CHAPTER 27

A HYBRID EVOLUTIONARY ALGORITHM FOR KNOWLEDGE DISCOVERY IN MICROARRAY EXPERIMENTS

L. Jourdan, M. Khazaoui, C. Dhaene and E. Talbi
LIFL, CNRS Bâtiment M3, Cité scientifique, 59655 Villeneuve d’Ascq - France

In this chapter, we present a hybrid evolutionary algorithm for knowledge discovery in biological data from microarray analysis. The problem of analyzing microarray data is actually a major issue in genomics. Often used techniques to analyze microarray data are clustering and classification. We propose to analyze those data through association rules, modeling this problem as a rule mining problem and using a metaheuristic to explore the large search space associated. Hence, we expose the main features of the proposed Genetic Algorithm scheme for association rule problem (encoding, mutation and crossover operators) and its multicriteria aspects. Results on real data are given.

1. Introduction

Experiments with DNA microarray technology generate a large amount of data as they are able to measure the expression levels of thousands of genes in a single experiment. To explore these data it is necessary to develop knowledge discovery techniques that can extract biological significance and use the data to assign functions to genes. This is the goal of data mining (also known as Knowledge Discovery in Databases - KDD) that has been defined as “The non trivial extraction of implicit, previously unknown, and potentially useful information from data.”

Several kinds of representations to express knowledge that can be extracted from microarray data can be found in the literature. Most current approaches in this area group genes, using clustering algorithms such as hierarchical clustering or Kohonen maps, i.e., self-organizing maps. Association rule mining is an advanced data mining technique that is useful in deriving meaningful rules from a given dataset. Rule mining, as well as other datamining approaches, may be seen as a NP-hard combinatorial
problem that can be solved with combinatorial optimization methods such as evolutionary algorithms.

In this chapter, we propose a multicriteria hybrid metaheuristic for microarray data analysis. We firstly introduce the biological context of the work. Then we set the rule mining context. After this positioning, we present a multicriteria hybrid algorithm and describe every feature of the algorithm (representation, operators, ...). Finally, results on real datasets are presented and analyzed.

2. Microarray analysis

Microarray technology (also called DNA microarrays, DNA arrays, DNA chips, gene chip, ...) is now widely used in many areas of biomedical research. It provides access to expression levels of thousands of genes at once in order to identify co-expressed genes, relationships between genes, patterns of gene activity, changes in gene activity under some medical treatments, etc. A microarray is typically a glass (or some other material) slide, on to which probes are attached at fixed locations (spots). There may be tens of thousands of spots on a single array. The technology consists in hybridizing DNA of a reference sample and a test sample, previously labeled with different fluorescent dyes, with probes that match a single gene. Hence, measuring the fluorescence allows to calculate the relative abundance of DNA of each gene and evaluate their expression level. Arrays are scanned and images are produced and analyzed to obtain an intensity value for each probe.

There are two variants of the DNA microarray technology: Synteni/Stanford chips and Affymetrix chips. They differ in:

- How DNA sequences are put down (spotting / photolithography).
- Length of DNA sequences (complete sequences or a serie of fragments).

In the Synteni/Stanford chips, probes of cDNA (500/5,000 bases long) are immobilized to a solid surface such as glass using robot spotting and exposed to a set of targets either separately or in a mixture. Affymetrix chips are powerful technology for re-sequencing DNA and polymorphisms detection. They use photolabile agnets and photolithography techniques.

Although fundamental differences exist between these two technologies, their strength lies in a massively parallel analysis of thousands of genes.
and the generation of a lot of data.

Analyzing DNA microarray data requires a preprocessing phase\(^6,36,49\). Figure 1 recalls the different steps required to produce new biological assumptions from microarray experiments. Major steps are described below.

![Diagram of microarray analysis process](image)

**Fig. 1.** Process of hypothesis extraction from microarray.

**Relative gene expression:** The differential gene expression is calculated by dividing the intensity of the gene in the sample under study by its intensity level in the control. This intensity ratio has a highly asymmetric distribution. To avoid this problem, a \(\log_2\)-transformation is usually used to make a normal like distribution:

\[
\text{Gene expression} = \log_2 \frac{\text{Expression of a gene in the sample}}{\text{Expression of the same gene in the control}} \tag{1}
\]

**Normalization:** There are many sources of variations of measures in microarray experiments (variations in cells or individuals, mRNA extraction, isolation, hybridization condition, optical measurement, scanner noise etc...). The purpose of normalization is to adjust (or correct) a signal in order to make the comparison with other signals more meaningful. Many techniques for normalization aim to make the data more normally distributed (Log-transformation per chip and per gene). This is an important issue to
be able to analyze data.

**Gene filtering and discretization:** We are interested in finding genes that show significant differences between two groups of patients. Hence, the filtering process may remove genes that do not differentiate the sample under study from the test sample (their relative expression is not significant). Most of the time, gene expression data is discretized into under / over expressed genes thanks to cutoffs.

Hence after these different phases, data may be considered as large tables indicating, with discretized values, the relative gene expression for thousands of genes under different experiments.

### 3. Rule mining

In this work, we propose to use a rule mining approach to explore the data provided by microarray experiments. We first introduce what rule mining is, and a classical rule mining approach. Then, we present a state of the art of bioinspired methods for rule mining. Finally, we justify the use of rule mining, in analysis of microarray experiments.

#### 3.1. Association rules

The search of association rules is an important task in knowledge discovery from databases\(^1,34,39\). Originally, proposed for Market Basket data to study consumer-purchasing patterns, it has potential applications in many areas. Given a database describing instances (eg. transactions) according to different attributes (eg. items purchased), this problem consists in discovering rules in the form: *IF C THEN P* or in a more detailed expression, *IF Cond\(_1\) AND Cond\(_2\) AND ... AND Cond\(_m\) THEN P*. The condition C is a conjunction of terms. For binary representation, a term indicates whether the corresponding attribute has been chosen or not. In nominal data, where an attribute may take different values, a term may be of the form \(<\text{attribute operator value}>\). The prediction P is also represented by a conjunction of terms.

The association rule search problem is a complex combinatorial problem as every attribute may be candidate to participate, with one of its possible values, to the rule. Hence the number of possible combinations is exponential regarding to the number of attributes. Moreover, this problem has been
shown to be NP-hard by Angiulli et al.\textsuperscript{3}. The problem has been reduced into a MAX clique problem that is a well known NP-hard problem\textsuperscript{17}. Therefore, algorithms dedicated to this problem have to be carefully designed.

### 3.2. A constructive method for rule mining

A classical approach for finding association rules is to look for frequent itemsets (set of items - or terms - that often occur simultaneously) as done by the Apriori\textsuperscript{2} algorithm. This algorithm relies on a fundamental property of frequent itemsets, called the Apriori property: “Every subset of a frequent itemset is also a frequent itemset”. This monotony allows to incrementally construct interesting sets of items in a very efficient way. Then items belonging to a same set are combined in order to obtain rules. There exists numerous variants of Apriori\textsuperscript{7,37,44} that allow to deal with problems of interesting sizes, but each of them requires the monotony of the criterion used.

### 3.3. Bioinspired methods and rule mining

As the number of rules that are candidate for extraction is exponential\textsuperscript{60}, the use of an evolutionary algorithm such as a genetic algorithm is well adapted to explore the large search space of candidate rules. These algorithms have already shown their ability to solve large combinatorial optimization problems and have already been applied with success to the rule discovery problem\textsuperscript{28}.

The paradigm of evolutionary algorithms consists of stochastic search algorithms that are based on an abstraction of the process of Darwinian evolution\textsuperscript{11}. An evolutionary algorithm maintains a population of “individuals”, each of them representing a potential solution to a given problem. Each individual is evaluated by a fitness function, which measures the quality of the solution (its adaptation to the problem). Individuals evolve toward better and better individuals via a selection procedure, based on the fitness of individuals, thanks to crossover (recombination) and mutation operators, and the replacement procedure.

In previous studies on association rule discovery with genetic algorithms, we can find two major representations: the Michigan\textsuperscript{21} representation and the Pittsburg\textsuperscript{39} one. Each representation has its drawbacks and advantages. The Pittsburg representation, where an individual is a set of rules, is more expensive in memory but has been used in several genetic algorithms: GABI\textsuperscript{26}, GIL\textsuperscript{24} and HDPDCS\textsuperscript{55}. In the Michigan representation each individual of
the population of the genetic algorithm represents a candidate rule. The main drawback of the Michigan approach is that it could, without the use of specific mechanisms, converge to a single rule and not to a set of rules. For example, in Weiss uses the sharing fitness to avoid the convergence to a single rule in order to predict rare events. Classical genetic algorithms for rule mining using the Michigan representation are COGIN\textsuperscript{10} and REGAL\textsuperscript{18}.

### 3.4. Interest of rule mining for microarray analysis

Microarray data format is very similar to the Market Basket data format. Microarray data may be represented in two different ways\textsuperscript{32}:

- **Gene table:** The genes constitute the rows of the data table whereas experiments to which genes were exposed are the columns. The values represent the abundance of transcript for each spot on the microarray. Clustering and classification may be applied to the gene table to divide the dataset into clusters/classes by grouping the rows (genes).

- **Treatment table:** The reverse way to present data is to flip the gene table. Thus, genes are now the columns whereas treatments are the rows. The values are the gene expression level. The objective is then to find associations between columns (genes). Therefore, association rule mining is a great challenge. This data mining model is general and allows to find, for example, associations between subsets of genes. Moreover, thanks to the condition and prediction notions, relations obtained are precise and give more information than only grouping genes together.

There has been some recent works on using association rule mining to analyze gene expression data. Chen et al.\textsuperscript{9} apply association rules to mine the transcription factors essential to certain gene expressions using the Apriori algorithm. They manage, with a small value of support to extract a small number of rules they can analyze. Creighton and Hanash\textsuperscript{10} propose to use also the Apriori algorithm to reveal biological relevant associations between different genes or between environmental effects and gene expression. They apply it with success to a yeast database. Kotala et al.\textsuperscript{32} introduce a new approach to mine association rules from microarray gene expression data using Peano count tree. Ieee et al.\textsuperscript{23} focus on the combinatorial analysis of motifs involved in transcriptional control and introduce a
A hybrid metaheuristic for knowledge discovery in microarray experiments

7

notion of association rules with distance information.

In our study we will consider data in the treatment table form (genes are the columns, treatments - or comparison of individuals of different status - are the rows. Our goal is to look for rules combining genes where a term can be in the form \(<gene = value>\). The value belongs to the discretized gene expression level. An example of a rule could be: IF \(gene_{12} = over\text{-}expressed\) AND \(gene_{604} = under\text{-}expressed\) THEN \(gene_{8734} = over\text{-}expressed\).

4. A hybrid multiobjective metaheuristic

The algorithm proposed has been developed thanks to the open source framework paradisEO (PARAllel and DIstributed Evolving Objects)\(^8\) http://www.lifl.fr/~cahon/logiciels.html that allows, once you have designed your algorithm, to easily develop multicriteria evolutionary algorithms and to use parallelism and distributed mechanisms. The originality of the proposed approach is to deal with rule mining through a multiobjective point of view. Hence, we first recall some principles of multiobjective combinatorial optimization problem, then we present the modeling of the rule mining problem as a multiobjective problem. Finally we present the algorithm and its hybridization with an exact enumeration.

4.1. A Multicriteria model for rule mining

4.1.1. Multicriteria definitions

We first describe and define Multi-Objective combinatorial Problems (MOPs) in a general case. Assume that a solution to such a problem can be described by a decision vector \((x_1, x_2, ..., x_n)\) in the decision space \(X\). A fitness function \(f : X \rightarrow Y\) evaluates the quality of each solution by assigning it an objective vector \((y_1, y_2, ..., y_p)\) in the objective space \(Y\) (Fig. 2)\(^{51}\). So, multiobjective optimization consists in finding the solutions, in the decision space, optimizing (minimizing or maximizing) \(p\) objectives.

For the following definitions, we consider the minimization of \(p\) objectives. For maximization problems, definitions are similar.

In the case of a single objective optimization, the comparison between two solutions \(x^1\) and \(x^2\) is immediate. If \(y^1 < y^2\) then \(x^1\) is better than \(x^2\). For
multiobjective optimization, comparing two solutions \( x^1 \) and \( x^2 \) is more complex. Here there exists only a partial order relation, known as the Pareto dominance concept illustrated by figure 3:

**Definition 1:** A solution \( x^i \) dominates a solution \( x^j \) if and only if:

\[
\begin{align*}
\forall k \in [1..p], & \quad f_k(x^i) \leq f_k(x^j) \\
\exists k \in [1..p], & \quad f_k(x^i) < f_k(x^j)
\end{align*}
\]

**Definition 2:** A solution is Pareto optimal if it is not dominated by any other solution of the feasible set.

The set of optimal solutions in the decision space \( X \) is denoted as the Pareto set, and its image in the objective space is the Pareto front. In MOP, we are looking for all the Pareto optimal solutions.

In the Pareto front two types of solutions may be distinguished: the supported solutions (that are on the convex hull of the set of solutions and that may be found by a linear combination of criteria), and non-supported solutions\(^{45}\). These solutions are important, because for some problems only few Pareto solutions are supported (the extremes) and to get a good compromise between the two criteria, it is necessary to choose one of the non-supported solution.

4.1.2. **Multicriteria association rules**

In order to solve association rule discovery problem as a combinatorial optimization problem, the optimization criterion has to be defined. A lot
of measures exist for estimating the quality of association rules. For an overview, readers can refer to Freitas\textsuperscript{16}, Tan et al.\textsuperscript{43} or Khabzaoui et al.\textsuperscript{30}. In a previous work\textsuperscript{30}, we made a statistical study of different criteria found in the literature. This study lead us to determine five groups where each group represents correlated criteria. We choose to select one criterion of each group and obtain five complementary criteria that allow to evaluate rules in a complete way: Support, Jmeasure, Interest, Surprise and Confidence. Those criteria are described below.

Rules are evaluated for a set of $N$ instances, where $|C|$ (resp. $|P|$) represents the number of instances satisfying the $C$ (resp. $P$) part of the rule and $|C\&P|$ the number of instances satisfying simultaneously the $C$ and the $P$ parts of the rule.

**Support ($S$):** It is the classical measure of association rules. It enables to measure rule frequency in the database. It is the percentage of transactions containing, both the $C$ part and the $P$ part, in the database. It is used to find frequent itemsets in Apriori.

$$S = \frac{|C\&P|}{N}$$ (3)

**Confidence ($C_f$):** The Confidence measures the validity of a rule. It is the conditional probability of $P$ given $C$. It is used in Apriori to find interesting

\[\text{In this example, points 1, 3 and 5 are non-dominated. Point 2 is dominated by point 3, and point 4 by points 3 and 5.}\]
rules in frequent itemsets.

\[ C_f = \frac{|C\&P|}{|C|} \quad (4) \]

**J-measure** \((Jm)\): Smyth and Goodman \(^{40}\) have proposed the Jmeasure, which estimates the degree of interest of a rule and combines support and confidence. It is used in optimization methods\(^{4,47}\).

\[ Jm = \frac{|P|}{N} \times \frac{|C\&P|}{|P|} \log \left( \frac{N \times |C\&P|}{|C| \times |P|} \right) \quad (5) \]

**Interest** \((I)\): The Interest measures the dependency while privileging rare patterns in the region of weak support.

\[ I = \frac{N \times |C\&P|}{|C| \times |P|} \quad (6) \]

**Surprise** \((R)\): It is used to measure the affirmation. It enables to search surprising rules.

\[ R = \frac{|C\&P| - |C\&P|}{|P|} \quad (7) \]

Hence using these five criteria allows to evaluate a rule in a multicriteria manner. As they are complementary, this model is interesting to select interesting rules and to reduce the numbers of possible rules.

### 4.2. The Genetic Algorithm

We present the general scheme of the algorithm and the genetic operators used. The algorithm presented here is a multicriteria version of ASGARD presented in\(^{27,28}\).

#### 4.2.1. General scheme

Figure 4 presents the scheme of the genetic algorithm with its multicriteria aspects. The algorithm starts with a set of randomly generated solutions. Then, solutions are selected according to their fitness quality to create new solutions (offspring). Best solutions encountered over generations are archived into a secondary population called the “Pareto Archive”.

#### 4.2.2. Genetic operators

Genetic operators allow diversification and intensification of the search.
Crossover: The proposed crossover operator has two versions:

- **Exchange crossover:** If two individuals \(X\) and \(Y\) have one or several common attribute(s) in the \(C\) parts, one common attribute is randomly selected. The value of the selected attribute in \(X\) is exchanged with its counterpart in \(Y\) (see figure 5).
- **Insert crossover:** Conversely, if \(X\) and \(Y\) have no common attribute, one term is randomly selected in the \(C\) part of \(X\) and inserted in \(Y\) with a probability inversely proportional to the length of \(Y\). The similar operation is performed to insert one term of \(Y\) in \(X\) (see figure 6).
Mutation: There are four mutation operators designed. The mutation operator called “Value mutation” replaces the value of an attribute by a randomly chosen one (see Figure 7). The second one called “Attribute mutation” replaces an attribute by another; the value of this attribute is randomly chosen (see Figure 8). The third one is a reduction mutation that randomly removes one term of the rule. The last one is an augmentation mutation that randomly adds a term to the rule.
4.3. Multicriteria mechanisms

In order to deal with multicriteria optimization problems, different mechanisms have to be used. For example, the notion of dominance has to be defined (to be able to compare solutions) and population management has to be carefully studied.

**Selection operator:** The classical roulette selection based on the ranking notion has been used. The probability of selection of a solution is proportional to its rank. We use two ranking methods:

1. **Pareto ranking:** The rank of a solution corresponds to the number of solutions, in the current population, by which it is dominated (see Figure 9 for a minimization problem)\(^{15}\).

2. **Non-Dominated Sorting GA (NSGA):** This method assigns ranks to solutions by first finding the set of non-dominated solutions in the current population. These solutions are removed from the population and assigned rank 1. As these solutions are removed, a new so-called front of non-dominated solutions is now present in the remainder of the original population. This second front is extracted and assigned rank 2. This procedure is repeated until no more solution is present in the population (see Figure 10 for a minimization problem)\(^{13}\).

![Fig. 9. Pareto ranking.](image)

![Fig. 10. NSGA ranking.](image)

Experiments done have shown that with the proposed algorithm, both ranking methods give interesting results with a small superiority for the Pareto ranking. Then, in the following, we will use the Pareto ranking for the selection.
Replacement operator: We use the elitist non dominated sorting replacement. The worst ranked solutions are replaced by the dominating solutions (if there exists any) generated by mutation and crossover operators (offsprings). The size of the population remains unchanged.

Archive: Non dominated association rules are archived into a secondary population called the "Pareto Archive" in order to keep track of them. It consists in archiving all the Pareto association rules encountered over generations. When a new Pareto solution is added to the archive, an update has to be done (some solutions may become dominated).

Elitism: The Pareto solutions (best solutions) are not only stored permanently, they also take part in the selection and may participate to the reproduction.

4.4. Hybridization

In order to increase the robustness of the approach, we hybridize it with an exact enumeration procedure. As the search space is large, this enumeration is realized for a small subspace defined thanks to solutions of the population. Hence, we design an operator which makes an exhaustive search on all the possible rules generated with attributes selected in two rules.

![Diagram](image)

Fig. 11. The enumeration operator.

This operator may be seen as a quadratic crossover operator. It takes as input two individuals, each coding a rule. It examines all the possible itemsets that can be derived from the items composing the rules, while
taking into account the different possible values of the attributes. All the possible rules that can be constructed from the generated itemsets are evaluated and introduced if necessary in a local Pareto archive. Finally, only the global Pareto solutions of the local Pareto archive are introduced in the global Pareto archive with the replacement operator. Two offsprings are candidate to take part in the population.

4.5. Adaptive rate

Probabilities of mutation are hard to set when several mutation operators compete and are often set experimentally. To overcome this problem, we implement an adaptive strategy for calculating the rate of application of each mutation operator. Many authors have worked on setting automatically probabilities of applying operator.\textsuperscript{12,29,20} Hong et al. proposed to compute the new rate of mutation by calculating the progress of the \( j^{th} \) application of mutation \( M_i \), for an individual \( \text{ind} \) mutated into an individual \( \text{mut} \) as follows\textsuperscript{22}:

\[
\text{progress}_j(M_i) = \text{Max}(\text{fitness(ind)}, \text{fitness(mut)}) - \text{fitness(ind)} \quad (8)
\]

Then for each mutation operator \( M_i \), assume \( N_{\text{mut}}(M_i) \) applications of the mutation are done during a given generation \( (j = 1, \ldots, N_{\text{mut}}(M_i)) \). Then we can compute the profit of a mutation \( M_k \):

\[
\text{Profit}(M_k) = \frac{\sum_j \text{progress}_j(M_k) / N_{\text{mut}}(M_k)}{\sum_i \left( \sum_j \text{progress}_j(M_i) / N_{\text{mut}}(M_i) \right)} \quad (9)
\]

We set a minimum rate \( \delta \) and a global mutation rate \( p_{\text{mutation}} \) for \( N \) mutation operators. The new mutation ratio for each \( M_i \) is calculated using the following formula\textsuperscript{22}:

\[
p(M_i) = \text{Profit}(M_i) \times (p_{\text{mutation}} - N \times \delta) + \delta \quad (10)
\]

The sum of all the mutation rates is equal to the global rate of mutation \( p_{\text{mutation}} \). The initial rate of application of each mutation operator is set to \( p_{\text{mutation}}/N \).

5. Experiments

5.1. Data

In order to evaluate the algorithm, we evaluate it on two microarray databases:
- a confidential microarray data containing 22376 human genes for 45 Affymetrix chips (DB1).
- a public database, the “MIPS yeast genome database” containing 2467 genes for 79 chips (YeastDB).

Genes expressions have been discretized and may take five values: Increase (I), Marginal Increase (MI), when the gene is over expressed, Decrease (D) and Marginal Decrease (MD) when it is under-expressed and No Change (NC), when the difference of expression is not significant. For a first study, a set of 514 genes (numbered from 1 to 514) that show an interesting differential expression over the set of experiments (filtered on the number of No Changes), have been selected for DB1. For the yeastDB all the genes (2467) have been considered.

**Missing value:** Missing values lead to problems in the data analysis, so that they influence the computation of statistical tests and quality criteria of association rules problems. There may have numerous missing values in microarray data due to the empty spots or because the background intensity is higher than the spot intensity. Two ways are commonly used for treatment of missing values: they may be replaced by estimated values (Median for example), or corresponding instances are deleted. In an association rule problem we propose that all genes that have missing values are kept without modification but when these missing values are on the attributes of a rule, these genes are excluded from the computation of the quality of this rule. So [N] may be different for each rule.

### 5.2. Evaluation protocol and parameters

#### 5.2.1. Configurations and parameters

Several propositions have been done in this paper in order to improve the standard genetic algorithm: elitism, adaptive strategy for mutation rates, hybridization. In order to evaluate the contribution of each of these mechanisms, several configurations of the genetic algorithm have been compared:

- **Conf A:** Pareto ranking
- **Conf B:** Pareto ranking + Adaptive strategy
- **Conf C:** Pareto ranking + Elitism
- **Conf D:** Pareto ranking + Adaptive strategy + Elitism
- **Conf E:** Conf D + Hybridization
- **Conf E’:** Conf D + Hybridization 1/10
In the Conf $E$ configuration, the crossover operator has been replaced by the enumeration operator presented for the hybridization. Using this operator for every crossover is very time consuming, hence we compared with Conf $E'$ algorithm where the hybridization operator replaces the classical crossover only one iteration over ten.

For all of these configurations, the same parameters have been used:

- Population size: around $N_{\text{Attribute}}/2$, which gives 250 for DB1, 1200 for YeastDB
- Selection in Pareto archive: 0.5
- Global Mutation rate: 0.5
- Crossover rate: 0.8
- Number of generations: 500

5.2.2. Evaluation measure used

In multicriteria optimization, solutions quality can be assessed in different ways. Some approaches compare the obtained front with the optimal Pareto front\textsuperscript{46}. Others approaches evaluate a front with a reference point\textsuperscript{25}. Some performance measures do not use any reference point or front to evaluate an algorithm\textsuperscript{31,52}, especially when the optimal Pareto front is not known at all.

Here, we use the contribution metric\textsuperscript{5,33} to evaluate the proportion of Pareto solutions given by each front.

The contribution of a set of solutions $PO_1$ relatively to a set of solutions $PO_2$ is the ratio $PO_2$ is the ratio of non-dominated solutions produced by $PO_1$ in $PO^*$, where $PO^*$ is the set of Pareto solutions of $PO_1 \cup PO_2$.

- Let $PO$ be the set of solutions in $PO_1 \cap PO_2$.
- Let $W_1$ (resp. $W_2$) be the set of solutions in $PO_1$ (resp. $PO_2$) that dominate some solutions of $PO_2$ (resp. $PO_1$).
- Let $L_1$ (resp. $L_2$) be the set of solutions in $PO_1$ (resp. $PO_2$) that are dominated by some solutions of $PO_2$ (resp. $PO_1$).
- Let $N_1$ (resp. $N_2$) be the other solutions of $PO_1$ (resp. $PO_2$): $N_i = PO_1 \setminus (PO \cup W_i \cup L_i)$.

\[
\text{Cont} (PO_1/PO_2) = \frac{\|PO\| + \|W_2\| + \|N_i\|}{\|PO^*\|} \tag{11}
\]
Let us remark that \( ||PO^*|| = ||PO|| + ||W_1|| + ||N_1|| + ||W_2|| + ||N_2|| \) and \( \text{Cont}(PO_1/PO_2) + \text{Cont}(PO_2/PO_1) = 1 \) (with \( \text{Cont}(PO_1/PO_2) \in [0,1] \)). Hence a contribution greater than 0.5 indicates that the Pareto front has been improved.

For example, we evaluate the contribution of the two sets of solution \( PO_1 \) and \( PO_2 \) on Fig. 12 solutions of \( PO_1 \) (resp. \( PO_2 \)) are represented by circles (resp. crosses). We obtain \( \text{Cont}(PO_1, PO_2) = 0.7 \) and \( \text{Cont}(PO_2, PO_1) = 0.3 \).

\[ \text{Fig. 12. Example of contribution.} \]

5.3. Results

Genetic algorithms are stochastic methods. Hence to evaluate the proposed approach, we have executed 10 runs for each configuration. Results are given with different indicators: mean of the value, min, max and its standard deviation. This allows a more efficient comparison.

5.3.1. Evaluation of the mechanisms

In order to evaluate the different mechanisms, comparisons of the final Pareto fronts obtained with the different configurations described before (Conf A, B, C, D, E, E') are reported in table 1. This table indicates the average of contribution as well as the minimum contribution, the maximum and the standard deviation.

Elitism

The gain of elitism may be evaluated by comparing similar configurations with and without elitism (here, C/A and D/B). Table 1 indicates that in average the Pareto fronts obtained using elitism are of better quality than
Table 1. Quality comparison of the different configurations (Contribution metric).

<table>
<thead>
<tr>
<th>Configuration</th>
<th>AVG</th>
<th>MIN</th>
<th>MAX</th>
<th>σ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conf B/ Conf A</td>
<td>0.55</td>
<td>0.39</td>
<td>0.78</td>
<td>0.09</td>
</tr>
<tr>
<td>Conf C/ Conf A</td>
<td>0.62</td>
<td>0.43</td>
<td>0.90</td>
<td>0.09</td>
</tr>
<tr>
<td>Conf D/ Conf B</td>
<td>0.72</td>
<td>0.32</td>
<td>0.86</td>
<td>0.07</td>
</tr>
<tr>
<td>Conf D/ Conf C</td>
<td>0.61</td>
<td>0.43</td>
<td>0.76</td>
<td>0.08</td>
</tr>
<tr>
<td>Conf E/ Conf D</td>
<td>0.74</td>
<td>0.50</td>
<td>0.87</td>
<td>0.06</td>
</tr>
<tr>
<td>Conf E/ Conf E'</td>
<td>0.60</td>
<td>0.45</td>
<td>0.81</td>
<td>0.08</td>
</tr>
<tr>
<td>Conf E/ Conf E'</td>
<td>0.58</td>
<td>0.41</td>
<td>0.74</td>
<td>0.09</td>
</tr>
</tbody>
</table>

those obtained without elitism (Cont(C/A)=0.62 and Cont(D/B)=0.72). Moreover this contribution may reach 0.9 for best improvements. This shows the interest of this mechanism.

Adaptive strategy

In a similar way, the gain of using the adaptive strategy may be evaluated by comparing similar configurations with and without it (here, B/A and D/C). Table 1 shows that using the adaptive strategy also improves, but in a smaller way, results obtained (Cont(B/A)=0.55 and Cont(D/C)=0.61). Let us remark that the improvement is greater for more complete configurations (Pareto ranking + elitism with and without adaptive strategy).

Hybridization

Two questions may arise according the use of the hybridization. Is it interesting to use such an operator and is it worth using it at each iteration. Hence we can compare configurations E/D and E/E'. Table 1 shows that using the hybridization allows to improve the more complete configuration (Pareto ranking + elitism + adaptive strategy). In this case, the minimum contribution encountered is equal to 0.59, which means that the hybridization always improves the Pareto front. When applying the operator only one generation over ten (Conf E'), results are still interesting (Cont(E'/D)=0.60), but Pareto front obtained are in general dominated by those produced by Conf E (Cont(E/E')=0.58), where the operator is applied at each generation. However, the drawback of using Conf E is the large amount of time required and a compromise between the quality of the solution and the computing time allowed has to be considered.
5.3.2. Example of rules obtained

Table 2 describes some rules of the Pareto front obtained thanks to Conf E for the YeastDB, whereas table 3 presents values of these rules for the five criteria considered (Support, Interest, Surprise, Confidence and Jmeasure). Interest of such rules is difficult to estimate in a biological point of view, but one idea would be to compare with relations already known between genes belonging to a same rule.

Table 2 shows that there exists genes that appear in several rules, and we may suppose that those genes are relevant for the problem under study. Those genes should be the first the biologists have to study.

Moreover, table 3 shows the interest of using the proposed multicriteria model as good solutions for one criteria are not always interesting for another one.

Table 2. Description of some Pareto solutions obtained with Conf E (YeastDB).

<table>
<thead>
<tr>
<th>Rules</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>IF ([VPS35=D] and [PRB1=I]) then [VPH1=D]</td>
</tr>
<tr>
<td>R2</td>
<td>IF ([DAR2=I] and [SVS1=D] and [KTR7=D] and [RP527A=D] and [PR02=D]) then [ARE2=I]</td>
</tr>
<tr>
<td>R3</td>
<td>IF ([POP3=D] and [TAF67=I] and [VPS36=D] and [SRP14=D]) then [MA1=ALPHA21=I]</td>
</tr>
<tr>
<td>R4</td>
<td>IF ([RF56A=D] and [RP026=D] and [SVS1=D] and [KTR7=D]) then [AK2=I]</td>
</tr>
<tr>
<td>R5</td>
<td>IF ([RN4=I] and [IPT1=I] and [GPM1=I] and [YTA12=I] and [PUP3=I] and [SAP185=D] and [ESC2=D]) then [GAR1=I]</td>
</tr>
<tr>
<td>R6</td>
<td>IF ([PSU1=D]) then [POP3=D]</td>
</tr>
</tbody>
</table>

Rules indicate the level of expression of the genes: I=Increase, D=Decrease.

Table 3. Quality of Pareto solutions presented table 2.

<table>
<thead>
<tr>
<th>Rules</th>
<th>S</th>
<th>Cj</th>
<th>I</th>
<th>R</th>
<th>Jm</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>0.582</td>
<td>0.978</td>
<td>1.288</td>
<td>2.968</td>
<td>0.147</td>
</tr>
<tr>
<td>R2</td>
<td>0.177</td>
<td>0.875</td>
<td>3.638</td>
<td>0.200</td>
<td>0.228</td>
</tr>
<tr>
<td>R3</td>
<td>0.278</td>
<td>0.916</td>
<td>2.203</td>
<td>0.425</td>
<td>0.227</td>
</tr>
<tr>
<td>R4</td>
<td>0.253</td>
<td>0.909</td>
<td>2.762</td>
<td>0.338</td>
<td>0.257</td>
</tr>
<tr>
<td>R5</td>
<td>0.063</td>
<td>1.000</td>
<td>7.900</td>
<td>0.072</td>
<td>0.130</td>
</tr>
<tr>
<td>R6</td>
<td>0.835</td>
<td>0.929</td>
<td>1.005</td>
<td>10.16</td>
<td>0.004</td>
</tr>
</tbody>
</table>

S=Support, I=Interest, R=Surprise, Cj=Confidence, Jm=Jmeasure.
6. Conclusion

In this work we have presented a multiobjective genetic algorithm for rule mining problems. Therefore, a multicriteria model has been proposed for association rules mining. Then, we have proposed a genetic algorithm able to look for the Pareto solutions regarding to the five selected criteria. We have presented here its application to analyze microarray experiment data. Through the experiments, advanced mechanisms proposed have been validated, and in particular the hybridization with an exact enumerative procedure.

In order to improve the use of such an algorithm and to speed up executions that may take several hours, we now work on a parallel implementation of the method. The parallelism should allow to have results in a more reasonable time, but should also allow to execute more intensification searches with the hybridization operator. Hence we will be able to give to biologists different hypothesis they will be able to evaluate.

References


8. S. Cahou, N. Melah, E-G Talbi, and M. Schoenauer. Paradisoo based design of


