In 2000 Congress on Evolutionary Computation (CEC'2000), San Diego, California, July 2000. IEEE Service Center.

A Multiobjective Evolutionary Setting for Feature Selection and a Commonality-Based Crossover Operator

C. Emmanouilidis, A. Hunter and J. MacIntyre
Centre for Adaptive Systems,
School of Computing, Engineering and Technology
University of Sunderland,
St. Peter's Campus, Sunderland, SR6 ODD, UK
{christos.emmanouilidis, andrew.hunter, john.macintyre}@sunderland.ac.uk

Abstract- Feature selection is a common and key problem in many classification and regression tasks. It can be viewed as a multiobjective optimisation problem, since, in the simplest case, it involves feature subset size minimisation and performance maximisation. This paper presents a multiobjective evolutionary approach for feature selection. A novel commonality-based crossover operator is introduced and placed in the multiobjective evolutionary setting. This specialised operator helps to preserve building blocks with promising performance. Selection bias reduction is achieved by resampling. We argue that this is a generic approach, which can be used in many modelling problems. It is applied to feature selection on different neural network architectures. Results from experiments with benchmarking data sets are given.

1 Introduction

Feature selection is the process of selecting a subset of available features to use in empirical modelling. The solution to the feature selection problem is neither trivial, nor unique. The set of optimal features can be different for different hypothesis spaces. Therefore, optimality of a feature subset should only be defined in the context of the family of admissible modelling functions from which it is intended to select the one that is finally deployed [1]. Even for a fixed family of admissible functions, optimal feature selection can only be guaranteed by exhaustive search of all possible feature subsets. This is infeasible when the problem involves a large number of features. Simple heuristic approaches, such as the stepwise methods, are often used. Forward selection, however, may fail to select interdependent features [2]. The root problem with forward selection and backwards elimination is that there is no reason why the best subset of p variables, should contain the (p-1) variables which give the best performance among all subsets of (p-1) variables. The situation improves with sequential replacement [3], where each feature deletion or addition step is followed by a feature replacement step. However, the interrelationships between features can be such that a fixed number of replaced features at each step may not be adequate, and this led to the development of floating search methods [4]. Approaches that maintain a population of solutions, such as evolutionary algorithms (EAs), are less likely to be restricted by interdependencies among features and may speedily perform efficient searches in high dimensional spaces [5]. Evolutionary algorithms have been used many times to aid the selection of feature subsets in various classification tasks (e.g. [6] [7]).

This work introduces a multiobjective evolutionary algorithm (MOEA) setting for the feature selection problem. Multiobjective genetic algorithms have previously been suggested for feature selection in neurofuzzy modelling [8]. Here we extend this approach to MOEA neural network feature selection. In addition, a commonality-based crossover operator is proposed, and employed in the multiobjective evolutionary setting. The specialised operator helps to preserve building blocks with promising performance. Standard operators tend to yield offspring where the number of features selected is the average of the number selected in the parents, thus over-sampling medium-sized feature sets. In contrast, using the new operator, each offspring has on average the same number of features as one particular parent. Thus, the offpring population maintains diversity in the feature subset size space (complexity space) and consequently the algorithm has exploratory power in wider areas of the search space. Selection bias reduction is achieved by means of resampling. We employ a variation of the Niched Pareto Genetic Algorithm (NPGA) [9]. Focusing on the feature selection problem, we employ a sharing function in both the Hamming distance and the complexity space and a specialised operator settings control strategy, tailored to the new crossover operator. To demonstrate that this feature selection approach is generic, we apply it to two different neural modelling approaches, probabilistic neural networks (PNN) [10] and multilayer perceptrons with sigmoid activations (MLP). The developed methodology is computationally simple, and can be applied to problems of significant dimensionality. Results from experimentation with two benchmarking data sets are given.

2 Multiobjective Evolutionary Feature Selection

In multiobjective optimisation, a key concept is that of Pareto optimality. Solutions are compared against each other in terms of Pareto dominance, i.e. a solution is dominant over another only if it has better performance in at least one criterion and non-inferior performance in all criteria. A solution is said to be Pareto optimal if it cannot be dominated by any other solution in the search space. In complex search spaces, wherein exhaustive search is infeasible, it is very difficult to guarantee Pareto optimality. Therefore, instead of the true set of optimal solutions (Pareto Set), one usually aims to derive a set of non-dominated solutions with objective values as close as possible to the objective values (Pareto Front) of the Pareto Set. Feature selection is well-suited to multiobjective optimisation. In the simplest case, it involves two objectives: minimisation of the number of features and maximisation of the modelling performance. In classification tasks, performance can be assessed in terms of the misclassification rate. A common approach is to combine the objectives in a composite function [11]. This may yield solutions good enough on average but not in each one of the objectives separately. Alternatively, multiple runs can be performed to optimise one objective, while keeping the rest at a desired level. For example, it is possible to seek to optimise performance for a given subset size. This can be pursued with EAs in a number of ways but, in principle, it would involve evolving a population of solutions increasingly concentrated around the desired subset size. Such an approach would limit the possibility for creative recombination of the genetic material between individuals whose complexity differs from the desired one. However, a significant part of the exploratory power of a genetic algorithm is attributed to the recombination operator and its ability to discover good solutions by building on existing schemata of promising performance. By imposing restrictions on the subset size, there is a danger of eliminating useful population diversity; thus, chromosomes of diverse subset size may not stand a chance to pass on well performing schemata to the next generation. Such population diversity can be maintained by aiming both at subset size minimisation and performance maximisation, without specifying which objective is more important. We therefore treat feature selection as a multiobjective optimisation problem, in the Pareto sense.

We concentrate on classification and consider a dual modelling performance criterion consisting of two terms: the estimated misclassification rate and the cost function. The former is common regardless of the classifier and the training algorithm employed, whereas the latter depends on the choice of classifier and algorithm. We employ a variation of the niched Pareto GA (NPGA) [9]. This is known to be a fast MOEA [12] since tournament domination is determined by a random subsample of the population. However, any MOEA could be employed in this setting. We introduce some modifications to the main NPGA algorithm to tailor it to the feature selection problem. In the remainder of this section, we describe the main characteristics of the MOEA employed.

- **Offspring and Parent populations**: The offspring population is double the size of the parent population. Mating pairs are selected at random and produce two offspring. Every individual in the parent population has the chance to mate twice on average, at each generation. This polygamy, together with the crossover operator we introduce in the next section and the sharing technique employed, increases the chance that there will be individuals across a greater range of the Pareto front.
- **Elitism**: The formulation of elitism in MOEAs is different from that for the single objective EAs, since instead of a single elite individual there is now an elite set of non-dominated solutions. The MOEA employed here maintains such an elite set and updates it each time a new offspring population is created. The individuals in the set are the first to be inserted into the next generation parent population. The rest are selected by random sampling tournament selection. This is a viable approach as long as the elite set is not too big compared to the size of the population. In the latter case, clustering techniques can be used to reduce the size of the elite set that is copied to the mating pool [13]. Here, we set a minimum population size depending on the chromosome length. The population size is allowed to change to ensure that it is no less than ten times the size of the elite set.
- **Tournament Selection**. The NPGA employs binary tournament selection. In this paper the offspring population is double the size of the parent population. Tournament groups of three individuals are employed, to reduce the chance that an individual from the parent population may not be selected for mating. The tournament sampling set size is set to a tenth of the parent population size [9].
- **Fitness sharing**: In the NPGA algorithm, the tournament winner is determined by checking each competing individual for domination against the sampling tournament set. When the outcome of the tournament is not clear cut, the winner is nominated by performing sharing. Thus, the individual with the smallest niche count is selected, where the niche count is a measure of how crowded the neighbourhood around it is, in the partially filled mating pool. Calculating the niche count based on the Hamming distance helps niche formation primarily in the Hamming space and only secondarily in the complexity space, which is one of the dimensions of the objective space in feature selection. We would like to boost the selection chances of those individuals which lie in a less crowded area of the complexity space. We therefore employ sharing in both the Hamming and subset size space:

$$s(d_{ij}) = \begin{cases} \left[1 - \left(\frac{d_{ij}}{\sigma_S} \right)^{\alpha_s} \right] & if \quad d_{ij} < \sigma_s \\ 0 & otherwise \end{cases} \tag{1}$$

$$d_{ij} = \sqrt{{d_{ijh}}^2 + {d_{ijc}}^2} \tag{2}$$

$$m_i = \sum_{j=1}^{N} s(d_{ij}) \tag{3}$$

where m_i is the niche count of the *i*th individual in the tournament group, d_{ij} is the Euclidean distance of the above individual to the *j*th of the *N* individuals already present in the mating pool, d_{ijh} and d_{ijc} are the corresponding distances in the Hamming and complexity space respectively, α_s is often set to one to correspond to triangular sharing function, and σ_s is the sharing threshold, below which two individuals are considered similar enough to affect the niche count. This niching technique allows the selection of dissimilar subsets that may achieve good performance at the same complexity level, while preventing domination of the population by a single very fit individual at early stages of the process. It also balances the population distribution across the subset size space. Hence, it enables more exploration to take place across a wider range of the non-dominated front.

- **Adaptive operator settings**. General guidelines often suggest values of crossover no less than 0.6 and mutation rate equal to $1/l_c$, where l_c is the chromosome length [14]. When employing the crossover operator introduced in this paper, there is no need for adapting the crossover rate, while there is a simple way to control the mutation rate.

3 Subset Size-Oriented Common Features Crossover Operator

Common uniform or *n*-point crossover operators can be disruptive since they may result in breaking up useful building blocks. When the aim is to identify good subsets of features at different complexity levels, i.e. to develop a range of solutions across the neighbourhood of the Pareto front, common crossover operators can have an additional negative side effect. A standard crossover operating on two individuals, coding subsets of size *n* and *m*, tends to yield offspring with complexity approximately (*n*+*m*)/2. Therefore, the EA tends to explore mostly medium-sized subsets, while the edges of the non-dominated front are less well explored. Improved performance can be achieved if mating restrictions are applied, so that mating between too-dissimilar individuals is discouraged [15] [16]. Dissimilarity here applies both to the Hamming distance between chromosomes and to the size of the coded subsets.

This paper introduces a novel crossover operator that helps to preserve building blocks of promising performance. It also produces offspring populations with relatively even distribution, across the range of the Pareto front. This is achieved whithout any need for mating restrictions. It exploits the concept that preserving the maximal common schema of two parents results in a more creative recombination strategy, compared to standard crossover. This concept has been recently termed the Commonalty-Based Crossover Framework [17] and has been employed for feature selection in conjuction with Random Sample Climbing (RSC), a mutation-based strategy that performs a local search in the neighborhood of each individual in a population of solutions [18]. The Common Features / Random Sample Climbing (CF/RSC) approach maintains a small population of individuals, as starting seeds for the RSC hill-climbing procedure. The CF operator yields a single offspring for each mating pair and each parent is allowed to mate twice in order to fill up the population for the next generation. The offspring inherits the common features of the parents. In a somewhat similar fashion, the common features are preserved by the crossover operator employed in the CHC algorithm, where half of the differing bits are crossed at random [19]. Thus, this operator also tends to average the number of selected bits. CHC feature selection has been examined in [20]. In both CF/RSC and CHC the aim is to identify a single solution.

In this work, instead of aiming at a single solution, we seek to obtain a range of solutions across the Pareto front. In the simplest case, these are non-dominated solutions in a two-dimensional complexity-performance space. Our common features operator utilises the subset size of each mating parent as the desirable target state for each offspring and we therefore call it Subset Size Oriented Common Features crossover operator (SSOCF). The functionality of the SSOCF operator is illustrated in Figure 1. Both offpring preserve the common features of their parents. The non-shared features are inherited by the offspring corresponding to the *i*th parent with probability (n_i-n_c)/n_u, where n_i is the subset size of the *i*th parent, n_c is the number of commonly selected features across both mating partners and n_u is the number of non-shared selected features. Those non-shared features which are not inherited by the first offspring are inherited by the second. Due to the inherent randomness of the non-common features assignment, the subset size of the offspring can be somewhat different from that of their parents, although it will be very close to it and on average equal to it. Following this procedure, each individual in the population becomes a starting point for exploration around its own complexity level. However, this exploration is more flexible than that of stepwise or sequential replacement methods. It is worth noting that this crossover operator has no effect when all bits are common, or if one parent feature

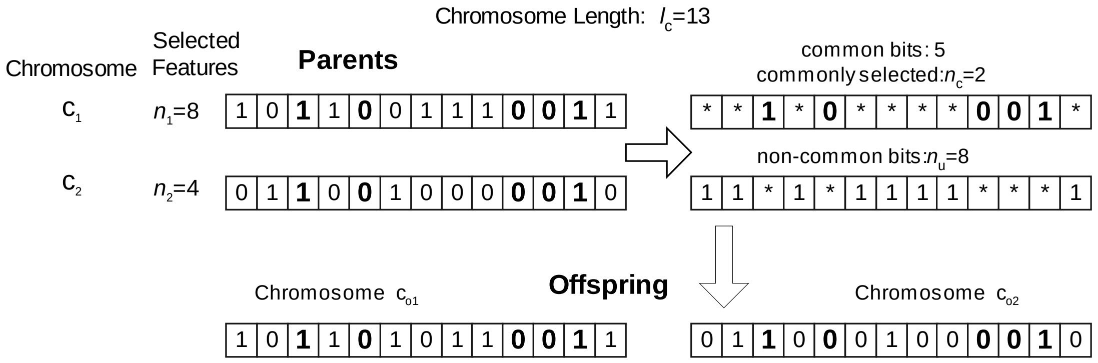

Figure 1: Example of the functionality of the Subset Size Oriented Common Features Operator

set is a subset of its mating partner. In such cases, any potential modification is the result of consequent mutation.

We define as the *subsethood ratio* the ratio of common bits to the subset size of the smallest complexity mating partner. The subsethood ratio becomes one when the feature set defined by one of the mating partners is a subset of the feature set defined by the other. In the extreme case this corresponds to identical partners. The *average subsethood ratio* over all mating partners in a single generation is a measure of the impact of the SSOCF crossover operator on the genetic material recombination. When random mating is applied, the average subsethood ratio, is not uniquely determined by the current population but depends on the exact mating pairs formed. However, in general, the closer it gets to one, the lower the impact of the crossover is. In such cases, it is desirable to allow more exploration to take place by increasing the mutation rate. This observation has led to the following mutation rate adaptation strategy:

- When progress is observed in a generation, i.e. when the non-dominated set is updated with newly found solutions, the mutation rate, p_m , is decreased geometrically, so that the effect of mutation is reduced considerably to allow crossover to exploit the novel genetic material.
- When no progress is observed over a number of successive generations, p_m is increased arithmetically, in order to infuse more randomness into the population. This will increase the hill-climbing opportunities.
- In any event, the mutation rate, is forced to lie within:

$$s_r \cdot \alpha_{lower} / l_c < p_m < s_r^3 \cdot \alpha_{upper} / l_c \qquad (4)$$

where s_r is the average subsethood ratio, l_c is the chromosome length and α_{lower} , α_{upper} are constant upper and lower bounds to prevent p_m becoming either too small to have any meaningful impact, or too large and, therefore, excessively disruptive. We empirically set in our experiments the geometric decrease factor to 0.6 to correspond to a drastic decrease, the arithmetic increase to $0.05/l_c$ so that it is a constant which becomes smaller for longer chromosomes, and α_{lower} to 0.8 and α_{upper} to 3.5, so that the mutation rate can fluctuate around the recommended $1/l_c$ value. These settings apply a small mutation rate as long as the crossover operator leads to progress, while the mutation rate is increased when this ceases to be the case. The average subsethood ratio increases gradually as more and more of the consequent generations' offspring inherit commonly featured bits. When performance fails to improve over a longer period, an increased mutation rate results in the introduction of more and more randomness in the genetic material. This disruption in the population results in a drop in the average subsethood ratio. However, lower average subsethood ratio values imply lower upper value for the mutation rate, which in turn allows the crossover operator to start taking over again, in order to explore the potential benefits of the newly inserted genetic material. This offers a simple way of controlling the the beneficial role of both crossover and mutation. It is worth noting that no crossover rate setting is needed by the SSOCF operator.

4 Neural Network Fitness Evaluation

The major computational cost, associated with the use of EAs for feature selection, is in the feature subset evaluation. This involves building and evaluating a model for a given subset. In order to reduce such costs, one can use a simpler form of model that can be evaluated more quickly during the feature selection stage. We distinguish two methods to reduce the computational cost. First, one can choose a model which has very low training requirements, such as probabilistic neural networks. Second, one can select a form of model such that a master model can be optimised at the beginning of the feature selection process. The master model uses the entire variable set, but can be deployed in such a way that unavailable features can be eliminated from consideration during feature subset evaluation, without requiring retraining of the master model. We demonstrate how the master model approach can be deployed with multilayer perceptrons with sigmoid activations. Fitness estimation noise is reduced by taking the average fitness over different subsamples of the data.

Probabilistic Neural Networks. Probabilistic neural networks (PNNs) have modest computational requirements for reasonably small data sets [10]. Based on kernel density estimation, equivalent to Parzen windows, the PNN uses Bayes rule to estimate posterior class probabilities, that an input vector **x** corresponds to the class ω_i. The probability density function for class ω_i is estimated by:

$$p(\mathbf{x} \mid \omega_i) = \frac{1}{(2\pi)^{N/2}\sigma^N} \frac{1}{m_i} \sum_{j=1}^{m_i} \exp\left[-\frac{(\mathbf{x}-\mathbf{x}_{ij})^T \cdot (\mathbf{x}-\mathbf{x}_{ij})}{2\sigma^2} \right] \quad (5)$$

where: ω_i =class, i=1..number of classes

j=pattern id number

m_i=total number of class ω_i training patterns

$\mathbf{x}_{ij}$=jth training pattern from class ω_i

σ =smoothing parameter

N=dimensionality of feature space

A good choice for the smoothing parameter is usually found after experimentation. The *a priori* probability of class ω_i, p_i , is usually not known. It can be estimated by the frequency of class ω_i patterns in the training set. For a two-class problem, class ω_1 will be selected by the PNN according to the rule $p(\mathbf{x} \mid \omega_1)/p(\mathbf{x} \mid \omega_2) > p_2/p_1$. The primary cost function employed in our experiments is the estimated misclassification rate, while the secondary is a sum squared error form:

$$E_{PNN} = \sum_i \sum_k \left\{ g_i(\mathbf{x}_{ik})^2 + \sum_{class \neq \omega_i} \left[1 - g_i(\mathbf{x}_{ik})\right]^2 \right\} \quad (6)$$

where $g_i(\mathbf{x}_{ik})$ is the network activation for each class, when the kth pattern belonging to class ω_i is inserted. The index k is employed instead of j to indicate that the cost function may be calculated over a data set different from the training set. PNNs are formed in one pass of the data.

Multilayer Perceptrons. A second approach is to employ a multilayer perceptron with sigmoid activations as a master model. The network is trained using all available input features before the feature selection algorithm begins. To evaluate a feature subset, we wish to "eliminate" unavailable features from this model. The simplest way is to substitute the sample mean of the feature from the training set. Other approaches, such as median substitution or even more complex data imputation methods could also be used. Here, in order to perform feature selection, we first train a *master* model to accept all input variables. MLP training in this work is done with the Quasi-Newton method (BFGS). To evaluate a feature subset, we test the master network on the validation data. Features that are selected in the feature mask are copied from the data set. Features that are not selected, instead have the sample mean value provided. The MLPs used during the MOEA search contain very few hidden nodes, to ensure that variation in the output is not due to overfit. Once the MOEA feature selection is complete, more accurate MLPs can be built based upon the feature subset selected.

5 Resampling and Fitness Assignment

Classifiers built without some of the useful features carry an *omission bias*. A second type of bias, more difficult to handle, is the *selection bias* [3]. This occurs as a result of the data-dependent nature of the subset selection process. Selection bias becomes more of a problem when the ratio of the number of training patterns to the number of potential predictor variables is small. A simple way of reducing selection bias is by resampling. Here we select a basic form of this approach, employing ten different random splits of the available data set, each into three subsets. The first set is employed for training; the second for validation during training and for assessing the impact of different subsets of inputs during the MOEA feature selection procedure. The same two sets are also used for building the final MLP models, based on the selected feature subsets. The third set (test set) is kept aside for independent evaluation of the final models. Fitness assignment during the MOEA search is performed by taking the average fitness over the different validation sets. A three-element fitness vector is passed to the MOEA. The first two values, the misclassification rate and the feature subset size, are the primary objectives to be minimised. The third value is the cost function and is treated as a secondary cost term, only employed to compare individuals achieving the same misclassification rate. An additional benefit of the resampling is that it reduces the effect of the noise in fitness evaluation.

6 Experimental Results

We demonstrate how our multiobjective evolutionary algorithm feature selection works on two benchmarking data sets of considerable dimensionality.

Ionosphere. This data set [21] consists of 351 patterns, with 34 attributes and one output with two classes, good or bad, with good implying evidence of some type of structure in the ionosphere and bad the lack of such evidence. Ten random permutations of this data set are employed. Each one is split in 3 subsets. The training set consists of 176 patterns, the validation set 88 and the evaluation set 87.

Sonar. (Mines vs. Rocks) The task is to train a network to discriminate between sonar signals bounced off a metal cylinder and those bounced off a roughly cylindrical rock [22]. The dataset has 208 cases, 60 inputs and one output. of two classes. Ten random splits of 104/52/52 (training/validation/evaluation) data are employed.

The ratio of available data to the number of inputs is very low for both data sets and therefore the selection bias when feature selection is based on a single data split is too high. In both PNN and MLP cases, different neural networks are built for each data split. The average misclassification rate during feature selection is estimated by taking the average rate over the validation sets. Final independent estimation is performed over the separate test sets.

We compare how the MOEA feature selection performs against sequential feature selection. The best non-dominated solutions found by forward selection and backwards elimination are compared against those found by the MOEA. In terms of computational requirements, MOEA is considerably more expensive. Backwards elimination complexity becomes a problem for feature sets of very large size. In sequential feature selection, when comparing two subsets which achieve the same missclassification rate at the same complexity level, we chose the one with the lowest cost function value. In domination tournaments such a situation is considered a tie. The sequential procedures always continue from the subset having the best performance at each step. A triangular sharing function is employed and the sharing threshold is set to $l_c / 4$. The smoothing factor for the PNNs is set to 0.2 and the number of hidden units in the MLPs is 2 and 3 for the ionosphere and sonar data respectively. The initial population has uniform distribution across the feature subset size and the different features. The minimum parent population size is 200, apart from the sonar MLP feature selection, where it is set to 250. In the latter case larger frontiers are created after a few generations and a larger initial population size can improve exploration. In both sequential and MOEA feature selection there are cases where an increase in the subset size does not improve performance. We have carried out 4 MOEA runs for each feature selection task. In all our experiments MOEA has been able to identify at least as good solutions as the sequential feature selection, for each complexity level. In addition, the experiments have shown that the MOEA consistently finds a large number of solutions missed by the sequential feature selection, with both the data sets and with both PNNs and MLPs (Figure 2). When examining all the non-dominated solutions found by all MOEA runs, it is also important to notice is that MOEA discovers a large number of the non-dominated solutions at early stages of the evolution process. This is illustrated in Figure 3, where the average number of non-dominated solutions found, out of a non-dominated set of size 9, is shown at different generations, when performing PNN feature selection on the ionosphere data.

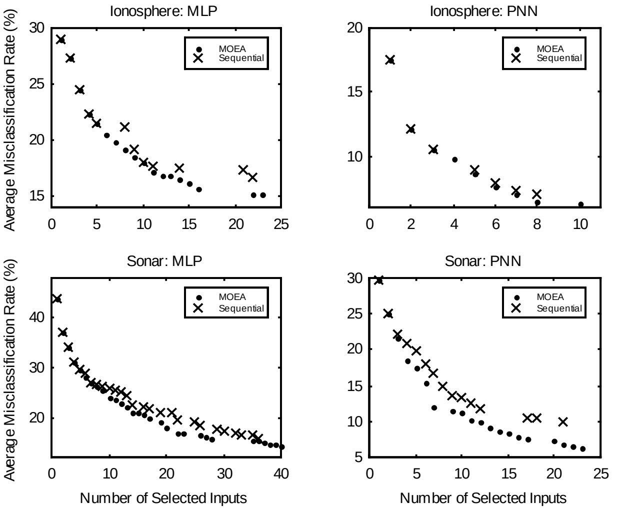

Figure 2: Non-dominated fronts found by MOEA and Sequential Feature Selection

Similar behaviour has been observed for the rest of our experiments, with the MLP sonar case appearing to be the most difficult for the MOEA.

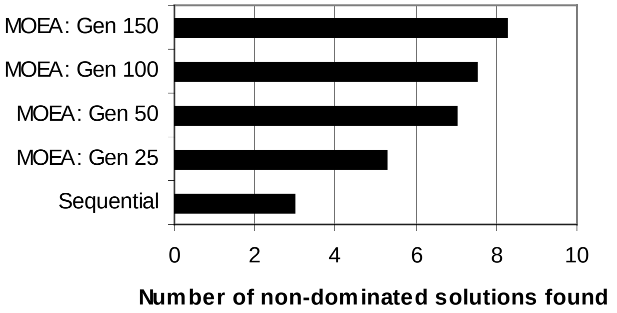

Figure 3: Ionosphere PNN feature selection

Due to space limitations, in the remainder we analyse the results obtained only from one set of MOEA runs. Figure 4 illustrates the subsethood ratio variation and the mutation rate adaptation for MLP feature selection with the ionosphere data.

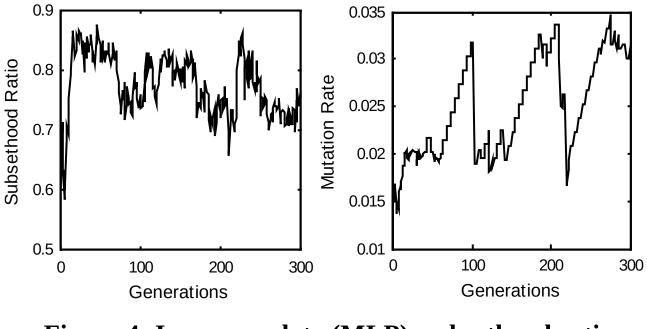

Figure 4: Ionospere data (MLP): subsethood ratio variation and mutation rate adaptation

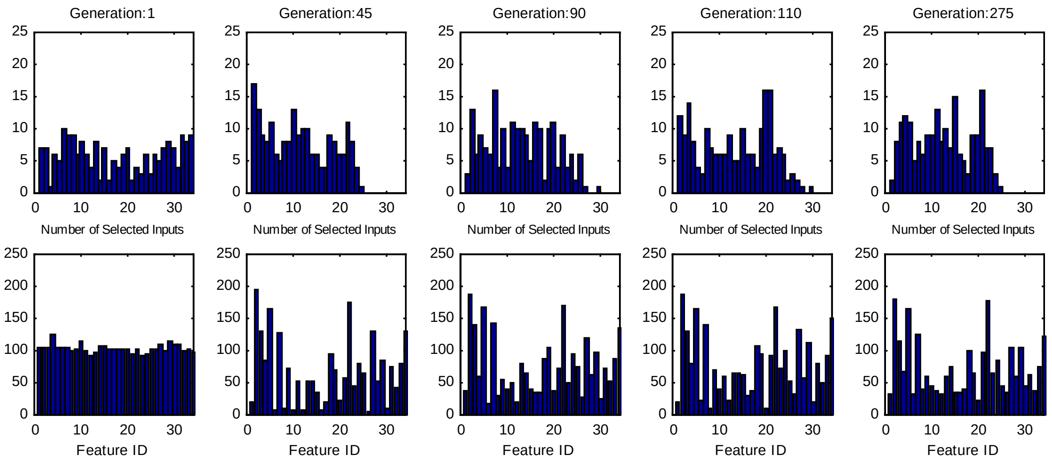

Figure 5: Snapshots of population evolution. Ionosphere (MLP) Upper row: Subset size distribution; dominated individuals of high complexity are phased out during evolution;. Lower row: Individual feature occurrence in the population. *Generation 1:* relatively even occurrence of all features and subset sizes. *Generations 45, 110:* a quite selective phase of the algorithm with low mutation rate and high subsethood ratio; *Generations 90 ,275:* high mutation rate and low subsethood ratio; more of the "less relevant" features do appear in an attempt to bring novelty to the genetic material.

Initially the MOEA progresses at almost every single generation. As more common bulding blocks appear in the population, the subsethood ratio, s_r, increases gradually, exceeding 0.85 just after 15 generations. According to the mutation adaptation strategy, this should bring down the mutation rate, p_m. However, there is a minimum value below which p_m is not allowed to drop and the MOEA has already started with p_m at its minimum. The minimum threshold increases with higher s_r, hence p_m increases up to 0.02, with s_r settling around 0.85. Improved solutions are found up to generation 60. The lack of further progress initiates step increases in p_m. This infuses randomness into the population, which, in turn, results in a gradual decrease in s_r. At generation 102, progress is again observed and p_m is reduced drastically. The low mutation rate allows the crossover operator to take over, in order to exploit the newly introduced genetic material. More and more common building blocks start to form, thus bringing s_r up again. The search is brought back to a productive phase. Progress is observed up to generation 150, where increases in the mutation rate are again imposed. Another large drop in p_m occurs after discovering an improved solution at a stage when s_r is relatively low (generation 219). Figure 5 (upper row) illustrates how the SSOCF operator together with the MOEA feature selection gradually eliminate dominated individuals from the population (ionosphere, MLP). The effect of the mutation adaptation policy can be seen in the charts showing the individual features distribution during evolution (Figure 5, lower row). Once feature selection is completed, final MLP models are built, based on the training and validation data. The sonar data set and, to a lesser extent the ionoshpere, is so sparse that employing a large number of hidden units seems to lead to overfit. Both MLP and PNN models are tested on the independent evaluation data.

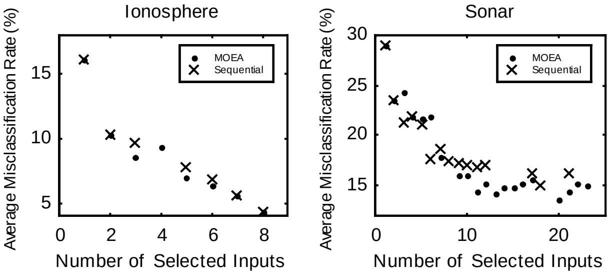

Figure 6: PNN performance on the evaluation data sets after feature selection

Effective selection bias reduction is achieved in the case of the ionosphere data set, while in the sonar case, the data is such that model selection is problematic. Figure 6 shows the performance over the test set for the PNN models. The subsets found by MOEA lead to improved performance with the ionosphere data. Feature selection in the sonar case is not consistent; feature subsets that have on average good performance on the validation data appear to perform poorly on the evaluation data. There are two reasons for that. First, the data is so sparse that even resampling did not offer significant selection bias reduction. Second, apart from very few features, most carry very little information with respect to the output. A summary of the results obtained with the MLPs is shown in the following table:

MLP Feature Selection	Ionosphere	Sonar
Features	11	7
Hidden Nodes	4	3
Misclassification (MOEA)	10.21 %	25.57 %
Misclassification (Sequential)	9.42 %	24.81 %

Overall, in all cases MOEA performed a more efficient search. Results are more consistent with PNNs, which is a full wraper approach [1], whereas MLP feature selecton was more noisy, especially with the sonar data. Evidently, the significance of a drop in the estimated misclasification rate, associated with increasing the feature subset size, should be interpreted with caution. One way of dealing with the problem in stepwise feature selection is by employing F-to-enter or F-to-delete values [3]. These are only employed *a priori* and play a direct part in guiding the search. In MOEA feature selection, F-values can be employed either *a priori* in an adequate multiobjective optimisation formulation, or *a posteriori* for the selection of the final set of non-dominated solutions, without influencing the search.

7 Conclusion

A multiobjective evolutionary approach and a commonality-based crossover operator have been introduced for feature selection. The key issue of treating feature selection as a multiobjective optimisation problem, in the Pareto sense has been discussed. The approach has a number of attractive features. First, it avoids imposing *a priori* restrictions on the search, such as those posed when the subset size and the performance are combined in an aggregating function or when single objective optimisation is pursued. Second, the algorithm exhibits exploratory power across the range of the non-dominated front. This is achieved as a result of the successful combination of the MOEA and the commonality crossover operator introduced in this work. Third, the method is quite generic and can be employed with different classifiers in problems of considerable dimensionality. The result is not a single solution but a range of non-dominated solutions. Therefore, a more informed decision can be taken regarding the features which are deemed to be important. A natural extension to this approach is the adoption of additional objectives for the search. These can be mimimisation of false negative or positive classifications, misclassification and data acquisition costs. However, an increase in the number of objectives increases the size of the Pareto front, and therefore some form of clustering is needed if elitism is to be used. Further work is needed towards comparing the novel crossover operator against other operators, as well as MOEA feature selection against other approaches.

Bibliography

[1] R. Kohavi and G. H. John, (1997) "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, pp. 273-324.

[2] K. N. Berk, (1978) "Comparing Subset Regression Procedures," *Technometrics*, vol. 20, pp. 1-6.

[3] A. J. Miller, (1990) *Subset selection in regression*: Chapman and Hall.

[4] P. Pudil, J. Novovicova, and J. Kittler, (1994) "Floating search methods in feature selection," *Pattern Recognition Letters*, vol. 15, pp. 1119-25.

[5] W. Siedlecki and J. Sklansky, (1989) "A note on genetic algorithms for large-scale feature selection," *Pattern Recognition Letters*, vol. 10, pp. 335-347.

[6] J. Bala, J. Huang, H. Vafaie, K. DeJong, and H. Wechsler, (1995) "Hybrid learning using genetic algorithms and decision trees for pattern classification," IJCAI 95, Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, Montreal, 19-25 August, 1995, pp. 719-724.

[7] A. Jain and D. Zongker, (1997) "Feature selection: evaluation, application, and small sample performance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 153-8.

[8] C. Emmanouilidis, A. Hunter, C. MacIntyre, and C. Cox, (1999) " Selecting Features in Neurofuzzy Modelling Using Multiobjective Genetic Algorithms," Proceedings of ICANN'99, the 9th International Conference on Artificial Neural Networks, 7-10 September 1999. Edinburgh, UK, pp. 749-754.

[9] J. Horn, N. Nafpliotis, and D. E. Goldberg, (1994) "A niched Pareto genetic algorithm for multiobjective optimization," Proceedings of the First IEEE Conference on Evolutionary Computation. pp. 82-87.

[10] D. F. Specht, (1990) "Probabilistic neural networks," *Neural Networks*, vol. 3, pp. 109-118.

[11] J. Yang and V. Honavar, (1998) "Feature subset selection using a genetic algorithm," *IEEE Intelligent Systems*, vol. 13, pp. 44-49.

[12] C. A. C. Coello, (1999) "An updated survey of evolutionary multiobjective optimization techniques: state of the art and future trends," CEC99, Proceedings of the 1999 Congress on Evolutionary Computation CEC99, Washington, D.C., 6-9 July 1999, pp. 3-13.

[13] E. Zitzler and L. Thiele, (1999) "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, pp. 257-71.

[14] A. A. Eiben, Hinterding R. and Michalewicz, Z., (1999) "Parameter control in evolutionary algorithms," *IEEE transactions on evolutionary computation*, vol. 3, pp. 124-141.

[15] R. Hinterding and Z. Michalewicz, (1998) "Your brains and my beauty: parent matching for constrained optimisation," *Proceedings of the 5th International Conference on Evolutionary Computation,* Anchorage, Alaska, May 4-9, 1998, pp.810-815.

[16] C. M. Fonseca and P. J. Fleming, (1995) "An overview of evolutionary algorithms in multiobjective optimization," Evolutionary Computation, vol. 3(1), pp. 1-16.

[17] S. Chen and S. Smith, (1999) "Introducing a New Advantage of Crossover: Commonality-Based Selection," GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference.

[18] C. Chen, C. Guerra-Salcedo, and S. Smith, (1999) "Non-Standard Crossover for a Standard Representation -- Commonality-Based Feature Subset Selection," GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference.

[19] L. J. Eshelman and J. D. Schaffer, (1991) "Preventing Premature Convergence in Genetic Algorithms by Preventing Incest," ICGA 1991, Proceedings of the Fourth International Conference on Genetic Algorithms, San Diego, July 13-16, 1991, pp. 115-122.

[20] C. Guerra-Salcedo and D. Whitley, (1998) "Genetic Search for Feature Subset Selection: A Comparison Between CHC and GENESIS," SGA'98. Symposium on Genetic Algorithms, July 22 - 25, 1998, University of Wisconsin, Madison, USA.

[21] C. Blake, E. Keogh, and C. J. Merz, "UCI repository of machine learning databases. http://www.ics.uci.edu/~mlearn/MLRepository.html. Irvine, CA: University of California, Dept. Information and Computer Science.," , 1998.

[22] R. P. Gorman and T. J. Sejnowski, (1988) "Analysis of hidden units in a layered network trained to classify sonar targets," *Neural Networks*, vol. 1, pp. 75-89.