

Evolutionary Search of Thresholds for Robust Feature Set Selection: Application to the Analysis of Microarray Data

Carlos Cotta¹, Christian Sloper², and Pablo Moscato³

¹ Dept. Lenguajes y Ciencias de la Computación, University of Málaga, ETSI Informática, Campus de Teatinos, 29071 – Málaga, Spain

² Department of Informatics, University of Bergen, HIB, 5020 Bergen, Norway

³ Newcastle Bioinformatics Initiative,
School of Electrical Engineering and Computer Science,
The University of Newcastle, Callaghan, NSW, 2308, Australia

Contact email: `ccottap@lcc.uma.es`

Abstract. We deal with two important problems in pattern recognition that arise in the analysis of large datasets. While most feature subset selection methods use statistical techniques to preprocess the labeled datasets, these methods are generally not linked with the combinatorial properties of the final solutions. We prove that it is NP -hard to obtain an appropriate set of thresholds that will transform a given dataset into a binary instance of a robust feature subset selection problem. We address this problem using an evolutionary algorithm that learns the appropriate value of the thresholds. The empirical evaluation shows that robust subset of genes can be obtained. This evaluation is done using real data corresponding to the gene expression of lymphomas.

1 Introduction

The problems discussed in this paper are central to pattern recognition and data mining tasks. Although their relevance is not restricted to Bioinformatics, they are of particular importance for this latter field, and more precisely in the analysis of microarray data. Consider the following typical situation: data from microarray experiments have been collected; these data contain gene-expression patterns for several samples from different classes. For simplicity, let us assume that the data samples are obtained from some healthy individuals and from some individuals affected by some illness. It is obviously interesting –for the purposes of prognosis, development of genic therapies, etc.– to determine which genes are related to this illness. This problem could be formalized as follows [1]:

k -FEATURE SET (decision version)

- **Input:** A set X of m examples (which are composed of a binary value specifying the value of the *target feature* and a vector of n binary values specifying the values of the other features) and an integer $k > 0$.

- **Question:** Does there exist a set S of *non-target* features (i.e., $S \subseteq \{1, \dots, n\}$) such that:
 - $|S| \leq k$,
 - No two examples in X that have identical values for all the features in S have different values for the target feature ?

Given a set of measurements obtained by means of a microarray experiment on m samples/conditions, the 0-1 values on each one of the features would correspond to under or over expressed genes. It is clear that a decision should be made as to a given threshold on the measured value such that all measures below to that value are assumed to correspond to a ‘1’ while the others would correspond to ‘0’. We will return to this issue later on in the paper.

Notice at this point the nature of the data being analyzed. Despite the enormous advances that are taking place in the microarray technology, the data we are dealing with is inherently noisy, since measurements can have errors. This implies that robust feature identification methods are essential. In other words, we must seek gene subsets such that their discriminatory power is kept even if some genes of this subset are later shown to have either measurement errors or spurious correlation with the target feature. This robust feature subset selection problem will be formalized in the next section. We will analyze the complexity of the problem, both as an stand-alone combinatorial problem, and in connection with the discretization of the original data. As it will turn out, both problems are intrinsically hard, and hence heuristic approaches are required to tackle them. We will show how an evolutionary algorithm jointly with a reduction approach inspired by kernelization rules often used in the design of fixed-parameter algorithms can provide good solutions for this problem.

2 The $(\alpha, \beta) - k$ -Feature Set Problem

As mentioned before, we aim at obtaining subsets of features (i.e., genes – both terms are interchangeably used) of robust discriminatory power. To this end, we will now introduce a generalization of the k -FEATURE SET problem which accounts for finding these possibly larger, though more robust feature sets from the data.

$(\alpha, \beta) - k$ -FEATURE SET

- **Input:** An instance of the k -FEATURE SET problem and integers $\alpha, \beta \geq 0$.
- **Question:** Does there exist a set S of *non-target* features ($S \subseteq \{1, \dots, n\}$) such that:
 - $|S| \leq k$,
 - for any pair of examples in X that have different values for the target feature, there are at least α different values for the features in S ,
 - for any pair of examples in X that have the same value for the target feature, there are at least β equal values for the features in S .

For the sake of clarity, let us appropriately reformulate this problem by making its structure explicit:

$(\alpha, \beta) - k\text{-FEATURE SET}$

- **Instance:** A set of m examples $X = \{x^{(1)}, \dots, x^{(m)}\}$, such that for all i , $x^{(i)} = \{x_1^{(i)}, x_2^{(i)}, \dots, x_n^{(i)}, t^{(i)}\} \in \{0, 1\}^{n+1}$, and three integers $k > 0$, and $\alpha, \beta \geq 0$.
- **Question:** Does there exist an $(\alpha, \beta) - k$ -feature set S , $S \subseteq \{1, \dots, n\}$, with $|S| \leq k$ and such that:
 - for all pairs of examples $i \neq j$, if $t^{(i)} \neq t^{(j)}$ there exists $S' \subseteq S$ such that $|S'| \geq \alpha$ and for all $l \in S'$ $x_l^{(i)} \neq x_l^{(j)}$?
 - for all pairs of examples $i \neq j$, if $t^{(i)} = t^{(j)}$ there exists $S' \subseteq S$ such that $|S'| \geq \beta$ and for all $l \in S'$ $x_l^{(i)} = x_l^{(j)}$?

We note that in the definition above the set S' is not fixed for all pairs of examples, but it is a function of the pair of examples chosen, so we mean $S' = S'(i, j)$.

The basic idea of this problem is to achieve robustness via some redundancy in example discrimination. Thus, we seek to have at least α genes for differentiating between any two samples of different classes. Similarly, we want to have at least β genes with consistent values for any two samples of the same class.

Let us now study the complexity of this problem. First of all, we consider its classical complexity within the P vs NP paradigm. The following theorem establishes that the problem is NP -hard:

Theorem 1. *The $(\alpha, \beta) - k\text{-FEATURE SET}$ problem is NP -hard.*

Proof. The proof is simple, and based on the fact that this problem generalizes the basic $k\text{-FEATURE SET}$ problem. This latter problem was shown to be NP -hard by Davies and Russell [1]. Notice now that the $k\text{-FEATURE SET}$ problem can be expressed as an $(1, 0) - k\text{-FEATURE SET}$ problem. Ergo, the $(\alpha, \beta) - k\text{-FEATURE SET}$ problem is in general NP -hard. \square

This NP -hardness result implies that there currently exists no polynomial-time algorithm for this problem. A possibility for dealing with this complexity barrier is resorting to the realm of parameterized complexity [2]. This paradigm tries to study the complexity of a combinatorial problem once some *parameter* has been extracted from it. This parameter may represent some structural property of the problem, or a property of the solutions sought. The basic idea here is trying to see whether the complexity of the problem can be isolated within this parameter, thus allowing for efficiently solving the problem for fixed values of the parameter ⁴. This is known as fixed-parameter tractability (*FPT*) [4].

In this case, the cardinality of the feature subset is the natural parameter, since this cardinality is always intended to be kept as small as possible. Unfortunately, it can be shown that the problem is also probably intractable from a parameterized point of view:

⁴ The $k\text{-VERTEX COVER}$ problem is a good example of this situation. The brute-force approach is $\Omega(n^{k+1})$ while a clever algorithm can solve it in $O(1.286^k + n)$, i.e., in linear time for any fixed k [3].

Theorem 2. *The $(\alpha, \beta) - k$ -FEATURE SET problem is not fixed-parameter tractable for parameter k .*

Proof. Again the proof relies on results for the basic k -FEATURE SET problem. Cotta and Moscato [5] have recently proved that this latter problem is $W[2]$ -complete. $W[2]$ is a parameterized class comprising substantially harder problems than FPT . Under the currently open –but widely believed conjecture in parameterized complexity– that $FPT \neq W[1]$, and being $W[1] \subseteq W[2]$ we can not expect that a clever fixed-parameter algorithm can be found for the $(\alpha, \beta) - k$ -FEATURE SET problem. \square

Once both the classical intractability and the parameterized intractability of the problem have been established, it becomes evident that heuristic algorithms are the most appropriate choice for tackling this problem. For example, evolutionary algorithms could be used here. However, recall that solving this $(\alpha, \beta) - k$ -FEATURE SET problem is not our final goal. On the contrary, it is a problem that arises in an intermediate stage, once we have determined the thresholds for discretizing to 0/1 the values of the gene expression levels. Since it is likely that many different choices of thresholds have to be tested, we must strive for finding a fast resolution method for dealing with this subsidiary problem. The next section will describe a greedy heuristic coupled with reduction rules for this purpose.

3 Fast Heuristic Resolution of the $(\alpha, \beta) - k$ -Feature Set Problem

In order to approach the heuristic resolution of this problem in its optimization version, we will use the underlying relationship between it and the RED-BLUE DOMINATING SET problem. To illustrate this relationship, let us begin by considering the $(1, 0) - k$ -FEATURE SET problem, i.e., the basic k -FEATURE SET problem. An instance I of this problem is reducible to RED-BLUE DOMINATING SET by using the following procedure: let $G(V_1 \cup V_2, E)$ be a bipartite graph such that

- there is a red vertex $g_i \in V_2$ for each feature/gene in I , i.e., $|V_2| = n$.
- there is a blue vertex $p_{jk} \in V_1$ for each pair of examples $x^{(j)}$ and $x^{(k)}$ such that $t^{(j)} \neq t^{(k)}$.
- there is an edge (g_i, p_{jk}) whenever $x_i^{(j)} \neq x_i^{(k)}$.

It can be clearly seen that I is a yes-instance if, and only if, there exists a red dominating set $D \subseteq V_2$ such that $|D| \leq k$. This construction can be generalized to the $(\alpha, 0) - k$ -FEATURE SET problem by requesting that D be α -dominating, i.e., that at least α vertices in D dominate each vertex in V_1 [6]. The final generalization to the $(\alpha, \beta) - k$ -FEATURE SET problem is easy from here: a tripartite graph $G(V_1 \cup V_2 \cup V_3, E)$ is constructed such that V_1 , V_2 , and the edges among vertices in them are defined as above, and

- there is a blue vertex $c_{jk} \in V_3$ for each pair of examples $x^{(j)}$ and $x^{(k)}$ such that $t^{(j)} = t^{(k)}$.

- there is an edge (g_i, c_{jk}) whenever $x_i^{(j)} = x_i^{(k)}$.

According to this construction, an instance I would be a yes-instance if, and only if, $D \subseteq V_2$ α -dominates V_1 , β -dominates V_3 , and $|D| \leq k$.

Once this translation mechanism has been performed, the resulting graph has to be solved. Obviously, finding the generalized dominating set is NP -hard as well, so a greedy heuristic will be utilized. This greedy heuristic will be jointly applied with some reduction rules. These rules are aimed at simplifying the problem instance, without altering its solution space, i.e., any solution of the original instance will be a solution of the simplified instance, and vice versa. As shown in [7], the application of reduction rules can in some cases turn large instances of NP -hard problems into trivial instances or small instances solvable by hand or by enumeration.

Let r_v be an integer associated with each vertex $v \in V_1 \cup V_3$; initially, $r_p = \alpha$ for each $p \in V_1$, and $r_c = \beta$ for each $c \in V_3$; let $G(v) = \{g \in V_2 \mid (g, v) \in E\}$ be the set of vertices (genes) dominating vertex $v \in V_1 \cup V_3$; conversely, let $N(g) = \{v \in V_1 \cup V_3 \mid (g, v) \in E\}$ be the vertices in $V_1 \cup V_3$ dominated by gene $g \in V_2$. Now, the rules considered for this problem are the following:

- R1. For each $v \in V_1 \cup V_3$ such that $r_v = |G(v)|$ do
 - i. For each $g \in G(v)$, mark g as belonging to the solution.
 - ii. Delete v from G .
- R2. For each $v \in V_1 \cup V_3$ such that $r_v \leq 0$ delete v from G .
- R3. For each $v_1, v_2 \in V_1 \cup V_3$, $v_1 \neq v_2$ such that $r_{v_1} \geq r_{v_2}$ and $G(v_1) \subseteq G(v_2)$, delete v_2 from G .

In these rules, the following actions are taken whenever a gene is marked, or a vertex is deleted:

- Gene marking* $[g]$: For each $v \in N(g)$ do
- i. $r_v \leftarrow r_v - 1$.
 - ii. $G(v) \leftarrow G(v) \setminus \{g\}$.

- Vertex deleting* $[v]$: For each $g \in G(v)$ do $N(g) \leftarrow N(g) \setminus \{v\}$

These rules greatly simplify the original instance by marking genes that are bound to appear in the final solution, and removing *subsumed* vertices, i.e., vertices that will be dominated for sure upon domination of another vertex. The application of these rules is interleaved with the use of a greedy insertion rule for adding a new gene to the solution any time the graph cannot be further simplified. More precisely, the process would be as follows:

1. **while** $V_1 \cup V_3 \neq \emptyset$ **do**
 - (a) Apply reduction rules.
 - (b) Let $g \in V_2$ be the gene for which $|N(g)|$ is maximal. Mark g as belonging to the solution.

Notice that the fact that indispensable genes are directly added to the solution, and that subsumed vertices are deleted from every $N(g)$, makes the greedy selection be more meaningful than otherwise.

4 Threshold Selection for Robust Feature Selection

As anticipated in Sect. 1, the $(\alpha, \beta) - k$ -FEATURE SET problem is a subsidiary problem that emerges once thresholds for discretizing gene-expression values have been set. We will now discuss the computational complexity of this latter problem, i.e., setting the threshold value for each gene such that an $(\alpha, \beta) - k$ -FEATURE SET problem is obtained:

(α, β) -THRESHOLD SELECTION

- **Input:** A $m \times n$ \mathbb{R} -matrix M , class identifiers $t[i] \in \mathbb{N}$ for every row i , $1 \leq i \leq m$, and two integers $\alpha, \beta \geq 0$.
- **Question:** Does there exist an array of n thresholds (one for each of the columns in M) such that each entry in the i th column of M greater than the i th threshold is given the value 1, and 0 otherwise, and such that
 1. $\forall i, j, t[i] \neq t[j]$, the number of columns where $M_{i,l} \neq M_{j,l}$ (disagree) is at least α , and
 2. $\forall i, j, t[i] = t[j]$, the number of columns where $M_{i,l} = M_{j,l}$ (agree) is at least β ?

We note that this is a necessary but not a sufficient condition to create a yes-instance of the $(\alpha, \beta) - k$ -FEATURE SELECTION problem. Notice also that for the sake of generality we have considered natural class identifiers rather than simply binary tags.

Theorem 3. (α, β) -THRESHOLD SELECTION is NP -complete.

Proof. The membership of this problem to NP is straightforward: given a solution –i.e., a list of n thresholds– the corresponding binary matrix can be computed in $O(mn)$, and its feasibility can be checked in $O(m^2n)$, which is polynomial in the input size. As to the NP -hardness of the problem, it can be established by reduction from k -VERTEX COVER. To avoid certain difficult cases we will reduce from a slightly modified version k -VERTEX COVER_M. The modification only guarantees that the vertex cover has cardinality greater or equal to 3, and it should be clear that this variation is NP -complete.

k -VERTEX COVER_M

- **Input:** A graph $G = (V, E)$ where the minimum vertex cover of G has at least 3 vertices, and an integer $k > 0$.
- **Question:** Does $G = (V, E)$ have a vertex cover of size k ?

Given an instance of a k -VERTEX COVER_M $G = (V, E)$, let $n = |V|$ and $m = |E|$. We will construct an instance of (α, β) -THRESHOLD SELECTION. Let $\alpha = 1$ and $\beta = (n - k)$ and construct a $(m + 2) \times n$ matrix M as follows:

$$\begin{aligned} \forall i \leq m, \forall j \leq n \quad M_{i,j} &= \begin{cases} 1 & \text{if } v_j \in e_i \\ 0 & \text{otherwise} \end{cases} \\ \forall j, \quad M_{(m+1),j} &= 0 \\ \forall j, \quad M_{(m+2),j} &= 1 \end{aligned}$$

The target values are set as follows:

$$T[i] = \begin{cases} 0 & \text{if } i \leq m \\ 1 & \text{if } i \geq m + 1 \end{cases}$$

We will now argue that this instance of (α, β) -THRESHOLD SELECTION has a solution if and only if k -VERTEX COVER_M has a vertex cover of size at most k .

(\Rightarrow) Given a k -VERTEX COVER_M $S \subseteq V(G')$ we can find a yes-instance of (α, β) -THRESHOLD SELECTION. In the matrix M set the thresholds as follows:

$$\text{Threshold}[i] = \begin{cases} 0.5 & \text{if } v_i \in S \\ 1.5 & \text{otherwise} \end{cases}$$

We argue that this assignment of thresholds to each of the columns satisfy the requirements for α and β .

Claim (1). β is satisfied for all pairs of rows.

Proof. Since $|S| \leq k$ at least $(n - k)$ columns have threshold 1.5. This sets all values in these columns to 0. Thus any pair of rows agree in at least $(n - k)$ places.

Claim (2). For all pairs of examples (i, j) for which $t[i] \neq t[j]$, the number of columns for which $M_{i,k} \neq M_{j,k}$ is at least α .

Proof. It is sufficient to argue that an arbitrary row representing an edge has at least one 1 and at least one 0 entry in the columns representing the vertex cover S . By definition a vertex cover must be incident to all edges. Thus there is at least one 1 in every row. Further, since every solution S is of at size at least 3 and every row contains at most 2 ones it follows that every row has at least one 0.

(\Leftarrow) Given a valid assignment of thresholds for each of the columns of M we can find a vertex cover in S . We remark that setting thresholds in a 0-1 matrix leaves only three possibilities: (1) setting an entire column to 0, (2) leaving the column with no changes, and (3) setting the entire column to 1. For purposes regarding agreement/disagreement (1) and (3) can be considered equivalent. Note that a changed column can never have any disagreements. To achieve at least $(n - k)$ agreements between row $(m + 1)$ and row $(m + 2)$ we must change at least $(n - k)$ columns.

We will argue that the at most k remaining unchanged columns represent a vertex cover in G . Since no changed column has any disagreements we must find the required disagreements in these remaining columns. Thus, each row (representing an edge) must have at least one 1 to disagree with row $(m + 1)$ (which are all 0). This is clearly a vertex cover. \square

This NP -completeness result does obviously imply the NP -hardness of the corresponding optimization version of the problem, i.e., finding the thresholds for obtaining the smallest (α, β) -feature subset. Hence, we will tackle this problem using a metaheuristic approach.

5 Learning Thresholds with Evolutionary Algorithms

The problem we will now address can be viewed as a continuous optimization problem. Let l_i and u_i be the lower bound and the upper bound for the expression values of gene g_i . We therefore have a search space $\mathcal{S} = \mathcal{D}_1 \times \cdots \times \mathcal{D}_n$, where $\mathcal{D}_i = [l_i, u_i]$. Let $\mathbf{x} \in \mathcal{S}$ be a vector of thresholds for each gene. When this vector is applied on the matrix of gene-expression values, a binary matrix is obtained. Together with the class-identifiers, this matrix constitutes an instance of the (α, β) -FEATURE SELECTION problem (its optimization version) that can be solved via the heuristic approach described in Sect. 3. The size $s(\mathbf{x})$ of the so-obtained gene subset indicates the absolute quality of vector \mathbf{x} . We would be interested in finding $\mathbf{x} \in \mathcal{S}$ such that $s(\mathbf{x})$ is minimal.

This continuous optimization problem has been attacked using an evolutionary algorithm specialized for this task: an evolution strategy (ES). To be precise, a (16, 100)-ES with binary tournament selection, flat recombination, and independent self-adaptive stepsizes for each variable has been used. The stepsizes are mutated following the guidelines shown in [8], i.e., a global learning rate $\tau = 1/\sqrt{2n}$, and a local learning rate $\tau' = 1/\sqrt{\sqrt{2n}}$. The fitness function (to be minimized) is defined as

$$\text{Fitness}(\mathbf{x}) = s(\mathbf{x}) + \left(1 - \frac{\min_{v \in V_1 \cup V_3} \{|G(v)|\}}{n+1}\right). \quad (1)$$

The rationale behind this fitness function is the following: first of all, notice that the second term is always smaller than 1, and hence, minimizing $s(\mathbf{x})$ is sought in the first place. In case two sets of thresholds induce gene subsets of the same size, the one that maximizes the minimum degree of a vertex in $V_1 \cup V_3$ is preferred. The underlying idea is that this way, the density of the graph is increased, and the chances that a small gene subset dominates $V_1 \cup V_3$ are higher.

The experiments have been done using both simulated data and data from real microarray experiments. Due to space constraints, we will focus here on the latter, which is taken from [9]. This dataset corresponds to different types of diffuse large B-cell lymphoma, and comprises gene-expression profiles for 2984 genes. There exist two classes in these data, each one containing 4 samples. Figure 1 (left) shows the matrix of gene-expression values. The evolutionary algorithm has been run on this dataset using different values for α and β . Concretely, we have considered $\alpha = \beta \in \{100, 200, 300\}$. For each value, 10 runs have been performed, for a maximum number of 50,000 fitness evaluations.

For $\alpha = \beta = 100$, the ES manages to find a gene subset of exactly 100 genes in all runs. A subset of this size is the optimal solution for this parameter setting, although we notice that this optimum is not unique. Figure 1 (middle-left) shows the original data after being normalized to $[-1, +1]$ using the thresholds obtained in one of the runs as the zero-level, and Fig. 1 (middle-right and right) shows the selected genes in this same run. As it can be seen, the gene subset selected exhibits a clearly distinctive expression profile in each of the classes, thus providing a robust mechanism for classifying samples.

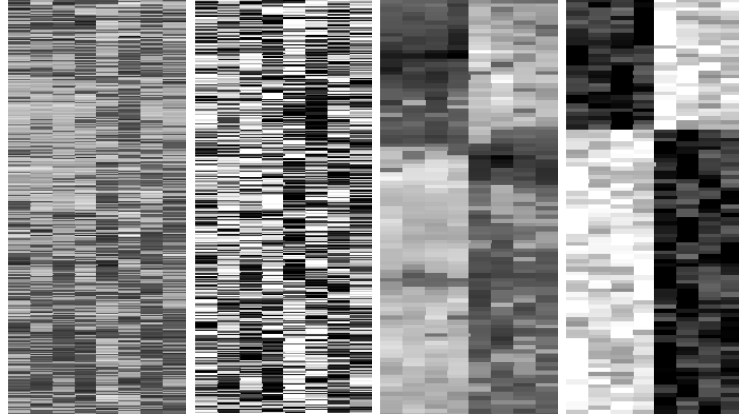


Fig. 1. (Left) Original dataset used in the experimentation. (Middle-left) The original dataset after applying the thresholds obtained in a typical run for $\alpha = \beta = 100$. (Middle-right) Genes selected in a typical run for $\alpha = \beta = 100$. (Right) Genes selected in a typical run for $\alpha = \beta = 100$ (thresholds applied).

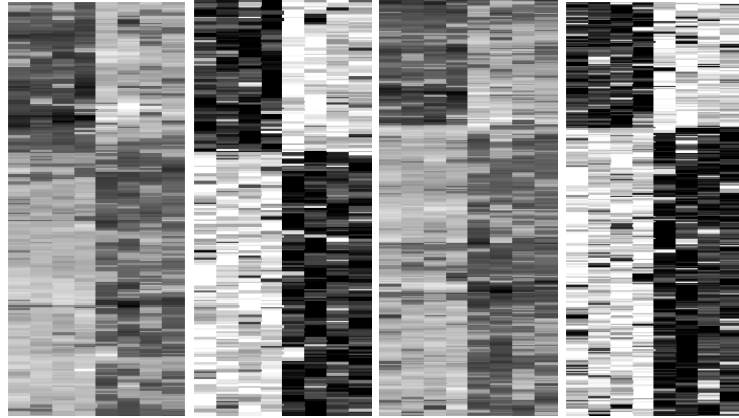


Fig. 2. (Left) Genes selected in a typical run for $\alpha = \beta = 200$. (Middle-left) Genes selected a typical run for $\alpha = \beta = 200$ (thresholds applied). (Middle-right) Genes selected in a typical run for $\alpha = \beta = 300$. (Right) Genes selected a typical run for $\alpha = \beta = 300$ (thresholds applied).

For $\alpha = \beta = 200$ and $\alpha = \beta = 300$, the ES finds gene subsets of 227.3 and 360.5 genes on average respectively. Figure 2 shows the outcome of average runs of the algorithm. Again, the results are very satisfactory, since well-differentiated expression profiles are found for each class. As a comparison, a plain greedy heuristic using 0.0 as the discretization threshold for all genes produces subsets of 129, 292, and 452 genes for $\alpha = \beta = 100, 200$, and 300 respectively, i.e., above 25% larger gene subsets on average.

6 Conclusions

We have presented both a theoretical and an applied study of two problems arising in the analysis of microarray data. We have shown that these problems are inherently hard, suggesting the need for heuristics. Since no pre-existing algorithms were available (recall these are novel problems that depart for classical feature selection tasks), a two-level heuristic approach has been proposed: on the first level, an evolutionary algorithm explores different thresholds for gene-expression values; on the second level, a greedy algorithm coupled with reduction rules solves a subsidiary combinatorial problem. The evaluation of this combined heuristic has been encouraging, providing robust subsets of genes for identifying different classes of lymphoma.

On the theoretical line, future work will be directed to determine whether some subclasses of the general (α, β) -THRESHOLD SELECTION problem can be found to be in the *FPT* class. From an applied standpoint, we plan to study several computational issues of the evolutionary algorithm proposed. In particular, we intend to test the performance of the algorithm in parallel environments. The fact that this application is computationally intensive lead us to believe that it can greatly benefit from the use of distributed programming techniques.

Acknowledgements Carlos Cotta is partially supported by Spanish MCyT, and FEDER under contract TIC2002-04498-C05-02. We thank Mike Fellows and Luke Mathieson for useful discussions, and Evosolve (UK Registered Charity No. 1083684) for supporting previous work.

References

1. Davies, S., Russell, S.: *NP-completeness of searches for smallest possible feature sets*. In Greiner, R., Subramanian, D., eds.: AAAI Symposium on Intelligent Relevance, New Orleans, AAAI Press (1994) 41–43
2. Downey, R., Fellows, M.: *Parameterized Complexity*. Springer-Verlag (1998)
3. Chen, J., Kanj, I., Jia, W.: *Vertex cover: further observations and further improvements*. In: *Proceedings of the 25th International Workshop on Graph-Theoretic Concepts in Computer Science*. Number 1665 in *Lecture Notes in Computer Science*, Springer-Verlag (1999) 313–324
4. Downey, R., Fellows, M.: *Fixed parameter tractability and completeness I: Basic theory*. *SIAM Journal of Computing* **24** (1995) 873–921
5. Cotta, C., Moscato, P.: *The k -FEATURE SET problem is $W[2]$ -complete*. *Journal of Computer and Systems Science* **67** (2003) 686–690
6. Harant, J., Pruchnewski, A., Voigt, M.: *On dominating sets and independent sets of graphs*. *Combinatorics, Probability and Computing* **8** (1999) 547–553
7. Weihe, K.: *Covering trains by stations or the power of data reduction*. In Battiti, R., Bertossi, A., eds.: *Proceedings of Algorithms and Experiments (ALEX 98)*, Trento, Italy (1998) 1–8
8. Bäck, T.: *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, New York (1996)
9. Alizadeh, A., et al.: *Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling*. *Nature* **403** (2001) 503–511