Extraction efficace des représentations condensées de motifs: Applications aux skypatterns et aux clusterings conceptuels

Charles Vernerey**, Samir Loudni**, Noureddine Aribi * Yahia Lebbah*

*University of Oran1, Lab. LITIO, 31000 Oran, Algeria
**TASC (LS2N-CNRS), IMT Atlantique, FR – 44307 Nantes, France

Résumé. Nous proposons dans ce papier un cadre générique basé sur la programmation par contraintes pour découvrir des représentations condensées de motifs par rapport à un ensemble de mesures. Pour cela, nous introduisons une nouvelle contrainte globale avec un algorithme de filtrage de complexité polynomiale. Nous démontrons l'utilité de notre contrainte globale en l'exploitant dans deux modèles à contraintes que nous proposons pour la découverte de skypatterns et pour le clustering conceptuel. Les expérimentations menées sur différents jeux de données démontrent l'efficacité de notre approche et les avantages significatifs qu'elle présente comparé aux approches existantes.

1 Introduction

La fouille de motifs vise à découvrir des régularités intéressantes dans des bases de données (Novak et al., 2009; Wrobel, 1997). La majorité des approches existantes réalise une énumération complète des motifs qui respectent un ensemble de contraintes. Cependant, le nombre important de motifs rend l'analyse de ces derniers très compliquée pour l'utilisateur. Une solution à ce problème repose sur le principe de *représentation condensée*. Cette approche a été utilisée principalement avec la mesure de fréquence (Calders et al., 2004) et il y a peu d'études sur les autres mesures (Giacometti et al., 2002; Soulet et al., 2004). Soulet et Crémilleux (2008) ont étendu le principe de représentation condensée de motifs à un ensemble de mesures. Ils ont proposé l'algorithme MICMAC pour la fouille de représentations condensées adéquates grâce à un nouvel opérateur de fermeture basé sur la notion de *fonction condensable*. Cependant, le problème principal de cette approche est son passage à l'échelle.

Les autres approches pour réduire le nombre de motifs sont basées sur les préférences de l'utilisateur. L'approche la plus populaire est la procédure top-k, qui retourne les k meilleurs motifs par rapport à une mesure choisie par l'utilisateur (Ke et al., 2009; Wang et al., 2005). D'autres méthodes basées sur la *Pareto dominance*, ou *skylines*, ont été proposées. Soulet et al. (2011) ont utilisé la notion de requêtes skylines (Börzsönyi et al., 2001) afin de découvrir les motifs Pareto (skypatterns). L'approche proposée, intitulée AETHERIS, exploite une représentation condensée adéquate de motifs et la notion de *skylineabilité* afin de réduire le temps d'exécution. Dans (Ugarte et al., 2017), une méthode (intitulée CP+SKY) qui utilise des contraintes dynamiques a été proposée. Elle exploite un modèle réifié pour encoder les motifs (Raedt et al., 2008). Cependant, l'utilisation de contraintes réifiées dans le modèle constitue un frein majeur pour son passage à l'échelle.

Récemment, la programmation par contraintes (PPC) a été utilisée avec succès pour modéliser différents problèmes de fouilles de données (Guns et al., 2011; Hien et al., 2020; Lazaar et al., 2016; Vernerey et al., 2022). L'avantage principal d'utiliser la PPC pour la fouille de données est sa déclarativité et sa flexibilité, ce qui permet d'ajouter de nouvelles contraintes spécifiées par l'utilisateur sans avoir à modifier le système sous-jacent.

Dans cet article, nous proposons une nouvelle contrainte globale, dénommée ADEQUATECLOSURE, pour extraire efficacement des représentations condensées adéquates de motifs par rapport à un ensemble de mesures. Cela est possible grâce à un opérateur de fermeture qui exploite le concept de mesure préservante. Nous démontrons l'utilité de notre contrainte globale pour la découverte de skypatterns et pour le clustering conceptuel. Enfin, nous présentons une étude expérimentale qui compare notre approche à celles existantes pour la fouille de représentations condensées adéquates de motifs et la fouille de skypatterns afin de démontrer son efficacité et son passage à l'échelle. Nous terminons par quelques résultats qualitatifs sur le clustering conceptuel optimisant simultanément plusieurs fonctions objectifs.

Une version décomposée et efficace de la contrainte globale ADEQUATECLOSURE, permettant d'une part de séparer le filtrage de la contrainte de clôture sur différentes mesures du filtrage de la contrainte de fréquence minimale et d'autre part de ne plus à gérer des variables supplémentaires tel que la fréquence du motif dans la signature de la contrainte, a été publiée aux deux conférences IJCAI 2022 (Vernerey et al., 2022) et EGC 2023 (Vernerey et al., 2023) (cf. remarque 1, Sect. 3). Dans ces deux articles, nous avons proposé une application pour l'extraction de règles d'association minimales non-redondantes (MNRs) (Bastide et al., 2000). Ainsi, l'exemple d'application au clustering conceptuel proposé dans cet article étendu est nouveau. Par ailleurs, l'étude expérimentale de la section 7.2, permettant de comparer notre contrainte globale (noté ADEQUATE-CI) à CP+CLOSED et MICMAC pour l'extraction de représentations condensées adéquates par rapport à un ensemble de mesures, est également nouvelle.

Ce papier est organisé comme suit. La section 2 rappelle les préliminaires. La section 3 introduit notre contrainte globale ADEQUATECLOSURE avec son algorithme de filtrage. La section 4 décrit un premier cas d'usage de notre contrainte ADEQUATECLOSURE pour la découverte de skypatterns. La section 5 présente un modèle PPC, basé sur une approche en deux étapes, pour faire du clustering conceptuel en exploitant les motifs extraits par rapport à une représentation condensée. La section 6 passe en revue les travaux connexes et dans la section 7, nous présentons un ensemble complet d'expériences. Enfin, nous concluons en section 8.

2 Préliminaires

2.1 Fouille de motifs

Soit $\mathcal{I} = \{1, ..., n\}$ un ensemble de n *items*, un *motif* P est un sous-ensemble non vide de $\mathcal{I}$. Le langage des motifs correspond à $\mathcal{L}_\mathcal{I} = 2^\mathcal{I} \backslash \emptyset$. Un jeu de données transactionnel $\mathcal{D}$ est un ensemble de transactions, où chaque *transaction* t est un sous ensemble de $\mathcal{I}$, i.e., $t \subseteq \mathcal{I}$; $\mathcal{T} = \{1, ..., m\}$ est un ensemble de m indices de *transactions*. 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 : $\boldsymbol{t}_\mathcal{D}(P) = \{t \in \mathcal{D} \mid P \subseteq t\}$. Le *support* de P dans $\mathcal{D}$ est la taille de sa couverture : $sup_\mathcal{D}(P) = |\boldsymbol{t}_\mathcal{D}(P)|$. Un motif P est dit *fréquent* dans $\mathcal{D}$ si $sup_\mathcal{D}(P) \geq \theta$,

où θ est un seuil minimal fixé par l'utilisateur. Étant donné $T \subseteq \mathcal{D}$, $\boldsymbol{i}(T)$ est l'ensemble des items qui sont communs à toutes les transactions de T : $\boldsymbol{i}(T) = \{i \in \mathcal{I} \mid \forall t \in T, i \in t\}$. On définit par $\boldsymbol{c}$ un *opérateur de fermeture*, tel que $\boldsymbol{c}(P) = \boldsymbol{i} \circ \boldsymbol{t}(P) = \boldsymbol{i}(\boldsymbol{t}(P))$. La *fermeture* d'un motif P est l'ensemble des items qui sont contenus dans toutes les transactions de $\boldsymbol{t}(P)$: $\boldsymbol{c}(P) = \{i \in \mathcal{I} \mid \forall t \in \boldsymbol{t}(P), i \in t\}$. Un motif P est dit *clos* (Pasquier et al., 1999) ssi $\boldsymbol{c}(P) = P$. L'opérateur de fermeture permet de définir les classes d'équivalence, et donc la représentation *condensée* des motifs.

Plusieurs mesures basées sur la fréquence sont utilisées afin d'évaluer l'intérêt d'un motif. Soit $\mathcal{D}$ un jeu de données partitionné en deux sous-ensembles $\mathcal{D}_1$ et $\mathcal{D}_2$. Le taux de croissance (gr_1) est une mesure qui permet de mettre en valeur les motifs dont la fréquence augmente significativement d'un sous-jeu de données à l'autre (Novak et al., 2009). Le support disjonctif d'un motif p est $sup_\vee(P) = |\{t \in \mathcal{D} \mid \exists i \in P : i \in t\}|$ et $size$ sa cardinalité. Des informations additionnelles (tel que des valeurs numériques associées aux items) peuvent également être utilisées. Étant donné une fonction $val : \mathcal{I} \to \mathbb{R}_+$, nous l'étendons à un motif P et nous notons $P.val$ l'ensemble $\{val(i) \mid i \in P\}$. Ce type de fonction peut être utilisé avec les primitives usuelles telles que sum, $\min$ et $\max$. Par exemple, $sum(P.val)$ est la somme des val pour chaque item de P.

La condensation des motifs vient du fait qu'il y a des dépendances entre eux. Le concept de *mesure préservante* (Soulet et Crémilleux, 2008), qui révèle cette dépendance entre un motif et ses spécialisations, est à la base des représentations condensées basées sur la fermeture.

Définition 1 (Mesure préservante) *Une mesure m est dite préservante ssi $\forall i \in \mathcal{I}$ et pour chaque motif $P \subseteq Q$ si $m(P \cup \{i\}) = m(P)$ alors $m(Q \cup \{i\}) = m(Q)$.*

Proposition 1 (Quelques mesures préservantes) $\min$*, sup,* $sup_\vee$, $\max$*,* $mean$ *et* sum *sont des mesures préservantes.*

Un *opérateur de fermeture adéquat* à des mesures autre que la fréquence a été proposé dans (Soulet et Crémilleux, 2008). Cette opérateur exploite la notion de mesure préservante.

Définition 2 (Opérateur de fermeture adéquat) *Soit M un ensemble de mesures préservantes. La fermeture d'un motif P par rapport à M, noté $\texttt{clos}_M(P)$, est l'ensemble d'items tel que $\texttt{clos}_M(P) = \{i \in \mathcal{I} \mid \forall m \in M, m(P \cup \{i\}) = m(P)\}$.*

Proposition 2 (Motifs clos) *Soit M un ensemble de mesures préservantes, $\texttt{clos}_M$ est un opérateur de fermeture. De plus, P est clos par rapport à M ssi $\texttt{clos}_M(P) = P$.*

Exemple 1 *Soit le jeu de données de la table 1 et $M = \{sup, \min\}$. B est un motif clos par rapport à M car $\texttt{clos}_M(B) = B$, i.e. $\nexists i \in \mathcal{I}$ tel que $sup(B) = sup(B \cup \{i\}) \wedge min(B.val) = min(B \cup \{i\}.val)$. Cependant, A n'est pas un motif clos par rapport à M car $\texttt{clos}_M(A) = AE$, i.e. il existe AE tel que $sup(A) = sup(AE) = 3$ et $min(A.val) = min(AE.val) = 30$. La table 1c montre les 17 motifs clos par rapport à $M = \{sup, \min\}$ avec $\theta = 1$.*

Trans.	Items						
t_1		B			E	F	$\mathcal{D}_1$
t_2		B	C	D			
t_3	A				E	F	
t_4	A	B	C	D	E		
t_5		B	C	D	E		$\mathcal{D}_2$
t_6		B	C	D	E	F	
t_7	A	B	C	D	E	F	

(a)

Item	val
A	30
B	40
C	10
D	40
E	70
F	50

Name	Definition
$area$	$X \mapsto sup(X) \times size(X)$
$mean$	$X \mapsto \frac{\min(X.val)+\max(X.val)}{2}$
min	$X \mapsto \min(X.val)$
$size$	$X \mapsto \lvert X \rvert$
$bond$	$X \mapsto \frac{sup(X)}{sup_\vee(X)}$
gr_1	$X \mapsto \frac{\lvert\mathcal{D}_2\rvert}{\lvert\mathcal{D}_1\rvert} \times \frac{sup_{\mathcal{D}_2}(X)}{sup_{\mathcal{D}_2}(X)}$

(b)

Itemset	$\langle sup, \min \rangle$	Itemset	$\langle sup, \min \rangle$
B	$\langle 6, 40 \rangle$	E	$\langle 6, 70 \rangle$
AE	$\langle 3, 30 \rangle$	BD	$\langle 5, 40 \rangle$
BE	$\langle 5, 40 \rangle$	EF	$\langle 4, 50 \rangle$
AEF	$\langle 2, 30 \rangle$	BCD	$\langle 5, 10 \rangle$
BDE	$\langle 4, 40 \rangle$	BEF	$\langle 3, 40 \rangle$
$ABDE$	$\langle 2, 30 \rangle$	$BCDE$	$\langle 4, 10 \rangle$
$BDEF$	$\langle 2, 40 \rangle$	$ABCDE$	$\langle 2, 10 \rangle$
$ABDEF$	$\langle 1, 30 \rangle$	$BCDEF$	$\langle 2, 10 \rangle$
$ABCDEF$	$\langle 1, 10 \rangle$		

(c)

TAB. 1 – Un jeu de données transactionnel (a). Une valeur est associée à chaque item. Examples de mesures (b). Les motifs fréquents et clos par rapport à $M = \{sup, \min\}$ et leurs valeurs pour $\theta = 1$ (c).

2.2 Fouille de skypatterns

Définition 3 (Dominance Pareto) *Soit $M = \{m_1, \ldots, m_n\}$ un ensemble de n mesures et $N = \{1, .., n\}$ un ensemble d'indices. Un motif P est caractérisé par un vecteur d'utilité $u(P) = (m_1(P), ..., m_n(P)) \in \mathbb{R}^n$. On compare généralement les vecteurs d'utilité à l'aide d'une relation de dominance Pareto ($\mathcal{P}$-dominance). La weak-$\mathcal{P}$-dominance $\succsim_\mathcal{P}$ entre deux motifs P, P' est définie par : $P \succsim_\mathcal{P} P' \Leftrightarrow [\forall i \in N, m_i(P) \geq m_i(P')]$, tandis que la strict $\mathcal{P}$-dominance $\succ_\mathcal{P}$ entre P et P' est définie par : $P \succ_\mathcal{P} P' \Leftrightarrow [P \succsim_\mathcal{P} P' \wedge not(P' \succsim_\mathcal{P} P)]$.*

Un motif P est Pareto-optimal (a.k.a *Skypattern*) ssi il n'existe pas de motif Q qui domine P.

Définition 4 (Archive) *Une archive $\mathcal{A}$ est un ensemble de solutions tel qu'il n'y a pas de solution dans $\mathcal{A}$ qui domine une autre solution dans l'archive : $\nexists y, y' \in \mathcal{A} : y \succ y'$.*

Exemple 2 *Considérons l'exemple dans la table 1a avec M=$\{sup, min\}$. Le motif E domine le motif B par rapport à M car $sup(B)$=$sup(E)$=6 et $min(E.val)$>$min(B.val)$.*

Définition 5 (Opérateur Sky) *Étant donné un ensemble de motifs $\mathcal{S} \subseteq \mathcal{L}_\mathcal{I}$ et un ensemble de mesures M, un skypattern de $\mathcal{S}$ par rapport à M est un motif de $\mathcal{S}$ qui n'est pas dominé par rapport à M. L'opérateur de motifs Pareto Sky($\mathcal{S}$,M) retourne tous les skypatterns de $\mathcal{S}$ par rapport à M : Sky($\mathcal{S}$,M) $= \{P \in \mathcal{S} \mid \nexists Q \in \mathcal{S}, Q \succsim_\mathcal{P} P\}$.*

Le problème de fouille de skypatterns peut être formulé ainsi : *Étant donné un ensemble de mesures M, le problème consiste à évaluer la requête $Sky(\mathcal{L}_\mathcal{I}, M)$.* Le problème de fouille de skypatterns est difficile en raison du nombre exponentiel de candidats potentiels (i.e. $|\mathcal{L}_\mathcal{I}|$) Yang (2004). Pour réduire le coût d'évaluation de la requête $Sky(\mathcal{L}_\mathcal{I}, M)$, nous proposons d'appliquer l'opérateur Sky sur un ensemble réduit mais pertinent de motifs $S \subseteq \mathcal{L}_\mathcal{I}$ qui contient tous les motifs Pareto, *i.e.* $S \subseteq Sky(\mathcal{L}_\mathcal{I}, M)$. Les représentations condensées peuvent être utilisées pour réduire le temps de calcul sans perte de précision.

Skylinéabilité. Bien que les représentations condensées réduisent le temps de calcul, pour certaines mesures, telles que $area$ ou $size$, la représentation condensée est égale à $\mathcal{L}_\mathcal{I}$. Par conséquent, calculer une représentation condensée pour chaque mesure $m \in M$ rendrait le processus d'extraction non efficace. Pour résoudre ce problème, Soulet et al. (2011) ont proposé la notion de *skylinéabilité*. L'idée majeure est de trouver un sous ensemble de mesures $M' \subseteq M$ tel que les motifs Pareto par rapport à M peuvent être récupérés à partir de la représentation condensée par rapport à M'. Un opérateur, noté $\bar{c}$, permettant d'obtenir M' à partir

Trans.	Items						
t_1	A			D		F	
t_2	A				E	F	
t_3	A				E		G
t_4	A				E		G
t_5		B			E		G
t_6		B			E		G
t_7			C		E		G
t_8			C		E		G
t_9			C		E		H
t_{10}			C		E		H
t_{11}			C			F	G	H

(a) La base de transactions $\mathcal{T}$.

Sol.	P_1	P_2	P_3
s_1	{C, F, G, H}	{E}	{A, D, F}
s_2	{A, F}	{C, H}	{E, G}
s_3	{A}	{B, E, G}	{C}

(b) Ensemble de clusterings conceptuels pour k=3 extraits à partir de la base de transactions de la table 2a.

Trans.	Items	
t_1	A	B
t_2	A	B
t_3	A	B
t_4		C
t_5		C

Item	val
A	10
B	20
C	30

(c) Exemple de motivation pour le clustering conecptuel à base de motifs clos sur plusieurs mesures (cf. section 5).

TAB. 2 – Exemple d'illustration.

de M est introduit dans Soulet et al. (2011). Il retourne un ensemble M' qui garantit que pour tout motif $P \subset Q$, si $P =_{M'} Q$, alors $Q \succeq_M P$ (voir (Ugarte et al., 2017)).

2.3 Programmation par contraintes

La programmation par contraintes (PPC) offre une approche générique pour modéliser les problèmes combinatoires. Un modèle PPC consiste en un ensemble de variables $X = \{x_1, \ldots, x_n\}$, un ensemble de domaines finis D pour chaque variable $x_i \in X$, et un ensemble de contraintes $\mathcal{C}$ sur X. Une contrainte $c \in \mathcal{C}$ est une relation qui spécifie les combinaisons autorisées de valeurs pour les variables $X(c)$. Une instanciation d'un sous-ensemble de variables $Y \subseteq X$ est une affectation de valeurs $v \in dom(x_i)$ à chaque variable x_i. Une solution est une instanciation de X satisfaisant toutes les contraintes $\mathcal{C}$. Les solveurs de contraintes utilisent des méthodes de recherche par retour-arrière pour explorer l'espace de recherche. Le concept principal utilisé pour accélérer la recherche est la propagation de contraintes à l'aide d'*algorithmes de filtrage*. En effet, à chaque instanciation d'une variable, l'algorithme de filtrage réduit l'espace de recherche tout en garantissant certaines propriétés 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.

Contrainte globale CLOSEDPATTERNS. La majorité des méthodes déclaratives utilisent un vecteur x de variables booléennes $(x_1, \ldots, x_{|\mathcal{I}|})$ pour représenter les motifs, où x_i représente la présence de l'item $i \in \mathcal{I}$ dans le motif. Nous utiliserons la notation suivante : $x^+ = \{i \in \mathcal{I} \,|\, dom(x_i) = \{1\}\}$, $x^- = \{i \in \mathcal{I} \,|\, dom(x_i) = \{0\}\}$ and $x^* = \{i \in \mathcal{I} \,|\, i \notin x^+ \cup x^-\}$.

Définition 6 (CLOSEDPATTERNS) *Soit x un vecteur de variables booléennes, θ un support minimal et $\mathcal{D}$ un jeu de données. La contrainte globale* CLOSEDPATTERNS$_{\mathcal{D},\theta}(x)$ *est respectée ssi x^+ est clos par rapport à $\{sup\}$ et fréquent par rapport à θ.*

2.4 Clustering conceptuel

Le clustering est une tâche importante en fouille de données dont l'objectif est de partitionner un ensemble de données (transactions) en groupes (clusters) d'une manière telle que les transactions appartenant à un même cluster soient similaires mais différentes de celles appartenant aux autres. Le **clustering conceptuel** consiste à fournir une description distincte de chaque cluster, c. à. d. le concept caractérisant l'ensemble des transactions qu'il contient. Les

motifs fermés sont de bons de candidats pour la recherche de clusterings à base d'associations. Ce problème peut être formulé de la façon suivante : trouver un ensemble de k motif fermés P_1, P_2, ..., P_k (i.e., clusters) couvrant toutes les transactions sans chevauchement des couvertures respectives de ces motifs. Par exemple, la table 2b illustre les différents clusterings conceptuels pour $k = 3$. Une fonction d'évaluation f est nécessaire pour évaluer la qualité d'un clustering. Ainsi, le clustering conceptuel cherche un ensemble disjoint de clusters sur $\mathcal{T}$ qui optimisent un certain critère défini par la qualité du clustering. Des fonctions d'évaluation typiques sont par exemple $minFreq$ et $minSize$, où $minFreq = min_{i=1}^{k} sup(P_i)$ et $minSize = min_{i=1}^{k} size(P_i)$. Par exemple, pour la base de transactions de la table 2a et $k = 3$, maximiser $minSize$ fournit un clustering s_2 avec une valeur optimale 2.

3 La contrainte globale ADEQUATECLOSURE

Cette section présente une nouvelle contrainte globale ADEQUATECLOSURE pour la fouille de motifs fréquents et fermés par rapport à un ensemble de mesures préservantes M.

Définition 7 (ADEQUATECLOSURE) *Soit x un vecteur de variables booléennes, f et f_1 deux variables entières, θ un support minimum, $\mathcal{D}$ un jeu de données transactionnel, et M un ensemble de mesures préservantes. La contrainte globale* $\text{ADEQUATECLOSURE}_{\mathcal{D},M,\theta}(x, f, f_1)$ *est respectée ssi* $\texttt{clos}_M(x^+) = x^+$ *et x^+ est frequent par rapport à θ (i.e. $f \geq \theta$).*

Les variables f et f_1 permettent de stocker les valeurs de $sup(x^+)$ et $sup_{\mathcal{D}_1}(x^+)$. Elles sont utilisées pour imposer des contraintes sur le support et le taux de croissance du motif. Nous introduisons l'opérateur d'inclusion de fermeture $\texttt{cl}_{inc}$ qui est utilisé par notre contrainte globale pour la fouille de représentations condensées adéquates par rapport à M. Dans le cas où la mesure de croissance n'est pas spécifiée dans la liste M, alors la contrainte globale est réduite à $\text{ADEQUATECLOSURE}_{\mathcal{D},M,\theta}(x, f)$.

Définition 8 (Closure inclusion) *Soit x une instanciation partielle des variables $x_1, \ldots, x_{|\mathcal{I}|}$, M un ensemble de mesures préservantes et i un item libre (i.e. $i \in x^*$).* $\texttt{cl}_{inc}(x^+, i, M)$ *retourne* **vrai** *ssi* $\forall m \in M, m(x^+ \cup \{i\}) = m(x^+)$*, i.e.* $\texttt{cl}_{inc}(x^+, i, M) \Leftrightarrow i \in \texttt{clos}_M(x^+)$.

Le lemme 1 caractérise une instanciation partielle cohérente par rapport à la contrainte ADEQUATECLOSURE, c'est à dire une instanciation partielle qui peut être étendue à une instanciation complète qui satisfait la contrainte.

Lemme 1 (Instanciation partielle cohérente) *Soit x une instanciation partielle des variables $x_1, \ldots, x_{|\mathcal{I}|}$ et M un ensemble de mesures préservantes. x est une instanciation partielle cohérente ssi x^+ est fréquent par rapport à θ et $\nexists j \in x^-$ tel que* $\texttt{cl}_{inc}(x^+, j, M)$ *est vérifiée.*

Preuve 1 *Si $sup(x^+) < \theta$, x^+ ne peut pas être étendu à un motif fréquent par rapport à θ. Considérons une instanciation partielle x et $j \in x^-$ tel que* $\texttt{cl}_{inc}(x^+, j, M)$ *soit vrai, ce qui signifie que $j \in \texttt{clos}_M(x^+)$. Comme x^+ ne peut pas être étendu à un motif clos sans ajouter j (j appartenant à x^-), le résultat suit.* □

Proposition 3 (Règles de filtrage de ADEQUATECLOSURE) *Étant donné une instanciation partielle cohérente x, un ensemble de mesures préservantes M, pour tout $i \in x^*$, les règles $(1-3)$ suppriment les valeurs inconsistantes de $dom(x_i)$:* (1) *si* $\texttt{cl}_{inc}(x^+, i, M) \Rightarrow 0 \notin dom(x_i)$*;* (2) *si* $|\boldsymbol{t}_{\mathcal{D}}(x^+ \cup \{i\})| < \theta \Rightarrow 1 \notin dom(x_i)$*;* (3) *si* $\exists j \in x^-$ *s.t.* $\texttt{cl}_{inc}(x^+ \cup \{i\}, j, M) \Rightarrow 1 \notin dom(x_i)$.

Preuve 2 *Soit x une instanciation partielle cohérente et i un item libre.*
1) Supposons que $\texttt{cl}_{inc}(x^+, i, M)$ est vrai. Il suit que $i \in \texttt{clos}_M(x^+)$. A partir du Lemme 1, nous savons que x^+ ne peut pas être étendu à un motif fermé par rapport à M si $i \in x^-$. Donc $0 \notin dom(x_i)$.
2) Si $|\boldsymbol{t}(x^+ \cup \{i\})| < \theta$, alors (Lemme 1) $x^+ \cup \{i\}$ ne peut pas être étendu à un motif fermé par rapport à M et donc $1 \notin dom(x_i)$.
*3) (***preuve par l'absurde***) Supposons que $j \in x^-$ tel que $\texttt{cl}_{inc}(x^+ \cup \{i\}, j, M)$, ce qui signifie que $j \in \texttt{clos}_M(x^+ \cup \{i\})$. Comme $j \in x^-$, ce qui signifie que $j \notin \texttt{clos}_M(x^+)$, il suit que $j \notin \texttt{clos}_M(x^+ \cup \{i\})$.* □

Notre contrainte globale ADEQUATECLOSURE propage à partir des variables booléennes aux variables entières qui représentent la fréquence d'un motif dans les jeux de données $\mathcal{D}$ et $\mathcal{D}_1$. Ainsi, deux autres règles similaires à (Schaus et al., 2017) sont appliquées pour mettre à jour les bornes de f et f_1 :

(4) règles UB : $\begin{cases} si\ |\boldsymbol{t}_{\mathcal{D}}(x^+)| < UB(f) \Rightarrow UB(f) \leq |\boldsymbol{t}_{\mathcal{D}}(x^+)| \\ si\ |\boldsymbol{t}_{\mathcal{D}_1}(x^+)| < UB(f_1) \Rightarrow UB(f_1) \leq |\boldsymbol{t}_{\mathcal{D}_1}(x^+)| \end{cases}$

(5) règles LB : $\begin{cases} si\ |\boldsymbol{t}_{\mathcal{D}}(x^+ \cup x^*)| > LB(f) \Rightarrow LB(f) \geq |\boldsymbol{t}_{\mathcal{D}}(x^+ \cup x^*)| \\ si\ |\boldsymbol{t}_{\mathcal{D}_1}(x^+ \cup x^*)| > LB(f_1) \Rightarrow LB(f_1) \geq |\boldsymbol{t}_{\mathcal{D}_1}(x^+ \cup x^*)| \end{cases}$

Pour connecter ensemble les variables f et f_1, nous définissons une nouvelle contrainte indépendante de la contrainte ADEQUATECLOSURE, qui s'exprime par $f_2 = f - f_1$, où f_2 est une variable entière qui représente la taille de la couverture des motifs qui sont contenus dans le jeu de données $\mathcal{D} \setminus \mathcal{D}_1$. La contrainte sur la variable f_2 est activée seulement si le taux de croissance apparaît dans M. Dans ce cas, les bornes de la variable f_1 sont mis à jour.

Exemple 3 *Soit le jeu de données de la table 1a avec $M = \{sup, min\}$, $\theta = 2$, et $dom(f) = \{0, \ldots, 6\}$. La contrainte $f \geq \theta$ met à jour le minorant de f à 2, i.e. $LB(f) = 2$. Soit l'instanciation partielle $x^+ = \emptyset$, $x^- = \{E\}$ and $x^* = \{A, B, C, D, F\}$. Graçe à la règle (3), la valeur 1 est filtrée de $dom(x_A)$ car $E \in \texttt{clos}_M(x^+ \cup \{A\})$ et $E \in x^-$. Soit l'instanciation partielle $x^+ = \{A, B, C, D\}$, $x^- = \emptyset$, $x^* = \{E, F\}$, $LB(f) = 2$ et $UB(f) = 6$, la règle (1) filtre la valeur 0 de $dom(x_E)$ car $\texttt{cl}_{inc}(ABCD, E, M)$ est vrai, i.e. $E \in \texttt{clos}_M(ABCD)$. La règle (2) filtre la valeur 1 de $dom(x_F)$ car $\boldsymbol{t}(ABCDF) = 1 < \theta$. Finalement, la règle (4) met à jour le majorant de f à $|\boldsymbol{t}(ABCDE)|$, i.e. $UB(f) = 2$.*

L'algorithme 1 montre la propagation de ADEQUATECLOSURE. Il prend en entrée le jeu de données transactionnel $\mathcal{D}$, les variables d'items x, les deux variables entières f et f_1, le support minimum θ et l'ensemble de mesures M. Il commence par calculer la couverture du motif x^+ et vérifie si l'instanciation partielle actuelle est inconsistante (Lemme 1), c'est à dire, si x^+ est soit infréquent (ligne 2) ou x^+ ne peut pas être étendu à un motif clos par rapport à M sans ajouter i ($i \in x^-$) (ligne 3), dans ce cas la contrainte n'est pas respectée et on retourne un échec. L'algorithme 1 supprime les items $i \in x^*$ qui ne peuvent pas

Algorithme 1 : Filtrage pour ADEQUATECLOSURE

```
   Input : 𝒟 : base transactionnelle ; θ : support minimal ; M : ensemble de mesures ;
   InOut : x = {x_1...x_n} : Variables d'items booléennes ; f, f_1 : Variables entières ;
1  begin
2      if |t_𝒟(x^+)| < θ then return faux ;
3      if ∃ i ∈ x^- s.t. closureInclusion(x^+, i, M) then return faux ;
4      foreach i ∈ x^* do
5          if |t_𝒟(x^+ ∪ {i})| < θ then
6              dom(x_i) ← dom(x_i) − {1} ; x^- ← x^- − ∪{i} ; x^* ← x^* \ {i} ;
7          else if closureInclusion(x^+, i, M) then
8              dom(x_i) ← dom(x_i) − {0} ; x^+ ← x^+ ∪ {i} ; x^* ← x^* \ {i} ;
9      end
10     foreach j ∈ x^- do
11         foreach i ∈ x^* do
12             if closureInclusion(x^+ ∪ {i}, j, M) then
13                 dom(x_i) ← dom(x_i) − {1} ; x^- ← x^- ∪ {i} ; x^* ← x^* \ {i} ;
14             end
15         end
16     end
17     updateBounds(f, |t_𝒟(x^+ ∪ x^*)|, |t_𝒟(x^+)|);
18     updateBounds(f_1, |t_𝒟_1(x^+ ∪ x^*)|, |t_𝒟_1(x^+)|);
19     return vrai ;
20 end
21 Function closureInclusion(x,i,M) : Boolean
22     foreach m ∈ M do
23         if m(x ∪ {i}) ≠ m(x) then return faux;
24     end
25     return vrai ;
```

appartenir à une solution qui contient x^+. Pour cela, nous testons en premier si $x^+ \cup \{i\}$ est infréquent par rapport à θ (ligne 5). Si c'est le cas, nous supprimons 1 de $dom(x_i)$ et nous mettons à jour x^- et x^* (ligne 6). Ensuite, pour chaque mesure $m \in M$, la fonction $\texttt{closureInclusion}(x^+, i, M)$ vérifie si ajouter l'item i ne modifie pas la valeur de m pour la spécialisation x^+ (lignes 22-23). Si c'est le cas, la fonction retourne **vrai** (ligne 25), supprime 0 de $dom(x_i)$ et met à jour les ensembles x^+ et x^* (ligne 8). Troisièmement, en appliquant la règle (3), nous supprimons 1 du domaine de chaque variable d'item $i \in x^*$ tel que $x^+ \cup \{i\}$ ne peut pas être étendu à un motif clos par rapport à M sans ajouter un item absent $j \in x^-$ (lignes 10-16). Enfin, nous mettons à jour les bornes des variables f et f_1 en appliquant les règles (4) et (5).

Proposition 4 (Complexité de l'algorithme 1) *Étant donné une base de données transactionnelle $\mathcal{D}$ qui contient n items et m transactions, un support minimal θ et un ensemble de mesures M qui contient c mesures basées sur sup. L'algorithme 1 assure le filtrage des domaines des variables en temps $\mathcal{O}(n^2mc)$.*

Preuve 3 (Complexité de l'algorithme 1) *Soit n le nombre d'items et m le nombre de transactions dans la base de données. Calculer $m(x^+)$ nécessite au plus $O(nm)$ pour chaque mesure basée sur sup (telle que sup ou $sup_\vee$) et $O(n)$ pour chaque mesure basée sur des valeurs (comme min, max ou sum). Vérifier les règles 1 et 2 nécessite $O(nm)$ pour chaque*

mesure basée sur sup et $O(n)$ pour chaque mesure basée sur des valeurs. Enfin, vérifier la règle 3 nécessite $O(n^2m)$ pour chaque mesure basée sur sup et $O(n^2)$ pour chaque mesure basée sur des valeurs. Par conséquent, si c est le nombre de mesures basées sur sup dans M, la complexité dans le pire des cas est $O(n^2mc)$. □

Remarque 1 *Dans (Vernerey et al., 2022, 2023), nous avons proposé une version décomposée et efficace de la contrainte globale* AdequateClosure *permettant de déléguer la gestion des règles de filtrage (2), (4) et (5) à la contrainte globale* CoverSize *(Schaus et al., 2017), qui permet de modéliser la mesure $sup(x)$, i.e. $f = |\boldsymbol{t}(x^+)|$ et la contrainte de support minimale (cf. la règle (2). Ainsi, la gestion des variables supplémentaires f et f_1 n'est plus gérée dans la signature de* AdequateClosure *mais plutôt grâce à deux contraintes additionnelles* CoverSize $_{\mathcal{D}}(x, f)$ *et* CoverSize $_{\mathcal{D}_1}(x, f_1)$ *respectivement. Cela permet de tirer parti de la déclarativité de la PPC en réutilisant des contraintes existantes.*

4 Fouille de Skypatterns avec AdequateClosure

Cette section décrit un premier cas d'usage de notre contrainte AdequateClosure pour la découverte de skypatterns. L'idée principale est d'utiliser une archive $\mathcal{A}$ de skypatterns pour supprimer les solutions qui sont dominées par au moins une solution de $\mathcal{A}$ (voir la définition 4). Soit $M = \{m_1, \ldots, m_l\}$ un ensemble de mesures à maximiser. Le modèle ClosedSky$_{\mathcal{D},M,\mathcal{A}}(x, obj)$ est donné comme suit :

$$\text{ClosedSky}_{\mathcal{D},M,\mathcal{A}}(x, obj) \equiv \begin{cases} \text{Pareto}_{\mathcal{A}}(obj) & (1) \\ \text{AdequateClosure}_{\mathcal{D},M',\theta}(x, f, f_1) & (2) \\ obj_i = m_i(x),\ i = 1..l & (3) \end{cases}$$

(a) Variables. Notre modèle contient : (i) n *variables booléennes* x modélisant le motif inconnu, où x_i représente la présence de l'item $i \in \mathcal{I}$ dans le motif. (ii) l *variables objectives* : obj est un vecteur de variables entières, t.q. obj_i représente la valeur d'une mesure $m \in M$. (iii) *Couverture de* x : nous utilisons deux variables entières f et f_1 pour stocker les valeurs de $sup(x)$ et $sup_{\mathcal{D}_1}(x)$.

(b) Contraintes. Notre modèle exploite un ensemble de contraintes pour lesquelles des algorithmes de filtrage efficaces existent.
- **Contrainte (1)** est une contrainte d'optimisation globale, qui est utilisée pour extraire des skypatterns sans ajout de contraintes dynamiques supplémentaires (contrairement au modèle de (Ugarte et al., 2017)). Formellement, $\text{Pareto}_{\mathcal{A}}(obj) \equiv \bigwedge_{y \in \mathcal{A}} \bigvee_{i=1..l} obj_i > y_i$. Elle impose que le vecteur objectif suivant $obj = (obj_1, ..., obj_l)$ ne soit pas dominé par rapport à l'archive $\mathcal{A}$, c'est-à-dire $\nexists y \in \mathcal{A} : y \succ obj$. Cette contrainte est détaillée dans Schaus et Hartert (2013). Contrairement au travail de (Ugarte et al., 2017), il n'est pas nécessaire d'avoir une deuxième étape de traitement des motifs, car tous les non skypatterns sont systématiquement filtrés.

- **Contrainte (2)** est introduite comme une nouvelle contrainte globale pour s'assurer que le motif x soit fermé par rapport à un ensemble de mesures condensables M'. Elle permet d'extraire des représentations condensées de motifs (sans utiliser des contraintes réifiées). M' est automatiquement calculé tel que M est maximalement M'-skylineable [1]. Les règles de

1. L'ensemble de mesures M' est calculé à partir de M avec l'opérateur de conversion $\bar{c}$ (voir Soulet et al. (2011)).

filtrage de cette contrainte sont données dans la section 3. Les deux variables f et f_1 sont utilisées pour imposer des contraintes sur les mesures de fréquence et de taux de croissance d'un ensemble d'un motif. Par exemple, pour imposer que x doit être fréquent par rapport à θ, on peut simplement ajouter la contrainte $f \geq \theta$.

- **Contraintes (3)** sont utilisées pour contraindre chaque variable obj_i à être égale à la valeur de la i^{eme} mesure, i.e. $obj_i = m_i(x),\ i \in [1, l]$. Elles impliquent également la définition de chaque mesure $m \in M$ (éventuellement via des contraintes globales existantes). Par exemple, si $M = \{sup, area\}$, alors $obj_1 = sup$ (i.e. $obj_1 = |\boldsymbol{t}(x)|$) et $obj_2 = obj_1 \times \sum_{i \in \mathcal{I}} x_i$ (i.e. modélise la mesure *area* comme le produit de son support (stocké dans obj_1) par sa taille).

Heuristique de branchement. Pour ordonner les variables, nous choisissons l'item libre i (*i.e.* $i \in x^*$) tel que $|\boldsymbol{t}_{\mathcal{D}}(x^+ \cup \{i\})|$ est minimal. Cette heuristique nous permet d'activer le plus tôt possible nos règles de filtrage (voir l'algorithme 1), donc de réduire l'espace de recherche.

5 Modèle PPC pour le clustering conceptuel condensable

Dans cette section, nous proposons un modèle PPC, basé sur approche en deux étapes, pour faire du clustering conceptuel en exploitant des représentations condensées selon des mesures préservées M'. Nous commençons par un exemple de motivation pour montrer l'intérêt de ce type de motifs pour le clustering conceptuel.

Exemple de motivation. Considérons la base de transactions de la table 2c, nous avons un seul clustering conceptuel possible, $\{AB, C\}$, qui a une $minFreq$ de 2 et une $minSize$ de 1. L'inconvénient des motifs fermés uniquement par rapport à sup est qu'ils ne prennent pas en compte toutes les informations sur les motifs. Par exemple, pour la base de transactions de la table 2c, le motif B n'est pas fermé par rapport à sup (car $sup(B) = sup(AB)$) mais il l'est par rapport à $\{sup, mean\}$ (car $mean(B) > mean(AB)$). Un tel motif peut s'avérer intéressant si l'on cherche un clustering qui maximise une certaine mesure de qualité qui dépend par exemple des valeurs numériques associées aux items. Soit $minMean = min_{i=1}^{k} mean(P_i.val)$. Ainsi, avec un clustering issu des motifs fermés par rapport à sup, nous avons $minMean = 15$ car $mean(AB) = \frac{10+20}{2} = 15$ et $mean(C) = 30$. En considérons les motifs fermés par rapport à $\{sup, mean\}$, il existe un autre clustering conceptuel plus intéressant $\{B, C\}$ avec $minMean = 20$ car $mean(B) = 20$.

Ainsi, l'idée clé est de faire du clustering conceptuel biaisé selon M'. L'avantage est d'exploiter les représentations condensées selon les mesures préservées M' pour extraire des descriptions (i.e. concepts) permettant de décrire des clsuters portant sur plusieurs mesures, ce qui permet de trouver des corrélations entre mesures tout en identifiant des clusterings plus intéressants, comme le montre l'exemple de motivation.

Soit I un ensemble d'items et T un ensemble de transactions. Une paire (I, T) telle que $\boldsymbol{t}_{\mathcal{D}}(I) = T$ et $\boldsymbol{i}(T) = I$ est appelée **concept formel** (ou motif fermé). L'ensemble de tous les concepts formels est noté $\mathcal{F}$. L'objectif du clustering conceptuel consiste à déterminer l'ensemble des k motifs $\mathcal{CC} = \{(I_1, T_1), ..., (I_k, T_k)\} \subseteq \mathcal{F}$, tel que $\{T_1, ..., T_k\}$ forme une *partition* de l'ensemble $\mathcal{T}$ des transactions. Notre approche, notée CONDENSED-CC, est basée sur un modèle PPC hybride (en deux étapes) proposé par (Chabert et Solnon, 2017), où le nombre de clusters n'est pas fixé à l'avance :

$$\text{CONDENSED-CC}_{\mathcal{D},M,\mathcal{A}}(x,\Phi,k,obj) \equiv \begin{cases} \mathbf{Etape_1} : \text{ADEQUATECLOSURE}_{\mathcal{D},M',\theta}(x,f) & (1) \\ \mathbf{Etape_2} : \text{PARETO}_{\mathcal{A}}(obj) & (2) \\ \qquad\qquad \wedge\ \text{HYBRID-CC}_{\mathcal{D},\mathcal{F}}(\Phi,k,obj) & (3) \end{cases}$$

(a) Variables. (i) une variable entière k qui modélise le nombre de clusters, avec $D(k) = [k_{min}, k_{max}]$, où k_{min} et k_{max} deux bornes données t.q. $2 \leq k_{min} \leq k_{max} < |\mathcal{T}|$. (ii) *une variable ensembliste* Φ qui modélise le sous ensemble de concepts définissant un clustering conceptuel optimal, avec $D(\Phi) = \mathcal{P}(\mathcal{F})$. (iii) *un vecteur de variables entières* $Concept$ qui représente, pour chaque transaction $t \in \mathcal{T}$, le concept $Concept_t$ qui couvre t avec $D(Concept_t) = \{P \in \mathcal{F} \mid t \in \boldsymbol{t}_{\mathcal{D}}(P)\}$. (iv) *un ensemble de variables objectifs* où chaque variable obj_i est associé au i^e critère objectif.

(b) Contraintes. Nous détaillons ci-après les contraintes par rapport à chaque étape :

- $\mathbf{Etape_1}$: Contrairement à l'approche hybride de (Chabert et Solnon, 2017) qui utilise LCM pour extraire tous les motifs fermés selon sup, cette étape se base sur la contrainte ADEQUATECLOSURE pour générer l'ensemble de tous les motifs fermés $\mathcal{F}$ par rapport à M'.

- $\mathbf{Etape_2}$: Cette étape permet de modéliser le problème du clustering conceptuel en exploitant les *concepts* $\mathcal{F}$ extraits à l'étape précédente et en optimisant plusieurs critères objectifs. Ce modèle comprend les contraintes suivantes :
  - la contrainte de **Pareto**, détaillée en section 4, pour extraire les solutions non-dominées.
  - les contraintes du clustering conceptuel (**Hybrid-CC**). Deux types de contraintes :
    (i) une contrainte qui impose que chaque transaction $t \in \mathcal{T}$ soit couverte par un concept $Concept_t$ appartenant l'ensemble des motifs du clustering optimal Φ;
    (ii) une contrainte qui assure que les concepts sélectionnés forment une *partition* de $\mathcal{T}$ (sans chevauchement) : chaque $t \in \mathcal{T}$ est couverte par exactement un seul concept de Φ.

(c) Fonctions objectives. Pour extraire les solutions du Front de Parero, nous avons utilisé deux fonctions : $minFreq$ et $minMean$.

6 Travaux connexes

Le concept de **représentations condensées** a été largement adopté dans plusieurs approches de fouille de données (Calders et al. (2004); Mannila et Toivonen (1997)). Soulet et Crémilleux (2008) ont proposé MICMAC pour la fouille de représentations condensées adéquates par rapport à un ensemble de mesures. Néanmoins, cette méthode présente un problème de passage à l'échelle. Ugarte et al. (2017) ont proposé un modèle PPC réifié pour encoder la contrainte de fermeture pour certaines mesures (sup, $\min$ et $\max$). Notre contrainte globale ne nécessite ni contraintes réifiées ni variables auxiliaires.

Calcul des skypatterns. Soulet et al. (2011) ont proposé AETHERIS, qui procède en deux étapes. Premièrement, les représentations condensées par rapport à un ensemble de mesures M' sont extraites. Ensuite, l'opérateur Sky est appliqué. Ugarte et al. (2017) ont aussi proposé une approche en deux étapes (intitulée CP+SKY), mais contrairement à AETHERIS, CP+SKY construit dynamiquement une représentation condensée grâce à des contraintes sur la relation de dominance,qui sont ajoutées pendant le processus de fouille. Notre approche ne nécessite

pas d'étape de post-traitement et exploite une contrainte globale afin d'extraire efficacement les représentations condensées de motifs.

Clustering conceptuel. Plusieurs approches déclaratives utilisant la PPC (Chabert et Solnon, 2017) ou la programmation linéaire (PLNE) Ouali et al. (2016) ont étés développées pour le clustering conceptuel. La plupart de ces approches combinent deux étapes : dans une première étape, un outil de fouille dédié (i.e., LCM) est utilisé pour calculer l'ensemble de tous les concepts formels et, dans une deuxième étape, la PPC ou la PLNE est utilisée pour sélectionner les meilleurs concepts optimisant un critère donné. Notre travail exploite la représentation condensées de motifs sur plusieurs mesures pour extraire des clusterings conceptuels.

7 Expérimentations

Nous avons mené des expérimentations pour répondre aux questions suivantes : (1) Quelles sont les performances (en temps CPU) de notre contrainte globale (noté ADEQUATE-CI) comparé à CP+CLOSED et MICMAC pour la fouille de motifs clos ? CP+CLOSED utilise la première étape de CP+SKY (modèle réifié). (2) Quelles sont les performances (en temps CPU) de notre approche (noté CLOSEDSKY) comparé à CP+SKY et AETHERIS pour la fouille de skypatterns ? (3) Quel est le nombre de skypatterns comparé au nombre de motifs clos ? (4) Comment les clusterings conceptuels basés sur les motifs fermés selon plusieurs mesures et selon la fréquence se comparent qualitativement ?

7.1 Protocole expérimental

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%). L'implémentation a été réalisée avec `Choco` (Prud'homme et al., 2016) version $4.10.5$, une librairie Java pour la programmation par contraintes. Nous avons utilisé les versions d'AETHERIS et MICMAC fournies par les auteurs (implantées en C++). Les expérimentations ont été menées sous un AMD Opteron 6174, 2.2 GHz avec une RAM de 256 Go et une limite de temps de 24 heures. La maximum heap size autorisée par la JVM est 30 Go. Nous considérons les ensembles de mesures suivants pour la fouille de motifs clos : $AC_1 : \{min(P.val), sup(P) \geq \theta\}$, $AC_2 : \{max(P.val), sup(P) \geq \theta\}$ et $AC_3 : \{min(P.val), max(P.val), sup(P) \geq \theta\}$. Les mesures qui utilisent des valeurs numériques, comme min ou max, sont appliquées à des valeurs générées aléatoirement dans l'intervalle $[0, 1]$.

7.2 Comparaison entre ADEQUATE-CI, CP+CLOSED et MICMAC

La table 3 compare les performances de ADEQUATE-CI, CP+CLOSED et MICMAC pour différentes valeurs de θ pour différents jeux de données et ensembles de mesures. Pour chaque méthode, nous reportons le temps CPU (en secondes), le nombre de motifs clos et le nombre de noeuds explorés. Concernant les temps d'exécution, ADEQUATE-CI arrive à terminer l'exécution sur toutes les instances contrairement aux autres méthodes qui obtiennent soit *Out Of Memory* ou *Time Out*. Sur 31 instances, MICMAC obtient 10 `OOM` et 1 `TO`, CP+CLOSED

Dataset $\lvert\mathcal{I}\rvert \times \lvert\mathcal{T}\rvert$	θ	#Motifs	#Nœud		Temps (s)		
		(1)(2)(3)	(1)	(2)	(1)	(2)	(3)
CHESS	0.3	730,6791	**14,613,581**	14,890,173	**1545**	50751	6283
75 × 3,196	0.2	30,120,283	**60,240,565**	-	**6536**	TO	26792
	0.1	153,073,913	**306,147,825**	-	**36606**	TO	TO
CONNECT	0.18	323,3691	**6,467,381**	-	**6884**	TO	OOM
129 × 67,557	0.15	5,084,539	**10,169,077**	-	**11159**	TO	OOM
	0.1	11,903,644	**23,807,287**	-	**26979**	TO	OOM
HEART-CLEVELAND	0.1	14,126,585	**28,253,169**	31,283,345	3297	10264	**1352**
95 × 296	0.08	26,812,645	**53,625,289**	58,338,937	6251	19423	**3096**
	0.06	53,854,923	**107,709,845**	114,916,691	12738	39720	**7473**
HEPATITIS	0.2	38,6831	**773,661**	847,343	60	159	**24**
68 × 137	0.1	1,949,759	**3,899,517**	4,115,027	333	799	**127**
	0.05	4,196,027	**8,392,053**	8,713,651	723	1675	**315**
KR-VS-KP	0.3	4,501,990	**9,003,979**	9,191,063	**997**	32293	5391
73 × 3,196	0.2	17,825,411	**35,650,821**	-	**3848**	TO	22417
MUSHROOM	0.01	39,676	**79,351**	130,235	25	2227	**11**
112 × 8,124	0.008	48,601	**97,201**	156,131	29	2595	**12**
	0.005	63,914	**127,827**	204,137	42	3263	**13**
PUMSB	0.8	46,495	**92,989**	-	**440**	TO	660
2,113 × 49,046	0.7	358,767	**717,533**	-	**3485**	TO	12015
SPLICE1	0.1	1,580	**3,159**	60,871	5	3043	2
287 × 3,190	0.05	30,473	**60,945**	2,541,737	155	51671	**41**
	0.02	565,780	**1,131,559**	-	1283	TO	**308**
T10I4D100K	0.005	1,074	**2,147**	-	**657**	TO	OOM
870 × 100,000	0.0025	7654	**15,307**	-	**1490**	TO	OOM
T40I10D100K	0.08	138	**275**	-	**13**	TO	OOM
942 × 100,000	0.05	317	**633**	-	**101**	TO	OOM
	0.01	65,237	**130,473**	-	**8801**	TO	OOM
BMS1	0.001	3,977	**7,953**	141,199	167	56872	**131**
497 × 59,602	0.0005	178,468	**356,935**	-	2067	TO	**559**
RETAIL	0.004	832	**1,663**	-	**185**	OOM	OOM
16470 × 88,162	0.002	2,692	**5,383**	-	**3428**	OOM	OOM

(a) AC_1

θ	#Motifs	#Nœud		Temps (s)		
	(1)(2)(3)	(1)	(2)	(1)	(2)	(3)
0.3	6,788,640	**13,577,279**	13,853,453	**2758**	52298	5782
0.2	30,521,170	**61,042,339**	-	**12196**	TO	27085
0.1	175,901,422	**351,802,843**	-	**78599**	TO	TO
0.18	6,581,293	**13,162,585**	-	**16886**	TO	OOM
0.15	10,320,509	**20,641,017**	-	**26497**	TO	OOM
0.1	24,476,915	**48,953,829**	-	**66785**	TO	OOM
0.1	22,598,666	**45,197,331**	48,301,365	10489	22502	**3198**
0.08	44,286,784	**88,573,567**	93,459,361	20900	43804	**4981**
0.06	92,532,208	**185,064,415**	-	45554	TO	**12909**
0.2	534,121	**1,068,241**	1,140,525	181	330	**31**
0.1	3,149,673	**6,299,345**	6,508,211	1071	1839	**195**
0.05	7,762,648	**15,525,295**	15,847,857	2619	4427	**576**
0.3	4,369,825	**8,739,649**	8,925,567	**1731**	32289	4400
0.2	17,997,787	**35,995,573**	-	**7237**	TO	19426
0.01	138,795	**277,589**	329,347	119	5771	**25**
0.008	174,672	**349,343**	409,301	156	7317	**29**
0.005	232,315	**464,629**	542,607	186	9424	**34**
0.8	46755	93509	-	964	TO	655
0.7	337740	675479	-	**7252**	TO	10660
0.1	1,580	**3,159**	60,871	8	2968	2
0.05	30,473	**60,945**	2,541,737	170	52622	**38**
0.02	565,846	**1,131,691**	-	1699	TO	**167**
0.005	1,074	**2,147**	-	**669**	TO	OOM
0.0025	7,698	**15,395**	-	**1610**	TO	OOM
0.08	138	**275**	-	**13**	TO	OOM
0.05	317	**633**	-	**85**	TO	OOM
0.01	65,237	**130,473**	-	**7094**	TO	OOM
0.001	3,982	**7963**	141,207	187	60778	**134**
0.0005	294,585	**589169**	-	3685	TO	**617**
0.004	832	**1663**	-	**306**	OOM	OOM
0.002	2,692	**5383**	-	**3626**	OOM	OOM

(b) AC_3

Dataset $\lvert\mathcal{I}\rvert \times \lvert\mathcal{T}\rvert$	θ	#Motifs	#Nœud		Temps (s)		
		(1)(2)(3)	(1)	(2)	(1)	(2)	(3)
CHESS	0.3	6,199,288	**12,398,575**	12,674,595	**1416**	46366	5815
75 × 3,196	0.2	27,341,083	**54,682,165**	-	**6216**	TO	25545
	0.1	150,302,539	**300,605,077**	-	**37159**	TO	TO
CONNECT	0.18	36,897,79	**7,379,557**	-	**8026**	TO	OOM
129 × 67,557	0.15	5,741,671	**1,1483,341**	-	**13258**	TO	OOM
	0.1	13,360,851	**267,21,701**	-	**31827**	TO	OOM
HEART-CLEVELAND	0.1	13,652,085	**27,304,169**	30,360,067	3377	9717	**1264**
95 × 296	0.08	25,776,190	**51,552,379**	56,330,623	6171	18830	**3055**
	0.06	51,168,896	**102,337,791**	109,707,247	12590	38049	**6692**
HEPATITIS	0.2	429,368	**858,735**	932,627	70	166	**25**
68 × 137	0.1	2,263,487	**4,526,973**	4,739,553	391	887	**143**
	0.05	5,031,535	**10,063,069**	10,378,429	874	1979	**377**
KR-VS-KP	0.3	4,345,059	**8,,690,117**	8,875,613	**925**	30780	5232
73 × 3,196	0.2	17,881,775	**35,763,549**	-	**3813**	TO	20553
MUSHROOM	0.01	45,766	**91,531**	141,203	28	2407	**18**
112 × 8,124	0.008	55,721	**111,441**	168,807	37	2795	**20**
	0.005	71,996	**143,991**	218,075	47	3525	**25**
PUMSB	0.8	36,450	**72,899**	-	**357**	TO	601
2,113 × 49,046	0.7	254,892	**509,783**	-	**2657**	TO	10862
SPLICE1	0.1	1,580	**3,159**	60,871	5	2986	2
287 × 3,190	0.05	30,473	**60,945**	2,541,737	146	53964	**41**
	0.02	56,5792	**1,131,583**	-	1286	TO	**202**
T10I4D100K	0.005	1074	**2,147**	-	**669**	TO	OOM
870 × 100,000	0.0025	7697	**153,93**	-	**1503**	TO	OOM
T40I10D100K	0.08	138	**275**	-	**11**	TO	OOM
942 × 100,000	0.05	317	**633**	-	**81**	TO	OOM
	0.01	65,237	**130,473**	-	**7431**	TO	OOM
BMS1	0.001	3,979	**7,957**	141,201	152	54370	**134**
497 × 59,602	0.0005	192,125	**384,249**	-	2164	TO	**578**
RETAIL	0.004	832	**1,663**	-	**196**	OOM	OOM
16470 × 88,162	0.002	2,692	**5,383**	-	**3852**	OOM	OOM

(c) AC_2

TAB. 3 – Analyse comparative pour l'extraction de motifs clos. " – " : résultats non disponibles. TO : Time Out; OOM : Out Of Memory. (1) : ADEQUATE-CI (2) : CP+CLOSED (3) : MICMAC. $AC_1 = \{min(P.val), sup(P) \geq \theta\}$, $AC_2 = \{max(P.val), sup(P) \geq \theta\}$, $AC_3 = \{min(P.val), max(P.val), sup(P) \geq \theta\}$

15 TO et 2 OOM. Comparé à MICMAC, ADEQUATE-CI a le meilleur temps d'éxecution sur 17 instances, avec un facteur d'accélération compris entre 3 et 5. Les seules exceptions sont HEART-CLEVELAND, HEPATITIS et MUSHROOM, où MICMAC est plus efficace. Par ailleurs, ADEQUATE-CI domine très largement CP+CLOSED. En effet, le nombre élevé de transactions

et d'items augmente sensiblement le temps de propagation pour le modèle réifié. La nature level-wise de MICMAC explique en partie les *Out Of Memory*, à cause du grand nombre de candidats qui doit être stocké pendant le processus de fouille. Ces résultats démontrent l'intérêt de notre approche pour la fouille de motifs clos. Nous rappelons que notre approche est générique et plus flexible : l'utilisateur peut facilement ajouter de nouvelles contraintes sans avoir à modifier le système sous-jacent.

Dataset $\lvert\mathcal{I}\rvert \times \lvert\mathcal{T}\rvert$	M	|Sky(M)|	# Nœud		Temps (s)		
		(1)(2)(3)	(1)	(2)	(1)	(2)	(3)
CHESS 75 × 3,196	ag	18	428,795	428,831	**142**	966	OOM
	fa	13	4,085	4,189	**4**	41	OOM
	fg	19	343,221	338,921	**78**	687	OOM
	fag	160	379,675	378,391	**146**	957	OOM
	agmM	577	1,582,829	1,541,931	**3559**	6183	OOM
	agnM	674	1,671,847	1,630,627	**3838**	6488	OOM
	agnm	1,459	1,700,829	1,648,195	**5778**	8630	OOM
	fagM	293	576,125	572,445	**241**	1152	OOM
	fagn	846	1,079,609	1,044,509	**944**	2721	OOM
	famM	245	24,213	24,815	**19**	155	OOM
	fanM	281	27,773	28,717	**24**	151	OOM
	fanm	219	12,169	12,389	**13**	97	OOM
	fgmM	249	1,406,647	1,346,019	**377**	2277	OOM
	fgnM	356	1,483,179	1,418,595	**434**	2312	OOM
	fgnm	446	1,560,195	1493,333	**577**	2516	OOM
	fagmM	2,464	1,606,675	1,564,011	**4862**	7595	OOM
	fagnM	3,014	1,689,931	1,645,885	**5287**	8297	OOM
	fagnm	3,630	1,715,569	1,663,603	**7286**	10150	OOM
	fagnMm	7,845	3,591,471	3,551,559	**59720**	66388	OOM
CONNECT 129 × 67,557	ag	45	4,775,201	-	**25641**	TO	OOM
	fa	17	3,715	3,745	**47**	1722	OOM
	fg	26	3,623,055	-	**20483**	TO	OOM
	fag	359	4,186,703	-	**25458**	TO	OOM
	agmM	903	6,618,189	-	**69963**	TO	OOM
	agnM	857	6,674,167	-	**68990**	TO	OOM
	agnm	-	-	-	TO	TO	OOM
	fagM	941	6,234,675	-	**43943**	TO	OOM
	fagn	-	-	-	TO	TO	OOM
	famM	142	6,403	6,519	**68**	2186	OOM
	fanM	155	6,565	6,681	**61**	2240	OOM
	fanm	367	25,161	25,407	**205**	4162	OOM
	fgmM	291	5,245,879	-	**37120**	TO	OOM
	fgnM	436	5,319,999	-	**36955**	TO	OOM
	fgnm	905	10,168,103	-	**83788**	TO	OOM
	fagmM	-	-	-	TO	TO	OOM
	fagnM	-	-	-	TO	TO	OOM
	fagnm	-	-	-	TO	TO	OOM
	fagnMm	-	-	-	TO	TO	OOM
HEART-CLEVELAND 95 × 296	ag	28	1,187,475	1,094,719	**326**	463	OOM
	fa	14	82,407	83,597	**14**	**14**	OOM
	fg	33	1,108,677	957,337	**237**	374	OOM
	fag	115	1,179,427	1,086,389	**468**	632	OOM
	agmM	479	1,476,827	1,530,369	**7327**	8576	OOM
	agnM	421	1,487,729	1,508,703	**7610**	9204	OOM
	agnm	2,893	1,505,917	1,688,261	**43747**	48893	OOM
	fagM	181	1,246,275	1,148,699	**1008**	1229	OOM
	fagn	752	149,0497	1,509,219	**9254**	10331	OOM
	famM	78	105,135	106,613	**39**	45	OOM
	fanM	130	142,153	143,821	**56**	61	OOM
	fanm	233	210,245	212,289	**130**	141	OOM
	fgmM	136	1,249,387	1,100,997	**844**	985	OOM
	fgnM	145	1,243,635	1,081,463	**976**	1116	OOM
	fgnm	417	867,935	835,749	**1468**	1687	OOM
	fagmM	830	1,502,943	1,559,133	**10033**	11464	OOM
	fagnM	795	1,512,021	1,536,235	**10401**	11872	OOM
	fagnm	4,204	1,708,165	1,966,619	**64454**	70359	OOM
	fagnMm	-	-	-	TO	TO	OOM

TAB. 4 – Analyse comparative pour l'extraction de skypatterns sur les datasets CHESS, CONNECT et HEART-CLEVELAND. " – " : resultats non disponibles (TO or OOM). TO : Time Out; OOM : Out Of Memory. (1) : CLOSEDSKY (2) : CP+SKY (3) : AETHERIS.

Dataset $\lvert\mathcal{I}\rvert \times \lvert\mathcal{T}\rvert$	M	|Sky(M)|	# Nœud		Temps (s)		
		(1)(2)(3)	(1)	(2)	(1)	(2)	(3)
HEPATITIS 68 × 137	ag	22	98,879	94,071	**20**	23	432
	fa	17	31,175	31,599	**4**	3	429
	fg	33	78,027	66,875	**14**	16	435
	fag	89	95,375	91,073	**30**	35	439
	agmM	280	125,091	133,999	**230**	264	814
	agnM	277	124,399	133,971	**203**	233	860
	agnm	1,070	176,333	193,791	**1123**	1184	1351
	fagM	200	105,053	103,539	**59**	67	460
	fagn	424	134,453	147,083	**291**	345	844
	famM	73	39,253	39,865	**11**	**11**	804
	fanM	78	40,687	41,339	**10**	**11**	861
	fanm	163	58,433	59,313	**26**	27	1342
	fgmM	127	82,947	76,411	**44**	51	807
	fgnM	130	80,619	76,343	**45**	51	855
	fgnm	300	89,991	89,295	**110**	122	1340
	fagmM	454	135,717	147,497	**375**	418	817
	fagnM	467	136,069	149,035	**341**	393	875
	fagnm	1,366	187,435	207,941	1564	1772	**1362**
	fagnMm	4,242	458,733	573,029	17951	18922	2092
KR-VS-KP 73 × 3,196	ag	18	464,439	495,993	**267**	810	OOM
	fa	13	3,095	3,189	**3**	32	OOM
	fg	42	30,3767	321,847	**104**	413	OOM
	fag	145	419,133	448,447	**460**	1079	OOM
	agmM	426	550,395	569,155	**1752**	2681	OOM
	agnM	491	611,563	631,283	**2756**	3849	OOM
	agnm	1,885	910,823	954,803	**10301**	12115	OOM
	fagM	435	513,537	547,759	**1477**	2438	OOM
	fagn	1,423	586,571	617,695	**3086**	4376	OOM
	famM	125	7,187	7,271	**7**	59	OOM
	fanM	179	7,795	7,889	**7**	61	OOM
	fanm	221	7,957	8,049	**10**	67	OOM
	fgmM	241	352,221	357,985	**230**	692	OOM
	fgnM	429	396,307	402,753	**384**	934	OOM
	fgnm	589	627,629	648,899	**918**	1709	OOM
	fagmM	1,639	602,813	633,661	**3368**	4575	OOM
	fagnM	2,285	674,531	706,523	**5576**	7171	OOM
	fagnm	5,854	957,721	1,007,825	**16484**	19022	OOM
	fagnMm	9,683	1,115,879	1,165,039	**68169**	75077	OOM
MUSHROOM 112 × 8,124	ag	15	15,881	30,299	**10**	355	15
	fa	5	711	1,169	**3**	97	15
	fg	17	15,405	29,585	**9**	328	15
	fag	59	15,761	30,175	**10**	356	15
	agmM	245	19,363	35,711	**29**	508	**28**
	agnM	240	19,339	35,803	**30**	513	**30**
	agnm	513	23,131	40,195	**72**	663	109
	fagM	110	16,031	30,865	**10**	393	16
	fagn	268	18,049	33,565	**20**	444	40
	famM	97	1,653	2,761	**5**	152	28
	fanM	137	1,781	3,141	**4**	151	40
	fanm	92	2,157	3,829	**6**	191	110
	fgmM	143	18,385	34,163	**19**	452	28
	fgnM	182	18,387	34,223	**20**	447	**30**
	fgnm	234	19,781	35,815	**30**	445	108
	fagmM	506	19,471	35,851	**34**	534	**29**
	fagnM	551	19,503	36,001	**33**	548	**30**
	fagnm	974	23,365	40,433	**88**	682	109
	fagnMm	1,021	25,501	44,911	107	780	**59**

TAB. 5 – Analyse comparative pour l'extraction de skypatterns sur les datasets HEPATITIS, KR-VS-KP et MUSHROOM. " − " : resultats non disponibles (TO or OOM). TO : Time Out ; OOM : Out Of Memory. (1) : CLOSEDSKY (2) : CP+SKY (3) : AETHERIS.

7.3 Comparaison entre CLOSEDSKY, CP+SKY et AETHERIS

Nous avons aussi comparé CLOSEDSKY, CP+SKY et AETHERIS pour la fouille de skypaterns avec différentes combinaisons de mesures parmi l'ensemble $\{sup : f, \max : M, \min : m, area : a, mean : n, growth\text{-}rate : g\}$. Pour chaque méthode et chaque combi-

naison sélectionnée, nous reportons le temps CPU, le nombre de skypatterns et le nombre de noeuds explorés par les deux méthodes PPC. Rappelons que AETHERIS et CP+SKY calculent en premier un ensemble représentatif de motifs par rapport à M' et appliquent ensuite l'opérateur Sky sur l'ensemble des motifs extraits, alors que notre méthode basée sur ADEQUATE-CI ne nécessite qu'une seule étape. Nous avons utilisé un *seuil de fréquence* de 1. Les tables 4 et 5 montrent les résultats obtenus.

Premièrement, les résultats montrent qu'il y a une grande différence entre le nombre de motifs clos (en millions, voir table 3) en comparaison avec le nombre de motifs Sky (en milliers). Cela démontre l'intérêt de la *dominance Pareto* pour réduire le nombre de motifs. Deuxièmement, en observant le temps d'exécution, on constate que CLOSEDSKY surpasse CP+SKY et AETHERIS sur toutes les instances considérées. CLOSEDSKY permet d'extraire les skypatterns là où les deux autres approches échouent. En effet, AETHERIS obtient des *Out Of Memory* sur 76 instances (sur un total de 114), alors que CP+SKY ne parvient pas à finir l'extraction sur 17 instances. Par comparaison, CLOSEDSKY échoue sur 6 instances (5 instances pour CONNECT et 1 instance pour HEART-CLEVLAND). Pour les jeux de données où CP+SKY et AETHERIS arrivent à terminer l'extraction, CLOSEDSKY obtient le meilleur temps d'exécution, sauf pour 2 instances où AETHERIS est plus efficace. Pour CHESS, CLOSEDSKY est 8 fois plus rapide que CP+SKY ; pour MUSHROOM, le facteur d'accélération est en moyenne de 23.86. Pour HEPATITIS, CLOSEDSKY est plus rapide que AETHERIS (en moyenne 25.56 fois plus rapide).

7.4 Impact de la règle (3) sur les performances de ADEQUATE-CI

ADEQUATE-CI assure la cohérence de domaine avec une complexité cubique mais avec un temps d'exécution plus long. Nous avons implémenté une nouvelle version qui ne prend en compte que les règles (1) et (2) (complexité quadratique). Nous avons testé cette nouvelle version (notée CLOSEDSKY-WC) sur CONNECT et SPLICE1. Les résultats sont disponibles dans (Vernerey et al., 2021). `WC` domine clairement `DC` en temps de calcul. Pour CONNECT, `WC` arrivent à trouver les motifs Sky pour 3 instances où `DC` n'arrivent pas à terminer l'extraction, `WC` étant en moyenne 9.5 fois plus rapide que `DC`. Pour SPLICE1, `WC` réussit à terminer l'extraction sur 7 instances (sur un total de 19). Comme seconde observation, le nombre de noeuds exploré par `DC` est à chaque fois plus petit que celui de `WC` mais la différence n'est pas significative contrairement au gain en temps d'exécution que procure `WC`. Par conséquent, un filtrage plus faible constitue un bon compromis pour les instances qui sont très difficiles à résoudre.

7.5 Analyse comparative des clusterings conceptuels Pareto optimaux

Nous avons utilisé les mêmes jeux de données de l'UCI testés dans Ouali et al. (2016). Nous considérons deux ensembles de mesures pour l'extraction de motifs clos $M_1 = \{sup(P) \geq \theta\}$ et $M_2 = \{sup(P) \geq \theta, mean(P.val)\}$, avec $\theta = 1$ et deux critères d'optimisation : maximisation de $minMean$ et $minFreq$. Nous évaluons l'intérêt de calculer le front de Pareto des clusterings non dominées par rapport aux deux critères $minMean$+$minFreq$. La valeur de k_{max}, le nombre de cluster maximal, a été fixée à $(m-1)$, m étant le nombre de transactions de l'instance. Nous reportons les résultats pour les jeux de données pour lesquels au moins un clustering optimal a été trouvé dans une limite de temps de 24 heures.

La figure 1a présente, pour chaque ensemble de mesures, le nombre de motifs clos générés à l'étape 1 et le temps de calcul des deux étapes nécessaire pour trouver toutes les solutions optimales. L'extraction de solutions optimales selon M_2 nécessite plus temps comparé à M_1. Cela provient probablement du grand nombre de motifs considérés (il y a en moyenne 2.9 fois plus de motifs selon M_2).

La figure 1b compare les fronts de Pareto des clusterings trouvés sur les datasets de la figure 1a. Pour chaque dataset, nous donnons les clusterings Pareto obtenus à partir des motifs clos selon M_1 et M_2. Les solutions communes sont indiquées par des points ronds. Nous pouvons remarquer que, pour les deux datasets LYMPH et SOYBEAN, les clusterings trouvés avec M_2 (points en triangle) dominent largement ceux extraits avec M_1 (points en carré). Pour PRIMARY-TUMOR, nous obtenons un nouveau clustering Pareto avec M_2 que nous ne trouvons pas avec M_1. Enfin, pour les deux datasets ZOO et CLEVE, les deux approches trouvent le même clustering. Ces résultats montrent clairement l'intérêt d'exploiter des motifs fermés par rapport à un ensemble de mesures pour extraire des clusterings conceptuels plus pertinents. Notons que toutes les solutions optimales de la figure 1b ont 2 clusters. Cela est notamment dû au fait que le support est une mesure anti-monotone et donc plus le nombre de clusters augmente et plus $minFreq$ diminue. Ainsi, les solutions Pareto par rapport à $minFreq$+$minMean$ ont tendance à favoriser des clusterings avec un nombre de clusters plus petit.

8 Conclusions

Nous avons proposé une nouvelle contrainte globale pour la fouille de motifs clos par rapport à un ensemble de mesures. Nous avons montré l'utilisation de notre contrainte pour l'extraction de skypatterns et l'extraction de clusterings conceptuels non-dominés. Nous avons mené des expérimentations sur plusieurs jeux de données de l'UCI qui ont démontré l'efficacité et le passage à l'échelle de notre approche comparé au modèle réifié CP+SKY et aux méthodes spécialisées MICMAC et AETHERIS. L'analyse qualitative des clusterings obtenus montre l'intérêt d'exploiter des motifs fermés selon plusieurs mesures.

Références

Bastide, Y., N. Pasquier, R. Taouil, G. Stumme, et L. Lakhal (2000). Mining minimal non-redundant association rules using frequent closed itemsets. In J. W. Lloyd, V. Dahl, U. Furbach, M. Kerber, K. Lau, C. Palamidessi, L. M. Pereira, Y. Sagiv, et P. J. Stuckey (Eds.), *Computational Logic - CL 2000, First International Conference, London, UK, 24-28 July, 2000, Proceedings*, Volume 1861 of *Lecture Notes in Computer Science*, pp. 972–986. Springer.

Börzsönyi, S., D. Kossmann, et K. Stocker (2001). The skyline operator. In *ICDE*, pp. 421–430.

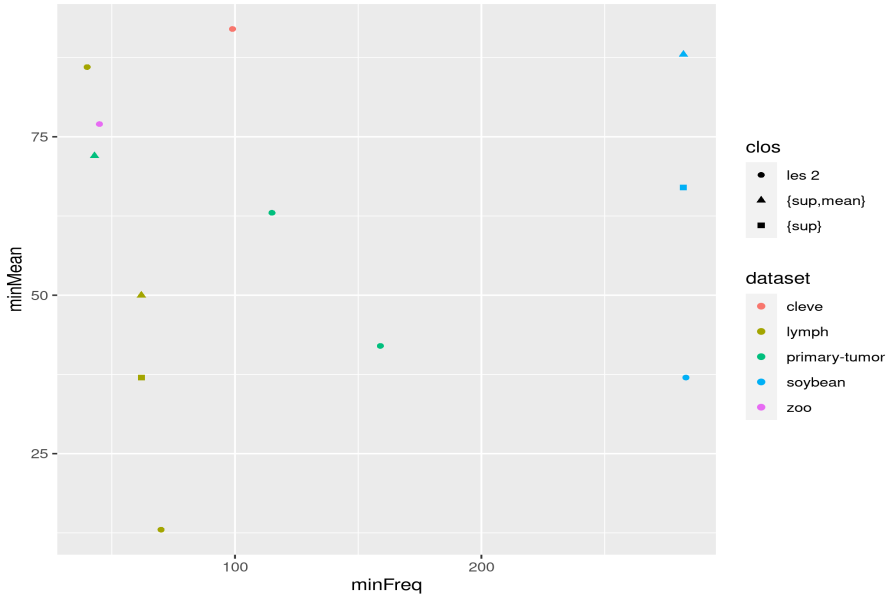
Calders, T., C. Rigotti, et J. Boulicaut (2004). A survey on condensed representations for frequent sets. In *Constraint-Based Mining and Inductive Databases*, pp. 64–80. Springer.

Chabert, M. et C. Solnon (2017). Constraint programming for multi-criteria conceptual clustering. In *CP 2017*, Volume 10416 of *LNCS*, pp. 460–476. Springer.

Giacometti, A., D. Laurent, et C. T. Diop (2002). Condensed representations for sets of mining queries. In *Proceedings of the 1st Int. Workshop on Inductive Databases*, pp. 5–19.

Dataset $\lvert\mathcal{I}\rvert \times \lvert\mathcal{T}\rvert$	M	nb clos	Temps (s.)
CLEVE	M_1	85035	114
43×303	M_2	156113	375
LYMPH	M_1	49655	18
59×142	M_2	174697	135
PRIMARY-TUMOR	M_1	87230	4506
31×336	M_2	143238	6027
SOYBEAN	M_1	31759	202
50×630	M_2	133344	3042
TIC-TAC-TOE	M_1	42711	TO
27×958	M_2	51970	122
WINE	M_1	26785	4900
45×178	M_2	66620	TO
ZOO	M_1	4567	3
36×101	M_2	18685	11

(a) Temps de calcul pour trouver les solutions non-dominées par rapport aux deux critères $minMean$+$minFreq$.



(b) Front de Pareto des clusterings conceptuels non-dominés par rapport aux deux critères objectifs $minMean$+$minFreq$.

FIG. 1 – Analyse comparative des clusterings conceptuels par rapport aux deux ensembles de mesures M_1 et M_2.

Guns, T., S. Nijssen, et L. De Raedt (2011). Itemset mining : A constraint programming perspective. *Artificial Intelligence 175*(12), 1951–1983.

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

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

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.

Mannila, H. et H. Toivonen (1997). Levelwise search and borders of theories in knowledge discovery. *Data Min. Knowl. Discov. 1*(3), 241–258.

Novak, P. K., N. Lavrac, et G. I. Webb (2009). Supervised descriptive rule discovery : A unifying survey of contrast set, emerging pattern and subgroup mining. *Journal of Machine Learning Research 10*, 377–403.

Ouali, A., S. Loudni, Y. Lebbah, P. Boizumault, A. Zimmermann, et L. Loukil (2016). Efficiently finding conceptual clustering models with integer linear programming. In *IJCAI 2016*, pp. 647–654.

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.

Prud'homme, C., J.-G. Fages, et X. Lorca (2016). Choco Solver Documentation.

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

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.

Schaus, P. et R. Hartert (2013). Multi-Objective Large Neighborhood Search. In *Proceedings of CP 2013*.

Soulet, A. et B. Crémilleux (2008). Adequate condensed representations of patterns. *Data Min. Knowl. Discov. 17*(1), 94–110.

Soulet, A., B. Crémilleux, et F. Rioult (2004). Condensed representation of emerging patterns. In *Proceedings of the 8th PAKDD*, pp. 127–132. Springer.

Soulet, A., C. Raïssi, M. Plantevit, et B. Crémilleux (2011). Mining dominant patterns in the sky. In *Proceedings of the ICDM 2011*, pp. 655–664. IEEE Computer Society.

Ugarte, W., P. Boizumault, B. Crémilleux, A. Lepailleur, S. Loudni, M. Plantevit, C. Raïssi, et A. Soulet (2017). Skypattern mining : From pattern condensed representations to dynamic constraint satisfaction problems. *Artif. Intell. 244*, 48–69.

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.

Vernerey, C., S. Loudni, N. Aribi, et Y. Lebbah (2023). Fouille de motifs sans seuil par optimisation multi-objectifs : Application aux règles d'association. In C. Faron et S. Loudcher (Eds.), *Extraction et Gestion des Connaissances, EGC 2023, Lyon, France, 16 - 20 janvier 2023*, Volume E-39 of *RNTI*, pp. 483–490. Editions RNTI.

Vernerey, C., S. Loudni, N. Aribi, et Y. Lebbah (April 2021). Supplementary Material : https://drive.google.com/file/d/1LwzEojaTCzMuVs4HFwGn7-JPlHjCWvmC/view?usp=sharing.

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.

Wrobel, S. (1997). An algorithm for multi-relational discovery of subgroups. In *PKDD*, Volume 1263 of *LNCS*, pp. 78–87. Springer.

Yang, G. (2004). The complexity of mining maximal frequent itemsets and maximal frequent patterns. In *In KDD '04 :*, pp. 344–353. ACM Press.

Summary

Condensed representations of patterns offer an elegant way to represent solution sets compactly, while minimizing the redundancy and the number of patterns. This approach has been mainly developed in the context of the frequency measure and there are very few works addressing other measures. We propose a generic framework based on constraint programming to efficiently mine adequate condensed representations of patterns w.r.t. a set of measures. For this, we introduce a new global constraint with a complete polynomial filtering. We show how this constraint can be exploited in association with Pareto dominance constraints to mine skypatterns and conceptual clustering. Experiments performed on standard datasets show the efficiency of our approach and its significant advantages over existing approaches.