Combining Evolutionary Algorithms and exact approaches for multi-objective knowledge discovery

Mohammed Khabzaoui, Clarisse Dhaenens, El-Ghazali Talbi

RAIRO-Oper. Res. 42 (2008) 69–83
DOI: 10.1051/ro:2008004

RAIRO Operations Research
www.rairo-ro.org

COMBINING EVOLUTIONARY ALGORITHMS AND EXACT APPROACHES FOR MULTI-OBJECTIVE KNOWLEDGE DISCOVERY

Mohammed Khabzaoui[1], Clarisse Dhaenens and El-Ghazali Talbi[1]

Abstract. An important task of knowledge discovery deals with discovering association rules. This very general model has been widely studied and efficient algorithms have been proposed. But most of the time, only frequent rules are seeked. Here we propose to consider this problem as a multi-objective combinatorial optimization problem in order to be able to also find non frequent but interesting rules. As the search space may be very large, a discussion about different approaches is proposed and a hybrid approach that combines a metaheuristic and an exact operator is presented.

Keywords. Hybridization, multi-objective optimization, knowledge discovery, association rules.

Mathematics Subject Classification. 90Cxx, 68XX.

1. Introduction

Knowledge discovery is an important issue in a world where data is easily obtained and has to be carefully studied in order to extract interesting information. Several tasks may be differentiated in knowledge discovery such as supervised classification, clustering, feature selection or association rules mining. This last task refers to a very general model that allows relationships to be found between items of a database.

Received October 30, 2006. Accepted October 30, 2007.

[1] Polytech'Lille, LIFL – CNRS / INRIA – Bâtiment M3, University of Lille 1, 59655 Villeneuve d'Ascq Cedex, France; Clarisse.Dhaenens@lifl.fr; El-Ghazali.Talbi@lifl.fr

Article published by EDP Sciences

The association rules problem was first formulated in [1] and was called the market-basket problem. The initial problem was the following: given a set of items and a large 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 different transactions.

Since this first application, many other problems have been studied with association rules that may be defined in a more general way. Let us consider a database composed of transactions (rows) described according to several – maybe many – attributes (columns). The value of the attributes may be a presence/absence marker (binary data), a description belonging to a small number of possibilities (nominal data) or a real value (numerical data).

Then, an association rule is an expression of the form: *IF C THEN P*. This kind of rules contains two parts: the IF part which is called the rule condition (C) and the THEN part which is called the rule prediction (P), where both parts, the C and the P parts, contain a conjunction of terms indicating specific values for specific attributes. Hence the general form of an association rule is $IF\ \langle term_1\rangle\ and\ \langle term_2\rangle\ and\ ...\ \langle term_p\rangle\ THEN\ \langle term_{p+1}\rangle\ and\ ...\ \langle term_n\rangle$, where $\langle term_i\rangle$ is of the form $\langle attribute_i = value_{i_j}\rangle$.

Association rules mining has been widely studied in the literature, but most of the time only frequent patterns are seeked. Here we propose to look for *interesting* associations and not only frequent ones. This research has been motivated by the type of data we are exploiting. Our applications deal with rule mining in micro-array data. In such genomic data, expression levels of thousands of genes are measured according to several experimental conditions and/or for several individuals (for example patients/healthy persons). In this context, the different conditions are the transactions (rows) and the genes are the attributes (columns). Biologists think that mining frequent rules is not always interesting as it may reveal already known associations. On the contrary, mining non frequent rules may reveal associations that may occur in a subset of experiments, or for a subset of individuals, and may explain some specific cases such as a specific disease.

In this context we have studied different quality criteria for association rules and we propose a multi-objective model for rules mining. This model will be briefly presented in this article.

The association rules problem may be seen as a combinatorial optimization problem, as rules are combinations of attributes, and in our applications the number of attributes (genes) may reach several thousands. Hence the combinatoric is very high. Moreover the chosen model deals with multi-objective optimization. Then, the question is: which optimization approach to choose? exact approach or (meta)heuristic? Regarding the advantages and the drawbacks of each type of approach, the choice has been made to hybridize both types of approaches.

Hence this article first presents the multi-objective model proposed to deal with association rules. Then, it exposes several methods available to solve the multi-objective association rules problem. It presents their advantages and drawbacks. Section 4 proposes a hybrid approach that combines both types of method. Experiments are presented in order to first find the best parameters for the hybridization

TABLE 1. Some quality measures.

Measure	**Formula**
Support (S)	$S = \frac{\|C \text{ and } P\|}{N}$
Confidence (Cf)	$C = \frac{\|C \text{ and } P\|}{\|C\|}$
Laplace (L)	$L = \frac{\|C \text{ and } P\|+1}{\|C\|+2}$
Interest (I)	$I = \frac{N*\|C \text{ and } P\|}{\|C\|*\|P\|}$
Conviction (V)	$V = \frac{\|C\|*\|\overline{P}\|}{N*\|C \text{ and } \overline{P}\|}$
Surprise (R)	$R = \frac{\|C \text{ and } P\|-\|Cand\overline{P}\|}{\|\overline{P}\|}$
Jaccard (ζ)	$\zeta = \frac{\|C \text{ and } P\|}{\|C\|+\|P\|-\|C \text{ and } P\|}$
Phi-coefficient (ϕ)	$\phi^2 = \frac{(\|C \text{ and } P\|*\|\overline{C} \text{ and } \overline{P}\|-\|C \text{ and } \overline{P}\|*\|\overline{C} \text{ and } P\|)^2}{\|C\|*\|P\|*\|\overline{C}\|*\|\overline{P}\|}$
Cosine (IS)	$IS = \frac{\|C \text{ and } P\|}{\sqrt{\|C\|*\|P\|}}$
Jmeasure (J)	$J = Pr(P) * [Pr(C\|P) \log(\frac{Pr(C\|P)}{Pr(C)}) + (1 - Pr(C\|P)) \log(\frac{1-Pr(C\|P)}{1-Pr(C)})]$
J1measure ($J1$)	$J1 = Pr(P) * [Pr(C\|P) \log(\frac{Pr(C\|P)}{Pr(C)})]$
JFmeasure(JF)	$JF = \frac{(w1\times J1+w2\times(\frac{N_{pu}}{NT}))}{w1+w2}$
PiatetskyShapiro (PS)	$PS = \frac{\|C \text{ and } P\|}{N} - \frac{\|C\|}{N} * \frac{\|P\|}{N}$

and then to show the contribution of the hybridization. Finally we conclude the paper in Section 6.

2. MULTI-OBJECTIVE ASSOCIATION RULES

In order to solve association rules 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 [8], Tan *et al.* [19] or Hilderman *et al.* [9]. Some of them are presented in Table 1.

Formulas are given for a set of N instances, where $|C|$ represents the number of instances satisfying the C part of the rule, $|P|$ the number of instances satisfying the P part of the rule, $|C \text{ and } P|$ is the number of instances satisfying simultaneously the C and P parts of the rule. Pr is a probability.

A statistical study (correlation analysis, principal composant analysis ...) [12] of these criteria showed that some of them are very correlated. This study leads us to determine five groups of criteria, where each group is composed of correlated

criteria [13]. We chose to take one criterion of each group and we obtained five complementary criteria that allow to evaluate rules in a complete way. Chosen criteria are *Support*, *Confidence*, *Jmeasure*, *Interest* and *Surprise*.

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.

Confidence (Cf). The *Confidence* measures the validity of a rule. It is the conditional probability of P given C.

Jmeasure. Smyth and Goodman [17] have proposed the Jmeasure, which estimates the degree of interest of a rule and combines support and confidence. It is used in optimization. During previous uses of genetic algorithms to extract rules, some authors have observed that in the first generations of a genetic algorithm using the $Jmeasure$ as an evaluating function, the quality was often equal to zero or rules covered any examples. Wang *et al.* [21] have proposed another measure: the $J1measure$. But the drawback of this formula is to cause a very low convergence of the algorithms. Araujo *et al.* [2] have then proposed the $JFmeasure$ where N_{pu} is the number of potentially useful attributes of the C part of the rule and NT the total number of attributes of the C part. An attribute is said potentially useful if it appears in the C part of at least one solution of the population. $w1$ and $w2$ are two user-specified parameters chosen between 0 and 1.

Interest (I). The $Interest$ measures the dependency while privileging rare patterns in the region of weak support. It takes values in the interval $[0\ ,\ \infty[$.

- C and P are independent if $Interest$ equals to 1.
- The rule is interesting if the $Interest$ is in the interval $]1\ ,\ \infty[$.

Surprise (R). It is used to measure the affirmation. It enables to search surprising rules. The $Surprise$ takes positive real values ($[0\ ,\ \infty[$).

Thanks to these five criteria, association rules may be evaluated in a complete way without privileging a specific criterion. With this multi-criteria model, the objective is to find, the Pareto solutions, that represent the best compromises between criteria.

3. Exact methods versus metaheuristics

Considering the association rules problem as an optimization one, may lead to different approaches to solve it. Either exact methods may be used in order to solve small instances, or heuristics, and particularly metaheuristics, are developed to approximate best solutions on large instances. In this section we propose to study the opportunity of developing exact methods for the multi-criteria rule mining problem and discuss of their limitations. Then we briefly present a genetic algorithm developed to deal with this problem.

3.1. *A priori*: an intelligent enumerative procedure

The most famous and the most commonly used algorithm to solve association rules is *a priori* [1]. The aim of this algorithm is to enumerate in a very efficient

way all the rules that respect a minimal given *Support* (frequent rules) and a minimal level of *Confidence* (truth of the rule). This method may be seen as an exact method, as it allows to enumerate all the possible rules satisfying predefined levels of *Support* and *Confidence*.

The efficiency of this algorithm is based on the fact that the *Support* is monotone. The *Support* of an itemset of size n is smaller than or equal to the *Support* of any of its subsets of size $n-1$. Hence the search of frequent itemsets (groups of items that are often together in the database) may be done in an increasing manner. Starting from itemsets of size 1, then itemsets of size 2, and so on, *a priori* looks for frequent itemsets of size n that are composed of frequent itemsets of size $n-1$. In a second phase, using frequent itemsets, *a priori* constructs rules that respect the minimal level of *Confidence*.

A lot of improvements of the initial method have been proposed, and in particular there exist parallel versions of it [22]. Now we can say that *a priori*-like methods are able to deal with problems with a large number of transactions and a quite large number of attributes.

Let us note that another interesting algorithm, based on a similar principle, is **Eclat**. The main differences between these two algorithms are how they construct the frequent itemsets and how they compute the *Support* [4].

3.2. Adapting the *A priori* algorithm to other criteria?

In a context of multi-objective optimization, an interesting question would be: how to adapt the *a priori* algorithm to other criteria?

In order to answer this question a study of the criteria should be done. It appears that no other criterion verifies the monotony property of the *Support*. For example, the *Confidence* of an itemset of size n may be greater than the *Confidence* of its subsets of size $n-1$. Hence the *a priori* algorithm may not be generalized to any other criterion.

In this context, where no domination properties may be found between an itemset and its subsets, the only way to exactly find the set of Pareto solutions is to use an enumerative procedure (EP) that constructs all the possible rules given a set of nb attributes and stores all the optimal solutions according to the classical dominance notion in multi-objective combinatorial optimization (Pareto rules).

We propose the following procedure:

```
EP: EnumProc(AttributeSet,nb)
C ← ∅                          // the C part of the rule under construction
P ← ∅                          // the P part of the rule under construction
ParetoSet ← ∅                          // Set of Pareto solutions found
first ← 1                      // indicates the first attribute to consider
ConstructRule(AttributeSet,nb,first,C,P,ParetoSet)
```

where `ConstructRule` builds recursively all the possible rules and update the Pareto set as follows:

```
ConstructRule(AttributeSet, nb, first, C, P, ParetoSet)
// first attribute in the rule
// in the C part
   for every value_i of attrib_first
      C = C ∪ < attrib_first = value_i >
      if (P ≠ ∅)
         Evaluate(C,P)
         UpdatePareto(C,P,ParetoSet)
      if (first < nb)
         ConstructRule(AttributeSet, nb, first + 1, C, P, ParetoSet)
      C = C \ < attrib_first = value_i >
// in the P part
   if (P = ∅)
      for every value_i of attrib_first
         P = < attrib_first = value_i >
         if (C ≠ ∅)
            Evaluate(C,P)
            UpdatePareto(C,P,ParetoSet)
         if (first < nb)
            ConstructRule(AttributeSet, nb, first + 1, C, P, ParetoSet)
         P = P \ < attrib_first = value_i >
// first attribute not in the rule
if (first < nb)
       ConstructRule(AttributeSet, nb, first + 1, C, P, ParetoSet)
```

Let us specify that in our application, the P part of the rule only contains a single term. But this proposed procedure may be generalized for any number of attributes in the P part. In this case, the condition *if* $(P = \emptyset)$ should be removed.

Moreover, we bring to the attention of the reader that this procedure is enumerative and can not deal with a large number of attributes in a reasonable time as the number of generated rules may become huge. For example, in the simple binary case (where attributes may only be '0' or '1'), the number of rules generated with nb attributes is equals to

$$C_1^{nb} \times \sum_{i=1}^{i=nb-1} C_i^{nb-1}.$$

Hence, to deal with a large number of attributes, a heuristic has to be developed. We propose to use an evolutionary approach.

3.3. A dedicated genetic algorithm

Genetic algorithms have already shown their capability to deal with large size multi-objective optimization problems [5] and we decided to develop a specific genetic algorithm for the multi-objective association rules problem.

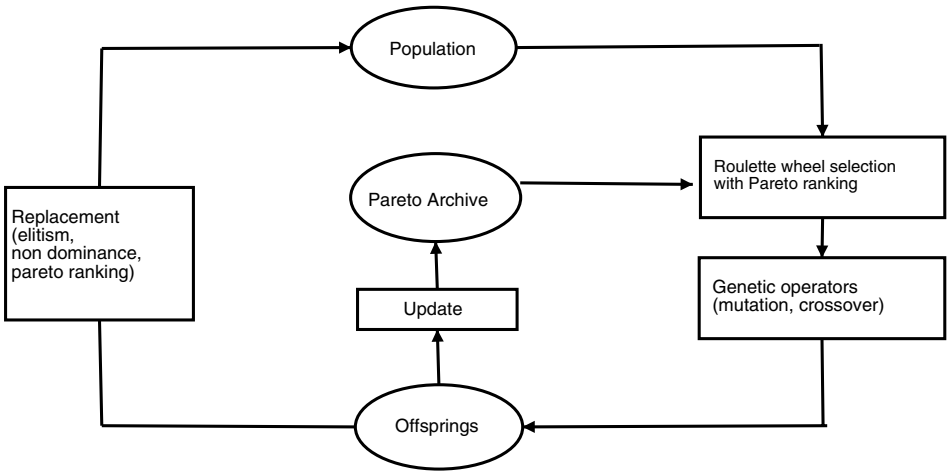

FIGURE 1. A multi-objective genetic algorithm.

Figure 1 presents the multi-objective genetic algorithm scheme adopted. As the aim is to look for all the solutions of best compromise, best solutions encountered over generations are filled into a secondary population called the "Pareto Archive". In the production process, elitism is applied in order to allow solutions from the "Pareto archive" to participate to the reproduction.

Specific operators have been proposed for the association rules problem and have been validated on several types of datasets. These operators are briefly explained below and to have more details about them, the reader may refer to [14].

Selection operator. We use the classical roulette selection based on the ranking notion where, the probability of selection of a solution is proportional to its rank. We use the Pareto ranking [7]. The rank of a solution corresponds to the number of solutions, in the current population, by which it is dominated.

Crossover. The crossover mixes the features of two rules by the combination of their attributes. The proposed crossover operator has two versions, to take into account the fact that the parents may share (or not) a common attribute:

- *Crossover by value exchange:* If two rules X and Y have one or several common attribute(s) in their C parts, one common attribute is randomly selected. The value of the selected attribute in X is exchanged with its counterpart in Y.
- *Crossover by insertion:* 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.

Mutation. Four mutation operators were implemented. The choice of the mutation operator is not made on advance, but the probability of appliance of a mutation operator is made in an adaptive manner (see next paragraph). At the

beginning of the algorithm, all the mutation operators have the same probability to be selected.

- The *Value mutation* that replaces an attribute value by a randomly chosen one.
- The *Attribute mutation* that replaces a term by another. The value of the new attribute is randomly chosen in its domain.
- The *Insertion operator* that adds a term (randomly chosen attribute with a randomly chosen value) in the rule.
- The *Delete operator* that removes a term of the rule (if the number of terms is greater or equal to 3).

Adaptive mutation rate. Setting the probabilities of appliance of these four mutation operators may be difficult. Moreover, the more interesting operator at a given time of the search may not be always the same. To overcome this problem, we implement an adaptive strategy for calculating the rate of application of each mutation operator according to the method proposed by Hong *et al.* for the mono-criterion case [10]. This method favors operators that often improve solutions. It calculates the "improvement ratio" of each operator and determines the probabilty of appliance of each operator, using this indication. Extending their method to the multi-objective case requires to define the progress made by an operator. We propose to evaluate this progress comparing the rank (Pareto ranking) of the solution obtained after the application of the operator with the rank of the initial solution. Then the new selection probabilities of the mutation operators are computed proportionally to the progress calculated. Details about the calculation may be found in [3].

Replacement operator. We use the elitist non dominated sorting replacement. The worst ranked solutions are replaced by dominating solutions (if there is any) generated by crossover and mutation operators (offspring). 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. This archive has to be updated each time a solution is added.

Elitism. The Pareto solutions are not only stored permanently, they also take part of the selection and may participate to the reproduction. Therefore a probability of selecting a parent from the archive is set

This approach has been previously tested on public knowledge discovery databases (from UCI, for example) and on micro-array data [14]. If it has shown a good convergence and a high diversity power, we can still wonder whether results obtained may be ameliorated using an hybrid approach.

4. A hybrid approach

We saw in Section 3 that exact methods were not usable on large problems and that the genetic algorithm developed offers a good diversity. In this section

we investigate whether the cooperation of the two types of methods can improve results.

4.1. Motivations

Recently, the interest for cooperative methods has become important. Cooperation or hybridization between methods may be expressed in different manners:

- methods of several types are combined in order to obtain better results than those obtained with a single type of these methods;
- a same algorithm is used several times in parallel and each copy cooperates;
- the two preceeding items may be combined.

Here we are interested in combining two types of methods in order to take advantage of both of them. Numerous such hybrid methods have been proposed. Most of the time a first heuristic gives solution(s) to a second heuristic which tries to ameliorate it(them). In the taxonomy of hybrid methods for mono-objective optimization proposed by Talbi [18], several hybridization schemes are proposed. A recent survey has been published by Puchinger and Raidl on this subject [16].

The main motivation of our approach has been driven by previous experiments. It appears that in our experimental context (discovering association rules in genomic data, coming from micro-array experiments), the search space is huge. Moreover, criteria used have no interesting properties that would help during the exploration of the search space. Hence it is necessary to have an efficient exploration agent, which will be the genetic algorithm. But exploring without exploiting regions would not lead to obtain very good solutions. Therefore, an intensification agent is also required and this will be the enumeration procedure proposed Section 3.2. This hybridization scheme is similar to the one used by Cotta and Troya, where a Branch and Bound is used as an operator within an evolutionary algorithm [6].

4.2. Description of the approach

The aim of this approach is to combine advantages of the two cooperating methods. During the execution of the Genetic Algorithm (GA), from times to times (this parameter will have to be settled) the GA will call the Enumerative Procedure (EP) in order to explore a specific region. This is represented by Figure 2, where the Enumerative Procedure (EP) is used as a crossover operator when the number of distinct attributes composing the two rules is not too large. Given two rules, the set of distinct attributes is extracted from these rules. If the cardinality of this set of attributes is not too large, those attributes will be used to enumerate all the possible rules they can form. Otherwise, the normal crossover of the GA is applied. This is described by the algorithm presented hereafter.

As a result of such an exact crossover, several (maybe a lot) local Pareto solutions are produced. All these Pareto solutions are stored in the local archive of the operator. This local archive is then used in order to update the global Pareto archive of the Genetic Algorithm by adding new Pareto solutions to the archive

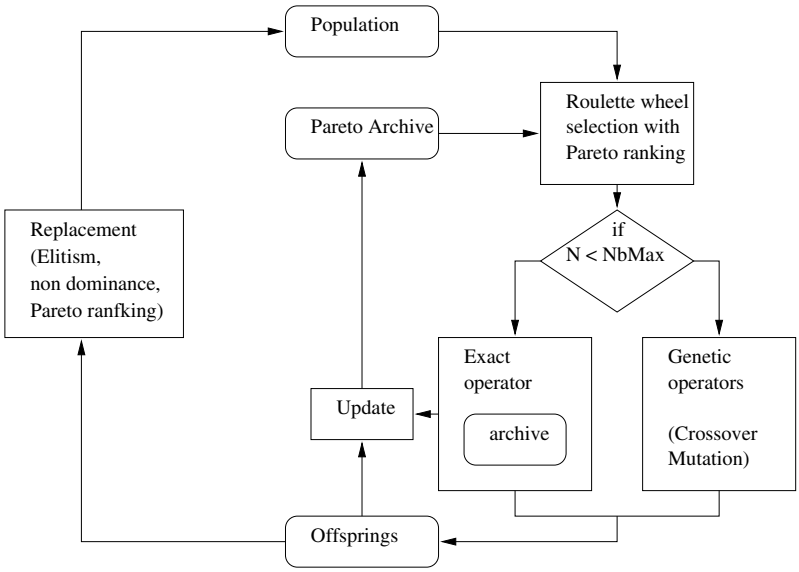

FIGURE 2. A multi-objective genetic algorithm.

and removing solutions that are dominated by a new inserted Pareto solution. Moreover, as this Enumerative Procedure is used as a crossover between two rules, two randomly chosen rules are selected from the local Pareto archive in order to become offspring of the crossover. The crossover may be described as follows:

`Crossover(`$Rule_1, Rule_2$`)`

$AttributeSet \leftarrow Attrib_{Rule_1} \cup Attrib_{Rule_2}$
`// Construction of the Attribute set`

$nb \leftarrow \mid AttributeSet \mid$ `// computation of the number of attributes`

`if (`$nb \leq MaxNb$`)`
`EnumProc(AttributeSet,nb)`
`else`
`NormalCrossover(`$Rule_1, Rule_2$`)`

Several parameters have to be settled in order to define the hybridization scheme:

- $MaxNb$ is the maximum number of attributes we allow. This parameter is linked to the maximum time we allow to the Enumerative Procedure.
- When does the Enumerative Procedure have to be used? Every generation? One over x generations?
- Does it have to be used on all the couples formed during a generation or only on a sample of them?

The next part will answer these questions by comparing several versions of the hybridization.

5. Experiments

To evaluate the algorithm and to test the different parameters, we applied it on a public micro-array database "MIPS Yeast Genome Database" containing 2467 genes for 79 chips. In this problem, 2467 attributes are candidate to form rules and 79 relations between those attributes are given to evaluate the rules.

5.1. Indicators used: the contribution and the D-Metric

In multi-objective optimization, solutions quality can be assessed in different ways. Some approaches compare the obtained front with the optimal Pareto front [20]. Others approaches evaluate a front with a reference point [11]. Some performance measures do not use any reference point or front to evaluate an algorithm [15], especially when the optimal Pareto front is not known at all.

Here, we want to compare two by two different versions of the algorithm, without knowing the true Pareto front. We propose to use two indicators: the contribution [3] and the D_{metric} based on the S_{metric} [23]. The contribution indicator quantifies the domination between two sets of non-dominated solutions. The contribution belongs to the interval $[0, 1]$. The S_{metric} calculates the hypervolume (in the objective space) of the k-dimensional region covered by the approximation set dominated by the solutions of the Pareto front. The D_{metric} allows to compare two fronts using the S_{metric}. It represents the part of the region dominated by the first front A and not dominated by the second front B ($D_{\text{metric}}(A, B) = S_{\text{metric}}(A, B) - S_{\text{metric}}(B)$).

5.2. Experimental design

Several versions of the algorithms are compared in order to determine the best parameters linked to the hybridization process to evaluate the efficiency of the hybridization.

For all the experiments, the default parameters of the GA have been defined regarding previous experiments realised on this method (see [14]). These default parameters are:

- Population size = 150.
- Selection in population = 1/3.
- Global Mutation rate = 0.5.
- Crossover rate = 0.8.
- Selection in Pareto archive (elitism) = 0.5.
- Minimal number of generations = 200.
- Maximal number of attributes for the enumeration procedure (MaxNb) = 10.

TABLE 2. Performances of the different versions.

Frequencies	Average contribution	Average Dmetric
0%	0.03	5.20
20%	0.19	4.44
50%	0.22	4.04
100%	0.29	3.80

The different versions of the hybridization deal with the frequency of application of the enumerative procedure (EP). In this context, one generation over x generations the normal crossover will be replaced by the Enumerative Procedure. This frequency will vary with the following parameters: Never (frequency 0%), One generation over five (frequency 20%), One generation over two (frequency 50%), Each generation (frequency 100%).

The more the EP is used, the more evaluations will be required to execute this procedure. The aim is to see if such a procedure, is able to still improve results obtained using a genetic algorithm dedicated to the problem under study. Hence these configurations will be compared in terms of quality of solutions produced.

5.3. RESULTS

In order to assess results, 20 executions of each version have been realized. Values given are the mean values obtained for the ten executions.

Table 2 indicates the contribution and D_{metric} comparing each version with the global front (generated using all the solutions found during all the executions of all the versions). The first column indicates the frequency of appliance ot the exact operator. The second column gives the average contribution of each version compared with the global front. It appears, as we could expect, that the more the Enumerative Procedure is used, the best are the results. The third column indicates the average D_{metric} of the global front with each version compared. Indeed, executing the Enumerative Procedure at each generation is the best version.

This may be confirmed by Figure 3 which shows for each frequency the average improvements regarding all the other frequencies.

But, this figure shows a more interesting fact. We can see that the improvement is non linear (logarithmic shape) compared to the increase of the frequency of application of the exact operator. On the contrary, the increase of computational time required for the execution of the algorithm is linear compared to the increase of the frequency of application. Indeed, the computational effort of the algorithm is dominated by the Enumerative Procedure (EP) as the computational time is almost devoted to it. In this context, executing the Enumerative Procedure one generation over five (20% of appliance) would require twice the time needed for a 10% appliance rate.

Experiments show that improvement of quality requires a lot of additional computational time. Hence, it is worth looking for a good compromise between time and quality. On the studied example, it can be seen that using the Enumerative

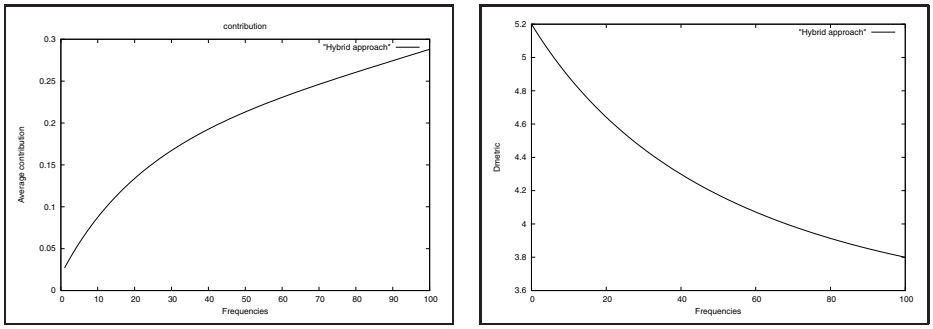

FIGURE 3. Evolution of the contribution and the Dmetric space with the frequency of appliance.

Procedure with a frequency of 50% (one iteration over two), for example, allows a good improvement of quality (only 30% above best performances) while dividing the execution time by two regarding to the 100% application rate. Hence an application rate of 50% could be a good compromise.

6. Conclusion

In this paper, the problem of association rules discovery has been modeled as a multi-objective combinatorial optimization problem. After a discussion about methods that may be used to solve such a problem, a hybrid approach, combining a dedicated genetic algorithm and an enumerative procedure is proposed. The frequency of application of the enumerative procedure is studied through experiments. It appears that the procedure is time consuming and a choice has to be made between quality of solutions and time allowed.

In order to accelerate the execution of the algorithm, it may not be interesting to execute the enumerative procedure at the beginning of the genetic algorithm. An idea would be to let the genetic algorithm converge and to launch the intensification search only when solutions produced by the GA are of good quality. It does not seem interesting to intensify the search around very bad solutions. Time will be wasted for nothing.

Moreover, another interesting perspective would be to parallelize the execution of the enumeration operator, as it is time consuming. At each iteration, quite a lot of enumerations should be done (one for each crossover). Waiting for all these enumerations to be sequentially executed is the task that is time consuming and that could be accelerated if several processors are dedicated to this task.

Finally, this Enumerative Procedure could also be used in order to exploit the neighborhood of a single solution. Hence it can be interesting to apply it at the end of the algorithm, on all solutions belonging to the archive of the Genetic Algorithm. This may generate new Pareto solutions that can dominate some solutions proposed by the GA.

References

[1] R. Agrawal and R. Srikant, Fast algorithms for mining association rules, in *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, edited by J.B. Bocca, M. Jarke, and C. Zaniolo, Morgan Kaufmann **12** (1994) 487–499

[2] D.L.A. Araujo, H.S. Lopes and A.A. Freitas, A Parallel Genetic Algorithm for Rule Discovery in Large Databases, in *Proc. 1999 IEEE Systems, Man and Cybernetics Conf.*, Vol. **III** (1999) 940–945, Tokyo, Japan.

[3] M. Basseur, F. Seynhaeve and E-G. Talbi, Adaptive mechanisms for multi-objective evolutionary algorithms. *IMACS multiconference, Computational Engineering in Systems Applications (CESA'03), IEEE Service Center, Piscataway, New Jersey*, S3-R-00-222:100–107 (2003).

[4] C. Borgelt, Efficient implementations of apriori and eclat, in *Workshop Frequent Item Set Mining Implementations (FIMI 2003, Melbourne, FL, USA)* **90** (2003).

[5] C.A. Coello, D.A. Van Veldhuizen and G.B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers (2002).

[6] C. Cotta and J.M. Troya, Embedding branch and bound within evolutionary algorithms. *Appl. Intell.* **18** (2003) 137–153

[7] C.M. Fonseca and P.J. Fleming, An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Comput.* **3** (1995) 1–16.

[8] A. Freitas, On rule interestingness measures. *Knowledge-Based Syst. J.* **12** (1999) 309–315.

[9] R. Hilderman and H. Hamilton, Knowledge discovery and interestingness measures: A survey, technical report cs 99-04. Technical report, Department of Computer Science, University of Regina, October (1999).

[10] T.P. Hong, H. Wang and W. Chen, Simultaneously applying multiple mutation operators in genetic algorithms. *J. Heuristics* **6** (2000) 439–455.

[11] A. Jaszkiewicz, On the performance of multiple objective genetic local search on the 0/1 knapsack problem. a comparative experiment. Technical Report RA-002/2000, Institute of Computing Science, Poznan University of Technology, Poznan, Poland (2000).

[12] M. Khabzaoui, C. Dhaenens, A. N'Guessan and E.-G. Talbi, Etude exploratoire des critères de qualité des règles d'association en datamining, in *Journées Françaises de Statistique* (2003) 583–587.

[13] M. Khabzaoui, C. Dhaenens and E.-G. Talbi, Association rules discovery for DNA microarray data. *Bioinformatics Workshop of SIAM International Conference on Data Mining* (2004) 63–71.

[14] M. Khabzaoui, C. Dhaenens and E.-G. Talbi, A Multicriteria Genetic Algorithm to analyze DNA microarray data, in *Congress on Evolutionary Computation (CEC)*, Vol. **II**, pp. 1874–1881, Portland, USA (2004). IEEE Service center.

[15] 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)* (2000) 869–878

[16] J. Puchinger and G.R. Raidl, Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification, in *First international Work-Conference on the Interplay between Natural and Artificial Computation (IWINAC)* **3562** (2005) 41–53.

[17] P. Smyth and R.M. Goodman, *Knowledge Discovery in Databases*, Chapter Rule Induction Using Information Theory, G. Piatetsky-Shapiro and J. Frawley (1991) 159–176.

[18] E.-G. Talbi, A taxonomy of hybrid metaheuristics. *Journal of Heuristics* **8** (2002) 541–564.

[19] 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).

[20] D.A.Van Veldhuizen and G.B. Lamont, On measuring multiobjective evolutionary algorithm performance, in *In 2000 Congress on Evolutionary Computation*. Piscataway, New Jersey, Vol. **1**, 204–211 (2000).

[21] K. Wang, S.H.W. Tay and B. Liu, Interestingness-based interval merger for numeric association rules, in edited by *Proc. 4th Int. Conf. Knowledge Discovery and Data Mining, KDD*, R. Agrawal, P. E. Stolorz, and G. Piatetsky-Shapiro, pp. 121–128. AAAI Press, (1998) 27–31. New York, USA.

[22] M.J. Zaki, Parallel sequence mining on shared-memory machines. *J. Parallel and Distrib. Comput.* **61** (2001) 401–426.

[23] E. Zitzler and L. Thiele, Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Trans. Evol. Comput.* **3** (1999) 257–271.