Variable Optimization for Flood Prediction

Wilfried Segretier* — **Martine Collard*** — **Laurent Brisson**** — **Jean-Emile Symphor*****

* *LAMIA - Université des Antilles et de la Guyane*
Campus de Fouillole B.P. 592
97159 Pointe-à-Pitre Cedex

** *Institut Télécom - Télécom Bretagne*
UMR CNRS 3192 Lab-STICC
Technopôle Brest Iroise CS 83818
29238 Brest Cedex 3

*** *Ceregmia-Martinique*
Campus de Schoelcher B.P. 7209
97275 Schoelcher Cedex

ABSTRACT. *In this paper, we present an heuristic based approach for feature selection in the context of flood prediction. Features are complex variables that represent aggregate values. We apply a preprocessing method on data in order to elicit relevant information that could not be easily accessible initially because it is split through several lines of a dataset. A genetic algorithm is used in order to search for the features that may prove the best performances for flood prediction.*

RÉSUMÉ. *Dans cette étude, nous présentons, une méthode permettant de découvrir, par le biais de variables agrégées et d'un algorithme génétique, des informations non connues a priori et utiles pour la prédiction de crues.*
Cette méthode se prête particulierement aux données dans lesquelles des informations à propos d'un même élément sont dispersées à travers plusieurs lignes du jeu de données initial.

KEYWORDS: *Data mining, Genetic Algorithm, Feature selection, Complex variables, Floods prediction*

MOTS-CLÉS : *Fouille de données, Algorithmes Génétiques, Sélection de caractéristiques, Variables complexes, Prédiction de crues*

1re soumission à *Revue ISI - Numéro spécial "Systèmes d'Information et de Décision pour l'Environnement"*, le December 5, 2010

1. INTRODUCTION

River floods are complex phenomena that may be due to multiple factors. As their impact on traffic and communications may be disastrous, forecasting has been considered as a major challenge for years. Main techniques used for designing flood forecasting systems are based on hydrologic models that take into account stream flow routing methods to predict flow rates and water levels for time periods ranging from a few hours to several days ahead. Data mining and optimization techniques may provide alternative or complementary solutions, particularly for selecting sound predictive features. Indeed optimized feature selection has been identified as a major issue in data mining processes since first research attempts in the domain.

In this paper, we focus on the example of a particular river for which an hydrologic forecasting system was designed. Limnimetric levels (water levels) are recorded by sensors on given spots all along the river course. Currently, flood alarms are triggered when one sensor is recording a limnimetric level reaching a predefined threshold. This system performs rather well to predict high limnimetric values on the whole river with low rates of false positives (FP) and false negatives (FN). But the inherent phenomenon of flood in the river basin surroundings is not well managed. With same water levels observed on same spots at a same time before a flood, sensitive areas down below the river bed may be under water or not depending on other factors obviously. The issue to address is thus to study the whole mechanism that result in flooding all these areas.
This work was initiated and funded by the General Council of the island of *La Martinique* in French West Indies who is much concerned by river flood problems since strategic places in the island (main roads, airport, industrial areas) are threatened.

Our final objective is to propose a new model for this natural system that we can assume to be complex since factors like water height, flow, rainfall, saturation rate, slope rate or ground types that behave rather independently seem to infer collectively the flood phenomenon. One practical advantage with such a model will be to optimize limnimetric and rainfall sensor locations on the river basin. The complex system approach we plan to adopt consists in a first stage on applying data mining and optimization techniques in order to extract the most relevant knowledge on each factor behaviour. A second stage will be to simulate these individual models and merge them as multi-agents in order to observe possible emerging collective phenomena inducing similar floods as those observed.

In this paper, we focus on the first stage of the project to propose an extensible method in order to optimize the selection of features for predicting high limnimetric levels on the last sensor downstream the river bed. This sensor state may be considered as the prediction variable since it is located very close to a threatened area. We assume thus that a threshold overflow on this sensor (called *event sensor*) is equivalent to

flood occurrence. The challenge is not only to globally optimize the system predictive performances but more precisely and according to a decreasing priority:
- to ensure the first requirement that the FN rate has to be very low since this kind of error may have dramatic consequences
- to extend as much as possible the flood anticipation time
- not to neglect the FP rate for the system relevance.

A simple approach could be to take natural variables that represent liminimetric levels observed upstream for predicting a high level on this given sensor downstream. And indeed this kind of predictive models may apparently perform well as we show in further sections. But they obviously cannot be considered as relevant and sound solutions since they are learnt on punctual data observed at a given time. We show in this paper how to define much more relevant variables that integrate levels observed on a time period. Thus we consider variables, called *complex variables*, that represent aggregate values on a time period rather than punctual values. These variables are defined by a set of parameters that give more flexibility allowing to consider different intervals of observation and prediction. The approach is extensible to other numeric records like rainfalls data or flow data.

In this work, we have trained available data that represent real levels recorded by sensors all along the river course. Unfortunately missing values were rather numerous due to very hard technical conditions. We have applied an optimization technique in order to select the best *complex variables* that represent aggregates of source raw values. A genetic algorithm has been employed to search for these best predictive variables among a wide space of potential solutions. The experimental results we obtained show good performances for selected variables .

The paper is organized in seven sections. Section 2 provides a quick overview of related works on feature selection. Section 3 describes source data, the data pre-processing method and a first exploratory analysis. Section 4 introduces the concept of complex variable and show how it is applied to the flood prediction context. Section 5 presents the genetic algorithm we used and all parameters involved. In Section 6 we discuss experimental results obtained and in Section 7 we conclude and present future works.

2. FEATURE SELECTION

Feature selection in data mining and machine learning has been widely studied for years (Fayyad *et al.*, 1996; Aha *et al.*, 1994; Chakrabarti *et al.*, 1998). In real life databases in which the dimension of tables may reach very high sizes in terms of variables, the selection of most representative features is mandatory for learning algorithms that are not able to afford so many variables. Techniques developed for eliciting the best subset of original features have proved to be efficient for

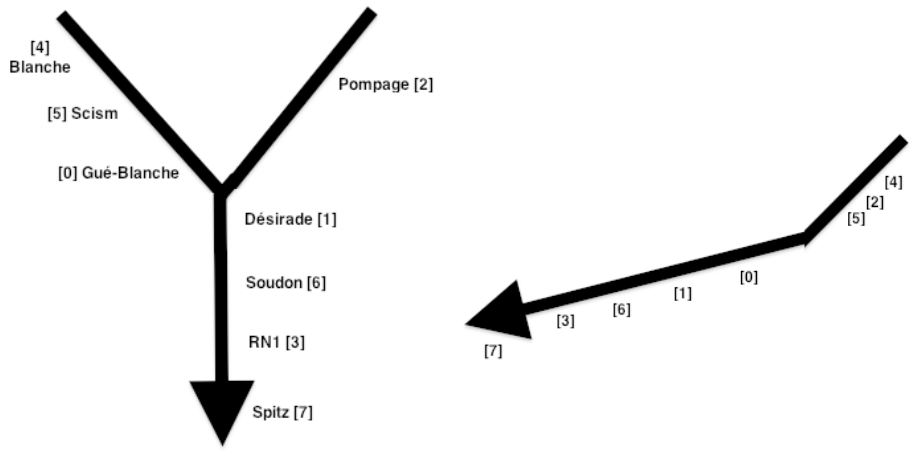
reducing the algorithm runtime and for improving resulting model performances. This kind of dimension reduction is generally processed in the context of supervised or unsupervised learning tasks for optimizing the learning process time and the model interpretability (Kudo *et al.*, 2000; Dy *et al.*, 2004; Devaney *et al.*, 1997; Chakrabarti *et al.*, 1998).
In supervised learning, the main issue for feature selection is to find a subset of features that produces higher classification accuracy. Feature selection in unsupervised learning is designed to find natural grouping of the examples in the feature space, that is a good subset of features that forms high quality of clusters for a given number of clusters.
A step further selection, so called *feature extraction* or feature preprocessing consists in building new variables on the basis of source data columns. The issue is generally to combine original variables in order to obtain better relevant features. For instance, the well known transformation of principal component analysis (PCA) performs a linear mapping of the data to a lower dimensional space.
Numerous search approaches have been proposed since the exhaustive evaluation of feature subsets is generally impractical because of the high computational complexity. Heuristic techniques like the branch and bound or greedy algorithms are popular while they do not always perform well. Among heuristic search methods, genetic algorithms have provided the best results for large datasets (Ralph, 2003).
For some kinds of datasets, such as transactional-data or time-series like the limnimetric data we study here, in which information about the same object may exist across several rows, the reduction may require to pre-process the data in order to *flatten* them before applying mining algorithms. In the following sections, we discuss the source data and the complex variables that can reduce the data dimension.

3. SOURCE DATA

As a frequent mandatory step in data mining tasks, source data have to be preprocessed. And indeed in this study, an entry in the source dataset represents an height level on a given sensor at a given time. No class label is included. These data have to be preprocessed in order to represent possible training and test sets for a predictive modelling algorithm. In this section, we show how data have been preprocessed and which first simple predictive models can be extracted from them.

3.1. *Source data*

The available source data we obtained, represent water height levels in millimeters recorded by sensors on several spots along the river course and its major tributaries from January 2006 to August 2010. Each measure on a given spot is differentiated by a timestamp as illustrated by Table 1 that shows a sample of data recorded by a given sensor. The default interval between two measures is six minutes. The second



(a) Top view (b) Side view

Figure 1. *Sensor locations on the river course*

column in Table 1 is the record time in minutes in the day. Nevertheless, due to hard technical conditions, sensors are not always working and thus data are missing. The whole volume of these raw data is ranging from 162000 to 338000 lines per sensor depending on the amount of missing values. Figure 1 illustrates schematically the sensors locations on the river course. The approximate average distance between two sensors is five kilometers.
In order to label data for supervised classification, we applied the predefined threshold (used in the current hydrologic system) on sensor measures to split examples in the *Flood* class (F) and the *Non-flood* class (N). In the following, we refer to a *flood event* on a sensor to indicate that the limnimetric level recorded reaches the threshold.

Table 1. *A sample of source data: Height levels in meters on a given sensor*

date	minutes	level(mm)
01/01/2006	990	320
01/01/2006	1002	321
01/01/2006	1008	320
	...	
31/08/2010	1020	267
31/08/2010	1026	300

As a preliminary data exploration, we wondered how would perform a simple predictive model only based on sensor(s) levels in order to predict the event "Flood" or "Non-flood" (F or N) on a sensor called the *event sensor*. We call *measure sensor* a

sensor used as a predictor.
While the last sensor *Spitz* located downstream the river bed should be the best candidate to be the event sensor, in this analysis, we chose the *Soudon* sensor (number 6) due to constraints on data. *Soudon* represents a better trade-off since recorded data on it are more numerous and more balanced among the two classes than data recorded on *Spitz* and is not far from the threatened area.

Table 2. *A sample of Dataset 1: Height levels on measure sensors recorded 60 minutes before with the Flood or Non-flood event on the event sensor Soudon*

0	1	2	3	4	5	6	7	class
207	689	373	836	702	687	578	990	F
204	542	355	544	266	321	297	442	N
				...				
371	716	367	1244	324	405	674	1968	F
93	482	355	464	194	227	734	618	F

3.2. *Data preprocessing*

We built several datasets structured as shown in Table 2, where each column except the last one, represents a measure sensor (different or not from the *event sensor*) and each line represents the *measure sensors* levels recorded M minutes before a *Flood* (F) or *Non-flood* (N) event observed on the *event sensor*. The *event sensor* may be included as a *measure sensor* since levels recorded on its spot before a flood event on it may be predictive too as shown by the example of Figure 2. As said earlier in this section, we mainly made experiments on the sensor *Soudon* which data are the most numerous and various. Indeed, in the current available data, we found 70 real threshold overflows on *Soudon* that we considered as *flood events*. *Flood events* are much more rare for other sensors. By "real threshold overflow", we mean that one flood and only one is said to occur as soon as the recorded level reaches the threshold and stays upper until it comes down.
For determining *Non-flood events* among under-threshold levels that are obviously more numerous in records (1 threshold overflow for 100 non overflow), data were collected randomly among under-threshold while observing a non significant time window around the event. We finally obtain 300 *Non-flood* examples.
Each dataset was determined by collecting measures among raw data and recorded M minutes before a *flood*/*Non-flood* event on the event sensor. The last column is the class label deduced from the event observed on the event sensor. The M minutes period is called the *prediction period*. Dataset 1 was built with a prediction period of 60 minutes for the event sensor *Soudon*.

To overcome the imbalance between classes in datasets, two common ways are :

1) either to assign distinct costs to class samples (usually higher costs for the minority class) (Pazzani *et al.*, 1994)

2) or re-sample the source dataset, either by over-sampling the minority class and/or under-sampling the majority class (Kubat *et al.*, 1997)

We followed the second approach to create balanced training sets. We over-sampled the *Flood* (F) class using the SMOTE algorithm (Chawla *et al.*, 2002) which consists in generating synthetic examples by randomly selecting points along the lines that join a minority class original sample and some of its nearest neighbors. We under-sampled the *Non-flood* (N) class by randomly picking-up such examples in the original data. Finally after applying SMOTE, we obtain samples that were equally balanced between the F and N classes. Models learned from these balanced data were tested on unbalanced original data in order to evaluate their performances in a more realistic situation.

3.3. *Exploratory analysis*

In order to get a first rough idea on which knowledge could be extracted from these data with simple methods, we applied different standard decision tree based algorithms such as C4.5 (Quinlan, 1993) , BF Tree (Shi, 2007) and FT (Gama, 2004). Figure 2 shows an example of decision tree obtained when using the C4.5 algorithm on Dataset 1. The second branch must be interpreted in this way: *if the water level on Soudon sensor is upper than 306 mm and the water level on Gue-blanche sensor is lower than 366 mm then the model predict Non-flood on Soudon 60 minutes later.*

Figure 2. *An example of decision tree learnt on Dataset 1 with the WEKA J48 algorithm*

```
soudon <= 306: N (192.0/13.0)
soudon > 306
|   gue_blanche <= 366: N (46.3/3.49)
|   gue_blanche > 366
|   |   soudon <= 674
|   |   |   pompage <= 440: N (52.58/23.29)
|   |   |   pompage > 440: F (77.82/25.13)
|   |   soudon > 674: F (232.3/24.77)
```

Table 3 shows average performances obtained from Dataset 1 and other similar datasets built on the same schema from raw data with different M values and for the event sensor *Soudon*. Columns give:
- the *prediction period* length M.
- the decision tree algorithm.

Table 3. *Average performances of decision tree models extracted from Dataset 1 and similar datasets*

Prediction period M	Algorithm	Test				
		FP	FN	TP	TN	Acc
60	C4.5	19,2	9,28	90,72	80,8	82,7
	FT	15,05	15	85	84,95	84,95
	BFT	14,8	10	90	85,2	86,1
120	C4.5	27,2	16,2	83,8	72,8	75
	FT	16,9	32,1	67,9	83,1	80,25
	BFT	26,2	15,2	84,8	73,8	75,7
180	C4.5	32,3	8,3	91,7	67,7	72,4
	FT	26,1	36,1	63,9	83,9	80
	BFT	30,4	7	93	69,6	74,1

- the FP, FN, True Positives (TP), True negatives (TN) and average accuracy rates obtained when testing models on different unbalanced original datasets.
Models were learnt by 10-fold cross validation with the three methods on SMOTE-extended balanced datasets. As we can observe, the best global accuracy rates are obtained with the shortest prediction period tested of 60 minutes. FP (1-TN) rates are increasing with M while FN (1-TP) ere very much varying according to M values and algorithms.
These results were obtained on few data with very unbalanced class distribution. We may obviously assume that they would generalize badly on new data since on one hand they only take into account punctual records and on the other hand the prediction period M is constant for any branch in the tree.

As stated in the introduction, our final objective is to model the collective contribution of different factors in flood occurrence. Thus we need to buid a flexible and extensible modelling method that should enable to integrate not only limnimetric punctual measures but other index measures considered on different time periods. This multi-objective challenge was the motivation for defining *complex variables* presented in the next section.

4. COMPLEX VARIABLES

The underlying idea is to test the predictive performance of features that would represent the river activity summarized on a time period by comparison to simple time measures. For instance, we may intuitively think that the water height level average and standard deviation computed over a given time period for a measure sensor before a flood event on an event sensor may be quite relevant. We adopted the same approach as in (Ralph, 2003) and we had to make an important pre-processing to adapt original

data. In this section, we define the notions of *structure* and *complex variable (CV)*, and then we show how they were applied in the flood prediction case.

4.1. *Definition*

A ***structure*** is a template for ***complex variables (CV)*** which values are aggregates. A **CV** is derived from a structure by instantiating its parameters.

A ***structure*** is defined by:

– an aggregate function (most of the time a statistical function)

– a set of aggregate attributes on which the aggregate function is applied

– a set of contextual conditions

– a *group-by* attribute

Let us note such a structure S as follows:

```
S=
<aggregation function>
<aggregate attribute>
<contextual conditions>
<group-by attribute>
```

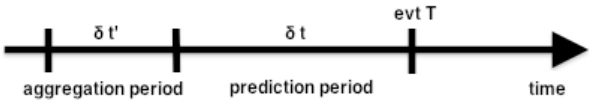
As an example applied to flood data, let us consider the structure S1 which models aggregates as standard deviation of measures for each measure sensor:

```
S1=
<standard deviation>
<sensor measure>
<cond1: aggregation period,
cond2: prediction period,
cond3: event sensor 6>
<measure sensor id>
```

S1 parameters are the *measure sensor* ***ms*** and the aggregation and prediction period lengths illustrated as on Figure 3. Figure 3 summarizes the principle applied to compute aggregate values according to the S1 structure: for a Flood or Non-flood event occurring at time T (evt T), an aggregate value is computed upon a period of $\delta t'$ minutes (*aggregation period*) before the beginning of the *prediction period* δt that ends itself with the event.

We could derive from S1, CVs that represent for instance:
(1) the standard deviation on **[60 minutes]** of measures recorded on the ***ms*** **[sensor**

Figure 3. *Aggregation computation procedure*



2] during the **[last 3 hours]** before the flood event occurring on the *event sensor* [sensor 6]
or
(2) the standard deviation on **[120 minutes]** of measures recorded on the ***ms*** **[sensor 5]** during **[90 minutes]** before a Non-flood event observed on the *event sensor* [sensor 6].

Thus a CV is defined by its structure and a tuple of parameters. For instance, CVs defined above in (1) and (2) are defined by S1 and the 3-tuple (ms, $\delta t'$, δt) where ms is the sensor measure, $\delta t'$ is the aggregation period length and δt is the prediction period length.
The search for best CVs that can be derived from a given structure is equivalent to a combinatorial search through the different possible values of its parameters. The main objective in this work, is **to search for best combinations of CVs derived from different structures**.

4.2. *Application*

In order to apply this approach to flood data, namely to obtain an efficient flood prediction from water levels, we have used structures similar to S1 from which derived CVs represent aggregates computed as shown on Figure 3. In this context, the only *aggregation attribute* to study is the water level recorded on a measure sensor, the *group-by attribute* is the measure sensor id and the *contextual conditions* are on the aggregation period, the prediction period and the event sensor. For instance, let us consider the S2 structure

```
S2=
<arithmetic mean>
<sensor measure>
<cond1: aggregation period,
cond2: prediction period,
cond3: event sensor 6>
<measure sensor id>
```

S1 and S2 represent the standard deviation and the arithmetic mean of sensor measures upon the aggregation period before the prediction period that ends with the event (F or N) occurring on the event sensor. Like S1 and S2, all the structures we have considered differ from one to each other only on the aggregation function and on the event sensor id. We have checked different aggregation functions like the Energetic Mean (EM), the Harmonic Mean (HM), the Arithmetic Mean (AM), the Standard Deviation (SD) and the Quadratic Mean (QM) as discussed in Section 6.

A complex variable derived from these structures is partly defined by the 3-tuple (*ms*, $\delta t'$, δt) where *ms* is the sensor measure, $\delta t'$ is the aggregation period length and δt is the prediction period length.
If we call ***independent CVs*** variables that refer to different measure sensors, the final objective is to find best combinations of independent variables for prediction. A CV has to be evaluated according to its efficiency to predict a class (Flood or Not-flood on a given event sensor). For this reason, values of CVs are discretized into bins and the evaluation is computed according to class distribution onto CV bins. In this experience, we have limited the discretization to two bins only. Thus the complete definition of a CV includes the bins limit.
In conclusion, in this application, a CVs is entirely defined by a given structure and a 4-tuple **(*ms*,** $\delta t'$**,** δt**,*B*)** where ***B*** is the bin limit in the variable value set.

Table 4 gives a sample of a resulting aggregate dataset obtained with values of different variables derived from the structure S2. In this dataset, columns give:

– the *measure sensor ms* on which the aggregation is computed

– the absolute and relative timestamp (date and min) ; these data are not directly used but allow to check if several lines refer to the same time

– the aggregation period length $\delta t'$

– the prediction period length δt

– the *event sensor es*

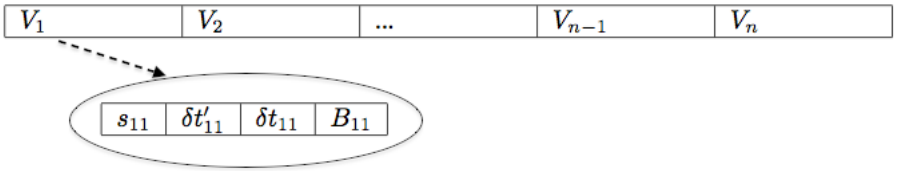
– the class (Flood or Non-flood) corresponding to the event sensor state

– the aggregate value A.

Table 4. *A sample of aggregate data as values of*

	ms	date	min	$\delta t'$	δt	es	C	A
	3	01/04/09	570	120	60	6	N	480
	3	17/01/09	543	120	60	6	N	651
	1	28/04/09	870	240	120	6	F	487
				...				
	2	08/01/09	810	240	180	6	N	462

Figure 4. *Complete chromosome sequence*



In the following sections 5 and 6, the only event sensor considered is *Soudon* because it is currently the best trade-off as explained in Section 3.

5. GENETIC ALGORITHM

As we have seen in the previous section, the search for best combinations of CVs derived from a given structure comes down to an optimizing search into the space of its parameters values combinations. However, when the number of parameters and the number of values for each parameter is increasing, the resulting combinatorial explosion does not allow to look exhaustively over all possible CVs in a reasonable time. For example, if there are only five parameters and each of them can take ten values, the number of CVs that can be derived reaches 100000.
Stochastic methods are thus indicated in this case. We decided to explore the solutions offered by Genetic Algorithms (GA) (Holland, 1992; Mitchell, 1998) that stochastic methods and use global search heuristics belonging to the family of evolutionnary algorithms. They are inspired by evolutionnary biology's main principles such as inheritance, mutation, selection and crossover. They have proved to be efficient when applied in a close context (Ralph, 2003) to solve combinatorial problems. In this section, we describe the specific GA we implemented with the EO library's functions (Keijzer *et al.*, 2001).

5.1. *Individual Encoding and Genetic Operators*

5.1.1. *Chromosome representation*

As seen in Section 4, a CV derived from a given structure can be encoded as the sequence (ms, $\delta t'$, δt, B) and evaluated according to its predictive power on a given sensor event. Since we are searching for different variables on different sensors, a solution (or chromosome) is defined as sequence of n independent CVs as illustrated on Figure 4. At this level, each CV codes a gene and CV parameters are sub-genes.

5.1.2. *Crossover*

The crossover principle consists in mating chromosomes (individuals) -the parents- in order to obtain new ones -the offsprings- made with their genetic heritage. The main purpose of this operator is to diversify an existing population in order to improve it.
In this work, the main operator we used is a quad crossover operator which consists in choosing two parents and computing two offsprings. This operator can be described as follows: when two individuals are selected for crossing, a user-defined number (generally n/2) of steps is accomplished. Each step takes place in the following way:

– First, in each individual, a gene (CV) is randomly chosen among its n genes.

– Then, two options are chosen with equal probability :

1) The two genes are simply switched

2) A randomly chosen number of their sub-genes (CVs parameters) are selected to be switched.

In these two cases, a control step is processed before validating the crossover in order to check the new offsprings *independence property*: new offsprings have to be sequences of independent CVs too.
Table 5 show an example of the application of this operator on two individuals. In this case, genes V_{11} and V_{21} have been entirely switched, whereas only the sub-genes $\delta t'$ and B that represent the aggregation period and the bin limit, have been exchanged between genes V_{12} and V_{22}.

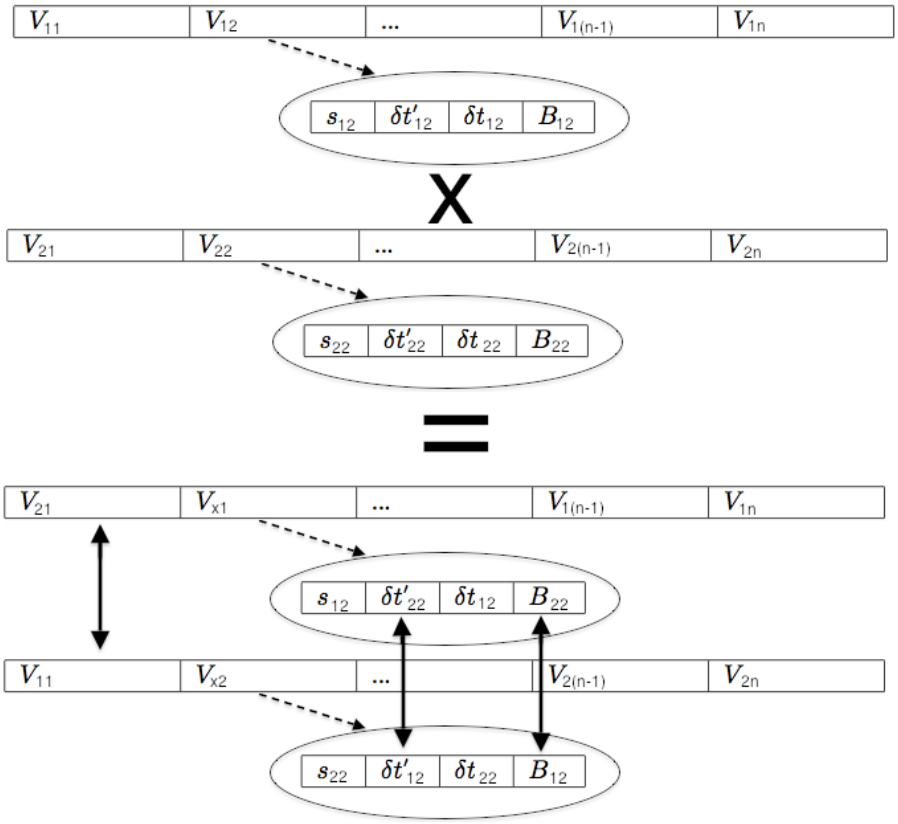
5.1.3. *Mutation*

The main purpose of the mutation operator is to prevent from the premature convergence of a population by allowing undirected jumps to slightly different areas of the search space. It is also useful in order to introduce a new genetic information that was not entered during the initialization step. Generally, mutations occurs randomly and very rarely with a probability under 0.01.
The mutation operator that we used consists simply in changing the value of a given sub-gene in an individual by either a randomly selected valid value or a close value. As for the crossover operator, the independence property is checked.

5.2. *Fitness Function*

The fitness function plays an essential role in a GA process as it assigns a score to each individual in the population in order to evaluate its quality regarding the problem to solve. Generally, the main challenge encountered while searching for an efficient fitness function is to elaborate a function which is not only computed from the individuals coding but is easily optimized too. For the particular problem of prediction, the fitness has to measure the predictive efficiency of individuals, that is their ability to separate classes. We have trained the three following functions. Let us

Figure 5. *Quad-crossover*



take:

– z as a given event sensor

– the complex variable CV=(w,x,y,L) for the event sensor z

– J, the set J of tuples available in the learning sample similar to Table 4 which schema is (*ms*,δt,$\delta t'$, *es, C, A*).

– $S = \{t \in J \mid t[ms] = w, t[\delta t'] = x, t[\delta t] = y, t[es] = z\}$

– I the set of bins defined for values of A

– for each i in I, $S_i = \{t \in S \mid t[A] \in i\}$

– $P_i(X)$ is the probability $P(t/t[C] = X \; and \; t \in S_i)$ for each class C

We have first considered two functions that tend to measure the class separability power induced by a variable and the discretization in bins selected for its values. F1 is directly dependent on the probability of each bin, so it has to be maximized. F2 is based on the entropy measure of each bin, so it has to be minimized.

$$F1(CV) = \frac{1}{|S|} \sum_{i \in I} \left| S_i \right| . \left| P_i(F) - P_i(N) \right| \quad [1]$$

$$F2(CV) = -\frac{1}{|S|} \sum_{i \in I} \left| S_i \right| . \Big[P_i(F) log_2(P_i(F)) + P_i(N) log_2(P_i(N)) \Big] \quad [2]$$

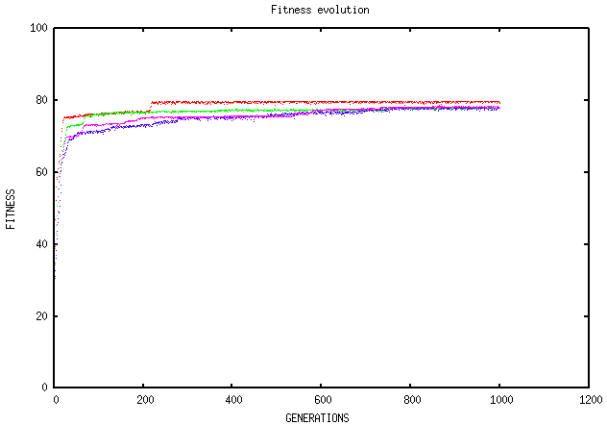
The fitness functions described above were implemented to evaluate the predictive power of a given CV. For individuals that are composed of several CVs, we define the fitness as the average fitness on its CVs.
For a combination of p CVs (CV_1, CV_2, ...CV_j...CV_p), we have also considered the fitness function (F3) defined as a simplification of the relative entropy (or Kullback divergence) that measures the difference between distributions of classes. It has to be maximized.

$$\text{F3}((CV_1, CV_2, ...CV_j...CV_p)) = \frac{(\mu(P_{ji}(F)) - \mu(P_{ji}(N))^2}{\frac{1}{2}\sigma(P_{ji}(F))^2 + \sigma(P_{ji}(N))^2} \quad [3]$$

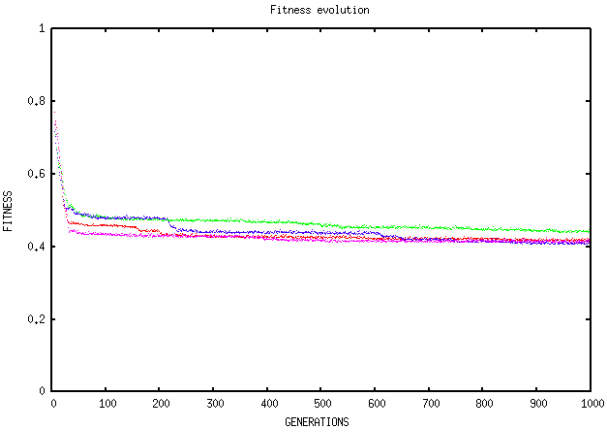
where $P_{ji}(X)$ is the probability $P_i(X)$ for CV_j.

5.2.1. *Fitness comparison*

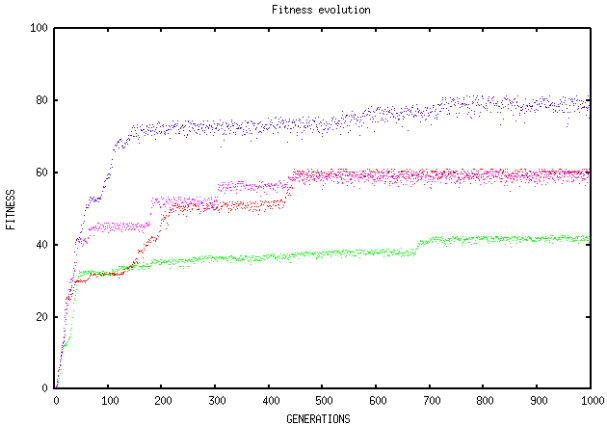
In this section we compare the behaviour of these three functions. Figures 6(a), 6(b) and 6(c) show, for given AG parameters and structure, each corresponding fitness evolution along generation. We have plotted the average fitness computed over the population at each generation for several experiments. We can observe that for F1 and F2, the values are rapidly close and converge approximately at the same time, after nearly 50 generations. We may think that this premature convergence is related to the relatively small size of the search space due to the fixed parameters, however the important point to note in this section is that they behave similarly. On F3 curves, we can see that the convergence occurs later, approximately between 100 and 500 generations, and that the values are varying from a run to another. Figures 7(a), 7(b) and 7(c) show the evolution of the fitness standard deviation into the population for the same experiments. We can check that the deviation is quite weak into populations generated with F1 and F2 and much more important into populations generated by F3.



(a) F1

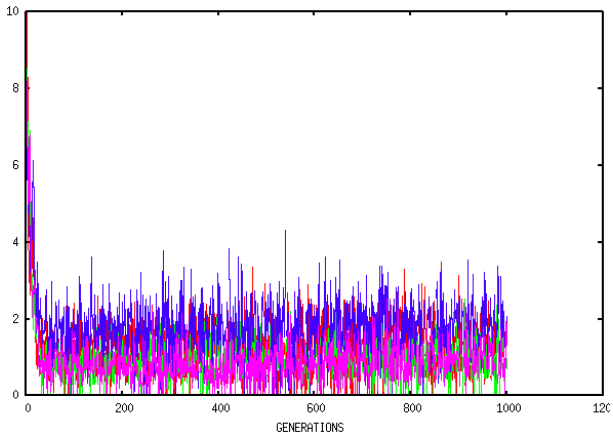


(b) F2

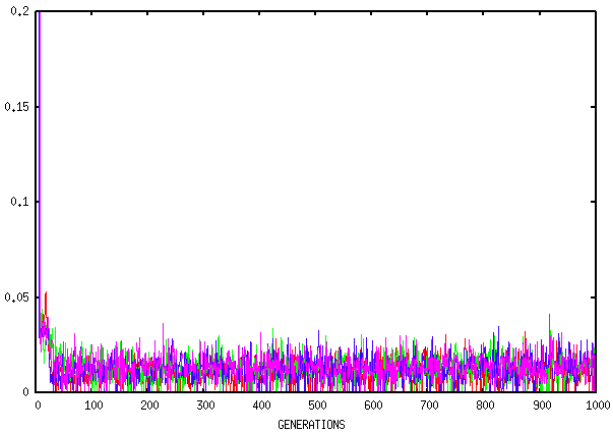


(c) F3

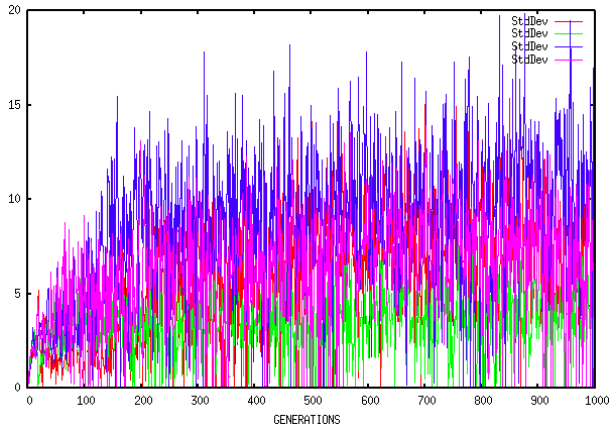
Figure 6. *Evolution of each fitness function over generations*



(a) F1



(b) F2



(c) F3

Figure 7. *Evolution of standard deviation of the population for each fitness function over generations*

6. Experimental results

In this section, we present results obtained with the GA for searching best solutions (chromosomes) as combinations of independent CVs. We checked various GA parameters (number of generations, operators probability, selection scheme, population size...). The best results presented here were obtained with the following parameters:

– 2000 generations
– a crossover probability of 0,6
– a mutation probability of 0,1
– tournament selection
– 100 individuals population

We trained different aggregation functions such as Energetic Mean (EM), Harmonic Mean (HM), Arithmetic Mean (AM), Standard Deviation (SD), Quadratic Mean (QM). In order to take into account the temporal evolution of the river, we linearly weighted these functions so that latest limnimetric levels recorded were more significant.

6.1. *Selection of best CVs combinations*

First, we have considered different CV structures separately and we have run the GA in order to obtain the best sets of variables regarding each structure. Then, among the best CV sets obtained, we picked with an exhaustive method, the best variables related to different structures in order to design new combinations composed of 3, 4, 5 or 6 CVs related to different structures. The exhaustive approach was made practicable thanks to the low number of potential structures and the size of the individuals. The *independence property* was checked in this step too and another constraint was checked in order to avoid more than 2 occurrences of the same measure sensor in a combination.

We have compared results obtained for combinations of different sizes (number of CVs). Tables 5, 6 and 7 give the composition of best individuals selected for each size. In these tables, the *ms* column indicates the sensor measure, the *Agg* column the aggregation function, $\delta t'$, δt and B columns represent the aggregation period, the prediction period and the bin limit for each CV. The *Avg Fitness* column gives the average fitness for the combination.
We can observe that the combination sizes 3 and 4 provide the best fitness. Whatever the size be, we can note that :

Table 5. *Composition and size comparison of best individuals obtained with F1*

Size	Best chromosome					Avg Fitness
	ms	Agg	$\delta t'$	δt	B	
3	6	HM	120	60	740	82,3
	6	EM	180	60	1333	
	5	EM	120	60	451	
4	6	HM	120	60	740	81,6
	1	SD	240	60	13	
	5	EM	120	60	451	
	6	EM	180	60	1333	
5	5	QM	60	120	455	80,9
	6	HM	120	60	740	
	6	EM	180	60	1333	
	5	EM	120	60	451	
	1	SD	240	60	13	
6	5	QM	60	120	455	79.8
	6	HM	120	60	740	
	6	EM	180	60	1333	
	5	EM	120	60	451	
	0	EM	240	60	658	
	1	SD	240	60	13	

– the energetic mean:

$$\bar{x} = 10.log_{10}(\sum_{i=1}^{n} 10^{x_i/10}) \quad [4]$$

often appearing in best individuals, suggesting that its "summarizing power" performs better than other functions, in this context.

– the best prediction periods are frequently 60 minutes, so that as expected, the short-term prediction is often better.<w

– the best aggregation periods are often 120 or 180 minutes.

– F1 and F2 provide very similar combinations either on aggregation functions or on measure sensors while results with F3 are slighly different with more variety in prediction periods δt.

– the measure sensor 6 *Soudon* and event sensor too, appears to be much predictive.

To overcome the problem of short prediction periods, we checked a constraint so that a given prediction period cannot appears more than once for a given measure sensor. Table 8, similar as Tables 5, 6 and 7, shows the new individuals obtained with

Table 6. *Composition and size comparison of best individuals obtained with F2*

Size	Best chromosome					Avg Fitness
	ms	Agg	$\delta t'$	δt	B	
3	6	HM	120	60	740	0,446
	6	EM	180	60	1333	
	5	EM	120	60	448	
4	6	HM	120	60	740	0,464
	0	HM	240	60	756	
	5	EM	120	60	448	
	6	EM	180	60	1333	
5	5	QM	60	120	455	0,481
	6	HM	120	60	740	
	6	EM	180	60	1333	
	5	EM	120	60	448	
	0	HM	240	60	756	
6	5	QM	60	120	455	0,49
	6	HM	120	60	740	
	6	EM	180	60	1333	
	5	EM	120	60	448	
	0	EM	240	60	658	
	0	HM	240	60	756	

more various δt values in the best combinations found with F1. As we see, the average fitness values stay very close to values obtained without the constraint in Table 5.

6.2. *Test of best combinations*

In order to test the performances of the best GA individuals ie combinations of CVs, we built *aggregate* datasets as illustrated by the sample of Table 9. In this table, each line represents a n-tuple (A_1, A_2,..., A_n, class) with n $\leq$ 6 where each A_i is the aggregate value of the *i-th* CV of a best individual and the class attribute refers to an event that occurred a while later on the event sensor *Soudon*.
A VERIFIER A dataset was generated for each best individual obtained with F1 and F3 as they appear on Tables 5, 7 and 8. Since F1 and F2 behaves similarly, we only keep F1.
In these datasets, each value is computed with its own parameters, such that two values of a same line may have been computed on different periods before the same event. Since the initial data contain missing values (for periods ranging from a few minutes to several months) these datasets also contain missing values.
Among the best combinations of Table 5, we have generated datasets only for size 3

Table 7. *Composition and size comparison of best individuals obtained with F3*

Size	Best chromosome				
	ms	Agg	$\delta t'$	δt	B
3	6	EM	120	60	1765
	0	HM	180	120	1011
	1	EM	60	60	678
4	6	EM	120	60	1765
	0	HM	180	120	1011
	1	EM	60	60	678
	1	SD	180	120	11
5	6	EM	120	60	1765
	0	HM	180	120	1011
	1	EM	60	60	678
	1	SD	180	120	11
	0	QM	120	240	763
6	6	EM	120	60	1765
	0	HM	180	120	1011
	1	EM	60	60	678
	1	SD	180	120	11
	0	QM	120	240	763
	6	EM	60	180	1751

and 4 in order to test the model performances for a fixed prediction period. Indeed these combinations contain CVs which $\delta t'$ is always 60 minutes. We have tested these same combinations on the two prediction periods of 60 minutes and 120 minutes. Results are presented in Tables 10 and 11.
We have applied on these datasets, the same tree based algorithms as in Section 3 on simple data. Thus predictive models learnt looks like this one:
PLACER ICI L ARBRE ET VERIFIER QUE LA PHRASE CI DESSOUS EST CORRECTE
We can interpret branches like the XXXX one as follows:
If the standard deviation of sensor 5 levels during 120 minutes overtakes its threshold and if the energetic mean of sensor Gue-blanche levels during 60 minutes overtakes its threshold and ...
Then the flood may occur on the event sensor Soudon with the probability p.

Tables 10 and 11 present performances obtained for prediction periods of 60 and 120 minutes as explained just above. These tables gather results obtained on simple data previously presented in Section 3 and on aggregate data. The first column *Method* indicates if the decision tree was learnt and tested on simple data or aggregate data. The second column indicates the learning algorithm applied.

Table 8. *Composition and size comparison of best individuals with prediction period constraint and F1*

Size	Best chromosome					Avg Fitness
	ms	Agg	$\delta t'$	δt	B	
3	6	EM	60	60	1118	81,5
	0	EM	120	60	619	
	5	QM	120	180	442	
4	6	EM	60	60	1118	80,4
	0	EM	120	60	619	
	6	HM	60	120	764	
	5	QM	120	180	442	
5	6	EM	60	60	1118	79,1
	0	EM	120	60	619	
	6	HM	60	120	764	
	5	QM	120	180	442	
	0	HM	240	120	616	
6	6	EM	60	60	1118	78,4
	0	EM	120	60	619	
	6	HM	60	120	764	
	5	QM	120	180	442	
	0	HM	240	120	616	
	3	QM	240	60	740	

Table 12 presents the performance rates obtained for mixed prediction period with the individuals presented in table 8 and F1. The column *Size* indicates the individual size. TABLE A INSERER SUR F3 Table **??** has the same composition as Table 12 and presents the performance rates obtained with the individuals presented in table 7 obtained with F3.
In the four tables, the column Algorithm indicates the decision tree technique and columns FP, FN, TP, TN and Acc present the performance rates of the tests on unbalanced data.

The results presented in Table 10 and Table 11 confirm the conjecture that aggregate variables perform well either on global accuracy or on TP or TN rates. Indeed for the 60 minutes prediction period with aggregate variables, the average accuracies (84,7%, 85,1%, 88,3%, 87,2%) and TN rates (85%, 86,9%, 89,9%) are quite good, but slightly improve results on simple variables. For the 120 minutes prediction period, while the global accuracies are not so good, we can observe the best FN rate of 6% obtained with C4.5.
But results of Table 12 are much more encouraging since they show that longer prediction periods and large CV size may be good predictors. Indeed, we can observe that the best individual of size 6 of Table 5 that is defined with prediction periods of

120 and 180 minutes and aggregation periods of 120 and 240 minutes perform very well since it provides the lowest FN rate of 9,9% with a correct FP rate of 14,4%.
As explained in Section **??**, TN rates minimization is a concern of first priority in this prediction problem. But the optimization of the prediction period is quite important too to provide better anticipation margin on the ground. According to these requirements, the *complex variable* approach gives encouraging results that confirm the a priori idea that summarizing the limnimetric activity on periods may provide sound predictions. Of course, these tests have to be extended to new and more numerous data we hope to obtain soon.

Table 9. *Example of Aggregate Data according to different CVs*

CV_1	CV_2	...	CV_n	Class
381.81	382.20	...	505.13	N
?	1520.71	...	392.77	N
409.24	?	...	469.90	N
?	1784.25	...	741.68	F
...	...	...	...	...
2455.80	2859.35	...	2660.03	F

Table 10. *Performance of decision trees for a 60 minutes prediction period*

Method	Algorithm	Test				
		FP	FN	TP	TN	Acc
Simple	C4.5	19,2	9,28	90,72	80,8	82,7
	FT	15,05	15	85	84,95	84,95
	BFT	14,8	10	90	85,2	86,1
Aggregate (Size 3)	C4.5	15	16,8	83,2	85	84.7
	FT	16.2	14,5	84,6	83,6	83,8
	BFT	14,6	15,7	84,3	85,4	85,1
Aggregate (Size 4)	C4.5	10,1	18,6	81,4	89,9	88,3
	FT	14,6	12,5	87,5	84,9	85,4
	BFT	12,8	11,4	88,6	86,9	87,2

7. CONCLUSIONS AND FUTURE WORK

In this paper, we have investigated the question of searching for complex variables that represent summarized values over time and evaluating their predictive performance for flood prediction. Since floods are complex phenomena due to multiple factors, the objective was to obtain an extensible and flexible solution that will enable to enter new types of data and new parameters easily.

Table 11. *Performance of decision trees for a 120 minutes prediction period*

Method	Algorithm	Test				
		FP	FN	TP	TN	Acc
Simple	C4.5	27,2	16,2	83,8	72,8	75
	FT	16,9	32,1	67,9	83,1	80,25
	BFT	26,2	15,2	84,8	73,8	75,7
Aggregate Size 4	C4.5	33,8	6	94	66,2	71,5
	FT	19,2	19,4	80,6	80,8	80,7
	BFT	22,5	14,5	85,5	77,5	79

Table 12. *Performance of decision trees for mixed prediction period on aggregate data*

Size	Algorithm	Test				
		FP	FN	TP	TN	Acc
3	C4.5	10,2	22,2	77,8	89,8	87,5
	FT	9,3	20,7	79,3	90,7	88,5
	BFT	17,9	10,5	89,5	82,1	83,5
4	C4.5	10,7	20,1	79,9	89.3	87,5
	FT	12,5	17,6	82,4	87,5	86,7
	BFT	15,3	11,3	88,7	84,7	85,4
5	C4.5	12,5	14,5	85,5	87,5	87,2
	FT	10,5	17,9	82,1	89,5	88
	BFT	12,3	12,4	87,6	87,7	87,7
6	C4.5	12,8	17,6	82,4	87,2	86,3
	FT	15,7	16,2	83,8	84,3	84,2
	BFT	14,4	9.9	90,1	85,6	86,5

In this work, we have assimilated a flood phenomenon to the occurrence of high water levels since our main objective was to demonstrate the advantage of the solution for prediction. Our approach was to follow a methodology in five steps:

– raw data were preprocessed in order to infer new datasets fitted to the selection of best complex variables,

– a genetic algorithm was designed in order to address the optimization problem in a large space of potential solutions,

– the best complex variables found were merged to provide best combinations that were considered as the new predictive variables,

– new datasets defined by complex variables were computed,

– predictive models learnt from these data proved to be efficient on the available data.

The results presented in this paper demonstrate the potential efficiency of the approach and open new perspectives for further works. This first attempt may be now extended to introduce informations on stream flow, rain flow and tides and may be other environmental factors in order to select more complete variables. The system has to be tested on new and more numerous data recorded recently. Further research axes will focus on other meta-heuristics that should be benchmarked with genetic algorithms and on multi-objective optimization techniques allowing to find best trade-offs among solutions.

8. ACKNOWLEDGEMENTS

We thank the General Council of *La Martinique* for its support and our special thanks are for Yves Sidibe, Bernard Naigre and Kléber Delbois.

9. References

Aha D. W., Bankert R. L., " A Comparative Evaluation of Sequential Feature Selection Algorithms", 1994.

Chakrabarti S., Dom B., Agrawal R., Raghavan P., " Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies", *The VLDB Journal*, vol. 7, n° 3, p. 163-178, 1998.

Chawla N. V., Bowyer K. W., Hall L. O., Kegelmeyer W. P., " SMOTE: Synthetic Minority Over-sampling Technique", *J. Artif. Intell. Res. (JAIR)*, vol. 16, p. 321-357, 2002.

Devaney M., Ram A., " Efficient Feature Selection in Conceptual Clustering", *ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, p. 92-97, 1997.

Dy J. G., Brodley C. E., " Feature Selection for Unsupervised Learning", *J. Mach. Learn. Res.*, vol. 5, p. 845-889, 2004.

Fayyad U. M., Piatetsky-Shapiro G., Smyth P., " Knowledge Discovery and Data Mining: Towards a Unifying Framework", *KDD*, p. 82-88, 1996.

Gama J., " Functional Trees", *Machine Learning*, vol. 55, n° 3, p. 219-250, 2004.

Holland J. H., *Adaptation in natural and artificial systems*, MIT Press, Cambridge, MA, USA, 1992.

Keijzer M., Merelo J. J., Romero G., Schoenauer M., " Evolving Objects: a general purpose evolutionary computation library", 2001.

Kubat M., Matwin S., " Addressing the Curse of Imbalanced Training Sets: One-Sided Selection", *in* D. H. Fisher (ed.), *ICML*, Morgan Kaufmann, p. 179-186, 1997.

Kudo M., Sklansky J., " Comparison of algorithms that select features for pattern classifiers", *Pattern Recognition*, vol. 33, n° 1, p. 25-41, 2000.

Mitchell M., " Handbook of Genetic Algorithms (L. D. Davis)", *Artificial Intelligence*, vol. 100, n° 1-2, p. 325-330, 1998.

Pazzani M. J., Merz C. J., Murphy P. M., Ali K., Hume T., Brunk C., " Reducing Misclassification Costs", *ICML*, p. 217-225, 1994.

Quinlan J. R., *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.

Ralph C., Data Spiders : Leveraging transactionnal data with Genetic algorithm, Technical report, Fair Isaac Corporation, 2003.

Shi H., Best-first decision tree learning, Technical report, University of Waikato, 2007.