Hybrid genetic algorithm for dual selection

F. Ros, Serge Guillaume, M. Pintore, J. Chrétien

HYBRID GENETIC ALGORITHM FOR DUAL SELECTION

FREDERIC ROS[1*], SERGE GUILLAUME[2], MARCO PINTORE[3], ~~ARNAUD DEMAN~~[3], JACQUES R. CHRETIEN[3]

1 – AXALTO (Schlumberger), avenue de la Pomme de Pin, St. Cyr en Val, 45060 Orléans cedex

2 – Cemagref, 34000 Montpellier, France

3 – BioChemics Consulting, 16 rue Leonard de Vinci, 45074 Orléans Cedex 2, France

**Author to whom correspondence should be addressed*

AXALTO (Schlumberger), avenue de la Pomme de Pin, St. Cyr en Val, 45060 Orléans cedex

E-Mail: fros@axalto.com
Fax : 0238499044
Phone : 0238499006

Abstract

In this paper, a hybrid genetic approach is proposed to solve the problem of designing a subdatabase of the original one with the highest classification performances, the lowest number of features and the highest number of patterns. The method can simultaneously treat the double problem of editing instance patterns and selecting features as a single optimization problem, and therefore aims at providing a better level of information. The search is optimized by dividing the algorithm into self-controlled phases managed by a combination of pure genetic process and dedicated local approaches. Different heuristics such as an adapted chromosome structure and evolutionary memory are introduced to promote diversity and elitism in the genetic population. They particularly facilitate the resolution of real applications in the chemometric field presenting databases with large feature sizes and medium cardinalities. The study focuses on the double objective of enhancing the reliability of results while reducing the time consumed by combining genetic exploration and a local approach in such a way that excessive computational CPU costs are avoided. The usefulness of the method is demonstrated with artificial and real data and its performance is compared to other approaches.

Keywords *feature selection, genetic algorithm, heuristics, classification, k-nearest neighbor method.*

1. Introduction

Automated database exploitation remains a challenge for pharmaceutical companies. It requires the selection of new compounds with potent specific biological properties in the search for new active leads [1,2]. These strategies involve the use of large compound libraries which are both costly and time-consuming, and require the development of mathematical models to manage chemical properties. It is now well established that despite technological progress, exploiting and mining large quantities of data requires the help of powerful algorithms. This entails incorporating pre-processing stages to avoid blind numerical search and to promote data interpretability. The relevance of pattern recognition systems is basically linked to the method of classification but is highly dependent on measured features representing the pattern. A feature selection stage can significantly reduce the size and complexity of data, and may provide an efficient preprocessing element to reduce the time or space required in practice by the algorithms. For many applications where comprehensibility and visualization are crucial issues, it would be well worth running a data reduction technique before any further algorithm in order to obtain such a size reduction. The basic challenge for pharmaceutical companies is to have a single system to estimate the classification potential of a given database in a reasonable time while having an understandable view of its internal structure. A database is presented as a series of sample patterns described by a set of features and one of the possible categories of the class label. By designing a sub database with the highest classification performances, the lowest number of features and the highest number of patterns, crucial information can be obtained. It is worth noting that the presence of too many "bad" patterns and irrelevant features is likely to make the traditional classification process applied on the whole database inefficient. In this case, the classification performances are very low and do not constitute a source of information even if more than 50% of patterns can be perfectly discriminated by the classes present. This removing process can firstly simplify the determination of the global behavior and secondly may provide guidance in discovering the causes of the abnormal behavior of the underlying applications, indirectly improving the interpretability.

In fact, the need consists in simultaneously selecting relevant features, and in "cleaning" the database by reducing the number of patterns, two basically divergent objectives. The first objective addresses feature selection [3-4], i.e. a problematic and challenging issue extensively studied in the literature. It can be done by complete, heuristic and random methods and aims at making the model performance estimation more reliable while improving the discrimination accuracy. The second objective is related to edition approaches [5] having the role of removing "bad patterns (examples)" or outliers coupled generally with condensing techniques [6] to select only "critical patterns". Outliers are defined as data points which are very different from the rest of the data based on some measurement. The real difficulty is to find tradeoffs between removing too many correct patterns and leaving some small overlap among classes. Many algorithms have been proposed in recent years for outlier detection [7] and this field of research seems to be of great interest in areas such as fraud (Credit card, Computer intrusion, Telecommunication Fraud, voting irregularity, public health…). Although many studies have been devoted to feature and pattern selection separately [8], very few algorithms have been presented that could cope with the particular double feature selection problem [9-10] studied in this paper. It can however be viewed as a search problem where the challenge is to obtain the optimal global solution with a minimum number of experiments. Many search algorithms such as simulated annealing [11], random recursive search [12], tabu search [13], hill climbing [14] and GAs [15] (genetic algorithms) can be good candidates to solve this type of problem. GAs are one of the best-known

techniques for solving optimization problems. Promising results have been reported in many areas and their reputation for selection problems is certain. In the dual selection problem, the data to be handled are often numerous and represented in high dimensional spaces. Although new technologies reduce the problem of excessive processing time, it is a fact that standard GAs may fail [16], particularly when applied to chemometric problems involving many features (sometimes several hundreds) and patterns. Furthermore, the precise role [17-19] of the genetic parameters can seriously obstruct obtaining global convergence, as the most appropriate parameters are dependent on the problem to be solved, the population model and the genetic algorithm performance used. Concretely, the main challenge in GAs is to maintain both an effective search and a good selection pressure. This challenge has attracted the attention of many researchers, and GA weaknesses are subject to various promising developments [20-22] to make them more applicable.

In this paper, a hybrid genetic approach is proposed to solve the problem of designing a subdatabase of the original one with the highest classification performances, the lowest number of features and the highest number of patterns. The GA will converge to a solution based on these three objectives. While the result may not be optimal with reference to the Bayes theory, it will nevertheless provide a more comprehensive view of the database internal structure than the initial set. The aim of this proposal is to provide all the elements to implement an operational system satisfying the double demands of efficiency and speed, from the definition of a valid chromosome to a method for intelligently combining genetic and local approaches to compensate for their respective weaknesses.

In order to do this, the paper is set out as follows. Section 2 summarizes the existing related work in different research areas and presents our contribution. In 3, we introduce our hybrid genetic approach and explain the different heuristics implemented in order to satisfy the double demands of efficiency and speed. In 4, we present the different data sets for the experiments and deal with the results and analysis. Section 5 concludes the paper.

2. Related work and contribution

2.1 Related work

2.1.1 Feature selection

There exists a vast amount of literature (See Dash [23] for a survey and Piramuthu [24] for recent comparisons) on feature selection as central in many areas involving classification problems. Feature selection methods are often classified into two categories [3][25]: filtering approaches which aim at selecting features independently of the learning algorithm, and the wrapper approach which uses a criterion dependent on the performances of the learning algorithm. The wrapper approaches considered here are globally better but only at great computational expense. Feature selection can be performed by complete, heuristic and random methods [26], and aims at making the model performance estimation more reliable while improving the discrimination accuracy. Complete approaches are not computationally feasible in practice and heuristic approaches are the most widely used. Among these heuristic methods one can mention the well-known relief method [27] and its variants, the focus method [28], which selects a pool of features on the basis of their individual power of discrimination. The relief assigns weights to features using the randomly sampled instances. The weights are calculated from the relative relevance of that feature to making class discrimination. Then, the algorithm consists in choosing all the features with a weight greater than or equal to a threshold. Despite good reported results, there appear to be some limitations to considering that the best feature space among all the possible combinations is the one which comprises only the best individual features, without taking into account their possible

synergy. Various statistical approaches [29][30] have also been proposed. Unlike the first approaches, they are rather considered as multiselective since they aim to select a pool of features from a multi-dimensional space. The most popular sequential search algorithms for feature selection are forward sequential and backward sequential selection [31]. These algorithms begin with a feature subset and sequentially add or remove features until some termination criterion is met. Despite their inherent simplicity and the fact that they are not optimal they continue to be widely used for different applications. They are however often very slow and very dependent on the initial directions, which limits the exploration of the feature space. For instance, a relevant feature is unlikely to be selected during the ascendant step as its contribution is masked by the current feature distribution. For the same reason, a relevant feature can be rejected during the descendant procedure. Random approaches are fairly recent [23] , but have reported interesting results despite their simplicity. For example, Skalak [32] selects prototypes and features simultaneously (RMHC-PF1) by random mutation hill climbing. By using a simple characteristic function bit vector representation, and allowing only one bit to be mutated at each iteration, he obtains respectable classification results on four known databases. In a similar category, a number of feature selection techniques based on evolutionary approaches [33] have also been proposed (See Jain and Zongher [34-35] for some evaluations). The classical way is to consider a representation in which individual chromosomes are bit-strings and the fitness function is related to the classification performances of the training algorithm. Random and evolutionary techniques appear promising and can lead to better spaces than systematic heuristic approaches. While they constitute a better exploration tool, they are however intrinsically less stable. As a result, convergence towards the optima is not always guaranteed. Zhang [36] has recently proposed a feature selection technique using the tabu search method that leads to interesting results compared to other approaches, including GAs.
It is a fact that feature selection is still a challenging issue despite a great amount of progress in this field. Hybrid techniques combining random and heuristic techniques are very complementary and hence likely to produce better results despite a more complex implementation.

2.1.2 Editing techniques and outlier detection

The main objective of edition methods is to reduce a native database in order to obtain a subset of relevant patterns leading to more accurate classifiers. It is important to distinguish between different editing techniques. Some aim at removing bad instances to eliminate outliers and possible overlap among classes from a given set of prototypes. Others aim at preserving classification competence as defined in [37] by discarding the cases which are superfluous or harmful to the classification process, thereby conserving only the critical instances. As editing methods are closely related to the nearest neighbors (NN) [38], they are often coupled with condensing methods or include data condensation to some extent in their processes. In this case, the ultimate objective is to find the smallest set of instances which enables the classifier to achieve a nearly similar (or better) classification accuracy to that of the original set. In any cases, this smallest set of instances enables to deduce training sets without irrelevant examples on the basis of well-classified patterns. Many methods have been proposed by different scientific communities, the first one being the “condensed nearest neighbor rule“ presented by Hart [39]. The idea of CNNR is to find incrementally a consistent subset S of the original database such that every member of the database is correctly classified when a 1NN rule is applied to S. By considering S as reference points for the 1NN rule, this instance selection scheme defines a very simple classifier requiring limited storage and gives a classification accuracy close to that obtained when the entire set is considered. See for example on the same principle the “reduced nearest neighbor rule” by Gates [40] or

the “iterative condensation algorithm” by Swonger [41]. A series of instance-based learning methods (IB) is presented in [42]. The basic idea of these methods is that they seek to discard superfluous instances which can be correctly classified by the KNN scheme with the remaining instances. The DROP family methods [43] are among the most popular in the pattern recognition community. Based on new heuristics, they aim at discarding the non-critical instances by starting with the original set and reduce each instance in an ordered way if at least as many of its associates can be correctly classified without it. More recently we can find several approaches using evolutionary algorithms which give promising results [44-45]. Generally surveys [46-48] on the subject show that there appears to be no clear scheme that is superior to all the others. In this paper, we are rather interested in editing techniques, which aim at removing only “bad” instances or outliers. It is worth mentioning that different communities (roughly the “pattern recognition” and the “data mining” community) have proposed editing methods independently, and more surprisingly, there has been to our knowledge no comparison reported in the literature between them. This is probably because the research area and the context (supervised/unsupervised, feature space dimensionalities, dataset sizes…) from which they originate are different. The difficulties, however, are the same. Outliers are basically defined as data points, which are very different from the rest of the data based on some measure and therefore considered as atypical data. The reasons for the presence of outliers are diverse: these data can be completely inconsistent as resulting from noise, exception or simply so far from the other data to explain the underlying mechanism generated by the selected features. We describe the work of the data mining community (for a complete review see [49]), and give more details concerning the methods proposed by the pattern recognition community.

Data mining area

Some methods model data sets as a collection of points in a multi-dimensional space, and provide tests based on concepts such as distance, density, and convex-hull depth (see [50] for a review of these methods). Distance based outlier approaches are the most well known and probably the simplest, as they do not require any knowledge about the pattern distribution. They are generally based on the study of the k nearest examples calculated from a given metric. Different techniques are known: they use different heuristics and manipulate appropriate metrics, the basic idea being to consider a point as normal when it is relatively close to its neighbors and abnormal in the opposite case.

Other methods assume an underlying probability model representing the data and find outliers based on the relationship with this model. Recent work by Shekhar *et al* [51] introduced a method for detecting spatial outliers in graph data sets. The method is based on the distribution property of the difference between an attribute value and the average attribute value of its neighbors. Chang-Tien Lun *et al* [52] propose three spatial outlier detection algorithms to analyze spatial data in order to reduce the risk of falsely claiming regular spatial points as outliers. The idea is to compare the attribute of each point with the attribute values of its neighbors by means of a comparison function. The first two algorithms (r and z algorithms) are iterative and differ by their comparison functions. Once an outlier has been detected, its attribute value is modified so that it will not impact the subsequent iterations negatively. Then, by replacing the attribute value of the outlier by the average attribute value of its neighbors it avoids normal points close to the true outliers being claimed as possible outliers. The median algorithm defines the neighborhood function differently. The attribute value is chosen to be the median of the attribute values of its k nearest neighbors, the motivation being that the median is a robust estimator of the center of the sample. The “editing by ordered projection” EOP proposed by Jesus S. Aguilar *et al* [53] is based on the projection of the examples in each dimension. It presents some interesting characteristics such as a considerable reduction in the number of examples from the database, lower

computational cost due to the absence of distance calculations, and conservation of the decision boundaries. Despite its simplicity, the results reported used as a preprocessing method for the C4.5 classifier tree [54] are very interesting. As mentioned in [55], most of the proposed methods are more applicable to low dimensional versions and lose their algorithmic effectiveness for the high dimensional case due to the sparseness of the data.

Pattern recognition area
Wilson editing [56] consists in removing any examples misclassified by its k nearest neighbors. It assumes that these examples are noisy and then acts as a noise-filtering pass. It leads to smoother boundaries between classes, and as mentioned in [57], Wilson reported improved classification accuracy over a large class of problems when using the edited set rather than the original, unfiltered set. Repeating Wilson editing is identical to Wilson's approach. It consists in repeating the Wilson editing method until there is no change in the reference set. Multi-edit [58] is also a derived version of Wilson's approach. It consists in repeating Wilson editing to N random subsets of the original dataset until no more examples are removed. Citation-editing [59] is derived from Wilson editing. Instead of considering only the k nearest neighbors of each example y_i for the removal, the method also considers the c nearest cities having among their k nearest neighbors y_i. If the class of the majority among the (k + c) examples is different from the class of y_i then y_i is removed from the dataset.
The Depuration algorithm [60] is based on a different philosophy. It consists in removing some "bad" examples while changing the class labels of some other examples. Two parameters k and k' have to be set according to (k+1) / 2 <= k'<= k. The idea is to consider the k nearest neighbors of each y_i example of the database. If a class label c is held by at least k' nearest neighbors, y_i is set to c otherwise it is removed from the database. Two new editing approaches have been derived for the Depuration algorithm. The RelabelOnly algorithm is a version of the Depuration algorithm without the removing step. The RemoveOnly algorithm is a version of the Depuration algorithm without the "relabel" step.
The Neural Network Ensemble Editing algorithm [61] follows the same scheme as the RelabelOnly algorithm using the neural networks generalization capability even in the presence of noise. By combining the classification results of a set of neural networks trained on the dataset, it is possible to change the label of a given example if needed.

2.1.3 Main trends in genetic algorithms

The particular double feature selection problem studied in this paper can be viewed as a search problem where the challenge is to obtain the optimal global solution with a minimum number of experiments. The reputation of GAs for solving multi optimization problems makes them good candidates. Diversity of individuals and a selecting pressure within a genetic population are two key elements in GAs. Although they aim at opposite goals, they need to cohabit to encourage the best algorithm convergence. The first element promotes the presence of chromosomes at different parts of the search space to enable an efficient exploration. Without an active diversity in the search, the search is likely to be trapped in a local optimum as chromosomes are too "alike", and make the genetic transformations inefficient. The second element encourages the survival of the best chromosomes, and therefore the creation of similar chromosomes in a very small subset of the space. Without a minimum of pressure, the search is similar to a random walk search, and has little chance of converging towards one optimum. While one of the most interesting features in GAs is the flexibility of the technique, choosing the right genetic parameters to control population evolution is time consuming and sometimes impractical. Parameters are not independent, are application dependent and should vary to match the evolution process. The issue of controlling the values of various parameters of an evolutionary algorithm is one of the most

important and promising areas of research in evolutionary computation (see [62] for a review). Most of the work in parameter adaptation [63-65] has focused on adaptating mutation, crossover rates and population sizes and despite encouraging results it seems difficult to extract general rules for a given problem.

Niching methods (See [66] for a complete introduction) have been developed to counteract the convergence of the population to a single solution by maintaining a diverse population of members throughout its search. By analogy with nature, a niche can be viewed as a subspace in the environment that can support different types of life [67]. The two most popular niche methods are *sharing* and *crowding*. Fitness sharing was introduced by Goldberg and Richardson [68] and applied successfully to a number of difficult and real-world problems [69]. Fitness sharing modifies the search landscape by adapting chromosome fitness values so that the regions with a dense population are penalized, and the others rewarded. Typically, the shared fitness of an individual i is defined as $f_{sh,i} = f_i / m_i$ where m_i is the niche count, given by $m_i = \sum_{j=1}^{n} sh(d_{ij})$, where n is the number of chromosomes in the population, d_{ij} represents a distance between the i^{th} and j^{th} chromosome based on genotype or phenotype, and sh() defines a decreasing function (from 1 to 0) measuring the amount of sharing or similarity between two chromosomes. The most widely used sharing function is defined by $sh(x) = 1 - x/d$ if $x < d$ otherwise $sh(x) = 0$, d representing a threshold distance expected to delineate the niche regions.

It should ideally produce high values inside the same niche ("intra") and low values "inter" niches in order to develop the potential of each niche independently without overlapping. This remains an open problem, even if different ways [70-71] of improving the sharing functions have been proposed. Fitness sharing can also be accomplished via intelligent crossover (IC). For example, Bo Yuan [72] proposes a new crossover operation where the crossover points may be different in two parents in order to create offspring on different parts of the two parents. His idea is to force the exploration of other regions of a search space even when most of the individuals are located in the same region. For example, two identical parents can produce one different offspring using an asymmetric two-point crossover, which is impossible in standard two-point crossover. Although the efficiency of the method seems to be application-dependent, the paper nevertheless shows through different simulations that the approach can outperform standard two-point crossover.

One of the most widely implemented crowding techniques is tournament selection [73]. In tournament selection, a set of individuals is randomly chosen from the current population and the best of this subset is placed in the next population without undergoing other genetic operations, the size of the tournament controlling the amount of selection pressure and hence convergence speed. The basic idea of crowding methods is then to encourage the insertion of new chromosomes in the population by replacing the most similar ones. The initial work of De Jong [74] consisted in replacing the most similar chromosome of a random subset of the entire population. Given the difficulty of maintaining more than two local optima in the population due to the stochastic errors in the replacement of population members, Mahfoud [73] proposed deterministic crowding (DC) which introduces a notion of competition between children and parents. Each child c_i (i=1,2), resulting from the crossover between two parents p_1 and p_2 and optionally from mutation operations, replaces the nearest parent if it has a higher fitness. DC results in two sets of tournaments: p_1 against c_1 and p_2 against c_2 or p_1 against c_2 and p_2 against c_1. The set of tournaments that yields the closest competitions is held. DC is reputed to be better than sharing approaches but can however suffer from crossover interactions between different niches. Restricted Tournament Selection [76] initially selects two chromosomes A and B from the population, and forms two new chromosomes A' and B' through crossover and mutation operations. A' and B' are then

placed into the population as in a steady state GA (only two offspring are produced at each generation). For each of A' and B', w (windowsize) more members of the population are scanned, and the closest among the group to A' and B' is saved for further processing (say A" and B"). A" competes against A' and B" competes against B' and the winners are inserted in the new population. Several methods including variants of DC such as elitist recombination [77], keep-best reproduction [78] and correlation family-based selection [79] are presented and compared in [80] through six test functions and three-world problems. In this paper, the author proposes a new replacement strategy for steady-state genetic algorithms based on a measure for the contribution of diversity and the fitness function, which outperforms other replacement strategies presented in the literature. Despite the progress in the GA field and its promising results, it is now well "established" (chiefly by practioners) that pure GA are not well suited to fine tuning search in complex search spaces, and particularly that the amount of parameterization can lead to extremely high computation costs to obtain good efficiencies. This entails incorporating additional techniques to obtain reliable results in the context of real applications. Even if experimental researchers and theoreticians are particularly divided on the issue of hybridization [81], several techniques have been reported in the GA literature. These include Genetic Local Search, often called memetic/hybrid algorithms [82], Random Multi-Start and others. Random multi start local search has been one of the most commonly used techniques. In this technique, a number of solutions are generated randomly at each step, local search is repeated on these solutions, and the best solution found during the entire optimization is kept. Complete and introductory studies related to hybrid approaches can be found in [83-85], applications in the field of chemometrics in [86-87] and more recent advances in [86]. Although a huge number of papers dealing with memetic algorithms architectures and design principles have appeared in the last 10 years, the diversity of algorithmic design space explored is relatively small [89]. Many of them are too time-consuming as they require considerable tuning of the local search and evolutionary parts of the algorithm. Although the philosophy of memetic algorithms is always the same, it appears that each particular application requires its own memetic algorithm. In [90] for example memetic algorithms are presented for the Traveling Salesman Problem (TSP), Quadratic Assignment Problem (QAP), Minimum Graph Coloring Problem (MSG) and for Protein Structure Prediction (PSP).

2.1.4 Dual selection

The particular double feature selection problem studied in this paper can be viewed as a search problem where the challenge is to obtain the highest classification perfomance with the best datasubset. It can be then seen as a specific multi objective problem. While traditional mathematical approaches offer a variety of solutions, evolutionary algorithms seem particularly suitable to solve multiobjective optimization problems because they deal simultaneously with a set of possible solutions in a single run of the algorithm. A number of multi-objective evolutionary algorithms have been successfully reported in the literature for several years. The techniques can be classified in Non-Pareto and Pareto Techniques [91]. Non-Pareto techniques do not directly incorporate the concept of Pareto Optimum and are generally efficient but better adapted to managing only a few objectives. The most well-known are the weighting approaches which aim at combining all the objectives into a single objective by aggregation. The VEGA method [92] proposed by Schaffer consists in stratifying the population in several sub-populations, each having different objectives to manage. As mentioned in [93], this method is a criterion-based approach where each time a chromosome is chosen for reproduction, potentially a different objective will decide which member of the population will be copied into the new population. Techniques which directly incorporate the concept of Pareto Optimum were first proposed by Goldberg [94]. Some

approaches use for example the dominance rank, i.e., the number of chromosomes by which a chromosome is dominated, to determine the fitness value. Diversity and elitism preservation are central in multiobjective problems, and the most recent and widely-used approaches [95] integrate these aspects. As the problem of selecting simultaneous features and instance patterns are essentially handled by the aggregating approach, we limit this review to the most closely related methods and leave the reader interested in this global area to read dedicated papers (See [96-98] for tutorials/surveys and [100] for a more specific paper). Papers related to the simultaneous selection of features and instances are very few in number, and unquestionably the most famous are those by Skalak [101] and Kuncheva [102]. All of them aim to design optimal nearest classifiers. It is worth mentioning that our objective is different as we aim at finding the largest pattern set which will predict the ability of the database to discriminate the classes, and not the smallest pattern set containing critical instances for a 1NN classification. They do however address a similar double selection problem in terms of problem complexity and are therefore considered as the closest references for this paper. We have already mentioned the work of Skalak in 2.1 as his idea of performing selection by random mutation hill climbing can serve for selecting prototypes (RMHC) and features separately or simultaneously (RMHC-PF1). Even if this solution appears incomplete as there is no mechanism to drive the final size of the different sets, the results obtained are impressive related to its simplicity. Kuncheva [102] has proposed initial work for the simultaneous editing of patterns and selecting of features by aggregating objectives. Based on a similar philosophy, her work has lead to other studies such as those by S.Y Ho [103] and very recently J.R Cano [104] and Chen [105] which incorporates an intelligent crossover to improve the population diversity and apply another multiobjective approach. In Kuncheva [102], the goal was to design optimal nearest neighbor classification by minimizing the sizes of both prototype and feature sets. She applied a standard GA well designed to perform edition and selection in a single step and showed that it could achieve a good compromise with both high discrimination and a moderate number of features. The results presented were better by far than all the other combinations of tested approaches where selection and edition were applied in different steps. According to the literature results, various promising approaches based on GAs seem to be helpful in managing simultaneous feature and instance patterns. The drawback of GAs remains the difficulty of setting up and driving the algorithm to obtain good solutions in a reasonable time, especially with a large database. Although these aspects are crucial for practitioners, there is no clear guide or any mention in the literature about how to set the genetic parameters, nor about the time required for the algorithms to converge toward good solutions.

2.2 Contribution of the paper

In this paper, we propose to treat the twofold problem of editing instance patterns and selecting features as a single optimization problem by the use of a specific hybridized genetic algorithm. Selecting optimal features is absolutely necessary for classification purposes.
In fact, the presence of irrelevant or redundant features may confuse the learning algorithm and lead to bad classification performances. Moreover, there is no ambiguity to say that reducing the original training by removing “bad instances“ is likely to increase the classification performances. As the problems of instance reduction and feature selection are not always independent, we propose a way to handle this double problematic as a single problem formalized as follows:

Problem formulation
Let $X=\{X_1,\ldots X_f\}$be the set of features describing objects as n-dimensional vectors and $Z=\{z_1,\ldots,z_p\}$, $z_j \in R^f$, be the data set. Associated with each item z_j, j = 1,…,p is a class label

from the set L={1,…,l}. Given a classifier C, the objectives of data reduction and feature selection are to find subsets $S_1 \subseteq X$ and $S_2 \subseteq Z$ such that the classification accuracy of S_2 is maximal and at the same time optimize the sizes of the reduced sets to have $|S_1|$ minimal and $|S_2|$ maximal, where |.| denotes cardinality. The formulation is then the following:
Find S_1 and S_2 in the combined space to manage three different objectives in the same algorithm such that:

$$\begin{cases} C(S_2) \text{ is maximal} \\ |S_2| \text{ is maximal} \\ |S_1| \text{ is minimal} \end{cases} \quad (1)$$

To solve this problem, we propose a hybrid GA having the double objective of reducing (examples) and selecting (features) while reaching the highest classification score. By adapting the chromosome structure, GAs can integrate a feature scheme able to select a pool of features from a multi dimensional space, and at the same time a pattern selection scheme. The GA presented in this paper integrates dedicated heuristics and mechanisms for the dual selection problem. To the best of our knowledge, there appears to be no reported method which simultaneously treats this double problem of instances reduction and features selection to achieve this particular objective. The reader should observe that the approach is technically close to that of methods devoted to designing nearest neighbor classifiers but the heuristics and mechanisms introduced to make the method efficient and practical are different. In section 3, we then present the hybrid GA and specifically the different heuristics implemented to solve the dual selection problem.

3. The hybrid algorithm

The whole procedure is made up of two distinct steps. It is summarized in the diagram of Fig. 1. The first one, which can be called a preliminary phase, is a pure GA. The goal is to promote diversity within the chromosome population S in order to remove the unused features and to prepare the second step, called the convergence phase. Then, the objective is to find a set of possible solutions. Instead of diversity, internal mechanisms are introduced to favour elitism and some local tuning is combined with the GA during the convergence phase. In this phase, computing resources dedicated to local tuning are progressively increased.
It should be noted that the transition between the preliminary and the convergence phase is automatic.
Preserving both elitism and diversity constitutes the main challenge for a GA.
The aim of our partitioning phase is firstly to encourage diversity and secondly elitism through the choice of known genetic algorithms. The management of diversity and elitism is also managed inside each phase. We have incorporated two mechanisms, i.e.: an archive population and a breaking mechanism in order to auto balance diversity and elitism.
The archive population is used as a repository of solutions, provides an extra source of results and favours more elitism.
Each time a sign of premature convergence is detected in the current population, the breaking mechanism integrated with the main objective of preventing premature convergence encourages diversification by re-seeding selected chromosomes.
The time feature is an essential factor for the use of GAs.
Hybridation with local approaches can quickly become unpractical. Most of the known memetic applications deal with relatively small systems. Unused and worse features are removed at the end of the first phase in order to avoid needlessly heavy calculations. Furthermore, local approaches are incorporated in such a way that computational CPU costs

are avoided, and the reliability of the results enhanced. As shown in the diagram, several "tricks" are incorporated to reduce both the number of solutions to which local search is applied and the number of inspected chromosome components.
The first subsection goes into the GA details while the second is dedicated to the hybrid component.

3.1 The Genetic algorithm

Chromosome:

As the optimization procedure deals with two distinct spaces, the feature space and the pattern space, both are managed by the GA. A chromosome represents the whole solution. It is encoded as a string of bits, whose length is f+p, f being the number of available features and p the number of patterns in the training set. In a chromosome a 1 for the *ith* feature or pattern stands for its selection, while a 0 means it is not taken into account.
As the number of features is likely to be smaller than the number of patterns, in order to speed up the procedure and to improve the exploration power of the algorithm, the two spaces are managed independently at each iteration by the genetic operators such as crossover and mutation. This means the whole chromosome is the union of two distinct subchromosomes, the first one to encode the feature space and the second one the pattern space. In each subchromosome a classical one-point crossover is applied. It processes inside comparable fields. This way doing is likely to yield better exploration and results.
We superimpose some restrictions for a chromosome to represent a valid solution. The first one is obvious: the number of selected features is not zero, $|S_1| \geq 1$. Otherwise, no input space would be defined. The other condition aims at ensuring that all the classes are managed by the system, whatever their cardinality. The number of prototypes of a given class has to be greater than a defined proportion $freq_{rep}$. Without this kind of constraint, bad results with small classes could be compensated by good results with larger ones.
In the case of large vector sizes such as those found in the chemometric field, the random solution is not very appropriate as not only the time taken by the process increases, but the performance of designed classifiers will not be guaranteed. Then, the initial chromosomes are not generated in a completely random way: the number of active bits is limited for both spaces. The intervals are $[a_1 p, a_2 p]$ and $[1, \min(a_3, f)]$ (typical values are $a_1=0.2$, $a_2=0.9$ and $a_3=30$).

Fitness function:

The choice of the fitness function is of prime importance in a GA design. The one we propose takes into account the three contradictory objectives: maximize the classification results, maximise the number of prototypes and minimize the number of features. It is, of course, defined for valid chromosomes only. C being the selected classifier, its analytical expression, to maximize, is as follows:

$$F = \begin{cases} C(S_2) * \lambda_f * \lambda_p & \text{if the chromosome is valid} \\ 0 \end{cases} \quad (2)$$

(λ_f, λ_p) compensate each other and they can be seen as penalty terms for $C(S_2)$. They are respectively maximal for a minimum of features and a maximum of patterns introduced in the chromosome. They are designed in a similar way. λ_p (see Fig. 2) depends on two parameters μ_p and Δ_p between 0 and 1, μ_p defines the lowest value for λ_p and Δ_p is a threshold:

$$\lambda_p = \begin{cases} \mu_p \ \ (\mu_p < 1) & \text{if } (\frac{p_a}{p}) \le \Delta_p \\ a_p * (\frac{p_a}{p}) + b_p & \text{else} \end{cases} \quad (3)$$

where p_a is the number of patterns in the current chromosome. a_p and b_p are calculated so that (λ_p) is 1 when $p_a/p = 1$ and (λ_p) is μ_p when p_a/p is Δ_p. The values of μ_p and Δ_p have to be chosen carefully as they are representative of the importance dedicated to the pattern set. Typical values are $\Delta_p = 0.5$, and $\mu_p = 0.95$.
In the same way, λ_f (see Fig. 3) depends on three parameters μ_f, Δ_{f1}, Δ_{f2}. μ_f defines the lowest value for λ_f, Δ_{f1} and Δ_{f2} ($\Delta_{f1}<\Delta_{f2}$) are two thresholds:

$$\lambda_f = \begin{cases} 1 & \text{if } f_a \le \Delta_{f1} \\ \mu_f & \text{if } f_a \ge \Delta_{f2} \\ a_f * (\frac{f_a}{f}) + b_f & \text{else} \end{cases} \quad (4)$$

where f_a is the number of features in the current chromosome. a_f and b_f are calculated so that (λ_f) is 1 when $f_a=\Delta_{f1}$, and (λ_f) is μ_f when the number of features is Δ_{f2}. Typical values are $\Delta_{f1} = 2$, $\Delta_{f2} = 15$ and $\mu_f = 0$ expressing that λ_f decreases linearly according to a_f and b_f.

Population evolution:

Most methods such as Determinist Crowding (DC), Restricted Tournament Selection (RTS) and others are continuously looking for a balance between elitism and diversity in the current population. We propose to use two distinct populations with different evolution rules and no direct interaction. The first one is called the current population, pop_c, its evolution is managed using classical genetic schemes (elitism, DC, RTS). The second one is called the archive population, pop_a, it acts as an evolutionary memory. It is a repository of good chromosome solutions found during the evolution. At each generation, pop_a is updated and may be used to partially regenerate pop_c if needed. The final pop_a constitutes the output of the GA.
The current population needs to be reseeded when a diversity index drops below a given threshold. The breaking mechanism is then used to produce major changes in the current population by including chromosomes from the archive population or applying a high mutation rate to refresh the chromosome.

The diversity index is based on the chromosomes similarities. Two chromosomes are said to be similar if their hamming distance is less than a predefined threshold. As a chromosome is the union of two subchromosomes, the hamming distances are computed in the two different spaces. The similarity between the *ith* and *jth* chromosomes is:

$$s(i,j) = \begin{cases} 1 & \text{if } d_h^f(i,j) < n_f \text{ and } d_h^p(i,j) < n_p \\ 0 & \text{else} \end{cases} \quad (5)$$

where $d_h^f(i,j)$ (resp. $d_h^p(i,j)$) stands for the hamming distance in the feature (resp. pattern) space, and n_f (resp. n_p) is a predefined threshold.

The proportion of chromosomes similar to the *ith* one is given by:

$$P_s(i) = \frac{1}{(s-1)} \sum_{j=1, j\neq i}^{s} s(i,j) \quad (6)$$

where *s* is the population size.

The breaking mechanism is active when there are a lot of similar chromosomes within the population. The $P_s(i)$ are thresholded to compute the diversity index:

$$DI = \frac{1}{s} * \sum_{i=1}^{s} S(i) \quad \text{where} \quad S(i) = 1 \text{ if } P_s(i) > th_{min} \text{ and 0 else} \quad (7)$$

When the diversity index, DI, is too low, some of the chromosomes which have a lot of similar ones in the population, some of the *ith* ones for which $S(i) = 1$, are either replaced by ones randomly chosen in the archive population or re-generated with a high mutation probability.

The update of the archive population takes into account both elitism and diversity. The decision to include a given chromosome in pop_a is based on two criteria, the first one is the fitness score. If there exists a chromosome in the archive population with a much lower score than the candidate, it is replaced by the candidate. This is the elitist side of the process. If the candidate score is slightly better than others, the candidate replaces the chromosome with the most comparable structure, the one with the closest hamming distance. Even if the candidate score is a little worse than that of the archive population, it can be used to replace one of a set of similar chromosomes, in order to increase the diversity level. Balance between elitism and diversity can be adapted during the genetic life.

As previously stated the whole procedure is made up of two steps. For the preliminary phase, whose main objective is to promote diversity, we have selected the RTS genetic scheme for pop_c evolution, the diversity level being controlled by the breaking mechanism. There is no hybridization with local approaches within this preliminary phase.

This genetic phase is driven by p_c and p_m, respectively the probability of crossover and mutation. Details of the native algorithm can be found in [76].

This stage automatically ends when there is a large enough number of “competent” and diverse chromosomes in the population. This condition can be formulated as follows. Let S' be the set of chromosomes whose fitness score is greater than a threshold, and F_{div} (resp P_{div}) a diversity measure in the feature (resp. pattern) space.

The fulfilment of the condition states that the three indexes, $s'=|S'|$, F_{div} and P_{div} have to be sufficiently high. The first condition expresses the level of “competence” of the whole population and the others the level of diversity. It should be noted that the three conditions have to be independently satisfied, that is to say the measures have to be all above specific thresholds.

The diversity measure we use is:

$$F_{div} = \frac{1}{s'} \sum_{i=1}^{s'} \sum_{j=1, j\neq i}^{s'} d_h^f(i,j) \quad (8)$$

An analog definition stands for P_{div}.

A cautious implementation also controls the end of the first phase by the number of iterations.

At the end of this step, the worst features, i.e. those which are selected with a low frequency (below min_{freq}), are discarded from the candidate set. This selection is based on a feature histogram of f dimensionality cumulating the feature vector of each explored

chromosome presenting a fitness score greater than a predefined threshold ($min_{fitness}$). This filter contributes to making the GA selection easier and particularly faster.

In the next step, the convergence phase, an elitist approach is preferred to select an accurate solution, diversity remaining controlled by the archive population, and the GA is combined by local search procedures.
An elitism approach has therefore been preferred to promote convergence, the diversity among the population remaining controled by pop_a and encouraged by the breaking process. More computing resources are progressively allocated in the use of local approaches.

The elitism scheme we have selected is driven by p_c and p_m but also p_s and p_r, respectively the probability of selection and rejection. In this scheme, the (p_s * s) best chromosomes from generation n are copied in the population n+1, (p_r * s) are discarded and replaced in a random way in the population n+1.

Then, p_c *(1 - p_r) * s parent chromosomes are submitted to the crossover and operator providing p_c *(1 - p_r) * s children. The children and the remaining chromosomes (1 - p_c)*(1 - p_r) * s are updated via the mutation operator according to p_m. Parents and children are put together and the best (1- p_r) * s chromosomes are placed in the population n+1 with the remaining chromosomes. This scheme is illustrated in Fig. 4.

3.2 Local tuning

As previously stated, recent literature reports that GAs are not easy to tune when dealing with large systems. The objective of a GA is twofold: space exploration and solution tuning. Reaching both of these objectives may prove difficult. The hybrid part of the algorithm is devoted to helping the GA in the tuning phase. Thus, the GA is in charge of the space exploration, it is likely to find a set of acceptable solutions, and the local procedures aim at improving these solutions by an exhaustive search in their neighborhood.

Of course, extensive search is time consuming, and local tuning has to be applied carefully, only when the expected gain is higher than the cost.

The local tuning includes two different phases: an ascending and a descending procedure.

The ascending phase aims at aggregating new elements, features or prototypes, in a given chromosome while the goal of the descending phase is, on the contrary, to remove features or prototypes from the chromosome description. Both procedures are random free. They are based on the population yielded by the GA.

Let us first consider the ascending step. It can be applied to the feature or the prototype space. Let S' be the set of chromosomes in the current population whose fitness score is higher than a given threshold (min_{local}), S“ a subset of S' randomly selected, $S'_1 \subseteq X$ be the set of features included in at least one chromosome (from S') description and $S'_2 \subseteq Z$ be the set of prototypes corresponding to at least one chomosome found out in the genetic history whose fitness score is higher than a given threshold.

The ascending procedure consists, for each chromosome in S“, in aggregating each of the features in S'_1 (resp. each of the prototypes in S'_2) to the chromosome and selecting the ones that improve the classification results. The process is repeated until no improvement is possible or a maximal number of ascending iterations is reached.
It should be mentioned that the number of features and prototypes to be tested is reasonably small as some features have been discarded by the first phase of the GA, and among the others, only those which are currently part of one of the best chromosomes are used. This remark highlights the complementary roles played by the GA and the local approach.

However, depending on the evolution stage, the cardinalities of S'_1 and S'_2 may be important. In this case, in order to control the ascending procedure computational cost, the number of features or prototypes tested by the procedure is limited. The selected ones are randomly chosen in S'_1 or S'_2 to form $S"_1$ and $S"_2$.

The descending phase is only applied to S". For each chromosome each of the selected features is removed if its removal does not affect the classification results while improving the fitness function.

In order to save time, ascending and descending procedures are carried out periodically within the so called "convergence phase".

Different strategies are likely to yield comparable results. In our implementation, the convergence phase is organized as a sequence of the following operations:

(i) A tuning phase including an ascending procedure followed by a descending one: The preferred options aggregate new prototypes and remove features as the lower the feature dimension space the better the interpretability. This complete mode is quite expensive, it is run with a large period.

(ii) A tuning phase with a descending procedure in only one space: The feature and prototype spaces are alternatively managed.

(iii) A pure GA.

At each genetic operation, the sequence can be indifferently applied to each chromosome independently or to S". We introduce p_{all}, p_{des} and p_{pure} as the probabilities of applying one of the three operations.

It should be noted that during the ascending procedure which aims at increasing the number of active bits in the chromosome, priority is given to increasing $|S_2|$ and then to increasing $|S_1|$. If there is a competition between two candidate chromosomes A (related to $|S_2|$) and B (related to $|S_1|$) giving similar fitness scores, A wins the competition. On the contrary in the descending procedure, the priority is given to decreasing $|S_1|$ to favour this criterion and afterwards to decreasing $|S_2|$.

Finally, when local approaches are applied, in addition to the weights of each operation, four processes to reduce the time are investigated:

(i) Only chromosomes (of S') close to solution are concerned by local approaches.

(ii) Only a fraction of the selected chromosomes is considered at a given step: this acts as a population reduction driven by p_{sol}, the probability to decrease the number of solutions to which local search is applied. $|S"| = \min(p_{sol} * |S|, |S'|)$

(iii) Only a variable subset of chromosomes components is evaluated: this acts as a chromosome reduction. We introduce ($p_{f\text{-}search}$, $p_{p\text{-}search}$) to decrease the number of chromosome components inspected. $S"_1 = \min(p_{f\text{-}search} * |S_1|, |S'_1|)$ and $S"_2 = \min(p_{p\text{-}search} * |S_2|, |S'_2|)$.

(iv) For each stepwise sequence, the number of chromosome modifications is limited (max_{ope}). Space exploration is let to GA mechanisms.

During phase 2, the different levels of probability can be progressively increased (linearly or by steps) to allocate more resources to the local tuning.

4. Results and discussion

The proposed hybrid GA is now applied to various benchmarks and real world data sets. The results are compared with other approaches. The objective of this section is multiple:

- Comparing the GA performances with common editing/selecting and known genetic approaches.
- Analysing its performance to produce competent training sets.
- Analysing the effect of different mechanisms that have been introduced.

4.1 Data sets used

To test the proposed method, trials were conducted based on seven data sets. The following UCI repository datasets [106] were used in tests: Iris (150 patterns, 4 features, two classes), Wisconsin breast cancer (699 patterns, 9 features and two classes), wine (178 patterns, 13 features, 3 classes), Pima indians diabetes (768 patterns, 8 features, two classes), Ionosphere (351 patterns, 34 features, two classes), Gls (214 patterns, 9 features, 6 classes). It is needless to introduce them as widely used by many machine learning algorithms. In addition, a data set called Chem (568 patterns, 166 features, 4 classes) coming from the chemometric field has been selected. 568 compounds was derived from analyses of the chemicals in the fathead minnow acute toxicity database. A detailed description of the biological and chemical test protocols used in the study has been published [107]. Several chemical classes such as organophosphates, alkanes, ethers, alcohols, aldehydes, ketones, esters, amines and other nitrogen compounds, aromatic and sulfur compounds, and several modes of action, such as narcosis, oxidative phosphorylation uncoupling, respiratory inhibition, electrophile/proelectrophile reactivity, acetylcholines-terase (AChe) inhibition, and mechanisms of central nervous system (CNS) exposure are represented in this data set. A ninety-six-hour lethal concentration killing 50% of the fathead minnow population (96h-LC50) was used to characterize toxicity. Four toxicity classes were generated according to the intervals established by the European Community legislation [108].
Finally, the datasets are composed of various dimensional spaces from 3 to 166 and various degrees of complexity regarding classes overlapping. Incomplete fields are replaced by the average of remaining one.

4.2 Presentation of algorithms and genetic parameters

The most natural way to reach the multiobjective discussed in this paper is to manage the objectives separatively by applying feature selection and editing approaches. Selection approaches aim to reduce the feature number without losing classification and editing approaches enable to discard bad examples. Their combination called (SE) is likely to produce a reduced set. Different famous edition approaches are implemented: Wilson, Repeated Wilson and City edition. They are combined with three selection schemes: Forward, Backward and random mutation hill climbing. Concerning the mutation hill climbing, the strategy is the one described by Skalak[101].

Four basic and very known genetic strategies are also implemented: a classic elitism scheme, determinist crowding, the restricted tournament selection and the multi hill climbing algorithm. All are combined with the proposed fitness function. Concerning the elitism strategy implemented, the subset of children created after genetic operations compete with their parents. The best of the whole set, parents and children, survive in the next generation.

There is no restriction concerning the choice of the classifier C. However, we have restricted the tests to only one classifier to focus on the genetic part. The 1NN (nearest neighbor) algorithm has been selected for its recurrent attractivity, simplicity and because no assumption on class shape is needed.

Finally, the different approaches either genetic or not are:
1. FW: Forward Selection combined with Wilson Edition
2. FRW: Forward Selection combined with Repeated Wilson Edition
3. FC: Forward Selection combined with City Edition
4. BW: Forward Selection combined with Wilson Edition
5. BRW: Backward Selection combined with Repeated Wilson Edition
6. BC: Backward Selection combined with City Edition
7. MHCW: Multi Hill Climbing Algorithm combined with Wilson Edition
8. MHCRW: Multi Hill Climbing Algorithm combined with Repeated Wilson Edition
9. MHCC: Multi Hill Climbing Algorithm combined with City Edition
10. EA: Elitist approach
11. DC: Determinist Crowding
12. RTS: Restricted Tournament Selection
13. MHCM: Multi Hill Climbing Algorithm for Multi objective
14. HG: Our hybrid approach

The same and very common genetic parameters have been chosen whatever the database. the main genetic features are listed below:

- Number of chromosomes: 100
- Initial population: random bit generation with prob(0) = 0.5 and prob(1) = 0.5
- Crossover, mutation, selection and rejection probabilities: $p_c = 0.5$, $p_m = 0.05$, $p_s = 0.3$, $p_r = 0.05$.
- Terminal number of generations: 500
- Fitness function (penalty terms and validity): $\Delta_p = 0.4$, $\mu_p = 0.95$, $\Delta_{f1}=1$, $\Delta_{f2}=15$, $\mu_f = 0.2$, $freq_{rep} = 0.1$.

For the hybrid GA, these specific parameters are used
- Initial population: $a_1 = 0.1$, $a_2 = 0.9$, $a_3 = 20$
- Diversity index: $n_f = 1$, $n_p=0.1*p$, $th_{min}=0.65*s$
- Genetic life: terminal number of generations=200 and local tuning starts not after 100
- Features removal: feature histogram is generated with chromosomes having fitness score more than $min_{fitness} = 0.3$. Threshold frequency to remove: $min_{freq} = 1\%$.
- Fitness score threshold to apply local optimization: $min_{local} = 0.7$ for two classes and 0.5 for three classes and more.
- Maximum number of chromosomes selected for the stepwise procedures: $p_{sol} = 0.25$
- Distribution of local procedures: one scheme with ascending/descending ($p_{all} = 20\%$), descending ($p_{des} = 50\%$) and pure GA ($p_{pure} = 30\%$).
- Maximum number of ascending/descending iterations for one sequence: $max_{ope}=10$
- Other local parameters: $p_{p\text{-}search} = 0.3$, $p_{f\text{-}search} = 0.5$.

4.3 Comparison with editing/selecting approaches:

This section aims to analyse the performance of (HG) with a combination of selection and editing approaches applied separately.

We restrict the experiments to the following scheme:

- Random generation of 10 tests. Each test is composed of training and test files approximately with 80% and 20%, respectively. The ten training files are centered and normalized and the corresponding test files are obtained by applying the respective transformation.
- For each SE, apply a selection scheme followed by an edition one to reduce both spaces separatively.
- Application of the hybrid GA to manage the multiobjective directly.

By averaging the results obtained for the 10 databases, we have a single triplet representing respectively $C(S_2)$, $|S_2|$ and $|S_1|$ for each database, each SE and HG. The results obtained are summarized in Table 1. They show the plus of the simultaneous dual selection in the case of a database containing a great amount of noise. If we consider the results obtained for databases Iris, Breast and Wine where there is a very little noise, the triplets provided by the hybrid approach are roughly comparable to those obtained by selection/edition. However, for all the cases the hybrid approach is competitive, which is not the case for each SE which varies with the experiments.
Differences between the results are much more significant with the databases containing noise. The case of the chemometric database is very clear in this respect: firstly, the multi hill climbing and backward selection are unable to provide interesting feature space in terms of dimensionality. Whatever the SE configuration, the results obtained are far from those of HG ($C(S_2)$=0.97 $|S_2|$=277.7 $|S_1|$=2.3). The Forward selection approach provides a small feature number ($|S_1|$=2.4) but whatever the editing scheme $|S_2|<160$ which is very low compared to $|S_2|$=277.7 obtained via the genetic approach. Tuning the progressive coefficient for the backward approach and changing the initial setting for the hill climbing one did not improve the triplet relevance. This means that in the event of noise and probably irrelevant data, a single optimisation is better.

Competence analysis to provide training sets

The double reduction applied on the feature and instances patterns aims to discard irrelevant features and select “good“ instance patterns. Therefore, even if it is beyond the scope of this paper, this process is likely to produce reduced sets which are “competent“ to form a training set for a classification learning algorithm, especially in the presence of noise. We therefore carried out an experiment to empirically measure the quality of the so-called “filtering“ for classification compared to other well-known schemes.
We have analysed the classification errors obtained with the test files for the different SE configurations and the hybrid approach (Fig. 5). In the genetic algorithm, each chromosome of the final population defines a classifier. The chromosome selected to design the reference set was the one presenting the best classification score among the set of chromosomes. On this basis, test results are satisfactory but not optimal.
However, there is always one chromosome among the final pop_a of the hybrid approach which outperforms all the SEs. This underlines the potential of the hybrid approach to provide good and especially diverse chromosomes. We did not find general rules linking the chromosome performances and its generalization ability for classification. Noisy and irrelevant patterns are sometimes difficult to distinguish. Only a cross validation approach demonstrates stability and consistence: divide the training set in two subsets, one dedicated to the search of potential solutions and the another one for testing and selecting the best chromosome regarding the classification score.

4.4 Comparison with other genetic approaches:

This section aims to analyse the performance of HG with the four GAs.
Different experiments involving the same number of chromosomes and genetic iterations have been carried out. We tested various genetic algorithm versions and four different sets of penalty terms have been considered to assess performances. Let recall that Δ_p and μ_p stand for λ_p as Δ_{f1}, Δ_{f2} and μ_f stand for λ_f.
As the result of the GA procedure is a chromosome population, comparing genetic approaches comes to compare the corresponding populations. Unfortunately, no metric is available to achieve this goal within a multi objective framework.
The three considered objectives are the classification rate, which ranges into the unit interval, the proportion of selected patterns, |S2|/p, and the number of selected features, f. A given pair of chromosomes are considered of comparable performance with respect to one of these objectives if the difference between their scores is less than a predefined threshold. The reported tests use the following values: ε_c=0.01 for classification, ε_p=0.02*p for pattern selection, ε_f=min(0.1*f, 2) for feature selection.

Chromosome comparison yields a single value, v:

- v = 1, if one of the chromosomes gives a better result for at least one of the objectives, and comparable results for the others ;
- v = 0, if each of the chromosomes gives better results than the other in at least one objective;
- v = 0.5, if the performances are comparable for all the objectives.

Population comparison is done in a similar way. A population X is considered better than a population Y if there exists a chromosome in X whose comparison with all the elements of Y yields 1. In this case, the comparison assigns a 1 to X and a 0 to Y. Otherwise, the populations are said comparable, and both are assigned a 0.5 value.
The results of the 10 experiments and the four configuration weighs are then averaged. Table 2 shows this final index for all the studied data sets and the comparison of our hybrid algorithm with other genetic approaches.
Let us underline none of the reported values is less than 0.5, meaning the proposed algorithm never gives poorer results than any of the compared GA. Moreover, the more difficult the data set to manage, the higher the index. This is especially true for the chemometrics data.
We have voluntarily restricted the comparisons to very simple criteria in order to demonstrate the efficiency of our hybrid approach. The problem of multiple optimisation and criteria aggregation are entire topics [109-110] that are continually being tackled in the literature. Even if further considerations are beyond the scope of this paper, they are worth considering in depth in other specific studies.

4.5 Discussion about efficiency and time reduction

The role of hybridization with local approaches in genetic development is obvious, but getting a dual procedure of power is more problematic. In term of efficiency, local approaches are really relevant if used in appropriate contexts. To be practical, the hybridization needs to be monitored. A balance between a complete use (likely generating a better efficiency while being unpractical) and a pure GA (producing limited performances) is necessary. The different heuristics have been implemented to make the hybridized method practical. As the approach is modular, different versions can be derived.
Mechanisms to find preservation of both elitism and diversity, at different levels, help getting the appropriate context. Hybridization mechanisms help to optimize the local searchs and incorporate parameters of probability in order to control resources. So, according to the space

dimensionnalities, there are the necessary elements to estimate the processing time and therefore to make the method practical.
The iris database presents no interest in terms of performance analysis but is useful to test different configurations. We have evaluated the following versions:

- V_0: Basic and pure elitism GA without any mechanism
- V_1: $V_0 \cup$ chromosomes initialisation $\cup$ archive population
- V_2: $V_1 \cup$ breaking process
- V_3: $V_1 \cup$ RTS in phase 1
- V_4: $V_2 \cup$ optimized local approaches
- V_5: $V_3 \cup$ optimized local approaches
- V_6: $V_4 \cup V_5$
- V_7: $V_6 \cup$ full local approaches

All give good results and the differences stem for the diversity. The V1, V2 and V3 versions present more diversity than the V0 version which produces only one solution (Fig.6). V1 expresses the presence of the archive population. (V2, V3) respectively denote the plus of diversity via the breaking process and the RTS scheme: Fig. 7 illustrates that the cost of using full local approaches is higher than the gain. For similar performances, spent time for V7 is 20 times higher than for V6. Fig. 8 and 9 show the effects of the different mechanisms when applying V6: The presence of the breaking process is visible in the pop_c evolution and local tuning is particularly effective (around generation 35) on |S2| evolution.

5. Conclusion

Automated database exploitation remains a challenge for pharmaceutical companies and particularly for the selection of new compounds with potent specific biological properties. The solutions that systematically exploit large and complete compound libraries are powerful but costly and highly time-consuming. In contrast, many data reduction techniques, such as unsupervised projective approaches (for example factorial analysis [111] or Kohonen map [112]) are comfortable but intrinsically limited in their exploitation. The method proposed here is an intermediate reduction tool. It can provide sub databases where the patterns are projected in a reduced feature space: this twofold reduction (feature/pattern) makes interpretability easier (Fig. 10). The approach is supervised and has the potential to create competent training sets.
Our solution is genetic based, modular and hybrid. We believe, in keeping with many other scientists interested in applying genetic algorithms to real contexts, that pure genetic approaches are still difficult to apply. It is particularly difficult to maintain qualities of a genetic population namely both diversity and elitism.
Our whole process is then optimized by dividing the algorithm into two self-controlled phases with dedicated objectives in which several mechanisms are incorporated. These mechanisms act in different and compensating ways to reach the same objective.
Setting the parameters to ensure a trade-off between these two tasks within a reasonable time is difficult. Exploratory strategies require a lot of resources to give a good solution in high dimensional problems, while elitism based strategy may ignore interesting parts of the space. Our modular approach integrates these constraints.
As proved by the results and comparison with other approaches, this algorithm is likely to give satisfactory results within a reasonable time when dealing with medium size data sets. Although the method has been developed for chemometric applications, involving many features (several hundreds) and patterns (several thousands are possible), applications on other databases are possible. Coupling the approach with clustering or stratification techniques will make the method better for managing very large databases (more than ten thousand patterns) that can be found in the field of data mining.

This hybrid approach can be applied to other problems. For instance, it seems rather appropriate for designing optimal nearest classifiers in the presence of noise and irrelevant features. The majority of the methods available in the literature disregard the feature selection phase and are based on heuristics that work well provided the amount of noise is small. Therefore, in a context of many features and noise, it constitutes an interesting alternative.

REFERENCES

1. Fauchère L.J, Bouting J.A, Henlin J.M, Kucharczyk N and Ortuno J.C (1998) Combinatorial chemistry for the generation of molecular diversity and the discovery of bioactive lead. Chem. Intell. Lab. Syst 43:43-68
2. Borman S. Reducing time to drug discovery (1999) Recent advances in solid phase synthesis and high-throughpout screening suggest combinatorial chemistry is coming of age. *CENEAR* 77(10): 33-48
3. Guyon I, Elisseeff A (2003) An Introduction to Variable and Descriptor Selection. Journal of Machine Learning Research 3:1157-1182
4. Ng. A.Y (1998) Descriptor selection: learning with exponentially many irrelevant descriptors as training examples. 15th International Conference on Machine Learning, San Francisco, CA, 404-412
5. Dasarathy B.V. (1990) Nearest Neighbor (NN) Norms: NN Pattern Recognition Techniques. IEEE Computer Society Press, Los Alamitos, California
6. Dasarathy B.V. (1994) Minimal consistent set (MSC) identification for optimal nearest neighbor decision system design. IEEE Transaction on System Man Cybernetics 24: 511-517
7. Ramaswamy S., Rastogi R. and Shim K. (2000) Efficient algorithms for mining outliers from large data sets. Proc. of the ACM SIGMOD Conference 427-438
8. Dasarathy B.V., Sanchez J.S. and Townsend S. (2003) Nearest Neighbour Editing and Condensing Tools-Synergy Exploitation. Pattern Analysis & Applications 3: 19-30
9. Kuncheva L.I, Jain L.C. (1999) Nearest Neighbor Classifier: Simultaneous Editing and Descriptor Selection. Pattern Recognition Letters 20 (11-13): 1149-1156
10. Ho S.Y, Chang X.I. (1999) An efficient generalized multiobjective evolutionary algorithm (1999) Proceedings of the Genetic and Evolutionary Computation Conference, Morgan Kaufmann Publishers, Los Altos, CA, 871-878
11. Davis T.E., Principe J.C. (1991) A simulated annealing-like converge theory for the simple genetic algorithm, in ICGA: 174-181
12. Ye. T., Kaur H.T. and Kalyanaraman S. (2003) A recursive Random Search Algorithm for Large Scale Network Parameter Configuration, SIGMETRICS 2003, San Diego, California
13. Glover F. (1989) Tabu Search. ORSA Journal of Computing 1(3): 190-206
14. Boyan J., Moore A. (2000) Learning Evalutation functions to improve optimisation by local search. Journal of Machine Learning Research, 1:77-112
15. Goldberg D.E. (1989) Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Boston, MA.
16. Forrest S., Mitchell M. (1993) What Makes a Problem Hard for a Genetic Algorithm? Some Anomalous Results and Their Explanation. Machine Learning 13: 285-319
17. Glicman M.R, Sycara K. (2000) Reasons for Premature Convergence of Self-Adapting Mutation Rates. Proc. of the Congress on Evolutionary Computation, San Diego, USA, 1: 62-69
18. Schaffer J., Caruana R., Eshelman L. and Das R. (1989) A study of control parameters affecting online performance of genetic algorithms for function optimization, Proc. Of 3rd International Conference on Genetic Algorithm, Morgan Kaufman, 51-60
19. Costa J, Tavares R and Rosa A. (1999) An experimental study on dynamic random variation of population size. Proc. of IEEE Systems, Man and Cybernetics Conf., Tokyo, Japan, 6: 607-612
20. Tuson A, Ross P. (1998) Adapting Operator Settings. Genetic Algorithms in Evolutionary Computation 6(2): 161-184
21. Pelikan M, Lobo F.G. (2000) Parameter-less Genetic Algorithm: A Worst-case Time and Space Complexity Analysis. Proc. of the Genetic and Evolutionary Computation Conference, San Francisco, CA, 370-377

22. Eiben A.E, Marchiori E and Valko V.A. (2004) Evolutionary Algorithms with on-the-fly Population Size Adjustment. Proc. of the 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII), Birmingham, UK, 41-50
23. Dash M., Liu H. (1997) Feature Selection for Classification. Intelligent Data Analysis 1: 131-156
24. Piramuthu S. (2004) Evaluating Feature selection methods for learning in data mining application. European Journal of Operational Research 156: 483-494
25. Kohavi R and John G. (1997) Wrappers for feature selection. Artificial Intelligence, 273-324.
26. Stracuzzi D.J., Utgoff P.E (2004) Randomized Variable Elimination. Journal of Machine Learning Research 5: 1331-1362
27.Kira K., Rendell LA (1992) The Feature Selection Problem: Traditional Methods and a New Algorithm. Proc. of the Ninth National Conference on Artificial Intelligence 129-134
28. Almuallim H., Diettrerich T.G. (1994) Learning boolean concepts in the presence of many irrelevant feautres. Artificial Intelligence 69(1-2): 279-305
29. Ratanamahatan A. and Gunopulos D. (2003) Feature selection for the naive bayesian classifier using decision trees. Applied Artificial Intelligence 17: 475-487
30. Shalkoff R. (1992) Pattern Recognition statistical, structural and neural approaches. John Wiley and Sons, Singapore
30. Devijver P.A. and Kittler J. (1982) Pattern Recognition: A statistical approach, Prentice Hall
31. Caruana R. and Freitag D (1994) Greedy attibute selection. Proceedings of Eleventh International Conference on Machine Learning, Morgan Kaufman, New Jersey 28-36
32. Shalak D.B (1994) Prototype and Feature Selection by Sampling and Random Mutation Hill Climbing Algorithms. Proc. of the Eleventh International Conference on Machine Learning, New Brunswick, Morgan Kaufman 293-301
33. Collins R.J, Jeferson D.R. (1991) Selection in massively parallel genetic algorithms. Proc. of the fourth International Conference On Genetic Algorithms, San Diego, CA, 244-248.
34. Jain A.K. and Zongker D. (1997) Feature Selection: Evaluation, Application, and Small Sample Performance. IEEE Trans. On Pattern Analysis and Machine Intelligence 19 (2): 153-158
35. Zongker D and Jain A.K. (2004) Algorithms for Feature Selection: An Evaluation. IEEE Trans. On Pattern Analysis and Machine Intelligence 26 (9): 1105-1113
36. Zhang H., Sun G. (2002) Optimal reference subset selection for nearest neighbor classification by tabu search. Pattern Recognition 35: 1481-1490
37. Brighton H., Mellish C (2002) Advances in Instance Selection for Instance-Based Learning Algorithms. Data Mining and Knowledge Discovery 6: 153-172
38. Dasarathy B.V (1994) Minimal consistent subset (MCS) identification for optimal nearest neighbor decision systems design. IEEE Transactions on Systems, Man and Cybernetics 24: 511-517
39. Hart P.E. (1968) The condensed nearest neighbor rule. IEEE Trans. on Inf. Theory 16: 515-516
40. Gates G.W. (1972) The reduced nearest neighbor rule. IEEE Trans. Inf. Theory 18 (3) 431-433
41. Swonger C.W. (1972) Sample set condensation for a condensed nearest neighbour decision rule for pattern recognition. S. Watanabe (Ed.), Orlando, FA: Academic Press 511-519
42. Aha D., Kibler D. Albert M.K. (1991) Instance-Based Learning Algorithms. Machine Learning 6: 37-66
43.Wilson D.R, Martinez T.R (2000) Reduction Techniques for Instance-Based Learning Algorithms. Machine Learning 38(3): 257-286

44. Kuncheva L.I. (1997) Fitness functions in Editing k-NN reference set by genetic algorithms. Pattern Recognition 30(6): 1041-1049
45. L. Guo, D.S Huang and W.Zhao (2003) Combining genetic optimization with hybrid learning algorithm for radial basis function neural networks. Electronics Letters OnLine 39(22).
46. Bezdek J.C. and Kuncheva L.I (2000) Nearest prototype classifier designs: An experimental study. International Journal of Intelligent Systems 16(12): 1445-1473
47. Bezdek J.C. and Kuncheva L.I (2000) Some Notes on Twenty One (21) Nearest Prototype Classifiers. SSPR&SPR, F.J. Ferri et al (Eds), Springer-Verlag Berlin Heidelberg 1-16
48. Kim S.W and Oommen B.J (2003) A brief taxonomy and ranking of creative prototype reduction schemes. Pattern Anal Applic 6: 232-244
49. Shekhar S., Lu C.T. and Zhang P (2003) A Unified Approach to Detecting Spatial Outliers. GeoInformatica 7(2): 139-166
50. Knorr E.M., Ng R.T. and Tucakov V. (2000) Distance-based outliers: algorithms and applications. VLDB Journal 8(3-4): 237-253
51. Shekhar S., Lu C.T. and Zhang P. (2002) Detecting Graph-Based Spatial Outliers
52. Lun C-T, Chen and Kou Y. (2003) Algorithms for Spatial Outliers Detection. Proc. of the Third IEEE International Conference on Data Mining
53. Aguilar J.C, Riquelme J.C and Toro M. (2001) Data Set Editing by Ordered Projection. Intelligent Data Analysis 5(5): 1-13
54. Quinlan J., C4.5 programs for machine learning, San Francisco, Morgan Kaufman 1992.
55. Kim S.W and Oommen B.J. (2003) Enhancing Prototype reduction schemes with recursion: a method applicable for “Large“ data sets, IEEE Trans. On Syst. Man and Cyber. 34(3): Part B
56. Wilson D.L. (1972) Asymptotic properties of nearest neighbor rules using edited data. IEEE Trans. on System Man Cybernetics 2: 408-421.
57. Francesco J.F, Jesus V. and Vidal A (1999) Considerations About Sample-Size Sensitivity of a Family of Edited Nearest-Neighbor Rules. IEEE Trans. On Syst. Man and Cyber. 29(4) Part B
58. Devijver P and Kittler J (1980) On the Edited Nearest Neighbor Rule. IEEE Pattern Recognition 1: 72-80
59. Garfield E. (1979) Citation Indexing: its Theory and Application in Science, Technology and Humanities. John Wiley and Sons, New York
60. Barandela R. and GascaE (2000) Decontamination of training samples for supervised pattern recognition methods. Ferri F.J, Inesta Quereda J.M., Amin A., Paudil P (Eds): Lecture Notes in Computer Science, Springer, Berlin 1876: 621-630
61. Jiang Y and Zhou Z.H. () Editing Training Data for kNN Classifiers with Neural Network Ensemble
62. Eiben A.E, Hinterding R and Michalewicz Z. (1999) Parameter control in evolutionary algorithms. IEEE Trans. on Evolutionary Computation 3(2): 124-141
63. Tuson A, Ross P. Adapting Operator Settings (1998) Genetic Algorithms in Evolutionary Computation; 6(2): 161-184
64. Costa J, Tavares R and Rosa A. (1999) An experimental study on dynamic random variation of population size. Proc. of IEEE Systems, Man and Cybernetics Conf., Tokyo, Japan 6: 607-612
65. Arabas J, Michalewicz Z, and Mulawka J. (1994) A genetic algorithm with varying population size. Proc. of the First IEEE Conference on Evolutionary Computation, Piscataway, NJ 73–78
66. Deb K., Goldberg D.E. (1989) An investigation of niche and species formation in genetic function optimisation. Proc. of the third Int. Conf. on Genetic Algorithms, Schaffer J.D., Ed San Mateo, CA, Morgan Kaufmann 42-50

67. Beasley D., Bull D.R and Martin R.R (1993) A Sequential Niche Technique for Multimodal Function Optimization. Evolutionary Computation 1(2): 101-125
68. Goldberg D.E. and Richardson J. (1987) Genetic Algorithms with sharing for multimodal function optimisation. Proc. of the 2nd Int. Conf. on Genetic Algorithms, J.J Grefensette, Ed. Hillsdale 41-49
69. Deb K. (1989) Genetic Algorithm in multimodal function optimisation. MS thesis, TCGA Report n°89002 University of Alabama
70. Miller B.L. and Shaw M.J. (1996) Genetic Algorithms with Dynamic Sharing for Multimodal Function Optimization. Proc. of Int. Conf. Evolutionary Computation, Piscataway 786-791
71. Sareni B., Krahenbuhl L. (1998) Fitness Sharing and Niching Methods Revisited. IEEE Trans. On Evolutionary Computation 2(3)
72. Youang B. (2002) Deterministic Crowding, Recombination and Self-Similarity. Proc. IEEE
73. Li J.P, Balazs M.E, Parks G.T.and Clarkson P.J (2002) A Species Conserving Genetic Algorithm for Multimodal Function Optimization. Evolutionary Computation 10(3): 207-234
74. DeJong K.A. (1975) Analysis of the behavior of a class of Genetic Adaptive Systems. PhD thesis, University of Michigan, August
75. Mahfoud S.W. (1992) Crowding and preselection revisited. 2nd Conf. Parallel problem Solving from Nature (PPSN'92), Brussels, Belgium, 2: 27-36
76. Harik G. (1995) Finding Multimodal Solutions Using Restricted Tournament Selection. Proc.of 6th Int. Conf. On Genetic Algorithms, Eshelman L.J. Ed. San Mateo, CA: Morgan Kaufman 24-31
77. Deb K., Pratap A., Agarwal S and Meyarivan T. () A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II, KanGal (Kanpur Genetic Algorithm Laboratory) Report No. 200001
78. Wiese K. and Goodwin S.D. (1998) Keep-best reproduction: a selection strategy for genetic algorithms. Proc. Of the 1998 symposium on Applied Computing 343-348
79. Matsui K. (1999) New selection method to improve the population diversity in genetic algorithms Systems, Man and Cybernetics. IEEE International Conference 1: 625-630
80. Lozano M., Herrera F and Cano J.R. (2004) Replacement Strategies to Preserve Useful Diversity in Steady-State Genetic Algorithms, Elsevier Science (in press)
81. Knowles J.D. (2002) Local Search and Hybrid Evolutionary Algorithms for Pareto Optimization. PhD Thesis, University of Reading, January
82. Zitzler E, Teich J. and Bhattacharyya (2000) Optimizing the Efficiency of Parameterized Local Search Within Global Search: A Preliminary Study. Proc. of the Congress on Evolutionary Computation, San Diego, California 365-372
83. P. Moscato. (1999) Memetic algorithms: A short introduction, D. Corne, F.Glover, and M.Dorigo (eds.), New Ideas in Optimization, McGraw-Hill, Maidenhead, 219-234
84. W.E. Hart. (1994) adaptative global optimization with local search. PhD. Thesis, University of California, San Diego
85. M. W. S. Land. (1998) Evolutionary algorithms with local search for combinatorial optimization. PhD Thesis, University of California, San Diego
86. Ros F, Pintore M and Chretien J.R. (2002) Molecular description selection combining genetic algorithms and fuzzy logic: application to database mining procedures. Journal of Chem. Int. Lab. Systems 63: 15-22
87. Leardi R. and Gonzalez A.L. (1998) Genetic algorithms applied to feature selection in PLS regression: how and when to use them. Chem. and Intelligent Laboratory Systems 41(2): 195-207
88. Merz P. (2000) Memetic Algorithms for Combinatorial Optimization Problems: Fitness Landscapes and Effective Search Strategies. PhD thesis, University of Siegen

89. Merz P. and Freisleben (1999) A comparison of memetic algorithms, tabu search and ant colonies for the quadratic assignment problem. Proc. of the Int. Congress of Evolutionary Computation, Washington DC, USA
90. Krasnogor N. (2002) Studies on the Theory and Design Space of Memetic Algorithms, Thesis University of the West of England, Bristol
91. Zitzler E., Laumanns M and Bleuler S. () A Tutorial on Evolutionary Multiobjective Optimization
92. Goldberg D.E (1989) Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, MA: Addison-Wesley
93. Schaffer J.D. (1985) Multiple objective optimization with vector evaluated genetic algorithms. Proc. of the 1th Int. Conference on Genetic Algorithms 93-100
94. Horn J., Nafpliotis N. And Goldberg D.E (1994) A Niched Pareto Genetic Algorithm for Multiobjective Optimization. Proc.of the first IEEE Conference on Evolutionary Computation 1: 82-87
95. Laumanns M., Thiele L., Deb K. And Zitzler E. (2000) On the Convergence and Diversity-Preservation Properties of Multi-Objective Evolutionary Algorithms. Evolutionary Computation 8(2): 149-172
96. Mitsuo G. And Runwei C (1997) Genetic Algorithms and Engineering Design. John Wiley and Sons, Inc., NewYork
97. Coello C.A, Van Veldhuizen and Lamont G.B. (2002) Evolutionary Algorithms for Solving Multi-Objective Problems, Kluwer, New York
98. Zitzler E. (1999) Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. PhD Thesis, Shaker Verlag, Aachen, Germany
99. Coello (1999) A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. Knowledge and Information Systems 1(3) 129-156
100. Tamaki H., Mori M., Araki M. and Ogai H. (1995) Multicriteria optimization by genetic algorithms: a case of scheduling in hot rolling process. Proc of the 3rd APORS 374-381
101. Skalak D.B. (1997) Prototype selection for composite nearest neighbor classifiers, Phd Thesis, University of Massachuset Amherst
102. Kuncheva L.I, Jain L.C. (1999) Nearest Neighbor Classifier: Simultaneous Editing and Descriptor Selection. Pattern Recognition Letters 20 (11-13): 1149-1156
103. Ho S-H, Lui C-C, Liu S. (2002) Design of an optimal nearest neighbor classifier using an intelligent genetic algorithm. Pattern Recognition Letters 23: 1495-1503
104. Cano J.R, Herrera F., Lozano (2003) Using evolutionary algorithms as instance selection for data reduction in kdd: an experimental study. IEEE Transactions on Evolutionary Computation 7(6): 193-208
105. Chen J.H, Chen H.M., Ho S.Y (2005) Design of nearest neighbor classifiers: multi-objective approach. Int. Journal of Approximate Reasoning (in press)
106. Blake C., Keogh E., Merz C.J. (1998) UCI repository of machine learning databases [http://www.ics.uci.edi/~mlearn/MLRepository.html], Departement of Information and Computer Science, University of California, Irvine, CA
107. Geiger D.L., Brooke L.T. and Call D.J. (Eds) (1990) Acute Toxicities of Organic Chemicals to Fathead Minnows (Pimephales promelas), Center for Lake Superior Environmental Studies, University of Wisconsin, Superior (USA).
108. Directive 92/32/ECC (1992), the seventh amendment to directive 67/548/ECC, OJL 154 of 5.VI.92, p1.
109. Knowles J.D., Corne D.W (2000) Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. Evolutionary Computation 8(2): 149-172
110. Jacquet-Lagrèze E. (1990) Interactive assessment of preferences using holistic judgements : the PREFCALC system. C.A. Bana e Costa (ed), Readings in Multiple Criteria Decision Aid, Springer-Verlag 336-350

111. Blayo F, Demartines P (1991) Data analysis: How to compare Kohonen neural networks to others techniques? International Workshop in Artificial Neural Networks (IWANN 1991), Barcelona, Spain, June 1993, Lectures Notes on Computer Science, Springer-Verlag, 469-476
112. Kireev D, Bernard D., Chretien J.R., Ros F. (1998) Application of Kohonen Neural Networks in Classification of Biologically Active Compounds. SAR and QSAR in Environmental Research 8: 93-107

Table 1. Comparison with selection/edition approaches. For each base, the first row is the number of selected feature, the second $C(S_2)$ and the third $|S_2|$. All the results are the average of 10 tests.

	FW	FRW	FC	BW	BRW	BC	MHCW	MHCRW	MHCC	HG
Iris		1.8			1.4			1.7		1.5
	0.95	1	1	0.95	1	0.99	0.96	1	0.99	0.99
	112.1	111.8	92.2	111.8	110.7	92.2	114.8	114.2	95.4	118
Breast		1.5			6.1			5		3.3
	0.94	1	1	0.96	1	1	0.96	1	0.99	0.99
	507.5	507.3	408.3	533.2	532.2	444.3	534.2	533.2	427.3	553
Wine		3.2			12			9		3.7
	0.92	1	0.99	0.98	1	0.97	0.99	0.97	1	0.99
	129	127.8	85.9	135.8	135.3	110	137.3	136.9	120.8	136.4
Pim		1.3			6.5			3		2.7
	0.71	1	0.99	0.69	1	0.98	0.7	1	0.98	0.94
	354.8	352.8	236.3	358.9	342.7	227.1	353.7	342.7	231.2	465.2
Ionosphere		2.4			23.2			15		3.2
	0.88	1	0.99	0.91	1	0.99	0.91	1	0.99	0.99
	219.8	218	185.3	205.6	201.2	189.2	229.8	223.6	196.4	260.2
Gls		4.5			6.5			6		3.4
	0.79	1	0.99	0.79	1	0.99	0.79	1	0.98	0.98
	109.8	106	65.7	107.5	104	68.5	112.7	109.1	71	132.5
Chem		2.8			68			78		2.3
	0.6	1	0.99	0.59	1	0.97	0.6	1	0.96	0.97
	154.5	146.9	86.6	156.2	137.0	81.8	163.1	145.1	85	277.7

Table 2. Comparison results with other genetic approaches to generate editing bases. Each coefficient presents the average of 10 comparisons, each comparison giving 1, 0.5 or 0.

	EA	DC	RTS	MHCM
Iris	0.5	0.55	0.5	0.8
Breast	0.55	0.525	0.512	0.65
Wine	0.613	0.537	0.537	0.9
Pim	0.587	0.587	0.575	0.75
Ionosphere	1	0.962	0.987	1
Gls	0.787	0.612	0.587	0.5
Chem	0.962	1	0.987	1

Fig 1 General schematic of the hybrid GA.

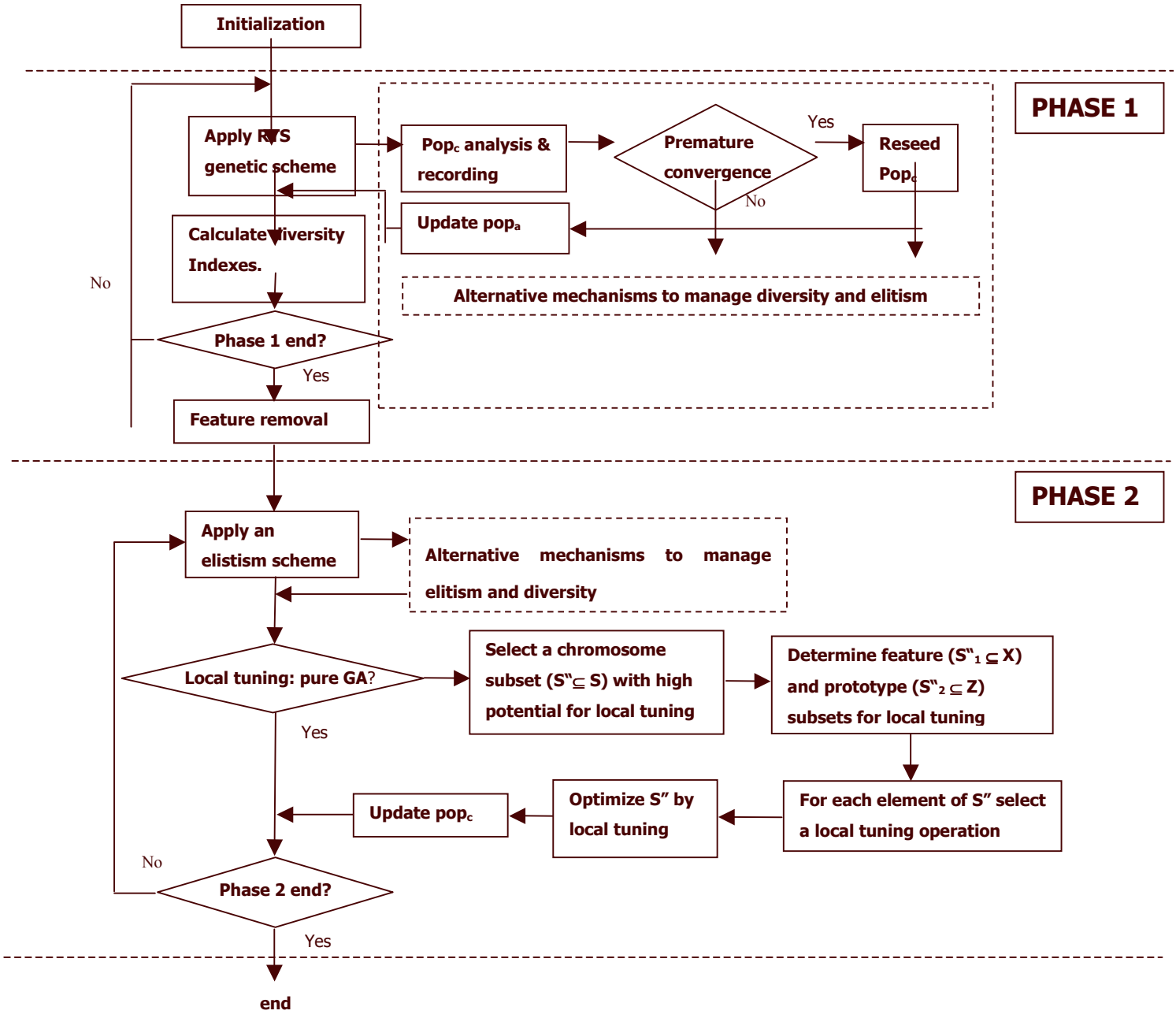

Fig 2 Variation of λ_p with μ_p and Δ_p

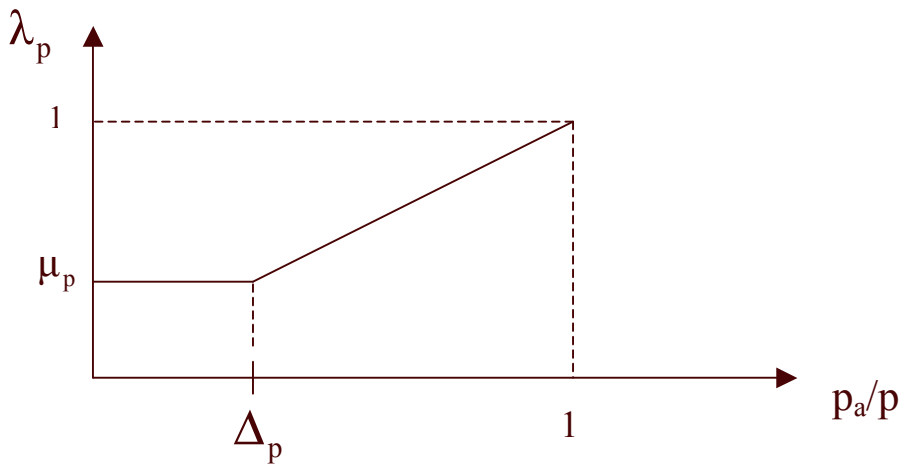

Fig 3 Variation of λ_f with μ_f and $(\Delta_{f1}, \Delta_{f2})$

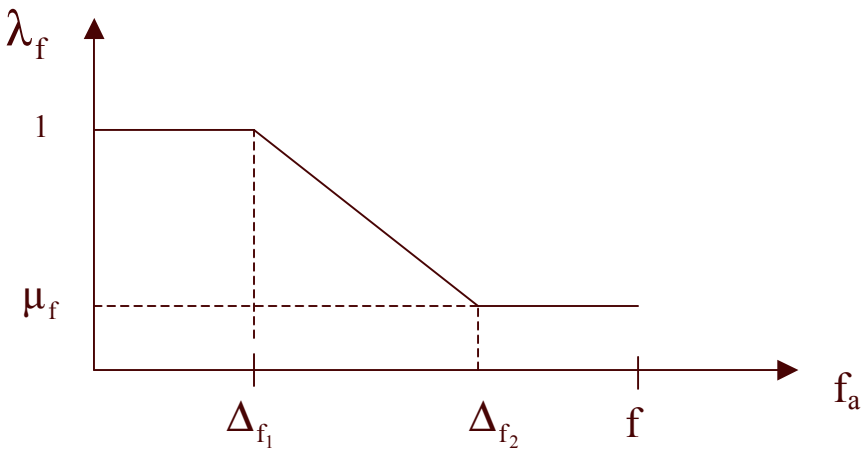

Fig 4 Elitism approach implemented

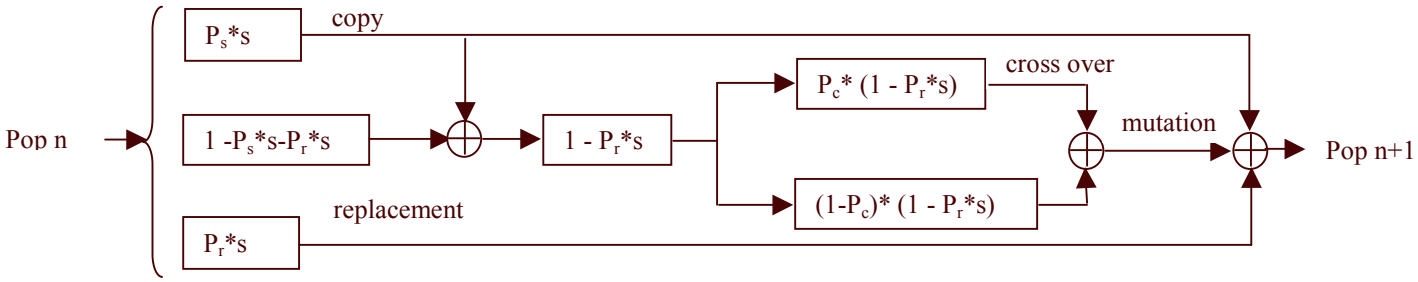

Fig 5 Competent analysis of the training set generated. HG/Bestpopa represents the best chromosome on the pop_a population, HG/BestClass the one providing the best classification score and HG/Cross the one obtained by dividing the training set into subsets.

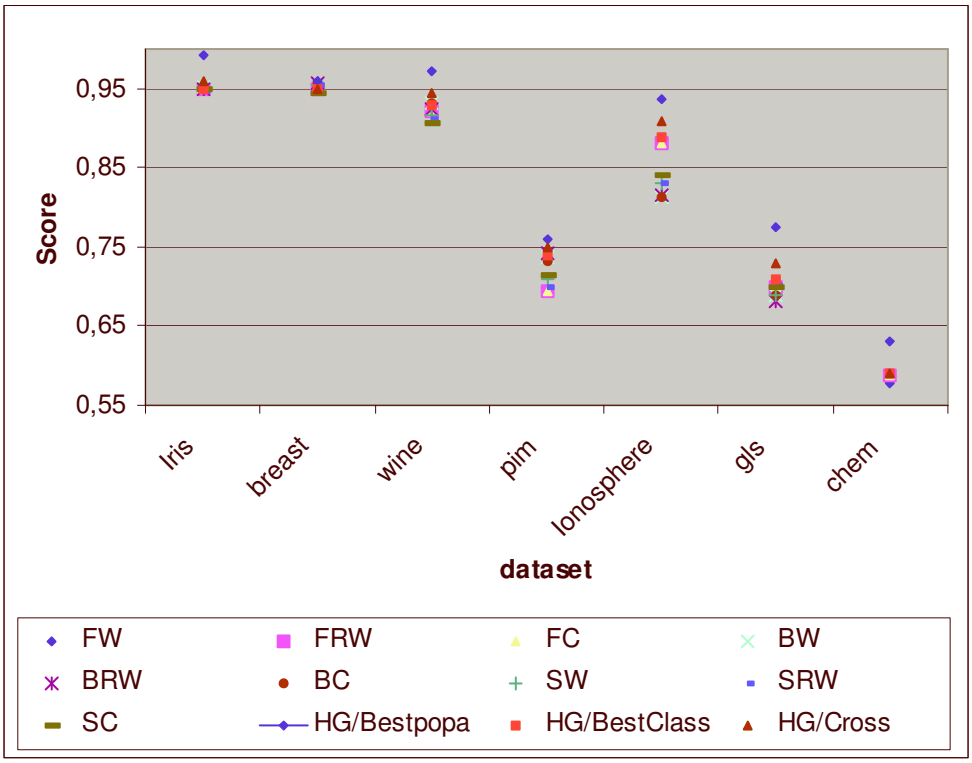

Fig 6 Population fitness score for 7 different versions

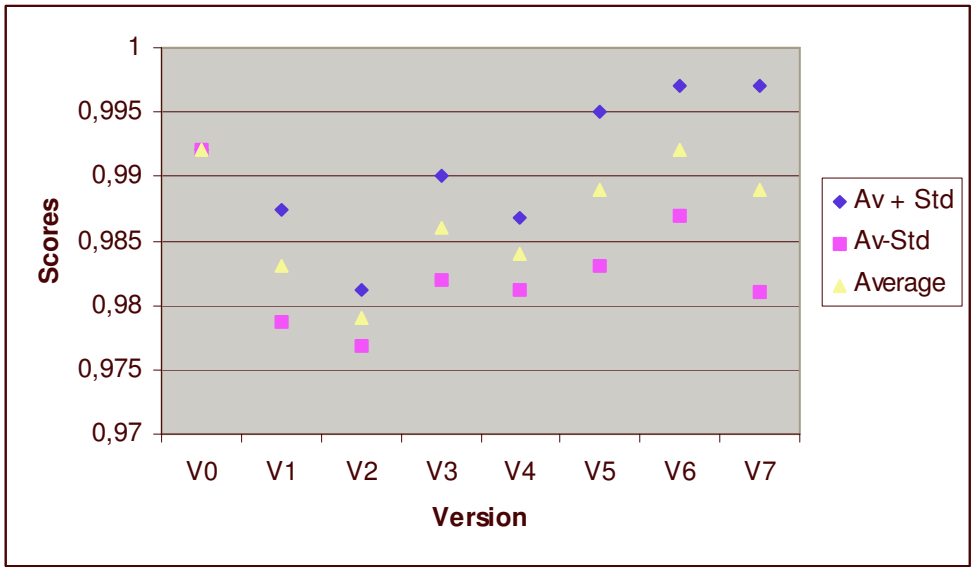

Fig 7 Fitness score and processing time for 7 configurations (average and maximum fitness scores are represented)

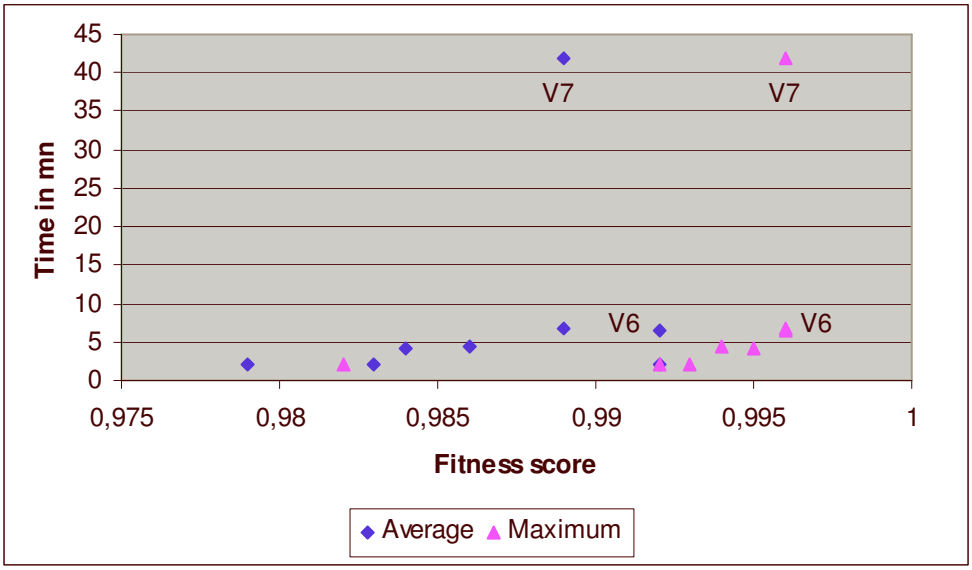

Fig 8 Score evolution (average for pop_a and pop_c) during the genetic process. The breaking mechanism affects pop_c in a significant way by incorporating diversity but has no real incidence on pop_a.

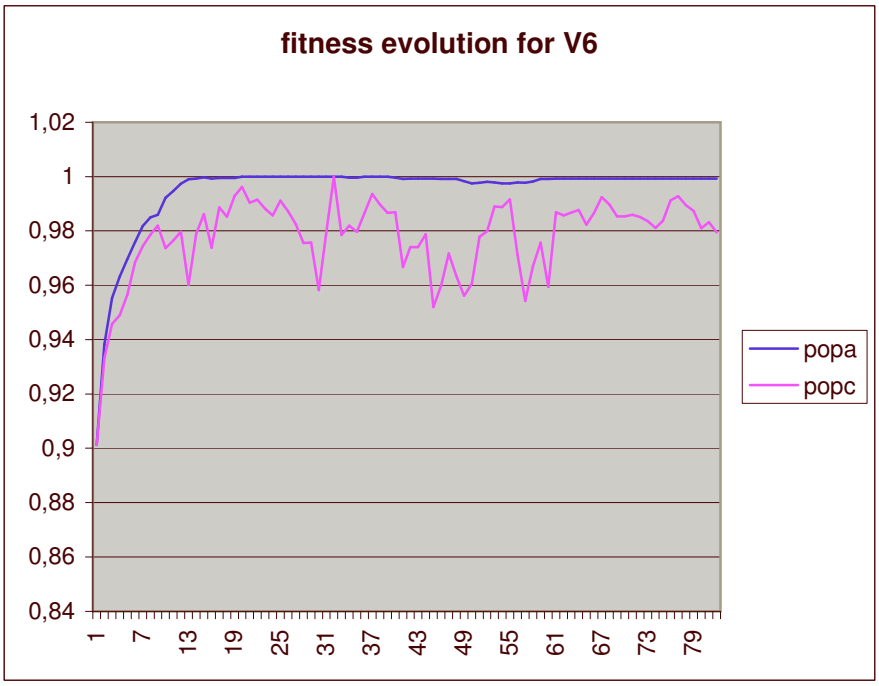

Fig 9 |S2| evolution (average for pop_a and pop_c) during the genetic process for the Iris database. The curves have similar evolution but the breaking mechanism affects more pop_c than pop_a.

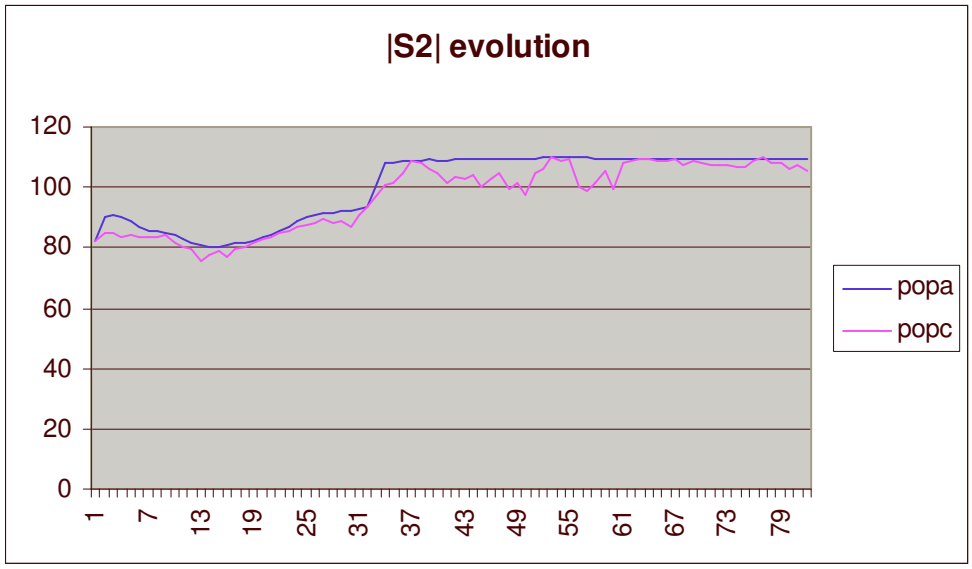

Fig 10 Example of graphical results obtained with the chemometrics database. With two features and about 25% of discarded patterns we can obtain a simple view of the database where the classes are separable.