Minimum Multiple Characterization of Biological Data using Partially Defined Boolean Formulas

Fabien Chhel
LERIA
University of Angers
France
chhel@info.univ-angers.fr

Adrien Goëffon
LERIA
University of Angers
France
goeffon@info.univ-angers.fr

Frédéric Lardeux
LERIA
University of Angers
France
lardeux@info.univ-angers.fr

Frédéric Saubion
LERIA
University of Angers
France
saubion@info.univ-angers.fr

ABSTRACT

In this paper, we adress a characterization problem coming from plant biology. We consider different groups of experiments, each corresponding to the indentification of a given bacteria with regards to a given set of characters for diagnosis purposes. We have to compute simultaneously a complete minimal set of characterization formulas for each group. We propose two different approaches, based on Boolean functions, that allow us to study the satisfiability and the underlying complexity of this problem.

Keywords

Boolean characterization, formula minimization, NP-hard, diagnostic tests.

1. INTRODUCTION

Knowledge acquisition as sets of Boolean vectors is an easy way to collect results from experiments in various application domains. In this paper we define the multiple characterization problem (MCP), where each Boolean variable represents the presence or the absence of the differents features in a diagnosis process on several groups of items. Once data have been collected, the experimenter wants to use its results for testing to wich group an incomming item belongs to. This membership test requires thus the exact characterization of each group and, for practical purpose, the number of features that are used must be as small as possible.

We focus here on bacterial strains of *Xanthomonas*, which is a genus of bacterias, many of which cause plant diseases. The name pathovar is a subdivision of the phytopathogenic bacterial species that corresponds to the strains causing the same symptoms on plant species or varieties of plant species. In particular, *Xanthomonas* are used in many studies because they include hundred of different pathovars. For example, the *Xanthmonas axonopodis* comes in pathovar *citri* that causes citrus canker but also pathovar *vesicatoria*, which is responsible for bacterial spot on pepper. However, the phylogeny of the strains is not sufficient to explain the host specificity, i.e. the plant species that are attacked by the strain. In particular, some genetically close pathovars can have some very different hosts and vice versa. The approach consists in identifying, among the directory of strains, the relevant genes (virulence genes) and in analyzing the correlation between the presence / absence of these genes and the host specificity of the pathovars (groups of bacterial strains) [8].

Recently, the description of 35 directories of virulence genes in a collection of 132 strains of *Xanthomonas* among 21 groups and with different host specificities showed a correlation between the virulence genes of a strain of *Xanthomonas* and its hosts. Within *Xanthomonas*, some pathovars are listed on the quarantine lists or even on lists concerning bioterrorism, and are thus subject to strict laws.

In this context, the characterization problem corresponds to the identification of a group of strains against other groups based on the presence or absence of particular genes. A strain is therefore a vector of binary values that reflects the presence (value 1) or absence (value 0) of these genes. More practically, a problem instance with 5 strains, divided into 3 groups based on a set of 4 genes can be illustrated by Fig. 1.

Solving this problem consists in characterizing each group. Therefore, for each group, we must find a combination of presence or absence of genes that is valid for all strains of group and not valid for all other strains of other groups. In the example in Fig. 1, group 1 is characterized by the simultaneous presence of genes x_1, x_2 and x_3.

There exists a real need to develop new approaches to provide characterization tools that take into account simultaneously several genes. In addition, biologists are interested in two specific properties of the solutions:

SAC'12 March 25-29, 2012, Riva del Garda, Italy.

Strain	Group	Genes			
		x1	x2	x3	x4
e1	g1	1	1	1	0
e2	g1	1	1	1	1
e3	g2	0	0	1	0
e4	g2	0	1	1	1
e5	g3	1	1	0	0

Figure 1: Example of instance

- A solution that minimizes the number of used characters: this is especially important for building diagnostic tests based on DNA chips [16]. The number of observed genes must be minimized for cost reasons, for avoiding long experiments and for insuring reliability. Another point is that it is easier to detect the presence of a gene rather than its absence.

- The computation of all solutions: it should be useful, in terms of biological interpretation, to have a representation of all possible solutions as it could highlight a special relationship between genes and explain some functional characteristics of the bacteria (for phenotypic considerations).

Once this problem has been settled, various methods can be of course considered. In particular, the DCC method (Diagnostic of Capacity Coefficient [3]), in which the genes are sorted by relevance thanks to a statistical study. The advantage of this approach is its computation simplicity but it deals with a single gene and it is difficult to generalize to the simultaneous identification of several genes (especially when dealing with complex combinations).

Of course our MCP is related to different exiting works or Boolean functions learning and minimization but nevertheless, several differences exists :

- **Automatic learning of Boolean functions**: This problem has been extensively studied for many years [14, 7, 10]. The initial works of Valiant [18] defines the probability approximately correct (PAC) learning model. On the one hand, in our context, we slightly differ from the PAC model since we want to obtain an exact characterization of the model. On the other hand, results on exact learning of Boolean functions [2] often relies on the class of function to which the function to be learnt belong. Here we cannot determine a priori the class of the hidden Boolean functions. Moreover, minimization of the learnt function is generally not handled.

- **Machine learning techniques**: Different machine learning techniques can be used to learn from examples, such as classifiers (e.g., Support Vector Machine [13]) but they do not always provide exact results and moreover they are often difficult to use in presence of very large sets of variables. Here, we do not consider the problem of generalization of the learnt function since we consider that the data set is exhaustive. Another possibility could be to use the FOIL system [15] that has been designed to learn Horn clauses from examples and then it may appear well-suited to our problem but the notion of minimality is not addressed by this system.

- **Minimization of Boolean expressions**: This problem is also important, for instance in electronic circuit design, and has deserved much works in order to provide efficient minimization techniques. Nevertheless, note that this problem is then slightly different from the main goal of electronic circuit design, where the purpose is to minimize the logical components of the circuit with respect to a fixed number of inputs and outputs [12].

We definitely have to insist on the fact that, apparent from the instances' size problem, we do need to provide exact and minimal (and not approximate learning based) characterizations using mutually partially defined Boolean function, which induce an extra level of algorithmic complexity. Informally, we consider a set $\mathcal{G}$ of n sets (called groups) g_i of entities. Each g_i must be satisfied by a Boolean formula, which is falsified by all the other entities that are in $\mathcal{G} \setminus g_i$. This can be interpreted as a multiple exclusive characterization of n groups of entities. Moreover, we require that the number of variables that is used in each formula is minimum.

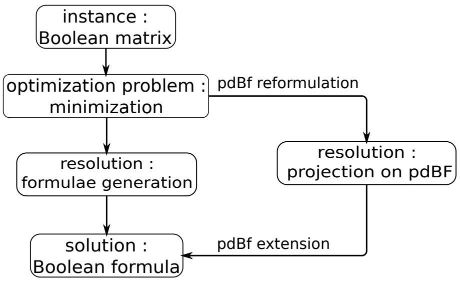

Figure 2: General Overview of the Proposed Approaches

We study here two ways of handling the problem that correspond to two possible formulations as illustrated by Figure 2. The initial problem is defined by a Boolean matrix. In the left part, we address the problem by means of Boolean formulas. We first study the satisfiability of the problem and its complexity. In the right part, the problem is translated as a problem of projection and related to approach of [11], where the problem of computing partially defined Boolean functions (pdBf) is stated and fully studied. This provides us another way to compute more compact solutions. The transformation between the two approaches is polynomial. In the first approach, solutions are expressed as propositional logic formulas, which induce several difficulties concerning their size. In particular, it might be difficult to control redundancies and tautologies when searching for solutions. Moreover, the search space might be infinite without defining some reduction rules to have normal forms. Therefore, the second formalism that uses projections provides a more compact representation of the solutions. Nevertheless, our first modeling allows us to clearly study the satisfiability of the problem and to provide a canonical solution.

In the last part of the paper, we show how this problem is applied to plant biology with some experimental results. These experiments have been conducted with biologists and have led to the development of a patented diagnosis kit for the identification of bacterial diseases.

The remaining of this paper is organized as follows. In section 2, we state the Multiple Characterization Problem. Two resolution approaches are then presented in section 3. In section 4, we study the complexity of the problem. Application to biological data is then described in section 5 and experimental results are provided.

2. THE MULTIPLE CHARACTERIZATION PROBLEM

The description of the multiple characterization problem (MCP) by means of presence or absence of several characters, has led us to naturally use a propositional logic formalism. We consider an instance $\mathcal{I}$ as a Boolean matrix corresponding to n characters and m entities:

$$\mathcal{I} \equiv \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \vdots & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix}$$

Each row of this matrix represents an entity, characterized by the presence or absence of a set of characters. We consider then the characters as Boolean variables and the entities as Boolean assignments.

For every column index $j \in \{1, \dots, n\}$, we define a propositional variable x_i which corresponds to a character. $\mathcal{X}$ is the initial set of propositional variables. For each row $i \in \{1, \dots, m\}$, we consider an entity e_i as the corresponding Boolean interpretation, i.e. a mapping from $\mathcal{X}$ to $\{0, 1\}$ (false, true), such that $\forall i, j, e_i(x_j) = a_{ij}$. We denote $\mathcal{E}$ the set of all entities. Given a propositional formula ϕ on $\mathcal{X}$ and $e \in \mathcal{E}$, we denote $e \models \phi$ the fact that the interpretation e satisfies the formula.

Note, in real cases, that preprocessing techniques can be applied to reduce the size of the original matrix. Therefore the set of variables that will be used for characterization may be smaller than the original set. In particular, if there is a character j such that $\forall i, k \in \{1, \dots, m\}, a_{ij} = a_{kj}$, then the corresponding column j can be removed from the matrix since this character cannot obviously be used to distinguish between entities.

The definition of the groups corresponds to sets of rows of the matrix $\mathcal{X}$. If there are two identical lines belonging to the same group, we may remove one of them. In this case, each group is then a subset of entities of $\mathcal{E}$. Note that at this time, two identical entities may belong to two different groups. We will study this aspect latter with regards to the satisfiability of the problem.

As usual, a literal is a variable $x \in \mathcal{X}$ or its negation, denoted $\neg x$. A clause is a disjunction of literals. We note $\mathcal{L}$ the set of literals built on $\mathcal{X}$. A formula ϕ is said to be in conjunctive normal form (CNF) if it is a conjunction of clauses. A formula is said to be in disjunctive normal form (DNF) if it is a disjunction of conjunctions of literals.

We can now define an instance $\mathcal{I}$ of a MCP.

Definition 1. **Instance of MCP**
An instance of a multiple characterization problem is defined by a tuple $\mathcal{I} \equiv (\mathcal{X}, \mathcal{E}, \mathcal{G})$ where $\mathcal{X}$ is a set of propositional variables, $\mathcal{E}$ is a set of entities defined over $\mathcal{X}$ and $\mathcal{G} \subseteq 2^{\mathcal{E}}$.

We now focus on defining the characterization of a group that should allow to recognize its own entities and to discriminate (i.e., not to accept) the entities of other groups (from a logical point of view it will thus correspond to the satisfaction or refutation of formulas).

Definition 2. **Group Characterization**
Given an instance $(\mathcal{X}, \mathcal{E}, \mathcal{G})$, a formula ϕ_g is said to characterize group $g \in \mathcal{G}$ iff:

$\forall e \in g, e \models \phi_g$ (accepts of the group's entities)
and
$\forall g' \in \mathcal{G} \setminus \{g\}, \forall e' \in g', e' \not\models \phi_g$ (discriminates other groups' entities).

By extension, we denote $g \models \phi_g$ the fact that ϕ_g characterizes g according to the previous definition. $Sol(g)$ represents the set of all the characterizations for a group g. $Sol(g) = \{\phi_g | g \models \phi_g\}$

Definition 3. **Solution of a MCP**
Given an instance $\mathcal{I} \equiv (\mathcal{X}, \mathcal{E}, \mathcal{G})$, an admissible solution of a multiple characterization problem is a $|\mathcal{G}|$-tuple of formulas $\Phi = (\phi_1, \cdots, \phi_{\mathcal{G}})$ such that $\forall i \in 1..|G|, g_i \in \mathcal{G}$, $g_i \models \phi_i$.

$SOL(\mathcal{I})$ is the set of all multiple characterizations for all groups. $SOL(\mathcal{I}) = Sol(g_1) \times \cdots \times Sol(g_{|\mathcal{G}|})$. Given a tuple of formulas $\Phi = (\phi_1, \cdots, \phi_{|\mathcal{G}|})$ and a set of groups $\mathcal{G}$, we denote by extension $\mathcal{G} \models \Phi$ the fact that $\forall i \in 1..|\mathcal{G}|, g_i \in \mathcal{G}$, $g_i \models \phi_i$.

Definition 4. **Satisfiability of a MCP**
An instance $(\mathcal{X}, \mathcal{E}, \mathcal{G})$ is satisfiable (resp. unsatisfiable) iff $\forall g \in \mathcal{G}, Sol(g) \neq \emptyset$ (resp. $\exists g \in \mathcal{G}, Sol(g) = \emptyset$).

We recall now some definitions on the size and length of a formula.

Definition 5. **Size and Length of Formulas**
Let ϕ be a formula and $var(\phi)$ the set of variable of ϕ.

- *The size of ϕ is $|var(\phi)|$.*
- *The length of ϕ, denote by $len(\phi)$, is the number of leaves of the term (formula) ϕ.*

Example 1. *Let $\phi = a \wedge b \wedge (\neg a \vee c)$, we have $|var(\phi)| = 3$ and $len(\phi) = 4$*

Now we define the size of a tuple of formulas.

Definition 6. **Size of a tuple of formulas**
For a tuple $\Phi = (\phi_1, \cdots, \phi_n)$ we have $|\Phi| = |\bigcup_{\phi_i} var(\phi_i)|$

Definition 7. **k-MCP (decision problem)**
Let an instance $\mathcal{I} \equiv (\mathcal{X}, \mathcal{E}, \mathcal{G})$ a minimal multiple characterization of size k is a set of formulas $\Phi \in Sol(\mathcal{I})$ and $k \in \mathbb{N}^+$ such that $|\Phi| \leq k$

The Minimum Multiple Characterization Problem for a size k (MIN-MCP-k) does not necessarily corresponds to a minimal solution $(\phi_1, \cdots, \phi_n)$ such that each ϕ_i is a minimal element of $Sol(g_i)$. Note that such a global minimal solution is a minimal element of $SOL(\mathcal{I})$, with regards to definition 6, whose computation is much more expensive.

Definition 8. **MIN-MCP (optimization problem)**
Let an instance $\mathcal{I} \equiv (\mathcal{X}, \mathcal{E}, \mathcal{G})$, a (global) minimal multiple characterization is a set of formulas $\Phi^* \in Sol(\mathcal{I})$ such that $\forall \Phi \in Sol(\mathcal{I}), |\Phi^*| \leq |\Phi|$

When building such global solution, we will be interested in the characterization of one group against the other ones. We may thus define a restriction of the problem, which is thus just a MIN-CP problem. for a group g

Definition 9. **MIN-CP(g)**
Let an instance $\mathcal{I} \equiv (\mathcal{X}, \mathcal{E}, \mathcal{G})$, a minimal characterization is a formula $\phi^* \in Sol(g)$ such that $\forall \phi \in Sol(g), |\phi^*| \leq |\phi|$

When observing the results, we are mostly interested in the size of the solution of the MIN-MCP (resp. MIN-CP), which corresponds thus to the number of variables used in the corresponding set, which will be denoted Min-MCP-opt (resp. Min-CP-opt).

3. RESOLUTION APPROACHES

In this section, we focus on the resolution of MIN-MCP. We first show the limits of the naive approach using directly Boolean formulas. We propose afterward to encode the problem in a different way in order to overcome theses limits.

3.1 The Boolean formulas approach for MIN-MCP

Intuitively, the Boolean formulas approach could seems the most suitable to solve the MCP problem.

Our first goal is to study the satisfiability of an instance of a MCP.

Proposition 1. *Satisfiability of an Instance*
Let an instance $\mathcal{I} \equiv (\mathcal{X}, \mathcal{E}, \mathcal{G})$, we have:

$$SOL(\mathcal{I}) \neq \emptyset \text{ iff } \forall g \in \mathcal{G} \begin{cases} \forall e \in g, e \models \phi_g \\ and \\ \forall g' \in \mathcal{G} \setminus \{g\}, \forall e' \in g', e' \not\models \phi_g \end{cases}$$

Proof. We propose thus a canonical formula that will be a solution for any satisfiable problem. Then, we will use this canonical formula to exhibit a necessary and sufficient condition for the satisfiability.

For each entity e in g, we build the formula $\phi_e^+ \equiv \bigwedge_{x \in X} \delta(e, x)$ where $\delta : \mathcal{E} \times X \to L$ is a function such that $\delta(e, x) = \neg x$ if $e \models \neg x$ and $\delta(e, x) = x$ otherwise. We note $\phi_g^+ \equiv \bigvee_{e \in g} \phi_e^+$. We may remark that ϕ_g^+ is in DNF. Similarly, we define $\phi_e^- \equiv \bigvee_{x \in X} \neg\delta(e, x)$ and $\phi_g^- \equiv \bigwedge_{g' \in \mathcal{G} \setminus \{g\}} \bigwedge_{e' \in g'} \phi_{e'}^-$, which is a CNF formula. We denote $\phi_g \equiv \phi_g^+ \wedge \phi_g^-$.

The proof is rather simple and is based on the following corollaries and lemmas obtained from this formula.

Lemma 1. $\forall e, e' \in \mathcal{E}, e \models \bigwedge_{x \in X} \delta(e', x) \Leftrightarrow e = e'$

Corollary 1. *Given an instance $\mathcal{I} \equiv (\mathcal{X}, \mathcal{E}, \mathcal{G})$:*

$\exists g, g' \in \mathcal{G}, g \neq g', \exists e \in g, \exists e' \in g', e = e' \Leftrightarrow SOL(\mathcal{I}) = \emptyset$

Corollary 2. *Given a instance $\mathcal{I} \equiv (\mathcal{X}, \mathcal{E}, \mathcal{G})$:*

$(\mathcal{X}, \mathcal{E}, \mathcal{G})$ is satisfiable $\Leftrightarrow \forall g, g' \in \mathcal{G}, g \neq g', g \cap g' = \emptyset$

□

We can thus exhibit a canonical solution Φ_{Max}^P for any satisfiable problem $\mathcal{I} \equiv (\mathcal{X}, \mathcal{E}, \mathcal{G})$:

$$\Phi_{Max}^{\mathcal{I}} \equiv (\phi_{g_1}, \ldots, \phi_{g_{|\mathcal{G}|}})$$

This canonical solution is also a maximal solution.

Proposition 2. *Given an instance $\mathcal{I} \equiv (\mathcal{X}, \mathcal{E}, \mathcal{G})$, we have the following properties:*

1. $SOL(\mathcal{I}) \neq \emptyset \Leftrightarrow G \models \Phi_{Max}^{\mathcal{I}}$
2. $\forall \Phi \in SOL(\mathcal{I}), \Phi \leftrightarrow \Phi_{Max}^{\mathcal{I}}$
3. $\forall \Phi \in SOL(\mathcal{I}), |\Phi_{Max}^{\mathcal{I}}| \geq |\Phi|$

Given a MCP $\mathcal{I}$, we aim first at computing a minimal solution $(\phi_1, \cdots, \phi_n)$ such that each ϕ_i is a minimal element of $Sol(g_i)$. Note that a global minimal solution is a minimal element of $SOL(\mathcal{I})$, with regards to definition 6, whose computation is much more expensive. In order to guarantee the computation of a minimal characterization for each group, we have implemented a complete search algorithm (EXACT-FORM-CHAR) that aim at exploring the whole search space in order to build a minimal solution. As usual this kind of tree-based exploration will be faced to computational space and time limits. The EXACT-FORM-CHAR algorithm allows us to obtain the shortest formula (in CNF) for a given group. The computation of this formula is achieved by trying all possible formulas from the shortest ones (one variable) to the largest ones (all the variables). However, many formulas are equivalent but it is hard to avoid dealing with these equivalent formulas since the generation of CNF without redundancy is akin to the redundant clauses detection problem, which is coNP-complete [1, 5]. The commutativity of logic connectors constitutes one of the reasons of these equivalences. Therefore, we may greatly limit the number of considered formulas by ordering the clauses. We use a notion of patterns of clauses which corresponds to the possible distributions of the literals in the clauses.

Example 2. *The possible patterns for CNF using two occurrences of literals are:*

- $()_1 \wedge ()_1$ *: 2 clauses of 1 occurrence.*
- $()_2$ *: 1 clause of 2 occurrences.*

These patterns are then instantiated by literals.

Example 3. *Based on the patterns of the previous example, considering a formula using two occurrences of literals with 2 variables x and y, we can build the next CNF:*

- $()_1 \wedge ()_1$ *:* $(x) \wedge (y)$, $(\neg x) \wedge (y)$, $(x) \wedge (\neg y)$ *and* $(\neg x) \wedge (\neg y)$
- $()_2$ *:* $(x \vee y)$, $(\neg x \vee y)$, $(x \vee \neg y)$ *and* $(\neg x \vee \neg y)$

Similarly to the clauses, the literals are also ordered in each clause. During the instantiation of the patterns, a simplification mechanism checks if the literals do not appear in unit clauses. In this case, the CNF is equivalent to another shorter one. We distinguish two possible cases:

- subsumption: CNF is $x \wedge ... \wedge (x \vee w \vee ... \vee z)$ (same parity), it is equivalent to $x \wedge ....$
- CNF is $\neg x \wedge ... \wedge (x \vee w \vee ... \vee z)$ (opposite parity), it is equivalent to $\neg x \wedge ... \wedge (w \vee ... \vee z)$.

It is interesting to remark that the number of patterns is given by the partition function noted p. For an integer n, $p(n)$ return the number of distinct ways (independently of the order) to represent n as a sum of natural integers. An asymptotic approximation of this function was proposed by

Hardy and Ramanujan [9], which can be used to compute the number of formulas studied by EXACT-FORM-CHAR.

EXACT-FORM-CHAR (Fig. 3) examines formulas from the shortest ones to the largest ones. We can thus be sure that equivalent CNFs obtained by the simplification mechanism are already tested by the algorithm. Then, the first formula that characterizes a group is a shortest one.

```
Require: an instance $(\mathcal{X}, \mathcal{E}, \mathcal{G})$ and a group $g \in \mathcal{G}$
  for $s$ from 1 to $|X|$ do
    for all formula $\phi$ with a size $s$ do
      if $\phi$ characterizes $g$ then
        return $\phi$
      end if
    end for
  end for
```

Figure 3: The EXACT-FORM-CHAR algorithm

Nevertheless, using this first direct approach we have been faced to three dificulties.

- With respect to the propositional logic recursive definitions, we can build terms (formula) of infinite size but we can restrict to normal form like DNF or CNF to bound the size (grow exponentially in number of variables).
- Many formulas are equivalent but it is hard to avoid dealing with these equivalent formulas since the generation of CNF without redundancy is akin to the redundant clauses detection problem, which is coNP-complete [1, 5]. The commutativity of logic connectors constitutes one of the reasons of these equivalences. Therefore, we may greatly limit the number of considered formulas by ordering the clauses.
- The major problem is to reason globally on $\mathcal{I}$ to find $min(SOL(\mathcal{I}))$. Find formula for each group with a global constraint (minimizing the total number of variables) is very costly. We want a global mechanism to solve our problem.

Therefore, we have turned to another resolution process.

3.2 The PdBf Approach

We first recall several notations and concepts related to partially defined Boolean functions (see [11] for more details) and we show the equivalence between the two approaches.

Definition 10. **Boolean Function**
A Boolean function f is a mapping $f : \mathcal{B}^n \mapsto \mathcal{B}$ of arity n.

Definition 11. **Partially Defined Boolean Function**
A partially defined Boolean function (pdBf) pf is defined as a subset $\mathcal{C}^+ \cup \mathcal{C}^- \subseteq \mathcal{B}^n$, where $\forall e \in \mathcal{C}^+$ (resp. $\forall e' \in \mathcal{C}^-$), we have $pf(e) = \top$ (resp. $pf(e') = \bot$) and $\mathcal{C}^+$ (resp. $\mathcal{C}^-$) is called the set of positive examples (resp. negative example). A $pdBf_g$ is defined as a pdBf with $\mathcal{C}^+ = g$ and $\mathcal{C}^- = G \setminus \{g\}$

We extend the notion of consistency for propositional logic to the pdBfs.

Proposition 3. **Consistency and Inconsistency of a pdBf**
A pdBf is consistent (resp. inconsistent) iff $\mathcal{C}^+ \cap \mathcal{C}^- = \emptyset$ (resp. $\mathcal{C}^+ \cap \mathcal{C}^- \neq \emptyset$).

Definition 12. **pdBf Extension**
An extension of pdBf is a formula ϕ, such that ϕ is satisfied (resp. unsatisfied) by all elements in $\mathcal{C}^+$ (resp. $\mathcal{C}^-$)

In the above definition, we remark that the pdBf's arity is equal to its extension's size. In [4], the authors develop polynomial time techniques to find extensions for many defined classes. We introduce now the notion of projection, which is a key concept for computing compact representation of solutions.

Definition 13. **Projection**
A projection is a function $\pi : \mathcal{B}^n \mapsto \mathcal{B}^k$ where $k \leq n$.

k is the dimension of the projection. π_{pdBf_g} is the projection associated to $pdBf_g$ (as defined above). Then we note for short $g \models \pi$ (i.e., π_{pdBf_g} is consistent) the fact that we have:

$$\cup_{e \in \mathcal{C}^+} \pi_{pdBf_g}(e) \bigcap \cup_{e \in \mathcal{C}^-} \pi_{pdBf_g}(e) \neq \emptyset$$

In order to preserve the results of previous sections, we have to prove that the formulas and projections (on pdBf) approaches are equivalent.

Proposition 4. **Projectability and Satisfiability**
For instance $\mathcal{I} \equiv (\mathcal{X}, \mathcal{E}, \mathcal{G})$ is satisfiable iff $\mathcal{I}$ is projectable on $\mathcal{G}$, ie $\forall g \in \mathcal{G}, \exists \phi, g \models \phi \Leftrightarrow \exists \pi, g \models \pi$

Proof. Let $g \models \phi \Leftrightarrow \forall e \in g, e \models \phi_g \wedge \forall e' \notin g, e' \not\models \phi$
$L \Rightarrow R$: inductive proof on the length of ϕ

- Basic case: $len(\phi) = 1$
  Let $\phi = x_i$ (resp. $\phi = \neg x_i$) and if $g \models \phi$ the only consistences pdBfs with x_i is $\forall e \in g, e(x_i) = \top$ (resp. $e(x_i) = \bot$) and $\forall e' \in \mathcal{G} \setminus \{g\}, e'(x_i) = \bot$ (resp. $e(x_i) = \top$). It exists π such as $g \models \pi_{x_i}$.
- Inductive case: For n>1, $len(\phi) = n - 1$ and $g \models \phi \Rightarrow g \models \pi$
  We can always build $\phi' = \phi \otimes x_i$ with $\otimes \in \{\wedge, \vee\}$, such that $g \models \phi'$ (according to the basic case's principle). We have then $\exists \pi', g \models \pi'$.

$R \Leftarrow L$: from a projection on a $pdBf_g$, we can always find an extension. □

We propose a complete algorithm to find an optimal $min(SOL(\mathcal{I}))$. The EXACT-PROJ-CHAR algorithm (Figure 4) computes projections from the shortest ones to the largest ones and returns extensions of the first consistent projection.

4. BOUNDS AND COMPLEXITY OF MCP

Firstly, we may easily provide lower and upper bounds for the MCP. Lower bound (LB) corresponds to the minimum number of variables needed to characterize all the groups. If we know the minimum number of variables needed to characterize each group, it is obvious that there is no smaller number of variables allowing to characterize all the groups. Concerning the upper bound (UB), we can be sure that there exists a solution for all the groups with at most as many variables as variables needed to characterize each group.
LB :
$|min(SOL(\mathcal{I}))| \geq max(|min(Sol(g_1))|, \ldots, |min(Sol(g_n)|)$.

```
Require: an instance $(\mathcal{X}, \mathcal{E}, \mathcal{G})$ and a group $g \in \mathcal{G}$
  for $s$ from 1 to $|X|$ do
    for all projection $\pi$ with a dimension $s$ do
      for all $g \in \mathcal{G}$ do
        consistent $\leftarrow$ true
        if $\pi_{pdBf_g}$ is inconsistent then
          consistent $\leftarrow$ false
          break
        end if
      end for
      if consistent then
        return $(ext(\pi(pdBf_{g_1})), \cdots, ext(\pi(pdBf_{g_n})))$
      end if
    end for
  end for
```

Figure 4: The EXACT-PROJ-CHAR algorithm

UB :
$|min(SOL(\mathcal{I}))| \leq |(min(Sol(g_1)), \ldots, min(Sol(g_n)))|$.

With regard to complexity, the minimal characterization problem for a group cannot be solved in a reasonable time. Indeed in [17], the author establishes that the minimization of a CNF formula is $\Sigma_2^p - complete$ (the proof is based on DNF formula but the results can be applied to CNF). This provides us an inclusion class of complexity. Considering that the minimization of the MCP correspond to the covering set problem, well-known to be NP-complete [6] (finding the smallest subset is NP-hard), the minimal characterization of a group is at least as hard as NP-complete problems.

Therefore, the idea is transform well-known problem SET-COVER ([6]) to k-MCP using the projection mechanism. In order to improve understanding, the number of group is fixed at 2 for the proof.

Proof. k-MCP is in NP, because we can easily check a solution with a polynomial time algorithm.

We recall the definition of SET-COVERT : Let C a collection of subsets of a set S, an $k \in \mathbb{N}^+$. Does $\exists C' \subseteq C$ with $|C'| \leq k$,such that $\bigcup_{c \in C'} c = S$?

We reduce to our problem :

Let $\mathcal{I} \equiv (\mathcal{X}, \mathcal{E}, \mathcal{G})$ an instance of MCP.
Does $\exists \pi \in \Pi$, the set of projection, with $|\pi| \leq k$ such that $\cup_{e \in \mathcal{C}^+} \pi_{pdBf_g}(e) \bigcap \cup_{e \in \mathcal{C}^-} \pi_{pdBf_g}(e) \neq \emptyset$?

We assign each subset of $c \in C$ to a propositional variable in $\mathcal{X}$ and each element of S corresponds to a pair of two entities $(e, e') \in \mathcal{E}^2$ of two different group. If we have two groups we can distribute just a single entity for the first one and $|C|$ for the second one and so $|\mathcal{E}| = |C|+1$. We construct $\mathcal{I}$ as follow :

Consider the only entity of the first group, we ground e_1, such that $\forall i \in \mathcal{X}, e_1(x_i) = 1$.

And for each pair $p_{0 \leq i \leq |c_j|}$ belongs to a subset $c_{0 \leq j \leq |C|}$ we have to differentiate the two entities of p_i with x_j. Therefore $e_i(x_j) = 0$. We ground also each x_j, such that $e_i'(x_j) = 1$

Example 4. *The collection* $C = \{\{1,4\},\{2\},\{2,3\}\}$ *is transform to the matrix* $\left(\begin{array}{ccc} 1 & 1 & 1 \\ \hline 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{array}\right)$

For $|\mathcal{G}| > 2$, we apply the transformation by setting $|S| = \binom{|\mathcal{G}|}{2}$, therefore each group contains only single entities.
We conclude that k-MCP is NP-complete.

□

From this first result, we get the following result for the minimization problem.

Proposition 5. **NP-hardness of MIN-MCP** *Minimum Multiple Characterization Problem is NP-hard.*

5. APPLICATION ON BIOLOGICAL DATA

The Min-MCP is of great interest to many biologists. For instance, the accurate characterization of collections of bacterial strains is a major scientific challenge, since bacteria are indeed responsible of significant plant diseases and thus subjected to official control procedures (e.g., in Europe, Directive 2000/29/EC). The development of diagnosis tests is therefore an important issue in order to routinely identify strains of these species. In this context, the characterization problem corresponds to the identification of a group of strains against other groups, based on the presence or absence of some particular characters. A strain is therefore a vector of binary values that reflects the presence (value 1) or absence (value 0) of these characters. Furthermore, to reduce the cost of expensive biological tests, we are interested in minimizing solutions.

The biologists have provided us with the four following instances:

- **A**: 8 groups, 108 entities (from 2 to 54 by group), 155 characters.
- **B**: 4 groups, 112 entities (from 5 to 69 by group), 155 characters.
- **C**: 7 groups, 112 entities (from 2 to 40 by group), 155 characters.
- **D**: 21 groups, 132 entities (from 2 to 40 by group), 37 characters.

A, B, C and D are instances coming from the BioMérieux API based on biochemical properties for *Ralstonia* species and the instance D coming from INRA tests on virulent genes for *Xhantomonas* species.

In Table 1, we present the experimental results that we have obtained with our algorithm. To find Min-CP-opt for each group g we use an alternate version of EXACT-PROJ-CHAR where we test the consistency of the $pdBf_g$. The lower and upper bounds are computed with the formula described in section 4 and the last line corresponds to the Min-MCP-opt value for each problem. When no solution is found in less than one hour, we note "-". To the best our knowledge, EXACT-PROJ-CHAR is the only one algorithm with exhaustive search dealing with MIN-MCP.

For the instance C, no result was found for the groups 1 and 2. 7 variables were tested by the algorithm when it was stopped so we can say that the lower bound is at least 7. However, we expect exponential time to compute these solutions, due to the NP-hardness of MIN-MCP.

Results in Table 1 show that the two approaches, Min-MCP-opt and Min-CP-opt, are complementary. Indeed, for

Instances	A								B				C						
Characters	155								155				155						
Entities	108								113				112						
Groups	1	2	3	4	5	6	7	8	1	2	3	4	1	2	3	4	5	6	7
Entites by groups	21	5	3	54	9	8	7	2	31	69	8	5	38	15	5	6	2	40	6
Min-CP-opt	2	2	2	4	4	3	3	3	3	6	5	2	-	-	5	4	2	5	4
Lower Bound	4								6				7						
Upper Bound	13								8				-						
Min-MCP-opt	6								6				-						

Instances	D																				
Characters	37																				
Entities	132																				
Groups	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Entites by groups	5	10	5	2	5	8	6	5	5	5	6	10	6	14	8	4	4	5	7	7	5
Min-CP-opt	3	2	4	2	4	2	4	2	3	3	3	3	2	3	3	3	3	2	2	1	1
Lower Bound	4																				
Upper Bound	16																				
Min-MCP-opt	9																				

Table 1: Application on real data with EXACT-PROJ-CHAR.

industrial applications where the minimization of the variables number is extremely important, it may be ask to characterize only one group but also to find a characterization for all the groups. In the first case Min-CP-opt is recommended and in the second case, it is more interesting to find Min-MCP-opt.

6. CONCLUSION AND FUTURE WORKS

In this paper we have defined the multiple characterization problem and we have proposed two formalisms in order to study its properties and to compute minimal solutions. An application to biological data has been described in order to highlight the possible uses of this general knowledge acquisition problem. We also propose a complete method to find minimal global solutions despite the NP-hardness of this problem. From the users' point of view, the formalisms appear to be useful since they clearly provide complementary information (which characters are important and formulas to implement a diagnostic test). It would be also very useful to be able to compute several different optimal solutions, for instance by means of a set of generators. Biologists are also interested in discovering relationships between characters, which could be achieved by means of logical consequences. Another perspective is related to the fact that data may be subjected to noise or uncertainty and one should be able to revise characterizations easily according to new experimental results.

7. REFERENCES

[1] Y. Boufkhad and O. Roussel. Redundancy in random SAT formulas. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 273–278. AAAI Press / The MIT Press, 2000.

[2] N. H. Bshouty. Exact learning via the monotone theory. In *34th Annual Symposium on Foundations of Computer Science*, pages 302–311. IEEE, 1993.

[3] P. Descamps and M. Véron. Une méthode de choix des caractères d'identification basée sur le théorème de bayes et la mesure de l'information. *Ann. Microbiol. (Paris)*, 132B, 1981.

[4] T. Eiter, T. Ibaraki, and K. Makino. Decision lists and related boolean functions. *Theor. Comput. Sci.*, 270:493–524, January 2002.

[5] O. Fourdrinoy, E. Grégoire, B. Mazure, and L. Sais. Eliminating redundant clauses in SAT instances. In P. V. Hentenryck and L. A. Wolsey, editors, *The Fourth International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pages 71–83, Brussels, Belgium, may 2007. Springer. lncs 4510.

[6] M. R. Garey and D. S. Johnson. *Computers and Intractability : a guide to the theory of NP-Completeness*. A series of books un the mathematical sciences. Freeman, New York, 1985.

[7] R. Gavaldà and D. Thérien. An algebraic perspective on boolean function learning. In *Algorithmic Learning Theory, 20th International Conference*, volume 5809 of *Lecture Notes in Computer Science*, pages 201–215. Springer, 2009.

[8] A. Hajri, C. Brin, G. Hunault, F. Lardeux, C. Lemaire, C. Manceau, T. Boureau, and S. Poussier. A "repertoire for repertoire" hypothesis: Repertoires of type three effectors are candidate determinants of host specificity in xanthomonas. *PLoS ONE*, 4(8):e6632, 08 2009.

[9] G. H. Hardy and Ramanujan. Asymptotic formulae in combinatory analysis. *Proceedings of the London Mathematical Society*, 2,17(17):75–115, 1918.

[10] L. Hellerstein and R. A. Servedio. On pac learning algorithms for rich boolean function classes. *Theor. Comput. Sci.*, 384(1):66–76, 2007.

[11] K. Makino, K. ichi Hatanaka, and T. Ibaraki. Horn extensions of a partially defined boolean function. *SIAM J. Comput.*, 28(6):2168–2186, 1999.

[12] E. McCluskey. Minimization of boolean functions. *Bell System Tech. J*, 35:1417–1444, 1956.

[13] D. Meyer, F. Leisch, and K. Hornik. The support vector machine under test. *Neurocomputing*, 55(1-2):169–186, 2003.

[14] B. K. Natarajan. On learning boolean functions. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, pages 296–304. ACM, 1987.

[15] J. R. Quinlan. Learning logical definitions from relations. *Machine Learning*, 5:239–266, 1990.

[16] M. Schena, D. Shalon, R. Davis, and P. Brown. Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, 270(5235):467–470., 2005.

[17] C. Umans. The minimum equivalent DNF problem and shortest implicants. *J. Comput. Syst. Sci.*, 63(4):597–611, 2001.

[18] L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, 1984.