CHAPTER 25

MULTI-OBJECTIVE ALGORITHMS FOR ATTRIBUTE SELECTION IN DATA MINING

Gisele L. Pappa and Alex A. Freitas

Computing Laboratory, University of Kent, Canterbury
CT2 7NF, UK , E- mail: {glp6,A.A.Freitas}@kent.ac.uk
http://www.cs.kent.ac.uk/people/staff/aaf

Celso A. A. Kaestner

Graduate Program in Applied Computer Science
Pontificia Universidade Catolica do Parana (PUCPR)
Rua Imaculada Conceicao, 1155
80215-901 Curitiba - PR - Brazil
E-mail: kaestner@ppgia.pucpr.br

Attribute selection is an important preprocessing task for the application of a classification algorithm to a given data set. This task often involves the simultaneous optimization of two or more objectives. In order to solve this problem, this chapter describes two multi-objective methods: a genetic algorithm and a forward sequential feature selection method. Both methods are based on the wrapper approach for attribute selection and were used to find the best subset of attributes that minimizes the classification error rate and the size of decision tree built by a well-known classification algorithm, namely C4.5.

1. Introduction

Attribute selection is one of the most important preprocessing tasks to be performed before the application of data mining techniques. In essence, it consists of selecting a subset of attributes relevant for the target data mining task, out of all original attributes. In this work the target task is classification, where the goal is to predict the class of an example (record) given the values of the attributes describing that example. Attribute selection became essential when researches discovered it can improve the data mining algorithm's performance (with respect to learning speed, classification rate and/or rule set simplicity) and at the same time remove noise and decrease data dimensionality.

In face of the importance of attribute selection, a variety of methods have been used in order to find a small attribute subset capable of obtaining a better classification rate than that obtained with the entire attribute set. These methods include sequential search[1], ranking techniques[2] and evolutionary algorithms[3].

Independent of the method used to solve the problem of attribute selection, solving this problem often requires the minimization of at least two objectives: the classification error rate and a measure of size — which can be a measure of size of the selected data (typically the number of selected attributes) and/or a measure of size of the classifier (say, a rule set) learned from the selected data. Many attribute selection methods optimize these objectives setting weights to each one and combining them in a single function.

However, the study of multi-objective optimization has shown that, in some tasks, a weighted combination of the objectives to be optimized in a single function is not the most effective approach to solve the problem. Mainly in tasks that deal with optimization of conflicting objectives, such as attribute selection, the use of the Pareto's dominance concept during optimization can be the best choice.

The optimization based on the Pareto's concept[4] suggests that, for each of the conflicting objectives to be optimized, exists an optimal solution. So, the final response of the optimization system is a set of optimal solutions instead of a single solution. This is in contrast with systems that intend to optimize a single objective. Hence, it is left to the user to decide which of the optimal solutions he/she considers the best to solve his/her problem, using his/her background knowledge about the problem.

In this spirit, this work presents two multi-objective attribute selection algorithms based on the Pareto's dominance concept. One of them is a multi-objective genetic algorithm, and the other one is a multi-objective version of the well-known forward sequential feature selection method. Both methods use the wrapper approach (see next section) in order to minimize the error rate and the size of the decision tree built by a well-known classifier, namely C4.5.

We report the results of extensive computational experiments with 18 public domain real-world data sets, comparing the performance of these two methods. The results show that both methods effectively select good attribute subsets — by comparison with the original set of all attributes —

and, somewhat surprisingly, the multi-objective forward sequential selection method is competitive with the multi-objective genetic algorithm.

2. Attribute Selection

As mentioned earlier, attribute selection is an important step in the knowledge discovery process and aims to select a subset of attributes that are relevant for a target data mining task. In the classification task, which is the task addressed in this work, an attribute is considered relevant if it is useful for discriminating examples belonging to different classes.

We can find in the literature a lot of attribute selection methods. These methods differ mainly in the search strategy they use to explore the space of candidate attribute subsets and in the way they measure the quality of a candidate attribute subset.

With respect to the search strategy, the methods can be classified as exponential (e.g. exhaustive search), randomized (e.g. genetic algorithms) and sequential. The exponential methods are usually too computationally expensive, and so are not further discussed here.

The sequential methods include the well-known FSS (forward sequential selection) and BSS (backward sequential selection)[5]. FSS starts with an empty set of attributes (features) and iteratively selects one-attribute-at-a-time — the attribute considered most relevant for classification at the current step — until classification accuracy cannot be improved by selecting another attribute. BSS starts with the full set of original attributes and iteratively removes one-attribute-at-a-time — the attribute considered least relevant for classification at the current step — as long as classification accuracy is not decreased. We have developed a multi-objective version of the FSS method, which will be described later.

With respect to randomized methods, in this chapter we are particularly interested in genetic algorithms, due to their ability to perform a global search in the solution space. In our case, this means that they tend to cope better with attribute interaction than greedy, local-search methods (such as sequential methods)[3]. We have also developed a multi-objective genetic algorithm (GA) for attribute selection, which will be described later.

The evaluation of the quality of each candidate attribute subset can be based on two approaches: the filter or the wrapper approach. The main difference between them is that in the wrapper approach the evaluation function uses the target classification algorithm to evaluate the quality of a candidate attribute subset. This is not the case in the filter approach, where the evaluation function is specified in a generic way, regardless of the classification algorithm. That is, in the wrapper approach the quality of a candidate attribute subset depends on the performance of the classification algorithm trained only with the selected attributes. This performance can be measured with respect to several factors, such as classification accuracy and size of the classifier learned from the selected data. Indeed, these are the two performance measures used in this work, as will be seen later.

Although the wrapper approach tends to be more expensive than the filter approach, the wrapper approach usually obtains better predictive accuracy that the filter approach, since it finds an attribute subset “customized” for the target classification algorithm.

The vast majority of GAs for attribute selection follow the wrapper approach. Table 1, adapted from Freitas[3], shows the criteria used in the fitness function of a number of GAs for attribute selection following the wrapper approach.

As can be observed in Table 1, there are many criteria that can be used in the fitness of a GA for attribute selection, but all the GAs mentioned in the table use classification accuracy, and many GAs use either the number of selected attributes or the size of the classifier learned from the data. Note that only one of the GAs mentioned in Table 1 is a multi-objective method — all the other GAs either try to optimize a single objective (predictive accuracy) or use some method (typically a weighted formula) to combine two or more objectives into a single objective to be optimized.

3. Multi-objective Optimization

Real world problems are usually complex and require the optimization of many objectives to reach a good solution. Unfortunately, many projects that should involve the simultaneous optimization of multiple objectives avoid the complexities of such optimization, and adopt the simpler approach of just weighing and combining the objectives into a single function. This simpler approach is not very effective in many cases, due to at least two reasons. First, the objectives are often conflicting with each other. Second, the objectives often represent different and non-commensurate aspects of a candidate solution’s quality, so that mixing them into a single formula is not semantically meaningful. Indeed, both reasons hold in our case, where the two objectives to be minimized — classification error rate and decision-tree size are to some extent conflicting and entirely non-commensurate.

According to the multi-objective optimization concept, when many ob-

Table 1. Main aspects of fitness functions of GAs for attribute selection

Reference	Criteria used in fitness function
[Bala et al. 1995][6]	predictive accuracy, number of selected attributes
[Bala et al. 1996][7]	predictive accuracy, information content, number of selected attributes
[Chen et al. 1999][8]	based first on predictive accuracy, and then on number of selected attributes
[Guerra-Salcedo & Whitley 1998][9]	predictive accuracy
[Guerra-Salcedo et al. 1999][10]	predictive accuracy
[Cherkauer & Shavlik 1996][11]	predictive accuracy, number of selected attributes, decision-tree size
[Terano & Ishino 1998][12]	subjective evaluation, predictive accuracy, rule set size
[Vafaie & DeJong 1998][13]	predictive accuracy
[Yang & Honavar 1997, 1998][14,15]	predictive accuracy, attribute cost
[Moser & Murty 2000][16]	predictive accuracy, number of selected attributes
[Ishibuchi & Nakashima 2000][17]	predictive accuracy, number of selected instances, number of selected attributes (attribute and instance selection)
[Emmanouilidis et al. 2000][18]	predictive accuracy, number of selected attributes (multi-objective evaluation)
[Rozsypal & Kubat 2003][19]	predictive accuracy, number of selected instances, number of selected attributes (attribute and instance selection)
[Llòra & Garrell 2003][20]	predictive accuracy

jectives are simultaneously optimized, there is no single optimal solution. Rather, there is a set of optimal solutions, each one considering a certain trade-off among the objectives[21]. In this way, a system developed to solve this kind of problem returns a set of optimal solutions, and can be left to the user to choose the one that best solves his/her specific problem. This means that the user has the opportunity of choosing the solution that represents the best trade-off among the conflicting objectives after examining several high-quality solutions. Intuitively, this is better than forcing the user to define a single trade-off before the search is performed, which is what happens when the multi-objective problem is transformed in a single-objective one. The Pareto's multi-objective optimization concept is used to find this set of optimal solutions. According to this concept, a solution S_1 dominates a solution S_2 if and only if[4]:

- Solution S_1 is not worse than solution S_2 in any of the objectives;
- Solution S_1 is strictly better than solution S_2 in at least one of the objectives.

Figure 1 shows an example of possible solutions found for a multi-objective attribute selection problem. The solutions that are not dominated by any other solutions are considered Pareto-optimal solutions, and they are represented by the dotted line in Figure 1.

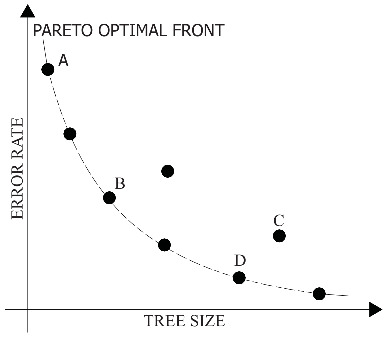

Fig. 1. Example of Pareto dominance in a two-objective problem

Note that Solution A has a small decision-tree size but a large error rate. Solution D has a large decision-tree size but a small error rate. Assuming that minimizing both objectives is important, one cannot say that solution A is better than D, nor vice-versa. On the other hand, solution C is clearly not a good solution, since it is dominated, for instance, by D.

4. The Proposed Multi-Objective Methods for Attribute Selection

In the last few years, the use of multi-objective optimization has led to improved solutions for many different kinds of problems[21]. So, in order to evaluate the effectiveness of the multi-objective framework in the attribute selection problem for the classification task, we proposed a multi-objective genetic algorithm[22] (MOGA) that returns a set of non-dominated solutions. We also proposed a multi-objective version of the forward sequential selection (FSS) method[23].

The goal of these proposed algorithms is to find a subset of relevant attributes that leads to a reduction in both classification error rate and complexity (size) of the decision tree built by a data mining algorithm.

The classification algorithm used in this paper is C4.5[25], a well-known decision tree induction algorithm. The proposed methods are based in the wrapper approach, which means they use the target data mining algorithm (C4.5) to evaluate the quality of the candidate attribute subsets. Hence, the methods' evaluation functions are based on the error rate and on the size of the decision tree built by C4.5. These two criteria (objectives) are to be minimized according to the concept of Pareto dominance.

The next subsections present the main aspects of the proposed methods. The reader is referred to Pappa[22,23] for further details.

4.1. *The Multi-Objective Genetic Algorithm (MOGA)*

A genetic algorithm (GA) is a search algorithm inspired by the principle of natural selection. It works evolving a population of individuals, where each individual is a candidate solution to a given problem. Each individual is evaluated by a fitness function, which measures the quality of its corresponding solution. At each generation (iteration) the fittest (the best) individuals of the current population survive and produce offspring resembling them, so that the population gradually contains fitter and fitter individuals — i.e., better and better candidate solutions to the underlying problem. For a comprehensive review of GAs in general the reader is referred to Michalewicz[24]. For a comprehensive review of GAs applied to data mining the reader is referred to Freitas[3].

The motivation for developing a multi-objective GA for attribute selection was that: (a) GAs are a robust search method, capable of effectively exploring the large search spaces often associated with attribute selection problems; (b) GAs perform a global search, so that they tend to cope better with attribute interaction than greedy search methods, which is also an important advantage in attribute selection; and (c) GAs already work with a population of candidate solutions, which makes them naturally suitable for multiobjective problem solving[4], where the search algorithm is required to consider a set of optimal solutions at each iteration.

4.1.1. *Individual Encoding*

In the proposed GA, each individual represents a candidate subset of selected attributes, out of all original attributes. Each individual consists of M genes, where M is the number of original attributes in the data being mined. Each gene can take on the value 1 or 0, indicating that the corresponding attribute occurs or not (respectively) in the candidate subset of selected attributes.

4.1.2. *Fitness Function*

The fitness (evaluation) function measures the quality of a candidate attribute subset represented by an individual. Following the principle of multi-objective optimization, the fitness of an individual consists of two quality measures: (a) the error rate of C4.5; and (b) the size of the decision tree built by C4.5. Both (a) and (b) are computed by running C4.5 with the individual's attribute subset only, and by using a hold-out method to estimate C4.5's error rate, as follows. First, the training data is partitioned into two mutually-exclusive data subsets, the building subset and the validation subset. Then we run C4.5 using as its training set only the examples (records) in the building subset. Once the decision tree has been built, it is used to classify examples in the validation set.

4.1.3. *Selection Methods and Genetic Operators*

At each generation (iteration) of the GA, the next population of individuals is formed as follows. First the GA selects all the non-dominated individuals of the current generation, which are then passed unaltered to the next generation by elitism[26]. Elitism is a common procedure in MOGAs. It avoids that non-dominated individuals disappear from the population due to the stochastic nature of selection operators. However, a maximum number of elitist individuals has to be fixed to avoid that the next population consist only of elitist individuals, which would prevent the creation of new individuals, stopping the evolutionary process. This maximum number of elitist individuals was set to half the population size. If the number of non-dominated individuals is larger than half the population size, that number of elitist individuals is chosen by the tie-breaking criterion explained later.

Once elitist reproduction has been performed, the remainder of the next generation's population is filled in with new "children" individuals, generated from "parent" individuals from the current generation. The parent individuals are chosen by tournament selection with a tournament size of 2. Then children are generated from parents by applying conventional uniform crossover and bit-flip mutation. The tournament selection procedure is adapted for multi-objective search as follows.

The fitness of an individual is a vector with values for two objectives:

the error rate and decision-tree size associated with the attribute subset represented by the individual. The selection of the best individual is based on the concept of Pareto dominance, taking into account the two objectives to be minimized. Given two individuals I_1 and I_2 playing a tournament, there are two possible situations. The first one is that one of the individuals dominates the other. In this case the former is selected as the winner of the tournament.

The second situation is that none of the individuals dominates the other. In this case, we use the following tie-breaking criterion to determine the fittest individual. For each of the two individuals I_i, i=1,2, the GA computes X_i as the number of individuals in the current population that are dominated by I_i, and Y_i as the number of individuals in the current population that dominate I_i. Then the GA selects as the best the individual I_i with the largest value of the formula: X_i - Y_i. Finally, if I_1 and I_2 have the same value of the formula X_i - Y_i (which is rarely the case), the tournament winner is simply chosen at random.

In all our experiments the probabilities of crossover and mutation were set to 80% and 1%, respectively, which are relatively common values in the literature. The population size was set to 100 individuals, which evolve for 50 generations. These values were used in all our experiments.

4.2. *The Multi-Objective Forward Sequential Selection Method (MOFSS)*

A single-objective optimization and a multi-objective optimization method differ mainly in the number of optimal solutions that they return. Hence, the first step to convert the traditional FSS into a multi-objective method is to make it able to return a set of optimal solutions instead of a single solution.

This first point was resolved by creating a list of all non-dominated solutions generated by the MOFSS until the current iteration of the algorithm. This concept of a external list of non-dominated solutions was inspired by some MOGAs in literature such as SPEA[27], that maintain all the non-dominated individuals in an external population.

The proposed MOFSS starts as the traditional FSS: a subset of solutions is created and evaluated. The evaluation of each solution considers both the error rate and the decision tree size generated by C4.5 during training. As in the proposed MOGA, the values of these objectives to be minimized are stored and later used to judge a solution as better or worse than other.

Each new solution of the current iteration is compared with every other solution of the current iteration, in order to find all non-dominated solutions in the current iteration. Then the non-dominated solution list, L, is updated. This update consists in comparing, through the Pareto's dominance concept, the solutions in the list with the non-dominated solutions of the current iteration. More precisely, for each non-dominated solution S of the current iteration, S will be added to the list L only if S is not dominated by any solution in L. It is also possible that S dominates some solution(s) in L. In this case those dominated solutions in L are, of course, removed from L.

The non-dominated solution list is the start point for generating new candidate solutions. At each iteration, each solution in the current list is extended with each new attribute (different from the ones that occur in the current solution), and the process starts again, until no more updates can be made in the non-dominated solution list.

5. Computational Results

Experiments were executed with 18 public-domain, real-world data sets obtained from the UCI (University of California at Irvine)'s data set repository[28]. The number of examples, attributes and classes of these data sets is shown in Table 2.

All the experiments were performed with a well-known stratified 10-fold cross-validation procedure. For each iteration of the cross-validation procedure, once the MOGA/MOFSS run is over we compare the performance of C4.5 using all the original attributes (the "baseline" solution) with the performance of C4.5 using only the attributes selected by the MOGA/MOFSS. Recall that the MOGA/MOFSS can be considered successful to the extent that the attributes subsets selected by it lead to a reduction in the error rate and size of the tree built by C4.5, by comparison with the use of all original attributes.

As explained before, the solution for a multi-objective optimization problem consists of all non-dominated solutions (the Pareto front) found. Hence, each run of the MOGA outputs the set of all non-dominated solutions (attribute subsets) present in the last generation's population and each run of the MOFSS outputs the solutions stored in the non-dominated solution list in the last iteration. In a real-world application, it would be left to the user the final choice of the non-dominated solution to be used in practice. However, in our research-oriented work, involving many differ-

Table 2. Main characteristics of the data sets used in the experiments

Data Set	# examples	# attributes	# classes
Arrhythmia	269	452	16
Balance-Scale	4	625	3
Bupa	6	345	2
Car	6	1717	4
Crx	15	690	2
Dermatology	34	366	6
Glass	10	214	7
Ionosphere	34	351	2
Iris	4	150	3
Mushroom	22	8124	2
Pima	8	768	2
Promoters	57	106	2
Sick-euthyroid	25	3163	2
Tic tac toe	9	958	2
Vehicle	18	846	4
Votes	16	435	2
Wine	13	178	3
Wisconsin breast-cancer	9	699	2

ent public-domain data sets, no user was available. Hence, we needed to evaluate the quality of the non-dominated attribute subsets returned by MOGA/MOFSS in an automatic, data-driven manner. We have done that in two different ways, reflecting two different (but both valid) perspectives, as follows.

The first approach to evaluate the set of non-dominated solutions returned by MOGA and MOFSS is called *Return All Non-Dominated Solutions*. The basic idea is that we return all the non-dominated solutions found by the method, and we compare each of them, one-at-a-time, with the baseline solution — which consists of the set of all original attributes. Then we count the number of solutions returned by the MOGA and MOFSS that dominate or are dominated by the baseline solution, in the Pareto sense — with respect to the objectives of minimizing error rate and decision-tree size, as explained above.

The second approach, called *Return the "Best" Non-Dominated Solution* consists of selecting a single solution to be returned to the user by using the tie-breaking criterion described earlier. From a user's point of view, this is a practical approach, since the user often wants a single solution. Moreover, this decision making process makes the solution of the multi-objective problem complete, following its 3 potential stages of development:

measurement, search and decision making[29].

There are many ways of setting preferences in a decision making process, as shown in Coello-Coello[29], but we did not follow any of those approaches. For both MOGA and MOFSS we return the solution in the non-dominated set of the last generation (or iteration) with the highest value of the tie-breaking criterion - which is a decision-making criterion tailored for our algorithms and underlying application. Note that, once the number of solutions that dominates the solutions in the non-dominated set is zero, the formula of the tie-breaking criterion is reduced to X_i. Therefore, instead of explicit ranking the objectives, we rank the non-dominated solutions according the number of individuals they dominate in the last generation. The solution chosen through this method was compared with the baseline solution.

There is one caveat when using this criterion in MOFSS. For this algorithm, we recalculate the tie-breaking criterion considering all the solutions generated in all the iterations of the method. That is, we calculate the number of solutions that are dominated by each of the solutions in the non-dominated solution list of the last iteration, considering all solutions generated by the method. The tie-braking criterion was recalculated because, for some data sets, the number of solutions in the non-dominated list at the beginning of the last iteration was small. As a result, few new solutions were generated in the last iteration. It was not fair to compare the solutions in that list just with those few solutions generated in the last generation, because the small number of solutions would lead to a low confidence (from a statistical point of view) in the result. In order to solve this problem, the tie-breaking criterion is recalculated using all generated solutions since the algorithm starts. There was no need to apply this procedure to MOGA, because this method has a larger number of solutions in the last iteration, providing enough solutions for a reliable computation of the tie-breaking criterion.

5.1. *Results for the "Return All Non-Dominated Solutions" Approach*

As explained earlier, the basic idea of this approach is that MOGA and MOFSS return all non-dominated solutions that they have found, and then we count the number of solutions returned by each of these methods that dominate or are dominated by the baseline solution.

Tables 3 and 4 show, respectively, the results found by MOGA and

MOFSS returning all the non-dominated solutions of the last generation (or iteration). Hereafter this version of the algorithms is called MOGA-all and MOFSS-all. In Tables 3 and 4 the second column shows the total number of solutions found by the method. The numbers after the "±" are standard deviations. The next columns show the relative frequency of the found solutions that dominate the baseline solution (column $F_{dominate}$), the relative frequency of the found solutions that are dominated by the baseline solution (column $F_{dominated}$) and the relative frequency of the found solutions that neither dominate nor are dominated by the baseline solution (column $F_{neutral}$).

Table 3. Results found with MOGA-all

	Solutions found with MOGA-all			
Data set	Total	$F_{dominate}$	$F_{dominated}$	$F_{neutral}$
Arrhythmia	3.9 ± 0.54	0.21	**0.33**	0.46
Balance-Scale	1.0 ± 0.0	**0.7**	0	0.3
Bupa	6.1 ± 0.38	0.31	0	0.69
Car	38.3 ± 0.76	0.002	0	0.998
Crx	4.55 ± 0.67	**0.56**	0.05	0.39
Dermatology	1.11 ± 0.11	**0.8**	0	0.2
Glass	46.9 ± 1.03	0	**0.06**	0.94
Ionosphere	1.14 ± 0.14	0.37	0.12	0.5
Iris	4.4 ± 0.16	**0.8**	0.02	0.18
Mushroom	1.9 ± 0.18	**0.68**	0	0.32
Pima	18.3 ± 1.15	0.34	0	0.66
Promoters	1.5 ± 0.16	0.33	0	0.67
Sick- euthyroid	25.4 ± 0.93	0.02	0.02	0.96
Tic tac toe	16.5 ± 1.0	0	0	1
Vehicle	6.1 ± 0.76	0.25	0.18	0.57
Votes	26.6 ± 1.63	**0.6**	0	0.4
Wine	4.66 ± 1.21	0.48	0.31	0.21
Wisconsin	9.3 ± 0.4	0.5	0.2	0.3

As can be observed in Table 3, there are 6 data sets where the value of $F_{dominate}$ is greater than 0.5 (shown in bold), which means that more than 50% of the MOGA-all's solutions dominated the baseline solution. In 9 out of the 18 data sets, no MOGA-all's solution was dominated by the baseline solution. There are only two data sets, namely arrhythmia and glass, where the value of $F_{dominate}$ is smaller than the value of $F_{dominated}$ (shown in bold), indicating that the MOGA was not successful in these two data sets. In any case, in these two data sets the difference between $F_{dominate}$ and $F_{dominated}$ is relatively small (which is particularly true in the case of glass), and the value of $F_{neutral}$ is greater than the values of both $F_{dominate}$ and $F_{dominated}$.

In summary, in 14 out of the 18 data sets the value of $F_{dominate}$ is greater than the value of $F_{dominated}$, indicating that overall MOGA-all was successful in the majority of the data sets. MOGA-all was very successful in 6 data sets, where the value of $F_{dominate}$ was larger than 0.5 and much greater than the value of $F_{dominated}$.

In Table 4, we can see that there are 7 data sets where the value of $F_{dominate}$ is greater than 0.5 (shown in bold), which means that 50% or more of the MOFSS-all's solutions dominated the baseline solution. Remarkably, there are only two data sets — namely wine and Wisconsin breast cancer — where the number of MOFSS-all's solutions dominated by the baseline solution was greater than zero, and in the case of wine that number is very close to zero, anyway. There are two data sets where all MOFSS-all's solutions are neutral, namely dermatology and mushroom. In summary, in 16 out of the 18 data sets the value of $F_{dominate}$ is greater than the value of $F_{dominated}$, indicating that overall MOFSS was successful in the vast majority of the data sets. MOFSS was very successful in 7 data sets, as mentioned above.

Table 4. Results found with MOFFS-all

	Solutions found with MOFFS-all			
Data set	Total	$F_{dominate}$	$F_{dominated}$	$F_{neutral}$
Arrhythmia	32.2 ± 10.82	**0.54**	0	0.46
Balance-Scale	1.8 ± 0.2	0.5	0	0.5
Bupa	2.9 ± 0.31	**0.65**	0	0.35
Car	4.3 ± 0.33	0.07	0	0.93
Crx	84.1 ± 2.05	**0.89**	0	0.11
Dermatology	76.5 ± 10.3	0	0	1
Glass	94.1 ± 5.24	**0.99**	0	0.01
Ionosphere	12.9 ± 6.23	0.14	0	0.86
Iris	3.5 ± 0.34	**0.86**	0	0.14
Mushroom	51.9 ± 11.88	0	0	1
Pima	11.1 ± 1.88	**0.95**	0	0.05
Promoters	66.6 ± 12.66	0.27	0	0.73
Sick-euthyroid	50.3 ± 6.44	0.1	0	0.9
Tic tac toe	8.1 ± 1.54	0.11	0	0.89
Vehicle	3.6 ± 0.16	0.17	0	0.83
Votes	98.4 ± 0.37	0.1	0	0.9
Wine	8.3 ± 6.1	**0.92**	0.01	0.07
Wisconsin	10.1 ± 4.76	0.45	0.37	0.18

5.2. *Results for the "Return the 'Best' Non-Dominated Solution" Approach*

Tables 5 and 6 show the results obtained by following this approach. These tables show results for error rate and tree size separately, as usual in the machine learning and data mining literature. Later in this section we show results (in Table 7) involving Pareto dominance, which consider the simultaneous minimization of error rate and tree size. In Tables 5 and 6 the column titled C4.5 contains the results for C4.5 ran with the baseline solution (all original attributes), whereas the columns titled MOGA-1 and MOFSS-1 contain the results for C4.5 ran with the single "best" non-dominated solution found by MOGA and MOFSS, using the criterion for choosing the "best" solution explained earlier. The figures in the tables are the average over the 10 iterations of the cross-validation procedure. The values after the "±" symbol represent the standard deviations, and the figures in bold indicate the smallest error rates/tree sizes obtained among the three methods. In the columns MOGA-1 and MOFSS-1, the symbol "+" ("-") denotes that the results (error rate or tree size) of the corresponding method is significantly better (worse) than the result obtained with the baseline solution. The difference in error rate or tree size between the columns MOGA-1/MOFSS-1 and C4.5 are considered significant if the corresponding error rate or tree size intervals — taking into account the standard deviations — do not overlap. The last two lines of Tables 5 and 6 summarize the results of these tables, indicating in how many data sets MOGA-1/MOFSS-1 obtained a significant win/loss over the baseline solution using C4.5 with all original attributes.

In Tables 5 and 6, the results of MOFSS-1 for the dataset Arrhythmia are not available due to the large number of attributes in this data set, 269. This leads to a too large number of solutions generated along all iterations of the algorithm, so that re-calculating the tie-breaking criterion considering all the generated solutions was impractical with the machine used in the experiments (a dual-PC with 1.1GHz clock rate and 3Gbytes memory).

The results in Table 5 show that MOGA-1 obtained significantly better error rates than the baseline solution (column "C4.5") in 8 data sets. In contrast, the baseline solution obtained significantly better results than MOGA-1 in just two data sets. MOFSS-1 has not found solutions with significantly better error rates than the baseline solution in any data set. On the contrary, it found solutions with significantly worse error rates than the baseline solution in 7 data sets.

Table 5. Error rates obtained with C4.5, MOGA-1 and MOFSS-1

	Error Rate (%)		
Data set	C4.5	MOGA-1	MOFSS-1
Arrhythmia	32.93 ± 3.11	26.38 ± 1.47 (+)	N/A
Balance-Scale	36.34 ± 1.08	**28.32 ± 0.71 (+)**	36.47 ± 1.84
Bupa	37.07 ± 2.99	**30.14 ± 1.85 (+)**	40.85 ± 1.45
Car	**7.49 ± 0.70**	16.65 ± 0.4 (-)	18.5 ± 0.70 (-)
Crx	15.95 ± 1.43	12.44 ± 1.84	15.04 ± 1.35
Dermatology	6.0 ± 0.98	**2.19 ± 0.36 (+)**	11.15 ± 1.60 (-)
Glass	1.86 ± 0.76	1.43 ± 0.73	1.86 ± 0.76
Ionosphere	10.2 ± 1.25	**5.13 ± 1.27 (+)**	7.98 ± 1.37
Iris	6.0 ± 2.32	**2.68 ± 1.1 (+)**	6.01 ± 2.09
Mushroom	0.0 ± 0.0	0.0 ± 0.0	0.18 ± 0.07 (-)
Pima	26.07 ± 1.03	23.07 ± 1.16	28.16 ± 1.72
Promoters	16.83 ± 2.55	**11.33 ± 1.92 (+)**	33.5 ± 6.49 (-)
Sick-euthyroid	2.02 ± 0.12	2.22 ± 0.18	2.32 ± 0.23
Tic tac toe	**15.75 ± 1.4**	22.65 ± 1.19 (-)	31.19 ± 1.69 (-)
Vehicle	26.03 ± 1.78	23.16 ± 1.29	33.74 ± 1.78 (-)
Votes	3.2 ± 0.91	2.97 ± 0.75	4.57 ± 0.89
Wine	6.69 ± 1.82	**0.56 ± 0.56 (+)**	6.07 ± 1.69
Wisconsin	5.28 ± 0.95	3.84 ± 0.67	7.16 ± 0.77 (-)
Wins over C4.5	-	8	0
Losses over C4.5	-	2	7

As can be observed in Table 6, the tree sizes obtained with the solutions found by MOGA-1 and MOFSS-1 are significantly better than the ones obtained with the baseline solution in 15 out of 18 data sets. In the other three data sets the difference is not significant.

In summary, both MOGA-1 and MOFSS-1 are very successful in finding solutions that led to a significant reduction in tree size, by comparison with the baseline solution of all attributes. The solutions found by MOGA-1 were also quite successful in reducing error rate, unlike the solutions found by MOFSS-1, which unfortunately led to a significant increase in error rate in a number of data sets. Hence, these results suggest that MOGA-1 has effectively found a good trade-off between the objectives of minimizing error rate and tree size, whereas MOFSS-1 minimized tree size at the expense of increasing error rate in a number of data sets.

Table 7 compares the performance of MOGA-1, MOFSS-1 and C4.5 using all attributes considering both the error rate and the tree size at the same time, according to the concept of significant Pareto dominance. This is a modified version of conventional Pareto dominance tailored for the classification task of data mining, where we want to find solutions that are not only better, but significantly better, taking into account the standard

Table 6. Tree sizes obtained with C4.5, MOGA-1 and MOFSS-1

	Tree Size (number of nodes)		
Data set	C4.5	MOGA-1	MOFSS-1
Arrhythmia	80.2 ± 2.1	65.4 ± 1.15 (+)	N/A
Balance-Scale	41.0 ± 1.29	16.5 ± 3.45 (+)	**7.5 ± 1.5 (+)**
Bupa	44.2 ± 3.75	7.4 ± 1.36 (+)	11.4 ± 2.78 (+)
Car	165.3 ± 2.79	29.4 ± 5.2 (+)	**17.7 ± 1.07 (+)**
Crx	29.0 ± 3.65	**11.2 ± 3.86 (+)**	24.6 ± 8.27
Dermatology	34.0 ± 1.89	25.2 ± 0.96 (+)	23.2 ± 2.84 (+)
Glass	11.0 ± 0.0	11.0 ± 0.0	11.0 ± 0.0
Ionosphere	26.2 ± 1.74	13.0 ± 1.4 (+)	14.2 ± 2.23 (+)
Iris	8.2 ± 0.44	5.8 ± 0.53 (+)	6.0 ± 0.68 (+)
Mushroom	32.7 ± 0.67	30.0 ± 0.89 (+)	**27.2 ± 1.76 (+)**
Pima	45.0 ± 2.89	11.0 ± 2.6 (+)	9.2 ± 1.85 (+)
Promoters	23.8 ± 1.04	11.4 ± 2.47 (+)	9.0 ± 1.2 (+)
Sick-euthyroid	24.8 ± 0.69	11.2 ± 1.35 (+)	9.6 ± 0.79 (+)
Tic tac toe	130.3 ± 4.25	21.1 ± 4.54 (+)	**10.6 ± 1.4 (+)**
Vehicle	134.0 ± 6.17	95 ± 3.13 (+)	**72.8 ± 10.98 (+)**
Votes	10.6 ± 0.26	5.4 ± 0.88 (+)	5.6 ± 1.07 (+)
Wine	10.2 ± 0.68	9.4 ± 0.26	**8.6 ± 0.26 (+)**
Wisconsin	28.0 ± 2.13	25 ± 3.71	**18 ± 1.53 (+)**
Wins over C4.5	-	15	15
Losses over C4.5	-	0	0

Table 7. Number of significant Pareto dominance relations

	C4.5	MOGA-1	MOFSS-1
C4.5	X	0	0
MOGA-1	14	X	7
MOFSS-1	8	0	X

deviations (as explained earlier for Tables 5 and 6). Hence, each cell of Table 7 shows the number of data sets in which the solution found by the method indicated in the table row significantly dominates the solution found by method indicated in the table column. A solution S_1 significantly dominates a solution S_2 if and only if:

- $obj_1(S_1) + sd_1(S_1) < obj_1(S_2) - sd_1(S_2)$ and
- not$[obj_2(S_2) + sd_2(S_2) < obj_2(S_1) - sd_2(S_1)]$

where $obj_1(S_1)$ and $sd_1(S_1)$ denote the average value of objective 1 and the standard deviation of objective 1 associated with solution S_1, and similarly for the other variables. Objective1 and objective2 can be instantiated with error rate and tree size, or vice-versa. For example, in the bupa dataset we can say that the solution found by MOGA-1 significantly dominates the solution found by MOFSS-1 because: (a) In Table 5 MOGA-1's error rate plus standard deviation (30.14+1.85) is smaller than MOFSS-1's error rate minus standard deviation (40.85-1.45); and (b) concerning the tree size (Table 6), the condition "not $(11.4 + 2.78 < 7.4 - 1.36)$" holds. So, both conditions for significant dominance are satisfied.

As shown in Table 7, the baseline solution (column "C4.5") did not significantly dominate the solutions found by MOGA-1 and MOFSS-1 in any dataset. The best results were obtained by MOGA-1, whose solutions significantly dominated the baseline solution in 14 out of the 18 datasets and significantly dominated MOFSS-1's solutions in 7 data sets. MOFSS-1 obtained a reasonably good result, significantly dominating the baseline solution in 8 datasets, but it did not dominate MOGA-1 in any dataset. A more detailed analysis of these results, at the level of individual data sets, can be observed later in Tables 8 and 9.

5.3. *On the effectiveness of the criterion to choose the "best" solution*

Analyzing the results in Tables 3, 4, 5 and 6 we can evaluate whether the criterion used to choose a single solution out of all non-dominated ones (i.e., the criterion used to generate the results of Tables 5 and 6) is really able to choose the "best" solution for each data set. We can do this analyzing the dominance relationship (involving the error rate and tree size) between the single returned solution and the baseline solution. That is, we can observe whether or not the single solution returned by MOGA-1 and MOFSS-1 dominates, is dominated by, or is neutral with respect to the baseline solution. Once we have this information, we can compare it with the corresponding relative frequencies associated with the solutions found by MOGA-all/MOFSS-all (columns $F_{dominate}$, $F_{dominated}$, $F_{neutral}$ of Tables 3 and 4). This comparison is performed in Tables 8 and 9, which refer to MOGA and MOFSS, respectively. In these two tables the first column contains the data set names, the next three columns are copied from the last three columns in Tables 3 and 4, respectively, and the last three columns are computed from the results in Tables 5 and 6, by applying the above-explained concept of significant Pareto dominance between the MOGA-1's/MOFSS-1's solution and the baseline solution.

As can be observed in Table 8, there are only 4 data sets in which the solutions found by MOGA-1 do not dominate the baseline solutions: car,

Table 8. Performance of MOGA-all versus MOGA-1

	Performance of MOGA-all's solutions wrt baseline solution			Performance of MOGA-1's solution wrt baseline solution		
Data set	F_{dom}	F_{dom_ed}	F_{neut}	Dom	Dom_ed	Neut
Arrhythmia	0.21	0.33	0.46	X		
Balance-Scale	0.7	0	0.3	X		
Bupa	0.31	0	0.69	X		
Car	0.002	0	0.998			X
Crx	0.56	0.05	0.39	X		
Dermatology	0.8	0	0.2	X		
Glass	0	0.06	0.94			X
Ionosphere	0.37	0.12	0.5	X		
Iris	0.8	0.02	0.18	X		
Mushroom	0.68	0	0.32	X		
Pima	0.34	0	0.66	X		
Promoters	0.33	0	0.67	X		
Sick- euthyroid	0.02	0.02	0.96	X		
Tic tac toe	0	0	1			X
Vehicle	0.25	0.18	0.57	X		
Votes	0.6	0	0.4	X		
Wine	0.48	0.31	0.21	X		
Wisconsin	0.5	0.2	0.3			X

glass, tic-tac-toe and wisconsin. For these 4 data sets the solutions found by MOGA-1 were neutral (last column of Table 8), and the value of $F_{neutral}$ was respectively 0.998, 0.94, 1 and 0.3. Therefore, in the first three of those data sets it was expected that the single solution chosen by MOGA-1 would be neutral, so that the criterion used for choosing a single solution cannot be blamed for returning a neutral solution. Only in the wisconsin data set the criterion did badly, because 50% of the found solutions dominated the baseline solution but a neutral solution was chosen.

The criterion was very successful, managing to chose a solution that dominated the baseline, in all the other 14 data sets, even though in 8 of those data sets less than 50% of the solutions found by MOGA-all dominated the baseline. The effectiveness of the criterion can be observed, for instance, in arrhythmia and sick-euthyroid. Although in arrhythmia the value of $F_{dominated}$ was quite small (0.21), the solution returned by MOGA-1 dominated the baseline solution. In sick-euthyroid, 96% of the solutions found by MOGA-all were neutral, but a solution that dominates the baseline solution was again returned by MOGA-1. With respect to the effectiveness of the criterion when used by MOFSS-1, unexpected negative results were found in 2 data sets of Table 9, namely crx and glass. For both data

Table 9. Performance of MOFSS-all versus MOFSS-1

	Performance of MOFSS-all's solutions wrt baseline solution			Performance of MOFSS-1's solution wrt baseline solution		
Data set	F_{dom}	F_{dom_ed}	F_{neut}	Dom	Dom_ed	Neut
Arrhythmia	0.54	0	0.46	-	-	-
Balance-Scale	0.5	0	0.5	X		
Bupa	0.65	0	0.35	X		
Car	0.07	0	0.93			X
Crx	0.89	0	0.11			X
Dermatology	0	0	1			X
Glass	0.99	0	0.01			X
Ionosphere	0.14	0	0.86	X		
Iris	0.86	0	0.14	X		
Mushroom	0	0	1			X
Pima	0.95	0	0.05	X		
Promoters	0.27	0	0.73			X
Sick- euthyroid	0.1	0	0.9	X		
Tic tac toe	0.11	0	0.89			X
Vehicle	0.17	0	0.83			X
Votes	0.1	0	0.9	X		
Wine	0.92	0.01	0.07	X		
Wisconsin	0.45	0.37	0.18			X

sets, despite the high values of $F_{dominate}$, the solutions chosen by MOFSS-1 were neutral. The opposite happened in ionosphere, sick-euthyroid and votes, where $F_{neutral}$ had high values, but single solutions better than the baseline solution were chosen by MOFSS-1.

The relatively large number of neutral solutions chosen by MOFSS-1 happened because in many data sets the tree size associated with the solution chosen by MOFSS-1 was smaller than the tree size associated with the baseline solution, whilst the error rates of the former were larger than the error rates of the latter.

Overall, the criterion for choosing a single solution was moderately successful when used by MOFSS-1, and much more successful when used by MOGA-1. A possible explanation for this result is that the procedure used for tailoring the criterion for MOFSS, described earlier, is not working very well. An improvement in that procedure can be tried in future research.

It is important to note that, remarkably, the criterion for choosing a single solution did not choose a solution dominated by the baseline solution in any data set. This result holds for both MOGA-1 and MOFSS-1.

6. Conclusions and Future Work

This chapter has discussed two multi-objective algorithms for attribute selection in data mining, namely a multi-objective genetic algorithm (MOGA) and a multi-objective forward sequential selection (MOFSS) method. The effectiveness of both algorithms was extensively evaluated in 18 real-world data sets. Two major sets of experiments were performed, as follows.

The first set of experiments compared each of the non-dominated solutions (attribute subsets) found by MOGA and MOFSS with the baseline solution (consisting of all the original attributes). The comparison aimed at counting how many of the solutions found by MOGA and MOFSS dominated (in the Pareto sense) or were dominated by the baseline solution, in terms of classification error rate and decision tree size. Overall, the results (see Tables 3 and 4) show that both MOGA and MOFSS are successful in the sense that they return solutions that dominate the baseline solution much more often than vice-versa.

The second set of experiments consisted of selecting a single "best" solution out of all the non-dominated solutions found by each multi-objective attribute selection method (MOGA and MOFSS) and then comparing this solution with the baseline solution. Although this kind of experiment is not often performed in the multi-objective literature, it is important because in practice the user often wants a single solution to be suggested by the system, to relieve him from the cognitive burden and difficult responsibility of choosing one solution out of all non-dominated solutions.

In order to perform this set of experiments, this work proposed a simple way to choose a single solution to be returned from the set of non-dominated solutions generated by MOGA and MOFSS. The effectiveness of the proposed criterion was analyzed by comparing the results of the two different versions of MOGA and MOFSS, one version returning all non-dominated solutions (results of the first set of experiments) and another version returning a single chosen non-dominated solution. Despite its simplicity, the proposed criterion worked well in practice, particularly when used in the MOGA method. It could be improved when used in the MOFSS method, as discussed earlier.

In the future we intend to analyze the characteristics of the data sets where each of the proposed methods obtained its best results, in order to find patterns that describe the data sets where each method can be applied with greater success.

References

1. Aha, D.W., Bankert, R.L.: A Comparative Evaluation of Sequential Feature Selection Algorithms. In: Fisher, D., Lenz, H.J. (eds.) Learning from Data: AI and Statistics V. Springer-Verlag, Berlin Heidelberg New York, (1996), 1–7.
2. Guyon, I., Elisseeff, A. : An Introduction to Variable and Feature Selection. In: Kaelbling, L. P. (ed.) Journal of Machine Learning Research 3, (2003), 1157–1182.
3. Freitas, A.A.: Data Mining and Knowledge Discovery with Evolutionary Algorithms. Springer-Verlag (2002).
4. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, England (2001).
5. Liu, H., Motoda, H.: Feature Selection for Knowledge Discovery and Data Mining, Kluwer, (1998).
6. Bala, J., De Jong, K., Huang, J.,Vafaie, H., Wechsler, H.: Hybrid learning using genetic algorithms and decision trees for pattern classification. In: Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI-95), (1995), 719–724.
7. Bala, J., De Jong, K., Huang, J., Vafaie, H., Wechsler, H.: Using learning to facilitate the evolution of features for recognizing visual concepts. Evolutionary Computation 4(3),(1996), 297–312.
8. Chen, S., Guerra-Salcedo, C., Smith, S.F.: Non-standard crossover for a standard representation - commonality-based feature subset selection. In: Proc. Genetic and Evolutionary Computation Conf. (GECCO-99), Morgan Kaufmann, (1999), 129–134.
9. Guerra-Salcedo, C., Whitley, D.: Genetic Search for Feature Subset Selection: A Comparison Between CHC and GENESIS. In: Proc. Genetic Programming Conference 1998, (1998), 504–509.
10. Guerra-Salcedo, C., Chen, S., Whitley, D., Smith, S.: Fast and accurate feature selection using hybrid genetic strategies. In: Proc. Congress on Evolutionary Computation (CEC-99),Washington D.C., USA. July (1999), 177–184.
11. Cherkauer, K.J., Shavlik, J.W.: Growing simpler decision trees to facilitate knowledge discovery. In: Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD-96), AAAI Press, (1996), 315–318.
12. Terano, T. , Ishino, Y. :Interactive genetic algorithm based feature selection and its application to marketing data analysis. In: Liu, H. ,Motoda, H. (Eds.) Feature Extraction, Construction and Selection,Kluwer, (1998), 393–406.
13. Vafaie, H., DeJong, K.:Evolutionary Feature Space Transformation. In: Liu, H., Motoda, H. (Eds.) Feature Extraction, Construction and Selection, Kluwer, (1998), 307–323.
14. Yang, J. ,Honavar, V.: Feature subset selection using a genetic algorithm. Genetic Programming 1997: Proc. 2nd Annual Conf. (GP-97), Morgan Kaufmann, (1997), 380–385.
15. Yang J., Honavar V.: Feature subset selection using a genetic algorithm. In: Liu, H., Motoda, H. (Eds.) Feature Extraction, Construction and Selection,

Kluwer,(1998), 117–136.

16. Moser A., Murty, M.N.: On the scalability of genetic algorithms to very large-scale feature selection. In: Proc. Real-World Applications of Evolutionary Computing (EvoWorkshops 2000). Lecture Notes in Computer Science 1803, Springer-Verlag, (2000), 77–86.
17. Ishibuchi, H., Nakashima, T.: Multi-objective pattern and feature selection by a genetic algorithm. In: Proc. 2000 Genetic and Evolutionary Computation Conf. (GECCO-2000), Morgan Kaufmann, (2000), 1069–1076.
18. Emmanouilidis, C., Hunter, A., MacIntyre, J.: A multiobjective evolutionary setting for feature selection and a commonality-based crossover operator. In: Proc. 2000 Congress on Evolutionary Computation (CEC-2000), IEEE, (2000), 309–316.
19. Rozsypal, A., Kubat, M.: Selecting Representative examples and attributes by a genetic algorithm. Intelligent Data Analysis 7, (2003), 290–304.
20. Llòra, X.,Garrell, J.: Prototype Induction anda attribute selection via evolutionary algorithms. Intelligent Data Analysis 7, (2003), 193–208.
21. Coello Coello, C.A., Van Veldhuizen, D.A., Lamont, G.B.: Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer Academic Publishers, New York (2002).
22. Pappa, G.L., Freitas, A.A., Kaestner, C.A.A.: Attribute Selection with a Multiobjective Genetic Algorithm. In: Proc. of 16^{th} Brazilian Symposium on Artificial Intelligence, Lecture Notes in Artificial Intelligence 2507, Spring-Verlag, (2002), 280–290.
23. Pappa, G.L., Freitas, A.A., Kaestner, C.A.A.: A Multiobjective Genetic Algorithm for Attribute Selection. In: Proc. of 4^{th} International Conference on Recent Advances in Soft Computing (RASC), University of Nottingham, UK, (2002), 116–121.
24. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. $3rd$ edn. Springer-Verlag, Berlin Heidelberg New York, (1996).
25. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, (1993).
26. Bhattacharyya, S.: Evolutionary Algorithms in Data mining: Multi-Objective Performance Modeling for Direct Marketing. In: Proc of 6^{th} ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2000), ACM Press (2000), 465–471.
27. Zitzler, E., Thiele, L.: Multiobjective Evolutionary Algorithms: A Comparative Study and the Strength Pareto Approach. In: IEEE Transactions on Evolutionary Computation 3(4), (1999), 257–271.
28. Murphy, P.M., Aha, D.W.: UCI Repository of Machine Learning databases. [http://www.ics.uci.edu/∼mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science, (1994).
29. Coello Coello, C.A.: Handling Preferences in Evolutionary Multiobjective Optimization: A Survey. In: Proc. of Congress on Evolutionary Computation (CEC-2002), IEEE Service Center, New Jersey (2000), 30–37.