Extraction de motifs n-aires utilisant la PPC

Mehdi Khiari, Patrice Boizumault, Bruno Crémilleux

GREYC (CNRS - UMR 6072) – Université de Caen
Boulevard du Maréchal Juin
14000 Caen
{Prénom.Nom}@info.unicaen.fr

Résumé

Dans cet article, nous proposons une approche PPC permettant d'extraire des motifs n-aires (i.e. combinant plusieurs motifs locaux) en fouille de données. Dans un premier temps, l'utilisateur modélise sa requête à l'aide de contraintes portant sur plusieurs motifs locaux. Puis, un solveur de contraintes génère l'ensemble correct et complet des solutions. Notre approche permet de modéliser de manière flexible des ensembles de contraintes portant sur plusieurs motifs locaux et ainsi de découvrir des motifs plus synthétiques et ainsi plus recherchés par l'utilisateur. A notre connaissance, il s'agit de la première approche générique pour traiter ce problème. Les expérimentations menées montrent la pertinence et la faisabilité de l'approche proposée.

1 Introduction

L'Extraction de Connaissances dans les Bases de Données (ECBD) a pour objectif la découverte d'informations utiles et pertinentes répondant aux intérêts de l'utilisateur. L'extraction de motifs sous contraintes est un cadre proposant des approches et des méthodes génériques pour la découverte de motifs locaux [2]. Mais, ces méthodes ne prennent pas en considération le fait que l'intérêt d'un motif dépend souvent d'autres motifs et que les motifs les plus recherchés par l'utilisateur (cf. section 2.2) sont fréquemment noyés parmi une information volumineuse et redondante. C'est pourquoi la transformation des collections de motifs locaux en modèles globaux tels que les classifieurs ou le clustering [13] est une voie active de recherche et la découverte de motifs sous contraintes portant sur des combinaisons de motifs locaux est un problème majeur. Dans la suite, ces contraintes sont appelées *contraintes n-aires*, et les motifs concernés, *motifs n-aires*.

Peu de travaux concernant l'extraction de motifs n-aires ont été menés et les méthodes développées sont toutes ad hoc [20]. La difficulté de la tâche explique l'absence de méthodes génériques : en effet, si l'extraction de motifs locaux nécessite déjà le parcours d'un espace de recherche très conséquent, celui-ci est encore plus grand pour l'extraction de motifs n-aires (le passage de un à plusieurs motifs augmente fortement la combinatoire). Ce manque de généricité est un frein à la découverte de motifs pertinents et intéressants car chaque contrainte n-aire entraîne la conception et le développement d'une méthode ad hoc.

Dans cet article, nous proposons une approche générique pour modéliser et extraire des motifs n-aires grâce à la Programmation par Contraintes (PPC). Notre approche procède en deux étapes. Tout d'abord, l'utilisateur modélise sa requête à l'aide de contraintes portant sur plusieurs motifs locaux. Ces contraintes traduisent des propriétés ensemblistes sur les items et les motifs (inclusion, appartenance, ...) ou des propriétés numériques sur les fréquences et tailles de ces motifs. Puis, un solveur de contraintes génère l'ensemble correct et complet des solutions. Un grand avantage de cette approche est de pouvoir modéliser de manière flexible des ensembles de contraintes portant sur plusieurs motifs locaux et ainsi de découvrir des motifs plus appropriés aux besoins de l'utilisateur. Il n'est plus nécessaire de développer une méthode ad hoc chaque fois que l'on veut extraire de nouveaux motifs n-aires. A notre connaissance, il s'agit de la première approche générique pour traiter ce problème.

La fertilisation croisée entre l'extraction de motifs et la PPC est un domaine de recherche émergeant. Un travail fondateur [6] propose une formulation PPC des contraintes sur les motifs locaux, mais il ne traite pas de l'extraction des motifs n-aires. Dans des travaux

Trans.	Items					
o_1	A	B			c_1	
o_2	A	B			c_1	
o_3			C		c_1	
o_4			C		c_1	
o_5			C		c_1	
o_6	A	B	C	D		c_2
o_7			C	D		c_2
o_8			C			c_2
o_9				D		c_2

TAB. 1 – Exemple de contexte transactionnel $\boldsymbol{r}$.

antérieurs [10, 11], nous avons proposé une approche hybride, reposant sur l'utilisation jointe d'un extracteur de motifs locaux et des Constraint Satisfaction Problems (CSP), pour extraire les motifs n-aires. Dans cet article, nous montrons l'apport d'une nouvelle approche fondée uniquement sur la PPC.

L'article est organisé comme suit : la section 2 présente le contexte général et introduit quelques définitions. Notre approche est décrite dans la section 3, et nous l'illustrons en modélisant plusieurs contraintes n-aires. La section 4 présente la modélisation des contraintes n-aires à l'aide de CSP. La section 5 dresse un bref état de l'art sur l'extraction de motifs n-aires et présente notre approche hybride. Dans la section 6, nous présentons notre nouvelle approche fondée uniquement sur la PPC. La section 7 compare l'approche PPC avec l'approche hybride. Enfin, la section 8 conclut en dressant quelques perspectives sur l'utilisation de la PPC pour l'extraction de motifs.

2 Contexte et motivations

2.1 Définitions

Soit $\mathcal{I}$ un ensemble de littéraux distincts appelés *items*, un motif ensembliste[1] d'items correspond à un sous-ensemble non vide de $\mathcal{I}$. Ces motifs sont regroupés dans le langage $\mathcal{L}_\mathcal{I} = 2^\mathcal{I} \backslash \emptyset$. Un contexte transactionnel est alors défini comme un multi-ensemble de motifs de $\mathcal{L}_\mathcal{I}$. Chacun de ces motifs, appelé transaction, constitue une entrée de la base de données. Ainsi, le tableau 1 présente un contexte transactionnel $\boldsymbol{r}$ où 9 transactions étiquetées $o_1, \ldots, o_9$ sont décrites par 6 items $A, \ldots, D, c_1, c_2$.

L'extraction de motifs a pour but la découverte d'informations à partir de tous les motifs ou d'un sous ensemble de $\mathcal{L}_\mathcal{I}$. L'extraction sous contraintes cherche la collection de tous les motifs de $\mathcal{L}_\mathcal{I}$ présents dans $\boldsymbol{r}$ et satisfaisant un prédicat appelé *contrainte*. Ces motifs sont appelés *motifs locaux* ; ce sont des régularités observées dans certaines parties des données. La localité de ces motifs provient du fait que, vérifier s'ils satisfont une contrainte donnée, peut s'effectuer indépendamment des autres motifs.

Il y a de nombreuses contraintes permettant d'évaluer la pertinence et la qualité des motifs locaux. Un exemple bien connu est celui de la contrainte de fréquence qui permet de rechercher les motifs X ayant une fréquence $freq(X)$ supérieure à un seuil minimal fixé $minfr > 0$. De nombreux travaux [16] remplacent la fréquence par d'autres mesures permettant d'évaluer l'intérêt des motifs locaux recherchés. C'est le cas de la mesure d'aire : soit X un motif, $aire(X)$ est le produit de la fréquence de X par sa taille, i.e., $aire(X) = freq(X) \times long(X)$ où $long(X)$ désigne la longueur (i.e., le nombre d'items) de X.

2.2 Motivations

En pratique, l'utilisateur est très souvent intéressé par la découverte de motifs plus riches que les motifs locaux et qui révèlent des caractéristiques et propriétés de l'ensemble de données étudié. De tels motifs portant sur plusieurs motifs locaux sont appelés *motifs n-aires* et permettent l'expression de *contraintes n-aires*.

Défintion 1 (contrainte n-aire)**.** *Une contrainte c est dite n-aire si elle porte sur plusieurs motifs locaux.*

Défintion 2 (motif n-aire)**.** *Un motif X est dit n-aire si il apparaît dans (au moins) une contrainte n-aire.*

Les contraintes n-aires permettent de modéliser un large ensemble de motifs utiles à l'utilisateur tel que la découverte de règles d'exceptions [20] ou la détection de règles susceptibles de donner lieu à des conflits de classifications dans le contexte de la classification associative [23]. D'autres contraintes n-aires sont présentées à la section 3.

Exemple 1 : E. Suzuki s'est intéressé à la découverte de paires de règles incluant une règle d'exception [20]. Une règle d'exception est une règle qui exprime une situation déviant d'un comportement général modélisé par une autre règle : l'intérêt de cette définition est d'expliciter la nature d'une exception par rapport à un comportement général et admis. Formellement, les règles d'exception sont définies comme suit (I est un item, par exemple une valeur de classe, X et Y sont des motifs locaux) :

$$e(X,Y,I) \equiv \begin{cases} vrai & \text{si } \exists Y \in \mathcal{L}_\mathcal{I} \text{ tq } Y \subset X, \\ & (X \backslash Y \rightarrow I) \wedge (X \rightarrow \neg I) \\ faux & \text{sinon} \end{cases}$$

[1] Dans cet article, nous nous intéressons au cas des motifs ensemblistes

Dans une telle paire de règles, $X \backslash Y \rightarrow I$ est une règle générale et $X \rightarrow \neg I$ est une règle d'exception qui révèle ainsi une information inattendue. Cette définition demande à ce que la règle générale soit de forte fréquence et de forte confiance tandis que la règle d'exception est peu fréquente mais de très forte confiance (la confiance d'une règle $X \rightarrow Y$ est mesurée par le rapport $freq(X \cup Y)/freq(X)$). La comparaison entre la règle générale et la règle d'exception ne peut pas être modélisée par une approche reposant uniquement sur les motifs locaux. Par contre, elle se modélise aisément à l'aide des contraintes n-aires. Donnons un exemple de règle d'exception à partir du tableau 1. En prenant 2/3 comme seuil de confiance d'une règle, la règle $AC \rightarrow \neg c_1$ est une règle d'exception puisque nous avons conjointement $A \rightarrow c_1$ et $AC \rightarrow \neg c_1$. E. Suzuki a proposé une méthode fondée sur une estimation probabiliste [20] pour extraire de telles paires de règles. Mais, cette approche est totalement dédiée à ce type de motifs.

Exemple 2 : Considérons l'exemple du transcriptome et de l'analyse d'expressions de gènes : le biologiste est vivement intéressé par la recherche de groupes de synexpressions. Les motifs locaux, composés de tags (ou gènes), qui satisfont la contrainte d'aire (cf. section 2.1), sont susceptibles de donner lieu à des groupes de synexpressions [12]. D'autre part, il faut être capable de prendre en compte l'incertain qui est présent dans ce type de données [1]. Les contraintes n-aires sont une façon naturelle de concevoir des motifs tolérants aux fautes et candidats à être des groupes de synexpressions : ceux-ci sont définis par l'union de plusieurs motifs locaux satisfaisant une contrainte d'aire et ayant un fort recouvrement entre eux. Plus précisément, à partir de deux motifs locaux X et Y, on définit la contrainte n-aire suivante :

$$c(X,Y) \equiv \begin{cases} aire(X) > min_{aire} \wedge \\ aire(Y) > min_{aire} \wedge \\ aire(X \cap Y) > \alpha \times min_{aire} \end{cases}$$

où min_{aire} est le seuil minimal d'aire et α est un paramètre fourni par l'utilisateur pour fixer le recouvrement minimal entre motifs locaux.

3 Exemples de contraintes n-aires

Dans cette section nous présentons plusieurs exemples de contraintes n-aires. Certaines d'entre elles ont déjà été introduites à la section 2.2.

3.1 Règles d'exception

Soient X et Y deux motifs, et I un item tel que I et $\neg I \in \mathcal{I}$ (I et $\neg I$ peuvent représenter deux classes présentes dans le jeu de données). Soient les seuils de fréquence *minfr* et *maxfr* et les seuils de confiance δ_1 et δ_2. La contrainte n-aire relative aux règles d'exception se modélise comme suit :

- $X \backslash Y \rightarrow I$ doit être une règle fréquente de forte confiance : $freq((X \backslash Y) \sqcup I) \geq minfr \wedge freq(X \backslash Y) - freq((X \setminus Y) \sqcup I) \leq \delta_1$.
- $X \rightarrow \neg I$ doit être une règle rare de forte confiance : $freq(X \sqcup \neg I) \leq maxfr \wedge (freq(X) - freq(X \sqcup \neg I)) \leq \delta_2$.

En résumé :

$$e(X,Y,I) \equiv \begin{cases} \exists Y \subset X \text{ tq :} \\ freq((X \setminus Y) \sqcup I) \geq minfr \wedge \\ (freq(X \setminus Y) - freq((X \setminus Y) \sqcup I)) \leq \delta_1 \wedge \\ freq(X \sqcup \neg I) \leq maxfr \wedge \\ (freq(X) - freq(X \sqcup \neg I)) \leq \delta_2 \end{cases}$$

3.2 Règles inattendues

Padmanabhan et Tuzhilin ont introduit dans [18] la notion de règle *inattendue* $X \rightarrow Y$ par rapport à une croyance $U \rightarrow V$ où U et V sont des motifs. Une règle inattendue est définie dans [18] par :

1. $Y \wedge V$ n'est pas valide,
2. $X \wedge U$ est valide (XU est un motif fréquent),
3. $XU \rightarrow Y$ est valide ($XU \rightarrow Y$ est une règle fréquente et de confiance suffisante),
4. $XU \rightarrow V$ n'est pas valide (soit $XU \rightarrow V$ n'est pas une règle fréquente, soit $XU \rightarrow V$ est une règle de faible confiance).

Etant donnée une croyance $U \rightarrow V$, une règle inattendue $un(X,Y)$ est modélisée par :

$$un(X,Y) \equiv \begin{cases} freq(Y \cup V) = 0 \wedge \\ freq(X \cup U) \geq minfr_1 \wedge \\ freq(X \cup U \cup Y) \geq minfr_2 \wedge \\ freq(X \cup U \cup Y)/freq(X \cup U) \geq minconf \wedge \\ (freq(X \cup U \cup V) < maxfr \vee \\ freq(X \cup U \cup V)/freq(X \cup U) < maxconf) \end{cases}$$

3.3 Groupes de synexpressions

La recherche de groupes de synexpressions à partir de n motifs locaux se modélise à l'aide de la contrainte n-aire suivante :

$$synexpr(X_1, ..., X_n) \equiv \begin{cases} \forall\ 1 \leq i < j \leq n, \\ aire(X_i) > min_{aire} \wedge \\ aire(X_j) > min_{aire} \wedge \\ aire(X_i \cap X_j) > \alpha \times min_{aire} \end{cases}$$

où min_{aire} désigne la surface minimale (définie à la section 2.1) et α est un seuil, défini par l'utilisateur,

permettant de quantifier le recouvrement minimal souhaité. Cet exemple montre comment on peut modéliser des motifs complexes et tolérants aux fautes tels que les groupes de synexpressions.

3.4 Conflits de classification

La combinaison des motifs locaux est un point clé de la qualité d'un classifieur à base d'associations qui est généralement construit à partir de règles fréquentes et de forte confiance. Typiquement, les paires de règles ayant un important chevauchement entre leurs prémisses et concluant sur des classes distinctes sont particulièrement susceptibles de donner lieu à un conflit de classification. En effet, lorsqu'une règle d'une telle paire est déclenchée par un exemple à classer, l'autre règle concluant sur une autre valeur de classe est fortement susceptible d'être également déclenchée car les prémisses des deux règles sont relativement similaires. Ce double déclenchement conduira à un conflit de classification. En étant capable de prendre en compte plusieurs motifs locaux, les contraintes n-aires permettent de modéliser de façon naturelle de tels conflits de classification. Soient $X \rightarrow c_1$ et $Y \rightarrow c_2$ deux règles fréquentes et de forte confiance, une paire de règles susceptibles de donner lieu à un conflit de classification s'exprime de la façon suivante :

$$c(X,Y) \equiv \begin{cases} freq(X) \geq minfr \wedge \\ freq(Y) \geq minfr \wedge \\ (freq(X \sqcup \{c_1\})/freq(X)) \geq minconf \wedge \\ (freq(Y \sqcup \{c_2\})/freq(Y)) \geq minconf \wedge \\ long(X \cap Y) \geq (long(X) + long(Y))/4 \end{cases}$$

Les quatre premières contraintes d'inégalité portent sur la fréquence et la confiance des règles de classification. La dernière contrainte décrit le chevauchement souhaité : les deux règles doivent avoir en commun au moins la moitié des items de leurs prémisses. Notons qu'il est simple pour l'utilisateur de modifier les paramètres de la contrainte n-aire et/ou ajouter de nouvelles contraintes pour modéliser des types de conflits de classification plus spécifiques.

4 Modélisation sous forme de CSP

4.1 Aperçu général

Soit $\boldsymbol{r}$ un jeu de données ayant nb transactions et $\mathcal{I}$ l'ensemble de ses items. La recherche de motifs ensemblistes peut se modéliser à l'aide de deux CSP $\mathcal{P}$ et $\mathcal{P}'$ inter-reliés :

1. CSP ensembliste $\mathcal{P}=(\mathcal{X}, \mathcal{D}, \mathcal{C})$ où :
   - $\mathcal{X}=\{X_1, ..., X_n\}$. Chaque variable X_i représente un motif ensembliste inconnu.
   - $\mathcal{D}=\{D_{X_1}, ..., D_{X_n}\}$. Le domaine initial de chaque variable X_i est $[\{\} \,..\, \mathcal{I}]$.
   - $\mathcal{C}$ est une conjonction de contraintes ensemblistes formulées à l'aide d'opérateurs ensemblistes ($\cup$, $\cap$, $\setminus$, $\in$, $\notin$, ...).
2. CSP numérique $\mathcal{P}'=(\mathcal{F}, \mathcal{D}', \mathcal{C}')$ où :
   - $\mathcal{F}=\{F_1, ..., F_n\}$. Chaque variable F_i est la fréquence du motif X_i.
   - $\mathcal{D}'=\{D_{F_1}, ..., D_{F_n}\}$. Le domaine initial de chaque variable F_i est $[1 \,..\, nb]$.
   - $\mathcal{C}'$ est une conjonction de contraintes numériques telles que : $<, \leq, \neq, =, ...$

4.2 Exemple des règles d'exception

Contrainte	Formulation
$freq((X \setminus Y) \sqcup I) \geq minfr$	$F_2 \geq minfr$ $\wedge$ $I \in X_2$ $\wedge$ $X_1 \subsetneq X_3$
$freq(X \setminus Y) - freq((X \setminus Y) \sqcup I) \leq \delta_1$	$F_1 - F_2 \leq \delta_1$ $\wedge$ $X_2 = X_1 \sqcup I$
$freq(X \sqcup \neg I) \leq maxfr$	$F_4 \leq maxfr$ $\wedge$ $\neg I \in X_4$
$freq(X) - freq(X \sqcup \neg I) \leq \delta_2$	$F_3 - F_4 \leq \delta_2$ $\wedge$ $X_4 = X_3 \sqcup \neg I$

TAB. 2 – Formulation des contraintes

La table 2 décrit l'ensemble des contraintes primitives modélisant les règles d'exception.

- Les variables ensemblistes $\{X_1, X_2, X_3, X_4\}$ représentent les motifs recherchés :
  - $X_1 : X \setminus Y$, et $X_2 : (X \setminus Y) \sqcup I$ (règle générale),
  - $X_3 : X$, et $X_4 : X \sqcup \neg I$ (règle d'exception).
- Les variables entières $\{F_1, F_2, F_3, F_4\}$ représentent leurs fréquences.
- Contraintes ensemblistes : $\mathcal{C} = \{(I \in X_2), (X_2 = X_1 \sqcup I), (\neg I \in X_4), (X_4 = X_3 \sqcup \neg I), (X_1 \subsetneq X_3)\}$
- Contraintes numériques : $\mathcal{C}' = \{(F_2 \geq minfr), (F_1 - F_2 \leq \delta_1), (F_4 \leq maxfr), (F_3 - F_4 \leq \delta_2)\}$

5 Etat de l'art

5.1 Découverte de motifs en fouille de données

Alors qu'il existe de nombreux travaux traitant de la découverte de motifs locaux sous contraintes [5, 16], très peu d'approches considérant simultanément plusieurs motifs locaux ont été proposées : citons les "patterns teams" [14], les ensembles sous contraintes de motifs [7] ou encore la sélection de motifs suivant leur intérêt compte tenu d'autres motifs déjà sélectionnés [3]. Même si ces approches comparent explicitement les motifs locaux entre eux, elles sont principalement fondées sur la réduction de la redondance entre motifs ou

poursuivent des objectifs spécifiques tels que la classification. De part leur flexibilité, les contraintes n-aires permettent à l'utilisateur d'exprimer, dans un cadre unique de modélisation, des biais de recherche variés et donc des types de motifs très divers. Notons qu'il existe des cadres génériques pour la construction de modèles globaux à partir de motifs locaux [13, 9]. Mais, ces cadres ne proposent pas de méthode d'extraction : ils permettent seulement de mieux comparer les approches existantes.

5.2 Une approche hybride

Nous avons proposé [10, 11] une première approche fondée sur l'utilisation conjointe d'un extracteur de motifs locaux et de la PPC pour extraire des motifs n-aires. Dans cette section, nous donnons une vue d'ensemble de cette approche hybride avant de détailler chacune de ses trois étapes en considérant l'exemple des règles d'exception (cf. la section 4.2).

5.2.1 Aperçu général

La figure 1 présente une vue d'ensemble des trois étapes de notre approche :

1. Modéliser la contrainte n-aire sous forme de CSP, puis distinguer les contraintes locales (portant sur un seul motif) des autres (portant sur plusieurs motifs).
2. Résoudre les contraintes locales à l'aide d'un extracteur de motifs locaux (Music-dfs[2]) [19] qui produit une représentation condensée par intervalles de tous les motifs satisfaisant les contraintes locales.
3. Résoudre les autres contraintes[3] à l'aide du solveur de contraintes ECL^iPS^e[4]. Le domaine de chaque variable résulte de la représentation condensée par intervalles calculée dans la seconde étape.

5.2.2 Partitionner les contraintes

L'ensemble de toutes les contraintes $(\mathcal{C} \cup \mathcal{C}')$ est divisé en deux sous ensembles :

- $\mathcal{C}_{loc}$ est l'ensemble des contraintes locales à résoudre par Music-dfs. Les solutions sont fournies sous forme d'une représentation condensée par intervalles.

[2] `http://www.info.univ-tours.fr/~soulet/music-dfs/music-dfs.html`

[3] Ces contraintes portent sur plusieurs variables, et sont donc n-aires au sens PPC. Pour éviter toute confusion, le terme n-aire sera réservé aux contraintes portant sur plusieurs motifs locaux au sens de la fouille de données (cf. la définition 1).

[4] `http://www.eclipse-clp.org`

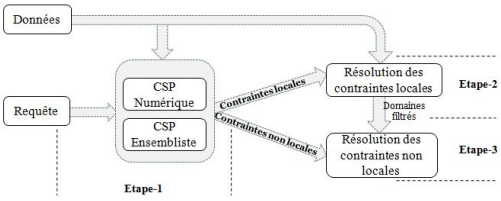

Fig. 1 – Aperçu des 3 étapes

- $\mathcal{C}_{autres}$ est l'ensemble des contraintes restantes à résoudre par ECL^iPS^e. Les domaines des variables X_i et F_i sont déduits de la représentation condensée par intervalles calculée dans l'étape précédente.

Pour les règles d'exception (cf. la table 2), cette partition s'effectue comme suit :

- $\mathcal{C}_{loc} = \{(I \in X_2), (F_2 \geq \mathit{minfr}), (F_4 \leq \mathit{maxfr}), (\neg I \in X_4)\}$
- $\mathcal{C}_{autres} = \{(F_1 - F_2 \leq \delta_1), (X_2 = X_1 \sqcup I), (F_3 - F_4 \leq \delta_2), (X_4 = X_3 \sqcup \neg I), (X_1 \subsetneq X_3)\}$

5.2.3 Résolution des contraintes locales

Music-dfs est un extracteur correct et complet de motifs locaux qui permet d'extraire efficacement l'ensemble des motifs satisfaisant un large éventail de contraintes locales [19]. L'efficacité de Music-dfs est due à sa stratégie de recherche en profondeur d'abord et à l'élagage de l'espace de recherche se basant sur la propriété d'anti-monotonie dans l'espace des motifs candidats. Les motifs locaux sont produits sous forme de représentation condensée composée d'intervalles disjoints où chaque intervalle synthétise un ensemble de motifs satisfaisants la contrainte [19].

Pour la méthode hybride, les contraintes locales sont résolues avant et indépendamment des autres contraintes. Ainsi, l'espace de recherche des autres contraintes sera réduit à l'espace des solutions des contraintes locales.

L'ensemble des contraintes locales $\mathcal{C}_{loc}$ est partitionné en une union disjointe de $\mathcal{C}_i$ (pour $i \in [1..n]$) où chaque $\mathcal{C}_i$ est l'ensemble des contraintes locales portant sur X_i et F_i. Chaque C_i peut être résolu de manière séparée et indépendante. Soit CR_i la représentation condensée sous forme d'intervalles issue de la résolution de $\mathcal{C}_i$ par Music-dfs, $CR_i = \bigcup_p (f_p, I_p)$ où chaque I_p est un intervalle ensembliste vérifiant : $\forall x \in I_p,\ \mathit{freq}(x) = f_p$. Les domaines des variables X_i et F_i sont définis par :

- $D_{F_i} = \{f_p \mid f_p \in CR_i\}$,
- $D_{X_i} = \bigcup_{I_p \in CR_i} I_p$.

Exemple : Considérons $\mathcal{C}_{loc}$ l'ensemble des contraintes locales relatives aux règles d'exception, et prenons comme valeurs respectives de $(I, \neg I, minfr, \delta_1, maxfr, \delta_2)$ les valeurs suivantes : $(c_1, c_2, 2, 1, 1, 0)$. L'ensemble des contraintes locales relatives à X_2 et F_2, $\mathcal{C}_2 = \{c_1 \in X_2, F_2 \geq 2\}$, est résolu par MUSIC-DFS par la requête suivante, où les paramètres peuvent être directement déduits de $\mathcal{C}_2$:

```
---------------
music-dfs data -q "{c1} subset X2 and freq(X2)>=2;"
X2 in [A, c1]..[A, c1, B ] U [B, c1] -- F2 = 2 ;
X2 in [C, c1] -- F2 = 3
---------------
```

5.2.4 Résolution des contraintes non locales

Les domaines des variables X_i et F_i sont obtenus à partir de la représentation condensée sous forme d'intervalles de tous les motifs satisfaisant les contraintes locales. La résolution de ces contraintes par ECL^iPS^e permettra ainsi d'obtenir l'ensemble de tous les motifs satisfaisant l'intégralité des contraintes.

Exemple : Considérons $\mathcal{C}_{autres}$ l'ensemble des contraintes non locales pour les règles d'exception, les valeurs respectives de $(I, \neg I, minfr, \delta_1, maxfr, \delta_2)$ étant toujours les mêmes. La session ECL^iPS^e ci-dessous illustre comment toutes les paires de règles d'exception peuvent être obtenues par backtracking :

```
---------------
[eclipse 1]:
?- exceptionsRules(X1, X2, X3, X4).
X1=[A,B], X2=[A,B,c1], X3=[A,B,C], X4=[A,B,C,c2];
X1=[A,B], X2=[A,B,c1], X3=[A,B,D], X4=[A,B,D,c2];
.../...
---------------
```

6 Une approche PPC

Cette section présente une nouvelle approche fondée uniquement sur la PPC. Cette approche repose, elle aussi, sur la modélisation sous forme de CSP (cf Section 4). Mais, à la différence de la méthode hybride (cf. la section 5.2), cette approche dite "PPC" n'utilise pas d'extracteur de motifs locaux. La section 7 compare en profondeur ces deux approches.

La modélisation s'effectue en trois étapes :

1. Lier les transactions et les motifs n-aires,
2. Modélisation des motifs n-aires recherchés,
3. Reformulation des contraintes ensemblistes et des contraintes numériques.

Cette approche a été mise en oeuvre en Gecode[5]. Pour la première étape (cf. section 6.1), nous avons utilisé l'implantation de l'extracteur de motifs FIM_CP[6] qui est une approche PPC pour l'extraction de motifs [6]. Cette approche traite dans un cadre unifié un large ensemble de motifs locaux et de contraintes telles que la fréquence, la fermeture, la maximalité, les contraintes monotones ou anti-monotones et des variations de ces contraintes. Mais FIM_CP ne traite pas les contraintes n-aires.

[5] http://www.gecode.org

[6] http://www.cs.kuleuven.be/~dtai/CP4IM/fim_cp.php

6.1 Transactions et les motifs n-aires

Nous avons utilisé une technique similaire à celle de FIM_CP pour établir le lien entre l'ensemble des transactions et les motifs n-aires que nous recherchons.

Soit $\boldsymbol{r}$ un contexte transactionnel où $\mathcal{I}$ est l'ensemble de ses n items et $\mathcal{T}$ l'ensemble de ses m transactions. Soit $\boldsymbol{d}$ la matrice 0/1 de dimension (m, n) telle que $\forall t \in \mathcal{T}, \forall i \in \mathcal{I}$, $(d_{t,i} = 1)$ ssi $(i \in t)$. Soit M le motif (unique) recherché par FIM_CP. Deux sortes de variables de décision (de domaine $\{0, 1\}$) sont utilisées :

- $\{M_1, M_2, ..., M_n\}$ où $(M_i = 1)$ ssi $(i \in M)$,
- $\{T_1, T_2, ..., T_m\}$ où $(T_t = 1)$ ssi $(M \subseteq t)$.

Ainsi, $freq(M) = \sum_{t \in \mathcal{T}} T_t$ et $long(M) = \sum_{i \in \mathcal{I}} M_i$.

La relation entre le motif recherché M et $\mathcal{T}$ est établie via des contraintes réifiées [6] imposant que, pour chaque transaction t, $(T_t = 1)$ ssi $(M \subseteq t)$, ce qui se reformule en :

$$\forall t \in \mathcal{T}, (T_t = 1) \Leftrightarrow \sum_{i \in \mathcal{I}} M_i \times (1 - d_{t,i}) = 0 \qquad (1)$$

Le filtrage associé à chaque contrainte réifiée (cf Equation 1) procède comme suit : si l'on peut déduire que (T_t=1) (resp. T_t=0), alors la somme soit être nulle (resp. non-nulle). La propagation s'effectue de manière analogue de la partie droite vers la partie gauche de l'équivalence.

6.2 Modélisation des k motifs n-aires recherchés

Soit $\boldsymbol{r}$ un contexte transactionnel où $\mathcal{I}$ est l'ensemble de ses n items et $\mathcal{T}$ l'ensemble de ses m transactions. Soit $\boldsymbol{d}$ la matrice 0/1 de dimension (m, n) telle que $\forall t \in \mathcal{T}, \forall i \in \mathcal{I}$, $(d_{t,i} = 1)$ ssi $(i \in t)$. Soient $X_1, X_2, ..., X_k$ les k motifs n-aires recherchés.

Tout d'abord, chaque motif n-aire inconnu X_j est modélisé par n variables de décision $\{X_{1,j}, X_{2,j}, ..., X_{n,j}\}$ (de domaine $\{0, 1\}$) telles que $(X_{i,j} = 1)$ ssi l'item i appartient au motif X_j :

$$\forall i \in \mathcal{I}, (X_{i,j} = 1) \Leftrightarrow (i \in X_j) \qquad (2)$$

Puis, m variables de décision $\{T_{1,j}, T_{2,j}, ..., T_{m,j}\}$ (de domaine $\{0, 1\}$) sont associées à chaque motif n-aire inconnu X_j telles que $(T_{t,j} = 1)$ ssi $(X_j \subseteq t)$:

$$\forall t \in \mathcal{T}, (T_{t,j} = 1) \Leftrightarrow (X_j \subseteq t) \qquad (3)$$

Ainsi, $freq(X_j)=\sum_{t\in\mathcal{T}} T_{t,j}$ et $long(X_j)=\sum_{i\in\mathcal{I}} X_{i,j}$.

La relation entre le motif recherché X_j et $\mathcal{T}$ est établie via des contraintes réifiées imposant que, pour chaque transaction t, $(T_{t,j}=1)$ ssi $(X_j \subseteq t)$, ce qui se reformule en :

$$\forall j \in [1..k], \forall t \in \mathcal{T}, (T_{t,j}=1) \Leftrightarrow \sum_{i\in\mathcal{I}} X_{i,j}\times(1-d_{t,i}) = 0 \quad (4)$$

6.3 Contraintes numériques et ensemblistes

Considérons un opérateur $\texttt{op} \in \{<, \leq, >, \geq, =, \neq\}$; les contraintes numériques se reformulent comme suit :

- $freq(X_p)$ op $\alpha \rightarrow \sum_{t\in\mathcal{T}} T_{t,p}$ op α
- $long(X_p)$ op $\alpha \rightarrow \sum_{i\in\mathcal{I}} X_{i,p}$ op α

Certaines contraintes ensemblistes (telles que égalité, inclusion, appartenance,...) se reformulent directement à l'aide de contraintes linéaires :

- $X_p = X_q \rightarrow \forall i \in \mathcal{I}, X_{i,p} = X_{i,q}$
- $X_p \subseteq X_q \rightarrow \forall i \in \mathcal{I}, X_{i,p} \leq X_{i,q}$
- $i_o \in X_p \rightarrow X_{i_0,p} = 1$

Les autres contraintes ensemblistes (telles que intersection, union, différence,...) se reformulent aisément à l'aide de contraintes booléennes en utilisant la fonction de conversion $(b::\{0,1\} \rightarrow \{False, True\})$ et les opérateurs booléens usuels :

- $X_p \cap X_q = X_r \rightarrow \forall i \in \mathcal{I}, b(X_{i,r}) = b(X_{i,p}) \wedge b(X_{i,q})$
- $X_p \cup X_q = X_r \rightarrow \forall i \in \mathcal{I}, b(X_{i,r}) = b(X_{i,p}) \vee b(X_{i,q})$
- $X_p \backslash X_q = X_r \rightarrow \forall i \in \mathcal{I}, b(X_{i,r}) = b(X_{i,p}) \wedge \neg b(X_{i,q})$

Enfin, l'ensemble des contraintes, qu'elles soient réifiées, numériques ou ensemblistes, est traité par *Gecode*.

6.4 Expérimentations

Cette section a pour but de montrer la faisabilité et les apports pratiques de l'approche PPC présentée dans cet article.

6.4.1 Protocole expérimental

Différentes expérimentations ont été menées sur plusieurs jeux de données de l'UCI *repository*[7] ainsi que sur un jeu de données réelles nommé `Meningitis` et provenant de l'Hôpital Central de Grenoble. `Meningitis` recense les pathologies des enfants atteints d'une méningite virale ou bactérienne. La table 3 résume les différentes caractéristiques de ces jeux de données.

Les expérimentations ont été menées sur plusieurs contraintes n-aires : règles d'exception, règles rares et conflits de classification. La machine utilisée est un PC 2.83 GHz Intel Core 2 Duo processor (4 GB de RAM) sous Ubuntu Linux.

[7] `http://www.ics.uci.edu/~mlearn/MLRepository.html`

Jeu de données	#trans	#items	densité
`Mushroom`	8142	117	0.18
`Australian`	690	55	0.25
`Meningitis`	329	84	0.27

TAB. 3 – Description des jeux de données

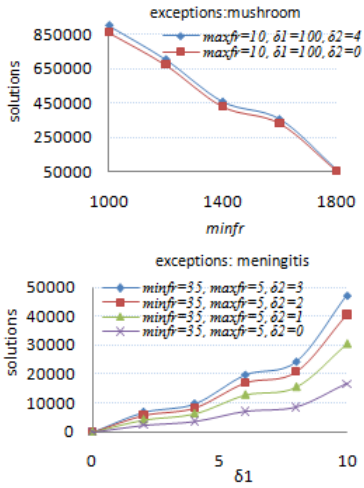

FIG. 2 – Evolution du nombre de règles d'exception

6.4.2 Correction et complétude de l'approche

Comme la résolution effectuée par le solveur de contraintes est correcte et complète, notre approche permet d'extraire l'ensemble correct et complet des motifs satisfaisant n'importe quelle contrainte n-aire. La figure 2 décrit l'évolution du nombre de paires de règles d'exception en fonction des seuils $minfr$ et δ_1 pour les jeux de données `Mushroom` et `Meningitis`. La figure 3 décrit l'évolution du nombre de conflits de classification en fonction des seuils $minfr$ et $minconf$ pour les jeux de données `Mushroom` et `Australian`. Nous avons aussi testé d'autres valeurs de ces paramètres ainsi que d'autres jeux de données. Mais, comme les résultats obtenus sont similaires, ils ne sont pas indiqués ici. Comme attendu, plus $minfr$ est petit, plus le nombre de règles d'exception est grand. Les résultats sont similaires quand δ_1 varie. Plus δ_1 est grand, plus le nombre de règles d'exception augmente (quand δ_1 augmente, la confiance décroît et il y a donc un plus grand nombre de règles générales).

6.4.3 Mise en valeur de motifs demandés par les utilisateurs

Les règles d'exception sont un cas particulier des règles rares (cf. la section 2.2). Même s'il existe quelques travaux permettant d'extraire les règles rares [21], il est impossible de distinguer les règles d'ex-

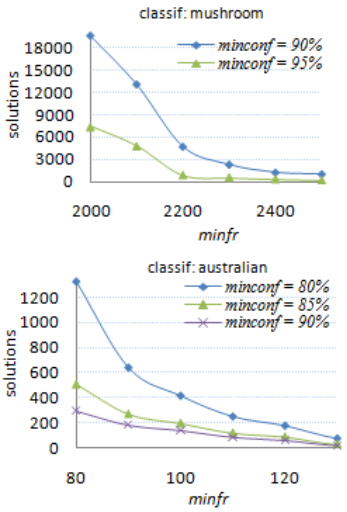

FIG. 3 – Evolution du nombre de conflits de classification

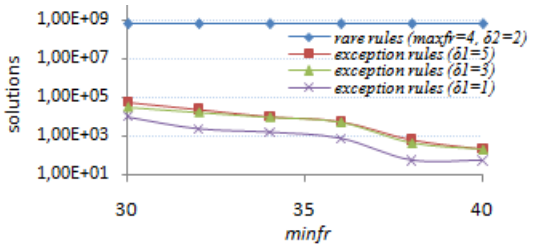

FIG. 4 – Nombre de règles d'exception vs nombre de règles rares (`Meningitis`)

ception à partir de l'ensemble des règles rares. C'est une limitation forte car la plupart des règles rares sont peu fiables, d'où l'intérêt des règles d'exception et de leur extraction. La figure 4 compare, sur le jeu de données `Meningitis`, le nombre de règles d'exception par rapport au nombre de règles rares en fonction du seuil de fréquence $minfr$ (le nombre de règles rares correspond à la ligne en haut de la figure 4). Savoir rechercher directement les règles d'exception permet de réduire de manière drastique (plusieurs ordres de magnitude) le nombre de motifs obtenus (noter que l'axe des ordonnées suit une échelle logarithmique).

Les règles inattendues sont également sources d'informations utiles. Un exemple d'une telle règle sur le jeu de données `Meningitis` est la règle dont la prémisse est composée d'un taux élevé de polynucléaires neutrophiles, de l'absence de signes neurologiques et dont la conclusion est une valeur normale pour le taux de polynucléaires immatures. Cette règle est inattendue par rapport à la croyance que des valeurs élevées de la numération des leucocytes et du taux de polynucléaires impliquent une méningite d'étiologie bactérienne. Les experts apprécient de disposer des contraintes n-aires pour la découverte de tels motifs.

6.4.4 Efficacité

Ces expérimentations permettent de quantifier les temps de calcul et le passage à l'échelle de notre approche. En pratique, les temps de calcul varient en fonction de la taille des jeux de données et de la dureté des contraintes[8]. C'est le cas des contraintes définies par des seuils de fréquence et de confiance élevés.

Pour `Meningitis` et `Australian`, l'ensemble de toutes les solutions est obtenu en quelques secondes (moins d'une seconde dans la plupart des cas). Pour `Mushroom`, les temps de calcul varient de quelques secondes pour les contraintes dures à environ 1 heure pour des seuils très faibles de fréquence et de confiance. La figure 5 décrit les temps de calcul obtenus sur `Mushroom` en fonction des seuils de fréquence et de confiance retenus. Plus une contrainte est dure, plus les temps de calcul sont petits. Les temps de calcul dépendent aussi de la taille du jeu de données traité : plus il est grand, plus le nombre de contraintes est élevé (cf. la section 7.2).

Bien évidemment, notre approche pourrait être utilisée pour extraire des motifs locaux. Nous obtenons dans ce cas les mêmes temps de calcul que [6] qui sont compétitifs vs les extracteurs de motifs locaux. Enfin, pour les règles d'exception, nous n'avons pas pu effectuer de comparaison avec la méthode ad hoc car les temps de calcul ne sont pas indiqués dans [20].

7 Comparaison des deux approches

Si l'approche hybride (cf. section 5.2) et l'approche PPC présentée dans cet article (cf. section 6) reposent sur la même modélisation, leurs mises en oeuvre diffèrent. L'approche hybride utilise un extracteur de motifs locaux afin de construire la représentation condensée sous forme d'intervalles de tous les motifs satisfaisant les contraintes locales. Cette représentation condensée permet de construire les domaines des variables du CSP ensembliste. L'approche PPC établit directement le lien entre l'ensemble des transactions et les motifs n-aires recherchés à l'aide de contraintes réifiées. Le CSP obtenu est directement résolu (sans l'aide d'un extracteur de motifs locaux).

[8] Une contrainte est dure si le ratio entre son nombre solutions et la cardinalité du produit cartésien des domaines de ses variables est faible.

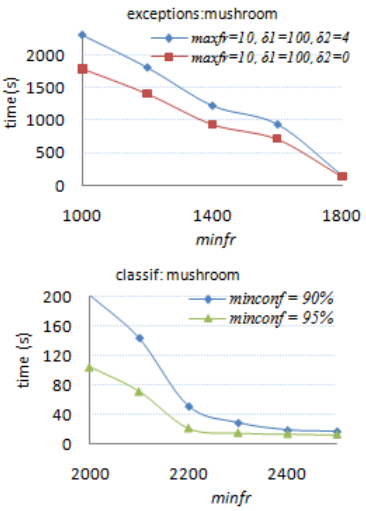

FIG. 5 – Temps d'exécution

7.1 Approche hybride : Pro/Cons

Avec l'approche hybride, la modélisation d'une contrainte n-aire peut être directement fournie au solveur de contraintes sans aucune re-formulation. Ainsi, un prototype a pu être rapidement développé en utilisant ECL^iPS^e. De plus, cette approche permet de bénéficier des avancées sur les extracteurs de motifs locaux.

Mais, les solveurs de CSP ensemblistes [8] ont du mal à gérer les unions d'intervalles. Afin d'établir la consistance aux bornes, l'union de deux intervalles ensemblistes est approximée par leur enveloppe convexe[9]. Afin de contourner ce problème, pour chaque variable X_i ayant pour représentation condensée $CR_i = \bigcup_p(f_p, I_p)$, la recherche s'effectue successivement sur chacun des I_p. Mais, si cette approche demeure correcte et complète, elle ne permet pas de profiter pleinement du filtrage car les retraits de valeurs non-viables se propagent uniquement sur les intervalles traités, et non pas sur l'intégralité des domaines.

Cette constatation pourrait paraître rédhibitoire, mais le nombre d'intervalles ensemblistes décroît rapidement lors de la prise en compte des contraintes locales. La table 4 décrit l'évolution du nombre d'intervalles ensemblistes constituant le domaine de la variable X_2 au fur et à mesure de la prise en compte des contraintes locales. (cf. l'exemple des règles d'exception à la section 5.2).

Une approche alternative serait d'utiliser des représentations condensées non exactes afin de réduire le nombre d'intervalles ensemblistes par domaine, comme

[9] L'enveloppe convexe de $[lb_1 .. ub_1]$ et $[lb_2 .. ub_2]$ est définie par $[lb_1 \cap lb_2 .. ub_1 \cup ub_2]$.

Contraintes locales	Nombre d'intervalles de D_{X_2}
-	3002
$I \in X_2$	1029
$I \in X_2 \wedge freq(X_2) >= 20$	52
$I \in X_2 \wedge freq(X_2) >= 25$	32

TAB. 4 – Nombre d'intervalles pour différentes contraintes locales (cas de D_{X_2})

par exemple une représentation condensée reposant sur les motifs fréquents maximaux [4]. Dans ce cas, le nombre d'intervalles par domaine sera beaucoup plus petit, mais en raison de l'approximation de l'union de deux intervalles par leur enveloppe convexe, il sera nécessaire de gérer les valeurs non viables qui en résultent.

7.2 Approche PPC : Pro/Cons

Tout d'abord, il est possible d'automatiser la reformulation des contraintes n-aires vers les contraintes primitives (cf. section 6.3). De plus, le propagateur des contraintes réifiées implanté en *Gecode* est très efficace. Mais, le nombre total de contraintes peut être très élevé pour des jeux de données de très grande taille. Considérons un jeu de données ayant n items, m transactions et une contrainte n-aire portant sur k motifs inconnus. Lier les transactions et les motifs n-aires nécessite $(k \times m)$ contraintes, chacune d'entre elles portant sur au plus $(n+1)$ variables (cf. l'équation 4 à la section 6.2). Ainsi, le nombre total de contraintes nécessaires à la modélisation de problèmes de très grande taille pourrait s'avérer prohibitif. En pratique, l'approche PPC permet de traiter des problèmes de grande taille (cf. section 6.4).

La différence fondamentale entre les deux approches réside dans leur façon de considérer l'ensemble des transactions. L'approche hybride utilise un extracteur de motifs locaux et le CSP résultant possède un très petit nombre de contraintes et de variables, mais ces variables ont des domaines de grande taille. À, l'inverse, l'approche PPC requiert un grand nombre de contraintes portant sur des variables de décision. Une voie médiane reste à trouver entre ces deux approches afin de pouvoir traiter des problèmes de très grande taille. L'extraction de motifs en utilisant la PPC est un domaine de recherche émergeant pour lequel il y a actuellement peu de travaux [6, 10, 11, 17].

8 Conclusions et perspectives

Nous avons proposé une approche PPC permettant de modéliser et d'extraire les motifs n-aires en fouille

de données. A notre connaissance, il s'agit de la première approche générique pour traiter ce problème. La modélisation décrite à la section 3 illustre la généricité et la flexibilité de notre approche. Les expérimentations menées (cf. section 6.4) montrent sa pertinence et sa faisabilité.

Les variables d'un CSP sont toutes quantifiées existentiellement Mais, certaines contraintes n-aires requièrent la quantification universelle pour être modélisées de manière concise et élégante, comme par exemple la contrainte *peak*. Un *peak* motif est un motif dont tous les voisins ont une valeur (selon une mesure) inférieure à un seuil. Pour cela, les QCSP [15, 22] nous semblent une piste prometteuse.

D'autre part, la découverte de connaissances dans les bases de données est un processus itératif et interactif guidé par l'utilisateur. Ce dernier ne devrait avoir qu'une vision de haut niveau du système de découverte de motifs en ayant à sa disposition un ensemble de contraintes (elles aussi de haut niveau) pour spécifier de façon déclarative les motifs désirés. Bien que nouvelle, l'approche PPC nous semble très prometteuse pour construire de tels systèmes.

Remerciements. Nous remercions le Dr P. François de l'Hôpital Central de Grenoble qui nous a fourni le jeu de données `Meningitis` et Arnaud Soulet pour les discussions fructueuses et MUSIC-DFS. Ce travail est partiellement financé par l'ANR (projet Bingo 2 ANR-07-MDCO-014).

Références

[1] J. Besson, C. Robardet, and J-F. Boulicaut. Mining a new fault-tolerant pattern type as an alternative to formal concept discovery. In *ICCS'06*, pages 144–157, Aalborg, Denmark, 2006. Springer-Verlag.

[2] F. Bonchi, F. Giannotti, C. Lucchese, S. Orlando, R. Perego, and R. Trasarti. A constraint-based querying system for exploratory pattern discovery. *Inf. Syst.*, 34(1) :3–27, 2009.

[3] B. Bringmann and A. Zimmermann. The chosen few : On identifying valuable patterns. In *12th IEEE Int. Conf. on Data Mining (ICDM-07)*, pages 63–72, 2007.

[4] D. Burdick, M. Calimlim, J. Flannick, J. Gehrke, and T. Yiu. Mafia : A performance study of mining maximal frequent itemsets. In *FIMI*, volume 90 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2003.

[5] T. Calders, C. Rigotti, and J-F. Boulicaut. A survey on condensed representations for frequent sets. In *Constraint-Based Mining and Inductive Databases*, volume 3848 of *LNAI*, pages 64–80, 2005.

[6] L. De Raedt, T. Guns, and S. Nijssen. Constraint Programming for Itemset Mining. In *ACM SIGKDD Int. Conf. KDD'08*, Las Vegas, Nevada, USA, 2008.

[7] L. De Raedt and A. Zimmermann. Constraint-based pattern set mining. In *7th SIAM Int. Conf. on Data Mining*. SIAM, 2007.

[8] C. Gervet. Interval Propagation to Reason about Sets : Definition and Implementation of a Practical Language. *Constraints*, 1(3) :191–244, 1997.

[9] A. Giacometti, E. Khanjari Miyaneh, P. Marcel, and A. Soulet. A framework for pattern-based global models. In *IDEAL'09*, pages 433–440, 2009.

[10] M. Khiari, P. Boizumault, and B. Crémilleux. Allier CSPs et motifs locaux pour la découverte de motifs sous contraintes n-aires. In *EGC*, volume RNTI-E-19, pages 199–210. Cépaduès-Éditions, 2010.

[11] M. Khiari, P. Boizumault, and B. Crémilleux. Combining CSP and constraint-based mining for pattern discovery. In *ICCSA'10*, volume 6017 of *LNCS*, pages 432–447, 2010.

[12] J. Kléma, S. Blachon, A. Soulet, B. Crémilleux, and O. Gandrillon. Constraint-based knowledge discovery from sage data. *In Silico Biology*, 8(0014), 2008.

[13] A. Knobbe, B. Crémilleux, J. Fürnkranz, and M. Scholz. From local patterns to global models : The lego approach to data mining. In *Int. Workshop LeGo co-located with ECML/PKDD'08*, pages 1–16, Antwerp, Belgium, 2008.

[14] A. Knobbe and E. Ho. Pattern teams. In *proceedings of the 10th ECML/PKDD'06*, volume 4213 of *LNAI*, pages 577–584, Berlin, Germany, 2006.

[15] N. Mamoulis and K. Stergiou. Algorithms for quantified constraint satisfaction problems. In *proceedings of the 10th Int. Conf. CP'04*, 2004.

[16] R. T. Ng, V. S. Lakshmanan, J. Han, and A. Pang. Exploratory mining and pruning optimizations of constrained associations rules. In *proceedings of ACM SIGMOD'98*, pages 13–24. ACM Press, 1998.

[17] S. Nijssen, T. Guns, and L. De Raedt. Correlated itemset mining in ROC space : a constraint programming approach. In *KDD'09*, pages 647–655, 2009.

[18] Balaji Padmanabhan and Alexander Tuzhilin. A belief-driven method for discovering unexpected patterns. In *KDD*, pages 94–100, 1998.

[19] A. Soulet, J. Klema, and B. Crémilleux. *Post-proceedings of the 5th Int. Workshop KDID'06*, volume 4747 of *LNCS*, pages 223–239. 2007.

[20] E. Suzuki. Undirected Discovery of Interesting Exception Rules. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 16(8) :1065–1086, 2002.

[21] L. Szathmary, A. Napoli, and P. Valtchev. Towards Rare Itemset Mining. In *Proc. of the 19th IEEE ICTAI '07*, volume 1, Patras, Greece, 2007.

[22] G. Verger and C. Bessière. Guiding search in QCSP$^+$ with back-propagation. In *CP'08*, volume 5202 of *Lecture Notes in Computer Science*, pages 175–189, 2008.

[23] X. Yin and J. Han. CPAR : classification based on predictive association rules. In *proceedings of the 2003 SIAM Int. Conf. on Data Mining (SDM'03)*, 2003.