



HAL
open science

Utilisation de l'opérateur OWA pour le clustering conceptuel équitaine

Nouredine Aribi, Abdelkader Ouali, Yahia Lebbah, Samir Loudni

► To cite this version:

Nouredine Aribi, Abdelkader Ouali, Yahia Lebbah, Samir Loudni. Utilisation de l'opérateur OWA pour le clustering conceptuel équitaine. 14èmes Journées Francophones de Programmation Par Contraintes, Jun 2018, Amiens, France. hal-01818827

HAL Id: hal-01818827

<https://hal.science/hal-01818827v1>

Submitted on 19 Jun 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Utilisation de l'opérateur OWA pour le clustering conceptuel équitable

Noureddine Aribi¹ Abdelkader Ouali² Yahia Lebbah¹ Samir Loudni²

¹ Lab. LITIO, University of Oran 1, 31000 Oran, Algeria.

² Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France.

{aribi.noureddine,lebbah.yahia}@univ-oran1.dz {abdelkader.ouali,samir.loudni}@unicaen.fr

Résumé

Nous proposons une approche de clustering conceptuel qui exploite la notion d'équité en optimisation multi-agents. L'idée consiste à s'assurer que les clusters obtenus soient équilibrés, i.e. approximativement de même taille. Dans notre approche, chaque agent représente un concept et l'utilité d'un agent correspondant à une mesure spécifique (par exemple la fréquence du concept). Le problème consiste alors à trouver la meilleure satisfaction cumulative tout en mettant l'accent sur un compromis équitable entre tous les agents individuels. Pour déterminer la solution optimale de meilleur compromis, nous avons exploité une formulation équitable de l'opérateur des moyennes pondérées ordonnées (en anglais, Ordered Weighted Averages - OWA), avec un jeu de poids particulièrement adapté à la recherche de solutions équitables. Les expériences effectuées sur les jeux de données de l'UCI et sur des instances issues d'une application réelle (ERP) montrent que notre approche trouve efficacement des clusterings de bonne qualité.

Abstract

We propose an equitable conceptual clustering approach based on multi-agent optimization. In the context of conceptual clustering, each cluster is represented by an agent having its own satisfaction and the problem consists in finding the best cumulative satisfaction while emphasizing a fair compromise between all individual agents. The fairness goal is achieved using an equitable formulation of the Ordered Weighted Averages (OWA) operator. Experiments performed on UCI datasets and on instances coming from real application ERP show that our approach efficiently finds clusterings of consistently high quality.

1 Introduction

Structurer les données dans le processus de découverte des connaissances est une tâche fondamentale qui

permet de mieux comprendre les données et d'identifier des groupements d'objets (appelés clusters) en fonction d'une mesure de similarité prédéfinie. En pratique, les utilisateurs aimeraient souvent effectuer d'autres actions, telle que l'interprétation sémantique de chaque cluster. Les méthodes telles que le clustering conceptuel répondent à cette question en tentant de trouver des descriptions des clusters au moyen de concepts formels.

De nombreuses approches ont été proposées pour le clustering conceptuel. Les approches traditionnelles [12, 8] combinent la formation des clusters et des descriptions. D'autres techniques [19, 18] ont plutôt choisi de découpler la recherche des descriptions - avant ou après l'étape du clustering. Plus récemment, des approches de programmation par contraintes (PPC) [4] et de programmation linéaire en nombres entiers (PLNE) [16] ont été proposées pour résoudre le problème du clustering conceptuel optimal dans un cadre déclaratif. Elles combinent deux techniques exactes : dans un premier temps, un outil de fouille de données dédié (i.e. LCM [21]) est utilisé pour calculer l'ensemble de tous les concepts formels et, dans un deuxième temps, la PLNE ou la PPC est utilisée pour sélectionner les meilleurs k clusters (i.e. les concepts) qui optimisent un critère donné. La plupart des mesures d'optimisation utilisées dans ces approches conduisent à des clusters déséquilibrés. S'assurer que les clusters obtenus soient (approximativement) équilibrés, permet de rendre les résultats du clustering plus utiles et exploitables [2, 23].

Cet article introduit le concept d'équité et de solutions équitables efficaces pour le problème du clustering conceptuel dans un contexte multi-agents, où chaque agent représente un concept et possède sa propre utilité liée à une mesure spécifique (e.g. la fré-

quence). Ici, l'équité correspond à l'idée de favoriser des solutions qui partagent équitablement la satisfaction entre les agents [9]. L'équité a été entièrement étudiée par la communauté d'optimisation multicritères [10], et formalisée à travers les trois propriétés : **(i) La symétrie** qui signifie que tous les agents ont la même importance. Par exemple, les deux vecteurs d'utilités $(5, 3, 0)$ et $(0, 3, 5)$ sont équivalents. **(ii) La Pareto-monotonie** exprime qu'une solution (x_1, x_2, \dots, x_n) est meilleure que la solution (y_1, y_2, \dots, y_n) , si et seulement si, $x_i \geq y_i$ pour tout i , avec au moins une inégalité stricte. **(iii) Le Principe de transfert** formalise une distribution équitable des utilités [20]. L'intuition est que tout transfert entre deux utilités inéquitables x_i et x_j , qui préserve la moyenne des utilités, améliorerait l'utilité globale.

Une manière courante de traiter le concept de solutions équitablement efficaces consiste à définir des fonctions d'agrégation qui remplissent les propriétés ci-dessus. Ceci définit une famille d'agrégations équitables qui sont *Schur-convexe* [11]. Dans la littérature, il existe plusieurs fonctions pour agréger les utilités des agents au moyen de *fonction d'utilité collective* (CUF). Les agrégations les plus utilisées sont **maxMin**, **maxSum** et **minDev**. La fonction **maxSum** combine linéairement des utilités, tandis que la fonction **minDev** minimise l'écart entre le meilleur et le pire des utilités. Le principe de transfert n'est pas assuré par **maxMin** et **minDev**, sur toutes les utilités, conduisant ainsi à *l'effet de noyade* [7]. La fonction **maxSum** est entièrement compensatoire et ne capture donc pas l'idée d'équité.

Dans ce papier, nous proposons une approche efficace pour le clustering conceptuel équitable, qui fait appel à deux techniques exactes : (1) extraction des motifs fermés (en utilisant l'algorithme LCM [21]); (2) sélection des meilleurs clusters à l'aide de la Programmation Linéaire en Nombres Entiers (PLNE) qui exploite une fonction d'agrégation équitable basée sur **OWA**, remplissant les trois propriétés d'équité cités ci-dessus. Notre opérateur **OWA** utilise des poids spécifiques proposés par Golden et Perny [9] qui assurent la propriété de Schur-convexité. Les expériences réalisées sur des ensembles de données UCI et sur un ensemble d'instances provenant d'une application réelle ERP montrent que notre approche basée sur **OWA** trouve efficacement des clusterings de hautes qualités, par rapport aux approches concurrentes.

La section 2 introduit les concepts utilisés dans ce document. La section 3 décrit nos modèles PLNE pour le clustering conceptuel équitable. Nous discutons les travaux connexes dans la section 4 avant de montrer les performances de notre approche dans la section 5. La section 6 conclut et esquisse les orientations de nos recherches futures.

2 Préliminaires

2.1 Concepts formels et clustering conceptuel

Concepts formels. Soit \mathcal{D} un ensemble de m transactions (numérotées de 1 à m), \mathcal{I} un ensemble de n items (numérotés de 1 à n), et $R \subseteq T \times \mathcal{I}$ une relation binaire qui lie les transactions aux items : $(t, i) \in R$ si la transaction t contient l'item i : $i \in t$. Un itemset (ou *motif*) est un sous-ensemble non nul de \mathcal{I} . Par exemple, la table 1a contient une base transactionnelle \mathcal{D} avec $m = 11$ transactions t_1, \dots, t_{11} décrit par $n = 8$ items. L'*extent* d'un ensemble $I \subseteq \mathcal{I}$ d'items est l'ensemble des transactions contenant tous les items de I , i.e. $ext(I) = \{t \in \mathcal{D} \mid \forall i \in I, (t, i) \in R\}$. L'*intent* d'un sous-ensemble $T \subseteq \mathcal{D}$ est l'ensemble des items contenus dans toutes les transactions dans T , c'est-à-dire, $int(T) = \{i \in \mathcal{I} \mid \forall t \in T, (t, i) \in R\}$. Ces deux opérateurs induisent une connexion de Galois entre $2^{\mathcal{D}}$ et $2^{\mathcal{I}}$, i.e. $T \subseteq ext(I) \Leftrightarrow I \subseteq int(T)$. Une paire telle que $(I = int(T), T = ext(I))$ s'appelle **concept formel**. Cette définition définit une propriété de **fermeture** sur l'ensemble de données \mathcal{D} , $closed(I) \Leftrightarrow I = int(ext(I))$. Un ensemble d'items I pour lequel $closed(I) = vrai$ s'appelle *motif fermé*. En utilisant $ext(I)$, on peut définir la *fréquence* d'un concept : $freq(I) = |ext(I)|$, sa *diversité* : $divers(I) = \sum_{t \in ext(I)} |\{i \in \mathcal{I} \mid (i \notin I) \wedge (i \in t)\}|$, et sa *taille* : $size(I) = |\{i \mid i \in I\}|$. On note \mathcal{C} l'ensemble de tous les concepts formels.

Clustering Conceptuel. Le clustering consiste à partitionner un ensemble de transactions en groupes relativement homogènes. Le clustering conceptuel vise à fournir une description distincte de chaque groupe - le concept caractérisant les transactions qu'il contient. Ce problème peut être formulé comme suit : "trouver un ensemble de k clusters, chacun décrit par un motif fermé P_1, P_2, \dots, P_k , couvrant toutes les transactions sans aucun chevauchement entre les clusters". Par exemple, la table 1c illustre trois classifications possibles pour $k = 3$. Une fonction d'évaluation f qui optimise un critère donné peut être utilisée pour exprimer la qualité du clustering. Différents critères d'optimisation peuvent être considérés : maximiser la *somme des fréquences* des concepts sélectionnés ; minimiser la *somme des diversités* des concepts sélectionnés. Par exemple, pour la base transactionnelle \mathcal{D} et $k = 3$, minimiser $f(P_1, \dots, P_k) = \sum_{1 \leq i \leq k} divers(P_i)$ fournit un clustering s_1 , avec une valeur optimale égale à 18 (voir la table 1c). La solution $s_1 = (1, 1, 9)$ engendre un gros cluster (de taille 9) couvrant la plupart des transactions, et deux clusters qui couvrent une seule transaction. Toutefois, cette solution est moins intéressante que celle où tous les clusters sont de taille comparable.

Trans.	Items							
t_1	A	B		D				
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) Données transactionnelles \mathcal{T} .

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
t_1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
t_2	1	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0
t_3	1	0	1	0	1	0	0	0	0	0	0	0	0	0	1	1	0	1
t_4	1	0	1	0	1	0	0	0	0	0	0	0	0	0	1	1	0	1
t_5	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1	0	1
t_6	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1	0	1
t_7	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	1	0	1
t_8	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	1	0	1
t_9	0	0	0	0	0	0	0	1	1	0	1	0	0	1	1	0	0	0
t_{10}	0	0	0	0	0	0	0	1	1	0	1	0	0	1	1	0	0	0
t_{11}	0	0	0	0	0	0	0	1	0	0	0	1	1	1	0	0	1	1

(b) $(a_{t,c})$ matrice associée à l'ensemble de données \mathcal{D} .

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

(c) 3 clustering conceptuels pour $k = 3$.

TABLE 1 – Exemple de motivation.

La façon la plus courante pour obtenir des clusters plus équilibrés est de considérer des fonctions d'optimisation dédiées :

- *maximiser la fréquence minimale* (**maxMin**). On cherche des solutions dans lesquelles la fréquence minimale des concepts est la plus grande possible.

- *minimiser l'écart entre les fréquences des clusters* (**minDev**). On impose une petite différence entre les fréquences des clusters : $\text{Min max}(\text{freq}(P_1), \dots) - \text{min}(\text{freq}(P_1), \dots)$.

Cependant, ces deux fonctions souffrent de l'*effet de noyade* [7]. Sur le **maxMin** (resp. **minDev**), le principe de transfert n'est assuré que sur l'utilité **min** (resp. **max**), et donc les utilités intermédiaires ne sont pas nécessairement équitables. Pour assurer l'équité, on considère, dans la section suivante, un opérateur sophistiqué qui raisonne sur l'ensemble des utilités.

2.2 Optimisation multi-agent équitable

Soit $N = \{1, \dots, n\}$ un ensemble de n agents. Une solution d'un problème d'optimisation multi-agent est caractérisée par un vecteur d'utilités $x = (x_1, \dots, x_n) \in \mathbb{R}_+^n$, où x_i représente l'utilité (degré de satisfaction) du i^{me} agent. Les vecteurs d'utilités sont généralement comparés en utilisant la relation de dominance de Pareto (P -dominance). La P -dominance faible \succsim_P entre deux vecteurs d'utilités x, x' est définie par : $x \succsim_P x' \Leftrightarrow [\forall i \in N, x_i \geq x'_i]$, alors que la P -dominance stricte \succ_P entre x et x' est donnée par : $x \succ_P x' \Leftrightarrow [x \succsim_P x' \wedge \text{not}(x' \succsim_P x)]$. Une solution x^* est Pareto-optimale (a.k.a *efficace*) si et seulement s'il n'y a pas de solution x qui domine x^* . L'ensemble des solutions Pareto-optimales forme le front de Pareto $P = \{x \mid \nexists x', x' \succ_P x\}$. La P -dominance peut être formulée comme suit : $\text{max} \{(x_1, \dots, x_n) : x \in Q\}$, où Q est l'ensemble des solutions réalisables. La P -

dominance peut conduire à un grand ensemble de solutions incomparables. De plus, la P -dominance est insensible aux solutions extrémales (*outliers*). Pour favoriser les vecteurs d'utilités *équilibrés*, il est nécessaire de raffiner la relation de P -dominance. L'intuition principale derrière le concept d'équité consiste à choisir des solutions qui partagent équitablement la satisfaction entre les agents [20]. Formellement, une relation de dominance équitable \succsim_{\parallel} doit remplir trois propriétés principales [11, 9] : **(i) Symétrie**. On considère un vecteur d'utilité $x \in \mathbb{R}_+^n$. Pour toute permutation σ sur N , on a $(x_{\sigma(1)}, \dots, x_{\sigma(n)}) \sim (x_1, \dots, x_n)$. Cela signifie que tous les agents ont la même importance. Par exemple, le vecteur d'utilité $(5, 3, 0)$ est considéré comme équivalent au vecteur d'utilité $(0, 3, 5)$. **(ii) P-Monotonie**. Pour tous $x, y \in \mathbb{R}_+^n$, $x \succsim_P y \Rightarrow x \succsim_{\parallel} y$ et $x \succ_P y \Rightarrow x \succ_{\parallel} y$. **(iii) Principe de transfert**. (a.k.a transfert *Pigou-Dalton* dans la théorie du choix social) Soit $x \in \mathbb{R}_+^n$ et $x_i > x_j$ pour certains $i, j \in N$. Soit e^z un vecteur tel que $\forall i \neq z, e_i^z = 0$ et $e_z^z = 1$. Pour tout ϵ où $0 < \epsilon \leq \frac{x_i - x_j}{2}$, on obtient $x - \epsilon e^i + \epsilon e^j \succ_{\parallel} x$. Toute légère amélioration de x_j au détriment (réduction) de x_i , qui préserve la *moyenne des utilités*, produirait une meilleure distribution des utilités parmi les agents et améliorerait par conséquent l'utilité globale de la solution. Par exemple, si l'on considère deux vecteurs d'utilités $x = (11, 10, 7, 10)$ et $y = (9, 10, 9, 10)$, alors le principe de transfert implique que $y = (9, 10, 9, 10)$, car il y a un transfert de taille $\epsilon = 2$ (i.e. $\frac{x_1 - x_3}{2}$), qui permet d'avoir y à partir de x . La combinaison de la P -monotonie et du principe de transfert conduit à ce que l'on appelle *dominance de Lorenz généralisée* définie dans [5] (pour plus de détails, voir [9, 11]).

2.3 Fonctions d'agrégation équitables

Une manière d'évaluer la qualité d'un vecteur d'utilité consiste à agréger les utilités individuelles avec une *fonction d'utilité collective* [13] $G : \mathbb{R}_+^n \rightarrow \mathbb{R}_+$, ce qui améliore le bien-être global avec $\max\{G(x) : x \in Q\}$. La fonction G peut être une combinaison linéaire d'utilités individuelles (i.e. $G(x) \stackrel{\text{def}}{=} \text{sum}(x)$), ce qui ne convient pas au contexte d'équité. Une autre façon de construire G est basée sur la fonction \min (i.e. $G(x) \stackrel{\text{def}}{=} \min(x)$), mais elle est sensible à l'*effet de noyade* [7]. D'autres raffinements de la fonction \min existent (e.g. le \min augmenté, lexmin [3]), mais ne résolvent pas vraiment le problème, car tous sont sensibles à l'*effet de noyade*. Afin de garantir des agrégations équitables, G devrait être conforme aux trois propriétés d'équité. La manière la plus connue est d'utiliser une fonction Schur-convexe ψ , qui préserve les trois propriétés d'équité : $x \succ_{\parallel} y \Leftrightarrow \psi(x) \geq \psi(y)$. Précisément, quand une fonction d'agrégation G est Schur-convexe [11], alors il s'agit d'une agrégation équitable [10]. Ainsi, les fonctions Schur-convexes jouent un rôle clé dans les agrégations équitables (pour plus de détails, voir [11, 10]). Dans la section suivante, nous introduirons une fonction d'agrégation qui assure l'équité.

2.4 Moyennes pondérées ordonnées (OWA)

Cette section se concentre sur les moyennes pondérées ordonnées (en anglais, Ordered Weighted Averages – OWA) [22] définies comme suit :

$$G^w(x) = \sum_{i=1}^n w_i x_{\sigma(i)} \quad (1)$$

où $w = (w_1, \dots, w_n) \in [0, 1]^n$ et $x_{\sigma(1)} \leq x_{\sigma(2)} \leq \dots \leq x_{\sigma(n)}$. OWA fournit une famille de compromis entre les opérateurs sum et \min . Le premier peut être obtenu par $G^{(1/n, 1/n, \dots, 1/n)}$ et le second par $G^{(1, 0, \dots, 0)}$. Golden et Perny [9] proposent des coefficients pour la fonction d'agrégation OWA pour qu'elle soit Schur-convexe :

Théorème 1 [9] *Soient les coefficients suivants de l'agrégation OWA : $W(x) = \sum_{k=1}^n \sin(\frac{(n+1-k)\pi}{2n+1}) x_{\sigma(k)}$. W est une fonction Schur-convexe.*

Le théorème 1 est fondamental, puisque les fonctions Schur-convexes assurent l'équité [10, 9].

3 Modèles PLNE

Cette section décrit les différents modèles PLNE pour trouver un clustering conceptuel équitable. Notre

approche suit l'approche en deux étapes de [16] : (1) un outil dédié à l'extraction de motifs fermés (i.e. LCM [21]) est utilisé pour calculer l'ensemble \mathcal{C} de tous les motifs fermés ; (2) La PLNE est utilisée pour sélectionner un sous-ensemble de \mathcal{C} qui est une partition de l'ensemble \mathcal{D} de transactions et qui optimise un critère donné. Pour favoriser les clusterings équitables, nous améliorons la deuxième étape avec des contraintes supplémentaires assurant une agrégation OWA équitable.

3.1 Modèles PLNE pour l'opérateur OWA

Cette section présente notre première formulation PLNE, appelée modèle OWA de base, pour le clustering conceptuel équitable utilisant un opérateur OWA équitable. Ensuite, nous montrons comment ce modèle de base peut être amélioré en post-traitant les contraintes OWA. Soit \mathcal{D} une base transactionnelle avec m transactions définies sur un ensemble de n items \mathcal{I} . Soit \mathcal{C} l'ensemble de p motifs fermés (par rapport à la mesure de fréquence) représentant les clusters candidats. Soit $a_{t,c}$ une matrice binaire $m \times p$ où $(a_{t,c} = 1)$ ssi $c \subseteq t$, i.e., la transaction t appartient à l'extension du motif fermé c . La matrice $(a_{t,c})$ associée à l'ensemble de données \mathcal{D} de la table 1a est décrite avec la table 1b. Soit v la liste des utilités des motifs fermés (e.g., fréquence, diversité, etc.). Pour chaque motif fermé ($c \in \mathcal{C}$), une variable binaire x_c est associée t.q. $(x_c = 1)$ ssi le cluster c est sélectionné.

(a) Modèle PLNE de base pour OWA. La figure 1a donne le modèle PLNE pour un clustering conceptuel équitable. Il utilise deux types de contraintes : des contraintes de clustering et des contraintes OWA modélisant l'opération de tri requise par l'opérateur OWA :

- **Contraintes de clustering conceptuel.** Les contraintes (C1) exigent que le sous-ensemble de motifs fermés sélectionnés soit une partition de \mathcal{D} . Les contraintes (C2) imposent une borne inférieure k_{\min} et/ou une borne supérieure k_{\max} sur le nombre de motifs fermés sélectionnés.

- **Contraintes OWA.** La fonction objectif et les contraintes (O1) et (O2) implémentent une formulation linéaire connue [15] de l'opérateur OWA pour le clustering conceptuel, où les coefficients ω sont fixés par le théorème 1. Comme expliqué dans la section 2.4, OWA est une somme pondérée sur des utilités triées. C'est pourquoi nous avons introduit r , qui représente la version triée du vecteur d'utilités v . M est une constante suffisamment grande. Soit z une matrice booléenne de taille $|\mathcal{C}|^2$ utilisée pour formuler les contraintes de tri (O1) et (O2). Ces contraintes imposent que le vecteur d'utilités $v \cdot x$ des motifs fermés soit trié par ordre croissant correspondant aux coefficients ω de OWA. Ces contraintes de tri sont expliquées

en détail dans [15]. Il s’ensuit que la k^{me} plus petite valeur d’utilité r_k aura le k^{me} poids le plus élevé ω_k . La fonction objectif maximise la somme pondérée en utilisant les poids ω de OWA donnés par le théorème 1.

(b) **Modèle PLNE amélioré pour OWA.** Afin de trouver un clustering conceptuel équitable, nous proposons un modèle optimisé (voir la fig. 1b) comme suit : (1) Les contraintes de tri (O1) et (O2) sont utilisées lorsque les valeurs d’utilité sont données en compréhension. Comme les valeurs d’utilité des concepts formels sont connues à l’avance, le tri peut être effectué immédiatement après la recherche de motifs fermés. Nous notons v^\uparrow la version triée de v par ordre croissant. (2) Nous associons les poids ω de OWA aux valeurs d’utilité triées, de sorte que toutes les utilités égales auront le même poids. Pour nos expériences, nous avons utilisé le modèle OWA amélioré. Nos résultats préliminaires ont montré que le modèle OWA amélioré surclasse clairement le modèle OWA de base en termes de temps CPU. Ceci est dû au fait que n^2 contraintes supplémentaires et n^2 variables supplémentaires sont utilisées pour coder les contraintes de tri du modèle OWA. Ceci constitue une forte limitation en termes de taille des données qui pourraient être considérées.

Proposition 1 *Les deux modèles PLNE basique et amélioré sont équivalents.*

Preuve 1 *Les deux modèles OWA utilisent les poids ω du théorème 1, ce qui assure une agrégation équitable. OWA amélioré est une optimisation du modèle de base : (1) Il utilise un tri à priori des utilités (pas besoin de contraintes de tri) ; (2) Le même poids est attribué aux utilités égales (même niveau de satisfaction), ce qui préserve directement la conformité avec le théorème 1. Ainsi, les deux modèles OWA sont équivalents. \square*

(c) **Stabilité numérique du modèle PLNE.** L’ensemble des motifs fermés est souvent énorme, ce qui conduit à un très grand vecteur ω dans le modèle OWA de base, et affecte la stabilité numérique du solveur. Le modèle OWA optimisé s’attaque à ce problème, en attribuant le même poids aux utilités égales. Cela permet de résoudre des instances du monde réel dans nos expérimentations exposées dans la section 5.

3.2 Autres modèles PLNE

Comme décrit dans la section 2.1, une agrégation linéaire d’utilités individuelles $\max\{sum(x) : x \in Q\}$ n’assure pas l’équité. Cela suggère de recourir à des opérateurs d’agrégation non linéaires, en particulier le \maxMin et le \minDev . L’agrégation \maxMin $\max\{min(x) : x \in Q\}$ s’attaque à l’équité en améliorant la pire des utilités. Cette fonction peut être

linéarisée en maximisant une variable $z \geq 0$, i.e. une borne inférieure du vecteur d’utilité $v \cdot x$ (voir la figure 2a, inégalité C3), où v est le critère à optimiser (e.g. la fréquence). Le modèle PLNE est donnée dans la figure 2a.

Une autre façon d’assurer l’équité consiste à minimiser l’écart maximal entre la meilleure et la plus mauvaise utilité : $\min\{\max(x) - \min(x) : x \in Q\}$. Le modèle \minDev peut être linéarisé en introduisant $2 \times n$ contraintes et deux variables de décision $z_{max} \geq 0$ et $z_{min} \geq 0$ pour maintenir les valeurs \max et \min du vecteur d’utilités $v \cdot x$ (voir la figure 2b, inégalités C4-C5). Le modèle PLNE résultant est donné par la figure 2b.

4 Travaux connexes

Approches heuristiques. Plusieurs méthodes ont exploré l’idée de séparer la classification de la recherche des descriptions conceptuelles. Pensa *et al.* [18] commencent par extraire les motifs fermés, puis effectuent le clustering k-Means sur ces motifs. Perkowitz et Etzioni [19] inversent les deux phases : la phase *clustering* utilise d’abord une technique de clustering pour former des clusters. À partir des clusters résultants, les descriptions sont apprises par une technique d’apprentissage des règles. Toutes ces techniques sont de nature heuristique et sont fortement influencées par les conditions d’initialisation, nécessitant plusieurs redémarrages, ce qui augmente les coûts de calcul.

Approches déclaratives. Récemment, [16, 17] ont développé des cadres déclaratifs avec la PLNE, qui trouvent des clusterings conceptuels optimaux, où les clusters correspondent à des concepts. Plus tard, Chabert *et al.* ont introduit deux nouveaux modèles PPC pour le clustering conceptuel optimal. Le premier modèle (noté FullCP2) peut être vu comme une amélioration de [6]. Le second modèle (noté HybridCP) suit l’approche en deux étapes de [16] : la première étape est exactement la même ; la deuxième étape utilise la PPC pour sélectionner les concepts formels. Notre travail est différent dans le sens qu’il garantit que le clustering conceptuel trouvé est optimal et *équitable*.

Clustering basé sur la distance vise à trouver des clusters homogènes uniquement sur la base d’une mesure de dissimilarité entre objets. Différents cadres déclaratifs ont été développés, qui s’appuient sur la PPC [6] ou la PLNE [1, 14]. Il existe quelques approches pour obtenir des clusters équilibrés. La plus importante est l’approche proposée par [2]. Elle comporte trois étapes : (1) échantillonnage ; (2) partitionnement de l’ensemble échantillonné et (3) satisfaction des contraintes d’équilibre imposées sur les clusters. Notre adoption des motifs fermés réduit la redondance

$$\begin{array}{l}
\text{Max } \sum_{c=1}^{|\mathcal{C}|} \omega_c \cdot r_c \\
\text{s.t. } \left\{ \begin{array}{l}
\text{Clustering. } \left\{ \begin{array}{l}
\text{(C1)} \quad \sum_{c=1}^{|\mathcal{C}|} a_{t,c} \cdot x_c = 1, \quad \forall t \in \mathcal{D} \\
\text{(C2)} \quad k_{min} \leq \sum_{c=1}^{|\mathcal{C}|} x_c \leq k_{max}
\end{array} \right. \\
\text{Tri OWA. } \left\{ \begin{array}{l}
\text{(O1)} \quad r_c - (v_i \cdot x_i) \leq M z_{c,i}, \quad \forall i, c = 1, \dots, |\mathcal{C}| \\
\text{(O2)} \quad \sum_{i=1}^{|\mathcal{C}|} z_{c,i} \leq c - 1, \quad \forall c = 1, \dots, |\mathcal{C}|
\end{array} \right. \\
x_c \in \{0, 1\}, r_c \in \mathbb{R}_+, \quad \forall c = 1, \dots, |\mathcal{C}| \\
z_{c,i} \in \{0, 1\}, \quad \forall i, c = 1, \dots, |\mathcal{C}|
\end{array} \right.
\end{array}
\quad \text{(a) Mod\`ele OWA de base.}$$

$$\begin{array}{l}
\text{Max } \sum_{c=1}^{|\mathcal{C}|} \omega_c \cdot (v_c^\uparrow \cdot x_c^\uparrow) \\
\text{s.t. } \left\{ \begin{array}{l}
\text{(C1)}, \quad \text{(C2)} \\
x_c \in \{0, 1\}, \\
\forall c = 1, \dots, |\mathcal{C}|
\end{array} \right.
\end{array}
\quad \text{(b) Mod\`ele OWA am\`elior\`e.}$$

FIGURE 1 – Mod\`eles PLNE pour le clustering conceptuel \`equitable bas\`e sur l’op\`erateur OWA.

$$\begin{array}{l}
\text{Max } z \\
\text{s.t. } \left\{ \begin{array}{l}
\text{(C1)}, \quad \text{(C2)} \\
\text{(C3)} \quad z \leq v_c \cdot x_c, \quad \forall c = 1, \dots, |\mathcal{C}| \\
x_c \in \{0, 1\}, \quad \forall c = 1, \dots, |\mathcal{C}| \\
z \geq 0
\end{array} \right.
\end{array}
\quad \text{(a) Mod\`ele PLNE } \mathbf{maxMin}.$$

$$\begin{array}{l}
\text{Max } z_{max} - z_{min} \\
\text{s.t. } \left\{ \begin{array}{l}
\text{(C1)}, \quad \text{(C2)} \\
\text{(C4)} \quad z_{max} \geq v_c \cdot x_c, \quad \forall c = 1, \dots, |\mathcal{C}| \\
\text{(C5)} \quad z_{min} \leq v_c \cdot x_c, \quad \forall c = 1, \dots, |\mathcal{C}| \\
x_c \in \{0, 1\}, \quad \forall c = 1, \dots, |\mathcal{C}| \\
z_{max} \geq 0, z_{min} \geq 0
\end{array} \right.
\end{array}
\quad \text{(b) Mod\`ele PLNE } \mathbf{minDev}.$$

FIGURE 2 – Mod\`eles PLNE pour le clustering conceptuel.

par rapport \`a d’autres fa\`cons de s\`electionner les clusters candidats. De plus, OWA donne des garanties plus fortes sur les clusterings obtenus en termes d’\`equit\`e.

5 Exp\`erimentations

L’\`evaluation exp\`erimentale a pour objectif de r\`epondre aux questions suivantes : (1) passage \`a l’\`echelle des mod\`eles PLNE sur les jeux de donn\`ees consid\`er\`es ; (2) qualit\`e des clusters r\`esultants et leur description ; (3) comparaison (en temps CPU) avec les mod\`eles PPC introduits par Chabert *et al.* [4].

Protocole exp\`erimental. Toutes les exp\`erimentations ont \`ete men\`ees au centre de calcul Cerist¹, o\`u chaque n\`oeud a deux CPUs Xeon E5-2650 avec 16 cores \`a 2.00GHz et 64 Go de RAM. Nous avons utilis\`e LCM pour extraire tous les motifs ferm\`es et CPLEX v.12.6.1 pour r\`esoudre les diff\`erents mod\`eles PLNE. Un *Timeout* de 24 heures a \`ete fix\`e.

Jeux de donn\`ees. Nous avons utilis\`e des jeux de donn\`ees connus provenant du d\`ep\`ot de l’UCI. Nous avons \`egalement consid\`er\`e les m\`emes instances (appel\`ees ERP-*i*, avec $i \in [1, 7]$) utilis\`ees dans [4] et provenant d’un cas d’application r\`eel², qui vise \`a extraire

des concepts de configuration \`a partir d’un progiciel de gestion int\`egr\`e (ERP). Le tableau 2 d\`ecrit les caract\`eristiques de tous les jeux de donn\`ees.

Pour \`evaluer la qualit\`e d’un clustering, nous avons \`evalu\`e la coh\`erence des transactions couvertes par le clustering, qui est mesur\`ee par la similarit\`e intra-cluster (*ICS*) et la dissimilarit\`e inter-clusters (*ICD*), les deux devraient \`etre aussi grandes que possible. La mesure de similarit\`e entre deux transactions t et t' est d\`efinie par : $s : \mathcal{D} \times \mathcal{D} \mapsto [0, 1]$, $s(t, t') = \frac{|t \cap t'|}{|t \cup t'|}$, $ICS(P_1, \dots, P_k) = \frac{1}{2} \sum_{1 \leq i < j \leq k} (\sum_{t, t' \in P_i} s(t, t'))$ et $ICD(P_1, \dots, P_k) = \sum_{1 \leq i < j \leq k} (\sum_{t \in P_i, t' \in P_j} (1 - s(t, t')))$

Pour \`evaluer l’\`equilibre des clusters par rapport \`a la fr\`equence, nous avons utilis\`e trois mesures : (1) Le ratio entre la fr\`equence du plus petit concept et la fr\`equence moyenne (i.e. *Min/Avg*). Pour les m transactions couvertes par les k clusters, *Avg* est calcul\`ee par (m/k) ; (2) L’\`ecart-type sur la fr\`equence (i.e. *StdDev*); (3) La d\`eviation entre la plus petite et la plus grande description des concepts s\`electionn\`es (i.e. *devSize*). Nous notons que ces trois mesures ne consid\`erent pas l’efficacit\`e des solutions.

(a) Analyse qualitative des clusterings. La Fig. 3a compare qualitativement, avec la mesure

1. http://www.rx-racim.cerist.dz/?page_id=26.

2. Ces jeux de donn\`ees sont disponibles sur [http://liris.](http://liris.cnrs.fr/csolnon/ERP.html)

[cnrs.fr/csolnon/ERP.html](http://liris.cnrs.fr/csolnon/ERP.html).

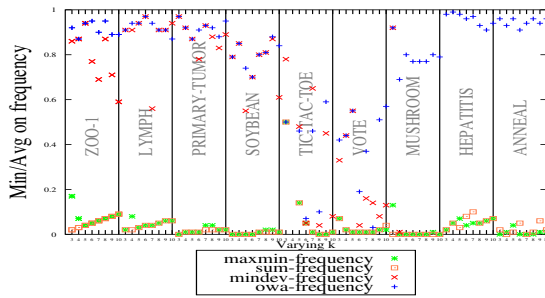
Jeu de données	# \mathcal{D}	# \mathcal{I}	Densité(%)	# \mathcal{C}
Soybean	630	50	32	31,759
Primary-tumor	336	31	48	87,230
Lymph	148	68	40	154,220
Vote	435	48	33	227,031
tic-tac-toe	958	27	33	42,711
Mushroom	8124	119	18	221,524
Zoo-1	101	36	44	4,567
Hepatitis	137	68	50	3,788,341
Anneal	812	93	45	1,805,193

(a) Jeux de données UCI.

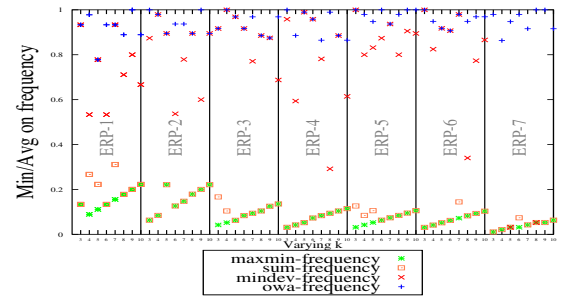
Dataset	# \mathcal{D}	# \mathcal{I}	Density(%)	# \mathcal{C}
ERP-1	50	27	48	1,580
ERP-2	47	47	58	8,1337
ERP-3	75	36	51	10,835
ERP-4	84	42	45	14,305
ERP-5	94	53	51	63,633
ERP-6	95	61	48	71,918
ERP-7	160	66	45	728,537

(b) Jeux de données ERP.

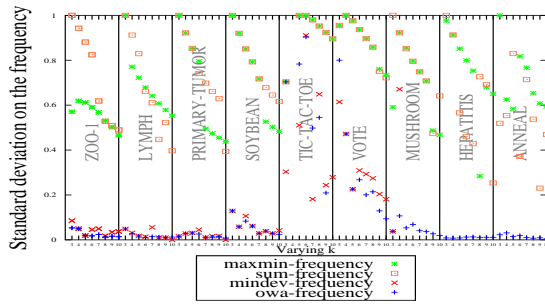
TABLE 2 – Description des jeux de données. Chaque ligne donne le nombre de transactions ($\#\mathcal{D}$), le nombre d'items ($\#\mathcal{I}$), la densité et le nombre de motifs fermés extraits ($\#\mathcal{C}$).



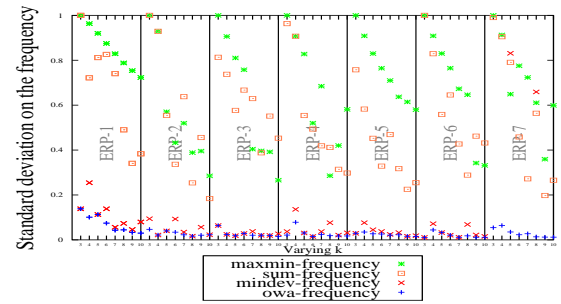
(a) Évaluation (Min/Avg) sur les jeux de données UCI.



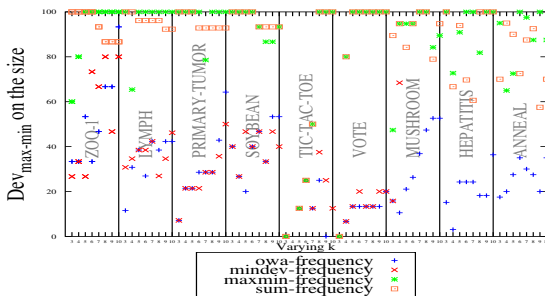
(b) Évaluation (Min/Avg) sur les instances ERP.



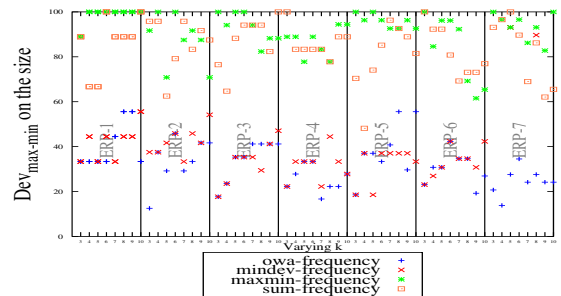
(c) Évaluation $StdDev$ sur les jeux de données UCI.



(d) Évaluation $StdDev$ sur les jeux de données ERP.



(e) Évaluation $devSize$ sur les jeux de données UCI.



(f) Évaluation $devSize$ sur les jeux de données ERP.

FIGURE 3 – Qualité de l'équilibre des clusters résultants des différents modèles PLNE.

\mathcal{D}	k	OWA		minDev		maxMin		maxSum	
		ICS	ICD	ICS	ICD	ICS	ICD	ICS	ICD
Soybean	3	0.447	0.784	0.447	0.784	1.000	0.026	1.000	0.026
	4	0.331	0.865	0.331	0.865	1.000	0.026	1.000	0.026
	5	0.259	0.895	0.284	0.905	1.000	0.026	1.000	0.026
	6	0.231	0.940	0.231	0.940	1.000	0.026	1.000	0.026
	7	0.195	0.964	0.195	0.964	0.959	0.108	0.959	0.108
	8	0.186	0.987	0.186	0.987	0.671	0.474	0.959	0.108
	9	0.166	1.000	0.166	1.000	0.671	0.474	0.959	0.108
	10	0.136	0.999	0.142	0.999	0.670	0.474	0.959	0.108

(a) Maximisation de la fréquence.

\mathcal{D}	k	OWA		minDev		maxMin		maxSum	
		ICS	ICD	ICS	ICD	ICS	ICD	ICS	ICD
soybean	3	0.447	0.776	0.447	0.776	1.000	0.026	0.447	0.776
	4	0.334	0.839	0.338	0.854	1.000	0.026	0.406	0.831
	5	0.296	0.900	0.301	0.900	1.000	0.026	0.389	0.843
	6	0.257	0.929	0.265	0.934	1.000	0.026	0.398	0.851
	7	0.240	0.956	0.240	0.956	0.959	0.106	0.330	0.909
	8	0.220	0.971	0.198	0.978	0.959	0.106	0.323	0.918
	9	0.183	0.991	0.184	0.989	0.959	0.106	0.216	0.975
	10	0.170	0.999	0.157	1.000	0.959	0.106	0.213	0.980

(b) Minimisation de la diversité.

TABLE 3 – Comparaison de la qualité des clusterings résultants en termes de l’ICS et de l’ICD.

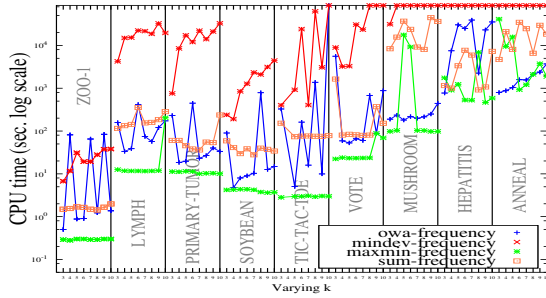
Min/Avg, les clusterings résultants des différents modèles PLNE pour différentes valeurs de k sur les jeux de données de l’UCI. **maxMin** et **maxSum** produisent des clusterings avec un déséquilibre fort par rapport à **OWA** et **minDev** (**maxMin** et **maxSum** atteignent toujours des valeurs *Min/Avg* plus petites). Il est intéressant de noter que **OWA** et **minDev** ont presque la même performance sur les jeux de données avec un nombre de motifs fermés compris entre 10^3 et 10^5 . Cependant, pour les trois jeux de données les plus difficiles – Mushroom, Hepatitis et Anneal – la disparité entre les deux modèles PLNE devient plus prononcée : **OWA** obtient toujours des clusterings les plus équilibrés (valeurs *Min/Avg* proches de 1). Sur ces jeux de données, **minDev** ne parvient pas à trouver une solution même pour des petites valeurs de k . Le même comportement est observé sur les jeux de données ERP (voir Fig. 3b). Sur ERP-7, **minDev** n’a pas été en mesure de trouver une solution. Cela s’explique en partie par le nombre de motifs fermés (10^6), qui génère un nombre important de contraintes par rapport à d’autres jeux ERP (de 10^3 à 10^5). En considérant la mesure *stdDev* (voir les figures 3c et 3d), **OWA** et **minDev** atteignent des valeurs plus petites de *StdDev* sur tous les jeux de données, mais **OWA** est légèrement mieux que **minDev**. Quand on examine la taille des descriptions (voir les figures 3e et 3f), nous pouvons voir que **maxMin** et **maxSum** conduisent à des valeurs *devSize* plus grandes. Ceci est indicatif d’un (ou de quelques) motif(s) ayant une grande fréquence et une petite taille, ou inversement, de motifs ayant une grande taille et une petite fréquence. Ces résultats sont conformes à nos conclusions précédentes. Cependant, pour **minDev** et **OWA**, les solutions optimales trouvées par les deux modèles PLNE tendent à offrir de meilleurs compromis entre les deux critères. Enfin, la Tab. 3 compare³ les quatre modèles selon l’ICS et l’ICD. Nous pouvons voir que **minDev** et **OWA** minimisent l’ICS pour obtenir des valeurs plus élevées de l’ICD. Ce comportement traduit des clusters plus équilibrés : l’ICS est nécessairement

limitée par le nombre de transactions dans un cluster mais l’ICD augmente s’il y a plus de transactions dans d’autres clusters à comparer. **maxMin** et **maxSum** montrent un comportement opposé, produisant un (ou quelques) grands clusters dominants, et de nombreux clusters plus petits.

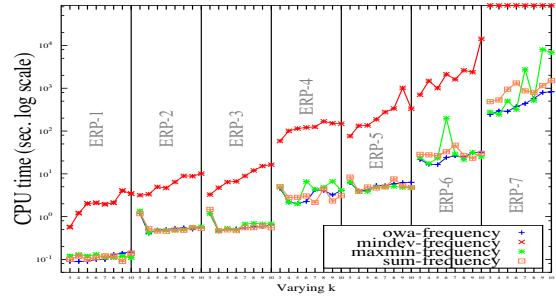
(b) **Passage à l’échelle.** Les figures 4a et 4b comparent les temps CPU pour calculer les clusterings optimaux pour différentes valeurs de k et sur les jeux de données de l’UCI et de l’ERP. Les clusterings maximisent la somme des fréquences des motifs sélectionnés. Les temps CPU comprennent le temps passé par LCM pour extraire tous les motifs fermés. Sur les jeux de données de l’UCI, les performances de **minDev** sont très inférieures par rapport aux autres modèles PLNE. Bien que les résultats qualitatifs de **minDev** soient satisfaisants, ce modèle reste entravé par de longues durées de résolution : il dépasse le *Timeout* sur 32 instances (parmi 72), notamment sur les trois jeux de données les plus difficiles – Mushroom, Hepatitis and Anneal – (voire la Fig. 4a). Cela provient probablement du fait que des contraintes supplémentaires ($2 \times n$) sont utilisées pour capturer la déviation minimale. Cependant, **OWA** donne des résultats assez compétitifs, tout en obtenant des clusterings équitables et optimaux (voir l’analyse qualitative). Il est capable de résoudre tous les jeux et vient en deuxième position. Globalement, **maxMin** obtient les meilleures performances. Cependant, comme noté ci-dessus, les clusterings optimaux trouvés sont loin d’être équilibrés ; ils correspondent à des solutions extrêmes (les pires cas). Ceci pourrait être expliqué en partie par l’approche locale de **maxMin** qui est moins contraignante en obtenant ainsi de bonnes performances. Le même comportement est observé pour **minDev** sur les jeux de données ERP. Enfin, les trois modèles PLNE – **OWA**, **maxMin** et **maxSum** – ont des performances très similaires sur les jeux de données de l’ERP. Nous concluons que le modèle **OWA** offre un bon compromis entre la qualité de la solution et le temps de calcul.

(c) **Modèles PLNE vs. Modèles PPC.** Les figures 5a et 5b comparent les performances de **maxMin**

3. Voir <https://loudni.users.greyc.fr/Cclustering.html> pour d’autres résultats.



(a) Instances de l'UCI : maximisation de la fréquence.



(b) Instances de l'ERP : maximisation de la fréquence.

Instance	OWA avec k non fixé $k \in [3, 10]$		OWA avec k fixé		OWA avec k non fixé $k \in [3, D - 1]$	
	meilleur k	Temps (s.) (2)	meilleur k	Temps (s.) (2)	meilleur k	Temps (s.) (2)
Soybean	10	27.09	10	14.82	501	15.76
Primary-tumor	10	26.81	10	33.34	215	14.52
Lymph	10	77.97	10	173.00	147	20.61
Vote	10	89.8	10	879.22	342	42.3
tic-tac-toe	9	2,104.07	9	9.95	956	11.07
Mushroom	10	377.21	10	442.34	8,123	982.95
Zoo-1	10	5.47	10	1.37	59	0.8
Hepatitis	10	8,462.45	10	35,498.2	136	607.51
Anneal	10	3,674.89	10	3,666.82	459	1,453.04

(c) Maximisation de la fréquence.

Instance	OWA avec k non fixé $k \in [3, 10]$		OWA avec k fixé		OWA avec k non fixé $k \in [3, D - 1]$	
	meilleur k	Temps (s.) (2)	meilleur k	Temps (s.) (2)	meilleur k	Temps (s.) (2)
Soybean	10	13.7	10	165.42	501	9.61
Primary-tumor	10	46.19	10	210.01	215	18.5
Lymph	10	123.84	10	569.63	145	22.05
Vote	10	146.72	10	786.84	342	45.7
tic-tac-toe	9	37,882.31	9	293.82	956	7.21
Mushroom	10	274.62	10	667.99	8,123	1,086.13
Zoo-1	10	0.89	10	1.82	59	0.8
Hepatitis	10	37,915.3	8	6,275.23	136	630.91
Anneal	10	6,839.68	10	25,760.25	459	2,311.01

(d) Minimisation de la diversité.

FIGURE 4 – Analyse des temps CPU.

par la PLNE avec les deux modèles PPC (FullCP2 et HybridCP) qui maximisent la fréquence minimale d'un cluster sur les jeux de données de l'UCI et de l'ERP. Les temps CPU d'HybridCP incluent ceux de l'étape de prétraitement. Le modèle PLNE maxMin est plus performant que FullCP2 et HybridCP sur tous les jeux de données. Aucun des deux modèles PPC ne passe à l'échelle : ils ne parviennent pas à trouver une solution dans la limite de temps pour ($k \geq 4$), sauf pour 4 jeux de données. De plus, le modèle PLNE bat clairement les deux modèles PPC. Enfin, notez que FullCP2 est légèrement mieux que HybridCP.

(d) **Modèle OWA avec k non fixé.** Nous évaluons dans cette expérimentation la capacité du modèle OWA à trouver la solution optimale lorsque k n'est pas fixé. Nous avons sélectionné deux paramètres : $k \in [3, 10]$ (OWA-1) et $k \in [3, |D| - 1]$ (OWA-2). Les Figs. 4c et 4d comparent les temps CPU quand k n'est pas fixé (Colonnes 3 et 7), et quand k est fixé (Col. 5) sur les jeux de données de l'UCI. La Col. 4 rapporte les meilleures valeurs trouvées pour k ($3 \leq k \leq 10$) qui optimisent les deux mesures. Pour tous les jeux de données sauf deux, OWA-1 et OWA-2 sont les approches les plus performantes. OWA-1 est capable de résoudre des instances 5 (resp. 7) plus rapidement en maximisant la fréquence (ou la diversité). Notons que OWA-1 et OWA (avec k fixé) sont similaires sur la meilleure valeur de k . En se comparant avec OWA-1, OWA-2 passe bien à l'échelle, en par-

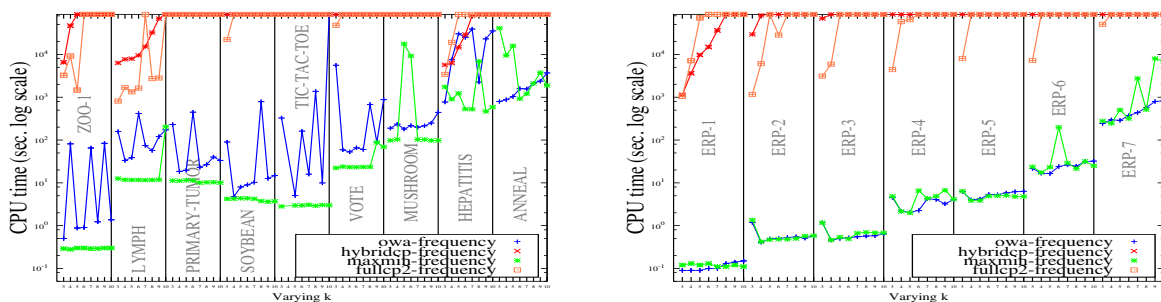
ticulier sur les deux jeux de données les plus difficiles – Anneal et Hepatitis – (*speed-up* jusqu'à 60, 09). En effet, des valeurs plus élevées de k permettent de trouver un clustering plus rapidement qu'avec des valeurs plus petites de k : il y a ($|D| - 1$) clusters pour 3 jeux de données, alors que pour le reste des jeux de données k est plutôt élevé.

6 Conclusion

Nous avons proposé une approche efficace pour la clustering conceptuel équitable. Cette approche utilise l'extraction de motifs fermés pour découvrir des candidats pour les descriptions. L'approche PLNE implémente une fonction d'agrégation équitable basée sur OWA pour sélectionner les meilleurs clusters à fréquences équilibrées. Contrairement aux opérateurs maxMin et minDev, notre approche offre un bon compromis entre la qualité de la solution et le temps de calcul. Nous prévoyons d'étendre notre approche au cas multicritère, où les utilités ne sont pas comparables.

Références

- [1] B. Babaki, T. Guns, and S. Nijssen. Constrained clustering using column generation. In *CPAIOR 2014*, pages 438–454, 2014.



(a) Instances de l'UCI : maximisation de la fréquence. (b) Instances de l'ERP : maximisation de la fréquence.
 FIGURE 5 – Comparaison des temps CPU du modèle PLNE de maxMin avec les deux modèles PPC.

- [2] A. Banerjee and J. Ghosh. Scalable clustering algorithms with balancing constraints. *Data Min. Knowl. Discov.*, 13(3) :365–395, 2006.
- [3] S. Bouveret and M. Lemaître. Computing leximin-optimal solutions in constraint networks. *Artif. Intell.*, 173(2) :343–364, 2009.
- [4] M. Chabert and C. Solnon. Constraint programming for multi-criteria conceptual clustering. In *CP 2017*, volume 10416 of *LNCS*, pages 460–476. Springer, 2017.
- [5] K. M. Chong. An induction theorem for rearrangements. *CJM*, 28 :154–160, 1976.
- [6] T-B-H Dao, K-C Duong, and C. Vrain. Constrained clustering by constraint programming. *Artif. Intell.*, 244 :70–94, 2017.
- [7] D. Dubois and P. Fortemps. Computing improved optimal solutions to max-min flexible constraint satisfaction problems. *EJOR*, 118 :95–126, 1999.
- [8] D. H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2(2) :139–172, 1987.
- [9] B. Golden and P. Perny. Infinite order lorenz dominance for fair multiagent optimization. In *AA-MAS*, pages 383–390, 2010.
- [10] Michael M. Kostreva, Wlodzimierz Ogryczak, and Adam Wierzbicki. Equitable aggregations and multiple criteria analysis. *EJOR*, 158(2) :362–377, 2004.
- [11] W. Marshall and I. Olkin. *Inequalities : Theory of Majorization and its Applications*. Academic Press, London, 1979.
- [12] R. S. Michalski and R. E. Stepp. Learning from observation : Conceptual clustering. In *Machine Learning*, pages 331–363. Springer, 1983.
- [13] Hervi Moulin. *Axioms of Cooperative Decision Making*. Number 9780521360555 in Cambridge Books. Cambridge University Press, 1989.
- [14] M. Mueller and S. Kramer. Integer linear programming models for constrained clustering. In *DS 2010*, pages 159–173, 2010.
- [15] Włodzimierz Ogryczak and Tomasz Śliwiński. On solving linear programs with the ordered weighted averaging objective. *EJOR*, 148(1) :80 – 91, 2003.
- [16] A. Ouali, S. Loudni, Y. Lebbah, P. Boizumault, A. Zimmermann, and L. Loukil. Efficiently finding conceptual clustering models with integer linear programming. In *IJCAI 2016*, pages 647–654, 2016.
- [17] A. Ouali, A. Zimmermann, S. Loudni, Y. Lebbah, B. Crémilleux, P. Boizumault, and L. Loukil. Integer linear programming for pattern set mining ; with an application to tiling. In *PAKDD 2017*, pages 286–299, 2017.
- [18] R. G. Pensa, C. Robardet, and J-F. Boulicaut. A bi-clustering framework for categorical data. In *PKDD 2005*, pages 643–650, 2005.
- [19] M. Perkowitz and O. Etzioni. Adaptive web sites : Conceptual cluster mining. In *IJCAI 99*, pages 264–269, 1999.
- [20] A.K. Sen and J.E. Foster. *On economic inequality*. Clarendon Press, Oxford, 1997.
- [21] T. Uno, T. Asai, Y. Uchida, and H. Arimura. An efficient algorithm for enumerating closed patterns in transaction databases. In *DS 2004*, pages 16–31, 2004.
- [22] Ronald R. Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Trans. Systems, Man, and Cybernetics*, 18(1) :183–190, 1988.
- [23] Y. Yang and B. Padmanabhan. Segmenting customer transactions using a pattern-based clustering approach. In *ICDM*, pages 411–418, 2003.