# A Novel Probabilistic Encoding for EAs Applied to Biclustering of Microarray Data

Marcozzi, Michaël; DIVINA, Federico; AGUILAR-RUIZ, Jesús S.; Vanhoof, Wim

*Published in:*
GECCO '11

*DOI:*
10.1145/2001576.2001623

*Publication date:*
2011

*Document Version*
Peer reviewed version

# A Novel Probabilistic Encoding for EAs Applied to Biclustering of Microarray Data

Michaël Marcozzi*
Faculty of Computer Science
University of Namur
Namur, Belgium
mmr@info.fundp.ac.be

Federico Divina
School of Engineering
Pablo de Olavide University
Seville, Spain
fdivina@upo.es

Jesús S. Aguilar-Ruiz
School of Engineering
Pablo de Olavide University
Seville, Spain
aguilar@upo.es

Wim Vanhoof
Faculty of Computer Science
University of Namur
Namur, Belgium
wva@info.fundp.ac.be

## ABSTRACT

In this paper we propose a novel representation scheme, called probabilistic encoding. In this representation, each gene of an individual represents the probability that a certain trait of a given problem has to belong to the solution. This allows to deal with uncertainty that can be present in an optimization problem, and grant more exploration capability to an evolutionary algorithm. With this encoding, the search is not restricted to points of the search space. Instead, whole regions are searched, with the aim of individuating a promising region, i.e., a region that contains the optimal solution. This implies that a strategy for searching the individuated region has to be adopted. In this paper we incorporate the probabilistic encoding into a multi-objective and multi-modal evolutionary algorithm. The algorithm returns a promising region, which is then searched by using simulated annealing. We apply our proposal to the problem of discovering biclusters in microarray data. Results confirm the validity of our proposal.

## Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*; I.5.3 [**Pattern Recognition**]: Clustering—*Algorithms*; J.3 [**Computer Applications**]: Life and medical sciences—*Biology and genetics, Health*

## General Terms

Algorithms

*F.R.S.-FNRS Research Fellow


*GECCO'11,* July 12–16, 2011, Dublin, Ireland.


## Keywords

Probabilistic Encoding, Multi-Modal and Multi-Objective Evolutionary Computation, Simulated Annealing, Biclustering, Microarray Data

## 1. INTRODUCTION

Microarray is a technology that allows measuring the expression level of thousands of genes under hundreds of different experimental conditions. The obtained data is usually organized in a matrix, called expression matrix, and have enormous potential in gene profiling, facilitating the prognosis and the discovering of subtypes of diseases [5]. Biclustering [19] of the expression matrix is one of the most used techniques for analyzing these kind of data. When a group of genes present an expression level that shows strikingly similar up-regulation and down-regulation under a set of conditions [8], it is a hint of a possible biological correlation among these genes under this set of conditions. With biclustering, one aims at individuating such subsets of potentially correlated genes and conditions in the matrix, commonly named biclusters. Various techniques have been used in order to discover meaningful biclusters in microarray data [19], and among these, Evolutionary Algorithms (EAs) has obtained encouraging results, see, e.g., [6, 11, 20]. Evolutionary Computation (EC) proposes a generic heuristic mechanism to solve combinatorial optimization problems, inspired by the neo-darwinian evolution process. An important and difficult aspect of the design of an EA is to define a formal encoding for the candidate solutions of the optimization problem [13]. The representation of solutions is important as it plays a similar role in an EA as the role played by genetic code in neo-darwinian evolution.

Biologists often refer to the genetic code of an individual as its genotype, and to the set of traits of the individual as its phenotype. The genotype of an individual determines its phenotype, and the phenotype determines the fitness of the individual in a given environment. Evolution occurs from recombinations and mutations of existing genotypes, and from selection of the fittest resulting phenotypes. The choice of a "right" encoding for the considered optimization problem is thus a particularly hard and sensitive aspect in the im-

plementation of an EA. Choosing a particular encoding will indeed condition the way new potentially better solutions can be generated by the EA, and influence thus the performance and the efficiency of the EA for exploring the search space.

It is now often admitted that there is no universal encoding mode that is suitable for every optimization problem [4]. Nevertheless, there exist many standard encoding schemes studied in the literature (see, e.g., [4]). Such schemes have been applied to many different problems, and specific genetic operators have been developed and tested for each kind of representation. One of the simplest and most established encoding schemes is represented by binary strings. Each candidate solution in the search space is represented by a different string of bits, typically of identical length. The use of this binary encoding mode is convenient for problems whose solutions can be naturally modeled using a set of boolean decision variables, e.g. biclustering. Often, however, the problem to solve is characterized by uncertainty. In such a case, a binary encoding may not be the best approach, since it will not allow to represent the implicit uncertainty of the problem. Biclustering is a problem that is characterized by a certain amount of uncertainty that can be due to various reasons, for example the presence of missing values and noise in the expression matrix. Other reasons will be discussed in the remainder of the paper.

This observation motivates us to introduce a new encoding paradigm, called probabilistic encoding. This novel representation extends the traditional binary encoding mode, since probabilistic encoding allows to represent the uncertainty that can exist over the truth value of some boolean decision variables in the optimal solutions of some optimization problems. Probabilistic encoding can also potentially increase the exploration power of evolutionary algorithms, compared to classical binary encoding. Other approaches, in particular Estimation of Distribution Algorithms (EDAs) [18] have been applied when there is a certain amount of uncertainty in the problem. Our approach differentiates from EDAs since it relies on genetic operators to evolve the population, following the classical EAs scheme. Recently, EDAs have been applied to biclustering [14], where EDAs was combined with the NSGA-II multi-objective approach.

In order to test our proposal on the biclustering problem, we incorporate the probabilistic encoding in a multi-objective and multi-modal EA. With this, we aim at discovering promising regions, i.e., regions containing interesting biclusters. Such regions will then be exploited, by means of a local search algorithm, in order to identify good biclusters. This strategy would allows us to deal with the intrinsic uncertainty that characterize the biclustering problem.

This paper is organized as follows: in Section 2 the new probabilistic encoding is introduced, section 3 provides the basic concepts of biclustering of gene expression data and motivates our application of probabilistic encoding to the biclustering problem. Section 4 describes the algorithm used, while experiments and results are discussed in section 5. Finally, some conclusions and future work are provided in section 6.

## 2. PROBABILISTIC ENCODING

In some optimization problems, each candidate solution can be naturally and efficiently modeled as a particular valuation of a single set of boolean decision variables (BDV). Classical examples of such problems are all the existing instances of the maximum-independent-set problem in graphs, the set covering problem and the knapsack problem [4]. Formally, we can define a BDV optimization problem in the following way:

DEFINITION 1. *An optimization problem $\Phi$ defined on a search space $S$ is a **BDV optimization problem** iff $\exists$ $V$, a set of $n$ boolean variables $v_i$, such that $\forall$ candidate solution $s \in S$, $s$ can be naturally and efficiently modeled as a unique valuation of the variables in $V$: $s \equiv \{v_1 = b_1^s, ..., v_i = b_i^s, ...v_n = b_n^s\}$ (with $b_i^s \in \{false, true\}, \forall i \in [1, n]$)*

We can see each boolean decision variable $v_i$ in a BDV problem as indicating the presence of the $i^{th}$ of the $n$ possible traits that can be exhibited by the candidate solutions of the problem. A particular candidate solution, i.e., a particular phenotype, can be defined by choosing a particular valuation of all the decision variables, i.e. by specifying which of the possible traits it possesses and which ones it does not. BDV optimization problems are considered as the context to apply the standard binary encoding as representation [13]. Each bit in a genotype will indicate the presence or not of its associated trait in the represented solution.

On the other hand, in some optimization problems, it is not possible to evaluate the quality of the solutions without a given level of uncertainty [26]. This typically implies that the objective function of the problem will exhibit a limited level of precision in its quality assessment of solutions. As a consequence, the algorithm may not always be able to distinguish between the quality of some different solutions. In such a case, only an optimal region containing a set of good solutions, typically sharing a set of interesting characteristics, can be individuated. However, the exact location of the optimal solution within the region remains uncertain.

Uncertainty can arise for many reasons. In some cases, uncertainty is inherent to the problem, for example, the quality measurement of the solutions is subject to noise and tolerances [3], or it can vary dynamically in an unpredictable manner across the time [22]. In other cases, uncertainty is introduced deliberately, for example when approximations of the quality criteria are used to reduce the computational burden [21].

In a BDV optimization problem, such an uncertainty over the quality measurement of the solutions can mean uncertainty of the impact that the presence of some traits can have over the quality of the optimal solution. In such a case, binary encoding becomes no longer a suitable option to represent such uncertain solutions. In fact, with binary encoding, the presence of all of the possible traits has indeed to be specified in an exhaustive and certain way. In uncertain BDV problems, it should be possible for the EA to specify the presence of the traits in the solutions with a given level of uncertainty. In this work, we propose thus to replace the standard binary encoding with a probabilistic encoding as representation for uncertain BDV problems. In a nutshell, probabilistic encoding replaces the bits of binary encoding with probabilities. Thus, each element of the genotype specifies the probability of the presence of its associated trait in the solution.

It should however be remarked that replacing bits with probabilities totally modifies what genotypes represent. With binary encoding, each genotype represents a single and precise solution. On the other hand, many different solutions

can be represented by a single genotype expressed with probabilistic encoding, and such a genotype represents thus a region of solutions in the search space. Formally, a probabilistic genotype can be seen as a traits probability function:

DEFINITION 2. *A **traits probability function** $p_{traits}$ of a BDV optimization problem $\Phi$ is a total discrete function that associates to each decision variable $v_i$ of the problem a probability value.*

The probability for each solution of the search space to be represented by a probabilistic genotype (and thus to be part of the region it represents) can be computed from its set of traits, and from the probability of presence of the traits in the genotype:

DEFINITION 3. *The **probability for the solution** $s \equiv \{v_1 = b_1^s, ..., v_i = b_i^s, ...v_n = b_n^s\}$ of a BDV optimization problem $\Phi$ **to be represented by a traits probability function** $p_{traits}$ is:*

$$
\begin{aligned}
P(s) &= P(v_1 = b_1^s) * ... * P(v_i = b_i^s) * ... * P(v_n = b_n^s) \\
&\quad \textit{(Replacing the } b_i s \textit{ by particular values...)} \\
&= P(v_1 = true) * ... * P(v_i = false) * ... * P(v_n = true) \\
&= p_{traits}(v_1) * ... * (1 - p_{traits}(v_i)) * ... * p_{traits}(v_n) \\
&\quad \textit{(Thus, more generally...)} \\
&= \underbrace{\left[\prod_{i|b_i^s = true} p_{traits}(v_i)\right]}_{\textit{Conjoint probability of presence of the traits in } s} \\
&\quad * \underbrace{\left[\prod_{j|b_j^s = false} (1 - p_{traits}(v_j))\right]}_{\textit{Conjoint probability of absence of the traits not in } s}
\end{aligned}
$$

Now that we have established how probabilistic encoding allows to represent regions of the search space instead of single solutions, the interest of probabilistic encoding for uncertain BDV problems becomes even more evident. Probabilistic encoding allows an EA to individuate a promising region of the search space, when the exact position of the optimum is uncertain. Concretely, the EA should evolve the population towards individuals where the probability associated to one trait measures its frequency of appearance in the optimal region to discover. If we suppose that the solutions in this optimal region share an important set of common characteristics, i.e. share and avoid a same important set of traits, these traits should see their probability converge respectively to 1 and 0. For the remaining traits, whose interest cannot be certainly assessed, the algorithm gives an estimated interest rate, which is the ratio of optimal solutions that exhibit this trait.

As an example, suppose that the problem is to find the optimal combination of three objects $A, B$ and $C$. If a binary representation were to be used, an example of individual could be "1,1,0", which represents the solution $\{A, B\}$. Instead, if probabilistic encoding were used, an example of individual could be "0.95 0.52 0.04". This individual identifies a region of the search space, and table 1 gives the probability for each solution in the search space to be represented by this genotype. If an EA converges to a population composed of genotypes similar to the one presented here, it indicates that the optimal solution should contain the item A ($P(A = true) = 0.95$) but not the item C ($P(C = true) = 0.04$), while there is a high level of uncertainty over the presence of B in the solution ($P(B = true) = 0.52$). The optimal region individuated by the algorithm is composed of the two solutions $\{A\}$ and $\{A, B\}$, as they have a much higher probability to be represented by the genotype.

**Table 1: Probabilities of each possible solution to belong to the region identified by individual "**0.95 0.52 0.04**".**

| $s \in S$ | $P(s)$ |
|---|---|
| $\emptyset$ | 0.02304 |
| $\{A\}$ | 0.43776 |
| $\{B\}$ | 0.02496 |
| $\{C\}$ | 0.00096 |
| $\{A, B\}$ | 0.47424 |
| $\{A, C\}$ | 0.01824 |
| $\{B, C\}$ | 0.00104 |
| $\{A, B, C\}$ | 0.01976 |

Independently of being more suitable to deal with uncertain problems, probabilistic encoding might offer another advantage over binary encoding: a better search space exploration power. As each individual evaluated by a probabilistic EA can represent a potentially large region of solutions, the algorithm can potentially explore at each generation a much wider part of the search space than it could do by adopting a binary encoding.

Using probabilistic encoding will of course impact the other implementation choices that have to be made while designing a particular algorithm. The main algorithmic components that will have to be adapted to deal with probabilistic encoding are the genetic operators. We will propose in section 4 a set of operators designed for the biclustering problem.

The evaluation mechanism of the candidate solutions quality should also be adapted. In fact, the algorithm has to evaluate the quality of probabilistically defined and potentially large regions of solutions, instead of single candidate solutions. In this work, we propose to measure the quality of a region using the average fitness of sample of solutions representative of the region. This sample set will be populated so that the number of times a solution has a chance to appear in the set is equal to its probability to be represented by the genotype defining the region. The size of the set will of course have to be sufficient to offer a proper sampling, and should thus be determined by the size of the region, i.e., the number of different solutions that have a significant relative probability to be represented by the genotype. Concretely, we propose to create each sample solution in the following way: for each possible trait $i$, we build the sample solution by picking a random number $r$ in $[0, 1)$ and comparing it to the probability $p_i$ associated to the trait $i$ in the probabilities string specifying the region. If $r < p_i$ the trait is attributed to the sample solution, and the other way round.

# 3. BICLUSTERING MICROARRAY DATA

As already mentioned, microarray experiments data are usually organized in an expression matrix $EM$. Typically, each element $EM_{ij}$ of $EM$ will indicate the level of expression of gene $i$ under condition $j$, measured in the experiment.

Within the framework of this work, a bicluster is essentially a sub-matrix of $EM$, defined by a subset $G$ of $g$ genes and by a subset $C$ of $c$ conditions, and biclustering is modeled as a $BDV$ optimization problem, where the set of possible traits to build a candidate solution is the set $I$ of genes and $J$ of conditions involved in the microarray experiment. Biclustering is then typically an NP-complete [8, 19] combinatorial problem, where the solutions are combinations between the genes and conditions of $EM$. As a consequence, the size of the expression matrix in real experimental datasets typically makes the problem intractable to solve using exact methods, and requires the use of efficient heuristic methods.

A key aspect is how to assess the quality of a bicluster. One of the most popular functions used for to this aim is the mean squared residue [8]:

DEFINITION 4. *The **mean squared residue** ($MSR$) of a bicluster $(G, C)$ of an expression matrix $EM$ is defined as:*

$$MSR((G,C)) = \frac{1}{g*c}\sum_{k\in G}\sum_{l\in C}((G,C)_{kl} - \mathcal{M}_k - \mathcal{M}^l + \mathcal{M})^2$$

*where $\mathcal{M}_k$, $\mathcal{M}^l$ and $\mathcal{M}$ are the averages of row $k$ of $(G,C)$, column $l$ of $(G,C)$ and of the whole bicluster, respectively.*

$MSR$ measures how much the expression level of the genes in the bicluster varies in a coherent manner under the same set of conditions. The lower the value of the $MSR$ and the better the bicluster is considered. If a bicluster has $MSR$ equal to zero then it is a perfect bicluster. However, constant biclusters, i.e., flat biclusters, have $MSR$ equal to zero. Such biclusters are not interesting, and for discarding them, usually the mean row variance is used in combination with $MSR$, in order to measure how much the expression level of the genes varies under the same set of conditions.

DEFINITION 5. *The **mean row variance** ($MRV$) of a bicluster $(G, C)$ of an expression matrix $EM$ is defined as:*

$$MRV((G,C)) = \frac{1}{g*c}\sum_{k\in G}\sum_{l\in C}((G,C)_{kl} - \mathcal{M}_k)^2$$

In addition to $MSR$ and $MRV$, it is also important to ensure that biclusters are of maximal size, i.e. that all the correlated genes and conditions of $EM$ that constitute the meaningful bicluster have been individuated, without any exception.

DEFINITION 6. *The **size** of a bicluster $(G, C)$ of an expression matrix $EM$ is defined as: $SIZE((G,C)) = g * c$*

These measures can be combined into a single objective function. However, this may not be the best strategy, as these measures are in conflict with each other, so that a better approach is to treat the biclustering problem as a multi-objective problem [20].

Biclustering can also be described as a multi-modal problem [24]. In some multi-objective problems [27], the goal is to find a good sampling of the Pareto front of the problem, i.e., extracting all the best different kinds of compromises between the different objectives. In biclustering, we are more interested in finding several different biclusters, identifying different underlying biological relations, even if these different biclusters exhibit similar kinds of compromise between objectives [10]. Different biclusters will obviously involve different sets of genes and conditions, but which can have some genes and conditions in common [23, 24], i.e., different biclusters can overlap in the expression matrix. Dealing with multi-modality means thus controlling the overlapping among the individuated biclusters. The individuated biclusters should not overlap too much, in order to represent different biological relations between different genes and conditions, but overlapping should not be totally forbidden. Some approaches, e.g., [8], control this by replacing elements of the expression matrix that are already included in previously discovered biclusters with random values. However, in our opinion, this strategy presents various drawbacks. First, it prevents the discovery of overlapping biclusters. Secondly, after having discovered various biclusters, the expression matrix could consist mostly of random values, a fact that will severely penalize the discovering of successive biclusters [24].

Finally, biclustering is an optimization problem that can exhibit uncertainty over the quality measurement of its solutions. The reason for this uncertainty is that the three objective functions of the problem are typically conflicting. First, $MSR$ and $MRV$ can be conflicting, as a perfect bicluster can be totally flat. $MSR$ and size are also typically conflicting, as when a bicluster has a non-zero $MSR$, it is always possible to remove a gene or a condition to lower the MSR [8]. On the one hand, a multi-objective EA will then be able to identify a set of incomparable (in the sense of Pareto) biclusters, typically sharing a single core set of highly coherent and high variance genes and conditions, but representing different kinds of compromises among objectives. On the other hand, as the EA has no mean to choose which of these compromises should be favored, it will face a total uncertainty over which of the identified biclusters is the best. To solve this problem, [8] chooses a particular compromise between $MSR$, *size* and $MRV$ to search for. They define a $MSR$ threshold $\delta$ below which biclusters are supposed to be coherent, and they do not consider $MRV$ while searching for biclusters. Their algorithm should thus find the largest bicluster with a $MSR$ smaller than $\delta$. However, in our opinion, this strategy presents two important drawbacks. First, the choice of the $MSR$ threshold $\delta$ must be made almost arbitrarily. Secondly, using such a threshold causes the biclusters to become incoherent in a discontinuous way. In fact, they are not considered good biclusters as soon as their $MSR$ exceeds the threshold, which is not consistent.

These observations motivate us to apply the probabilistic encoding to the biclustering problem. To this aim, we present a new biclustering approach called MOBPEOC, for Multi-Objective Biclustering with Probabilistic Encoding and Overlapping Control. MOBPEOC is a genetic algorithm that represents individuals with probabilistic encoding, and uses a new multi-objective and multi-modal adapted selection mechanism.

As MOBPEOC makes use of probabilistic encoding, it should be able to deal in a better way with the uncertainty introduced by the conflicting objectives of the problem, and to improve the exploration power of the EA, compared to the binary encoding used in most other evolutionary approaches. Notably, probabilistic encoding should allow to search for a Pareto-optimal region of biclusters and to extract the core

set of coherent and high variance genes and conditions they share. Individuated regions could then be explored for biologically meaningful biclusters by a human decision-maker. In this paper we propose an automatic decision-making process, based on simulated annealing, in order to demonstrate the efficiency of the method to discover interesting biclusters. MOBPEOC also makes use of the Niched Pareto selection operator proposed in [16], but with a niching distance and a niching radius defined in terms of overlapping between biclusters. Our goal is to provide the algorithm with a selection mechanism able to consider the three objectives simultaneously, and to allow the algorithm to find several different biclusters in one run, with a fine-grained control over the level overlapping of the individuated biclusters in the expression matrix, by tuning the niching radius parameter.

# 4. THE MOBPEOC ALGORITHM

The MOBPEOC algorithm follows the classical structure of genetic algorithms (GAs). Biclusters are represented using probabilistic encoding, with one probability associated to each of the genes or conditions in the analyzed dataset. The algorithm will consider three objectives to optimize: $MSR$, size and $MRV$ of a region, computed as detailed in section 3. Recently, it has been shown that $MSR$ may fail at individuating some kind of patterns in biclusters [1]. However, in this paper, we will use $MSR$, as a first attempt to assess the validity of applying probabilistic encoding to the problem of biclustering.

The population is initialized by creating individuals according to the following process. For each individual, a proportion of randomly selected genes and conditions receive a high probability value, while the remaining ones receive a low probability value. The proportion of selected genes and conditions, as well as the probabilities assigned to each gene or condition, are randomly picked between parametric minimal and maximal thresholds. These parameters allow to control the size of the regions created, as well as the size of the sub-matrices in these regions.

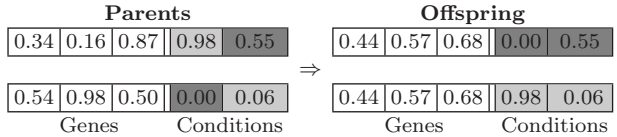
The fitness of an individual is based on the concept of Pareto dominance, and is used during the selection process. In fact, selection is enforced using the Niched Pareto selection mechanism proposed in [16]. This operator randomly selects two candidate individuals from the current generation, and returns one of them using a two step tournament. During the first step, the quality of the two individuals is evaluated using the domination relations in the current generation. If the first step of the tournament leads to a tie, continuously updated sharing is used to compute the niche count of both individuals in the provisional next generation. The individual that presents the lowest niche count is then selected. Differently from [16], sharing is enforced directly in the search space. This is because our goal is not to maintain diversity along the Pareto optimal front, but to individuate different biclusters in one run. The distance function $d$ between two genotypes (respectively encoding the traits probability functions $p^1_{traits}$ and $p^2_{traits}$) is defined as an estimated measurement of the overlapping rate between the typical biclusters that each genotype represents: $d(p^1_{traits}, p^2_{traits}) = 1 - overlap_{genes} * overlap_{conditions}$

with

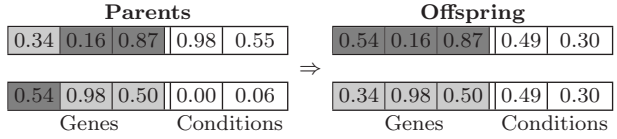
$$overlap_{genes} = \frac{|\{i \in I |\ (p^1_{traits}(i) \geqslant 0.5 \wedge p^2_{traits}(i) \geqslant 0.5)\}|}{|\{i \in I |\ (p^1_{traits}(i) \geqslant 0.5 \vee p^2_{traits}(i) \geqslant 0.5)\}|}$$

$$overlap_{conditions} = \frac{|\{j \in J |\ (p^1_{traits}(j) \geqslant 0.5 \wedge p^2_{traits}(j) \geqslant 0.5)\}|}{|\{j \in J |\ (p^1_{traits}(j) \geqslant 0.5 \vee p^2_{traits}(j) \geqslant 0.5)\}|}$$

Variation is enforced using three crossover operators, classical GAs uniform crossover, gene mean crossover and condition mean crossover, and two mutation operators, gene mutation and condition mutation. Gene mean crossover (respectively condition mean crossover) works in a similar way as uniform crossover for conditions (respectively genes) probability values, but for each gene (respectively condition), the probability values of the offspring are computed as the mean of the probability values of the two parents, see figure 1.Gene mutation replaces a randomly picked gene probability value by a randomly generated number in $[0, 1]$. In the same way, condition mutation replaces a randomly selected condition probability value by a random number in $[0, 1]$. Every time crossover or mutation has to be applied, one of the corresponding operators is chosen with a probability which is supplied as a parameter.




(a) Gene mean crossover.




(b) Condition mean crossover.

**Figure 1: Crossover operators.**

After a complete run, MOBPEOC returns a population of promising biclusters regions. We propose to search each of the individuated regions for a low $MSR$ bicluster, using a local search algorithm: simulated annealing (SA) [17]. SA has already been applied for biclustering of microarray data in [7], where it is shown to perform better than Cheng and Church's greedy approach [8]. Our implementation of SA starts with the bicluster containing all the genes and conditions with a probability $\geqslant 0.5$ in the genotype encoding the searched region. It then tries to improve the $MSR$ by repeatedly adding or removing one gene or one condition from the bicluster, according to the simulated annealing scheme. Genes and conditions with a probability in the genotype higher than a parametric threshold cannot be removed from the bicluster. Similarly, genes and conditions with a probability in the genotype lower than a parametric threshold cannot be added to the bicluster.

# 5. EXPERIMENTS AND RESULTS

In order to assess the effectiveness of our proposal, we conduct experiments on two well known datasets. The first dataset is the yeast Saccharomyces cerevisiae cell cycle expression dataset, [9]. The expression matrix contained in this dataset consists of 2884 genes and 17 conditions. The second dataset is the human B-cells expression data, [2], which contains 4026 genes and 96 conditions. The two datasets are taken from [8], where the original data are preprocessed. The most important preprocessing operation regards

missing values: missing values are replaced with random values, although it is known that these random numbers can affect the discovery of biclusters [25]. The expectation was that these random values would not form recognizable patterns.

Tables 2 and 3 show the parameter settings used in the experiments. These values have been obtained after several trial experiments on the two datasets. When the algorithm reaches the maximum number of generations, a bicluster is extracted from each of the returned regions.

**Table 2: MOBPEOC GA parameters.**

| Parameter | Value (Yeast/Human) |
|---|---|
| Main GA loop | |
| Number of generations | 1000 |
| Size of the population | 600 |
| (Re)Initialization of the population | |
| Min-max % of selected genes | 0.1-4 / 0.05-2 |
| Min-max % of selected conditions | 70-100 / 90-100 |
| Thresholds for high probabilities | 0.7-1.0 |
| Thresholds for low probabilities | 0.0-0.3 |
| % of population reinitialized [15] | 5 |
| Selection | |
| Size of regions sample set | 50 |
| Size of comparison set [16] | 510 |
| Niching radius [16] | 0.85 |
| Scaling factor [16] | 1 |
| Genetic operators | |
| % of crossover per generation | 85 |
| % of uniform crossover | 97.5 / 95 |
| % of gene mean crossover | 2.5 / 4 |
| % of condition mean crossover | 0 / 1 |
| % of mutation per generation | 5 |
| % of gene mutation | 60 |
| % of condition mutation | 40 |

**Table 3: Automatic decision-making parameters.**

| Parameter | Value (Yeast/Human) |
|---|---|
| Exploration of regions | |
| Min prob. of unremovable gene | 0.85 / 0.98 |
| Max prob. of unaddable gene | 0.15 |
| Min prob. of unremovable cond. | 0.85 / 0.95 |
| Max prob. of unaddable cond. | 0.15 |
| Simulated annealing | |
| Initial temperature | 0.08 / 2 |
| Final temperature | 0.015 / 0.0375 |
| Number of attempts per temp. | Number of addable or removable genes and conditions |
| Temperature decrease ratio | 0.90 / 0.85 |

For each dataset, we compute the average $MSR$ and size of the discovered biclusters. Then we compare these results with those obtained by C&C greedy approach [8], and by two EAs, SMOB [12] and SEBI [11]. SMOB and SEBI adopt a sequential covering strategy. The EA is run several times, and each time a bicluster is returned. These results are reported in table 4. It can be seen that MOBPEOC obtains the lowest MSR on the two datasets. This result is particularly encouraging, and it confirms the validity of our proposal. Notice that MOBPEOC does not make use of any threshold for limiting the MSR, as it happens in the other three algorithms, which used a threshold of 300 on the yeast dataset and of 1200 on the human dataset. Nevertheless, MOBPEOC discovered biclusters that contain genes that present a much more similar behavior under the same set of conditions, as reflected by the much lower MSR. Moreover, this has another important consequence, as it frees the user from the task of determining a specific threshold for a given dataset. This is an important consideration, since the use of a wrong threshold will exclude certain biclusters from the search process.

**Table 4: Averages MSR, size and total coverage obtained by the four algorithm on the yeast (Y) and the human (H) datasets. Standard deviation is reported between brackets.**

| | | MSR | size | % Cov. |
|---|---|---|---|---|
| MOBPEOC | Y | 88.83 (57.42) | 391.34 (481.63) | 39.39 |
| | H | 744.04 (337.21) | 1967.05 (2000.54) | 21.92 |
| **C&C** | Y | 204.29 (42.78) | 1576 (2178.46) | 81.14 |
| | H | 850.04 (153.91) | 4595.98 (3353.72) | 36.81 |
| **SMOB** | Y | 206.17 (15.82) | 453.48 (231.76) | 40.39 |
| | H | 1019.16 (120.78) | 709.13 (378.05) | 33.52 |
| **SEBI** | Y | 205.18 (4.49) | 209.92 (171.39) | 38.14 |
| | H | 1028.84 (29.19) | 615.84 (278.35) | 34.07 |

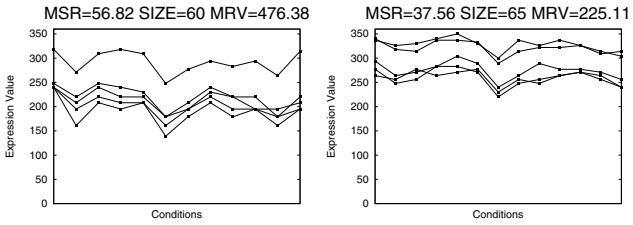



**Figure 2: Examples of biclusters obtained on the yeast dataset.**

Results concerning the volume are also satisfactory, as the average volume of the biclusters discovered by MOBPEOC is higher than those obtained by the other two EAs. Only on the yeast dataset, the volume obtained by SMOB is slightly higher. C&C obtains better results, as far as the volume is concerned. However, this is mainly due to the first biclusters that are found by this algorithm. In fact, these biclusters are characterized by a huge volume, but are not very interesting from the point of view of the MSR and MRV [8]. As far as the coverage is concerned, MOBPEOC obtains results that are comparable to those obtained by SMOB and SEBI on the yeast dataset, while on the human dataset SMOB and SEBI obtain better results. This despite that the average size of the biclusters found by MOBPEOC is higher than the average size of the biclusters found by the other two algorithms on this dataset. Another advantage that our proposal presents with respect to SMOB and SEBI, is that all the biclusters are found in one single run of the algorithm, while the other two EAs has to be run several times, since they adopt a sequential covering strategy.

Figures 2 and 3 present two examples of biclusters obtained on both the yeast and the human dataset, respectively, with their $MSR$, $MRV$ and size. The biclusters relative to the yeast dataset present a very similar behavior under all the conditions. In particular, in bicluster 540, we can clearly distinguish two groups of biclusters that varies

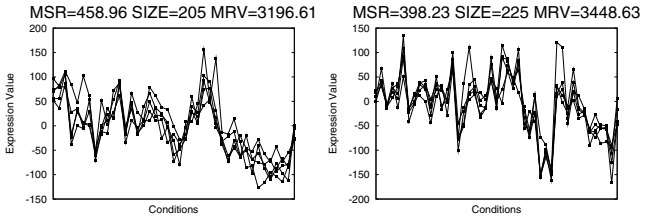



**Figure 3: Examples of biclusters obtained on the human dataset.**

in the same way. Such a behavior is know as a shifting pattern [1]. The biclusters relative to the human dataset, do not present such a behavior. Instead, their genes present steep, and yet simultaneous, variations in their expression levels. This kind of patterns are known as scaling patterns [1]. Thus, even with the limitation imposed by $MSR$ [1], MOBPEOC is able to discover biclusters containing shifting patterns.

Another aspect we wanted to address was the control of overlapping among biclusters. For this, we analyse the overlapping rates obtained on the yeast dataset, see table 5. The mean overlapping among the 600 biclusters returned using the MOBPEOC approach over the yeast dataset is 16.10%. This value seems to agree with the choice of a niching radius enforcing 15% of similarity between the niches established in the GA. In order to validate the efficiency of the overlapping control mechanisms in MOBPEOC, we ran the algorithm on the yeast dataset again with a niching radius equal to zero, i.e., which allows 100% of similarity between the niches. Figures 4 and 5 show the evolution of the minimal, average and maximal distance among the individuals for each of the 1000 generations performed. Figure 4 presents this result when the niching radius assumes value 0.85, while figure 5 when it assumes value 0. When the niching radius is set to 0.85, it

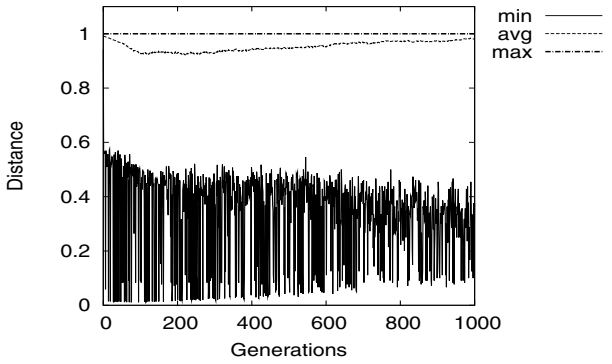



**Figure 4: Evolution of the distance between individuals in the population with Niching radius = 0.85.**

can be noticed that the mean overlapping distance decreases during the first 100 generations. We can suppose that this phase corresponds to the discovery of a core set of niches. In a second phase, the mean distance grows very slowly while the minimal overlapping distance continues to decrease. We can suppose that this phase corresponds to the convergence inside each niche towards a particularly good solution within the niche. On the other hand, when the niching radius is set

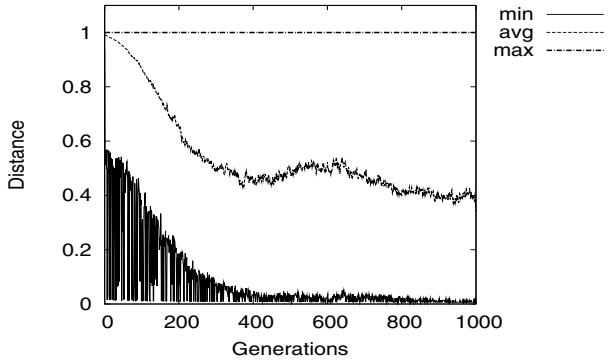



**Figure 5: Evolution of the distance between individuals in the population with Niching radius = 0.**

to zero, the mean and minimal overlapping distance collapse in the first 400 generations. This difference of behavior between the two settings is a clear hint of the efficiency of the overlapping control mechanism. We have also compared the biclusters obtained using MOPEOC with these two settings, and noticed that when the niching radius is set to zero, the returned biclusters become logically very similar, showing high levels of overlapping, and covering less elements in the expression matrix.

**Table 5: Overlapping rate of the discovered biclusters with Niching radius = 0.85 and 0.**

| N.r. | Min | Mean | Max | St. Dev |
|---|---|---|---|---|
| 0.85 | 0.0 | 16.10 | 100 | 22.67 |
| 0 | 8.17 | 60.59 | 100 | 18.63 |

# 6. CONCLUSIONS

In this work we have proposed a novel encoding scheme for EAs, called probabilistic encoding. In this scheme, each gene of the individual represents a probability that a trait of the problem has to belong to a solution encoded by the individual. Such an encoding allows to deal with uncertainty that characterizes many optimization problems. Moreover, probabilistic encoding allows to search for promising regions in the search space, instead of exploring single points. Once the most promising region has been detected, the region can be explored for searching for the optimal solutions. Thus, the individuated regions could represent starting points where to look for the optimal solution.

In order to test our proposal, we have incorporated it into a multi-objective and multi-modal EA, called MOBPEOC, and applied the algorithm to the problem of finding biclusters in microarray data. Once the EA has individuated promising regions, a local search procedure, based on simulated annealing, is applied to extract interesting biclusters from such regions. This allows to individuate various biclusters in one single run of the EA, which is an advantage with respect to other methods that have to apply a sequential covering strategy. The results obtained demonstrate the validity of our proposal. In fact, MOBPEOC obtained biclusters that are, in general, characterized by a higher vol-

ume and a lower MSR, with respect to other evolutionary approaches.

As future work, we intend to perform a biological evaluation of the biclusters, and to expand the experimentation to other microarray datasets, characterized by a higher dimensionality. Another possible development is to use other quality measures instead of the $MSR$, to test the kind of biclusters discovered in this way.

## Acknowledgments

This work has been funded by the Spanish Ministry of Science and Innovation under grant TIN2007-68084-C02-01 and by the Belgian Fund for Scientific Research (F.R.S.-FNRS). The authors would also like to thank Stéphane Amant for useful discussions and the anonymous reviewers for their valuable comments.

## 7. REFERENCES

[1] J. S. Aguilar-Ruiz. Shifting and scaling patterns from gene expression data. *Bioinformatics*, 21:3840–3845, 2005.

[2] A. A. Alizadeh and et al. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000.

[3] D. Arnold and H.-G. Beyer. A general noise model and its effects on evolution strategy performance. *IEEE Trans. Evolutionary Computation*, 10(4):380–391, 2006.

[4] T. Back, D. B. Fogel, and Z. Michalewicz, editors. *Evolutionary Computation 1: Basic Algorithms and Operators*. IOP Publishing Ltd., Bristol, UK, 1999.

[5] D. P. Berrar, W. Dubitzky, and M. Granzow. *A Practical Approach to Microarray Data Analysis*. Springer Publishing Company, Incorporated, 2003.

[6] S. Bleuler, A. Prelić, and E. Zitzler. An EA Framework for Biclustering of Gene Expression Data. In *Congress on Evolutionary Computation*, pages 166–173, Piscataway, NJ, 2004. IEEE.

[7] K. Bryan. Biclustering of expression data using simulated annealing. In *Proceedings of the 18th IEEE Symposium on Computer-Based Medical Systems*, CBMS '05, pages 383–388, Washington, DC, USA, 2005. IEEE Computer Society.

[8] Y. Cheng and G. M. Church. Biclustering of expression data. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, pages 93–103. AAAI Press, 2000.

[9] R. Cho, M. Campbell, E. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T. Wolfsberg, A. Gabrielian, D. Landsman, D. Lockhart, and R. Davis. A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell*, 2:65–73, 1998.

[10] G. P. Coelho, F. O. de França, and F. J. V. Zuben. Multi-objective biclustering: When non-dominated solutions are not enough. *J. Math. Model. Algorithms*, 8(2):175–202, 2009.

[11] F. Divina and J. S. Aguilar-Ruiz. Biclustering of expression data with evolutionary computation. *IEEE Trans. on Knowledge and Data Engineering*, 18(5):590–602, 2006.

[12] F. Divina and J. S. Aguilar-Ruiz. A multi-objective approach to discover biclusters in microarray data. In *Proceedings of the 16th Genetic and Evolutionary Computation Conference*, pages 385–392. ACM, 2007.

[13] A. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing*. SpringerVerlag, 2003.

[14] L. Fei and L. Juan. Biclustering of gene expression data with a new hybrid multi-objective evolutionary algorithm of nsga-ii and eda. In *2nd International Conference on Bioinformatics and Biomedical Engineering, 2008. ICBBE 2008.*, pages 1912–1915. IEEE, 2008.

[15] C. M. Fonseca and P. J. Fleming. Multiobjective optimization and multiple constraint handling with evolutionary algorithms-part i: A unified formulation. *IEEE Trans. on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 28:26–37, 1998.

[16] J. Horn, N. Nafpliotis, and D. E. Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In *IEEE World Congress on Computational Intelligence*, pages 82–87. IEEE, 1994.

[17] S. Kirkpatrick, J. Gelatt, C. D., and M. P. Vecchi. Optimization by Simulated Annealing. *Science*, 220(4598):671–680, 1983.

[18] P. Larrañaga and J. A. Lozano, editors. *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*, chapter An introduction to probabilistic graphical models. Kluwer Academic Publishers, 2002.

[19] S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 1(1):24–45, 2004.

[20] S. Mitra and H. Banka. Multi-objective evolutionary biclustering of gene expression data. *Pattern Recogn.*, 39(12):2464–2477, 2006.

[21] I. Paenke, J. Branke, and Y. Jin. Efficient search for robust solutions by means of evolutionary algorithm and fitness approximation. *IEEE Trans on Evolutionary Computation*, 10(4):405–420, 2006.

[22] D. Parrott and X. Li. Locating and tracking multiple dynamic optima by a particle swarm model using speciation. *IEEE Trans. Evol. Comput*, 10(4):440–458, 2006.

[23] T. Van den Bulcke and et.al. Probic: identification of overlapping biclusters using probabilistic relational models. Poster at ECCB/ISMB, 2007.

[24] J. Yang, H. Wang, W. Wang, and P. Yu. Enhanced biclustering on expression data. In *Proceedings of the Third IEEE Symposium on BioInformatics and BioEngineering (BIBE'03)*, pages 321–327, 2003.

[25] J. Yang, W. Wang, H. Wang, and P. S. Yu. $\delta$–clusters: Capturing subspace correlation in a large data set. In *Proceedings of the 18th IEEE Conference on Data Engineering*, pages 517–528, 2002.

[26] S. Yang, Y.-S. Ong, and Y. Jin, editors. *Evolutionary Computation in Dynamic and Uncertain Environments*, volume 51 of *Studies in Computational Intelligence*. Springer, 2007.

[27] E. Zitzler, M. Laumanns, and L. Thiele. Spea2: Improving the strength pareto evolutionary algorithm. Technical report, Computer Eng. and Networks Lab. - Swiss Federal Institute of Technology Zurich, 2001.