Fouille de motifs diversifiés : une approche basée sur la relaxation et l'échantillonnage

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

Fouille de motifs diversifiés : une approche basée sur la relaxation et l'échantillonnage

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

*TASC (LS2N-CNRS), IMT Atlantique, FR – 44307 Nantes, France
prenom.nom@imt-atlantique.fr
**Université Oran1, Lab. LITIO, 31000 Oran, Algeria
prenom.nom@edu.univ-oran1.dz
***Normandie Univ., UNICAEN, CNRS – UMR GREYC, France
prenom.nom@unicaen.fr

Résumé. Dans cet article, nous proposons une approche basée sur la programmation par contraintes pour l'extraction de 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 et 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. Enfin, nous montrons comment intégrer notre contrainte pour l'échantillonnage de motifs diversifiés. Les résultats expérimentaux sur plusieurs jeux de données démontrent l'efficacité en temps de calcul et en termes de diversité de notre approche par rapport aux approches concurrentes.

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 retours 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 (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 motifs. Cette dernière approche nécessite d'ajouter des contraintes supplémentaires pour assurer la diversité en élaguant les motifs non diversifiés. D'autres approches basées sur l'échantillonnage de motifs ont été récemment proposées (Dzyuba et al., 2017; Boley et al., 2012; Hasan et Zaki, 2009) pour répondre aux besoins des utilisateurs par rapport à la rapidité de l'extraction, au contrôle de la taille des sorties obtenues et à l'obtention d'un ensemble de motifs ayant une bonne diversité. Notons ici que la diversité découle principalement de la nature aléatoire de ce type de méthodes.

Dans cet article, nous proposons une nouvelle approche déclarative basée sur la Programmation par Contraintes pour extraire efficacement les motifs fréquents fermés et diversifiés. L'approche proposée, axée sur la diversité et l'échantillonnage, imposent à la fois une diversité contrainte par une mesure de dissimilarité entre motifs et une diversité méthodologique par un échantillonnage aléatoire. Le coeur de la méthode s'appuie sur le maintien d'un historique des motifs divers et retenus comme résultat. Dans ce travail, 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 difficile. 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. Enfin, nous montrons comment notre contrainte globale peut être intégrée dans FLEXICS, une méthode d'échantillonnage de motifs ensemblistes basée sur le formalisme logique SAT.

Ce papier est organisé comme suit. La section 2 rappelle les préliminaires. La section 3 présente une formalisation de la problématique de la fouille de motifs diversifiés. La section 4 introduit deux relaxations anti-monotones de l'indice de Jaccard et présente notre contrainte globale CLOSEDDIVERSITY avec son algorithme de filtrage. La section 5 montre comment intégrer notre contrainte pour l'échantillonnage de motifs diversifiés. La section6 discute le positionnement de nos contributions par rapport à l'état de l'art. Dans la section 7, nous présentons un ensemble complet d'expériences. La section8 présente une synthèse des différentes contributions et dresse quelques perspectives de recherches futures.

2 Préliminaires

Dans cette section, nous présentons les différents préliminaires portant sur la fouille de motifs (cf. section 2.1, et la programmation par contraintes (cf. section2.2. Ensuite, nous détaillons en sections 2.3 et 2.4 les approches PPC pour l'extraction et l'échantillonnage de motifs.

2.1 Fouille d'itemsets

Étant donné une base de données $\mathcal{D}$, un langage $\mathcal{L}$ définissant des sous-ensembles de données et un prédicat c (appelé *contrainte*) qui détermine si un élément $\phi \in \mathcal{L}$, la tâche est de

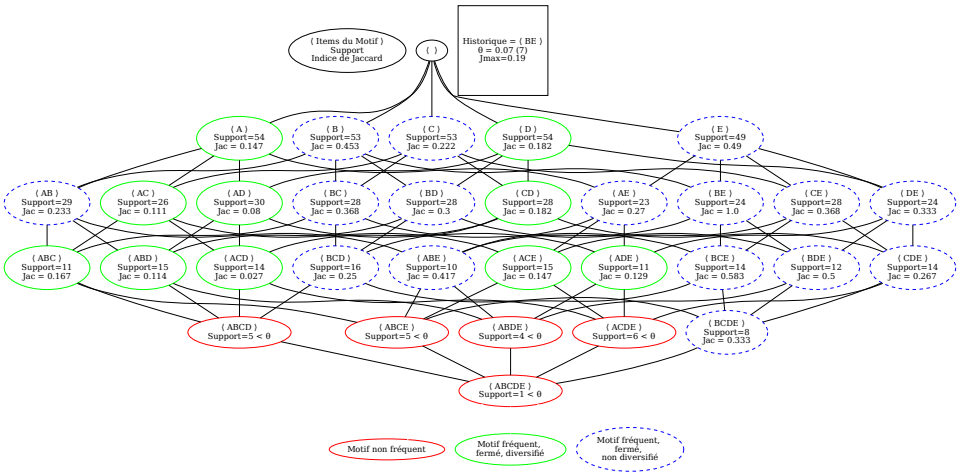


FIG. 1 – Treillis des motifs fermés et fréquents associé à la base transactionnelle $\mathcal{D}$ de l'exemple 1.

trouver la théorie $\mathcal{T}(\mathcal{L}, \mathcal{D}, c) = \{\phi \in \mathcal{L} \mid c(\mathcal{D}, \phi)\ is\ true\}$ (Nijssen et Zimmermann, 2014). Une tâche bien connue en fouille de données est *la fouille de motifs fréquents*.

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 prédicat bien connu est un seuil sur le support des motifs, *le support minimal* θ. La tâche consiste alors à calculer les *motifs fréquents* (Agrawal et Srikant, 1994; Borgelt, 2012) : $\{P \in \mathcal{L}_{\mathcal{I}} \mid sup(P) \geq \theta\}$.

D'autres contraintes sur les motifs individuels pour réduire la redondance entre les motifs, comme la contrainte de *clôture* (Pasquier et al., 1999), ont été proposées. 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 $Clos(P) = P$. 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 Programmation par contraintes

La programmation par contraintes (Hoeve et Katriel, 2006) (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 $v \in dom(y_i)$ à chaque variable $y_i \in Y$. Une solution est alors une instanciation de X satisfaisant toutes les

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          Sélectionner 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
```

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 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 pratique, une contrainte est équipée d'un ou plusieurs algorithmes de filtrage, nommés propagateurs. Ainsi, à chaque instanciation d'une variable à une valeur de son domaine, le propagateur d'une contrainte c réduit les domaines de $X(c)$ en filtrant les valeurs localement inconsistantes tout en garantissant un certain niveau 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.

2.3 Modèle PPC pour la fouille de motifs fermés

Plusieurs propositions ont étudié les relations entre la fouille de motifs et la programmation par contraintes (CP) pour revisiter les tâches de fouille de données de manière déclarative et générique (De Raedt et al., 2008; Lazaar et al., 2016; Schaus et al., 2017; Belaid et al., 2019; Vernerey et al., 2022).

Le premier modèle PPC utilisé pour la fouille de motifs fréquents 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 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 $x^* = \{i \in \mathcal{I} \,|\, i \notin x^+ \cup x^-\}$.

Définition 1 (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.

Définition 2 (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 2). 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 ($j \in x^-$).

2.4 FLEXICS : échantillonnage de motifs basé sur le formalisme SAT

L'échantillonnage de motifs est une autre approche qui a été récemment proposée (Boley et al., 2011, 2012; Hasan et Zaki, 2009; Bendimerad et al., 2020; Bhuiyan et Hasan, 2016) : au lieu d'énumérer tous les motifs, ces derniers sont échantillonnés un par un, selon une distribution de probabilité proportionnelle à une mesure d'intérêt donnée. Dans (Aggarwal et Han, 2014) (Chapitre 8), il est précisé qu'il faudrait rechercher un ensemble de motifs d'intérêt réduit (facilement interprétable) et non redondant (avec une grande diversité). C'est précisément l'objectif des approches d'échantillonnage en fouille de motifs. Les avantages attendus comprennent (Dzyuba et al., 2017) : 1) la *flexibilité*, permettant l'utilisation potentielle d'un large éventail de mesures de qualité ; 2) l'exploration *anytime* des motifs, où un ensemble représentatif croissant de motifs peut être généré et inspecté à tout moment ; 3) la *diversité*, puisque les ensembles de motifs générés sont échantillonnés indépendamment à partir de différentes régions de l'espace de solutions.

FLEXICS (Dzyuba et al., 2017) est une technique d'échantillonnage de motifs. Elle repose sur l'algorithme WEIGHTGEN (Chakraborty et al., 2014) pour l'échantillonnage de solutions satisfaisant une formule logique. L'idée de base est de partitionner l'espace des motifs satisfaisant une contrainte q en ajoutant récursivement à q m contraintes `XOR` aléatoires, puis de faire un tirage pondéré dans cet espace réduit. Ces contraintes portent sur la présence des items dans le motif. Elles sont de la forme $\bigotimes b_i.x_i = b_0$, b_0 étant le bit de parité et b_i l'apparition ou non de la variable d'item x_i dans la contrainte. Les m contraintes `XOR` identifient une cellule appartenant à un partitionnement de l'espace global des motifs en 2^m cellules. Pour échantillonner un motif, WEIGHTGEN commence par estimer la valeur de m jusqu'à avoir la bonne taille de cellule, génère aléatoirement les m contraintes `XOR` à ajouter, puis tire aléatoirement un motif parmi ceux satisfaisant toutes les contraintes du problème selon une pondération définie par une mesure d'intérêt. Un oracle est utilisé pour sélectionner les motifs dans la cellule : (i) un algorithme générique, GFLEXICS, qui s'appuie sur CP4IM (De Raedt et al., 2008), un cadre générique pour l'extraction de motifs sous contraintes basé sur la PPC, (ii) un algorithme spécialisé, EFLEXICS, qui utilise une version étendue de ECLAT. Pour la pondération des motifs, FLEXICS permet un choix entre différentes fonctions comme la fréquence ou la pureté. Par ailleurs, la méthode d'échantillonnage ajoute une diversité (implicite) entre les motifs échantillonnés. En effet, la partition de l'espace de recherche permet d'obtenir des cellules de tailles réduites dans lesquelles il est plus facile de réaliser un tirage des motifs. L'utilisation des contraintes `XOR` permet alors à FLEXICS de tirer des motifs qui décrivent différentes régions du

jeu de données, étant données que les différentes cellules sont différentes les unes des autres. Cependant, les cellules exploitées par FLEXICS pour tirer ses motifs ne sont pas toujours mutuellement exclusives. En effet, pour une paire de cellules $(Cell_i, Cell_j)$, il est possible d'avoir $Cell_i \cap Cell_j \neq \emptyset$. De ce fait, il est possible d'avoir des redondances dans l'ensemble de motifs final. Nous montrons à la section 5 comment contrôler de manière explicite la diversité des motifs échantillonnés.

3 Fouille de motifs diversifiés et relaxations

Dans cette section, nous présentons la problématique de la fouille de motifs diversifiés. Nous introduirons en section 3.1 la mesure de Jaccard pour l'évaluation de la redondance entre paires de motifs, suivie d'une formalisation de la contrainte de Jaccard maximum entre paires de motifs. Nous étendons en section 3.2 l'évaluation de la redondance sur tous les itemsets d'un ensemble de motifs. Nous montrons en section 3.3 que la mesure de Jaccard n'est ni monotone, ni anti-monotone et formalisons le problème d'extraction de motifs diversifiés permettant d'exploiter deux relaxations de la mesure de Jaccard.

3.1 Indice de Jaccard et contrainte de diversité entre paires de motifs

L'indice de Jaccard est une mesure de similarité classique sur les ensembles (Tan et al., 2005). Nous l'utilisons comme mesure d'évaluation de la redondance entre paires de motifs, i.e. pour quantifier le chevauchement des couvertures entre deux motifs.

Définition 3 (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. Pour limiter la redondance entre paires de motifs, la diversité peut être imposée avec une limite supérieure J_{max} sur l'indice de Jaccard par paire. Ceci définit la contrainte Jaccard maximum par paire suivante :

Définition 4 (Contrainte de Jaccard maximum entre paires de motifs) *Soient P et Q deux motifs. Étant donné la mesure Jac et un seuil de redondance maximum 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}.*

3.2 Évaluation de la diversité sur un ensemble de motifs

L'évaluation de la redondance entre les paires de motifs est intéressante car elle permet d'identifier de façon précise les sous-ensembles de motifs ayant apporté beaucoup de redondance à l'ensemble. Cependant, cette approche procède de façon locale sur chaque motif et ne permet pas d'évaluer la redondance de façon globale sur tout l'ensemble de motifs. Toutefois, en utilisant une fonction d'agrégation, il est possible d'avoir une vision globale de la redondance dans l'ensemble. Comme fonction d'agrégation, on peut utiliser la somme $\sum$, le maximum $\max$, le minimum $\min$ ou la moyenne Agv. L'indice de Jaccard est ici utilisé avec un seuil J_{max} afin de limiter la redondance entre les différentes paires de motifs. Par ailleurs,

pour incorporer la contrainte de Jaccard maximum par paire c_{Jac} durant l'énumération de l'espace de motifs, nous imposons que toutes les paires de motifs dans l'ensemble de résultats ne doivent pas dépasser une similarité de Jaccard maximum en termes de couverture. Pour cela, nous proposons de maintenir un *historique* $\mathcal{H}$ des différents motifs extraits afin de garantir que tous les nouveaux motifs soient diversifiés par rapport à ceux déjà extraits. Étant donné que l'indice de Jaccard évalue la similarité entre des paires de motifs, nous proposons d'agréger les valeurs de diversité des différentes paires de motifs.

Une première approche consiste à contraindre la somme de toutes les évaluations par paires entre le prochain motif à extraire P et tous les motifs déjà présents dans $\mathcal{H}$ dans une grande contrainte globale. L'inconvénient de cette approche est que de nombreuses évaluations entre paires de motifs de l'indice de Jaccard devraient être connues avant que la contrainte puisse propager un changement sur les autres évaluations. Une approche alternative consisterait à contraindre chaque similarité entre paires de motifs individuellement. Dans cet article, nous contraignons la plus grande similarité par paire (en utilisant l'opérateur $\max$) à être inférieure à un certain seuil J_{max} : si la plus grande similarité entre paires de motifs doit être inférieure au seuil, alors toutes les similarités doivent être inférieures à J_{max}. Ceci peut être formellement défini comme suit :

Définition 5 (Extraction d'un ensemble de motifs 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}, la tâche consiste à extraire un nouveau motif P tel que toutes les évaluations par paires entre P et les motifs dans $\mathcal{H}$ doivent être plus petites que J_{max}, c'est à dire, $\forall\, H \in \mathcal{H}$, $Jac(P, H) \leq J_{max}$, ce qui est équivalent à $max(Jac(P, H))_{H \in \mathcal{H}} \leq J_{max}$.*

Ainsi, pour être diversifiés, les motifs P devront avoir des indices de Jaccard par rapport aux motifs H de $\mathcal{H}$ inférieurs à J_{max}. 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 $\mathcal{H} = \{BE\}$. Les arêtes entre les nœuds représentent les relations de spécialisation ou de généralisation entre les différents nœuds. Ainsi, le motif BCD, qui est une une spécialisation des motifs BC, BD et CD, a un support de 16, un indice de jaccard de 0.25. Ainsi, ACE est un motif fréquent, fermé et diversifié (i.e., $Jac(ACE, BE) = 0.147 < 0.19$).

3.3 Monotonicité de lindice de Jaccard

Bien que l'(anti-)monotonicité s'aligne bien avec la relation de spécialisation entre motifs (Mitchell, 1982) lors de l'énumération de l'espace de recherche, commune à la plupart des algorithmes de fouille de motifs, pousser des contraintes qui ne sont pas construites sur une mesure (anti-)monotonique est souvent plus difficile et moins efficace. C'est le cas de la mesure de Jaccard, qui n'est ni monotone croissante, ni monotone décroissante (i.e anti-monotone) (voir la proposition 1).

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

Pour illustrer la proposition 1, considérons les motifs C, CD et BCD de la figure 1. Le motif C n'est pas diversifié par rapport à BE car $Jac(C, BE) = 0.222 \geq J_{max}$. Or, le motif CD est diversifié car $Jac(CD, BE) = 0.182 \leq J_{max}$. Par conséquent, la contrainte c_{Jac}

n'est donc pas monotone. Elle n'est pas non plus anti-monotone ($Jac(A, BE) < J_{max}$ mais $Jac(AE, BE) > J_{max}$).

La raison de l'indétermination de la monotonie de la mesure du Jaccard est que la spécialisation peut perdre des éléments à cause de l'intersection ainsi que de la différence des couvertures des motifs. Cependant, nous proposons dans ce papier 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.

3.4 Relaxation de l'indice de Jaccard

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 : $c^r_{Jac} : Th(c_{Jac}) \subseteq Th(c^r_{Jac})$. L'attrait de cette approche est que nous pouvons bénéficier de propriétés de monotonie appropriées de la contrainte relaxée qui peuvent être efficacement exploitées pour la réduction de l'espace de recherche. Par ailleurs, une telle approche nous permet de préserver les solutions (Bayardo, 2004) de c_{Jac} puisque l'élagage induit par la relaxation ne supprime pas les motifs satisfaisant c_{Jac}. Dans ce qui suit, nous formalisons cette intuition en définissant deux relaxations exploitant une *borne supérieure* $\overline{c}_{Jac}$ et une *borne inférieure* $\underline{c}_{Jac}$ de l'indice de Jaccard.

Définition 6 (Relaxation de l'indice de Jaccard) *Soit un historique $\mathcal{H} = \{H_1, \ldots, H_k\}$ de k motifs 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 $\forall H \in \mathcal{H}$, $LB_J(P, H) \leq J_{max}$. Lorsque $UB_J(P, H) \leq J_{max}$ pour tout $H \in \mathcal{H}$, alors, la contrainte de Jaccard est satisfaite.*

4 Relaxations anti-monotones de l'indice de Jaccard

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 (cf. section 4.1), (ii) une relaxation par la borne supérieure pour trouver les items menant vers des motifs diversifiés (cf. section 4.2). Nous montrons en section 4.3 comment exploiter la monotonie de la borne inférieure pour réduire efficacement l'espace de recherche au sein d'une nouvelle contrainte globale CLOSEDDIVERSITY. Enfin, nous proposons en section 4.2 une nouvelle heuristique de choix de variables exploitant l'anti-monotonie de la borne supérieure. Cette heuristique permet d'accélérer la recherche de motifs diversifiés. Les preuves des différentes propositions sont disponibles en annexe.

4.1 Borne inférieure de l'indice de Jaccard

À partir de la définition 3, nous formulons une borne inférieure de l'indice de Jaccard qui minimise le chevauchement entre les couvertures des deux motifs. Nous commençons par définir la notion de couverture résiduelle d'un motif.

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

Nous pouvons dériver une borne inférieure de l'indice de Jaccard entre un motif H et la spécialisation du motif P en considérant le cas où la couverture de l'intersection de ces deux motifs soit la plus petite possible, tandis que la couverture résiduelle de chaque motif reste la plus grande possible. La valeur de Jaccard la plus petite est celle qui réduit le numérateur de l'indice de Jaccard à 0, ce qui n'est pas toujours possible avec une contrainte de fréquence. Le dénominateur, d'autre part, se compose de $|\boldsymbol{t}(H)|$ (qui ne change pas) et d'une partie de la couverture de P qui ne couvre pas H, i.e. $\boldsymbol{t}_H^{pr}(P)$.

Le lemme 1 introduit une propriété sur la couverture résiduelle d'un motif qui sera utilisée dans les différentes preuves à venir.

Lemme 1 (COUVERTURE RÉSIDUELLE)
Considérons un motif H de l'historique $\mathcal{H}$. Soit P et Q deux motifs. Si $P \supseteq Q$, alors $|\boldsymbol{t}_H^{pr}(P)| \leq |\boldsymbol{t}_H^{pr}(Q)|$.

Comme indiqué précédemment, notre intuition est d'utiliser le Jaccard d'un motif pour dériver des propriétés à partir de la valeur minimum de Jaccard de toutes ses spécialisations. Dans (Hien et al., 2020a), nous avons présenté une première borne LB_J^{old} de Jaccard. Dans ce qui suit, nous proposons une nouvelle borne LB_J plus resserrée.

Proposition 2 (Nouvelle borne inférieure LB) *Considérant un motif H de l'historique $\mathcal{H}$. Soit P un motif rencontré pendant la recherche tel que $sup(P) \geq \theta$, et $\boldsymbol{t}_H^{pr}(P)$ la couverture résiduelle de P par rapport à H.*

$$LB_J(H,P) = \begin{cases} \frac{\theta - |\boldsymbol{t}_H^{pr}(P)|}{|\boldsymbol{t}(H)| + |\boldsymbol{t}_H^{pr}(P)|} & si\ (\boldsymbol{t}_H^{pr}(P) < \theta) \\ 0 & si\ non \end{cases}$$

est une borne inférieure de $Jac(H,P)$.

Proposition 3 *Soit $LB_J^{old}(P,H) = \frac{\theta - |\boldsymbol{t}_H^{pr}(P)|}{|\boldsymbol{t}(P)| + |\boldsymbol{t}(H)| + |\boldsymbol{t}_H^{pr}(P)| - \theta}$ la borne inférieure de Jaccard introduite dans (Hien et al., 2020a). La borne $LB_J(P,H)$ est plus resserrée que $LB_J^{old}(P,H)$.*

Proposition 4 (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 $LB_J(P,H) \leq LB_J(Q,H)$.*

Proposition 5 (Filtrage utilisant LB_J)
Soit H un motif de l'historique $\mathcal{H}$. Pour tout motif P, si $LB_J(H,P) > J_{max}$, alors $Jac(H,P) > J_{max}$ et P n'est pas diversifié. Par ailleurs, toute spécialisation $Q \supseteq P$ n'est pas non plus diversifiée.

La proposition 5 établit un résultat important permettant de définir une condition de filtrage exploitant la monotonie de la borne inférieure. Ainsi, en combinant une fonction monotone croissante et un seuil maximum rend la contrainte elle-même anti-monotone. Effet, si $LB_J(H,P) > J_{max}$, alors aucun motif $Q \supseteq P$ ne pourra satisfaire la contrainte de Jaccard maximum. De ce fait, nous pouvons filtrer le motif Q. Notre contrainte globale exploite cette règle de filtrage dans le but de réduire l'espace de recherche (voir la section 4.3).

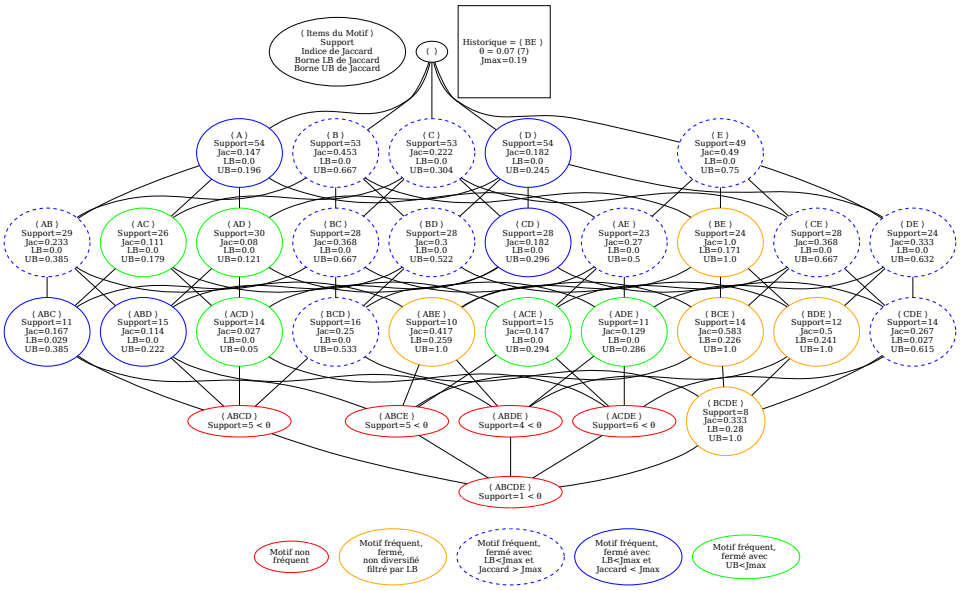


FIG. 2 – Treillis des motifs fermés et fréquents ($\theta = 7$)

La figure 2 illustre l'exploitation de la borne inférieure sur le même jeu de données que celui de la figure 1. Pour chaque motif, nous reportons la valeur de la borne inférieur LB de l'indice de Jaccard. Les motifs ayant un indice de Jaccard inférieur à $J_{max} = 0.19$ sont représentés par des nœuds de couleur bleu. Les motifs avec un indice de Jaccard supérieur à J_{max} et une borne LB inférieure à J_{max} sont représentés par des nœuds en pointillés bleu. Enfin, les motifs ayant une borne LB supérieure à J_{max} sont représentés en orange et correspondent aux motifs filtrés. Par exemple, le motif C n'est pas filtré comme dans la figure 1, ce qui permet de générer le motif diversifié CD. Par contre le motif BDE ainsi que toutes ses spécialisations (i.e. le motif $BCDE$ seront filtrés car non diversifiés, i.e. $LB_J(BDE, BE) = 0.241 > 0.19$.

4.2 Borne supérieure de l'indice de Jaccard

Comme notre relaxation approxime la théorie de la contrainte de Jaccard c_{Jac}, c'est-à-dire $Th(c_{Jac}) \subseteq Th(c^r_{Jac})$, on pourrait alors avoir un motif P (un *faux positif*) avec $LB_J(P, H) < J_{max}$ alors que $Jac(P, H) > J_{max}$ (voir les motifs marqués en bleu avec des traits pointillés dans la figure 2). Pour prendre en considération ce cas, nous proposons une borne supérieure sur l'indice de Jaccard pour évaluer la satisfaction de la contrainte de Jaccard maximum. Les motifs P tels que $UB_J(P, H) \leq J_{max}, \forall\, H \in \mathcal{H}$ sont alors appelés des *témoins positifs* (Kifer et al., 2006). Cette borne supérieure est alors utilisée dans une nouvelle heuristique de choix de variables afin de guider la recherche vers des ensemble de motifs diversifiés (voir la section 4.4).

Pour dériver une borne supérieure de l'indice de Jaccard, nous utilisons le raisonnement inverse de celui de la borne inférieure : le Jaccard le plus élevé possible sera atteint si $\boldsymbol{t}(H) \cap$

$\boldsymbol{t}(P)$ reste inchangé et que l'ensemble $\boldsymbol{t}_H^{pr}(P)$ est réduit au maximum (en dessous du seuil de fréquence minimum) afin de maximiser le dénominateur $\boldsymbol{t}(H) \cup \boldsymbol{t}(P)$. Si l'intersection est supérieure ou égale à θ, dans le pire des cas (conduisant au Jaccard le plus élevé), tout futur motif P' couvrira uniquement les transactions de l'intersection. Dans le cas contraire, le dénominateur devra contenir peu de transactions de $\boldsymbol{t}_H^{pr}(P)$, c'est-à-dire exactement $\theta - |\boldsymbol{t}(H) \cap \boldsymbol{t}(P)|$ transactions.

Proposition 6 (Borne supérieure)
Soit H un motif de l'historique $\mathcal{H}$, et P un motif tel que $sup(P) \geq \theta$.

$$UB_J(H, P) = \frac{|\boldsymbol{t}(H) \cap \boldsymbol{t}(P)|}{|\boldsymbol{t}_P^{pr}(H)| + \max\{\theta, |\boldsymbol{t}(H) \cap \boldsymbol{t}(P)|\}}$$

est une borne supérieure de $Jac(H, P)$.

Tous les motifs fréquents, fermés et diversifiés avec une valeur de UB_J inférieure à J_{max} sont marqués avec des lignes en vert dans la figure 2. Par ailleurs, le motif AC est un témoin positif car $UB_J(AC, BE) = 0.179 < J_{max}$.

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, si la borne supérieure est *monotone décroissante* (c'est à dire anti-monotone), alors P pourra être un témoin positif pour toutes ses spécialisations par rapport à la contrainte de Jaccard maximum.

Proposition 7 (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 $UB_J(P, H) \geq UB_J(Q, H)$.*

4.3 Contrainte globale CLOSEDDIVERSITY

Dans cette section, nous montrons comment exploiter notre relaxation LB au sein de de la contrainte globale ClosedPatterns afin d'extraire des motifs fréquents fermés et diversifiés. Nous proposons une nouvelle contrainte globale notée CLOSEDDIVERSITY qui tire partie de la monotonie de la borne inférieure afin de réduire l'espace de recherche.

Définition 8 (CLOSEDDIVERSITY) *Soit x un vecteur de variables booléennes représentant les items, $\mathcal{H}$ un historique de motifs fréquents, fermés et diversifiés (initialement vide), θ un seuil de support, J_{max} un seuil de diversité maximum 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, $\forall H \in \mathcal{H}, LB_J(x^+, H) \leq J_{max}$.*

Initialement vide, 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. En effet, il est possible d'avoir $LB_J(x^+, H) \leq J_{max}$ alors que $Jac(x^+, H) > J_{max}$.

Le propagateur de CLOSEDDIVERSITY exploite les règles de filtrage de CLOSEDPATTERNS (voir Sect. 2.3) auxquelles 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.

Algorithme 2 : Filtrage de la contrainte globale CLOSEDDIVERSITY

In : θ, J_{max} : seuil de fréquence et de diversité ; $\mathcal{H}$: historique des solutions trouvées pendant la recherche ;
InOut : $X = \{X_1 \dots X_n\}$: variables booléennes d'items ;
Début
  Si ($|\boldsymbol{t}(X^+)| < \theta \vee !\mathcal{P}Growth_{LB}(X^+, \mathcal{H}, J_{max})$) **alors retourner false**;
  Pour chaque $i \in X^*$ **faire**
    Si ($|\boldsymbol{t}(X^+ \cup \{i\})| < \theta$) **alors**
      $dom(X_i) \leftarrow dom(X_i) - \{1\}$;
      $X^-_{Freq} \leftarrow X^-_{Freq} \cup \{i\}$; $X^* \leftarrow X^* \setminus \{i\}$;
      continuer ;
    Si ($|\boldsymbol{t}(X^+ \cup \{i\})| = |\boldsymbol{t}(x^+)|$) **alors**
      $dom(X_i) \leftarrow dom(X_i) - \{0\}$;
      $X^+ \leftarrow X^+ \cup \{i\}$; $X^* \leftarrow X^* \setminus \{i\}$;
    Si ($!\,\mathcal{P}Growth_{LB}(X^+ \cup \{i\}, \mathcal{H}, J_{max})$) **alors**
      $dom(X_i) \leftarrow dom(X_i) - \{1\}$;
      $X^-_{Div} \leftarrow X^-_{Div} \cup \{i\}$; $X^* \leftarrow X^* \setminus \{i\}$;
      continuer ;
    Pour chaque $k \in (X^-_{Freq} \cup X^-_{Div})$ **faire**
      Si ($\boldsymbol{t}(X^+ \cup \{i\}) \subseteq \boldsymbol{t}(X^+ \cup \{k\})$) **alors**
        $dom(X_i) \leftarrow dom(X_i) - \{1\}$
        si $k \in X^-_{Freq}$ **alors**
          $X^-_{Freq} \leftarrow X^-_{Freq} \cup \{i\}$;
        sinon
          $X^-_{Div} \leftarrow X^-_{Div} \cup \{i\}$;
        $X^* \leftarrow X^* \setminus \{i\}$;
        arrêter ;
  retourner true;
Fonction $\mathcal{P}Growth_{LB}(x, \mathcal{H}, J_{max})$ *: Booléen* **Pour chaque** $H \in \mathcal{H}$ **faire**
  Si ($LB_J(H, x) > J_{max}$) **alors**
    retourner false
retourner true

Proposition 8 (Règles de filtrage) *Soit $\mathcal{H} = \{H_1, \dots, H_k\}$ un historique de k motifs fréquents, fermés et diversifiés, x une instanciation partielle des variables et une variable non instanciée $i \in x^*$, $x^+ \cup \{i\}$ ne pourra pas être étendu à un motif diversifié si l'une de deux conditions ci-dessous est vérifiée :*
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.*

Algorithme 2. 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 2.3) 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 29), 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 16). On met à jour x^-_{Div} et x^*, et on répète l'opération sur les autres variables non instanciées (items libres). 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 17-25), alors la variable i est filtrée.

Proposition 9 (Consistance de domaine et complexité) *L'algorithme 2 supprime toutes les valeurs inconsistantes avec une complexité temporelle en* $\mathcal{O}(n^2 \times m)$.

4.4 Motifs témoins et fréquences estimées

La contrainte globale CLOSEDDIVERSITY est exploitée au niveau de la propagation du solveur de contraintes, c'est-à-dire à la ligne 4 de l'algorithme 1. Dans cette section, nous exploitons les notions de motifs témoins et de *fréquences estimées* pour guider la recherche vers des motifs diversifiés. Pour cela, nous proposons deux nouvelles heuristiques de choix de variables décrites dans les lignes 7 à 10 de l'algorithme 1. Enfin, nous montrons comment exploiter la contrainte de Jaccard maximum ainsi que notre relaxation pour réduire le nombre de faux positifs dans les motifs diversifiés.

Fréquences estimées et heuristique de choix de variables MINCOV (Han et al., 2000). 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. Cette nouvelle heuristique de choix de variable sera notée MINCOV.

Témoins positifs et heuristique de choix de variables FIRSTWITCOV. 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 7), 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}.

5 Échantillonnage de Motifs Diversifiés

Dans cette section, nous montrons comment exploiter la contrainte globale CLOSEDDIVERSITY au sein d'un outil d'échantillonnage afin de contrôler de manière explicite la diversité des motifs échantillonnés. Pour cela, nous proposons d'utiliser WEIGHTGEN (Chakraborty et al., 2014) pour échantillonner des motifs en utilisant comme oracle pour l'énumération des motifs, la contrainte globale CLOSEDDIVERSITY. Le cadre obtenu est alors similaire à celui de FLEXICS (Dzyuba et al., 2017). Notre nouvel outils, noté SDIVJAX pour « **S***ample* **Div***erse pattern with* **Ja***ccard and* **X***or constraints* » c'est-à-dire « *Échantillonnage de motifs diversifiés avec l'indice de Jaccard et des contraintes* `XOR` », tire parti de la diversité implicite apportée par les contraintes `XOR` de WEIGHTGEN et la diversité explicite de la contrainte globale CLOSEDDIVERSITY.

Nous commencerons par présenter dans la section 5.1 l'algorithme WEIGHTGEN et les principales adaptations que nous avons apporté à cet algorithme. Ensuite, dans la section 5.2, nous détaillerons ces adaptations. En particulier, nous présenterons deux nouveaux oracles qui exploitent CLOSEDDIVERSITY pour l'énumération de motifs au niveau de chaque cellule et qui vont servir à l'étape d'échantillonnage Enfin, nous décrirons dans la section 5.3 l'algorithme de filtrage des contraintes `XOR` que nous avons implanté.

5.1 Adaptations de l'algorithme WEIGHTGEN

Basiquement, l'algorithme WEIGHTGEN réduit l'espace des solutions aléatoirement à l'aide de contraintes `XOR`, puis fait un tirage pondéré dans cet espace. Ces contraintes portent sur la présence des items dans le motif. Plus précisément, il partitionne l'espace de recherche des motifs en des cellules puis tire dans différentes cellules des motifs proportionnellement à leur poids. Pour obtenir la "bonne" taille de cellule [1] désirée, lors de l'étape d'initialisation, l'algorithme commence par estimer le nombre de contraintes `XOR` qu'il faut ajouter en moyenne durant la phase d'échantillonnage (voir la ligne 4, algorithme 3). Cette taille est calculée en additionnant les poids de toutes les solutions du problème (voir la ligne 15, algorithme 3). Finalement, vient la phase d'échantillonnage (voir la ligne 10, algorithme 3) où, pour chaque motif échantillonné, les opérations ci-dessous sont répétées :

1. générer les contraintes `XOR` estimées lors de la phase d'initialisation (ligne 8) ;
2. énumérer les solutions dans le sous-espace défini par les contraintes `XOR` (ligne 14) ;
3. calculer le poids de ces solutions (ligne 15) ;
4. Si le poids total ne convient pas, cellule trop grande ou trop petite, ajouter ou enlever une contrainte puis revenir à l'étape 2 ;
5. tirer un motif aléatoirement suivant la pondération choisie.

Modifications de WEIGHTGEN. Les modifications que nous avons apportés concernent principalement la prise en compte explicite de la diversité dans l'étape d'énumération des solutions dans chaque cellule (cf. la fonction CLOSEDXORSOLVE) et le contrôle du nombre de contraintes `XOR` dans le cas où la cellule devient trop petite (cf. les lignes 20-21, algorithme 3).

1. L'algorithme contient un paramètre κ, nommé "sampling error tolerance", qui correspond à la tolérance sur la taille de la cellule et qui permet d'avoir une solution plus rapidement au détriment du respect de la bonne distribution.

Algorithme 3 : Updated WEIGHTGEN

```
1  Entrée : 𝒟, θ, J_max, w : fonction de poids, κ : tolérance d'erreur, k : nombre de tirages
2  Entrée/Sortie : X = {X_1 ... X_n} : variables booléennes ;
3  Début
4      N_XOR ← EstimationXOR();
5      loThresh ≅ (1 + κ)/κ², hiThresh ≅ (1 + κ)³/κ² ;
6      i ← 1, ℋ ← ∅ ;
7      Pour (i ≤ k) faire
8          InitXORs ← {RandomXOR() × N_XOR} ;
9          ▷ Retourner une solution tirée aléatoirement
10         P ← GENERER (κ, [loThresh, hiThresh], InitXORs) ;
11         si (P ≠ NUL) alors ℋ ← ℋ ∪ P ;
12     retourner ℋ ;
13 Fonction GENERER(κ, [lT, hT], XORs) : Motif
14     Solutions ← CLOSEDXORSOLVE(X, θ, J_max, XORs)
15     PoidsCellule ← Σ_{s∈Solutions} w(s)
16     si (PoidsCellule ∈ [lT, hT]) alors
17         retourner ECHANTILLONNER(Solutions, w)
18     si (PoidsCellule > hT) alors
19         retourner GENERER(κ, [loThresh, hiThresh], XORs ∪ RandomXOR()
20     Si (|XORs| > 0)
21         retourner GENERER(κ, [loThresh, hiThresh], XORs − RandomXOR())
22     Sinon
23         retourner NUL
```

Ainsi, lorsque le nombre de contraintes XOR ne permet pas d'obtenir suffisamment de motifs (ligne 21), nous autorisons le retrait d'une contrainte afin de moins contraindre l'espace de recherche. Dans le cas où il n'y a pas de solutions et que le nombre de contraintes XOR est nulle (ligne 22), alors nous pouvons arrêter la recherche. En effet, ce cas correspond à une situation où il n'y a plus aucune solution diversifiée. Il est alors inutile de poursuivre la résolution.

5.2 Oracles de sélection des motifs diversifiés

L'exploitation de WEIGHTGEN pour l'échantillonnage des motifs permet à FLEXICS de tirer des motifs proportionnellement à leur poids (évalué grâce à une mesure de qualité). Par ailleurs, avec les contraintes XOR, il est possible d'obtenir une diversité implicite entre les motifs échantillonnés car ceux-ci sont extraits dans différentes zone de l'espace de recherche. Toutefois, cette diversité n'est pas garantie. En effet, étant donné le caractère aléatoire des contraintes XOR générées et le fait que chaque tirage soit indépendant, les cellules obtenues par l'application de ces contraintes peuvent se chevaucher. Il en résulte alors la possibilité de tirer deux motifs identiques ou très similaires dans différentes cellules. Pour ce faire, nous proposons deux oracles d'énumération des motifs permettant d'ajouter explicitement une contrainte de diversité : SDIVJAX-1 et SDIVJAX-2. Par ailleurs, en contraignant l'espace de recherche de chaque cellule par l'ajout de la contrainte de diversité, nous espérons réduire le temps d'exploration au niveau de chaque cellule.

(a) SDIVJAX-1. L'objectif de cet oracle est d'extraire un ensemble de motifs diversités localement à chaque cellule, puis échantillonner un motif parmi cet ensemble. Pour cela, dans chaque cellule, la contrainte globale CLOSEDDIVERSITY est utilisée pour énumérer les motifs fréquents fermés et diversifiés qui satisfont les contraintes XOR générées par WEIGHTGEN. Il en résulte alors un historique $\mathcal{H}_{local}$ de motifs diversifiés entre eux. Un motif est par la suite tiré de cet historique local et l'opération se répète jusqu'à l'obtention du nombre k de motifs demandés. Étant donné le caractère local de la diversité des motifs échantillonnés, l'historique $\mathcal{H}_{local}$ exploité par CLOSEDDIVERSITY est réinitialisé à l'ensemble vide, à chaque nouveau tirage de WEIGHTGEN. Dans chaque cellule i, CLOSEDDIVERSITY énumère donc un ensemble de motifs $\mathcal{H}^i_{local} = \{H^1_i, H^2_i, \ldots, H^n_i\}$ tel que :

$$\forall j, \ell \in [1, n] \wedge (j > \ell) : LB_J(H^j_i, H^\ell_i) \leq J_{max}$$

Un motif est alors tiré de cet ensemble proportionnellement à son poids et le processus se poursuit jusqu'à l'extraction complète du nombre k d'échantillons demandés. Cet ensemble $\mathcal{H}$ de k motifs se présente alors comme suit :$\mathcal{H} = \{H_1, H_2, \ldots, H_k\}$ avec $H_j \in \mathcal{H}^j_{local}$ où $\mathcal{H}^j_{local}$ est l'historique local de motifs extraits par CLOSEDDIVERSITY à la cellule j. Cette approche nous permet de bénéficier des performances de CLOSEDDIVERSITY et ainsi d'accélérer l'échantillonnage de chaque motif au sein de chaque cellule. Par ailleurs, le filtrage effectué avec la borne LB_J nous permet d'obtenir des cellules de tailles plus réduites par rapport à celles de EFLEXICS et GFLEXICS. De ce fait, moins de contraintes XOR sont nécessaires pour échantillonner chaque motif, ce qui permet un gain supplémentaire en termes de performance. Toutefois, en réinitialisant l'historique des motifs $\mathcal{H}_{local}$ à chaque nouvelle cellule, la méthode ne permet pas de garantir une diversité globale des motifs échantillonnés, c'est à dire ceux de $\mathcal{H}$. Nous proposons de remédier à cette situation avec l'approche SDIVJAX-2.

(b) SDIVJAX-2. Avec SDIVJAX-2, nous maintenons un historique $\mathcal{H}_{global}$ des différents motifs échantillonnés à travers les différentes cellules. Cet historique est mis à jour après chaque tirage de chaque motif et est utilisé pour assurer la diversité entre motifs déjà échantillonnés et les motifs solutions des prochaines cellules. Pour garantir la diversité entre les différents motifs échantillonnés, nous utilisons la contrainte globale CLOSEDDIVERSITY qui prend en paramètre un historique $\mathcal{H}_{global}$ initialement vide. Contrairement à l'approche SDIVJAX-1, $\mathcal{H}_{global}$ n'est pas mis à jour avec tous les motifs découverts dans une même cellule. Notons par $\mathcal{H}^{i-1}_{global}$ l'historique obtenu après l'échantillonnage des $(i-1)$ premières cellules.

Au départ, $\mathcal{H}^1_{global} = \emptyset$. Pour échantillonner un motif dans la ième cellule, nous procédons comme suit :

— extraction de l'ensemble $S = \{s_1, ..., s_n\}$ de motifs fréquents, fermés et diversifiés par rapport à $\mathcal{H}^{i-1}_{global}$, tel que $\forall\, s_j \in S, LB_J(s_j, \mathcal{H}^{i-1}_{global}) \leq J_{max}$.

— tirage d'un motif $s_j \in S$ proportionnellement à son poids et mise à jour de $\mathcal{H}^i_{global}$ comme suit : $\mathcal{H}^i_{global} \leftarrow \mathcal{H}^{i-1}_{global} \cup \{s_j\}$.

Soit k le nombre de tirages à réaliser, après l'échantillonnage des k motifs, nous avons $\mathcal{H}_{global} = \mathcal{H}^k_{global} = \{s_1, s_2, \ldots, s_k\}$, avec $\forall i, j \in [1, k] \wedge (i > j) : LB_J(s_i, s_j) \leq J_{max}$.

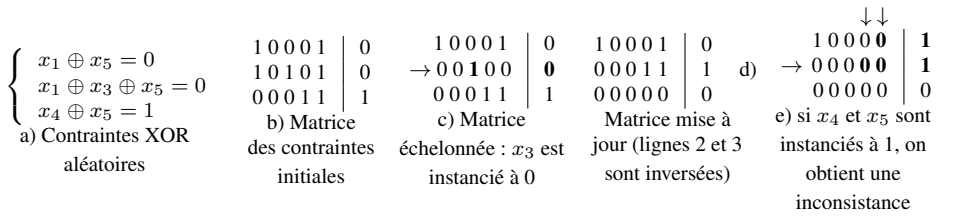


FIG. 3 – Exemple de propagation de contraintes XOR basé sur la procédure l'élimination de Gauss.

5.3 Filtrage de SDIVJAX

Le filtrage de SDIVJAX combine deux propagateurs, celui de CLOSEDDIVERSITY décrit par l'algorithme 2 et le propagateur des contraintes XOR. En PPC, au fur et à mesure que les variables sont modifiées par un propagateur lors de l'étape de filtrage, d'autres contraintes peuvent alors réagir à ces modifications en réveillant leur propre propagateur. Ce processus de réveil est géré de façon interne par le solveur en traduisant les informations de filtrage de la contrainte en événements compréhensibles par le solveur (retraits de valeurs, mise à jour des bornes, etc.). Dans ce qui suit, nous détaillons le propagateur des contraintes XOR. Il est basé sur le processus *d'élimination de Gauss*, un algorithme de résolution de systèmes d'équations linéaires. Chaque contrainte XOR de la forme $\bigotimes b_i.x_i = b_0$ (b_0 étant le bit de parité) peut être traduite en une somme de coefficients binaires de la forme $\sum b_i = b_0$. Les différentes contraintes XOR forment une matrice $\mathcal{M}$ de taille $(n, m+1)$, où n est le nombre de contraintes et m le nombre d'items du jeu de données, la dernière colonne représente le bit de parité. Ainsi, pour chaque contrainte k, si une variable x_i participe à la contrainte, alors $\mathcal{M}[k, i] = 1$ sinon $\mathcal{M}[k, i] = 0$.

La figure 3.e montre la représentation matricielle du système de contraintes XOR de la figure 3.a sur un jeu de données de 5 items. À chaque étape de propagation (Algorithme 4), la matrice $\mathcal{M}$ est mise à jour (ligne 5) à partir du motif partiel courant x^+. Pour toute variable x_i instanciée à 1 qui apparaît dans une contrainte XOR, le bit de parité de la contrainte est inversé et le coefficient de cette variable dans les lignes correspondantes de la matrice est mis à 0. La figure 3.d illustre cette opération aux lignes 1 et 2 de l'étape e et à la ligne 2 de l'étape c. Ensuite, on vérifie si la mise à jour de la matrice ne conduit pas à un cas d'inconsistance, i.e. une ligne vide avec un bit de parité égal à 1. Dans ce cas le système de contraintes XOR est insatisfiable (figure 3.e). Si ce test ne conduit pas à un échec, alors une opération d'échelonnement (ligne 12) est réalisée avec la méthode de Gauss-Jordan pour obtenir une matrice centrée-réduite (figure 3.c). Lors de l'échelonnement, deux situations permettent la propagation. Si une ligne devient vide alors que son membre de droite est égal à 1 le système est insatisfiable et la branche de recherche en cours se termine. Si une ligne ne contient qu'une seule variable libre, elle est alors affectée à son membre droit dans la ligne. Par exemple, à partir de la ligne 2 de la figure 3.c, on peut filtrer la valeur 1 du domaine de x_3.

6 Positionnement par rapport à l'état de l'art

La problématique de l'extraction d'ensembles de motifs diversifiés a été abordée dans la littérature, à la fois pour offrir des résultats plus intéressants et pour accélérer le processus d'ex-

Algorithme 4 : Propagateur de contraintes XOR

Entrée : $\mathcal{I}, \mathcal{T}, \mathcal{M}$: matrice des coefficients
Entrée/Sortie : $X = \{X_1 \dots X_n\}$: variables booléennes ;
Début
  $X^+ \leftarrow \{i \mid X_i = 1\}$, $X^- \leftarrow \{i \mid X_i = 0\}$, $X^* = \{i \in \mathcal{I} \mid i \notin X^+ \cup X^-\}$
  Pour $(i \in X^+ \wedge ligne\ r \in \mathcal{M})$ **faire**
    Si $(\mathcal{M}[r][i] = 1)$ **alors** $parity_r \leftarrow 1 - parity_r$
    $\mathcal{M}[r][i] \leftarrow 0$
  Si CHECKINCONSISTENCY () **alors** $fails()$;
  Si $(X^* = \emptyset)$
    retourner true;
  Sinon
    ECHELONNER ($\mathcal{M}$) ;
    Si CHECKINCONSISTENCY ()
      $fails()$;
    Sinon
      FIXERVARIABLES () ;
  retourner true;
Fonction CHECKINCONSISTENCY () *: Booléen*
  Pour chaque $ligne\ r \in \mathcal{M}$ **faire**
    Si $((parity_r = 1) \wedge (\sum_{i=1}^{n} \mathcal{M}[r][i] = 0))$ **alors**
      retourner true
  retourner false
Procédure FIXERVARIABLES ()
  Pour chaque $ligne\ r \in \mathcal{M}$ **faire**
    Si $(\sum_{i=1}^{n} \mathcal{M}[r][i] = 1)$ **alors**
      $dom(X_i) \leftarrow dom(X_i) - \{1 - parity_r\}$;
      $(X^+ \leftarrow X^+ \cup \{i\} \vee X^- \leftarrow X^- \cup \{i\}$;)

traction. Différentes approches basées sur la *recherche par faisceau* (*beam search* en anglais) proposent de peupler le faisceau pour la découverte de sous-groupes, en exploitant les meilleurs motifs partiels à étendre, tout en tenant compte du chevauchement de couverture (Van Leeuwen et Knobbe, 2012; Meeng et al., 2014; Vijayakumar et al., 2018). Contrairement à notre méthode exhaustive, la recherche par faisceau est heuristique, et comme elle extrait tous les motifs en même temps, des motifs partiels diversifiés peuvent conduire à un résultat final moins diversifié. Nous distinguons ci-dessous deux grandes catégories d'approches pour l'extraction d'un ensemble de motifs diversifiés, en termes de coût de calcul et de qualité des motifs extraits :

6.1 Approches par échantillonnage de motifs

Ces méthodes échantillonnent des motifs pour réduire les temps d'exécution et contrôler la taille des résultats. Avec une précision d'échantillonnage élevée, elles peuvent également produire un ensemble de résultats diversifiés provenant de différentes régions de l'espace des solutions. Les approches suivantes sont parmi les plus récentes et les plus reconnues :

— GIBBS (Bendimerad et al., 2020) s'appuie sur l'utilisation de mesures d'intérêt subjectif (Bie, 2011), qui peuvent être mises à jour avec les informations résultant des motifs déjà extraits afin de garantir que les futurs motifs apportent de nouvelles informations. Ce processus est sémantiquement similaire à l'utilisation de l'historique des motifs extraits pour calculer la mesure de Jaccard par paires, et devrait donc aboutir à un ensemble de motifs diversifiés. Cette mesure d'intérêt subjectif guide ensuite le processus d'échantillonnage de GIBBS pour sélectionner des motifs, ce qui est censé réduire les temps d'exécution.
— CFTP (Boley et al., 2012) échantillonne directement des motifs en utilisant une mesure de qualité. Cette méthode est conçue pour fournir rapidement des résultats qui peuvent, ou non, présenter de la diversité. Elle utilise une procédure d'échantillonnage aléatoire en deux étapes, adaptée à un ensemble restreint de tâches de recherche de motifs, en se basant sur une mesure de qualité φ. La probabilité d'échantillonnage des motifs est liée à leur score mais ne prend pas en compte les motifs déjà échantillonnés.
— FLEXICS (Dzyuba et al., 2017) FLEXICS (Dzyuba et al., 2017) utilise une collection de contraintes XOR aléatoires pour partitionner l'espace global des solutions en plusieurs bins non chevauchants, puis sélectionne un certain nombre de motifs à partir de ces bins. Les bins sont suffisamment petits pour être efficacement explorés à l'aide soit d'une approche CP, soit d'un algorithme de fouille de motifs dédié. Les contraintes XOR garantissent la diversité des données, ce qui théoriquement pourrait favoriser la diversité des motifs. Deux variantes de FLEXICS ont été proposées. D'une part, GFLEXICS s'appuie sur CP4IM (De Raedt et al., 2008). D'autre part, EFLEXICS utilise une extension d'ECLAT (Zaki et al., 1997). Bien que les auteurs discutent des ensembles de motifs, ils se limitent uniquement à une contrainte stricte de non-chevauchement des couvertures.
— Bosc et al. (2018) propose d'utiliser la recherche arborescente de Monte Carlo (MCTS) avec des bornes supérieures de confiance (UCB) pour orienter la recherche vers des régions intéressantes dans le treillis, en fonction de l'espace déjà exploré. Bien que la méthode MCTS soit nécessairement randomisée, elle permet une exploration *anytime*. Belfodil et al. (2019) considèrent les ensembles de descriptions de sous-groupes comme des *disjonctions* de tels motifs. En utilisant un algorithme glouton exploitant des bornes supérieures, les auteurs proposent d'extraire de manière itérative jusqu'à k descriptions de sous-groupes (similairement à notre travail). Toutefois, cette approche nécessite un attribut cible *et* une valeur cible sur laquelle se concentrer, tandis que notre approche permet une exploration non supervisée.

6.2 Approches de réduction de la redondance

Plusieurs travaux récents se concentrent sur l'extraction d'ensembles de motifs diversifiés (Van Leeuwen et Knobbe, 2012; Bosc et al., 2018; Belfodil et al., 2019) en traitant la réduction de la redondance comme une étape de post-traitement. Dans cette approche, les motifs doivent d'abord être extraits, puis les algorithmes sélectionnent un sous-ensemble de motifs parmi tous les itemsets extraits en utilisant plusieurs mesures de redondance, telles que l'entropie, mais uniquement pour des contextes d'exploration supervisée. Bien qu'il existe des travaux visant à réduire la *redondance* d'un ensemble de résultats dans un contexte d'exploration non supervisée (Knobbe et Ho, 2006; Bringmann et Zimmermann, 2009), ces méthodes

ne conduisent pas nécessairement à des ensembles *diversifiés* au sens de notre travail. Cette différence est liée au problème que CLOSEDDIVERSITY aborde, contrairement à la recherche d'*un ensemble de k motifs maximisant l'entropie* (PATTERNSTEAM) (Knobbe et Ho, 2006) ou d'un *ensemble de motifs non redondants* PICKER (Bringmann et Zimmermann, 2009), qui exploite des bornes pour prédire la présence de motifs dont la présence ou l'absence est la plus inattendue étant donné la partition actuelle. La première approche vise à fournir la quantité maximale d'informations à un analyste *humain* qui examine les motifs, tandis que la seconde tente de minimiser le nombre d'itemsets nécessaires pour qu'un *algorithme d'apprentissage* puisse modéliser toutes les informations disponibles.

Il existe également la méthode KRIMP (Vreeken et al., 2011), basée sur la Longueur Minimum de Description (MDL), qui effectue également un post-traitement des motifs. Cette méthode réduit implicitement la redondance en sélectionnant un ensemble de motifs qui compriment au mieux la base de données sous-jacente. Deux motifs qui sont syntaxiquement très similaires sont peu susceptibles d'être choisis ensemble, car leur contribution à la compression est probablement redondante. Cependant, deux itemsets qui sont syntaxiquement différents mais qui couvrent à peu près les mêmes transactions pourraient être sélectionnés ensemble, car leurs effets de compression seraient complémentaires. Tout comme dans le cas de PATTERNSTEAM et PICKER, KRIMP nécessite une première étape d'extraction potentiellement coûteuse et n'exploite pas la mesure de Jaccard.

7 Évaluation expérimentale

Nous avons évalué notre méthode en nous intéressant aux quatre points suivants :

1. le temps d'exécution et le nombre de motifs générés : pour cela, nous comparons d'une part CLOSEDDIV avec CLOSEDP et FLEXICS de (Dzyuba et al., 2017) et d'autre part SDIVJAX (les deux variantes) avec FLEXICS et CLOSEDDIV ;
2. réduction de la redondance : nous avons décidé de représenter visuellement la distribution des valeurs de Jaccard par paires pour les différentes méthodes. Nous avons opté pour des fonctions de densité cumulatives (CDF) sur les valeurs de Jaccard par paires ;
3. la qualité des motifs de CLOSEDDIV par rapport à ceux générés avec CLOSEDP et FLEXICS ;
4. 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%). Nous avons sélectionné pour chaque dataset des seuils de fréquence pour avoir différents nombres de motifs fermés et fréquents ($|Th(c)| \leq 15000$, $30000 \leq |Th(c)| \leq 10^6$, and $|Th(c)| > 10^6$). La seule exception concerne les jeux de données 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. La mise en oeuvre des différentes contraintes globales ont été réalisées en Java, sous `Choco`, une bibliothèque Java dédiée au développement des programmes à contraintes. Toutes les expérimentations ont été menées

sous Linux sur un **AMD Opteron** 6174**, 2.2 GHz** disposant d'une mémoire **RAM de** 256 **Go**. Nous avons utilisé une limite de temps de 24 heures et un espace mémoire alloué à la JVM de 30 Go. Comme seuil de diversité, nous avons fixé J_{max} à 5%. Nous avons également choisi d'utiliser comme heuristique par défaut de choix de variables MINCOV. Pour le choix des valeurs, le branchement se fait toujours d'abord sur la plus grande valeur du domaine de chaque variable. Tous les codes sources de nos méthodes sont disponibles à l'adresse https://github.com/lobnury/ClosedDiversity.

Pour évaluer la diversité d'un ensemble de motifs, nous continuons d'utiliser l'indice de Jaccard de chaque paire de motifs. Cependant, les méthodes d'agrégation statistiques de ces valeurs de Jaccard, comme le maximum , la moyenne ou le minimum, ne permettent pas d'avoir une vision d'ensemble sur la diversité de l'ensemble des motifs. Par exemple, un ensemble de motifs, ayant un Jaccard moyen de 0.5 sur les différentes paires pourrait être constitué de motifs qui se chevauchent tous à peu près à moitié mais également de motifs X qui, étant donné un autre motif Y, ont exactement la même couverture que Y ou ont une couverture totalement disjointe. Pour mieux rendre compte de la diversité d'un ensemble de motifs, nous avons choisi de représenter de façon visuelle la *distribution* des indices de Jaccard des différentes paires. Comme les fonctions de densité de probabilité peuvent être sujettes à des variation dans la distribution conduisant des formes visuellement plutôt distinctes, nous avons décidé d'utiliser à la place les *fonctions de répartition cumulative* (ou Cumulative Distribution Function (CDF)) sur les indices de Jaccard de chaque paire de motifs. Soit un ensemble de motifs $\mathcal{H} = \{P_1, \ldots, P_k\}$, la distribution est représentée comme suit :

$$CDF_{\mathcal{H}}(\tau) = \#\{(i,j) | Jac(P_i, P_j) \leq \tau, 1 \leq i < j \leq k\} \cdot \frac{2}{k(k-1)}$$

$\frac{2}{k(k-1)}$ est le facteur de normalisation permettant d'avoir une distribution comprise entre 0 et 1. Ainsi, pour tout indice de Jaccard τ, $CDF_{\mathcal{H}}(\tau)$ indique la proportion de paires de motifs ayant un indice de Jaccard *inférieur* à τ. Ainsi, les courbes les plus à gauche sur les graphiques représentent les ensembles les plus diversifiés car indiquant une valeur de Jaccard plus faible. De même pour les courbes les plus hautes.

7.1 Résultats expérimentaux avec CLOSEDDIVERSITY

7.1.1 Temps d'exécution

a)- CLOSEDDIV-MINCOV vs CLOSEDP : La table 1 compare CLOSEDDIV-MINCOV et CLOSEDPATTERNS pour différentes valeurs de θ. CLOSEDDIV-MINCOV présente différents comportements en fonction de la nature du jeu de données. Sur les jeux de données denses ($\rho \geq 30\%$), CLOSEDDIV-MINCOV est plus efficace que CLOSEDPATTERNS et jusqu'à trois ordres de grandeur plus rapide. Sur CHESS, l'accélération est de 903 pour $\theta = 20\%$. Pour les instances résultant entre 300 et 5000 motifs fréquents, fermés et diversifiés, CLOSEDDIV-MINCOV est souvent 17 fois plus rapide. Une autre observation importante qui peut être faite est que sur certaines instances (CHESS et KR-VS-KP) avec un seuil de fréquence minimum de $\%10$, CLOSEDPATTERNS ne parvient pas à terminer l'extraction dans la limite de temps. Les bonnes performances de CLOSEDDIV-MINCOV sont principalement dues aux règles de filtrage de LB qui permettent de réduire l'espace de recherche en filtrant plus de valeurs inconsistantes. En effet, CLOSEDDIV-MINCOV produit un nombre de nœuds toujours plus réduit que CLOSEDP. Sur

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	**65**	3072.25	**3.40**	45,617,249	**318**
75 × 3196	15	50,723,131	**238**	6164.81	**26.18**	101,446,261	**1154**
49.33%	10	OOM	**1,622**	OOM	**728,13**	OOM	**7,774**
HEPATITIS	30	83,048	**11**	5.84	**0.08**	166,095	**28**
68 × 137	20	410,318	**45**	32.62	**0.42**	820,635	**129**
50.00%	10	1,827,264	**1,018**	138.33	**33.79**	3,654,527	**2,545**
KR-VS-KP	30	5,219,727	**14**	775.82	**0.41**	10,439,453	**57**
73 × 3196	20	21,676,719	**141**	2108.37	**3.41**	43,353,437	**307**
49.32%	10	OOM	**1,609**	OOM	**744.49**	OOM	**7,703**
CONNECT	30	460,357	**19**	1765.59	**13.62**	920,713	**89**
129 × 67557	18	2,005,476	**141**	5790.21	**258.18**	4,010,951	**699**
33.33%	15	3,254,780	**297**	9349.59	**866.38**	6,509,559	**1,389**
HEART-CLEVELAND	10	12,774,456	**1,470**	1303.58	**73.39**	25,548,911	**3,735**
95 × 296	8	23,278,687	**4,761**	1836.99	**542.75**	46,557,373	**11,441**
47.37%	6	43,588,346	**20,490**	**3610.75**	7668.52	87,176,691	**46,506**
SPLICE1	10	1,606	**413**	**4.92**	27.75	3,211	**825**
287 × 3190	5	31,441	**7,920**	**95.66**	4214.48	62,881	**15,886**
20.91%	2	589,588	-	**1053.21**	-	1,179,175	-
MUSHROOM	5	8,977	**548**	**6.42**	52.21	17,953	**1,357**
112 × 8124	1	40,368	**9,935**	**23.37**	8976.82	80,735	**20,924**
18.75%	0.5	62,334	**23,931**	**32.60**	50646.09	124,667	**49,406**
T40I10D100K	8	138	**125**	**69.97**	346.24	275	**249**
942 × 100000	5	317	**284**	**253.75**	1514.76	633	**567**
4.20%	1	65,237	**7,217**	**5474.79**	53000.72	130,473	**14,517**
PUMSB	40	-	**4**	-	**58.78**	-	**15**
2113 × 49046	14	-	**15**	-	**246.80**	-	**59**
3.50%	20	-	**39**	-	**797.87**	-	**206**
T10I4D100K	5	11	11	**0.81**	5.05	21	21
870 × 100000	1	386	**360**	**424.10**	2481.72	771	720
1.16%	0.5	1,074	**607**	**704.66**	5491.64	2,147	**1,238**
BMS1	0.15	1,426	**592**	**10043.61**	61531.32	2,851	**1,186**
497 × 59602	0.14	1,683	**647**	**11480.74**	66287.70	3,365	**1,298**
0.51%	0.12	2,374	**778**	**13255.79**	74801.13	4,747	**1,560**
RETAIL	5	17	**12**	**10.74**	31.13	33	**23**
16470 × 88162	1	160	**105**	**304.19**	1599.69	319	218
0.06%	0.4	832	**515**	**6065.53**	31962.90	1,663	**1,071**

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-MINCOV

les jeux de données denses, les gains sont très importants avec une moyenne de gains d'environ 99 %. La seule exception est HEART-CLEVELAND pour lequel CLOSEDDIV-MINCOV est plus lent pour les valeurs de $\theta \leq 6\%$. Cela est principalement dû au nombre relativement important de motifs diversifiés ($\geq$ 20000), qui induit un sur-coût élevé pour le calcul de la borne inférieure. Nous observons le même comportement sur les deux jeux de données modérément denses SPLICE1 et MUSHROOM. Sur des jeux de données creux, CLOSEDDIV-MINCOV peut prendre beaucoup plus de temps pour extraire tous les motifs fréquents, fermés et diversifiés. Cela peut s'expliquer par le fait que sur ces instances presque tous les motifs fermés sont diversifiés par rapport à notre borne inférieure LB (en moyenne environ 65 % pour RETAIL et 37 % pour BMS1, (voir Table 1)). De ce fait, les motifs non diversifiés sont rarement filtrés, tandis que le sur-coût du calcul de la borne inférieure pénalise fortement la recherche des solutions.

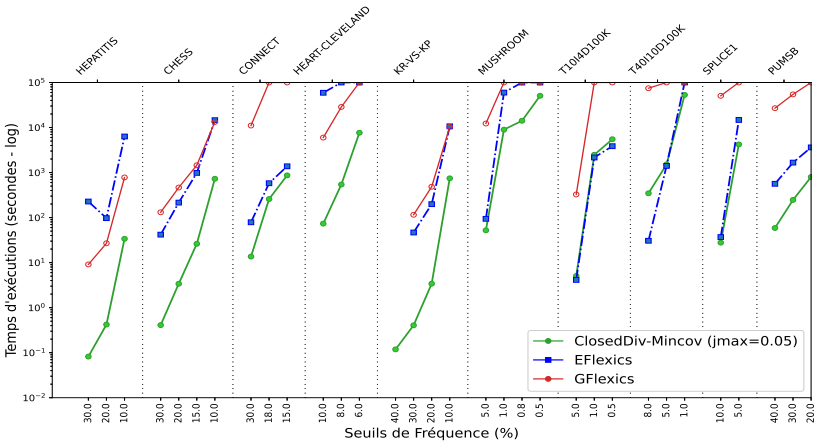


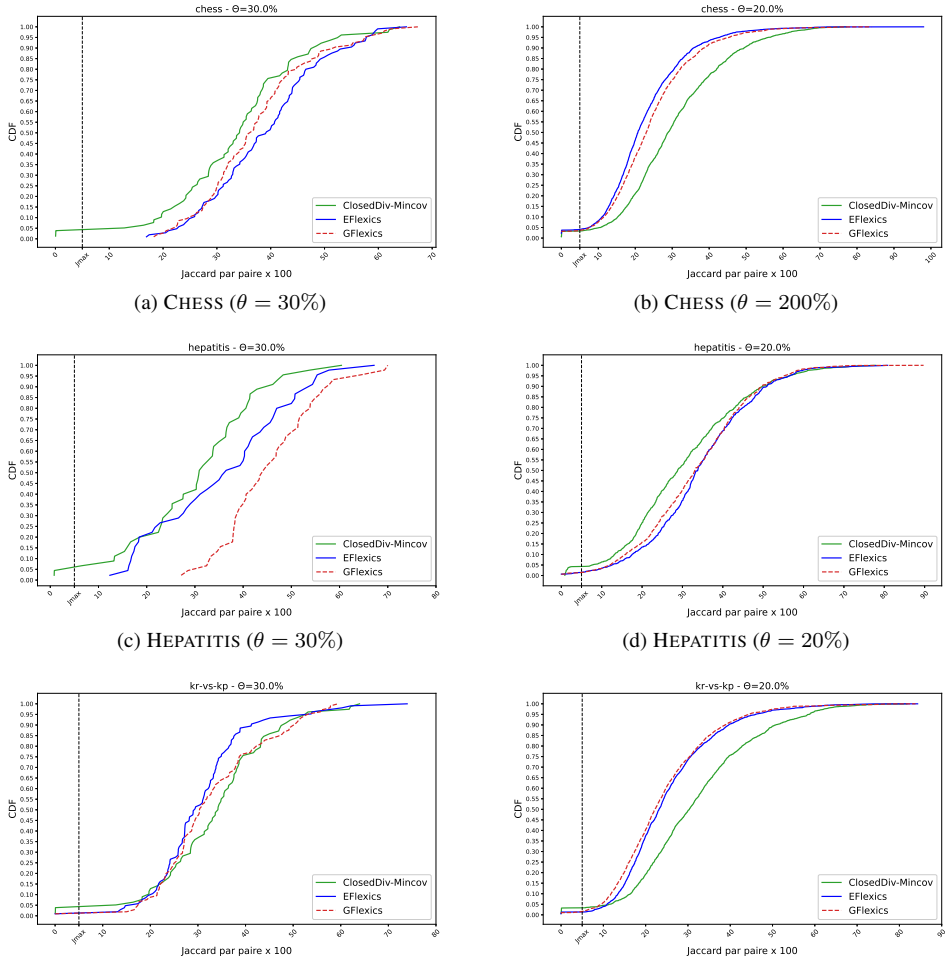
FIG. 4 – Analyse des performances : CLOSEDDIV vs FLEXICS.

Cela explique également la légère différence dans le nombre de nœuds exploré par les deux méthodes. Enfin, sur jeu de données très creuxPUMSB, notre approche est très efficace alors que CLOSEDPATTERNS ne parvient pas à terminer l'extraction.

b)- CLOSEDDIV-MINCOV vs FLEXICS : La figure 4 compare les temps d'exécution de notre approche CLOSEDDIV-MINCOV avec ceux des deux variantes de FLEXICS, EFLEXICS et GFLEXICS. CLOSEDDIV-MINCOV domine largement GFLEXICS. Sur plusieurs instances, GFLEXICS ne parvient pas à générer le nombre d'échantillons demandés pour des seuils de fréquence très bas dans la limite de 24 heures. Alors que EFLEXICS est presque toujours plus rapide que GFLEXICS, notre approche est toujours mieux classée que les deux variantes de FLEXICS. La seule exception est le jeu de données RETAIL, démontrant ainsi son utilité pour extraire des motifs diversifiés dans un contexte *anytime*.

7.1.2 Évaluations de la qualité de la diversification

Les figure 5 et 6 montrent les CDF des deux méthodes lorsqu'elles sont disponibles sur différents jeux de données. L'axe des x représente le Jaccard par paires (multiplié par 100 pour une meilleure lisibilité) et l'axe des y des proportions comprises entre 0 et 1. Pour toute valeur de Jaccard jc, les courbes indiquent quel pourcentage de paires de motifs ont des valeurs de Jaccard inférieures à jc. Sur les jeux de données denses et pour des valeurs de $\theta \geq 30\%$, les courbes de FLEXICS sont largement dominées par celles de CLOSEDDIV-MINCOV. Toutefois, pour des valeurs de $\theta \leq 30\%$, les CDF de FLEXICS sont bien meilleurs comme c'est le cas pour CHESS. Sur les jeux de données moyennement denses et très peu denses (cf. figure 6), nous pouvons observer que les courbes de CLOSEDDIV-MINCOV se situent au-dessus de celles de FLEXICS pour la plupart des jeux de données indiquant que CLOSEDDIV-MINCOV extrait toujours des couples de motifs pour lesquels le Jaccard par paire est très faible.



(a) CHESS ($\theta = 30\%$) (b) CHESS ($\theta = 200\%$)

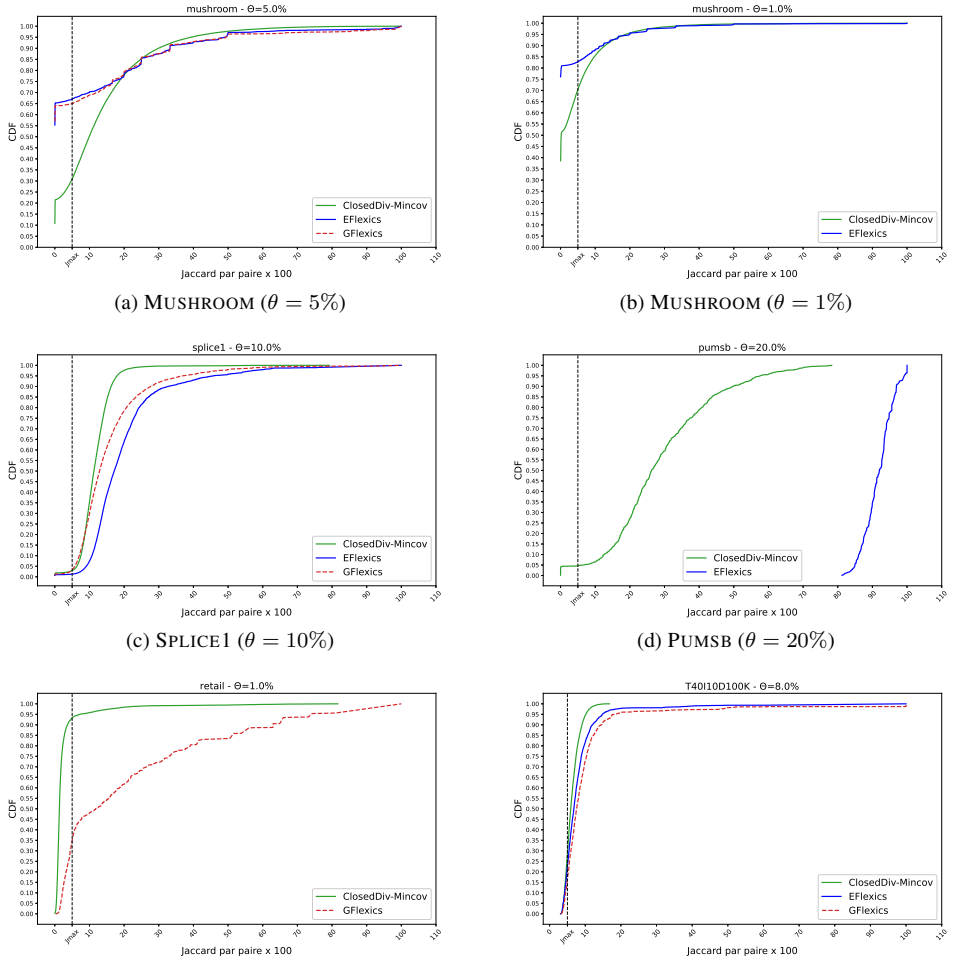
(c) HEPATITIS ($\theta = 30\%$) (d) HEPATITIS ($\theta = 20\%$)

(e) KR-VS-KP ($\theta = 30\%$) (f) KR-VS-KP ($\theta = 20\%$)

FIG. 5 – Évaluation de la redondance globale des paires de motifs de CLOSEDDIV-MINCOV et FLEXICS sur les instances denses. Les abscisses sont étiquetées avec les valeurs de Jaccard des paires de motifs (multipliées par 100 pour une meilleure lisibilité) et les ordonnées avec des valeurs de CDFs associées comprises entre 0 et 1. Pour toute valeur Jaccard donnée, les courbes indiquent quel pourcentage de paires de motifs a un indice de Jaccard inférieur à cette valeur.

7.1.3 Taille des ensembles de motifs

La table 1 indique, pour chaque jeux de données et seuil de fréquence, le nombre de motifs extraits par CLOSEDDIV-MINCOV et CLOSEDPATTERNS. Nous rapportons également le nombre de noeuds explorés. Les résultats mettent en évidence une grande différence entre les deux approches avec un nombre nettement inférieur de motifs générés par CLOSEDDIV-MINCOV (en milliers) par rapport à CLOSEDPATTERNS (en millions). Sur les jeux de données



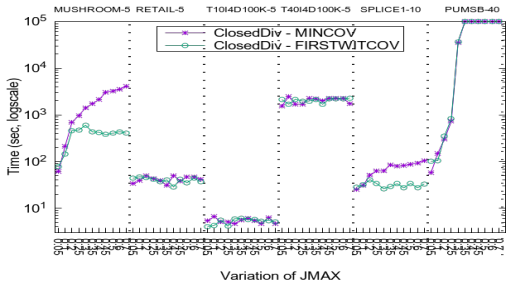
(a) MUSHROOM ($\theta = 5\%$) (b) MUSHROOM ($\theta = 1\%$)

(c) SPLICE1 ($\theta = 10\%$) (d) PUMSB ($\theta = 20\%$)

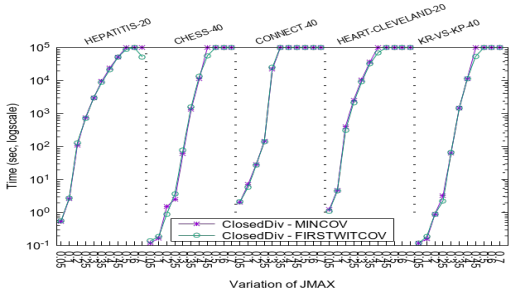
(e) RETAIL ($\theta = 1\%$) (f) T40I10D100K ($\theta = 8\%$)

FIG. 6 – Évaluation de la redondance globale des paires de motifs de CLOSEDDIV-MINCOV et FLEXICS sur les jeux de données moyennement denses et très peu denses.

denses et moyennement denses (de CHESS à MUSHROOM), l'écart est largement amplifié, en particulier pour les petites valeurs de θ. Par exemple, sur CHESS, le nombre de motifs extraits par CLOSEDDIV-MINCOV est réduit de 99.9% (de $\sim 50 \cdot 10^6$ motifs à 238) pour θ égal à 15%. La densité de ces bases explique le bon comportement de CLOSEDDIV-MINCOV. En effet, comme le nombre de motifs fermés augmente avec la densité, la redondance entre ces motifs augmente également. Sur les jeux de données très creux, CLOSEDDIV-MINCOV produit toujours moins de motifs que CLOSEDPATTERNS mais la différence est moins prononcée. Cela s'explique par le fait que sur ces jeux de données, nous avons peu de motifs et que presque tous les motifs sont diversifiés.



(a) Données peu denses : MINCOV vs FIRSTWITCOV.



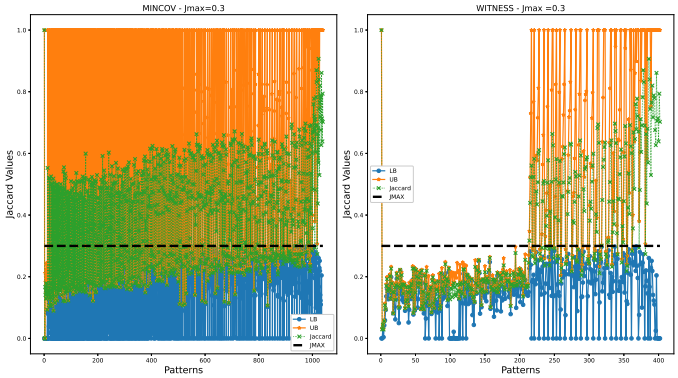
(b) Données denses : MINCOV vs. FIRSTWITCOV.

FIG. 7 – Analyse des performances : CLOSEDDIV vs FLEXICS.

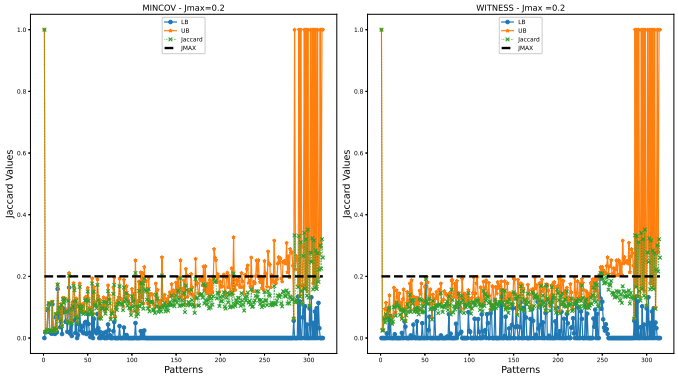
7.1.4 Évaluation des heuristiques de choix de variables

Dans cette expérimentation, nous comparons les deux heuristiques de choix de variables MINCOV et FIRSTWITCOV et nous étudions l'effet de la variation du paramètre J_{max}. La figure 7 montre l'évolution du nombre de motifs extraits pour différentes valeurs de J_{max} comprises entre 0.05 et 0.7. Ces résultats montrent que la taille de l'historique a un impact important sur les temps de calcul. En effet, la taille de l'historique $\mathcal{H}$ croît rapidement avec l'augmentation de J_{max}. Cela induit des sur-coûts importants dans les calculs des bornes inférieures et supérieures. Notons qu'en pratique, les utilisateurs ne sont intéressés que par des petites valeurs de J_{max} car la diversité des motifs obtenus est alors maximale et le nombre de motifs renvoyés devient raisonnable. Cela explique pourquoi nous avons fixé la valeur de J_{max} à 0.05 dans nos expérimentations.

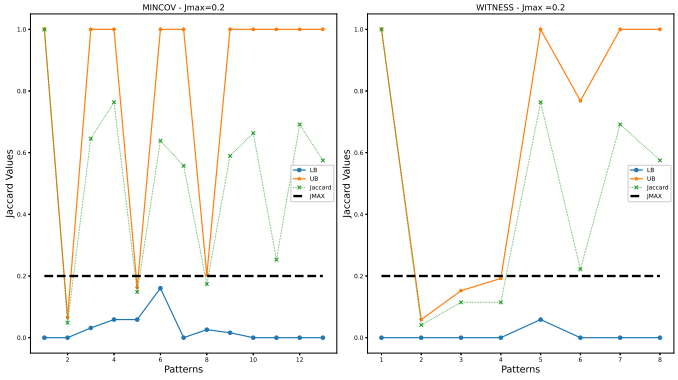
Les figures 7a et 7b comparent les deux heuristiques MINCOV et FIRSTWITCOV. Sur les jeux de données denses, les deux heuristiques se comportent de manière très similaire, avec un léger avantage pour FIRSTWITCOV pour $J_{max} \geq 0,45$. Par ailleurs, le nombre de motifs témoins extraits est très faible (moins de 1 % pour la plupart des instances) par rapport au nombre de motifs diversifiés extraits par les deux heuristiques (voir la figure 7). Cela explique pourquoi FIRSTWITCOV et MINCOV ont le même comportement. Ceci est dû à la stratégie FIRSTWITCOV qui évite l'exploration complète des différents sous-arbres témoins rencontrés lors de la recherche. Pour les jeux de données creux, nous constatons que les deux heuristiques ont presque les même performances, FIRSTWITCOV étant meilleur sur certaines instances. De plus, on peut observer (voir l'annexe complémentaire (Hien et al., 2020b)) que FIRSTWITCOV



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



(b) T40 ($\theta = 5\%$, $J_{max} = 0.2$)



(c) RETAIL ($\theta = 5\%$, $J_{max} = 0.2$)

FIG. 8 – Analyse qualitative de LB et UB

permet de générer des motifs de meilleure qualité grâce à sa capacité à guider la recherche vers des motifs témoins positifs.

7.1.5 Analyse qualitative des bornes

La figure 8 montre l'évolution des valeurs de LB_J, de l'indice de Jaccard et de UB_J des motifs générés par MINCOV et FIRSTWITCOV sur trois jeux de données (T40I10D100K, SPLICE1 et RETAIL) avec un seuil de fréquence $\theta = 5\%$ et un seuil de diversité $J_{max} \in \{0.2, 0.3\}$. Pour le premier motif extrait, l'indice de Jaccard et l'UB_J vaut 1, et la valeur de LB_J vaut 0. Les solutions sont ordonnées en fonction des valeurs de UB_J.

Nous constatons ainsi que la valeur de LB est toujours en-dessous du seuil J_{max} (représentées en rouge). Cela montre à quelle fréquence la règle de filtrage LB de CLOSEDDIVERSITY est utilisée. Cela confirme également la pertinence de la règle pour l'élagage des motifs fréquents, fermés non diversifiés. Concernant la valeur de UB_J, on peut constater qu'elle est toujours très proche de celle du Jaccard, ce qui signifie que notre borne supérieure du Jaccard fournit une relaxation serrée. De plus, un grand nombre de solutions ont des valeurs de UB_J inférieures ou très proches de J_{max}. Ceci est un indicateur de la qualité des motifs trouvés en termes de diversité. Par ailleurs, nous remarquons que les meilleures solutions (c'est-à-dire celles avec les plus petites valeurs de UB_J et de Jaccard) sont générées au début de la recherche. Ainsi, plus l'historique est grand, moins la qualité des solutions en termes de valeurs de Jaccard et de UB_J est bonne. Enfin, nous pouvons voir que FIRSTWITCOV permet de découvrir rapidement un ensemble de motifs de meilleure qualité en termes de valeurs de UB_J et de Jaccard comparativement à MINCOV. Cela démontre la force de notre heuristique de branchement UB_J pour favoriser les ensembles de motifs plus diversifiés.

7.2 Résultats expérimentaux avec SDIVJAX

Dans cette section, nous présentons l'étude expérimentale menée sur plusieurs jeux de données UCI, pour comparer et évaluer les apports pratiques de SDIVJAX et ses différentes variantes par rapport à l'approche FLEXICS. Nous discutons les résultats obtenus en termes de temps de calcul et de qualité de diversité des solutions.

La mise en œuvre de notre approche a été réalisée en Java pour la partie CLOSEDDIVERSITY et en Scala [2] pour la partie WEIGHTGEN. Nous avons utilisé une limite de temps de 24 heures et un seuil de diversité J_{max} fixé à 5%. Pour mesurer l'impact de l'étape d'estimation du nombre de contraintes XOR sur les deux variantes SDIVJAX, nous avons implanté une approche, dénommée SDIVJAX-CDIV, qui exploite la contrainte globale CLOSEDDIVERSITY comme oracle de nos deux méthodes SDIVJAX-1 et SDIVJAX-2 pendant l'étape d'estimation du nombre de contraintes XOR. Nous comparons nos deux oracles SDIVJAX-1 et SDIVJAX-2 aux deux variantes de FLEXICS (EFLEXICS et GFLEXICS). Pour cette dernière, nous avons utilisé les mêmes paramètres que précédemment. Pour toutes les méthodes, le nombres d'échantillons a été fixé au nombre de motifs extraits par CLOSEDDIVERSITY.

7.2.1 Temps d'exécution de SDIVJAX

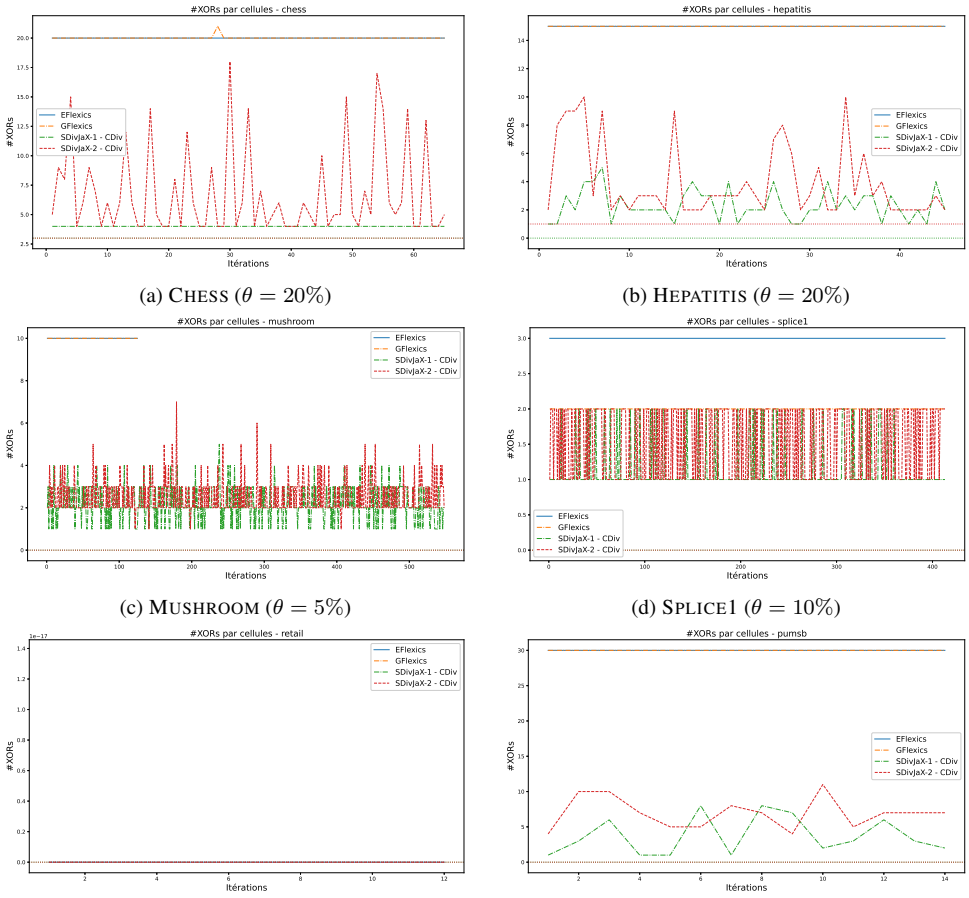
Le tableau 2 compare les temps d'exécution de SDIVJAX-1-CDIV et SDIVJAX-2-CDIV avec FLEXICS sur différents jeux de données et pour différentes valeurs de θ. Une première remarque est que pour des valeurs de θ élevées (exceptée HEART-CLEVELAND avec $\theta = 10\%$ et T40I10D100K avec $\theta = 8\%$), SDIVJAX-1-CDIV surpasse largement les autres approches,

2. https://www.scala-lang.org/

Dataset $\lvert\mathcal{I}\rvert \times \lvert\mathcal{T}\rvert$ $\rho(\%)$	$\theta(\%)$	CDiv		Flexics	
		(1)	(2)	EFLEXICS	GFLEXICS
HEPATITIS	30	**2.19**	*6.84*	227.34	9.11
68 × 137	20	*88.73*	110.84	97.16	**26.89**
50.00%	10	7119	7509	*6316*	**779.24**
CHESS	30	**4.08**	*15.80*	41.83	131.07
75 × 3196	20	*336.86*	639.94	**215.07**	465.12
49.33%	15	5457	2678.18	**981.49**	*1452.23*
	10	35342	61991	*14573*	**13305**
CONNECT	30	**43.26**	201.68	*78.69*	11073
129 × 67,557	18	-	*	**579.38**	-
33.33%	15	-	*20239*	**1377.09**	-
HEART-CLEVELAND	10	*12711*	40435	59096	**5984**
95 × 296	8	-	**8922**	-	*28629*
47.37%	6	-	*	-	-
KR-VS-KP	30	**3.71**	*15.32*	46.63	115.42
73 × 3196	20	*263.41*	619.75	**198.84**	484.07
49.32%	10	29426	40933	*10631*	**10341**
MUSHROOM	5	20479	16737	**93.79**	*12277*
112 × 8124	1	-	-	**59691**	-
18.75%	0.8	-	-	-	-
	0.5	-	-	-	-
T10I4D100K	5	**19.66**	*82.92*	UNS	326.67
	1	*4227*	-	**2163.16**	-
T40I10D100K	8	*555.35*	44683	**30.53**	74677
	5	*2080.23*	-	**1404.82**	-
SPLICE1	10	*4939*	4473	**36.95**	50622
	5	-	-	**14747**	-
RETAIL	5	**91.33**	*824.79*	UNS	1294.68
16470 × 88,162	1	**3184.75**	-	UNS	22279
0.06%	0.4	**50267**	-	UNS	-
PUMSB	40	**197.02**	*1818.32*	562.23	26880
2,113 × 49,046	30	*23144*	30267	**1662.70**	54418
3.50%	20	-	*62088*	**3619**	-

TAB. 2 – Analyse comparative des temps d'exécution de SDIVJAX avec FLEXICS. (1) : SDIVJAX-1-CDIV, (2) : SDIVJAX-2-CDIV. En gras, les résultats de la meilleure méthode. En italique, les résultats de la seconde meilleure méthode.

alors que SDIVJAX-2-CDIV est très souvent classée en seconde position. Toutefois, cette tendance s'inverse avec la diminution de la valeur de θ, où EFLEXICS obtient les meilleurs résultats sur la plupart des instances considérées. Notons, toutefois, les bonnes performances de SDIVJAX-1-CDIV, en particulier sur le jeu de données RETAIL, qui est très souvent classée en première position. Les résultats de SDIVJAX-2-CDIV ne sont pas compétitifs par rapport aux trois autres approches. Ce comportement peut s'expliquer par la stratégie de gestion de l'historique au niveau de SDIVJAX-CDIV. En effet, avec SDIVJAX-1-CDIV, l'historique évolue localement et plus rapidement lors de l'exploration d'une cellule, ce qui a pour conséquence de réduire l'espace de recherche et donc d'accélérer le temps d'exploration dans chaque cellule. Au contraire, l'historique maintenu par SDIVJAX-2-CDIV est plus globale et augmente que d'un seul motif d'une étape d'échantillonnage à une autre, ce qui pénalise fortement le temps global de résolution de SDIVJAX-2. Ce comportement est confirmé par les graphiques de la figure 9. En effet, après l'étape d'estimation (cf. traits en pointillés rouge), un sous-ensemble de contraintes XOR sont ajoutées (une contrainte `XOR` dans le cas de HEPATITIS). Ensuite, pour échantillonner un motif, SDIVJAX-2-CDIV nécessite de rajouter plus de contraintes `XOR`



(a) CHESS ($\theta = 20\%$) (b) HEPATITIS ($\theta = 20\%$)

(c) MUSHROOM ($\theta = 5\%$) (d) SPLICE1 ($\theta = 10\%$)

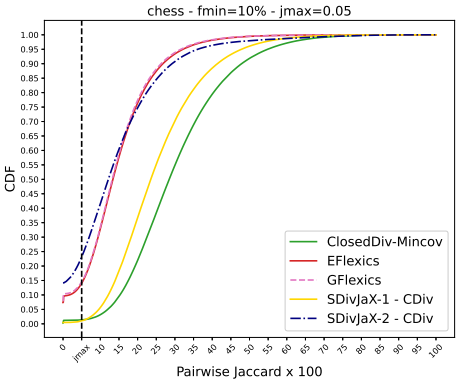
(e) RETAIL ($\theta = 5\%$) (f) PUMSB ($\theta = 30\%$)

FIG. 9 – Évolution du nombre de contraintes XOR utilisées par itération par SDIVJAX et FLEXICS.

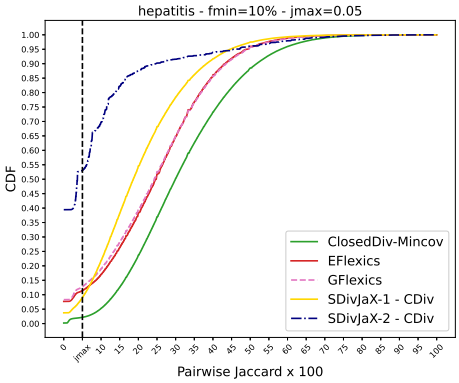
comparé à SDIVJAX-1-CDIV, jusqu'à 10 contraintes XOR dans le cas de HEPATITIS. Rappelons que l'ajout de contraintes XOR se fait lorsque la taille de la cellule, donnée par la somme des poids des motifs de la cellule, est beaucoup plus grande. Cette augmentation significative du nombre de contraintes XOR pénalise les performances de la méthode. Notons que sur le jeu de données RETAIL, aucune contrainte XOR n'a été rajoutée car ce dernier contient déjà peu de motifs par rapport aux seuils considérés.

7.2.2 Évaluation de la qualité de diversification

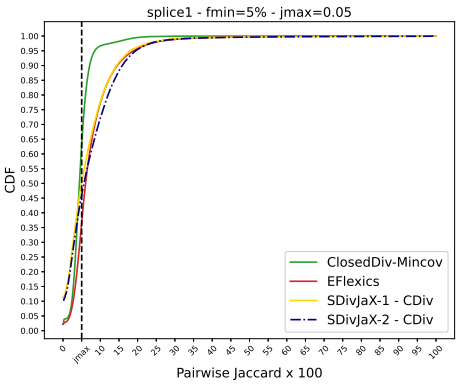
Dans la figure 10, nous représentons les distributions cumulatives d'indice de Jaccard de toutes les paires de motifs de CLOSEDDIV-MINCOV, FLEXICS et SDIVJAX-CDIV sur six jeux de données. Pour les jeux de données denses, les courbes des CDFs de SDIVJAX-CDIV se situent au-dessus de celles de CLOSEDDIV-MINCOV, indiquant que SDIVJAX-CDIV extrait des



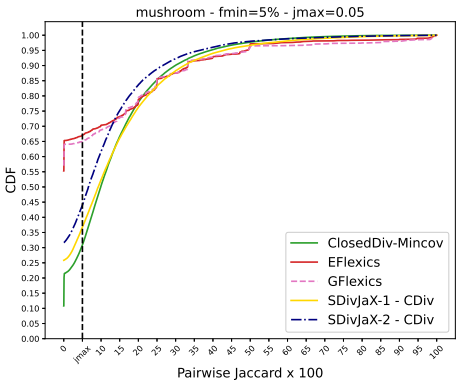
(a) CHESS ($\theta = 10\%$)



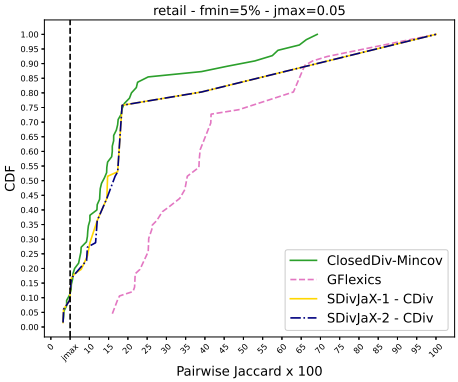
(b) HEPATITIS ($\theta = 10\%$)



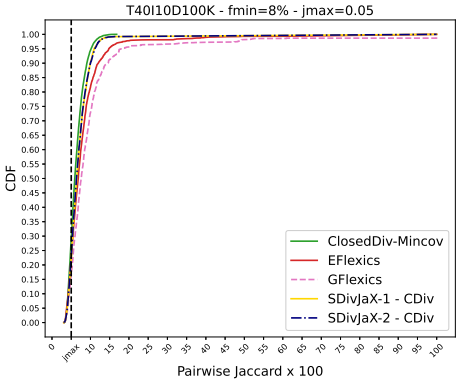
(c) SPLICE1 ($\theta = 5\%$)



(d) MUSHROOM ($\theta = 5\%$)



(e) RETAIL ($\theta = 5\%$)



(f) T10I4D100K ($\theta = 1\%$)

FIG. 10 – Évaluation de la redondance globale des paires de motifs de CLOSEDDIV-MINCOV, FLEXICS, et SDIVJAX-CDIV

ensembles de motifs pour lesquels l'indice de Jaccard des paires est relativement faible. Pour les instances creuses, les deux approches restent comparables. Comparé à FLEXICS, les CDFs de SDIVJAX-2 indiquent qu'il est plus facile de trouver des paires de motifs avec un faible Jaccard. Notons enfin que, sur la plupart des instances considérées, SDIVJAX-1-CDIV extrait des ensembles de motifs moins diversifiés que SDIVJAX-2-CDIV. En effet, l'idée de maintenir un historique $\mathcal{H}_{global}$ permet à SDIVJAX-2-CDIV d'extraire des ensembles de motifs de meilleure diversité.

8 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. Nous avons également proposé deux extensions de FLEXICS pour l'échantillonnage de motifs diversifiés. Nos expérimentations ont montré, d'une part, que notre approche CLOSEDDIVERSITY permet de réduire de façon significative le nombre de motifs par rapport à ceux générés par la contrainte globale CLOSEDPATTERNS et que les ensembles de motifs générés sont plus diversifiés. D'autre part, pour l'échantillonnage de motifs, SDIVJAX-2 constitue un meilleur compromis entre temps d'exécution et réduction de la redondance.

Notre travail constitue une première étape vers de nouvelles perspectives de recherche. Nous discutons ci-dessous des plus prometteuses.

Fouille interactive d'ensembles de motifs diversifiés. L'extraction d'un ensemble diversifié de motifs est particulièrement bien adaptée à la fouille interactive. En effet, la force de notre approche réside dans la proposition d'une collection réduite et variée de motifs à l'expert en données, qui peut les analyser rapidement. Il serait intéressant d'intégrer les retours des utilisateurs pour rendre le processus de fouille plus itératif et exploratoire. Une piste intéressante serait de proposer un moyen interactif d'ajuster la valeur de J_{max} en fonction des retours des utilisateurs. Cela peut être considéré comme une version de la boucle *Mine* (extraire une solution), *Interact* (interagir avec l'utilisateur), *Learn* (apprendre ses préférences) et *Repeat* (répéter avec les contraintes mises à jour) (van Leeuwen, 2014).

Extraction du $top-k$ ensemble de motifs diversifiés. En l'absence d'une fonction objective à optimiser, il peut exister plusieurs ensembles de motifs différents satisfaisant la contrainte CLOSEDDIVERSITY. Des travaux antérieurs ont tenté d'optimiser différentes métriques (c'est-à-dire maximiser la couverture par rapport au nombre de motifs) dans le cadre d'approches d'extraction de $top-k$ motifs (Ke et al., 2009; Wang et al., 2005) pour orienter la recherche vers un ensemble "optimal" de motifs. Une direction de recherche intéressante consiste à étendre notre approche à l'extraction des $top-k$ ensemble de motifs en utilisant l'entropie comme mesure d'intérêt d'un ensemble de motifs. Une telle approche permet également de gérer la difficulté du seuil J_{max} et de contrôler le nombre de solutions en sortie.

Langage des motifs. Notre cadre est suffisamment général pour être appliqué à une large variété de langages (comme les séquences, les arbres ou les graphes). Cependant, un changement de langage peut affecter l'efficacité des méthodes d'extraction. Par conséquent, l'extraction efficace d'ensembles de motifs diversifiés pour des langages plus complexes (par exemple, des

motifs séquentiels diversifiés) dépend fortement des avancées et progrès réalisés en modélisation par programmation par contraintes pour ces tâches.

Références

Aggarwal, C. C. et J. Han (2014). *equent pattern mining*. Springer.

Agrawal, R. et R. Srikant (1994). Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th VLDB*, San Francisco, CA, USA, pp. 487–499.

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

Bayardo, R. J. (2004). The hows, whys, and whens of constraints in itemset and rule discovery. In *European Workshop on Inductive Databases and Constraint Based Mining, Hinterzarten, Germany, March 11-13, 2004, Revised Selected Papers*, pp. 1–13.

Belaid, M., C. Bessiere, et N. Lazaar (2019). Constraint programming for association rules. In T. Y. Berger-Wolf et N. V. Chawla (Eds.), *Proceedings of the 2019 SIAM International Conference on Data Mining, SDM 2019, Calgary, Alberta, Canada, May 2-4, 2019*, pp. 127–135. SIAM.

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.

Bendimerad, A., J. Lijffijt, M. Plantevit, C. Robardet, et T. D. Bie (2020). Gibbs sampling subjectively interesting tiles. In *IDA 2020*, pp. 80–92.

Bhuiyan, M. et M. A. Hasan (2016). Interactive knowledge discovery from hidden data through sampling of frequent patterns. *Stat. Anal. Data Min. 9*(4), 205–229.

Bie, T. D. (2011). Maximum entropy models and subjective interestingness : an application to tiles in binary databases. *Data Min. Knowl. Discov. 23*(3), 407–446.

Boley, M., C. Lucchese, D. Paurat, et T. Gärtner (2011). Direct local pattern sampling by efficient two-step random procedures. In *KDD 2011*, pp. 582–590.

Boley, M., S. Moens, et T. Gärtner (2012). Linear space direct pattern sampling using coupling from the past. In *Proceedings of KDD '12*, pp. 69–77. ACM.

Borgelt, C. (2012). Frequent item set mining. *WIREs Data Mining Knowl. Discov. 2*(6), 437–456.

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.

Bringmann, B. et A. Zimmermann (2009). One in a million : picking the right patterns. *Knowl. Inf. Syst. 18*(1), 61–81.

Chakraborty, S., D. J. Fremont, K. S. Meel, S. A. Seshia, et M. Y. Vardi (2014). Distribution-aware sampling and weighted model counting for sat.

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.

Han, J., J. Pei, et Y. Yin (2000). Mining frequent patterns without candidate generation. *SIGMOD Rec. 29*(2), 1–12.

Hasan, M. A. et M. Zaki (2009). MUSK : uniform sampling of k maximal patterns. In *Proceedings of SDM 2009*, pp. 650–661. SIAM.

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/tree/master/Suppl_Material.

Hoeve, W. et I. Katriel (2006). Global constraints. In *Handbook of Constraint Programming*, pp. 169–208. Elsevier Science Inc.

Ke, Y., J. Cheng, et J. X. Yu (2009). Top-k correlative graph mining. In *SDM*, pp. 1038–1049. SIAM.

Kifer, D., J. Gehrke, C. Bucila, et W. White (2006). How to quickly find a witness. In *Constraint-Based Mining and Inductive Databases*, pp. 216–242. Springer Berlin Heidelberg.

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.

Meeng, M., W. Duivesteijn, et A. J. Knobbe (2014). Rocsearch - an roc-guided search strategy for subgroup discovery. In M. J. Zaki, Z. Obradovic, P. Tan, A. Banerjee, C. Kamath, et S. Parthasarathy (Eds.), *Proceedings of the 2014 SIAM International Conference on Data Mining, Philadelphia, Pennsylvania, USA, April 24-26, 2014*, pp. 704–712. SIAM.

Mitchell, T. M. (1982). Generalization as search. *Artif. Intell. 18*(2), 203–226.

Nijssen, S. et A. Zimmermann (2014). Constraint-based pattern mining. In C. C. Aggarwal et J. Han (Eds.), *Frequent Pattern Mining*, pp. 147–163. Springer.

Pasquier, N., Y. Bastide, R. Taouil, et L. Lakhal (1999). Discovering frequent closed itemsets for association rules. In *Proceedings of the 7th ICDT*, pp. 398–416.

Schaus, P., J. O. R. Aoga, et T. Guns (2017). Coversize : A global constraint for frequency-based itemset mining. In *Proceedings of the 23rd CP 2017*, pp. 529–546.

Tan, P.-N., M. Steinbach, et V. Kumar (2005). *Introduction to Data Mining*. Addison Wesley.

van Leeuwen, M. (2014). *Interactive Data Exploration Using Pattern Mining*, pp. 169–182. Berlin, Heidelberg : Springer Berlin Heidelberg.

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

Vernerey, C., S. Loudni, N. Aribi, et Y. Lebbah (2022). Threshold-free pattern mining meets multi-objective optimization : Application to association rules. In L. D. Raedt (Ed.), *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pp. 1880–1886. ijcai.org.

Vijayakumar, A. K., M. Cogswell, R. R. Selvaraju, Q. Sun, S. Lee, D. J. Crandall, et D. Batra (2018). Diverse beam search for improved description of complex scenes. In S. A. McIlraith et K. Q. Weinberger (Eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 7371–7379. AAAI Press.

Vreeken, J., M. Van Leeuwen, et A. Siebes (2011). Krimp : mining itemsets that compress. *Data Mining and Knowledge Discovery 23*(1), 169–214.

Wang, J., J. Han, Y. Lu, et P. Tzvetkov (2005). TFP : an efficient algorithm for mining top-k frequent closed itemsets. *IEEE Trans. Knowl. Data Eng. 17*(5), 652–664.

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.

Zaki, M., S. Parthasarathy, M. Ogihara, et W. Li (1997). New algorithms for fast discovery of association rules. In *Proceedings of KDD 1997, Newport Beach, California, USA, August 14-17, 1997*, pp. 283–286. AAAI Press.

Annexe

Cette section présente les preuves des différentes propositions du papier.

Preuve du Lemme 1 (Couverture résiduelle)

Preuve 1 *Comme* $Q \subseteq P$*, on obtient alors* $\boldsymbol{t}(P) \subseteq \boldsymbol{t}(Q)$*. Nous avons* $\boldsymbol{t}(P) = \boldsymbol{t}_H^{pr}(P) \cup \{\boldsymbol{t}(P) \cap \boldsymbol{t}(H)\}$*, avec* $\boldsymbol{t}_H^{pr}(P) \cap \{\boldsymbol{t}(P) \cap \boldsymbol{t}(H)\} = \emptyset$*. Par conséquent,* $\boldsymbol{t}_H^{pr}(P) = \boldsymbol{t}(P) \setminus \{\boldsymbol{t}(P) \cap \boldsymbol{t}(H)\}$.

Soit $t \in \boldsymbol{t}_H^{pr}(P)$*. Ainsi,* $t \in \boldsymbol{t}(P) \wedge t \notin \{\boldsymbol{t}(P) \cap \boldsymbol{t}(H)\} \cdots (1)$.
Comme $\boldsymbol{t}(P) \subseteq \boldsymbol{t}(Q)$*, à partir de* (1)*, on obtient* $t \in \boldsymbol{t}(P)$ *et* $t \in \boldsymbol{t}(Q) \cdots (2)$.
Comme $t \in \boldsymbol{t}(P) \wedge t \notin \{\boldsymbol{t}(P) \cap \boldsymbol{t}(H)\}$*, nous avons* $t \notin \boldsymbol{t}(H)$.
Puisque $t \notin \boldsymbol{t}(H) \Rightarrow t \notin \{\boldsymbol{t}(Q) \cap \boldsymbol{t}(H)\} \cdots (3)$
A partir de (2) *et* (3) *respectivement, on obtient* $t \in \boldsymbol{t}(Q)$ *et* $t \notin \{\boldsymbol{t}(Q) \cap \boldsymbol{t}(H)\} \cdots (4)$.
Nous avons $\boldsymbol{t}(Q) = \boldsymbol{t}_H^{pr}(Q) \cup \{\boldsymbol{t}(Q) \cap \boldsymbol{t}(H)\}$*. A partir de* (4) *On conclut que* $t \in \boldsymbol{t}_H^{pr}(Q)$.
Par conséquent, $\forall t \in \boldsymbol{t}_H^{pr}(P)$*, nous avons* $t \in \boldsymbol{t}_H^{pr}(Q)$ *et donc,* $\boldsymbol{t}_H^{pr}(P) \subseteq \boldsymbol{t}_H^{pr}(Q)$.

Preuve de la Proposition 2 (Nouvelle borne inférieure)

Preuve 2 $\forall H \in \mathcal{H}$ *nous avons :*

$$\begin{aligned} |\boldsymbol{t}(P)| \geq \theta &\Leftrightarrow |\boldsymbol{t}(H) \cap \boldsymbol{t}(P)| + |\boldsymbol{t}_H^{pr}(P)| \geq \theta \\ &\Leftrightarrow |\boldsymbol{t}(H) \cap \boldsymbol{t}(P)| \geq \theta - |\boldsymbol{t}_H^{pr}(P)| \end{aligned}$$

Puisque $|\boldsymbol{t}(H) \cup \boldsymbol{t}(P)| = |\boldsymbol{t}(H)| + |\boldsymbol{t}_H^{pr}(P)|$*, on obtient*

$$Jac(H,P) = \frac{|\boldsymbol{t}(H) \cap \boldsymbol{t}(P)|}{|\boldsymbol{t}(H) \cup \boldsymbol{t}(P)|} = \frac{|\boldsymbol{t}(H) \cap \boldsymbol{t}(P)|}{|\boldsymbol{t}(H)| + |\boldsymbol{t}_H^{pr}(P)|} \geq \frac{\theta - |\boldsymbol{t}_H^{pr}(P)|}{|\boldsymbol{t}(H)| + |\boldsymbol{t}_H^{pr}(P)|}$$

Comme $Jac(H,P) \geq 0$*, si* $|\boldsymbol{t}_H^{pr}(P)| \geq \theta$*, le numérateur devient* 0*, et* $\frac{0}{|\boldsymbol{t}(H)| + |\boldsymbol{t}_H^{pr}P|} = \frac{0}{\boldsymbol{t}(H)} = 0$.

Preuve de la Proposition 3

Preuve 3 *La preuve est triviale*

$$\begin{aligned} |\boldsymbol{t}(P)| \geq \theta &\Leftrightarrow |\boldsymbol{t}(P)| + |\boldsymbol{t}(H)| + |\boldsymbol{t}_H^{pr}(P)| - \theta \geq |\boldsymbol{t}(H)| + |\boldsymbol{t}_H^{pr}(P)| \\ &\Leftrightarrow \frac{1}{|\boldsymbol{t}(P)| + |\boldsymbol{t}(H)| + |\boldsymbol{t}_H^{pr}(P)| - \theta} \leq \frac{1}{|\boldsymbol{t}(H)| + |\boldsymbol{t}_H^{pr}(P)|} \end{aligned}$$

on obtient $LB_J^{old}(P,H) \leq LB_J(P,H)$.

Preuve de la Proposition 4 (Monotonie de LB_J)

Preuve 4 *Puisque $|\boldsymbol{t}_H^{pr}(P)| \leq |\boldsymbol{t}_H^{pr}(Q)|$ (voir lemme 1), on obtient*

$$\begin{aligned} LB_J(H,P) = \frac{\theta - |\boldsymbol{t}_H^{pr}(P)|}{|\boldsymbol{t}(H)| + |\boldsymbol{t}_H^{pr}(P)|} &\geq \frac{\theta - |\boldsymbol{t}_H^{pr}(Q)|}{|\boldsymbol{t}(H)| + |\boldsymbol{t}_H^{pr}(P)|} \\ &\geq \frac{\theta - |\boldsymbol{t}_H^{pr}(Q)|}{|\boldsymbol{t}(H)| + |\boldsymbol{t}_H^{pr}(Q)|} = LB_J(H,Q) \end{aligned}$$

Preuve de la Proposition 6 (Borne supérieure)

Preuve 5 *$\forall H \in \mathcal{H}$ nous avons :*

$$\begin{aligned} \Rightarrow Jac(H,P) &= \frac{|\boldsymbol{t}(H) \cap \boldsymbol{t}(P)|}{|\boldsymbol{t}(H) \cap \boldsymbol{t}(P)| + |\boldsymbol{t}_H^{pr}(P)| + |\boldsymbol{t}_P^{pr}(H)|} \\ \Rightarrow |\boldsymbol{t}_H^{pr}(P)| + |\boldsymbol{t}(H) \cap \boldsymbol{t}(P)| \geq \theta | \boldsymbol{t}_H^{pr}(P)| &\geq \theta - |\boldsymbol{t}(H) \cap \boldsymbol{t}(P)| \\ \Rightarrow Jac(H,P) &\leq \frac{|\boldsymbol{t}(H) \cap \boldsymbol{t}(P)|}{|\boldsymbol{t}(H) \cap \boldsymbol{t}(P)| + |\boldsymbol{t}_P^{pr}(H)| + \max\{0, \theta - |\boldsymbol{t}(H) \cap \boldsymbol{t}(P)|\}} \\ \Rightarrow Jac(H,P) &\leq \frac{|\boldsymbol{t}(H) \cap \boldsymbol{t}(P)|}{|\boldsymbol{t}_P^{pr}(H)| + \max\{\theta, |\boldsymbol{t}(H) \cap \boldsymbol{t}(P)|\}} \end{aligned}$$

Preuve de la Proposition 7 (Anti-monotonie de UB_J)

Preuve 6 *$\forall Q \supset P, \boldsymbol{t}(Q) \subseteq \boldsymbol{t}(P)$, et un motif de l'historique $H \in \mathcal{H}$:*

$$UB_J(H,P) = \frac{|\boldsymbol{t}(H) \cap \boldsymbol{t}(P)|}{|\boldsymbol{t}_P^{pr}(H)| + \max\{|\boldsymbol{t}(H) \cap \boldsymbol{t}(P)|, \theta\}}$$

Nous devons considérer trois cas :

1. $|\boldsymbol{t}(H) \cap \boldsymbol{t}(P)| > \theta, |\boldsymbol{t}(H) \cap \boldsymbol{t}(Q)| > \theta$:

$$\begin{aligned} UB_J(H,P) = \frac{|\boldsymbol{t}(H) \cap \boldsymbol{t}(P)|}{|\boldsymbol{t}_P^{pr}(H)| + |\boldsymbol{t}(H) \cap \boldsymbol{t}(P)|} &= \frac{|\boldsymbol{t}(H) \cap \boldsymbol{t}(P)|}{|\boldsymbol{t}(H)|} \\ &\geq \frac{|\boldsymbol{t}(H) \cap \boldsymbol{t}(Q)|}{|\boldsymbol{t}(H)|} = UB_J(H,Q) \end{aligned}$$

2. $|\boldsymbol{t}(H) \cap \boldsymbol{t}(P)| > \theta, |\boldsymbol{t}(H) \cap \boldsymbol{t}(Q)| \leq \theta \Rightarrow \boldsymbol{t}_P^{pr}(H) + \theta \geq \boldsymbol{t}(H)$:

$$\begin{aligned} UB_J(H,P) = \frac{|\boldsymbol{t}(H) \cap \boldsymbol{t}(P)|}{|\boldsymbol{t}_P^{pr}(H)| + |\boldsymbol{t}(H) \cap \boldsymbol{t}(P)|} &= \frac{|\boldsymbol{t}(H) \cap \boldsymbol{t}(P)|}{|\boldsymbol{t}(H)|} \\ &\geq \frac{|\boldsymbol{t}(H) \cap \boldsymbol{t}(Q)|}{|\boldsymbol{t}_Q^{pr}(H)| + \theta} = UB_J(H,Q) \end{aligned}$$

3. $|\boldsymbol{t}(H) \cap \boldsymbol{t}(P)| \leq \theta, |\boldsymbol{t}(H) \cap \boldsymbol{t}(Q)| \leq \theta \Rightarrow |\boldsymbol{t}(H) \cap \boldsymbol{t}(P)| \geq |\boldsymbol{t}(H) \cap \boldsymbol{t}(Q)| \boldsymbol{t}_P^{pr}(H) \leq \boldsymbol{t}_Q^{pr}(H)$:

$$UB_J(H,P) = \frac{|\boldsymbol{t}(H) \cap \boldsymbol{t}(P)|}{|\boldsymbol{t}_P^{pr}(H)| + \theta} \geq \frac{|\boldsymbol{t}(H) \cap \boldsymbol{t}(Q)|}{|\boldsymbol{t}_P^{pr}(H)| + \theta}$$
$$\geq \frac{|\boldsymbol{t}(H) \cap \boldsymbol{t}(Q)|}{|\boldsymbol{t}_Q^{pr}(H)| + \theta} = UB_J(H,Q)$$

Étant donné que la propriété d'anti-monotonie est vérifiée dans les trois cas, alors elle l'est également pour la borne supérieure.

Preuve de la Proposition 8 (Règles de filtrage de CLOSEDDIVERSITY)

Preuve 7 *La preuve de la règle de filtrage* (1) *est une conséquence de la proposition 5. Nous donnons ci-après la preuve de la règle* (2)*. Soit* $P = x^+ \cup \{i\}$*,* $Q = x^+ \cup \{k\}$ *t.q.* $i \in x^*$ *and* $k \in x^-_{Div}$ *et* $H \in \mathcal{H}$*.*

$$LB_J(H,P) = \frac{\theta - |\boldsymbol{t}_H^{pr}(P)|}{|\boldsymbol{t}(H)| + |\boldsymbol{t}_H^{pr}(P)|}$$
$$|\boldsymbol{t}(P)| \leq |\boldsymbol{t}(Q)| \Rightarrow |\boldsymbol{t}_H^{pr}(P)| \leq |\boldsymbol{t}_H^{pr}(Q)|$$
$$\Rightarrow LB_J(H,P) \geq \frac{\theta - |\boldsymbol{t}_H^{pr}(Q)|}{|\boldsymbol{t}(H)| + |\boldsymbol{t}_H^{pr}(P)|} \geq \frac{\theta - |\boldsymbol{t}_H^{pr}(Q)|}{|\boldsymbol{t}(H)| + |\boldsymbol{t}_H^{pr}(Q)|} = LB_J(H,Q)$$

Preuve de la Proposition 9 (consistance de domaine et complexité)

Preuve 8 ***(i) Consistance de domaine*** *L'algorithme 2 utilise deux groupes de règles de filtrage : (i) les règles portant sur la fermeture permettant de maintenir l'arc-consistance (voir (Lazaar et al., 2016)), et (ii) la règle basée sur la relaxation* LB *permettant de garantir que chaque item libre* $i \in x^*$ *ne pouvant étendre un motif courant* x^+ *vers un motif fréquent, fermé et diversifié, sera nécessairement filtré (voir Proposition 8). Par construction, toute valeur* $x_i \in x$ *ne conduisant pas à une solution satisfaisant la contrainte* CLOSEDDIVERSITY *sera supprimée de* $dom(x_i)$*. Par conséquent, la contrainte* CLOSEDDIVERSITY *maintient l'Arc-consistante sur les variables de* x*.*
(ii) Complexité temporelle. *La ligne 6 s'exécute en* $\mathcal{O}(n \times m)$*. La fonction* $\mathcal{P}Growth_{LB}$ *s'exécute en* $\mathcal{O}(m \times |\mathcal{H}|)$*, car chaque appel à* $LB_J(H,x)$ *s'exécute en* $\mathcal{O}(m)$*, ainsi le filtrage de domaine s'effectue en* $\mathcal{O}(n \times m \times |\mathcal{H}|)$*. Supposons que* $|\mathcal{H}| \leq n$*, alors le filtrage de domaine s'effectue en* $\mathcal{O}(n^2 \times m)$*. De plus, les filtrages portant sur la fréquence et la fermeture s'effectuent en* $\mathcal{O}(\frac{n^2}{4} \times m)$*, car (i) le filtrage des items non fréquents se fait en* $\mathcal{O}(n \times m)$ *car le test de la ligne 6 se fait en* $\mathcal{O}(m)$*; (ii) le filtrage des items n'apparaissant pas dans la clôture d'un motif se fait en en* $\mathcal{O}(n \times m)$ *car le test de la ligne 10 est réalisé en* $\mathcal{O}(m)$*; (iii) enfin, le filtrage des lignes 17-25 se fait en* $\mathcal{O}(\frac{n^2}{4} \times m)$ *car le test d'inclusion de la ligne 18 se fait en* $\mathcal{O}(m)$ *et* $|x^*| + |x^-|$ *vaut au plus* n*, ainsi cette règle de filtrage est alors vérifiée avec au plus* $\frac{n}{2} \times \frac{n}{2}$ *opérations (i.e.* $\mathcal{O}(\frac{n^2}{4})$*). Comme cette dernière règle de filtrage est dissociée de la règle de filtrage sur la diversité (c'est pour cette raison que nous continuons à la ligne 16), la complexité temporelle de l'algorithme de filtrage est alors de* $\mathcal{O}(n^2 \times m)$*.*

Fouille de motifs diversifiés: une approche Basée sur la relaxation et l'échantillonnage

Summary

In this paper, we use constraint programming 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. Finally, we show how to integrate our relaxation to sample diverse set of patterns. We demonstrate experimentally that our approach CLOSEDDIVERSITY significantly reduces the number of patterns and is very efficient in terms of running times, particularly on dense datasets. For the pattern sampling task, we show that SDIVJAX-2 constitutes a better compromise between running time and redundancy reduction.