A Hybridized Multiobjective Evolutionary Approach for Microarray Biclustering

Cristian Andrés Gallo[1], Jessica Andrea Carballido[1], and Ignacio Ponzoni[1,2]

[1] Laboratorio de Investigación y Desarrollo en Computación Científica (LIDeCC),
Departamento de Ciencias e Ingeniería de la Computación,
Universidad Nacional del Sur, Av. Alem 1253, 8000, Bahía Blanca, Argentina
{jac,ip}@cs.uns.edu.ar
[2] Planta Piloto de Ingeniería Química (PLAPIQUI) - UNS – CONICET
CCT BB, Co. La Carrindanga km.7, CC 717, Bahía Blanca, Argentina

Abstract. In this paper a new memetic algorithm that integrates a multiobjective evolutionary algorithm (the SPEA2) with a local search technique, for microarray data analysis, is presented. The algorithm explores the gene expression data matrix in order to find biclusters that fulfill several objectives. The performance of the new strategy was analyzed by means of a fair comparison with a well known method presented in the literature. The cases of study were two datasets corresponding to *Saccharomyces cerevisiae* and *human B-cells Lymphoma*. Our algorithm outperformed the previous method in terms of *Set Coverage* and *Spacing*, yielding to undeniably promising outcomes. Nonetheless, more experiments with data from other organisms are necessary, thus leading to more concluding results.

Keywords: gene regulation, biclustering, evolutionary algorithms

1 Introduction

The arrival of genomic high-technologies has triggered the generation of massive volumes of molecular biological datasets. In particular, the DNA microarray technology makes possible the simultaneously recording the activity of thousands of genes [1]. The basic information unit is the *expression level* of a gene, which measures the relative abundance of the mRNA of a gene under a specific experimental *condition* (or *sample*). The expression level of a great number of genes under various experimental conditions can be arranged into a matrix, namely *gene expression data matrix* (M_{GC}), where rows and columns correspond to genes and samples respectively. Each matrix entry e_{gc} is a real value that indicates the expression level of a gene under a particular condition.

An important aim of the gene expression data analysis consists in grouping together genes according to their expression levels under multiple conditions. This task is generally carried out by clustering techniques [2, 3]. Nowadays, one relevant application of the clustering of gene expression patterns is the discovering of gene regulatory networks [4]. These networks relate genes and gene products in a traditional graph [5]. An important cause of gene co-expression is the sharing of regulation mechanisms (co-regulation). Clustering of co-expressed genes into biologically meaningful classes is useful for inferring the biological role of an unknown gene that is co-expressed with known genes [4]. In

XXXIV Conferencia Latinoamericana de Informática (CLEI 2008), pp. 759-768, 2008.

general, genes are not relevant for all the experimental conditions, but groups of genes are often co-regulated and co-expressed only under some specific conditions. This important observation has turned the academic interest to the design of *biclustering* methods that simultaneously group genes and samples [2]. In this context, a satisfactory *bicluster* consists in a group of rows and columns of the M_{GC} that satisfies some similarity score [6] in conjunction with some other criteria.

As we will discuss in following sections, the elevated complexity of this task has motivated the development of several approximation methods to generate near optimal solutions. In particular, in this paper, a memetic multiobjective evolutionary algorithm called SPEA2^{LS} is proposed. The new algorithm hybridizes the SPEA2 [7] with a new variant of the local search algorithm proposed by Cheng & Church [6]. For each potential bicluster, SPEA2^{LS} evaluates the following objectives: the mean squared residue proposed by Cheng and Church [6] as a similarity measure for biclusters; the row variance; and the dimension (number of rows and columns). In order to assess the performance of our approach, a comparative study between SPEA2^{LS} and the Cheng & Church algorithm [6] is presented, using as case studies the *Saccharomyces cerevisiae* cell cycle expression dataset obtained from Tavazoie *et al.* [8], and the *human B-cells Lymphoma* expression dataset presented by Alizadeth *et al.* [9].

The paper is organized as follows: in the Section 2 some basic definitions about biclustering are introduced; next the core algorithms used to build the memetic approach are described; later the main features of the SPEA2^{LS} are presented; and finally, in Sections 5 and 6, the experimental results and conclusions are discussed.

2 Biclustering

In the context of this work, a bicluster is defined as a pair (G, C) where $G \subseteq \{1,\ldots,m\}$ is a subset of genes (rows) and $C \subseteq \{1,\ldots,n\}$ is a subset of conditions (columns) [6]. The main goal is to find the largest biclusters that does not exceed certain homogeneity constrain. It is also important to consider that the variance of each row in the bicluster should be relatively high, in order to capture genes exhibiting fluctuating coherent trends under some set of conditions. The bicluster size is the number of rows $f(G)$ and the number of columns $g(C)$. The homogeneity $h(G,C)$ is given by the mean squared residue score, while the variance $k(G, C)$ is the row variance [6]. Therefore, our optimization problem can be defined as follows:
maximize

$$f(G) = |G| \cdot \tag{1}$$

$$g(C) = |C| \cdot \tag{2}$$

$$k(G,C) = \frac{\sum_{g \in G, c \in C} (e_{gc} - e_{gC})^2}{|G| \cdot |C|} \cdot \tag{3}$$

subject to

$$h(G,C) \leq \delta \tag{4}$$

with $(G,C) \in X$, $X = 2^{\{1,\ldots,m\}} \times 2^{\{1,\ldots,n\}}$ being the set of all biclusters, where

$$h(G,C) = \frac{1}{|G| \cdot |C|} \sum_{g \in G, c \in C} \left(e_{gc} - e_{gC} - e_{Gc} + e_{GC}\right)^2 \tag{5}$$

is the mean squared residue score,

$$e_{gC} = \frac{1}{|C|} \sum_{c \in C} e_{gc}, \quad e_{Gc} = \frac{1}{|G|} \sum_{g \in G} e_{gc} \tag{6,7}$$

are the mean column and row expression values of (G,C) and

$$e_{GC} = \frac{1}{|G| \cdot |C|} \sum_{g \in G, c \in C} e_{gc} \tag{8}$$

is the mean expression value over all the cells that are contained in the bicluster (G,C). The user-defined threshold δ represents the maximum allowable dissimilarity within the cells of a bicluster. In other words, the residue quantifies the difference between the actual value of an element e_{gc} and its expected value as predicted for the corresponding row mean, column mean, and bicluster mean. If a bicluster has a mean square residue lower than δ, then we call the bicluster a δ-bicluster. The problem of finding the largest square δ-bicluster is NP-hard [6]. The high complexity of this problem has motivated development of heuristic techniques to generate near optimal solutions. In particular, evolutionary algorithms are well suited for addressing this class of problems [3, 4, 10].

3 Core algorithms

A multiobjective evolutionary algorithm (MOEA) is a global search heuristic, primarily used for optimization tasks. But when it is applied to biclustering, it may not find good solutions satisfying the homogeneity constraint. Moreover, it was observed that, in the absence of a local search strategy, stand-alone evolutionary algorithms could not generate satisfactory solutions [10]. In that context, we have hybridized a well known MOEA, called SPEA2 [7], with a local search method based on [6] to speed up the convergence.

3.1 Strength Pareto Evolutionary Algorithm (SPEA2)

A brief report of the SPEA2 used to globally explore the search space X is introduced here. For a more in-depth study of this method please be referred to [7]. A definition of the concept of Pareto non-domination [11] is given next, trailed by the overall algorithm.

Definition 1. If there are M objective functions, a solution x^1 is said to weakly dominate another solution x^2, if both conditions (a) and (b) are true:

(a) The solution x^1 is no worse than x^2 in all the M objective functions.
(b) The solution x^1 is strictly better than x^2 in at least one of the M objective functions.

Otherwise the two solutions are non-dominating to each other. When a solution *i* weakly dominates a solution *j*, we denote $i \preceq j$. If $\neg\exists\ i: i \preceq j$ then *j* is called a Pareto-optimal solution. Then the set of all Pareto-optimal solutions is called the Pareto-optimal front. The aim of a MOEA is to approximate the Pareto-optimal front.

Algorithm 1 (SPEA2 Main Loop)
Input: *N (population size), $\overline{N}$ (archive size), T (maximum number of generations)*
Output: *A (non-dominated set)*
Step 1: ***Initialization:*** *Generate an initial population P_0 and create the empty archive (external set) $P_0 = \phi$. Set $t = 0$.*
Step 2: ***Fitness assignment:*** *Calculate fitness values of individuals in P_t and $\overline{P}_t$ (cf. Section 3.1).*
Step 3: ***Environmental selection:*** *Copy all non-dominated individuals in P_t and $\overline{P}_t$ to $\overline{P}_{t+1}$. If size of $\overline{P}_{t+1}$ exceeds $\overline{N}$ then reduce $\overline{P}_{t+1}$ by means of the truncation operator, otherwise if size of $\overline{P}_{t+1}$ is less than $\overline{N}$ then fill $\overline{P}_{t+1}$ with dominated individuals in P_t and $\overline{P}_t$ (cf. Section 3.2).*
Step 4: ***Stopping:*** *If $t \geq T$ or another stopping criterion is satisfied then set A to the set of decision vectors represented by the non-dominated individuals in $\overline{P}_{t+1}$. Stop.*
Step 5: ***Mating selection:*** *Perform binary tournament selection with replacement on $\overline{P}_{t+1}$ in order to fill the mating pool.*
Step 6: ***Variation:*** *Apply recombination and mutation to the mating pool and set $\overline{P}_{t+1}$ to the resulting population. Increment generation counter and go to Step 2.*

Fitness Assignment. Each individual *i* in the archive $\overline{P}_t$ and the population P_t is assigned a strength value $S(i)$, representing the number of solutions it dominates:

$$S(i) = \left| \left\{ j \mid j \in P_t + \overline{P}_t \wedge i \succ j \right\} \right| \tag{9}$$

Using values of equation 9, the raw fitness *R(i)* of an individual *i* is calculated:

$$R(i) = \sum_{j \in P_t + \overline{P}_t, j \succ i} S(j) \tag{10}$$

The raw fitness is determined by the strengths of its dominators in the archive and in the population. Note that the fitness must be minimized. In order to be more accurate, additional density information is incorporated to discriminate between individuals having identical raw fitness values. The density estimation technique is based on the following idea: for each individual *i*, the distances (in the objective space) to all individuals *j* in the archive and in the population are calculated and stored in a list. After sorting the list in increasing order, the k^{th} element gives the distance sought, denoted as σ_i^k. We chose k=1 because it is often sufficient and leads to a more efficient implementation [12]. Then, the density $D(i)$ corresponding to *i* is defined by

$$D(i) = \frac{1}{\sigma_i^k + 2} \tag{11}$$

Finally, the fitness function *F(i)=R(i)+D(i)* can be obtained from equations 10 and 11.

Environmental Selection. During environmental selection, the first step is to copy all the non-dominated individuals from the archive and from the population to the archive of the next generation ($\overline{P}_{t+1}$). If the non-dominated front exactly fits into the archive the environmental selection step in completed. If the archive is too small, the best dominated individuals in the previous archive and population are copied to the new archive.

Otherwise (when the archive is too large) an archive truncation procedure is invoked which iteratively removes individuals from $\overline{P}_{t+1}$ to decrease its size to $\overline{N}$. At each repetition, an individual i is chosen for removal, for which $i \leq_d j$ for all $j \in \overline{P}_{t+1}$ with

$$i \leq_d j \quad :\Leftrightarrow \quad \forall \ \ 0 < k < \left|\overline{P}_{t+1}\right| : \sigma_i^k = \sigma_j^k \ \vee \\ \exists \ \ 0 < k < \left|\overline{P}_{t+1}\right| : \left[\left(\forall\, 0 < l < k : \sigma_i^l = \sigma_j^l\right) \wedge \sigma_i^k < \sigma_j^k\right] \tag{12}$$

where σ_i^k denotes the distance of i to its k-th nearest neighbor in $\overline{P}_{t+1}$.

3.2 Ad-hoc Local Search procedure

This subsection describes the local search procedure that hybridizes the SPEA2 giving place to SPEA2^{LS}. This greedy approach is based on [6], with some modifications introduced in order to consider the row variance and the overall efficiency of the proposal. The algorithm starts from a given bicluster (G, C). The genes or conditions having mean squared residue above (or below) a certain threshold are selectively eliminated (or added) according to the following algorithm:

Algorithm 2 (Local Search)
Input: (G, C) *(a bicluster)*
Output: (G, C)' *(an improved bicluster)*
Step 1: *Compute* e_{gC}, e_{Gc}, e_{GC} *and h(G, C) by equations 5-8.*
Step 2: *if* $h(G, C) < \delta$ *go to step 6.*
Step 3: *Remove all genes* $i \in G$ *satisfying*: $\frac{1}{C}\sum_{c \in C}\left(e_{gc} - e_{gC} - e_{Gc} + e_{GC}\right)^2 > \alpha \cdot \delta$.

Recalculate all means and perform the same operation on conditions. The equation for conditions is analogous.
Step 4: *Recompute* e_{gC}, e_{Gc}, e_{GC} *and h(G, C). If* $h(G, C) < \delta$ *go to step 6.*
Step 5: *Remove the node* i *(gene or condition) with the largest:*

$$d(i) = \frac{1}{C}\sum_{c \in C}\left(e_{ic} - e_{iC} - e_{Gc} + e_{GC}\right)^2 \cdot$$

The equation for the conditions is analogous. Go to step 4.
Step 6: *Recompute* e_{gC}, e_{Gc}, e_{GC} *and h(G, C).*
Step 7: *Add all conditions* $c \notin C$ *satisfying:* $\frac{1}{G}\sum_{g \in G}\left(e_{gc} - e_{gC} - e_{Gc} + e_{GC}\right)^2 \leq h(G, C)$

Step 8: *Recompute* e_{gC}, e_{Gc}, e_{GC}, *h(G, C) and k(G, C), this last by means of equation 3.*
Step 9: *Add all genes* $g \notin G$ *(or its inverse) satisfying:*

$$\frac{1}{C}\sum_{c \in C}\left(e_{gc} - e_{gC} - e_{Gc} + e_{GC}\right)^2 \leq h(G, C) \ \wedge \ \ k(G \cup \{g\}, C) \geq \mu \cdot k(G, C) \cdot$$

The equation for the inverse only differs in that the term e_{gc} *is multiplied by -1.*

The main differences with the implementation in [6] are the following:

* In Step 3 we remove multiple nodes considering a different threshold to delete them, $\alpha.\delta$ instead of $\alpha.h(G,C)$. As a consequence, Step 5 is performed a smaller num-

ber of times with respect to the original proposal in [6]. This is useful because, with a proper setting of the parameter α, the CPU time needed to optimize a bicluster is decreased. This is possible without loosing significant precision of the algorithm.

* In step 7 we incorporated the row variance, adding only the rows that will increase in a certain proportion the overall row variance of the individual.

* Finally, in the Steps 7-9 the original algorithm tries to add each row, each columns and each inverted row, in that order. In our case, we add each condition first. This increases (on average) the amount of conditions of the resulting bicluster since a column, in general, has more probability of being inserted in the solution when it contains less quantity of rows. In this way it is possible to obtain biclusters that contain a significant quantity of conditions and genes, since it is more relevant from the biological point of view to observe the behavior of the genes on a bigger amount of conditions.

Beside δ, two additional parameters need to be set for this algorithm. α determines how often multiple gene deletion is used. A higher α leads to less multiple gene deletion and thus, in general, requires more CPU time. The other parameter is μ that establishes a relationship between the number of genes and the row variance of the bicluster. A bigger μ results in individuals with a higher row variance and a smaller size. If the value μ=0 this step results equivalent to that of the original proposal in [6].

4 SPEA2^{LS}: a Memetic Approach

The aim of our study is to use the SPEA2 for approximating the Pareto front of biclusters from a given gene expression matrix, as this approach gives the best tradeoff between the objectives that we want to optimize. However, in view of the fact that the Pareto front also includes biclusters that do not satisfy the homogeneity restriction, we needed to guide the search to the area where this restriction is accomplished. In that context, we applied the aforementioned local search method to the archive $\overline{P}_t$ of the SPEA2 after each generation, thus guiding the search and speeding up the convergence of the MOEA by refining the chromosomes. In order to take into account the inverted rows, we have extended the classical representation of a bicluster and we have also modified the genetic operators. Then, our proposal performs over a double-sized search space, in contrast with the evolutionary biclustering methods found in the literature [3, 4, 10]. The importance of considering these inverted rows resides in that they form "mirror images" of the rest of the rows in the bicluster, and can be interpreted as opposite co-regulated (inhibitory regulation) [6]. In this way, our proposal is able to find biclusters that the former evolutionary methods cannot detect.

Representation. Each individual of the evolutionary algorithm represents one bicluster, which is encoded by a fixed size ternary string built by appending a string for genes with a binary string for conditions. The individual corresponds to a solution for the problem of optimal bicluster generation. If a bit is set to 1, it means that the relative row or column belongs to the encoded bicluster, otherwise it does not. To take into account the inverted rows we considered a negative value in the bit string for the genes. That is to say, a bit of the genes string is set to -1 when the relative inverted row belongs to the encoded solution. Fig. 1 shows encoding of genes and conditions for a random individual.

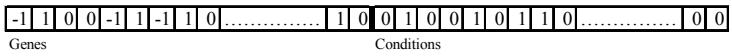

Fig. 1. An encoded individual representing a bicluster

Genetic operators. It is important to give a brief description of the genetic operators used in this approach, since they have a key influence in how the search is performed.

Mutation. This operator is implemented in the following way: first it is determined if the individual needs to be mutated by means of the probability assigned to the operator. In such case, a position of the binary string is selected at random, then proceeding to alter the bit in question. If the resulting position is a column, the corresponding bit is simply complemented. On the other hand if the resulting position is a row, then we have two cases: if the bit is in 0 then it is set to 1, and the sign is determined with a probability of 0.5. If the bit is in on (1 or -1) we simply change the value to 0.

Recombination. We chose a two-point crossover with a little restriction: one random point is selected on the rows and the other random point is selected on the columns. In this way, we ensure that the recombination is performed over both the genes' and the conditions' subspaces. Then, when both children are obtained by combining each one of the two parents' parts (i.e. the ends and the center), the individual that is selected to be the only descendant is the non-dominated one. If both are non-dominated, one of them is chosen at random.

Multiobjective fitness. As regards the objectives to be optimized, we observed that it was necessary to generate maximal sets of genes and conditions while maintaining the "homogeneity" of the bicluster with a relatively high row variance. These bicluster features, conflicting to each other, are well suited for multiobjective modeling. In that context, we decided to optimize the objectives defined by equations 1- 4: the quantity of genes, the quantity of conditions, the row variance, and the mean squared residue. The first three objectives are maximized while the last one is minimized.

In order to avoid solutions with too few conditions, rows and columns are considered individually. The aim is to have a better control on the quantity of each individual's samples. Otherwise, maximizing the size of the bicluster with rows and columns together (in a multiplicative way) gives as a result huge solutions, but that contain a lot of genes and few conditions (less than three in most cases). Individuals with these features are not very well suited as solutions from the biological point of view.

5 Experimental framework and results

A comparison between SPEA2^{LS} and the Cheng & Church algorithm [6] is presented here. For this analysis, we have used two sets of expression data, the *yeast Saccharomyces cerevisiae* cell cycle expression data from [8] and the *human B-cells Lymphoma* expression data from [9]. The *yeast* data contain 2.884 genes and 17 conditions and the expression values denote relative abundance. The data have been directly used in the preprocessed form as provided by the authors. All values are integers in the range between 0 and 600 replacing the missing values by sampling a random number from a uniform distribution between 0 and 800. The *Lymphoma* datasets contain 4.026 genes and 96

conditions. The expression levels are integers in the range between -750 and 650, where the missing values were replaced by a uniform distribution between -800 and 800.

Parameters setup. As regards of the parameter settings, we have determined the setup values which yielded the best results in a few preliminary runs, except for δ that was taken from [6]. Table 1 shows the parameters used for this study. Note that the MOEA parameters are equal for both datasets, whereas the local search parameters need to be tuned in each case. The fact that the parameter α is smaller for the *Lymphoma* dataset than for the *Yeast* dataset is due to efficiency reasons. We have set parameter μ on that value in order to obtain the best performance in the comparative study, although the exact determination will depend on the results looked for by the biologists, that is to say, bi-clusters with bigger size and smaller variance or the inverse situation.

Table 1. Default parameter setting for this study.

	SPEA2				Local Search		
	PopSize	Archive Size	N° of Gens.	Mutation Prob.	α	δ	μ
Yeast	100	100	75	0.3	1.8	300	0.998
Lymphoma					1.5	1200	0.999

Qualitative evaluation. The performance of the algorithms was assessed in terms of two metrics: Set Coverage and Spacing [11].

Set Coverage Metric: this measure can be used to find which one of the two methods *A* and *B* obtain solutions that are closer to the Pareto front. It calculates the proportion of solutions of *B* which are weakly dominated by solutions of *A*, as follows:

$$\mathcal{C}(A, B) = \frac{|\{b \in B \mid \exists a \in A : a \preceq b\}|}{|B|} \tag{13}$$

If $\mathcal{C}(A, B) = 1$, all members of *B* are weakly dominated by *A*. On the other hand, if $\mathcal{C}(A, B) = 0$, no member of *B* is weakly dominated by *A*. Since this operator is not symmetric, $\mathcal{C}(A, B)$ is not necessarily equal to 1- $\mathcal{C}(B, A)$. Thus, it is necessary to compute both to know how many solutions of *A* are covered by *B* and vice versa.

Spacing: this metric analyzes the diversity of the set of non-dominated solutions reported by a single algorithm. It is calculated with a relative distance measure between two consecutive solutions in the obtained non-dominated set, as follows:

$$\mathbf{S} = \sqrt{\frac{1}{|Q|}\sum_{i=1}^{|Q|}(d_i - \overline{d})^2} \tag{14}$$

where $d_i = \min_{k \in Q \wedge k \neq i} \sum_{m=1}^{M} |f_m^i - f_m^k|$ and $\overline{d} = \sum_{i=1}^{|Q|} d_i / |Q|$

The distance measure is the minimum value of the sum of other solutions in the non-dominated set. **S** measures the standard deviations of different d_i values. When the solutions are near uniformly spaced, **S** will be small. Thus, an algorithm finding a set of non-dominated solutions having a smaller *spacing* is better.

Comparative study. With respect to the experimental results, Cheng & Church [6] reported 100 biclusters (http://arep.med.harvard.edu/biclustering/) obtained by their node-deletion algorithm for each one of the two datasets. This deterministic method finds sub-matrices in the data gene expression matrix, which have low mean squared residue scores. It is important to remark that, taking into account our optimization objectives, we identified the subset of non-dominated biclusters from those reported

by [6] building a Pareto front comprised of 59 biclusters in the case of the *Yeast* dataset and 82 biclusters in the case of the *Lymphoma* dataset.

In table 2, the results of evaluating the metrics for the Pareto fronts obtained by SPEA2^{LS} in 25 runs and the Pareto front extracted from the biclusters reported by Cheng & Church (C&C) is presented for both datasets. The last column of each sub-table corresponds to the number of non-dominated biclusters found by SPEA2^{LS} on each run. The Spacing metric for the biclusters of C&C is calculated once, and the values are 0,0864807 for the *Yeast* dataset and 0,0489143 for the *Lymphoma* dataset.

Table 2. Values of the metrics for 25 runs of SPEA2^{LS} against the C&C Pareto front

	***Yeast* data set**					***Lymphoma* data set**				
	Set Coverage Metric			**Spacing**	**Size**	**Set Coverage Metric**			**Spacing**	**Size**
	A	SPEA2^{LS}	C&C	SPEA2^{LS}	SPEA2^{LS}	**A**	SPEA2^{LS}	C&C	SPEA2^{LS}	SPEA2^{LS}
Run	**B**	C&C	SPEA2^{LS}			**B**	C&C	SPEA2^{LS}		
1		0,0508475	0	0,042936	88		0,0243902	0	0,0290043	69
2		0,0508475	0,011236	0,0797892	89		0	0	0,0254372	67
3		0,135593	0,0212766	0,0716581	94		0	0	0,0234453	98
4		0,0677966	0,0246914	0,0476916	81		0,0365854	0	0,0197048	76
5		0,0169492	0,011236	0,0446212	89		0	0	0,0193862	91
6		0,0677966	0	0,0575045	84		0,0731707	0	0,0253784	89
7		0,0847458	0	0,0450482	85		0,0121951	0	0,0276477	64
8		0,220339	0,0224719	0,0451643	89		0,0853659	0	0,0209617	86
9		0,0169492	0	0,0472884	91		0	0	0,0240706	99
10		0,135593	0,011236	0,0669113	89		0	0	0,0236731	74
11		0,101695	0	0,0588176	83		0	0	0,0206069	97
12		0,101695	0,010989	0,0438995	91		0,097561	0	0,0243936	89
13		0,0338983	0,011236	0,043917	89		0	0	0,0254488	97
14		0,0508475	0,0111111	0,0428312	90		0,0243902	0	0,0276834	89
15		0,0847458	0	0,0594793	86		0,0365854	0	0,0235165	85
16		0,135593	0,0227273	0,0540757	88		0,0365854	0	0,0205129	80
17		0,152542	0	0,0769413	81		0,0609756	0	0,049318	90
18		0,0338983	0,0119048	0,051344	84		0,0487805	0	0,029582	83
19		0,101695	0,0365854	0,0572634	82		0	0	0,0183164	81
20		0,101695	0	0,0483056	83		0,0365854	0	0,0317309	61
21		0,0338983	0,0344828	0,0497456	87		0,0365854	0	0,0312944	81
22		0,0508475	0,0111111	0,0530636	90		0,0487805	0	0,0254533	78
23		0,0677966	0	0,0507114	89		0,0365854	0	0,0245224	83
24		0,169492	0,0229885	0,0473304	87		0	0	0,0235456	99
25		0,0508475	0,0340909	0,0772722	88		0	0	0,0210882	95
Avg.		0,0847458	0,01237499	0,05454442	87,08		0,0278049	0	0,0254289	84,04

The average set coverage of SPEA2^{LS} over C&C clearly outperforms the one obtained by C&C with respect to SPEA2^{LS} on both datasets. In particular, the biclusters obtained by SPEA2^{LS} exhibit better values for all the objectives, which becomes more evident in the case of the *Lymphoma* dataset (see Column 2 of the corresponding sub-table). As regards the Spacing metric, the values of the Pareto front obtained by SPEA2^{LS} are clearly better than the ones achieved by C&C on both datasets, which means that the solutions obtained by our approach are more sparsely situated in the frontier.

All the testing has been made on a *Pentium* III processor with 384 MB of RAM. Running times (on average) were: 660 seconds for the *Yeast* dataset and 1860 seconds for the *Lymphoma* dataset. We argue that running times of a half an hour are well acceptable, especially in comparison with the effort, time and economical costs needed to perform the biological experiments.

6 Conclusions

In this paper we have introduced the SPEA2^{LS}, a multiobjective framework for biclustering of gene expression data hybridized with a local search procedure. For one of the most relevant examples in the literature, the biclustering algorithm of Cheng & Church, we have demonstrated that the quality of those outcomes can been outperformed by our memetic approach. The comparative assessment of the results was performed over two benchmark gene expression datasets, *Yeast* and *Lymphoma*.

As a novel feature, we provide the biologists with an extra parameter to determine the form of biclusters they consider more relevant, giving them the possibility to adjust the size and the variance of the biclusters. Moreover, the evolutionary approaches for biclustering presented in the literature do not consider the inclusion of inverted rows, perhaps for efficiency reasons since the search space is duplicated. However, inverted rows are very important since they can be thought as co-regulated by receiving the opposite regulation. We have revealed that it possible to consider these "extra rows" thus improving the quality of the biclusters, without loss of efficiency.

Acknowledgments. Authors acknowledge the ANPCyT from Argentina, for Grant Nº11-1652, and SeCyT (UNS), for Grants PGI 24/N019, PGI 24/ZN15, PGI 24/ZN16.

References

1. Sohler, F.: Contextual Analysis of Gene Expression Data, Master Thesis dissertation (2006)
2. Madeira, S., Oliveira, A.L.: Biclustering Algorithms for Biological Data Analysis: A Survey. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 1 (2004) 24-45
3. Mitra, S., Banka, H.: Multi-objective evolutionary biclustering of gene expression data. Pattern Recognition, 39 (2006) 2464–2477
4. Divina, F., Aguilar-Ruiz, J.S.: Biclustering of Expression Data with Evolutionary Computation. IEEE Transactions on Knowledge and Data Engineering, 18, 5 (2006) 590- 602
5. Ponzoni, I., Azuaje, F., Augusto, J., Glass, D.: Inferring association rules between genes using a combinatorial optimization learning process and adaptive regulation thresholds. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 4 (2007) 624-634
6. Cheng, Y., Church, G.M.: Biclustering of Expression Data. Proceedings of the 8th International Conf. on Intelligent Systems for Molecular Biology, La Jolla, USA (2000) 93-103
7. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In Giannakoglou, Tsahalis, Periaux, Papailiou, and Fogarty (eds), Evolutionary Methods for Design, Optimisations, and Control, (2002) 19-26
8. Tavazoie, S., Hughes, J.D., Campbell, M., Cho, R.J., Church, G.M.: Systematic determination of genetic network architecture. Natural Genetics, 22 (1999) 281–285
9. Alizadeh, A.A., Eisen, M.B., Davis, R.E., Ma, C., Lossos, I.S., Rosenwald, A., Boldrick, J.C., Sabet, H., Tran, T., Yu, X., Powell, J.I., Yang, L., Marti, G.E., Moore, T., Hudson, J. Jr, Lu, L., Lewis, D.B., Tibshirani, R., Sherlock, G., Chan, W.C., Greiner, T.C., Weisenburger, D.D., Armitage, J.O., Warnke, R., Levy, R., Wilson, W., Grever, M.R., Byrd, J.C., Botstein, D., Brown, P.O., Staudt, LM. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. Nature. 403 (2000) 503-511
10. Bleuler, S., Prelic, A., Zitzler, E.: An EA framework for biclustering of gene expression data, in: Proceeding of Congress on Evolutionary Computation (2004) 166-173
11. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, Inc., New York, NY (2001)
12. Zitzler, E., Laumanns, M., Bleuler, S.: A Tutorial on Evolutionary Multiobjective Optimization (2004).