

Parallel Genetic Algorithms for multi-objective rule mining

Mohammed Khabzaoui* Clarisse Dhaenens* El-Ghazali Talbi*

*LIFL, Université des Sciences et Technologies de Lille
UMR CNRS 8022, Bâtiment M3
59655 Villeneuve d'Ascq Cedex, FRANCE
{khabzaou, dhaenens, talbi}@lifl.fr

1 Introduction

In this work, we study a general model (association rule) to discover and describe associations between items in large databases. The association rule problem is modeled as a multi-objective combinatorial optimization problem. We propose to solve it using a cooperative evolutionary algorithm based on genetic algorithms. Therefore, specific mechanisms (mutation and crossover operators, elitism,...) have been designed and a parallel model is proposed to introduce more diversity and to find better solutions. Experiments are carried out on microarray data, where relations between gene expression levels of thousands of genes are looked.

2 Association rules discovery

Association rules were first formulated in [1] and was called the market-basket problem: Given a set of items and a collection of sales records, which consist in a transaction date and the items bought in the transaction, the task is to find relationships between the items contained in the transactions. In a more general context, where instances are described according several attributes, an association rule is an expression of the form: *IF C THEN P*. The IF part is called the rule condition (*C*) and the THEN part is called the rule prediction (*P*). Both parts contain a conjunction of terms indicating specific values for specific attributes.

The most famous algorithm to solve association rules is Apriori [1] which is based on the *support* (frequency of the rule) and the *confidence* (truth of the rule). But, according to biologists, who are studying microarray data, frequent rules are not necessarily interesting and rare rules for which the *confidence* is very high are also interesting.

Hence the question of the quality criterion to use in order to evaluate rules arises. A lot of measures exist for estimating the quality of association rules. For a global overview, readers can refer to Freitas [3] or Tan et al. [6].

Vienna, Austria, August 22–26, 2005

In order to evaluate rules in a complete way, we have previously determined a coherent set of five complementary criteria (*Support, Confidence, J-measure, Interest, Surprise*) [4]. As the number of possible rules may be huge, when the number of attributes (here genes) is important, we choose to adopt a combinatorial optimization approach using a multi-objective evolutionary algorithm for the resolution.

3 A Parallel Multi-Objective Genetic Algorithm (PMOGA)

In order to deal with the multi-objective association rule problem we have developed a parallel Genetic Algorithm. The aim is to find all the solutions of best compromise between the objectives (Pareto solutions). The parallel scheme involves a specific GA designed to solve the multi-objective association rule problem. This GA has been described in details in [4]. Here we recall its main characteristics and present the parallel model proposed.

3.1 A multi-objective Genetic Algorithm

In order to deal with the multi-objective association rule problem we have developed a specific Genetic Algorithm. The algorithm starts with a set of randomly generated solutions (population). The populations size remains constant throughout the GA. At each iteration, solutions are selected, according to their fitness quality (ranking) to form new solutions (offspring). Offspring are generated through a reproduction process (Crossover, Mutation). These operators have been designed for the rule mining problem.

As in a multi-objective optimization, we are looking for all the solutions of best compromise, best solutions encountered over generations are filed into a secondary population called the "Pareto Archive". In the selection process, solutions can be selected also from this "Pareto Archive" (elitism). A part of the offspring solutions replace their parents according to the replacement strategy. To have details about multi-objective operators implemented, the reader may refer to [4].

3.2 A parallel model

Parallel genetic algorithms may be classified into three main models: Global, Cellular and Island models. The Global model uses parallelism to speed up the sequential genetic algorithm. It uses a global shared population and the fitness evaluation is done on different processors. In the Cellular model the population is separated into a large number of very small subpopulations, which are maintained by different processors. In the Island model the population is divided into few large independent subpopulations. Each island evolves their population using a serial GA. For each island, some solutions occasionally migrate to another island.

We choose the Island model because this model seems to be adapted for the problem under study, where the search space is very large and requires a good diversity.

Island model: The island model implemented has been designed according to the ring topology. The model typically runs a serial multi-objective GAs on each processor (island) with independent populations and Pareto Archives (see Figure 1). Each GA starts with its

own parameters (population, parameters of GAs). Periodically, each island sends some solutions from its Pareto archive (randomly selected) to the neighboring island (this neighbor is defined by the topology).

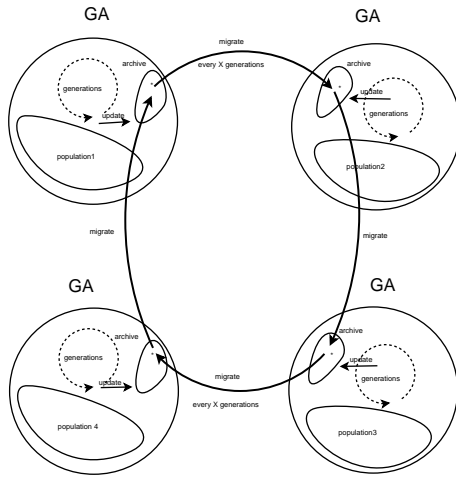


Figure 1: Island model.

In the proposed model each Island has to:

- create its population,
- evolve its population for a global number of generations and update its archive every generation,
- send some solutions from its "Pareto Archive" to the neighboring island,
- receive migrating solutions and replace the worst solutions by those immigrants (according to the ranking),

At the end, a specific island collects all the final Pareto Archives to create the global Pareto Archive.

An island model requires identification of the migration policy:

- What is the neighborhood
- How and when do migrations occur?
- How many solutions have to be sent?
- How to select emigrating solutions?
- Which solutions are to be replaced by the received solutions? structure?

4 Experiments

The aim here is to validate the interest of developing cooperative algorithms based on parallel genetic algorithms. First, we expose the indicators used to compare several models as they are specific to multi-objective optimization. Then, we describe the general experimental design. Finally we present results in two phases: we first try to find the best migration policy (when and how many solutions must be exchanged) and then we compare the cooperative model with non cooperative ones in order to assess the contribution of cooperation.

To evaluate the algorithm, we tested it on several microarray databases and expose here results for the public microarray database "MIPS Yeast Genome DB" containing 2467 genes for 79 chips.

Indicators for multi-objective experiments: In multi-objective optimization, solutions quality can be assessed in different ways. Some approaches compare the obtained front with the optimal Pareto front [7]. Other approaches evaluate a front with a reference point. Some performance measures do not use any reference point or front to evaluate an algorithm [5],

especially when the optimal Pareto front is not known at all.

Here, different versions of the proposed model will be compared, without knowing the true Pareto front. We propose to use two complementary performance indicators that allow to compare two by two the Pareto fronts obtained by different algorithms: the contribution and the entropy [2]. The contribution indicator quantifies the domination between two sets of non-dominated solutions. The contribution belongs to the interval $[0, 1]$ and a value greater than 0.5, indicates an improvement of the Pareto front. The entropy indicator gives an idea about the diversity of the solutions found. An entropy value belongs to the interval $[0, 1]$ and the more the entropy is close to 1, the better diversified is the front.

These indicators have been integrated to GUIMOO (**G**raphical **U**ser **I**nterface for **M**ulti-**O**bjective **O**ptimization (<http://www.lifl.fr/OPAC/guimoo>) which offers tools to analyze results of multi-objective methods.

Experimental design: Several versions of the algorithms will be compared. Default values of the parameters are: Population size = 300, Selection in population = $2/3$ (200), Mutation rate = 0.5, Crossover rate = 0.8, Selection in Pareto Archive (elitism) = 0.5.

The stopping criterion used is the non evolution of the archive during 10 generations, once the minimal number of generations has been overpassed.

Regarding the technical aspects, the used clustered machine is composed of 6 workstations with Intel Pentium IV - 3 GHz and 512 Mbytes main memory.

Selecting best parameters:

To select the best migration policy, we must answer to the questions “when” and “how many” solutions should have to be sent by the islands. Therefore these two parameters have been specifically studied.

Size of exchanges: The number of solutions sent by an island will be a proportion of the Pareto archive of this island. The tested proportions are : **2%**, **7%**, **10%**, **20%** and **50%**. Ten executions with each value have been executed. Results are reported in table 1. Each configuration is compared with all the others. Information about the average of the contribution is given. We can easily see here that the configuration 10% has all its contributions greater than 0.5. It seems to be the best configuration tested. These results show that too few or too much exchanges will not lead to a good cooperation and will not produce a very efficient front. Hence it seems important to choose a medium value for the number of solutions sent by the islands. For the rest of the experiments we will use the value of 10% of the local Pareto archive.

Table 1: Comparison of several scenarios: contribution

	“How many” scenario					“when” scenario					
	Average contribution					Average contribution					
	2%	7%	10%	20%	50%	5	10	25	50	80	
2%	-	0.47	0.47	0.51	0.51	5	-	0.50	0.48	0.46	0.54
7%	0.53	-	0.48	0.54	0.54	10	0.50	-	0.47	0.44	0.50
10%	0.53	0.52	-	0.54	0.56	25	0.52	0.53	-	0.46	0.49
20%	0.49	0.46	0.46	-	0.50	50	0.54	0.56	0.54	-	0.52
50%	0.49	0.46	0.44	0.50	-	80	0.46	0.50	0.51	0.48	-

Frequency of exchanges: The other very important question is “when” do islands have

Vienna, Austria, August 22–26, 2005

to exchange their Pareto solutions. In order to determine the best configuration, several tests have been realized with different occurrences. The different versions tested are the followings: every **5**, **10**, **25**, **50** and **80** iterations.

This parameter has a great importance as it allows to let islands evolving independently or to cooperate very often. Table 1 indicates the two by two contributions. We can see, one more time, that a parameter (here 50 generations) seems to overpass all the other, as its contribution is always greater than 0.5. One more time we can see that exchanging information too often or too rarely does not help the cooperation. It seems to be important to let each island evolving alone while receiving regularly good individuals from its neighbor.

Parallel Vs non Parallel: In order to assess the contribution of the cooperation, three different configurations have been tested (see figure 2). These configurations have been chosen in order to give to each configuration the same global population size.

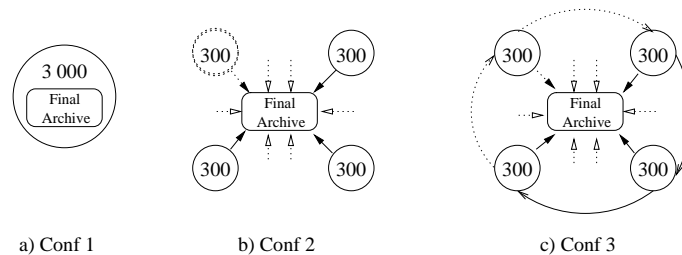


Figure 2: The three configurations tested.

- *Conf 1*: A single Genetic Algorithm, with a population size of **3 000** individuals (all the other parameters are the default parameters). The Pareto archive of the Genetic Algorithm is the final archive.
- *Conf 2*: Ten independent Genetic Algorithms with a population size of **300** individuals each (all the other parameters are the default parameters). The ten Genetic Algorithms contribute to the final Pareto archive.
- *Conf 3*: Ten cooperative Genetic Algorithms with a population size of **300** individuals each. The migration policy is to send 10% individuals every 50 iterations. The ten Genetic Algorithms contribute to the final Pareto archive.

	Average contribution			Average entropy		
	<i>Conf 1</i>	<i>Conf 2</i>	<i>Conf 3</i>	<i>Conf 1</i>	<i>Conf 2</i>	<i>Conf 3</i>
<i>Conf 1</i>	-	0.39	0.28	-	0.56	0.50
<i>Conf 2</i>	0.61	-	0.40	0.69	-	0.53
<i>Conf 3</i>	0.72	0.60	-	0.71	0.70	-

Table 2 indicates the two by two contributions and entropies of the different configurations tested. Contribution values show that *Conf 3* is the very best configuration. In fact, we can remark that $Conf\ 3 > Conf\ 2 > Conf\ 1$. This shows that the cooperation allows improving the quality of the Pareto front obtained. Moreover, these results show that as the *Conf 2* is better than the *Conf 1*, this problem requires a lot of diversity search, which may be difficult to achieve with a single genetic algorithm. The need of diversity may be confirmed by the

right part of table 2. Again, this table shows that the Pareto fronts obtained with *Conf 3* are the most diversified fronts. Hence, the cooperation allows obtaining efficient and diversified Pareto fronts, which was the initial objective of the study.

5 Conclusion

This work proposes to deal with a very challenging NP-hard problem: Rule mining. This interesting knowledge discovery task is a difficult problem and we adopt an optimization approach to obtain solutions.

As the search space of this rule mining problem is very large only heuristics are able to cope with it. Moreover, while defining the optimization criterion, the necessity of defining a multi-objective model appeared, which still makes the problem more complex! Hence we develop an evolutionary approach and propose a cooperative model, based on parallel genetic algorithms that evolve independently and exchange information.

This parallel model has been tested in order to determine the best parameters configuration. This model has also been compared with non cooperative approaches and results show the contribution of the cooperation.

References

- [1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proceedings of the 20th Intl. Conference on Very Large Databases, Santiago, Chile*, Sept 1994.
- [2] M. Basseur, F. Seynhaeve, and E-G. Talbi. Design of multi-objective evolutionary algorithms: Application to the flow-shop scheduling problem. In *Congress on Evolutionary Computation CEC'02*, pages 1151–1156, Honolulu, USA, 2002.
- [3] A. Freitas. On rule interestingness measures. *Knowledge-Based Systems journal*, 1999.
- [4] M. Khabzaoui, C. Dhaenens, and E-G. Talbi. A Multicriteria Genetic Algorithm to analyze DNA microarray data. In *Congress on Evolutionary Computation (CEC)*, volume II, pages 1874–1881, Portland, USA, juin 2004. IEEE Service center.
- [5] J. D. Knowles, D. W. Corne, and M. J. Oates. On the assessment of multiobjective approaches to the adaptive distributed database management problem. In *Proceedings of the Sixth International Conference on Parallel Problem Solving from Nature (PPSN VI)*, pages 869–878, september 2000.
- [6] P-N. Tan, V. Kumar, and J. Srivastava. Selecting the right interestingness measure for association patterns. In *Proceedings of the Eight ACM SIGKDD conference, Edmonton, Canada*, 2002.
- [7] D. A. Van Veldhuizen and G. B. Lamont. On measuring multiobjective evolutionary algorithm performance. In *In 2000 Congress on Evolutionary Computation, Piscataway, New Jersey*, volume 1, pages 204–211, Jul 2000.

Vienna, Austria, August 22–26, 2005