Fouille de Motifs Fermés et Diversifiés Basée sur la Relaxation

Arnold Hien, Samir Loudni, Noureddine Aribi, Yahia Lebbah, Amine Laghzaoui, Abdelkader Ouali, Albrecht Zimmermann

Fouille de Motifs Fermés et Diversifiés Basée sur la Relaxation

Arnold Hien**, Samir Loudni***, Noureddine Aribi * Yahia Lebbah*, Amine Laghzaoui*
Abdelkader Ouali**, Albrecht Zimmermann**

*Université Oran1, Lab. LITIO, 31000 Oran, Algeria
**Normandie Univ., UNICAEN, CNRS – UMR GREYC, France
***TASC (LS2N-CNRS), IMT Atlantique, FR – 44307 Nantes, France

Résumé. Dans cet article, nous proposons une nouvelle approche basée sur la programmation par contraintes pour l'extraction de motifs fréquents fermés et diversifiés (Hien et al., 2020a). La diversité est contrôlée par une contrainte de seuil sur l'indice de Jaccard. Nous montrons que cette mesure n'a pas de propriété de monotonie, ce qui rend le processus d'extraction infaisable. Pour y remédier, nous proposons une nouvelle contrainte globale, CLOSEDDIVERSITY, qui exploite une relaxation anti-monotone de l'indice de Jaccard pour élaguer les motifs non diversifiés. Une seconde relaxation, basée sur une borne supérieure, est exploitée via une nouvelle heuristique de branchement.

Mots-clés : Fouille de motifs, Contrainte globale, Diversité, Jaccard, Relaxation

1 Introduction

Ces dernières années, la fouille de motifs a changé peu à peu de paradigme pour évoluer vers un modèle plus centré utilisateur. Il s'agit de prendre en compte les préférences de l'utilisateur afin de guider la recherche vers des motifs plus intéressants pour lui. Cela est rendu possible par l'introduction de mécanismes de feedback qui permettent à l'utilisateur de spécifier ses préférences sur les motifs extraits (Dzyuba et van Leeuwen, 2013). Un élément important de ce paradigme est la capacité à pouvoir présenter rapidement à l'utilisateur des motifs diversifiés. En effet, lorsque les motifs sont similaires, ou si l'extraction des motifs prend beaucoup de temps, l'utilisateur risque de se lasser et il devient alors difficile pour lui d'exprimer ses préférences. Nous proposons une nouvelle approche déclarative exploitant la programmation par contraintes (PPC) pour extraire efficacement des motifs fréquents, fermés et diversifiés. L'utilisation de la PPC est motivée par son caractère déclaratif, permettant de combiner plusieurs contraintes au même temps, et par la richesse du langage de contraintes qu'elle offre. De plus, la PPC permet une gestion générique des variables et des contraintes ainsi que l'utilisation d'algorithmes efficaces de filtrage, ce qui permet une construction itérative des motifs.

Les travaux précédents sur l'extraction de motifs diversifiés ont proposé l'utilisation d'un post-traitement sur les motifs déjà extraits (voir (Knobbe et Ho, 2006)). Van Leeuwen et Knobbe (2012) ont quant à eux proposé d'utiliser une approche heuristique. Bosc et al. (2018); Belfodil et al. (2019) ont au contraire introduit la diversité dans le processus d'extraction de

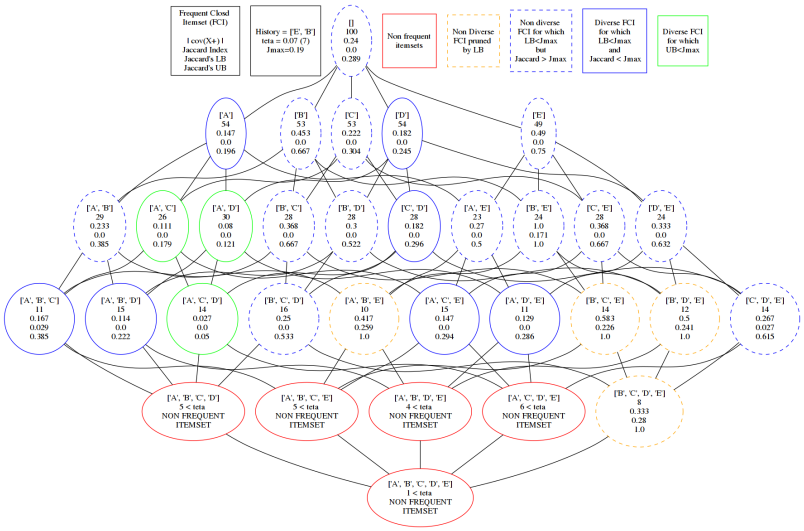


FIG. 1: Le treillis des motifs fréquents fermés associé à la base transactionnelle $\mathcal{D}$ de l'exemple 1.

motifs. Cette dernière approche nécessite d'ajouter des contraintes supplémentaires pour assurer la diversité en élaguant les motifs non diversifiés.

Dans cet article, nous utilisons la programmation par contraintes pour extraire efficacement les motifs fréquents fermés et diversifiés. La diversité est contrôlée par une contrainte de seuil sur l'indice de Jaccard. Nous montrons que cette mesure n'a pas de propriété de monotonie, ce qui rend le processus d'extraction infaisable. Pour y remédier, nous proposons une nouvelle contrainte globale, CLOSEDDIVERSITY, qui exploite une relaxation anti-monotone de l'indice de Jaccard pour élaguer les motifs non diversifiés. Une seconde relaxation, basée sur une borne supérieure, est exploitée via une nouvelle heuristique de branchement.

2 Préliminaires

2.1 Fouille d'itemset

Soit $\mathcal{I}$ un ensemble de n *items*, un *motif* P est un sous-ensemble non vide de $\mathcal{I}$. Une base transactionnelle $\mathcal{D}$ est un multi-ensemble de transactions sur $\mathcal{I}$, où chaque *transaction* t est un sous-ensemble de $\mathcal{I}$, i.e., $t \subseteq \mathcal{I}$. Un motif P *apparaît* dans une transaction t, ssi $P \subseteq t$. La *couverture* de P dans $\mathcal{D}$ est l'ensemble des transactions dans lesquelles il apparaît : $\boldsymbol{t}(P) = \{t \in \mathcal{D} \mid P \subseteq t\}$. Le *support* de P dans $\mathcal{D}$ est le cardinal de sa couverture : $sup(P)$ = $|\boldsymbol{t}(P)|$. Un motif P est dit *fréquent* si son support dépasse un seuil de fréquence minimal θ, $sup(P) \geq \theta$. La *clôture* d'un motif P, notée $Clos(P)$, est l'ensemble des items communs à toutes les transactions dans $\boldsymbol{t}(P)$: $Clos(P) = \{i \in \mathcal{I} \mid \forall t \in \boldsymbol{t}(P), i \in t\}$. Un motif P est dit *fermé* ssi $Clos(P) = P$.

Example 1 *La figure 1 montre le treillis de motifs fréquents fermés dérivés d'une base transactionnelle ayant* 5 *items et* 100 *transactions, avec* $\theta = 7$.

2.2 Mesure de Diversité

L'indice de Jaccard est une mesure de similarité classique sur les ensembles. Nous l'utilisons pour quantifier le chevauchement des couvertures entre deux motifs.

Definition 1 (Indice de Jaccard) *Soient deux motifs P et Q, l'indice de Jaccard mesure la proportion de chevauchement entre les couvertures des deux motifs :* $Jac(P,Q) = \frac{|\boldsymbol{t}(P) \cap \boldsymbol{t}(Q)|}{|\boldsymbol{t}(P) \cup \boldsymbol{t}(Q)|}$.

Un indice de Jaccard plus petit est synonyme d'une faible similarité en termes de couverture entre motifs et peut donc être utilisé comme mesure de diversité entre paires de motifs.

Definition 2 (Contrainte de Diversité/Jaccard) *Soient P et Q deux motifs. Étant donné la mesure Jac et un seuil de diversité J_{max}, on dit que P et Q sont diversifiés entre eux ssi $Jac(P,Q) \leq J_{max}$. Nous noterons cette contrainte c_{Jac}.*

Notre objectif est d'exploiter la contrainte de Jaccard durant la recherche pour élaguer les motifs non-diversifiés. Pour cela, nous maintenons un *historique* $\mathcal{H}$ de motifs extraits pendant la recherche et qui sont diversifiés entre eux. Les prochains motifs P extraits devront alors respecter la contrainte c_{Jac} par rapport à chaque motif $H \in \mathcal{H}$.

Definition 3 (k motifs fréquents et diversifiés) *Étant donné un historique $\mathcal{H} = \{H_1, \ldots, H_k\}$ de k motifs fréquents, fermés et diversifiés, la mesure Jac et un seuil de diversité J_{max}, le problème consiste à trouver de nouveaux motifs P tel que $\forall\, H \in \mathcal{H}$, $Jac(P,H) \leq J_{max}$.*

Example 2 *Le treillis de la figure 1 montre un ensemble de motifs fréquents fermés et diversifiés (représentés par des cercles bleus et verts) obtenus avec $J_{max} = 0.19$ et $\mathcal{H} = \{BE\}$. ACE est un motif fréquent, fermé et diversifié (i.e., $Jac(ACE, BE) = 0.147 < 0.19$).*

Proposition 1 *Soient P, Q et P' trois motifs avec $P \subset P'$. $Jac(P,Q)$ peut être plus petit, égal ou supérieur à $Jac(P',Q)$.*

Comme l'indique la proposition 1, la contrainte de Jaccard n'est ni monotone ni anti-monotone, ce qui implique un élagage limité lors de la recherche. Pour faire face à ce problème, nous proposons deux relaxations anti-monotones : (i) Une relaxation par la borne inférieure, permettant d'élaguer les motifs non-diversifiés lors de la recherche, (ii) une relaxation par la borne supérieure pour trouver les items menant vers des motifs diversifiés.

2.3 Programmation par contrainte

La programmation par contraintes (PPC) offre une approche générique pour modéliser les problèmes combinatoires. Un modèle PPC consiste en un ensemble de variables $X = \{x_1, \ldots, x_n\}$, un ensemble de domaines finis D pour chaque variable $x_i \in X$, et un ensemble de contraintes $\mathcal{C}$ sur X. Une contrainte $c \in \mathcal{C}$ est une relation entre différentes variables $X(c)$, qui précise les combinaisons possibles de valeurs pour ces variables. Une instanciation d'un sous-ensemble de variables $Y \subseteq X$ est une affectation de valeurs $v \in dom(x_i)$ à chaque variable x_i. Une solution est alors une instanciation de X satisfaisant toutes les contraintes $\mathcal{C}$. Pour la résolution, les solveurs utilisent des méthodes de recherche par retour-arrière pour explorer l'espace de recherche et instancier progressivement les variables. L'algorithme 1 donne le schéma général de résolution. À chaque nœud, *Recherche* sélectionne une

Algorithme 1 : Recherche(D)

```
1  In : X : variables de décision ; C : contraintes ;
2  InOut : D : domaines des variables ;
3  begin
4      D ← Filtrage(D, C)
5      if il existe x_i ∈ X t.q. dom(x_i) est vide then
6          return Echec
7      if il existe x_i ∈ X t.q. |dom(x_i)| > 1 then
8          Selectionner x_i ∈ X t.q. |dom(x_i)| > 1
9          forall v ∈ dom(x_i) do
10             Recherche(Dom ∪ {x_i ← {v}})
11     else
12         retourner la solution D
```

variable non instanciée (ligne 8) selon l'heuristique définie par l'utilisateur et l'instancie avec une valeur (ligne 9). Lorsqu'une instanciation ne respecte pas toutes les contraintes (lorsqu'un des domaines devient vide), un retour-arrière a lieu (ligne 5). On obtient une solution (ligne 12) lorsque tous les domaines $dom(x_i)$ ne contiennent que des singletons et que toutes les contraintes sont respectées. Afin d'accélérer la recherche, des *algorithmes de filtrages* sont utilisés. En effet, à chaque instanciation d'une variable à une valeur de son domaine, l'algorithme de filtrage réduit l'espace de recherche tout en garantissant une certaine propriété de consistance comme la *consistance de domaine*. La consistance de domaine garantit que pour chaque variable x_i d'une contrainte c ($x_i \in X(c)$) et pour chaque $v \in dom(x_i)$, il existe une instanciation $(x_i = v)$ qui satisfait c.

2.4 Modèle PPC pour la fouille de motifs fermés

Le premier modèle PPC utilisé pour la fouille de motifs fréquents et fermés a été proposé par De Raedt et al. (2008). Ce modèle est basé sur des contraintes réifiées (Apt, 2003) faisant intervenir les items et les transactions d'un jeu de données. Par la suite, Lazaar et al. (2016) ont proposé la première contrainte globale pour produire des motifs fréquents et fermés. Ils utilisent un vecteur x de variables booléennes $(x_1, \ldots, x_{|\mathcal{I}|})$ pour représenter les motifs. Chaque variable x_i représente la présence de l'item $i \in \mathcal{I}$ dans le motif. Nous utiliserons les notations suivantes : $x^+ = \{i \in \mathcal{I} \,|\, dom(x_i) = \{1\}\}$ l'ensemble des items présents, $x^- = \{i \in \mathcal{I} \,|\, dom(x_i) = \{0\}\}$ l'ensemble des items absents et $x^* = \{i \in \mathcal{I} \,|\, i \notin x^+ \cup x^-\}$.

Definition 4 (CLOSEDPATTERNS) *Soit x un vecteur de variables booléennes, θ un seuil de support minimum et $\mathcal{D}$ un jeu de données. La contrainte globale* CLOSEDPATTERNS$_{\mathcal{D},\theta}(x)$ *est vérifiée si et seulement si x^+ est à la fois fermé et fréquent.*

Definition 5 (Extension propre (Wang et al. (2003))) *Un motif non nul P est une extension propre de Q ssi $\boldsymbol{t}(P \cup Q) = \boldsymbol{t}(Q)$.*

Règles de filtrage. Lazaar et al. (2016) ont proposé trois règles de filtrage pour CLOSEDPATTERNS. La première règle permet d'étendre un motif x^+ avec un item i lorsque $x^+ \cup \{i\}$ est une extension propre de x^+ (voir Définition 5). Dans ce cas, on supprime la valeur 0 de $dom(x_i)$. La seconde règle permet de vérifier la fréquence du motif $x^+ \cup \{i\}$ et de supprimer la valeur 1

de $dom(x_i)$ si son support est inférieur au seuil θ. La troisième règle supprime la valeur 1 de $dom(x_i)$ lorsque $\boldsymbol{t}(x^+ \cup \{i\}) \subset \boldsymbol{t}(x^+ \cup \{j\})$, avec j un item absent ($j \in x^-$).

3 Extraction de Motifs Fréquents Fermés et Diversifiés

Cette section présente deux relaxations anti-monotones de l'indice de Jaccard : (i) Une relaxation par la borne inférieure pour élaguer les motifs non-diversifiés lors de la recherche, (ii) une relaxation par la borne supérieure pour trouver les items menant vers des motifs diversifiés. Les preuves des différentes propositions sont disponibles dans (Hien et al., 2020b).

3.1 Reformulation du problème

La proposition 1 établie que la contrainte de Jaccard n'est ni monotone ni anti-monotone. Nous proposons alors d'approximer la contrainte c_{Jac} par la collection de motifs solutions de sa relaxation : c^r_{Jac} : $Th(c_{Jac}) \subseteq Th(c^r_{Jac})$. Notre approche consiste à formuler une contrainte relâchée, ayant une propriété de monotonie, qui sera utilisée pour élaguer l'espace de recherche. Plus précisément, nous proposons d'exploiter des bornes inférieure et supérieure de l'indice de Jaccard dans le but de dériver une relaxation anti-monotone de c_{Jac}.

Definition 6 (Relaxation de l'indice de Jaccard) *Soit un historique $\mathcal{H} = \{H_1, \ldots, H_k\}$ de k motifs fréquents, fermés et diversifiés, un seuil de diversité J_{max}, une borne inférieure LB_J et une borne supérieure UB_J de l'indice de Jaccard, la relaxation du problème d'extraction de motifs diversifiés consiste à trouver les motifs candidats P tels que $\forall\, H \in \mathcal{H}$, $LB_J(P, H) \leq J_{max}$. La contrainte de Jaccard est satisfaite lorsque $\forall H \in \mathcal{H}, UB_J(P, H) \leq J_{max}$.*

3.2 Borne inférieure de l'indice de Jaccard

À partir de la définition 1, nous formulons une borne inférieure de l'indice de Jaccard qui minimise le chevauchement entre les couvertures des deux motifs et maximise la couverture propre de chaque motif.

Definition 7 (Couverture propre) *Soient P et Q deux motifs. La couverture propre de P par rapport à Q est définie par : $\boldsymbol{t}^{pr}_Q(P) = \boldsymbol{t}(P) \backslash \{\boldsymbol{t}(P) \cap \boldsymbol{t}(Q)\}$.*

Une borne inférieure LB de Jaccard minimise le numérateur et maximise le dénominateur du quotient donné dans la définition 1.

Proposition 2 (Borne inférieure LB) *Soit un motif $H \in \mathcal{H}$, P un motif partiel en cours de construction tel que $sup(P) \geq \theta$, et $\boldsymbol{t}^{pr}_H(P)$ la couverture propre de P par rapport à H. $LB_J(P, H) = \frac{\theta - |\boldsymbol{t}^{pr}_H(P)|}{|\boldsymbol{t}(P)| + |\boldsymbol{t}(H)| + |\boldsymbol{t}^{pr}_H(P)| - \theta}$ est une borne inférieure de $Jac(P, H)$.*

Cette borne inférieure de Jaccard nous permet de filtrer des motifs non diversifiés, c'est à dire ceux qui ont un LB_J supérieur à J_{max}. Ces motifs sont appelés des *témoins négatifs*.

Example 3 *Dans la figure 1, les motifs non diversifiés avec un LB_J supérieur à $J_{max} = 0.19$ sont représentés avec la couleur orange.*

Proposition 3 (Monotonie de LB_J) *Soit $H \in \mathcal{H}$ un motif. Pour tout motif P et Q tel que $P \subseteq Q$, alors nous avons $LB_J(P,H) \leq LB_J(Q,H)$.*

La propriété 3 est très importante car elle établit une condition nécessaire pour pouvoir filtrer les motifs non diversifiés (voir Section 3.4). En effet, lorsque $LB_J(P,H) > J_{max}$, alors aucun motif $Q \supseteq P$ ne pourra satisfaire la contrainte de Jaccard, ce qui rend la contrainte anti-monotone. On pourra donc filtrer le motif Q.

3.3 Borne supérieure de l'indice de Jaccard

En relâchant la contrainte de Jaccard et en approximant sa théorie $Th(c_{Jac})$ par $Th(c^r_{Jac})$ ($Th(c_{Jac}) \subseteq Th(c^r_{Jac})$), il est possible d'extraire des motifs P tel que $LB_J(P,H) < J_{max}$ alors que $Jac(P,H) > J_{max}$ (voir Figure 1). Pour remédier à cette situation (cas de faux positifs), nous définissons une borne supérieure UB de Jaccard qui évalue la satisfaction de la contrainte. Ainsi, les motifs P tels que $UB_J(P,H) \leq J_{max}$, $\forall H \in \mathcal{H}$ vont satisfaire la contrainte de Jaccard et seront appelés *témoins positifs*.

La borne UB a été construite en prenant la démarche inverse de celle de la borne inférieure : nous maintenons le numérateur $\boldsymbol{t}(H) \cap \boldsymbol{t}(P)$ inchangé et nous réduisons l'ensemble $\boldsymbol{t}^{pr}_H(P)$ afin de maximiser le dénominateur $\boldsymbol{t}(H) \cup \boldsymbol{t}(P)$. Ainsi, si l'intersection est supérieure ou égale à θ, les futurs motifs P' couvriront uniquement des transactions de l'intersection. Dans le cas contraire, le dénominateur devra contenir quelques transactions de $\boldsymbol{t}^{pr}_H(P)$ (exactement $\theta - |\boldsymbol{t}(H) \cap \boldsymbol{t}(P)|$ transactions).

Proposition 4 (Borne supérieure UB) *Étant donné un motif $H \in \mathcal{H}$, et un motif P tel que $sup(P) \geq \theta$. $UB_J(P,H) = \frac{|\boldsymbol{t}(H) \cap \boldsymbol{t}(P)|}{|\boldsymbol{t}^{pr}_P(H)| + \max\{\theta, |\boldsymbol{t}(H)| \cap |\boldsymbol{t}(P)|\}}$ est une borne supérieure de $Jac(P,H)$.*

Example 4 *Dans la figure 1, les motifs diversifiés avec un UB_J inférieur à $J_{max} = 0.19$ sont représentés avec la couleur verte.*

Notre borne supérieure UB peut être utilisée pour évaluer la contrainte de Jaccard pendant l'extraction des motifs. En effet, pendant l'étape d'énumération des motifs, lorsqu'un motif candidat P a une borne supérieure de Jaccard inférieure à J_{max} alors la contrainte c_{Jac} est satisfaite. Par ailleurs, comme nous le montrons dans la proposition 5, la borne supérieure UB est anti-monotone. De ce fait, tous les motifs $Q \supseteq P$ serons aussi diversifiés.

Proposition 5 (Anti-monotonie de UB_J) *Soit H un motif de l'historique $\mathcal{H}$. Pour tous les motifs P et Q, tels que $P \subseteq Q$, nous avons $UB_J(P,H) \geq UB_J(Q,H)$.*

3.4 Contrainte globale CLOSEDDIVERSITY

La contrainte globale CLOSEDDIVERSITY exploite la relaxation LB de l'indice de Jaccard pour extraire des motifs fréquents, fermés et diversifiés.

Definition 8 (CLOSEDDIVERSITY) *Soit x un vecteur de variables booléennes, $\mathcal{H}$ un historique de motifs fréquents, fermés et diversifiés (initialement vide), θ un seuil de support, J_{max} un seuil de diversité et $\mathcal{D}$ un jeu de données. La contrainte* CLOSEDDIVERSITY$_{\mathcal{D},\theta}(x, \mathcal{H}, J_{max})$ *est vérifiée si et seulement si : (1) x^+ est fermé ; (2) x^+ est fréquent, $sup(x^+) \geq \theta$; (3) x^+ est diversifié, $\forall H \in \mathcal{H}, LB_J(x^+, H) \leq J_{max}$.*

Algorithme 2 : Filtrage pour CLOSEDDIVERSITY

```
1  In : θ, J_max : seuils de fréquence et de diversité ; H : historique des solutions trouvées ;
2  InOut : x = {x_1 ... x_n} : variables booléennes ;
3  begin
4      if ( |t(x+)| < θ ∨ !PGrowth_LB(x+, H, J_max)) then return false;
5      foreach i ∈ x* do
6          if (|t(x+ ∪ {i})| < θ) then
7              dom(x_i) ← dom(x_i) − {1}; x⁻_Freq ← x⁻_Freq ∪ {i}; x* ← x* \ {i}; continue;
8          if (|t(x+ ∪ {i})| = |t(x+)|) then
9              dom(x_i) ← dom(x_i) − {0}; x+ ← x+ ∪ {i};    x* ← x* \ {i};
10         if (! PGrowth_LB(x+ ∪ {i}, H, J_max)) then
11             dom(x_i) ← dom(x_i) − {1}; x⁻_Div ← x⁻_Div ∪ {i}; x* ← x* \ {i}; continue;
12         foreach k ∈ (x⁻_Freq ∪ x⁻_Div) do
13             if (t(x+ ∪ {i}) ⊆ t(x+ ∪ {k})) then
14                 dom(x_i) ← dom(x_i) − {1}
15                 if k ∈ x⁻_Freq then x⁻_Freq ← x⁻_Freq ∪ {i};
16                 else x⁻_Div ← x⁻_Div ∪ {i};
17                 x* ← x* \ {i}; break;
18     return true;
19 Function PGrowth_LB(x, H, J_max) : Booléen
20     foreach H ∈ H do
21         if (LB_J(x, H) > J_max) then return false
22     return true
```

L'historique $\mathcal{H}$ est mis à jour de façon itérative en y ajoutant les motifs extraits avec CLOSEDDIVERSITY. La condition (3) est une condition nécessaire pour assurer la diversité des motifs. Nous montrerons en section 3.5 comment exploiter la borne supérieure UB pour garantir la satisfaction de la contrainte globale. CLOSEDDIVERSITY exploite les règles de filtrage de CLOSEDPATTERNS (voir Sect. 2.4) auxquels nous avons ajouté nos règles détaillées ci-dessous. On notera par x^-_{Freq} l'ensemble des variables non fréquentes, et par x^-_{Div} l'ensemble des variables filtrées par la règle LB.

Proposition 6 (Règles de filtrage) *Soit $\mathcal{H} = \{H_1, \ldots, H_k\}$ un historique de k motifs fréquents, fermés et diversifiés, x une instanciation partielle des variables et une variable non instanciée $i \in x^*$, la variable i sera filtrée si l'un des deux cas suivants est vérifié :*
1) si $\exists\, H \in \mathcal{H}$ s.t. $LB_J(x^+ \cup \{i\}, H) > J_{max}$, alors on filtre 1 du domaine $dom(x_i)$ de i.
2) si $\exists\, k \in x^-_{Div}$ s.t. $\boldsymbol{t}(x^+ \cup \{i\}) \subseteq \boldsymbol{t}(x^+ \cup \{k\})$, alors $LB_J(x^+ \cup \{i\}, H) > LB_J(x^+ \cup \{k\}, H) > J_{max}$ et on filtre 1 du domaine $dom(x_i)$ de i. Ces deux cas indiquent que le motif $x^+ \cup \{i\}$ n'est pas diversifié et sera donc filtré.

Algorithme. Le propagateur de CLOSEDDIVERSITY prend en paramètre les variables x, le support minimum θ, le seuil de diversité J_{max} et l'historique $\mathcal{H}$ initialement vide. Il commence par vérifier si le motif partiel x^+ est fréquent en comparant sa couverture à θ. Il teste également sa diversité avec la fonction $\mathcal{P}Growth_{LB}$. Si le motif n'est pas fréquent ou s'il n'est pas diversifié, alors la contrainte globale n'est pas respectée et la branche explorée est abandonnée (ligne 4). Par la suite, l'algorithme 2 applique les règles de filtrage de CLOSEDPATTERNS (voir Section 2.4) auxquelles on ajoute une règle de filtrage avec LB_J. Ainsi, $\forall H \in \mathcal{H}$, la

valeur de $LB_J(x^+ \cup \{i\}, H)$ est évaluée avec la fonction $\mathcal{P}Growth_{LB}(x^+ \cup \{i\}, \mathcal{H}, J_{max})$. S'il existe un motif H tel que $LB_J(x^+ \cup \{i\}, H) > J_{max}$ (ligne 21), alors la variable x_i est filtrée (la valeur 1 est supprimée de son domaine) car le motif $x^+ \cup \{i\}$ ne conduira pas vers un motif diversifié (ligne 11). On met à jour x^-_{Div} et x^*, et on répète l'opération sur les autres variables non instanciées. De même, lorsque la couverture d'un motif $x^+ \cup \{i\}$ est incluse dans la couverture du motif $x^+ \cup \{k\}$, tel que $k \in (x^-_{Freq} \cup x^-_{Div})$ (lignes 12-17), alors la variable i est filtrée.

Proposition 7 (Consistance de domaine et complexité) *L'algorithme 2 supprime toutes les valeurs inconsistantes avec une complexité temporelle en* $\mathcal{O}(n^2 \times m)$.

3.5 Motifs témoins et Fréquences estimées

Fréquences estimées. La fréquence d'un motif peut être calculée en faisant l'intersection des couvertures des items qui le constituent puis calculer leur cardinalité : $sup(x^+) = |\cap_{i \in x^+} \boldsymbol{t}(i)|$. Pour limiter les nombreuses et coûteuses opérations d'intersection qui correspondent à des *OU logiques*, nous proposons d'estimer la fréquence de chaque item $i \in \mathcal{I}$ en fonction des items du motif x^+. Cette estimation, notée $eSup_{\mathcal{D}}(i, x^+)$, constitue une *borne inférieure* de $|\boldsymbol{t}(x^+ \cup \{i\})|$. De ce fait, lorsque $eSup_{\mathcal{D}}(i, x^+) \geq \theta$ alors $|\boldsymbol{t}(x^+ \cup \{i\})| \geq \theta$. Ainsi, la fréquence du motif n'est calculée que lorsque $eSup_{\mathcal{D}}(i, x^+) < \theta$, ce qui nous permet des gains de performance non négligeables. Par ailleurs, avec les fréquences estimées, nous proposons une nouvelle heuristique de choix de variables notée MINCOV. Elle consiste, pour une itération donnée, à étendre le motif partiel courant avec la variable qui, à l'itération précédente, avait la plus petite fréquence estimée. En effet, ces variables sont les plus susceptibles d'activer rapidement une règle de filtrage (voir Algorithme 2) et donc de réduire l'espace de recherche.
Témoins positifs. Durant la recherche, nous calculons de façon incrémentale $UB(x^+ \cup \{i\}, H)$ de chaque extension du motif partiel x^+. Ainsi, avec la propriété d'anti-monotonie de UB_J (voir Proposition 5), si $\forall H \in \mathcal{H}, UB(x^+ \cup \{i\}, H) < J_{max}$ alors tous les sur-motifs de $x^+ \cup \{i\}$ satisferont la contrainte de Jaccard. Cette propriété nous permet de déduire une nouvelle heuristique de choix de variable que nous notons FIRSTWITCOV et qui consiste à étendre le motif partiel courant avec la variable i qui a un UB_J inférieur au seuil J_{max}.

4 Résultats expérimentaux

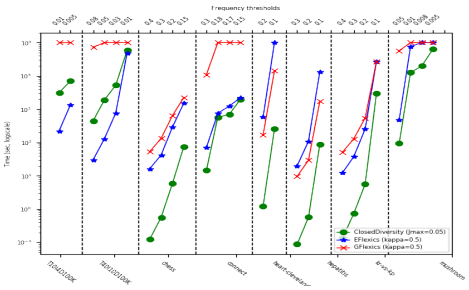
Nous avons évalué notre méthode en nous intéressant aux trois points suivants : (1) le temps d'exécution et le nombre de motifs générés : pour cela, nous comparons CLOSEDDIV avec CLOSEDP et FLEXICS de (Dzyuba et al., 2017) ; (2) la qualité des motifs de CLOSEDDIV par rapport à ceux générés avec CLOSEDP et FLEXICS ; (3) la qualité de nos bornes LB/UB : nous avons mesuré la distance qu'il y a entre ces deux bornes et l'indice de Jaccard. Nous avons utilisé les jeux de données UCI (fimi.ua.ac.be/data) et avons choisi des jeux de données de différentes tailles et densités. Certains jeux de données, comme HEPATITIS et CHESS sont très denses (resp. 50% et 49%). D'autres au contraire sont très peu denses, comme T10I4D100K et RETAIL (resp. 1% and 0.06%). Les expérimentations ont été menées sur une machine avec un processeur AMD Opteron 6174 de 2.2 GHz ayant 256 Go de RAM et avec une limite de temps d'exécution de 24 heures. Nous avons sélectionné pour chaque dataset des seuils de fréquence pour avoir différents nombres de motifs fermés et fréquents ($|Th(c)| \leq 15000$,

Dataset $\lvert\mathcal{I}\rvert \times \lvert\mathcal{T}\rvert$ $\rho(\%)$	$\theta(\%)$	#Motifs		Temps (s)		#Nœuds	
		(1)	(2)	(1)	(2)	(2)	(2)
CHESS	20	22,808,625	**96**	2838.30	**5.87**	45,617,249	**436**
75 × 3196	15	50,723,131	**393**	5666.03	**75.40**	101,446,261	**1,855**
49.33%	10	OOM	**4,204**	OOM	**3825.29**	OOM	**18,270**
HEPATITIS	30	83,048	**12**	9.64	**0.09**	166,095	**29**
68 × 137	20	410,318	**57**	42.00	**0.57**	820,635	**162**
50.00%	10	1,827,264	**2,270**	**169.59**	**76.91**	3,654,527	**5,256**
KR-VS-KP	30	5,219,727	**17**	682.94	**0.74**	10,439,453	**82**
73 × 3196	20	21,676,719	**96**	2100.79	**5.64**	43,353,437	**448**
49.32%	10	OOM	**4,120**	OOM	**3035.49**	OOM	**17,861**
CONNECT	30	460,357	**18**	1666.14	**14.81**	920,713	**77**
129 × 67557	18	2,005,476	**197**	5975.44	**573.66**	4,010,951	**900**
33.33%	15	3,254,780	**509**	9534.07	**1989.35**	6,509,559	**2,188**
HEART-CLEVELAND	10	12,774,456	**3,496**	1308.63	**257.39**	25,548,911	**7,977**
95 × 296	8	23,278,687	**12,842**	**2278.97**	2527.38	46,557,373	**28,221**
47.37%	6	43,588,346	58,240	**4126.84**	46163.06	87,176,691	124,705
SPLICE1	10	1,606	**422**	**6.55**	25.25	3,211	**843**
287 × 3190	5	31,441	**8,781**	**117.15**	5616.47	62,881	**17,594**
20.91%	2	589,588	-	**1179.55**	-	1,179,175	-
MUSHROOM	5	8,977	**727**	**10.02**	60.70	17,953	**1,704**
112 × 8124	1	40,368	**12,139**	**34.76**	12532.95	80,735	**25,154**
18.75%	0.5	62,334	**27,768**	**50.05**	64829.06	124,667	**56,873**
T40I10D100K	8	138	127	**75.91**	447.20	275	253
942 × 100000	5	317	288	**331.47**	1561.34	633	575
4.20%	1	65,237	7,402	**5574.31**	58613.88	130,473	14,887
PUMSB	40	-	**4**	-	**57.33**	-	**16**
2113 × 49046	30	-	**15**	-	**267.72**	-	**64**
3.50%	20	-	**52**	-	**852.39**	-	**250**
T10I4D100K	5	11	11	**1.73**	6.31	21	21
870 × 100000	1	386	**361**	**434.25**	3125.06	771	722
1.16%	0.5	1,074	**617**	**881.31**	7078.90	2,147	**1,257**
BMS1	0.15	1,426	**609**	**11362.71**	68312.38	2,851	**1,220**
497 × 59602	0.14	1,683	**668**	**11464.93**	68049.00	3,365	**1,339**
0.51%	0.12	2,374	**823**	**13255.79**	79704.88	4,747	**1,651**
RETAIL	5	17	**13**	**10.74**	33.44	33	25
16470 × 88162	1	160	**111**	**297.21**	1625.73	319	227
0.06%	0.4	832	**528**	**6073.53**	31353.23	1,663	**1,093**

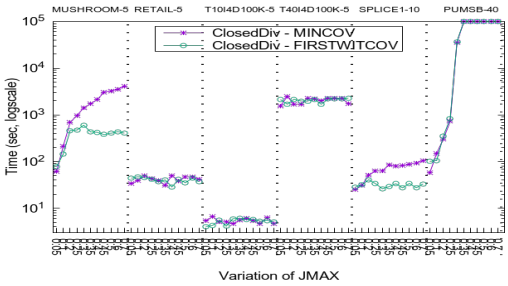
TAB. 1: CLOSEDDIV ($J_{max} = 0.05$) vs CLOSEDP. Pour les colonnes #Motifs and #Noeuds, les valeurs en gras indiquent une réduction dépassant 20% du nombre total de motifs et nœuds." − " s'affiche lorsque la limite de temps est dépassée. OOM : Mémoire insuffisante. (1) : CLOSEDP (2) : CLOSEDDIV

$30000 \leq |Th(c)| \leq 10^6$, and $|Th(c)| > 10^6$). La seule exception concerne les datasets très volumineux et peu denses RETAIL et PUMSB, où le nombre de solutions est petit. Nous avons utilisé CLOSEDPATTERNS comme base de référence pour déterminer les seuils appropriés utilisés par CLOSEDDIV. Pour évaluer la qualité de nos motifs, nous proposons de calculer l'ECR (Exclusive Coverage Ratio) qui mesure le taux moyen de couverture propre des motifs extraits : $ECR(P_1, ..., P_k) = avg_{1 \leq i \leq k}(\frac{sup(P_i) - |\boldsymbol{t}(P_i) \cap \bigcup_{j \neq i} \boldsymbol{t}(P_j)|}{sup(P_i)})$.

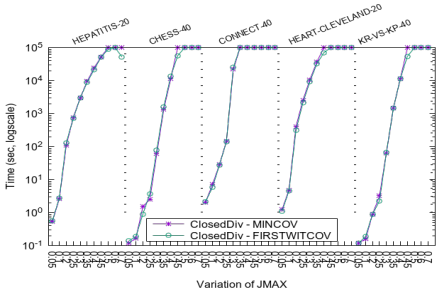
(a) Comparaisons entre CLOSEDDIV, CLOSEDP et FLEXICS. La table 1 compare CLOSEDDIV et CLOSEDP pour différentes valeurs de θ. Pour CLOSEDDIV, nous avons pris un seuil de diversité J_{max} égal à 0.05 et avons choisi MINCOV comme heuristique de choix de variables. Comme nous pouvons le constater, CLOSEDDIV permet de réduire considérablement le nombre de motifs, surtout avec les jeux de données denses et pour des valeurs faibles de θ. Ainsi, pour CHESS, CLOSEDDIV permet une réduction de près de 99% du nombre de motifs par rapport à CLOSEDP (de $\sim 50 \cdot 10^6$ à 393 motifs) pour $\theta = 15\%$. Ce résultat peut s'expliquer par la densité des jeux de données qui induisent de nombreuses redondances dans la



(a) Temps d'exécution de MINCOV et FLEXICS



(b) Jeux de données non denses.



(c) Jeux de données denses.

Jeux de données	$\theta(\%)$	Exclusive Coverage Ratio (ECR)		
		CLOSEDP	CLOSEDDIV–MINCOV	EFLEXICS
T40I10D100K	8	1.6E-01	**4.4E-01**	1.29E-01
	1	1.9E-01	**9.2E-01**	2.90E-01
SPLICE1	10	4.0E-02	**1.6E-01**	2.98E-02
	5	4.0E-02	**2.5E-01**	1.79E-01
CONNECT SPLICE1	18	0.00	**1.0E-01**	4.31E-02
	10	4E-02	**1.6E-01**	2.98E-02
	5	4E-2	**2.5E-01**	1.79E-01
CONNECT	18	1.25E-6	**1.0E-01**	4.31E-02
	15	0.00	**1.7E-01**	4.16E-02
MUSHROOM	5	3.3E-4	**5.6E-01**	2.63E-01
	1	3.73E-4	**4.3E-01**	4.2E-01
HEPATITIS	10	0.00	**2.5E-01**	1.0E-01
HEPATITIS	10	1.50E-3	**2.5E-01**	1.0E-01
T10I4D100K	1	5.9E-01	**9.3E-01**	5.72E-01
	0.5	4.8E-01	**9.6E-01**	4.63E-01

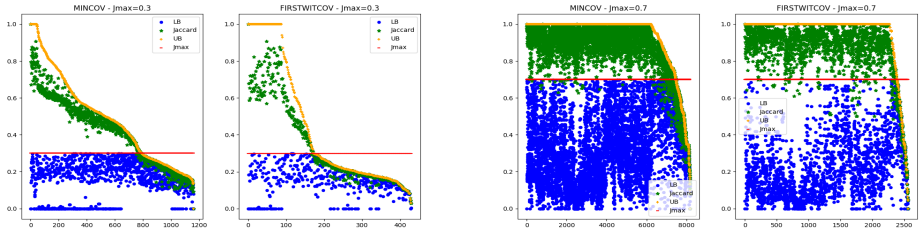
(d) ECR.

FIG. 2: Analyse des temps d'exécutions (MINCOV vs FIRSTWITCOV et CLOSEDDIV vs FLEXICS) et ECR

couverture des motifs fermés. Avec ces jeux de données, le temps d'exécution de CLOSEDDIV est également plus court car le filtrage des motifs redondants permet de réduire considérablement l'espace de recherche. Avec les jeux de données moins denses, la réduction du nombre de motifs par CLOSEDDIV est plus faible. En effet, avec ces jeux de données, il y a peu de motifs fermés et donc moins de redondances. Par conséquent, les calculs de LB_J pénalisent grandement CLOSEDDIV qui devient plus lent comparé à CLOSEDP.

La figure 2a montre une comparaison des temps d'exécution entre CLOSEDDIV et FLEXICS. FLEXICS est un outil permettant de faire de l'échantillonnage de motifs. Pour cela, il partitionne l'espace de recherche en cellules avec des contraintes XOR puis utilise un oracle pour faire un tirage pondéré des motifs dans différentes cellules. Il se décline en deux versions, EFLEXICS qui utilise ECLAT comme oracle et GFLEXICS qui utilise plutôt CP4IM (De Raedt et al., 2008). La partition de l'espace en cellules permet d'introduire une diversité dans les données, d'où notre intérêt pour cette approche. D'une part, CLOSEDDIV est plus rapide que GFLEXICS avec plus d'un ordre de grandeur. D'autre part, EFLEXICS est plus rapide que GFLEXICS, et notre approche est presque toujours classée en premier, illustrant son utilité pour extraire des motifs diversifiés de manière *anytime*.

(b) Impact de la variation de J_{max}. Nous avons étudié les deux heuristiques de choix de variables MINCOV et FIRSTWITCOV pour différentes valeurs du seuil J_{max}. Dans les figures 2b et 2c, nous pouvons constater que, de façon générale, le temps d'exécution augmente avec le



(a) SPLICE1 ($\theta = 10\%$, $J_{max} = 0.3$) (b) MUSHROOM ($\theta = 5\%$, $J_{max} = 0.7$)

FIG. 3: Analyse qualitative de LB et UB

seuil J_{max}. En effet, avec un seuil de diversité plus grand, CLOSEDDIV génère plus de motifs, ce qui induit un sur-coût dans le calcul de LB et UB. Nous pouvons constater néanmoins que lorsque J_{max} devient assez grand, CLOSEDDIV filtre beaucoup moins (voir figure 3). Par ailleurs, nous constatons que les deux heuristiques MINCOV et FIRSTWITCOV ont presque les même performances, FIRSTWITCOV étant meilleur sur certaines instances. De plus, on peut constater (voir l'annexe complémentaire (Hien et al., 2020b)) que FIRSTWITCOV permet de générer des motifs de meilleure qualité grâce à sa capacité de guider la recherche vers des motifs témoins positifs.

(c) Analyse qualitative des bornes LB et UB. La figure 3a montre l'évolution des valeurs de LB_J, Jac et UB_J des motifs générés par MINCOV et FIRSTWITCOV pour les jeux de données SPLICE1 et MUSHROOM. Pour les différentes courbes, les motifs ont été triés par ordre décroissant de leur UB_J. Nous constatons ainsi que la valeur de LB est toujours en-dessous de J_{max}. Concernant la valeur de UB_J, on peut constater qu'elle est toujours très proche de celle du Jaccard, ce qui dénote une bonne relaxation pour notre borne supérieure. Par ailleurs, nous pouvons noter que l'heuristique FIRSTWITCOV permet de générer rapidement des motifs de bonne qualité (avec un UB inférieur à J_{max}). Ces résultats démontrent l'intérêt de notre borne supérieure UB_J et de l'heuristique FIRSTWITCOV à générer des motifs plus diversifiés.

(d) Analyse qualitative des motifs. La figure 2d compare les ECR de CLOSEDDIV, CLOSEDP et FLEXICS, un ECR important traduit une plus grande diversité dans la couverture des motifs. Pour pallier au grand nombre de motifs générés par CLOSEDP qui complique l'évaluation de l'ECR, nous avons décidé d'effectuer le calcul sur des échantillons de $k = 10$ motifs, et répéter l'évaluation 100 fois afin d'évaluer un grand nombre de motifs. Nous reportons une moyenne des ECR calculés. CLOSEDDIV conduit clairement à des solutions avec une plus grande diversité entre motifs. Ceci est indicatif de motifs dont la couverture est (approximativement) mutuellement exclusive.

5 Conclusions

Dans ce papier, nous avons proposé une contrainte globale qui exploite deux relaxations LB/UB (anti-)monotones de l'indice de Jaccard pour l'extraction de motifs fréquents, fermés et diversifiés. La diversité est contrôlée par une contrainte de seuil mesurant la similarité des

occurrences des motifs. Nos expérimentations ont montré que notre approche permet de réduire de façon significative le nombre de motifs par rapport à ceux générés par la contrainte globale CLOSEDPATTERNS. Par ailleurs, les ensembles de motifs générés sont plus diversifiés.

Références

Apt, K. (2003). *Principles of Constraint Programming*. USA : Cambridge University Press.

Belfodil, A., A. Belfodil, A. Bendimerad, P. Lamarre, C. Robardet, M. Kaytoue, et M. Plantevit (2019). Fssd-a fast and efficient algorithm for subgroup set discovery. In *Proceedings of DSAA*, pp. 91–99.

Bosc, G., J.-F. Boulicaut, C. Raïssi, et M. Kaytoue (2018). Anytime discovery of a diverse set of patterns with monte carlo tree search. *Data mining and knowledge discovery 32*(3), 604–650.

De Raedt, L., T. Guns, et S. Nijssen (2008). Constraint programming for itemset mining. In *14th ACM SIGKDD*, pp. 204–212.

Dzyuba, V. et M. van Leeuwen (2013). Interactive discovery of interesting subgroup sets. In *International Symposium on Intelligent Data Analysis*, pp. 150–161. Springer.

Dzyuba, V., M. van Leeuwen, et L. De Raedt (2017). Flexible constrained sampling with guarantees for pattern mining. *Data Mining and Knowledge Discovery 31*(5), 1266–1293.

Hien, A., S. Loudni, N. Aribi, Y. Lebbah, M. Laghzaoui, A. Ouali, et A. Zimmermann (2020a). A relaxation-based approach for mining diverse closed patterns. In *Proceedings of ECML PKDD 2020*, Volume 12457, pp. 36–54. Springer.

Hien, A., S. Loudni, N. Aribi, Y. Lebbah, M. Laghzaoui, A. Ouali, et A. Zimmermann (June 2020b). Supplementary Material : https://github.com/lobnury/ClosedDiversity.

Knobbe, A. J. et E. K. Ho (2006). Pattern teams. In *Proceedings of ECML-PKDD*, pp. 577–584. Springer.

Lazaar, N., Y. Lebbah, S. Loudni, M. Maamar, V. Lemière, C. Bessiere, et P. Boizumault (2016). A global constraint for closed frequent pattern mining. In *Proceedings of the 22nd CP*, pp. 333–349.

Van Leeuwen, M. et A. Knobbe (2012). Diverse subgroup set discovery. *Data Mining and Knowledge Discovery 25*(2), 208–242.

Wang, J., J. Han, et J. Pei (2003). CLOSET+ : searching for the best strategies for mining frequent closed itemsets. In *Proceedings of the Ninth KDD*, pp. 236–245. ACM.

Summary

In this paper, we use constraint programming (well suited to user-centric mining due to its rich constraint language) to efficiently mine a diverse set of closed patterns. Diversity is controlled through a threshold on the Jaccard similarity measure. We show that the Jaccard measure has no monotonicity property, which prevents usual pruning techniques and makes classical pattern mining unworkable. This is why we propose antimonotonic lower and upper bound relaxations, which allow effective pruning, with an efficient branching rule, boosting the whole search process. We show experimentally that our approach significantly reduces the number of patterns and is very efficient in terms of running times, particularly on dense datasets.

Keywords: Pattern mining, Global Constraint, Diversity, Jaccard, Relaxations