated as the first, outermost layer, and a (dummy) fitness value is assigned to them, whose role is simply to characterise their degree of domination over the others in the population. The highest fitness value is assigned to the outermost layer, as the individuals in it exhibit the highest degree of domination (the actual value assigned is proportional to the population size; this and other details are being omitted here for brevity). Then, the remaining individuals are classified again, also based on the non-domination criterion, and the second layer is formed with a (dummy) fitness lower than the first one. This process continues until all individuals are classified in their respective layers.

Once the non-domination layers were obtained, in order to maintain the diversity of the individuals inside each layer, a sharing function method is used to assign fitness to each individual (Goldberg, 1989). The basic idea of the sharing function is to reduce the fitness of the individuals that have many neighbours in the same layer. The fitness reduction is by a factor that depends on the number and proximity of neighbouring individuals, so that, the larger the number of individuals lying close to each other, the higher their fitness reduction; analogously, the more isolated an individual, the smallest the influence of the fitness reduction. As a consequence, the sharing function induces the formation of niches, thus increasing diversity in the population. The sharing function value between two individuals in the same border is given by:

$$Sh(d_{ij}) = \begin{cases} 1 - \left(\dfrac{d_{ij}}{\sigma_{share}} \right)^{\alpha}, & if \quad d_{ij} < \sigma_{share} \\ 0, & otherwise \end{cases} \qquad (1)$$

where, d_{ij} is the distance between the phenotypes of individuals i and j of the same front; σ_{share} is the maximum distance allowed to two individuals being considered members of the same niche; and α is the power that defines as the sharing degree decreases with the proximity.

A *niche counter* parameter is calculated through the addition of the value of the sharing function mentioned above for all individuals of the current front. The shared fitness values of each individual are calculated by the division of its dummy fitness by its *niche counter*.

After the non-domination classification and fitness assignment are performed, a proportional selection scheme is used so that the outer the layer an individual is in, the likelier its chance to reproduce. Stochastic remainder selection was the actual scheme used, preserved here because it was present in the original proposition of the NSGA. In this method, the next generation is formed by taking as many copies of every individual in the current population as the result of the integer division of its fitness value by the average fitness of the population. Subsequently, if the population needs be filled in, the fractional part of the latter division is used (Goldberg, 1989). The other steps of the NSGA are very similar to the simple GA (Goldberg, 1989).

Figure 2 shows a schematic flow chart of the NSGA.

Therefore, no elitist strategy is used in NSGA, in the sense of an external archive that would retain the best non-dominated solutions found along the evolutionary process, that would directly pass on to the next generation, regardless of how well fitted those individuals might

already be. This is in contrast with various more recent, elitist, multi-objective evolutionary methods (Coello *et al.*, 2002), such as NSGA-II (Deb *et al.*, 2000), SPEA (Zitzler & Thiele, 1999), SPEA2 (Zitzler *et al.*, 2001), PAES (Knowles & Cornes, 1999) and PESA (Knowles *et al.*, 2000).
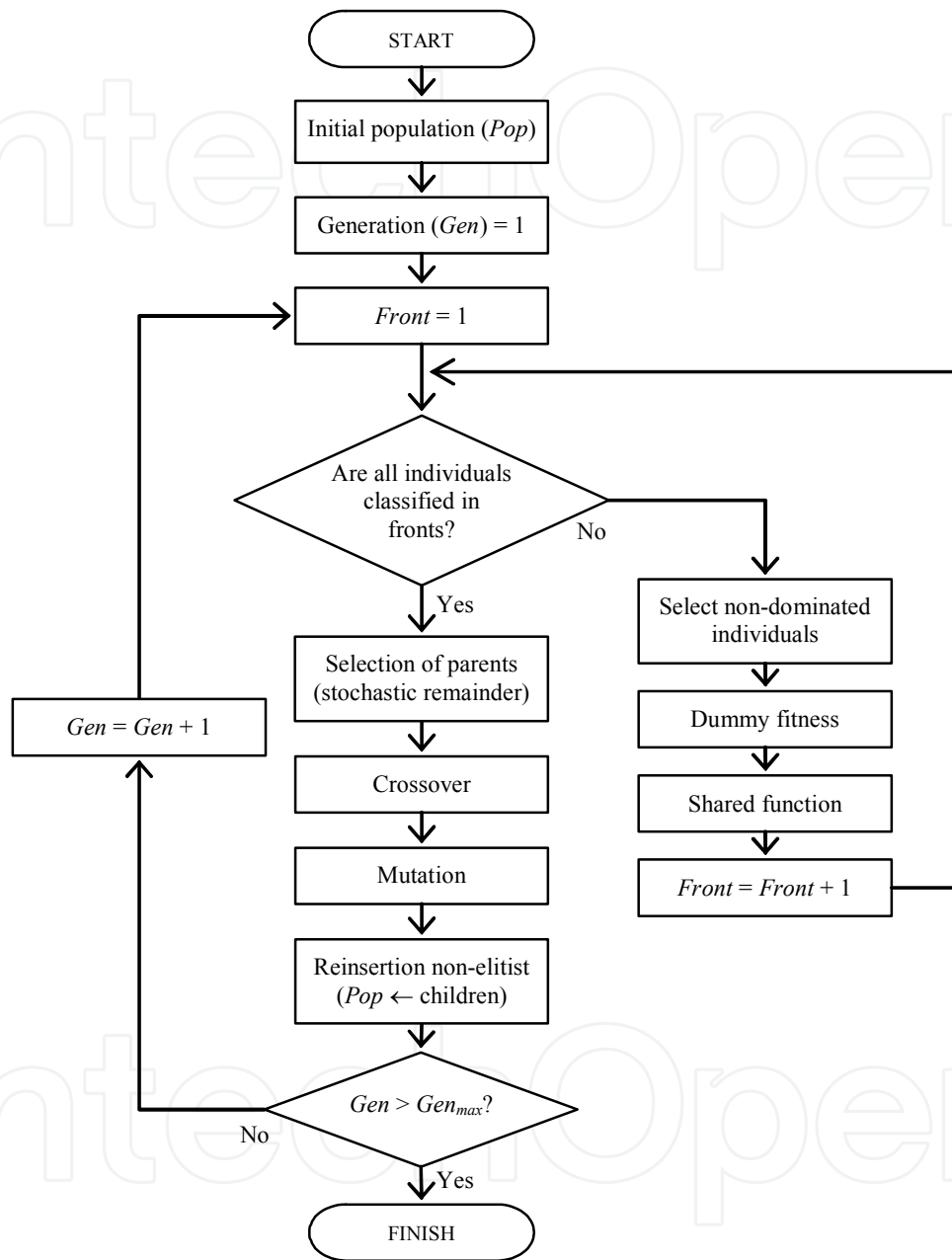

Fig. 2. NSGA's flow chart

3. Data mining

3.1 Knowledge discovery in databases

Data mining (DM) is the process of identification and extraction of useful information in typically large databases. DM aims to automatically discover the knowledge that is not easily perceivable. Data mining is considered the central step of the high-level process called

Knowledge Discovery in Databases (KDD). KDD includes includes the application of several preprocessing methods aimed at facilitating the application of the data mining algorithm and postprocessing methods aimed at refining and improving the discovered knowledge.

Data preprocessing aims to assure the data quality and to reduce the problem complexity, consolidating the relevant information. It commonly includes the following steps (Freitas, 2002):

i. data integration: involves, for instance, removing inconsistencies in attribute names or attribute value names between data sets of different sources.

ii. data cleaning: may involve detecting and correcting errors in the data, filling in missing values, etc.

iii. discretization: consists of transforming a continuous attribute into a categorical (or nominal) attribute, taking on only a few discrete values.

iv. attribute selection: consists of selecting a subset of attributes relevant for classification, among all original attributes.

Data postprocessing goal is to improve the understanding of the knowledge discovered, validating it through quality measures of the solution and the data analyst perception. It can be applied, for example, to simplify the discovered rule set by removing some rules and/or rule conditions, in order to improve knowledge comprehensibility for the user. Another goal of postprocessing can be to extract a subset of interesting rules, among all discovered ones. Methods for selection of interesting rules can be divided into subjective methods, user-driven and domain-dependent; and objective ones.

3.2 Data mining tasks

There are different types of tasks associated to data mining process. Each task can be thought of as a particular kind of problem to be solved by a data mining algorithm. The main types of tasks performed by DM algorithms are:

i. classification: prediction of the value (the class) of a user-specified goal attribute based on the values of other attributes, called the predicting attributes.

ii. association: discovery of association rules typically extracted from databases in which each record consists of a set of binary attributes called items. An association rule is a relationship of the form IF X THEN Y, where X and Y are sets of items and $X \cap Y = \varnothing$.

iii. clustering: creation of groups (clusters) determined by the data set, based on similarity measures or probabilistic models.

iv. Regression: association of an item to one or more prediction variables of real values.

v. sequences analysis: determination of the sequential characteristics, for example, in data with time dependence, which idea is the extraction and register of deviations and trends in the time.

vi. summarizing: description in compact form of a data subgroup.

vii. dependence modeling: prediction of the values of several user-specified goal attributes based on the values of other predicting attributes, being that a goal atribute X can be a predictive atribute of another goal atribute Y.

Dependence modelling is the task investigated in this work. The most popular DM tasks are classification and association. Classification is the most investigated one. It searches for the knowledge able to predict the value of a previously defined goal attribute based on other attributes (prediction attributes). This knowledge is often represented by IF-THEN rules.

Dependence modeling task can be seen as a generalization of the classification task. Dependence modeling task also aims to discover rules able to predict the goal attribute value, from values of the prediction attributes. However, in dependence modeling, there are more than one goal attribute. Initially, a small set of goal attributes whose prediction is considered interesting is specified. These attributes may occur in the consequent part of the rule (THEN) or in the antecedent part of the rule (IF). The others attributes occur only in the rule antecedent. As a consequence, dependence modeling task has a larger search space than the classification task, when one considers the same data set for both tasks.

Dependence modeling task is also different from the discovery of association rules. In association task, any attribute of the analyzed database can occur in the antecedent or in the consequent part of a rule. By other hand, only few pre-selected attributes – the goal attributes - can occur in the consequent part when the dependence modeling is considered. Also, in dependence modeling task, these same goal attributes can occurs in the antecedent part of a rule.

3.3 Evolutionary algorithms applied to data mining

Depending on the goal of the related mining process, a different class of DM algorithm is more appropriate. Some of the main techniques used in data mining are neural networks and decision trees. Evolutionary methods, especially the genetic algorithms (GA) (Goldberg, 1989), come also being investigated in several DM problems. Accordinly to Freitas (2002), the main motivation for using GAs in the discovery of high-level prediction rules is that they perform a global search and cope better with attribute interaction than the greedy rule induction algorithms often used in data mining.

The most relevant aspects in GA specification are the individual's encoding and the fitness function, which must represent the mined information and the quality of this information, respectively.

Two approaches have been used to represent IF-THEN rules as GA individuals: Michigan and Pittsburgh (Freitas, 2002). In the Michigan approach, each individual represents a single rule and the population as a whole represents a set of rules. In Pittsburgh approach, each individual represents a full set of rules. The kind of rule which is wanted to discover and the kind of data mining task addressed in the application can determine the choice between these two approaches. For example, in a classification task it is usually to evaluate the quality of the rule set as a whole, rather than the quality of a single rule. Therefore, the interaction among the rules is important indicating the Pittsburgh approach as more appropriated. The Michigan approach might be more adequate in other kinds of data mining problems as in a task where the goal is to find a small set of high-quality prediction rules, and each rule is often evaluated independently of other rules. Another aspect important to this choice is that using the Michigan approach, GA are more suitable for a good convergence as the individual encoding is simpler than in Pittsburgh's. In this work, the Michigan approach was adopted.

The evaluation of each individual is based on the metrics used to measure the quality of the rule codified. Several metrics can be used but usually only those related to the prediction accuracy of the rules are used, such as confidence factor, coverage, sensitivity and specification. For example, in (Fidelis *et al.,* 2000) the quality of the individual (rule) is measured by multiplying specificity and sensitivity. Freitas (2002) has pointed out the importance to include other metrics in the rules evaluation, such as, comprehensibility and

interestingness (Carvalho *et al.*, 2003). This raises the question of how to evaluate the trade-off between the accuracy of a model and other quality criteria. A typical example is a scenario where one wants to maximize both the accuracy and the simplicity of a classification model. This scenario configurates the data mining task as a multiobjective problem. The "conventional" approach to deal with this scenario is transforming the multi-objective problem into a single-objective one using a weighted formula. For example, a standard genetic algorithm was used by Noda and collaborators (1999) to search for rules evaluated by two metrics, one related to interestingness and another related to accuracy. Genetic algorithm aggregates theses two metrics into a single fitness, through of a pondered average.

Freitas (2004) presented a critical analysis about the three main multi-objective approaches to cope with data mining tasks:

i.   Transforming the original multi-objective problem into a single-objective one by using a weighted formula, as the single-objective GA previouly cited and described in (Noda *et al.*, 1999).

ii.   Manipulating simultaneous and independent objectives with the use of Pareto-based GA, as the approach investigated in this work and described in the next section.

iii.   The lexicographical approach, where the objectives are ranked in order of priority, as the work described in (Gonçalves *et al.*, 2005) and (Pila *et al.*, 2006).

Although the weighted formula approach is the most popular approach in the data mining literature, Freitas pointed out some drawbacks related to this approach. One of these drawbacks is that it mixes different non-commensurable model-quality criteria into the same formula. In particular, this has the disadvantage of producing model-quality measures that are not very meaningful to the user, going against the principle that in data mining discovered knowledge should be not only accurate but also comprehensible to the user. This and other drawbacks of the weighted-formula approach are avoided by the Pareto approach.

Pareto-based multi-objective GAs have been previously investigated in DM tasks, as the algorithms used in the search for nonlinear models of direct marketing in (Bhattacharyya, 2000) and to select characteristics for non-supervised learning in (Kim *et al.*, 2000). In (Iglesia *et al.*, 2003), an approach based on the NSGAII is employed in the classification task (only one goal attribute), using metrics related to the rules precision: accuracy and covering. A Pareto-based GA was implemented to mine rules from market-basket type databases (association task), using metrics for accuracy, comprehensibility and degree of interestingness (Ghosh & Nath, 2004). An approach based on the NSGAII was also applied to the association task, for the simultaneous maximization of covering and confidence metrics (Ishibuchi *et al.*, 2006a). Other related works that investigate multi-objective data mining are references (Francisci & Collard, 2003; Pappa *et al.*, 2004; Lane & Gobet, 2005; Ishibuchi *et al.*, 2006b; Berlanga *et al.*, 2006).

4. *Mo-miner*: a multi-objective data mining environment

A data mining environment was implemented to obtain interesting and accurate rules. It was named *MO-miner* (Multi-Objective-based miner). It employs a multi-objective GA inspired by NSGA and NSGAII methods that manipulates the metrics accuracy and degree of interestingness of the rules, which are kept as independent objectives. This environment is the major investigation of this work and it was implemented based on the model

described in (Noda *et al.*, 1999), called GA-nuggets. The major modification refers to the multi-objective approach, because GA-nuggets uses a single-objective GA and evaluates the metrics accuracy and degree of interestingness through a pondered average.

The individual's encoding represents one candidate rule (Michigan approach). The set of goal attributes must be predefined. The antecedent of the rule is formed by a conjunction of conditions of the form $A_i = V_{ij}$, where A_i is the *i-th* attribute and V_{ij} is the *j-th* value of this attribute's domain. The consequent consists on a simple condition of the form $G_k = V_{kl}$, where G_k is the *k-th* goal attribute and V_{kl} is the *l-th* value of its domain. Each individual represents the antecedent of a rule. The individual is encoded by a fixed size string, with *n* genes that represent the values of each one of the attributes, as illustrated in Figure 3. This encoding must represent a variable length rule. If an attribute does not occur in the rule antecedent, the value of its gene is "-1". For each individual, the GA automatically chooses the best goal attribute to put in the consequent ($G_k = V_{kl}$), that is, the values *k* and *l* in order to maximize rule's evaluation.

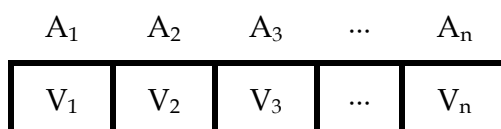| A_1	A_2	A_3	...	A_n
V_1	V_2	V_3	...	V_n

Fig. 3. Individual representation

The fitness function must reflect the goal of mining accurate and interesting rules. To do this, two metrics were used: *AntInt* and *PredAcc*. *AntInt* is the degree of interestingness of the antecedent part and *PredAcc* is the predictive accuracy of the rule. They are defined following.

The rule's predictive accuracy (*PredAcc*) is an adaptation of the sensitivity indicator, usually employed to measure the predictive ability of any rule:

$$PredAcc = \frac{(|A \& C| - 0.5)}{|A|} \tag{1}$$

where $|A\&C|$ is the number of registers in the database that satisfy both antecedent and consequent of the rule (true positives); and $|A|$ is the number of cases that satisfy only the rule antecedent (true positives plus false positives). The subtraction of 0.5 in the numerator was included to penalize rules covering few training examples. This factor is not used when the *PredAcc* is calculated in the test set, because it contains data unseen during training.

The degree of interestingness of the rule antecedent (*AntInt*) is calculated by an information-theoretical measure (Freitas, 1998) given by:

$$AntInt = 1 - \frac{\left(\sum_{i=1}^{n} InfoGain(A_i) \middle/ n\right)}{\log_2\left(|dom(G_k)|\right)} \tag{2}$$

where *n* is the number of attributes in the antecedent; $dom(G_k)$ is the number of possible values of the attribute G_k occurring in the consequent and $InfoGain(A_i)$ is the information gain associated to the attribute A_i, given by:

$$InfoGain(A_i) = Info(G_k) - Info(G_k|A_i) \qquad (3)$$

$$Info(G_k) = -\sum_{l=1}^{m_k}(\Pr(V_{kl})\log_2(\Pr(V_{kl}))) \qquad (4)$$

$$Info(G_k|A_i) = \sum_{j=1}^{n_i}(\Pr(V_{ij})(-\sum_{l=1}^{m_k}(\Pr(V_{kl}|V_{ij})\log_2(\Pr(V_{kl}|V_{ij}))))) \qquad (5)$$

where m_k is the number of possible values of the attribute G_k; n_i is the number of possible values of the prediction attribute A_i; $Pr(X)$ denotes the probability of X and $Pr(X|Y)$ denotes the conditional probability of X given Y.

The fitness calculation starts with two objectives, *AntInt* and *PredAcc*, which are kept separated, instead of working with an aggregation of these metrics. Subsequently, these objectives are used to perform the classification of non-dominated fronts, in the Pareto sense, in a similar way to that performed in NSGA. After the initial population of rules was generated, each individual is evaluated with respect to the two metrics by equations (1) and (2). These values are used to classify the individuals in hierarchical non-dominated fronts. Dummy fitness value is attributed to the individuals from each front. Later, these values are adjusted for each rule in a same front through the calculation of the sharing function and by the niche counter (Srinivas & Deb, 1994). After this, each individual finally receives a scalar fitness and the children generation is processed by four steps:

i.   Selection of crossover pairs,
ii. Uniform crossover with eventual repair of invalids chromosomes,
iii. Random mutation,
iv. Insertion and removal of genes.

These steps are detailed as follows.

A simple tournament selection is used, where a group (Tour) of individuals is randomly sorted and the one with the highest fitness is chosen for reproduction. It is important to note that a stochastic remainder selection was used in the original proposition of the NSGA. We adopted Tour = 3 in the tournament selection, in which each pair of individuals are selected from two groups of 3. The criterion used to define the tournament winner is: a pair <goal attribute, value> is drafted, the rules participants of the tournament are compared with respect to this pair and the one with the highest fitness is chosen.

After the selection, a uniform crossover operator was used with a probability of 70% for applying crossover to a pair of individuals and another probability of 50% for swapping each attribute's value in the rule antecedents of two individuals. After crossover is done, GA verifies if any invalid individual was created. If so, it uses a repair operator to produce valid individuals.

The mutation operator randomly transforms the value of an attribute into another value belonging to the domain of that attribute. The mutation rate used was 5%.

Besides crossover and mutation, there are the insertion and removal operators. These operators randomly insert or remove a condition in the antecedent. Their probabilities depend on the number of attributes in the antecedent. The insertion operator has a null probability of application when the rule antecedent has the maximum number of attributes (as specified by the user). The removal operator works in opposite way. The used insertion and removal rates are 50% and 70%, respectively.

The GA uses a kind of elitism (Goldberg, 1989) in the population reinsertion after each generation. Elitism strategy keeps the *M* best prediction rules for each pair <goal attribute, value> to the next generation, among *K* possible pairs, corresponding to an elitism factor of *K×M*. In the experiments described in section 6, the number of rules kept by elitism adopted was 3 rules for each pair.

It is important to note that the original NSGA is a non-elitist method. The fact that we centred the multi-objective approach herein around a non-elitist method, instead of using its more recent, elitist counterpart, NSGA-II (Deb *et al.*, 2000), derives from the problem being tackled. The point is that, the single-objective genetic algorithm presented in (Noda *et al.*, 1999) (GA-nuggets) – that has been used as the basis of our multi-objective GA – relies on a elitist strategy for reinsertion n of individuals in the population; hence, to maintain all the basic structure of that reference GA was a natural constraint, at the same time that using a non-elitist multi-objective algorithm seemed a sensible decision, so as not to overload the resulting algorithm as far as elitism is concerned. But now that the simpler, and to some extent naive multi-objective approach we adopted has shown to be successful, changing the actual multi-objective algorithm to a more sophisticated and well-established one, is clearly a natural step ahead.

A population size of 100 individuals and a number of 50 generations were used. Although *MO-miner* goal is to find a front with several non-dominated solutions, a criterion to select the best rule, in end of GA execution, must be established. We applied the same function adopted in (Noda *et al.*, 1999) as the single-objective fitness function and it consists of two parts. The first one measures the degree of interestingness of the rule as a whole (antecedent and consequent), while the second measures its predictive accuracy. Equation (6) was applied to decide which is the best rule in the highest non-dominated front:

$$Fitness = w_1 \times \frac{AntInt + ConsInt}{2} + w_2 \times PredAcc \qquad (6)$$

where *AntInt* is the degree of interestingness of the antecedent, *PredAcc* is the predictive accuracy of the rules; and w_1 and w_2 are user-defined weights and we used the same values adopted by Noda and collaborators (1999), $w_1 = 1/3$ and $w_2 = 2/3$, that gives a greater weight to the rule accuracy.

ConsInt is the degree of interestingness of the consequent part (Noda *et al.*, 1999) and it is calculated by:

$$ConsInt = \sqrt{1 - \Pr(V_{kl})} \qquad (7)$$

where $Pr(V_{kl})$ is the relative frequency of the goal attribute value V_{kl}.

Accordingly to Freitas (2004), one of the arguments against Pareto approaches to data mining algorithms is the difficulty of choosing a single "best" solution to be used in practice. A possible criticism of such kind of approach is that the data mining algorithm returns a set of non-dominated solutions, whereas in practice the user will often use a single solution. Therefore, a difficult problem associated with the Pareto approach is to decide how can one choose the "best" non-dominated solution, out of all non-dominated solutions in the final generation. Choosing the best solution based on Equation (6), we define GA's final output in an objective way and we avoid the problem cited above. Besides, the rules are selected with the same final criterion used in the single-objective environment implemented in (Noda *et al.*, 1999) what turns the results obtained by the two approaches suitable for a comparative analysis.

The metric *ConsInt*, given by equation (7), was not employed as an independent objective of our multi-objective approach; because it returns a constant value that depends only on the consequent pair <*goal attribute*, *value*> at issue and on the number of occurrences of this pair in the database. However, it is used in the final evaluation of the best individual to weight the degree of interestingness of the rule in the same way as done in (Noda *et al.*, 1999). Figure 4 presents a schematic flow chart of *MO-miner* environment.
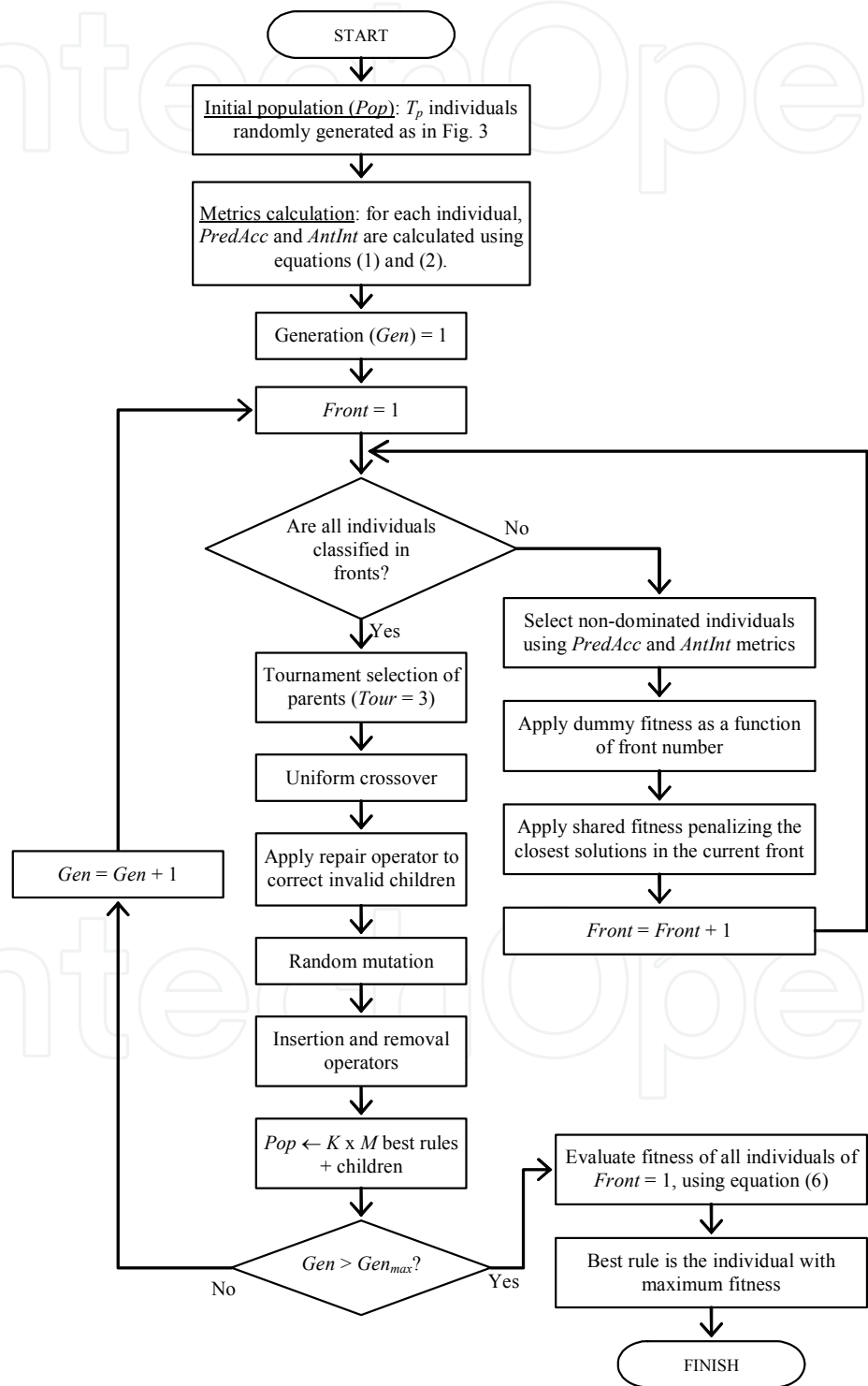

Fig. 4. *MO-miner*'s flow chart

5. Data sets

We used two data sets to evaluate *MO-miner*: *Nursery* and *Zoo*. These data sets were also explored in (Noda *et al.*, 1999). Both databases are public data sets and they were obtained from the UCI repository (www.ics.uci.edu/~mlearn/MLRepository.html) of machine learning databases.

Nursery database was obtained in http://archive.ics.uci.edu/ml/datasets/Nursery, available in this repository since 1997. It was derived from a hierarchical decision model originally developed to rank applications for nursery schools. It was used during several years in 1980's when there was excessive enrollment to these schools in Slovenia, and the rejected applications frequently needed an objective explanation. The final decision depended on three subproblems: occupation of parents and child's nursery, family structure and financial standing, and social and health picture of the family. The database contains 12960 registers and nine categorical attributes: parents, health, form, children, has_nurs, housing, finance, social and recommendation. Three attributes was used as goal: finance, social and recommendation. Finance has two possible values (convenient and inconv), social has three possible values (non-prob, slightly-prob and problematic) and recommendation has five values in its domain (not-recom, recommended, very-recom, priority, spec-prior). It results in rules with ten different consequents:
- Consequent 1: finance = convenient
- Consequent 2: finance = inconv
- Consequent 3: social=non-prob
- Consequent 4: social = slightly-prob
- Consequent 5: social =problematic
- Consequent 6: recommendation = not-recom
- Consequent 7: recommendation = recommended
- Consequent 8: re-commendation = very-recom
- Consequent 9: recommendation = priority
- Consequent 10: recommendation = spec-prior

Zoo is the second public database used and it was also obtained from the UCI repository (http://archive.ics.uci.edu/ml/datasets/Zoo), and it is available in this repository since 1990. Each instance of this database corresponds to an animal. It contains only 101 registers and a total of eighteen categorical attributes: hair, feathers, eggs, milk, backbone, fins, legs, tail, catsize, airbone, aquatic, breathes, venomous, toothed, predator, domestic and type. Three goal attributes was chosen: predator, domestic and type. Predator and domestic are binary attributes and type has seven possible numerical values (from 1 to 7). Therefore, there are eleven different consequents:
- Consequent 1: predator=0
- Consequent 2: predator=1
- Consequent 3: domestic=0
- Consequent 4: domestic=1
- Consequent 5: type=1
- Consequent 6: type=2
- Consequent 7: type=3
- Consequent 8: type=4
- Consequent 9: type=5

- Consequent 10: type=6
- Consequent 11: type=7

6. Experiments

6.1 Single-objective environment

Besides *MO-miner* detailed in Section 4, a second environment was implemented to supply comparative results to evaluate the effectiveness of the multi-objective approach. It was implemented strongly based on the model described in (Noda *et al.*, 1999), called GA-nuggets. It uses a single-objective standard genetic algorrithm with the same goal of *MO-miner*: to discover accurate and interesting rules.

Each individual is evaluated through the pondered average described by Equation (6). The components *PredAcc, AntInt* and *ConsInt* of Equation (6) was calculated in the same way as described to *MO-miner* and they are given by equations (1), (2) and (7), respectively. The user-defined weights w_1 and w_2 in Equation (6) was also adopted as in (Noda *et al.*, 1999): $w_1 = 1/3$ and $w_2 = 2/3$.

The single-objective GA also uses a population size of 100 individuals and a number of 50 generations per execution. After the initial population of rules was generated, each individual is evaluated with respect to Equation (6). The children generation is processed as in the multi-objective environment:

i.   Simple tournament selection to crossover (*Tour* = 3).
ii.   Uniform crossover with eventual repair of invalids chromosomes.
iii.   Random mutation.
iv.   Insertion and removal of the genes.

Some small modifications had been made in the single-objective GA with respect to the one described in (Noda *et al.,* 1999):

i.   We adopted *Tour* = 3 in the tournament selection.
ii.   The criterion used to define the tournament winner is: a pair <*goal attribute, value*> is drafted, the rules participants of the tournament are compared with respect to this pair and the one with the highest fitness is chosen.
iii.   The number of rules kept by elitism is 3 rules for each pair.

6.2 Experiments description

Two kinds of experiments had been executed with the single-objective GA and with *MO-miner*. The first one used the full data set during GA evolution, getting the better rules to predict each one of the possible consequents $G_k = V_{kl}$. In the second experiment, a cross-validation was used to evaluate the quality of the discovered rules. Hence, the data set is divided into five mutually exclusive and exhaustive partitions. This second kind of experiment is composed by five types of GA evolution, each one using a different partition as test set. All partitions, except the test set, had been used during the rules evolution as training set, corresponding to the data effectively used by the genetic algorithm during rule search. Later, the rules are evaluated concerning their prediction accuracy in the test partition. The final result is defined by the average results considering the five test partitions.

6.3 Results using *Nursery* database

The results of this section was previously discussed in (Oliveira *et al.,* 2008).

6.3.1 Full set experiment

Both single-objective GA and *MO-miner* had been applied to 100% of the registers from *Nursery* database. In each experiment, 10 different GA executions had been performed. Table 1 presents *AntInt* and *PredAcc* values for the best rule obtained in these executions, for each consequent. Figure 5 shows radial charts of these metrics to aid the comparative analysis: (a) *AntInt* and (b) *PredAcc*. In general, both environments had mined rules with a high degree of interestingness and a good accuracy for the majority of the consequents. The single-objective GA found rules with degree of interestingness below of 0.95 in consequents 6, 8, 9 and 10 and *MO-miner* only in consequents 8 and 10. In all the others consequents, *AntInt* exceeds 0.97. Both the single-objective and *MO-miner* had found bad rules (≤ 0.5) for consequents 3, 4 and 7 with respect to accuracy. These results are compatible with those reported in (Noda *et al.*, 1999).

In a comparative analysis, one can observe that the most significant change occurs with respect to the *AntInt*, while the metric *PredAcc* remains practically unchanged for both environments. Concerning degree of interestingness, *MO-miner* returned rules better than single-objective GA in seven consequents, with an increase above to 10% in two of them (consequents 6 and 9). With respect to accuracy, the multi-objective returns rules a little bit higher than the single-objective environment in three consequents (consequents 6, 9 and 10) and equivalent in the others.

We conclude that our multi-objective approach found rules with higher degree of interestingness, without damage on the prediction accuracy.

Table 2 presents the best rules discovered by *MO-miner*. Their quality of prediction is better estimated by the average of *PredAcc* obtained in cross-validation experiments, described as follows.

Consequent	Single-Objective		MO-miner	
	AntInt	*PredAcc*	*AntInt*	*PredAcc*
1	0.999	0.987	0.999	0.987
2	0.999	0.992	0.999	0.992
3	0.993	0.498	0.999	0.498
4	0.993	0.498	0.999	0.498
5	0.996	0.994	0.999	0.994
6	0.891	0.998	0.998	0.999
7	0.978	0.500	0.998	0.500
8	0.921	0.969	0.921	0.969
9	0.907	0.996	0.998	0.998
10	0.908	0.997	0.909	0.999

Table 1. *Nursery* Database: *AntInt* and *PredAcc* obtained with single-objective GA and *MO-miner* in full set experiment.

6.3.2 Cross-validation experiment

The cross-validation method was used to verify the generalization capability of the mined rules. *Nursery* database was divided into five partitions and five different GA evolutions had been performed. In each kind of evolution, a different partition was used as the test set

and the others four partitions are merged and they were used in the evolutionary search for the rules (training). Ten different GA executions had been performed for each test partition. After the best rules for each consequent were obtained, they had been applied to the test partition to evaluate its prediction over examples not seem during the evolution.

Table 3 presents *PredAcc* average for the best rules found for each environment, both in training and in test. Figure 6 shows radial charts of the metric *PredAcc* (a) in training and (b) in test.
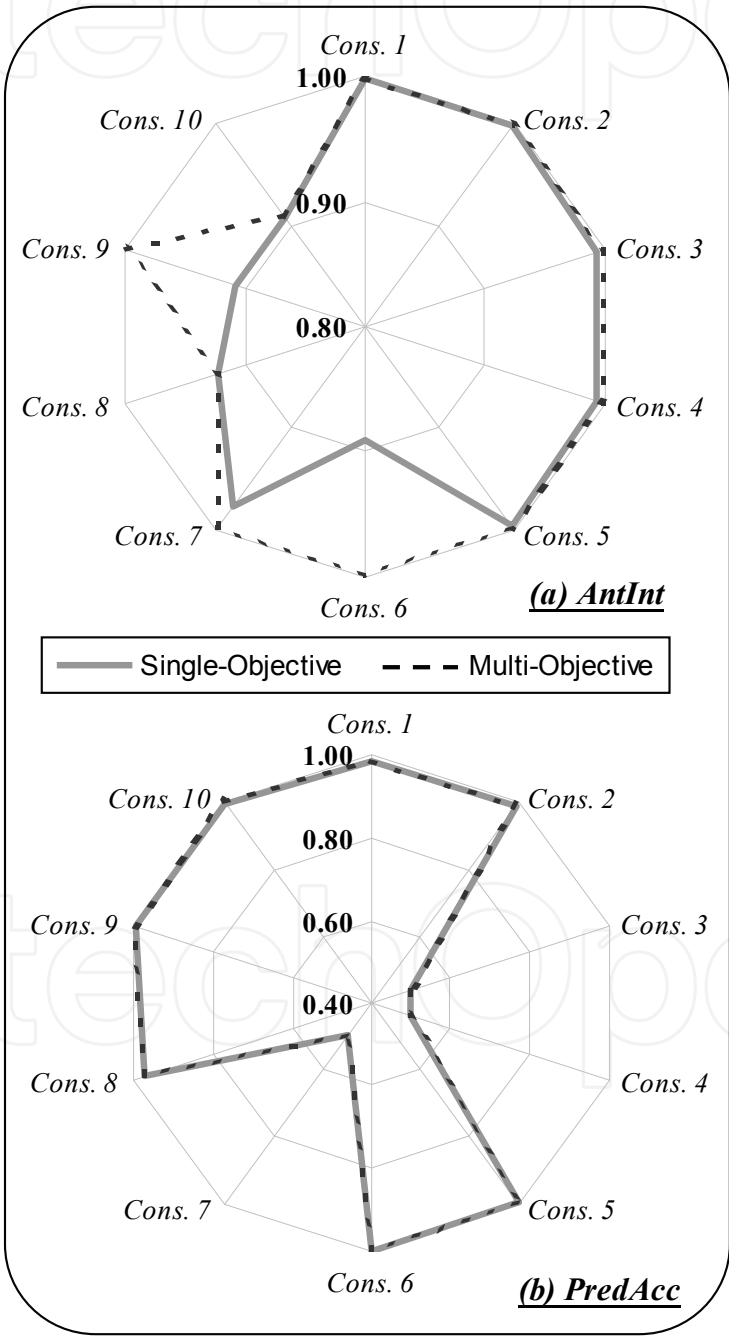
Fig. 5. Single-objective GA and MO-miner results in the full *Nursery* database: (a) *AntInt* (b) *PredAcc*.

IF (*health = recommended*) AND (*form = foster*) AND (*housing = convenient*) AND (*recommendation = very_recom*) THEN (*finance = convenient*)
IF (*housing = convenient*) AND (*social = nonprob*) AND (*health = recommended*) AND (*recommendation = spec_prior*) THEN (*finance = inconv*)
IF (*health = recommended*) AND (*recommendation = very_recom*) THEN (*social = non_prob*)
IF (*health = recommended*) AND (*recommendation = very_recom*) THEN (*social = slightly_prob*)
IF (*housing = convenient*) AND (*finance = convenient*) AND (*health = recommended*) AND (*recommendation = spec_prior*) THEN (*social = problematic*)
IF (*has_nurs = proper*) AND (*finance = convenient*) AND (*social = nonprob*) AND (*health = not_recom*) THEN (*recommendation = not_recom*)
IF (*parents=usual*) AND (*has_nurs=proper*) AND (*finance = convenient*) AND (*children=1*) AND (*form = complete*) AND (*housing = convenient*) AND (*social = nonprob*) AND (*health = recommended*) THEN (*recommendation = recommended*)
IF (*parents = usual*) AND (*has_nurs = less_proper*) AND (*housing = convenient*) AND (*finance = convenient*) AND (*social = slightly_prob*) AND (*health = recommended*) THEN (*recommendation = very_recom*)
IF (*has_nurs = proper*) AND (*housing = convenient*) AND (*finance = convenient*) AND (*health = priority*) THEN (*recommendation = priority*)
IF (*has_nurs = very_crit*) AND (*finance = inconvenient*) AND (*social = problematic*) AND (*health = recommended*) THEN (*recommendation = spec_prior*)

Table 2. Rules extracted by *MO-miner* from *Nursery* database.

Consequent	Single-Objective		MO-miner	
	Training	Test	Training	Test
1	0.984	0.400	0,985	0,600
2	0.992	0.700	0,991	0,700
3	0.499	0.263	0,499	0,333
4	0.498	0.200	0,498	0,367
5	0.994	0.686	0,993	0,733
6	0.997	0.800	0,999	1,000
7	0.500	0.007	0,500	0,007
8	0.968	0.000	0,972	0.000
9	0.993	0.644	0,999	0,450
10	0.995	0.590	0,998	0,590

Table 3. *Nursery Database: PredAcc* of training and test obtained with single-objective GA and *MO-miner* in cross-validation experiment.

In both environments, the average of *PredAcc* in the training set is close to the values obtained with the full set. However, the average of *PredAcc* in the test set falls substantially, indicating that the rules mined in training do not exhibits good generalization ability. When comparing results of the two environments one can notice that even so the performance has been very similar in the training sets, *MO-miner* returned higher *PredAcc* rules in the test sets. *PredAcc* average measured in test sets for *MO-miner* had been higher in five consequents (1, 3, 4, 5 and 6) and lower in only one (consequent 9). Thus, it is possible to conclude that the rules mined by *MO-miner* present better generalization ability than those obtained by simple-objective GA.
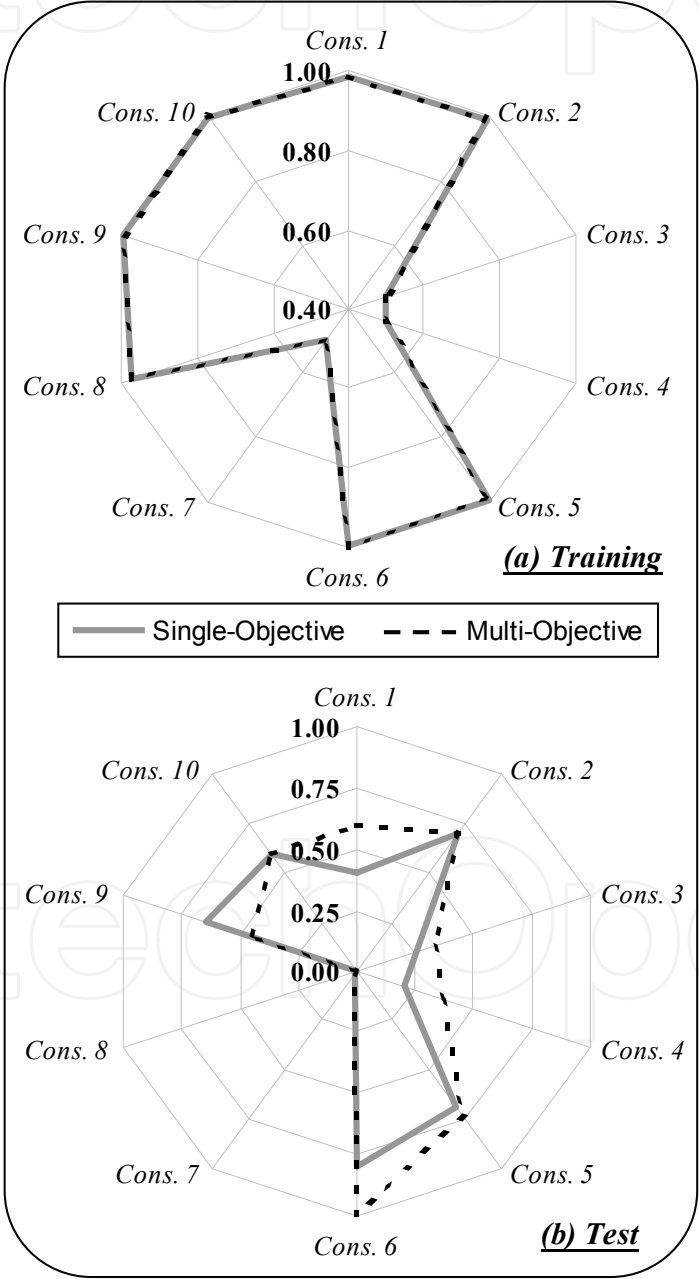
Fig. 6. *PredAcc* average values of the single-objective GA and MO-miner environments in cross-validation experiment using *Nursery* database: (a) training (b) test.

6.4 Results using *Zoo* database
6.4.1 Full set experiment

Table 4 shows the values of *AntInt* and *PredAcc* for the best rules found using the full Zoo database. Both environments (single-objective and *MO-miner*) were able to mine rules with high degree of interestingness and a good accuracy for almost all the consequents; the unique exception was the consequent 4. The single-objective GA found rules with *AntInt* below of 0.9 in consequents 5 to 11 while *MO-miner* only in consequents 7, 8 and 9 (but in all these case *AntInt* was at least 0.84). In the other cases, *AntInt* is above 0.95 for both environments. In respect to accuracy, single-objective and *MO-miner* had found bad rules (< 0.8) only for consequent 4. Again, these results are compatible with (Noda *et al.,* 1999). As observed in Nursery database, the most significant difference occurs with respect to *AntInt*, while *PredAcc* remains practically equivalent for both environments.

Figure 7 shows the radial charts to aid the comparative analysis: (a) *AntInt* and (b) *PredAcc*. Concerning *AntInt*, *MO-miner* returned rules better than single-objective GA in seven consequents, with an increase above to 10% in four of them (5, 6, 10 and 11). With respect to *PredAcc*, *MO-miner* returns rules a little bit higher in four consequents (3, 5, 6 and 8), a lit bit lower in one (consequent 7), and equivalent in the others. As observed in Nursery experiments, the major conclusion is that the multi-objective approach found rules with higher degree of interestingness, without damage on the prediction accuracy.

Table 5 presents the best rules discovered by *MO-miner* using the full Zoo database. Cross-validation experiments described in the next section can better estimate their quality of prediction.

Consequent	Single-Objective		MO-miner	
	AntInt	PredAcc	AntInt	PredAcc
1	0.956	0.929	0.999	0.929
2	0.983	0.958	0.999	0.958
3	0.990	0.980	0.950	0.986
4	0.996	0.500	0.996	0.500
5	0.880	0.984	0.982	0.988
6	0.893	0.971	0.982	0.975
7	0.849	0.875	0.881	0.833
8	0.887	0.958	0.840	0.962
9	0.883	0.875	0.863	0.875
10	0.848	0.937	0.982	0.937
11	0.887	0.937	0.982	0.937

Table 4. *Zoo* Database: *AntInt* and *PredAcc* obtained with single-objective GA and *MO-miner* in full set experiment.

6.4.2 Cross-validation experiment

As performed for the *Nursery* database, the *Zoo* database was divided into five partitions and five different GA evolutions had been performed. For each test partition, ten different GA executions had been performed. The best rules obtained for each consequent had been applied to the test partition to evaluate its prediction.

Table 6 presents *PredAcc* average for the best rules found (over the 5 partitions), both in training and in test. Figure 8 shows radial charts of the metric *PredAcc* (a) in training and (b) in test. The average of *PredAcc* in the training set is close to the values obtained with the full set, as for the single-objective GA as for *MO-miner*. But as observed in *Nursery* experiments, the average of *PredAcc* in the test set falls substantially, indicating that the rules mined in training do not exhibits good generalization ability, for both environments. Once more *MO-miner* returned higher *PredAcc* rules in the test sets even so the performance has been very similar in the training sets. *PredAcc* average measured in test sets for *MO-miner* had been higher in six consequents being this performance significatively in three of them. It is possible to conclude that, as concluded in *Nursery* experiments, the rules mined by *MO-miner* present better generalization ability.
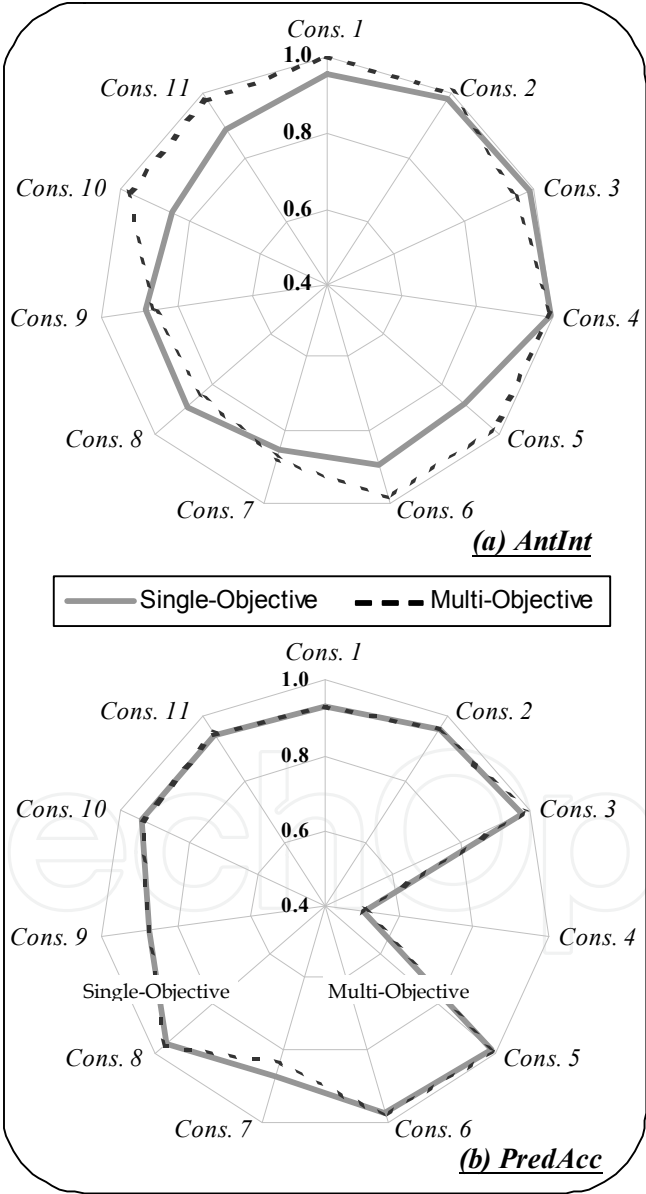
Fig. 7. Single-objective GA and MO-miner results in the full *Zoo* database: (a) *AntInt* (b) *PredAcc.*

IF (*domestic*=1) AND (*catsize*=0) THEN (*predator*=0)
IF (*airbone*=0) AND (*aquatic*=1) AND (*catsize*=1) THEN (*predator*=1)
IF (*hair*=0) AND (*predator*=1) THEN (*domestic*=0)
IF (*eggs*=0) AND (*venomous*=0) AND (*tail*=0) AND (*catsize*=0) THEN (*domestic*=1)
IF (*milk*=1) AND (*venomous*=0) THEN (*type*=1)
IF (*feathers*=1) AND (*venomous*=0) THEN (*type*=2)
IF (*hair*=0) AND (*aquatic*=0) AND (*predator*=1) AND (*toothed*=1) AND (*domestic*=0) THEN (*type*=3)
IF (*eggs*=1) AND (*fins*=1) AND (*domestic*=0) THEN (*type*=4)
IF (*feathers*=0) AND (*airbone*=0) AND (*aquatic*=1) AND (*breathes*=1) AND (*catsize*=0) THEN (*type*=5)
IF(*aquatic*=0) AND (*fins*=0) AND (*legs*=6) AND (*catsize*=0) THEN(*type*=6)
IF (*airbone*=0) AND (*predator*=1) AND (*backbone*=0) AND (*domestic*=0) THEN (*type*=7)

Table 5. Rules extracted by *MO-miner* from *Zoo* database.

Consequent	Single-Objective		MO-miner	
	Training	Test	Training	Test
1	0.919	0.200	0.922	0.433
2	0.947	0.900	0.948	0.933
3	0.979	0.975	0.983	0.978
4	0.471	0.050	0.500	0.080
5	0.980	1.000	0.985	1.000
6	0.964	1.000	0.970	1.000
7	0.842	0.400	0.842	0.430
8	0.948	0.800	0.950	0.800
9	0.833	0.600	0.858	0.600
10	0.915	0.500	0.922	0.600
11	0.917	1.000	0.920	1.000

Table 6. *Zoo Database: PredAcc* of training and test obtained with single-objective GA and *MO-miner* in cross-validation experiment.
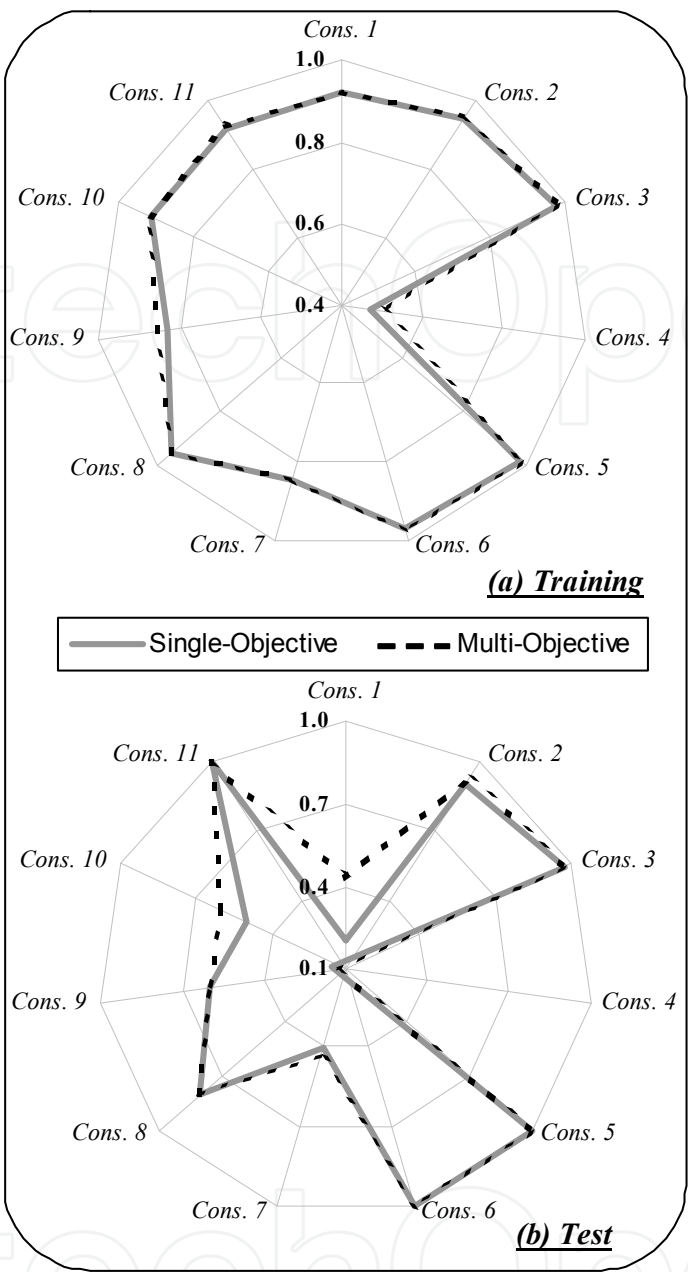
Fig. 8. *PredAcc* average values of the single-objective GA and MO-miner environments in cross-validation experiment using *Zoo* database: (a) training (b) test.

7. Conclusion

The application of a multi-objective GA-based environment named *MO-miner*, inspired by the family of algorithms NSGA and designed to discover accurate and interesting dependence modeling rules, was investigated in this work. Our investigation was performed by applying *MO-miner* to two distinct public data sets named *Nursery* and *Zoo*, composed by 12960 and 101 registers, respectively.

The accuracy and the interestingness of the rules mined by *MO-miner* were compared with those found by a single-objective standard genetic algorithm, based in its turn on the

environment implemented by Noda and collaborators (1999) named *GA-nuggets*. The results obtained with our single-objective GA, both for full database and for crossvalidation experiments, are very similar to those reported in (Noda *et al.*, 1999), in which the same data sets were mined.

Comparing the performance of the both environments, it was possible to observe that, when using the full database, they had found rules very similar with respect to accuracy. However, *MO-miner* returned more interesting rules. We believe that this result is due to the kind of rule evaluation performed by *MO-miner*, which kept metrics independents, and to *MO-miner*'s dynamics that drives the search toward the region of the highest values in both metrics. Conversely, in single-objective environment, the balance between the metrics was pre-defined, giving a higher weight to the accuracy with respect to the degree of interestingness.

In respect to the cross-validation experiments, it was evidenced that the results found by *MO-miner* were better, resulting in a lesser accuracy decay between training and test. This indicates that the rules obtained by *MO-miner* have greater generalization than those extracted using the single-objective environment.

Despite the fact that *Zoo* and *Nursery* data sets are very different among each other, specially because the first is a small database and the second is a large one, in the comparative experiments applying the two environments, we arrived at similar conclusions. Using both data sets, *MO-miner* found more interesting rules in the full set experiment and rules with better generalization ability in the cross-validation experiment.

Different paths related to multi-objective genetic algorithms and data mining can be explored as continuity of this study, such as: inclusion of other metrics as comprehensibility (Freitas, 2002); use of others multi-objective techniques (Coello *et al.*, 2002); use of other fitness functions, among others.

8. Acknowledgements

9. References

Berlanga, F.; del Jesus, M. J.; Gonzalez, P.; Herrera, F. & Mesonero, M. (2006) Multiobjective evolutionary induction of subgroup discovery fuzzy rules: A case study in marketing, *Lecture Notes in Artificial Intelligence (Advances in Data Mining)*, Vol. 4065, pp. 337-349, 2006, Springer-Verlag.

Bhattacharyya, S. (2000). Evolutionary algorithms in data mining: multi-objective performance modelling for direct marketing, *Proceedings of ACM SIGKDD International Conference KDD-2000*, pp. 465-473, 2000.

Carvalho, D. R.; Freitas, A. A. & Ebecken, N.F.F. (2003). A critical review of rule surprisingness measures, *Proceedings of Data Mining IV - Int. Conf. on Data Mining*, pp. 545-556, 2003.

Coello, C. A. C. (1999). A comprehensive survey of evolutionary-based multiobjective optimization techniques, *Knowledge and Information Systems*, Vol. 1, No. 3, 1999, pp. 269-308.

Coello,C. A. C.; Van Veldhuizen, D. A. & Lamont, G. B. (2002). Evolutionary algorithms for solving multi-objective problems, 2002, Kluwer Academic Publishers, New York.

Deb, K.; Agrawal,S.; Pratab, A. & Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pp. 849-858, 2000, Springer, Berlin.

Fidelis, M. V.; Lopes, H. S. & Freitas, A. A. (2000). Discovering comprehensible classification rules with a genetic algorithm, *Proceedings of Congress on Evolutionary Computation (CEC-2000)*, pp. 805-810, 2000, La Jolla, CA, USA.

Fonseca, C. M. & Fleming, P. J. (1993). Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. *Proceedings of the Fifth International Conference on Genetic Algorithms*, Forrest, S. (ed.), pp. 416-423, 1993, San Mateo, California.

Francisci, D. & Collard, M. (2003). Multi-Criteria Evaluation of Interesting Dependencies according to a Data Mining Approach, *Proceedings of Congress on Evolutionary Computation (CEC'2003)*, Vol. 3, pp. 1568--1574, Dec - 2003, IEEE Press, Canberra, Australia.

Freitas, A. A. (1998). On objective measures of rule surprisingness. *Principles of Data Mining & Knowledge Discovery (Proc. 2nd European Symp., PKDD'98). LNAI 1510*, pp. 1-9, 1998, Springer-Verlag.

Freitas, A. A. (2002). A survey of evolutionary algorithms for data mining and knowledge discovery, In: *Advances in Evolutionary Computation*, Ghosh and Tsutsui (ed.), 2002, Springer-Verlag.

Freitas, A. A. (2004). A critical review of multi-objective optimization in data mining: a position paper, *ACM SIGKDD Explorations Newsletter*, Vol. 6, 2004, pp. 2.

Ghosh, A. & Nath, B. (2004). Multi-objective rule mining using genetic algorithms. *Information Sciences*, Vol. 163, No. 1-3, pp. 123-133, 2004.

Goldberg, D. (1989). Genetic Algorithms in Search, *Optimization and Machine Learning*, Addison-Wesley.

Gonçalves, A. S.; Freitas, A. A.; Kato, R. E. & de Oliveira, R. (2005). Using genetic algorithms to mine interesting dependence modeling rules, *Proceedings of 23rd IASTED Int. Multi-Conference on Databases and Applications (DBA-2005)*, 2005.

Horn, J. and Nafpliotis, N. (1993). Multiobjective optimization using the Niched Pareto Genetic Algorithm. *IlliGA1 Technical Report 93005*, University of Illinois at Urbana-Champaign, 1993, Urbana, Illinois.

Iglesia, B.; Philpott, M.; Bagnall, A. & Smith, V. (2003). Data mining rules using multi-objective evolutionary algorithms, *Proceedings of Congress on Evolutionary Computation*, pp. 1552-1559, 2003.

Ishibuchi, H.; Kuwajima, I. & Nojima, Y. (2006a). Multiobjective association rule mining. *Proceedings of PPSN Workshop on Multiobjective Problem Solving from Nature*, 12 pages (Reykjavik, Iceland), 2006.

Ishibuchi, H.; Nojima, Y. & Kuwajima, I. (2006b). Multiobjective Genetic Rule Selection as a Data Mining Postprocessing Procedure, *Proceedings of Genetic and Evolutionary Computation Conference (GECCO'2006)*, Maarten Keijzer *et al.* (ed.), pp. 1591-1592, Vol. 2, 2006, ISBN 1-59593-186-4, ACM Press, Seattle, Washington, USA.

Kim, Y.; Street, W. N. & Menczer, F. (2000). Feature selection in unsupervised learning via evolutionary search, *Proceedings of ACM SIGKDD International Conference KDD-2000*, pp. 365-369, 2000.

Knowles, J. and Corne, D. (1999). The Pareto archived evolution strategy: a new baseline algorithm for multiobjective optimisation, *Proceedings of Congress on Evolutionary Computation*, Angeline, P. J. *et al.* (ed.), pages 98-105, 1999, IEEE Press, Washington, D.C.

Knowles, J.; Corne, D. & Oates, M. (2000). The Pareto-envelope based selection algorithm for multiobjective optimization, *Proceedings of the Sixth International Conference on Parallel Problem Solving from Nature (PPSN VI)*, Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J. J. and Schwefel, H.-P. (ed.), pp. 839-848, 2000, Springer, Berlin.

Lane, P. C. R. & Gobet, F. (2005). Discovering predictive variables when evolving cognitive models, Pattern Recognition and Data Mining, *Lecture Notes in Computer Science*, Vol. 3686, pp. 108--117, 2005, Springer.

Mitchell, M. (1997). An Introduction to Genetic Algorithms, *MIT Press*, Massachusetts.

Noda, E.; Freitas, A. A. & Lopes, H. S. (1999). Discovering Interesting Prediction Rules with a Genetic Algorithm, *Proceedings of Congress on Evolutionary Computation*, pp. 1322-1329, 1999.

Oliveira, G. M. B.; Takiguti, M. C. S. & Martins, L. G. A. (2008). Dependence Modeling Rule Mining using Multi-Objective Generic Algorithms. *Proceedings of 26th IASTED International Conference ARTIFICIAL INTELLIGENCE AND APPLICATIONS (AIA08)*, pp. 278-283, Feb - 2008, Innsbruck.

Pappa, G. L.; Freitas, A. A. & Kaestner, C. A. A. (2004). Multi-Objective Algorithms for Attribute Selection in Data Mining, in *Applications of Multi-Objective Evolutionary Algorithms*, Carlos A. C. Coello and Gary B. Lamont (ed.), pp. 603-626, 2004, World Scientific, Singapore.

Pila, A.; Giusti, R.; Prati, R. & Monard, M. (2006). Multi-Objective Evolutionary Algorithm to Build Knowledge Classification Rules with Specific Properties, *Proceedings of 6th International Conference on Hybrid Inteliggent Systems*, 2006, IEEE Computer Society, NY.

Shaffer, J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. In *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetics Algorithms*, pp. 93-100, 1985.

Srinivas, N. & Deb, K. (1994). Multiobjective Optimization using nondominated sorting in genetic algorithms, *Evolutionary Computation*, Vol. 2, No. 3, 1994, pp. 221.

Zitzler, E. & Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 4, 1999, pp. 257-271.

Zitzler, E.; Laumanns, M. & Thiele, L. (2001). SPEA2: Improving the {S}trength {P}areto {E}volutionary {A}lgorithm. *EUROGEN 2001: Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou and T. Fogarty (ed.), pp. 12-21, Sep - 2001, Greece.

Advances in Robotics, Automation and Control

Edited by Jesus Aramburo and Antonio Ramirez Trevino

The book presents an excellent overview of the recent developments in the different areas of Robotics, Automation and Control. Through its 24 chapters, this book presents topics related to control and robot design; it also introduces new mathematical tools and techniques devoted to improve the system modeling and control. An important point is the use of rational agents and heuristic techniques to cope with the computational complexity required for controlling complex systems. Through this book, we also find navigation and vision algorithms, automatic handwritten comprehension and speech recognition systems that will be included in the next generation of productive systems developed by man.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Gina M. B. de Oliveira, Luiz G. A. Martins and Maria C., S. Takiguti (2008). MO-Miner: A Data Mining Tool Based on Multi-Objective Genetic Algorithms, Advances in Robotics, Automation and Control, Jesus Aramburo and Antonio Ramirez Trevino (Ed.), ISBN: 978-953-7619-16-9, InTech, Available from: http://www.intechopen.com/books/advances_in_robotics_automation_and_control/mo-miner__a_data_mining_tool_based_on_multi-objective_genetic_algorithms

INTECH
open science | open minds