Microarray Biclustering: A Novel Memetic Approach Based on the PISA Platform

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
{cag,jac,ip}@cs.uns.edu.ar
[2] Planta Piloto de Ingeniería Química (PLAPIQUI) - UNS – CONICET
Complejo CRIBABB, Co. La Carrindanga km.7, CC 717, Bahía Blanca, Argentina

Abstract. In this paper, a new memetic approach that integrates a Multi-Objective Evolutionary Algorithm (MOEA) with local search for microarray biclustering is presented. The original features of this proposal are the consideration of opposite regulation and incorporation of a mechanism for tuning the balance between the size and row variance of the biclusters. The approach was developed according to the Platform and Programming Language Independent Interface for Search Algorithms (PISA) framework, thus achieving the possibility of testing and comparing several different memetic MOEAs. The performance of the MOEA strategy based on the SPEA2 performed better, and its resulting biclusters were compared with those obtained by a multi-objective approach recently published. The benchmarks were two datasets corresponding to *Saccharomyces cerevisiae* and *human B-cells Lymphoma*. Our proposal achieves a better proportion of coverage of the gene expression data matrix, and it also obtains biclusters with new features that the former existing evolutionary strategies can not detect.

Keywords: Gene regulation, biclustering, evolutionary algorithms, PISA.

1 Introduction

The study of complex interactions between macro-molecules during transcription and translation processes constitutes a challenging research field, since it has a great impact in various critical areas. In this context, the microarray technology arose as a fundamental tool to provide information about the behavior of thousands of genes. The information provided by this technology corresponds to the relative abundance of the mRNA of genes under a given experimental condition. The abundance of the mRNA is a metric that can be associated to the expression level of the gene. This information can be arranged into a matrix, namely gene expression data matrix, where rows and columns correspond to genes and experiments respectively. Each matrix entry is a real number that represents the expression level of a given gene under a given condition.

An important issue in gene expression data analysis consists in grouping genes that present a similar, or related, behavior according to their expression levels. The

C. Pizzuti, M.D. Ritchie, and M. Giacobini (Eds.): EvoBIO 2009, LNCS 5483, pp. 44–55, 2009.

achievement of this task helps in inferring the functional role of genes during protein transcription. Therefore, based on the data about the relations between genes and their products, the gene regulatory networks (GRNs) can be discovered.

In general, during the process of identifying gene clusters, all of the 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 leaded the attention to the design of biclustering methods that simultaneously group genes and samples [1]. In this context, a satisfactory bicluster consists in a group of rows and columns of the gene expression data matrix that satisfies some similarity score [2] in conjunction with other criteria.

In this paper, we propose a memetic multi-objective evolutionary approach implemented in the context of the PISA platform to solve the problem of microarray biclustering. Our technique hybridizes traditional multi-objective evolutionary algorithms (MOEAs) with a new version of a well-known Local Search (LS) procedure. To the best of our knowledge, this methodology introduces two novel features that were never addressed, or partially dealt-with, by other evolutionary techniques designed for this problem instance. The first contribution consists in the design of the individual representation that contemplates the mechanisms of opposite regulation. The other new characteristic is the incorporation of a mechanism that controls the trade-off between size and row variance of the biclusters. The rest of the paper is organized as follows: in the next section some concepts about microarray biclustering are defined; then, a brief review on existing evolutionary methods used to tackle this problem is presented; in Section 4 our proposal is introduced; then, in Section 5, all the experimental framework and the results are put forward; finally some conclusions are discussed.

2 Microarray Biclustering

As it was aforementioned, expression data can be viewed as a matrix **E** that contains expression values, where rows correspond to genes and columns to the samples or *conditions*, taken at different experiments. A matrix element e_{ij} contains the measured expression value for the corresponding gene i and sample j. In this context, a bicluster is defined as a pair (G, C) where $G \subseteq \{1, \ldots, \mathrm{m}\}$ is a subset of genes (rows) and $C \subseteq \{1, \ldots, \mathrm{n}\}$ is a subset of conditions (columns) [2]. In general, the main goal is to find the largest bicluster 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 [2]. Therefore, our optimization problem can be defined as follows:
maximize

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

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

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

subject to

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

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 a given value δ, then we call the bicluster a δ-bicluster. The problem of finding the largest square δ-bicluster is NP-hard [2]. The high complexity of this problem has motivated researchers to apply various approximation techniques to generate near optimal solutions. In particular, evolutionary algorithms (EAs) are well-suited for addressing this class of problems [3, 4, 5].

3 Microarray Biclustering with Evolutionary Algorithms

The first reported approach that tackled microarray biclustering by means of an EA was proposed by Bleuler *et al.* [5]. In this work, several variants are presented. They analyze the use of a single-objective EA, an EA combined with a LS strategy [2] and the LS strategy alone [2]. In the case of the EA, one novelty consists in a form of diversity maintenance that can be applied during the selection procedure. For the case of the EA hybridized with a LS strategy, they consider whether the new individual yielded by the LS procedure should replace the original individual (*Lamarckian* approach) or not (*Baldwinian* approach). As regards the LS as a stand alone strategy, they propose a new non-deterministic version, where the decision on the course of execution is made according to some probability.

Regarding the EA, a binary representation for the individuals where each individual stands for a given bicluster is adopted, and independent bit mutation and uniform crossover are used. For the definition of the fitness function, they distinguish two cases: whether the EA operates alone or if it works together with the LS strategy. For the first situation a better fitness value, obtained from the size of the bicluster, is assigned to those individuals that comply with the residue restriction. If the bicluster has a residue over a given threshold, namely δ, then a value greater than 1 is set. For the second case, as the residue constraint is considered by the LS strategy, they only look at the size of the biclusters for the fitness assignment. For the experiments, two datasets were used: *Yeast* [6] and *Arabidopsis thaliana* [7, 8]. The study of the results is organized considering whether the aim is to get a unique bicluster or a set of biclusters. For the analysis of a single bicluster, the evaluation is focused on the size of the biclusters, and the algorithm that performed better was the EA combined with the LS method by means of an updating policy. For the second case of analysis, a comparison of the results as regards the covering of matrix **E** is performed, and the hybridized EA with diversity maintenance combined with LS did better in this sense.

Another approach, called SEBI for Sequential Evolutionary BIclustering, was later proposed by Divina and Aguilar-Ruiz [4]. In this work, an EA is presented where the individuals represent biclusters by means of binary strings. The main idea of this sequential technique is that the EA is run several times. From each run, the EA yields the best bicluster according to its size, row variance and overlapping factors. If its residue value (as defined by Chung and Church [2]) is lower than δ, then the bicluster is added into an archive that they call *Results*. Whenever this is the case, the method keeps track of the elements of the bicluster so as to use this information to minimize overlapping during the next run of the EA.

As regards the details of the EA, the fitness function combines the aforementioned objectives by means of a non-Pareto aggregative function. Tournament selection is chosen and several options for the recombination operators were implemented. For the experimental studies, the EA was executed for two datasets: *Yeast* [6] and *Human B-cells* [9]. The comparison is performed against the biclusters found by Chung and Church as regards the covering of the whole gene expression matrix **E**. For the *Yeast* dataset, SEBI manages to cover 38% of **E**, while Chung and Church's covers 81%. Regarding the *Human* dataset, SEBI covers 34% while Chung and Church's biclusters cover 37%. The authors consider that these results can be explained as a consequence of the overlapping factor, since the consideration of this objective naturally goes in detriment of the other goals.

Finally, Mitra and Banka [3] present a MOEA combined with a LS [2] strategy. This method constitutes the first approach that implements a MOEA based on Pareto dominancy for this problem. The authors base their work on the NSGA-II, and look for biclusters with maximum size and homogeneity. The individual representation is the same as in the previously introduced methods; and uniform single-point crossover, single-bit mutation and crowded tournament selection are implemented. The LS strategy is applied to all of the individuals with a *Lamarkian* approach, at the beginning of every generational loop. The method is tested on microarray data consisting of two benchmark gene expression datasets, *Yeast* and *Human B-cell Lymphoma*. For the analysis of the results, a new measure called Coherence Index (CI) is introduced. CI is defined as "the ratio of mean squared residue score to the size of the formed bicluster". The biclusters are compared to those reported by Chung and Church and, in all the cases, Mitra and Banka's results indicate a better performance in terms of the bicluster size, while satisfying the homogeneity criterion in terms of δ. However, as regards coverage, Chung and Church's work produces better results.

4 Our Proposal

The aim of our study is to use a MOEA 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 need to guide the search to the area where this restriction is accomplished. In that context, we apply a LS technique based on Chung and Church's procedure after each generation, thus orienting the exploration and speeding up the convergence of the MOEA by refining the chromosomes. Besides, the results achieved by other authors [3, 5] reveal that MOEAs alone obtain poor biclusters.

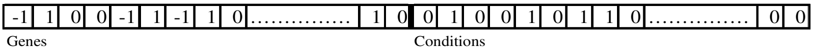
In order to consider inverted rows, we have extended the classical representation of a bicluster and we have also modified the genetic operators. Then, our proposal performs over a double-sized search space, in contrast with the evolutionary biclustering methods found in the literature [3, 4, 5]. The importance of including 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 [2]. In this way, our proposal is able to find biclusters that the former evolutionary methods cannot detect.

As regard to the implementation, the multi-objective strategy was built on the base of a platform called PISA [10]. PISA is a text-based interface for search algorithms. It splits an optimization process into two modules. One module contains all the parts that are specific to the optimization problem (e.g., evaluation of solutions, problem representation, and variation of solutions) and is called the *Variator*. The other module contains the parts of an optimization process which are independent of the optimization problem (mainly the selection process). This part is called the *Selector*. These two modules are implemented as separate programs which communicate through text files.

For this work, we have designed a *Variator* specific for the microarray biclustering application, and we have combined it with the *Selectors* corresponding to the IBEA [11], NSGAII [12] and SPEA2 [13] optimization algorithms. The reason for the selection of these MOEAs is that they are the most recommended evolutionary optimizers in the literature. In this way, we will assess the MOEA that exhibits the best performance for the problem. In the following sections, we will describe the main features of the implemented *Variator* and how the LS is incorporated into the search process.

Individual's Representation

Each individual represents one bicluster, which is encoded by a fixed size string built by appending a string for genes with another bit string for conditions. The individual corresponds to a solution for the problem of optimal bicluster generation. If a string position (*locus*) 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 also considerer the addition of negative values in the string for genes. That is to say, a *locus* of the string is set to -1 when the relative inverted row belongs to the encoded solution. Figure 1 shows an example of such encoding for a random individual.



-1 1 0 0 -1 1 -1 1 0 .... 1 0 0 1 0 0 1 0 1 1 0 .... 0 0

Fig. 1. An encoded individual representing a bicluster

Genetic Operators

It is important to give a brief description of the genetic operators, since they have a key influence in how the search is performed by the MOEA.

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 string is selected at random, then proceeding to alter the *locus* in question. If the resulting position is a column, the corresponding *locus* is simply complemented. On the other hand if the resulting position is a row, then we have two cases: if the *locus* is set to 0 then it is set to 1, and the sign is determined with a probability of 0.5. If the *locus* is in on (1 or -1) we simply change the value to 0.

Recombination. A two-point crossover was implemented 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 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.

Multi-Objective Fitness Function

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 multi-objective modeling. In that context, we decided to optimize the objectives defined by equations 1- 4 (see *Section 3*): 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.

Local Search

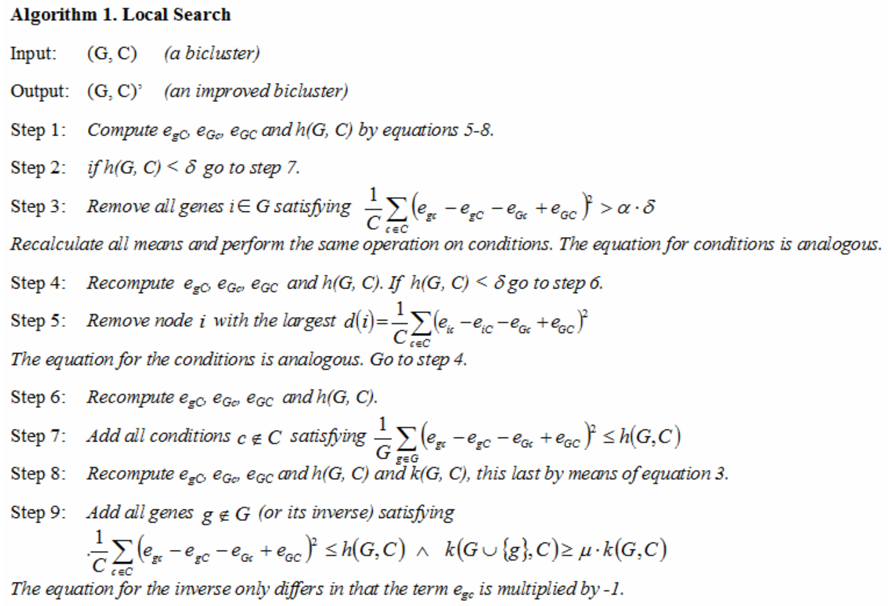
This subsection describes the LS procedure that hybridizes the selected MOEAs. The LS is applied into the *Variator* side to the biclusters that are selected by the *Selector* as the resulting individuals of each generation. Adding the LS to the *Variator* is the only way to hybridize a MOEA without altering the basic principles of PISA [10]. The greedy approach is based on Chung and Church's work [2], 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 Algorithm 1.

The main differences with Chung and Church's implementation are the following:

- In Step 3, we remove multiple nodes considering a different threshold, $\alpha.\delta$ instead of $\alpha.h(G,C)$. As a consequence, Step 5 is performed a smaller number of times with respect to the original proposal. 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 9, we incorporated the row variance, adding 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 column and each inverted row, in that order. In our case, we first attempt to add each condition. 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 if it contains less quantity of rows.

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 $\mu = 0$, this step results equivalent to that of the original proposal.



Algorithm 1. 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 7.
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 node $i$ with the largest $d(i) = \frac{1}{C}\sum_{c \in C}\left(e_{ic} - e_{iC} - e_{Gc} + e_{GC}\right)^2$
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 $e_{gC}$, $e_{Gc}$, $e_{GC}$ and 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
         $\frac{1}{C}\sum_{c \in C}\left(e_{gc} - e_{gC} - e_{Gc} + e_{GC}\right)^2 \leq h(G,C) \wedge k(G \cup \{g\}, C) \geq \mu \cdot k(G,C)$
The equation for the inverse only differs in that the term $e_{gc}$ is multiplied by -1.
```

5 Experimental Framework and Results

Two different goals were established for our study. First we need to determine which of the memetic MOEAs performs better in this class of problem. The analysis will be performed with the tools provided in Knowles *et al.* [14]. Then, we will compare the selected memetic evolutionary algorithm with the approach of Mitra and Banka [3] since, to the best of our knowledge, this is the only multi-objective evolutionary method for microarray biclustering found in the literature.

Performance Assessment

As it was aforementioned, the choice of the best memetic MOEA will be based on the results of the tools provided in Knowles *et al.* [14], which are well recognized in the area of multi-objective optimization. The metrics applied in the evaluation of the MOEAs are the *Dominance Ranking* [14], the *Hypervolumen Indicator* I_H^- [15], the multiplicative version of the *Unary Epsilon Indicator* I_ε^1 [16], and the *R2 Indicator* I_{R2}^1 [17]. The *Dominance Ranking* is useful in the assessment of quite general statements about the relative performance of the considered optimizers, since it merely relies on the concept of Pareto dominance and some ranking procedure. The *Quality Indicator I* measures the number of goals that have been attained for the optimizers being under consideration. Each one of the indicators empathize a different aspect in the preference of the solutions obtained. For the details of the previous metrics please be referred to Knowles *et al.* [14].

If a significant difference can be demonstrated using the *Dominance Rank*, the only purpose of the *Quality Indicators* is to characterize further differences in the approximation sets. On the other hand, if we cannot establish significant differences by means of the *Dominance Rank*, then the *Quality Indicators* can help us in the decision of which one of the optimizers is better. However, these results do not confirm that the selected method generates the better approximation sets. In order to make inferences about the results of the previous metrics we will apply the *kruskal-wallis test* [18], since more than two methods are tested [14].

For this analysis, we have used two microarray datasets, the *Saccharomyces cerevisiae* cell cycle expression data from [6] 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 mRNA abundance. All values are integers in the range between 0 and 600 replacing the missing values by 0. The *Lymphoma* dataset contains 4.026 genes and 96 conditions. The expression levels are integers in the range between -750 and 650, where the missing values were also replaced by 0. These datasets have been directly used as in [2].

First Experimental Phase

The three memetic MOEAs, IBEA, NSGA-II and SPEA2, have been evaluated with 50 runs and 75 generations over the two datasets. The Table 1 summarizes the parameters used in this benchmark. These values were selected from a few preliminary runs. In the case of the LS setup, δ was set with the same value as in [2], μ was set for the best tradeoff between size and row variance, and α was set considering the overall efficiency on each dataset. All the executions were controlled by the *Monitor* module [10]. In the case of the IBEA algorithm, we chose the *Additive Epsilon Indicator*, and the rest of the parameters were set to the default values. Since PISA assumes that all the objectives are minimized, the four objectives of our approach (see *equations 1-4*) were adapted accordingly. For the parameters of the indicators, we maintained the default values of the nadir and ideal points (appropriately extended to four objectives), since the objectives are automatically normalized to the interval [1..2] by the tools.

Table 1. Parameter's settings for this study

		Generations	Mutation Prob.	Crossover Prob.	δ	α	μ
Our variator	*Yeast*	75	0,3	0,9	300	1,8	0,998
	Lymphoma				1200	1,5	0,999

PISA	α	μ	λ
	100	50	50

Table 2. *Kruskal-wallis test* over the *Quality Indicators* I_H^- (left), I_ε^1 (middle) and I_{R2}^1 (right)

Hypervolumen Indicator

	IBEA	SPEA2	NSGA-II
IBEA	-	1	0,99
SPEA2	1,50E-07	-	0,16
NSGA2	1,09E-05	0,84	-

Multiplicative Epsilon Indicator

	IBEA	SPEA2	NSGA-II
IBEA	-	0,99	0,99
SPEA2	2,10E-04	-	0,61
NSGA2	2,10E-04	0,39	-

R2 Indicator

	IBEA	SPEA2	NSGA-II
IBEA	-	1	0,99
SPEA2	8,00E-09	-	0,09
NSGA2	4,00E-06	0,91	-

Table 3. Average of the objective's values of IBEA, SPEA2 and NSGA-II on the *Yeast* dataset (above) and on the *Lymphoma* dataset (below).

Yeast dataset

	average rows	average columns	average residue	average variance	average size
IBEA	1047,63	12,52	261,61	296,35	13116,33
SPEA2	794,59	10,37	224,31	296,47	8239,898
NSGA-II	646,34	9,92	204,75	236,04	6411,693

Lymphoma dataset

	average rows	average columns	average residue	average variance	average size
IBEA	655,93	60,71	1089,61	1135,93	39821,51
SPEA2	727,74	52,63	1048,91	1112,03	38300,96
NSGA-II	583,8	54,34	1046,68	1061,7	31723,69

As regards the experimental results, the *kruskal-wallis test* can not detect significant differences on the *Dominace Ranking* of the three MOEAs, assuming a statistically significant level α = 0.05. This situation is equal for both datasets. In fact, all the results of the executions are assigned to the higher rank, showing that none of the MOEAs generates better approximation sets with respect to the others. This demonstrates the high influence in the search process of the LS and how it guides the MOEAs to the same areas on the search space. Table 2 shows, for the *Yeast* dataset, the results of the *kruskal-wallis test* over three *Quality Indicators*. The table contains, for each pair of optimizers O_R (row) and O_C (column), the p-values with respect to the alternative hypothesis that the *Quality Indicator I* is significantly better for O_R than for O_C. For the *Lymphoma* dataset, differences between the algorithms are discovered, but none of them are statistically significant (α = 0.05).

As it is shown in Table 2, both SPEA2 and NSGA-II perform better than IBEA under all the indicators, but the differences between SPEA2 and NSGA-II are not statistically significant. In view of these results, no asseveration can be made with respect to which one of the hybridized MOEAs performs better in this context.

At this point, we advised the need of applying an *ad hoc* strategy in order to select one of the algorithms, i.e., we will play the role of a decision maker. The Table 3 shows the average of the objective's values for the biclusters found by each memetic MOEA executed with the parameters shown in Table 1. It is clear that IBEA obtains biclusters of a bigger size (on average) with respect to those obtained by SPEA2 and NSGA-II. Moreover, NSGA-II constitutes the approach that obtains the most homogeneous biclusters and SPEA2 is the one that obtains the best relation between residue and row variance. This behavior becomes more evident for the *Yeast* dataset than for the *Lymphoma* dataset. It is important to notice that, in general, biclusters with higher size have higher residue and lower row variance; whereas biclusters with small residue have sizes that tend to be smaller, independently of the row variance. The row variance is at least bigger in value than the residue in all the cases.

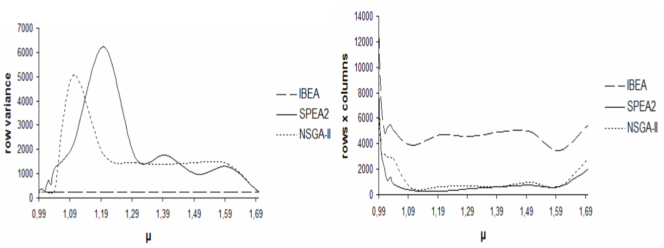


Fig. 2. Average row variance (left) and average size (right) for IBEA, NSGA-II and SPEA2 on the *Yeast* dataset when □ parameter varies between 0.99 and 1.7

Another characteristic of the MOEAs that can help on the selection of the method is constituted by how well the search can be oriented by means of the μ parameter on the LS. The Figure 2 shows, for the *Yeast* dataset, the average row variance (left) and the average size (right) of the biclusters obtained by each hybrid method when the μ parameter varies between 0.99 and 1.7. This threshold is related with the values of μ that have more effect over the results. As we can see, both SPEA2 and NSGA-II are well suited for being guided by the μ parameter, since as we increase the value of μ, the resulting biclusters have higher row variance in detriment of the size. In this regard, the SPEA2 is the algorithm with best performance. On the other hand, the only effect that can be observed on IBEA is the reduction of the size of the biclusters, since we can not observe any effect on the row variance. Perhaps this is due to the fact that IBEA is an Indicator-based MOEA whereas SPEA2 and NSGA-II are Pareto-based MOEAs. Therefore, a conjecture is that the small changes introduced by the μ parameter in the population in each generation are not well perceived by IBEA; probably because the concept of non-dominated solution set is not supported by the algorithm. The behavior observed on the *Lymphoma* dataset is similar.

For the comparative study of the next sub-section, we chose the memetic SPEA2 since it is more sensitive to the μ parameter than the others, whereas the average sizes of the biclusters are similar to those found by the memetic IBEA. Although IBEA can find some greater biclusters than SPEA2, it is not sensitive to the μ parameter.

All the testing has been made on a *Mobile Sempron* with 2 GB of RAM. The running time (on average) for the Yeast dataset was of 150s whereas for the *Lymphoma* dataset was of 660s. Since the execution time is mainly influenced by the LS procedure, the three MOEAs obtained these values.

Second Experimental Phase

A comparison between the memetic SPEA2 and Mitra and Banka's algorithm [3] is presented here. For this analysis, we used the results published by [3] in the paper. The parameter setups of our approach are those of the Table 1. The Table 4 shows the average results of the objective's values for the *Yeast* dataset for both approaches. The size of the largest bicluster found by each method and the coverage of the gene expression data matrix **E** are also shown. The row variance is not shown because in [3] it is not reported. As we can see, our proposal can obtain more homogeneous biclusters (on average) whereas the biclusters of Mitra and Banka's algorithm are bigger in size (on average). The largest bicluster found by the two methods is similar in

Table 4. Average objective's values for the bicluster found in the *Yeast* dataset by our memetic SPEA2 and Mitra and Banka's approach

	average rows	average columns	average residue	average size	largest bicluster size	coverage of cells
memetic SPEA2	794,59	10,37	224,31	8239,89	14602	72,50%
M&B's approach [3]	1095,43	9,29	234,87	10176,54	14828	51,34%

size. When we consider the coverage of **E**, our proposal obtains a significantly better coverage of cells with respect to Mitra and Banka's algorithm. It is important to remark that the biclusters found by our approach also include the inverted rows; therefore, the search is carried out over a doubled-size search space with regard to the other evolutionary methods for microarray biclustering found in the literature.

As regards to the *Lymphoma* dataset, in [3] the average results of the objective's values are not reported in the paper. For this dataset, they simply show the largest bicluster and the average coverage of **E**. In this regard, our proposal can find a bicluster greater than the one reported for Mitra and Banka's algorithm. This bicluster has 1009 genes, 63 conditions, a mean squared residue of 1181.06, a row variance of 1295.05, and a size of 63567; whereas the greatest bicluster that is reported in [3] is of a size of 37560. We can argue that, to the best of our knowledge, this bicluster is greater than any other bicluster found by any method reported in the existing literature. Also, the coverage of **E** achieved by our memetic SPEA2 is (on average) about 33.58% of cells; significantly better than the average of 20.96% obtained by Mitra and Banka's algorithm.

6 Conclusions

In this paper, we have introduced a general multi-objective framework for microarray biclustering hybridized with a LS procedure for finer tuning. In a first experimental phase, we have hybridized and compared three well known MOEAs (IBEA, SPEA2 and NSGA-II) based on the PISA platform, in order to establish which one obtains the best results. Since no conclusive result was obtained from this evaluation, we selected the SPEA2 since it was able to obtain relatively large biclusters with a high sensitivity to the μ parameter. Then, during a second experimental phase, we have demonstrated that the quality of the outcomes of the memetic SPEA2 outperformed the results reported by Mitra and Banka. The comparative assessment was carried out on two benchmark gene expression datasets to demonstrate the effectiveness of the proposed method.

Moreover, we provide to the biological scientists with an extra parameter to determine which biclusters they consider more relevant, giving them the possibility of adjusting the size and the row variance of the biclusters. Furthermore, the evolutionary approaches for biclustering found in the literature do not consider the inclusion of inverted rows, perhaps for efficiency reasons since the search space is duplicated. However, these inverted rows are very important because, they can be interpreted as co-regulated by receiving the opposite regulation. In this context, we have also demonstrated that it possible to take into account 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. They would also like to acknowledge SeCyT (UNS) for Grant PGI 24/ZN15.

References

1. Madeira, S., Oliveira, A.L.: Biclustering Algorithms for Biological Data Analysis: A Survey. IEEE-ACM Trans. Comput. Biol. Bioinform. 1, 24–45 (2004)
2. Cheng, Y., Church, G.M.: Biclustering of Expression Data. In: Proceedings of the 8th International Conf. on Intelligent Systems for Molecular Biology, pp. 93–103 (2000)
3. Mitra, S., Banka, H.: Multi-objective evolutionary biclustering of gene expression data. Pattern Recognit. 39, 2464–2477 (2006)
4. Divina, F., Aguilar-Ruiz, J.S.: Biclustering of Expression Data with Evolutionary Computation. IEEE Trans. Knowl. Data Eng. 18(5), 590–602 (2006)
5. Bleuler, S., Prelic, A., Zitzler, E.: An EA framework for biclustering of gene expression data. In: Proceeding of Congress on Evolutionary Computation, pp. 166–173 (2004)
6. Cho, R., et al.: A genome-wide transcriptional analysis of the mitotic cell cycle. Mol. Cell 2(1), 65–73 (1998)
7. Menges, M., Hennig, L., Gruissem, W., Murray, J.: Genome-wide gene expression in an Arabidopsis cell suspension. Plant Mol. Biol. 53(4), 423–442 (2003)
8. Laule, O., et al.: Crosstalk between cytosolic and plastidial pathways of isoprenoid bio systhesis in arabidopsis thaliana. PNAS 100(11), 6866–6871 (2003)
9. Alizadeh, A.A., et al.: Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. Nature 403, 503–511 (2000)
10. Bleuler, S., Laumanns, M., Thiele, L., Zitzler, E.: PISA - A Platform and Programming Language Independent Interface for Search Algorithms. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) EMO 2003. LNCS, vol. 2632, pp. 494–508. Springer, Heidelberg (2003)
11. Zitzler, E., Künzli, S.: Indicator-Based Selection in Multiobjective Search. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 832–842. Springer, Heidelberg (2004)
12. Deb, K., Agraval, S., Pratap, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) PPSN 2000. LNCS, vol. 1917, pp. 849–858. Springer, Heidelberg (2000)
13. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In: Giannakoglou, Tsahalis, Periaux, Papailiou, Fogarty (eds.) Evolutionary Methods for Design, Optimisations and Control, pp. 19–26 (2002)
14. Knowles, J., Thiele, L., Zitzler, E.: A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers. TIK Computer Engineering and Networks Laboratory (2005)
15. Zitzler, E., Thiele, L.: Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. IEEE Trans. Evol. Comput. 3(4), 257–271 (1999)
16. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C., Grunert da Fonseca, V.: Performance Assessment of Multiobjective Optimizers: An Analysis and Review. IEEE Trans. Evol. Comput. 7(2), 117–132 (2003)
17. Hansen, M., Jaszkiewicz, A.: Evaluating the quality of approximations to the non-dominated set. Technical University of Denmark (1998)
18. Conover, W.: Practical Nonparametric Statistics. John Wiley & Sons, New York (1999)