RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ M'HAMED BOUGARA – BOUMERDES

جامعة بومرداس
Université de Boumerdes
University of Boumerdes

FACULTÉ DES SCIENCES

THÈSE DE DOCTORAT

Présentée par :

BOUDANE Fatima

Spécialité : Informatique

Option : Informatique

Le problème de classification automatique de données : approches mono et multi-objectif

---

Devant le Jury :

M. MEZGHICHE Mohamed	Professeur	UMBB	Président
Mme BENBOUZID-SI TAYEBFatima	Professeur	ESI	Examinatrice
Mme BOUGHACI Dalila	Professeur	USTHB	Examinatrice
M. AMAD Mourad	M.C.A	UAMOB	Examinateur
M. BOULIF Menouar	Professeur	UMBB	Examinateur
M. BERRICHI Ali	Professeur	UMBB	Directeur de thèse

Année Universitaire : 2020/2021

Remerciements

Tout d'abord, je tiens à remercier Dieu le tout puissant et miséricordieux qui m'a donné la force et la patience pour accomplir ce travail.

Je remercier infiniment mon directeur de thèse Mr. Ali BERRICHI, professeur à l'UMBB, pour sa disponibilité, son précieux soutien, ses encouragements, ses conseils judicieux et la confiance dont il m'a fait part lors de la réalisation de ce travail.

Je remercie également les membres de mon jury de soutenance pour m'avoir fait l'honneur d'examiner ce travail ; Mr. Mohamed MEZGHICHE, professeur à l'UMBB, de m'avoir accordé l'honneur de présider mon jury de soutenance, Mme Dalila BOUGHACI, professeur à l'USTHB, Mme Fatima BENBOUZID-SI TAYEB, professeur à l'ESI, Mr. Mourad AMAD, maître de conférences à l'UAMOB et Mr. Menouar BOULIF, professeur à l'UMBB, pour l'intérêt qu'ils ont porté à mon travail en acceptant de l'évaluer.

Je tiens également à remercier Mr. Mohamed MEZGHICHE en sa qualité de directeur du laboratoire LIMOSE pour ses efforts, son aide et ses conseils aussi bien sur le plan pédagogique que scientifique.

Mes profonds et mes plus grands remerciements vont à ma famille : aux deux personnes qui me sont les plus chères au monde ; mes parents, à mes sœurs et frères. Merci pour être toujours présents pour m'encourager. Merci du fond du cœur.

Enfin, merci à tous ceux qui m'ont aidé, encouragé, soulagé, tout au long de mes années d'étude. Merci pour leur présence, leurs discussions, leurs conseils et suggestions éclairées.

Résumé

Le clustering est l'une des tâches les plus importantes et les plus étudiées en data mining. Bien que beaucoup d'algorithmes de clustering aient été proposés dans la littérature de recherche, la plupart d'entre eux ne peuvent pas traiter correctement des ensembles de données ayant des clusters de formes arbitraires et de densité variable. De plus, les plus connus des algorithmes dépendent des paramètres utilisateur qui sont difficiles à définir. Dans le cadre de cette thèse, nous considérons le problème de clustering traitant des ensembles de données avec un nombre inconnu de clusters, ayant des formes arbitraires, présentant des variations de densité et contenant des outliers. Notre motivation principale est de proposer de nouvelles approches permettant d'automatiser le processus de clustering en considérant des ensembles de données possédant toutes ces spécifications. Pour répondre à ces exigences, nous avons proposé, tout d'abord, un nouvel indice de validation du clustering basé sur la connectivité et la densité (CDBCVI), qui permet de faire face au cas de clusters de formes arbitraires et de différentes densités. Il facilite ainsi l'évaluation des algorithmes de clustering et la sélection de leurs paramètres appropriés. Ce nouvel indice est basé sur les relations de densité et de connectivité entre les objets de données, extraites sur la base du graphe de proximité de Gabriel. L'incorporation des relations de connectivité et de densité permet d'obtenir de bons résultats de clustering dans le cas de clusters de n'importe quelle forme, taille ou densité. Par la suite, nous avons proposé trois approches de clustering mono- et multi-objectif qui permettent d'automatiser le processus de clustering et d'améliorer la qualité de ses résultats. Ces approches utilisent un schéma de codage de solutions basé sur la densité, inspiré des algorithmes basés sur la densité NBC (Neighborhood-Based Clustering) et DBSCAN (Density Based Spatial Clustering of Applications with Noise) qui sont très efficaces dans le cas de clusters ayant des formes arbitraires et des densités différentes. La première approche consiste à utiliser la métaheuristique de recherche par voisinage variable (Variable Neighborhood Search (VNS)), afin de remédier à la difficulté du choix de la valeur du paramètre unique de l'algorithme NBC et améliorer ses résultats. La deuxième approche consiste à utiliser l'algorithme de colonie d'abeilles artificielles (Artificiel Bee Colonies (ABC)) afin d'automatiser et améliorer la qualité du clustering de l'algorithme NBC. Quant à la troisième approche, elle consiste à utiliser l'algorithme ABC afin d'automatiser et améliorer la qualité du clustering en s'inspirant de la procédure d'expansion de clusters de l'algorithme DBSCAN. Pour améliorer le processus d'évaluation des solutions de clustering au cours des itérations, nous avons défini plusieurs fonctions objectif basées sur des concepts de densité, vu que la prise en compte d'une seule fonction objectif peut ne pas être conforme aux ensembles de données ayant des clusters de formes complexes et des outliers.

Nous avons testé la performance des approches proposées par une expérimentation approfondie sur des ensembles de données réels et synthétiques. Les résultats expérimentaux démontrent l'efficacité et la supériorité des approches proposées par rapport à plusieurs d'autres approches de la littérature.

***Mots-clés* :** Clustering, optimisation multi-objectif, algorithme de colonie d'abeilles artificielles, densité, voisinage, connectivité, clusters de forme arbitraire, indice de validation du clustering, graphe de Gabriel.

Abstract

Clustering is one of the most important and well-studied data mining tasks. Although many clustering algorithms have been proposed in the literature, most of them cannot properly handle datasets having arbitrarily shaped clusters with varying density and depend on the user-defined parameters, which are hard to set. In the framework of this thesis, we consider the clustering problem in datasets with unknown number of clusters having arbitrary shapes and which present density variations and outliers. In the presence of all these specifications, our main motivation is to propose approaches allowing to automate the process of clustering, in order to avoid the difficulty encountered by the user in determining the best values of the input parameters in the case of density-based clustering algorithms and improve the quality of clustering results. To meet these requirements, we first propose a new clustering validation index, based on connectivity and density (CDBCVI), which makes it possible to deal with the case of clusters of arbitrary shapes and different densities and thus, facilitates the evaluation of clustering algorithms and the selection of their appropriate parameters. Our new index is based on the density and connectivity relationships between data objects, extracted based on the proximity graph Gabriel. Incorporating connectivity and density relationships allows good clustering results to be achieved, in the case of clusters of any shape, size or density. We propose, then, three mono- and multi-objective clustering approaches that automate the clustering process and improve the quality of its results. These approaches use a density-based solution encoding scheme, inspired from the NBC (Neighborhood-Based Clustering) and DBSCAN (Density Based Spatial Clustering of Applications with Noise) algorithms which are very efficient in the case of clusters having arbitrary shapes and different densities. The first approach consists to use the Variable Neighborhood Search (VNS) metaheuristic, in order to overcome the difficulty of choosing the value of the single parameter of the NBC algorithm and improve its results. The second consists to use the Artificial Bee Colonies (ABC) algorithm to automate and improve the quality of clustering of the NBC algorithm. As for the third, it consists to use the ABC algorithm to automate and improve the quality of clustering, based on the cluster expansion process of the DBSCAN algorithm. To improve the clustering solutions evaluating process during iterations of these approaches, we define several objective functions based on density concepts, since taking into account a single objective function may not conform to datasets having clusters of arbitrary shapes and outliers.

We tested the performance of the proposed approaches through in-depth experimentation on real and synthetic datasets. The experimental results demonstrate the effectiveness and the superiority of the proposed approaches compared to several other approaches in the literature.

Keywords **:** Multi-objective clustering, artificial bee colony algorithm, density-based clustering, neighborhood, connectivity, density, clusters of arbitrary shape, clustering validation index, Gabriel graph.

ملخص

يعد التجميع أحد أكثر مهام البحث عن البيانات إثارة للاهتمام والأكثر دراسة. على الرغم من أنه تم اقتراح العديد من خوارزميات التجميع، إلا أن معظمها لا يمكنها التعامل بشكل صحيح مع مجموعات البيانات التي تحتوي على مجموعات ذات شكل عشوائي و كثافة متفاوتة وتعتمد على عوامل يصعب تحديد قيمها من طرف المستخدم. في إطار هذه الأطروحة، نأخذ في الاعتبار مشكلة التجميع في مجموعات البيانات ذات عدد مجموعات غير معروف، ذات الأشكال العشوائية و التي تتضمن اختلافات في الكثافة وقيم متطرفة. في ظل وجود كل هذه المواصفات، فإن دافعنا الرئيسي هو اقتراح أساليب تسمح بتجنب الصعوبة التي يواجهها المستخدم في تحديد أفضل قيم لمعلمات الإدخال في حالة خوارزميات التجميع المستندة على الكثافة و بتحسين جودة نتائج التجميع. لتحقيق هذه المتطلبات، نقترح أولاً مؤشرا جديدًا للتحقق من صحة المجموعات، بناءً على الاتصال والكثافة، مما يجعل من الممكن التعامل مع حالة المجموعات ذات الأشكال العشوائية والكثافات المختلفة، وبالتالي يسهل تقييم خوارزميات التجميع واختيار قيم المعلمات المناسبة لها. يعتمد مؤشرنا الجديد على علاقات الكثافة والاتصال بين البيانات المستخرجة بناءً على الرسم البياني التقريبي غابرييل. يتيح دمج علاقات الاتصال والكثافة تحقيق نتائج تجميع جيدة للمجموعات ذات أي شكل أو حجم أو كثافة. ثم نقترح بعد ذلك، ثلاثة طرق تجميع فردية ومتعددة الأهداف تعمل على أوتوماتكية عملية التجميع وتحسين جودة نتائجها. تستخدم هذه الطرق مخطط ترميز حلول قائم على الكثافة، مستوحى من الخوارزميات القائمة على الكثافة وهي فعالة جدًا في حالة وجود مجموعات ذات أشكال عشوائية وكثافة مختلفة. تتمثل الطريقة الأولى في استخدام طريقة البحث المتغير في الجوار للتغلب على صعوبة اختيار قيمة المعلمة الفردية لخوارزمية التجميع القائم على الجوار ولتحسين نتائجها. الطريقة الثانية تتمثل في استخدام خوارزمية مستعمرات النحل الاصطناعية لأوتوماتكية وتحسين جودة تجميع خوارزمية التجميع القائم على الجوار. أما الطريقة الثالثة فهي تستخدم خوارزمية مستعمرات النحل الاصطناعية لأوتوماتكية وتحسين جودة التجميع من خلال استلهام الإلهام من إجراء توسيع المجموعات للخوارزمية القائمة على الكثافة للتجميع المكاني للتطبيقات ذات الضوضاء. لتحسين عملية تقييم حلول التجميع أثناء أداء هذه الطرق، نعرف العديد من الدوال الهادفة بناءً على مفاهيم الكثافة، نظرًا لأن مراعاة دالة هادفة واحدة قد لا تتوافق مع مجموعات البيانات التي تحتوي على مجموعات ذات الأشكال المعقدة والقيم المتطرفة.
اختبرنا أداء الأساليب المقترحة من خلال تجارب متعمقة على مجموعات بيانات حقيقية واصطناعية. أظهرت النتائج التجريبية فعالية وتفوق الأساليب المقترحة مقارنة بالعديد من الأساليب الأخرى المقترحة في المؤلفات.

الكلمات المفتاحية: التجميع متعدد الأغراض، خوارزمية مستعمرة النحل الاصطناعية، التجميع القائم على الكثافة، الجوار، الاتصال، الكثافة، مجموعات الشكل العشوائي، مؤشر التحقق من التجمعات، الرسم البياني غابرييل.

Table des Matières

CHAPITRE 4. L'indice de validation de clustering basé sur la connectivité et la densité

CHAPITRE 5. Clustering automatique à base de densité utilisant des métaheuristiques mono et multi-objetif

Liste des figures

Liste des tableaux

Introduction générale

Aujourd'hui, avec les progrès technologiques et les évolutions du stockage et du traitement des données, la quantité de données stockées sous format numérique a augmenté considérablement et rapidement dans plusieurs domaines d'application. D'où la nécessité de nouvelles approches efficaces pour l'extraction de connaissances cachées dans ces ensembles de données, utiles pour l'aide à la décision. L'exploration de données ou data mining est devenue un domaine interdisciplinaire pour l'extraction de modèles intéressants, précédemment inconnus et potentiellement utiles dans un ensemble de données. Il a été appliqué avec succès pour résoudre plusieurs problèmes réels dans différents domaines, tels que la gestion de la relation client (CRM pour *Customer Relationship Management*), la détection de fraudes, l'astronomie, la géographie et la fabrication.
Le data mining comprend plusieurs tâches telles que la classification, le clustering, la régression et la détection d'anomalies.

Dans cette thèse, nous nous concentrons sur le clustering qui est considéré comme la tâche la plus importante et la plus difficile du data mining (Han et al., 2012).

Le clustering consiste à grouper des objets dans des classes (ou clusters) tels que les objets similaires sont affectés au même cluster, tandis que ceux qui sont dissimilaires sont affectés à des clusters différents. Il s'agit d'une méthode d'apprentissage non supervisée (classification non supervisée) où les regroupements sont explorés sans aucune orientation ni utilisation d'informations externes (à la différence d'une méthode de classification supervisée dans laquelle, à partir de données *(individu, classe)*, il s'agit d'apprendre à classer un nouvel individu dans l'une des classes prédéfinies). Le clustrering a été appliqué dans différents domaines de recherche tels que le clustering de documents, la segmentation d'images et la reconnaissance de formes. Plusieurs difficultés se posent avec le problème de clustering, telles que : le nombre inconnu de clusters, le traitement des clusters de forme arbitraire, les variations de densité, le traitement de données mixtes, la définition des objectifs de clustering et les contraintes spécifiques dans certains domaines d'application. Afin d'aboutir à de bons résultats, l'objectif de clustering, le domaine d'application et les exigences de clustering spécifiques au domaine doivent être clarifiés explicitement. Ensuite, les exigences de clustering sont mises en correspondance avec les capacités et les hypothèses des approches de clustering pour sélectionner l'approche la plus appropriée.

Dans cette thèse, nous nous concentrons plus particulièrement sur les problèmes de clustering généralement illustrés dans les ensembles de données spatiales (les ensembles de données spatiales sont particulièrement rencontrés dans les systèmes d'information géographiques, les graphiques basés sur des points et les systèmes biomédicaux) où l'emplacement des objets et leurs relations géométriques sont importants. Autrement dit, la topologie, la proximité et la connectivité deviennent les problèmes clés du clustering. Dans le domaine spatial, une grande variété de caractéristiques de clusters, en particulier des formes arbitraires, des variations de densité et le nombre inconnu de clusters sont observées.

Dans la littérature, plusieurs méthodes ont été développées pour résoudre le problème de clustering. En général, les méthodes de clustering peuvent être classées principalement en

méthodes hiérarchiques et méthodes de partitionnement. Chaque catégorie contient plusieurs approches de résolution qui définissent le problème de clustering dans des perspectives différentes. Cependant, la majorité des algorithmes de clustering existants, tels que *k*-means (Hartigan & Wong, 1979), ne peuvent pas traiter correctement les clusters de forme arbitraire et contenant des données isolées. De plus, ils dépendent des paramètres définis par l'utilisateur et souffrent du problème bien connu des minima locaux. Bien que les algorithmes de clustering basés sur la densité (Ester et al., 1996; Zhou et al., 2005; Mishra & Mohanty, 2019; Shi et al., 2018) puissent trouver des clusters de formes arbitraires dans des ensembles de données, la plupart d'entre eux ne peuvent pas traiter correctement les ensembles de données de densité variable et ils dépendent des paramètres définis par l'utilisateur.

Récemment, plusieurs approches de clustering mono et multi-objectif, basées sur des métaheuristiques, sont proposées dans la littérature (Zhu et al., 2019; Kumar et al., 2017). Pour trouver de bonnes solutions de clustering en temps raisonnable, plusieurs travaux ont combiné des algorithmes de clustering classiques avec des métaheuristiques tels que l'optimisation par essaim de particules (Particle Swarm Optimization (PSO)) (Armano & Farmani, 2016; Guan et al., 2019), les algorithmes de colonie d'abeilles artificielles (Artificial Bee Colony (ABC)) (Ranjbar et al., 2015; Danish et al., 2019) et les algorithmes génétiques (Genetic Algorithms (GAs)) (Rahman et Islam, 2014; Sabau, 2012). Cependant, la majorité de ces approches de clustering ne peuvent pas traiter correctement des clusters avec des formes arbitraires et des densités différentes et ils dépendent aussi des valeurs des paramètres définis par l'utilisateur qui sont difficiles à estimer.

Dans le cadre de cette thèse, nous abordons le problème de clustering avec les caractéristiques suivantes :

(1) Le nombre de clusters est inconnu.
(2) Un cluster est composé d'objets de données connectés dans une région dense qui est entourée de régions de faible densité ou vice versa.
(3) Les clusters peuvent avoir des formes arbitraires.
(4) Il peut y avoir des variations de densité au sein des clusters tant qu'elles gardent une cohérence le long d'une direction.
(5) Différents clusters peuvent avoir des densités différentes.

Afin de résoudre le problème de clustering avec ces spécifications, les problèmes clés à prendre en compte deviennent la connectivité, la densité et la proximité. Nous visons, donc, à développer des approches de clustering sans paramètres pour des ensembles de données multidimensionnelles, en utilisant les principaux concepts du clustering spatial dans leur intégralité, à savoir la connectivité, la densité et la distance, tout en s'inspirant des avantages des métaheuristiques. Pour réaliser notre étude, nous avons choisi deux métaheuristiques qui sont simples et efficaces et elles ont un nombre limité de paramètres de contrôle, à savoir : l'algorithme de recherche par voisinage variable (Variable Neighborhood Search (VNS)) et l'algorithme de colonie d'abeilles artificielles (ABC). Dans cette thèse, nous avons développé plusieurs approches contribuant à la résolution du problème de clustering dont les résultats des tests expérimentaux obtenus sont très satisfaisants. Les contributions de cette thèse consistent en :

1. La proposition d'un nouvel Indice de Validation de Clustering basé sur la densité. Il permet de faire face au cas de clusters de formes arbitraires et de différentes densités. Il facilite ainsi l'évaluation des algorithmes de clustering et la sélection de leurs paramètres appropriés.
2. La proposition d'approches de clustering mono- et multi-objectif qui permettent de traiter les spécifications du problème posé. Plusieurs fonctions objectif sont proposées, permettant ainsi d'améliorer l'évaluation de la qualité des solutions de clustering en remplacement des indices de validité conventionnels, pouvant échouer dans le cas de clusters de formes arbitraires.

Cette thèse s'articule autour du plan suivant :

Dans le chapitre 1, nous présentons le problème de clustering et décrivons brièvement les concepts de base du clustering. Nous discutons, aussi, les caractéristiques et la classe des problèmes de clustering et pourquoi le clustering est un problème difficile.

Dans le chapitre 2, les propriétés et capacités des différentes approches de clustering existantes dans la littérature sont passées en revue. Les avantages et les inconvénients de ces approches sont donnés dans ce dernier.

Vu l'importance considérable des algorithmes de clustering à base de densité, notamment pour le traitement des clusters de formes arbitraires, nous présentons brièvement deux algorithmes bien-connus basés sur la densité, à savoir l'algorithme DBSCAN (Ester et al., 1996) et l'algorithme NBC (Zhou et al., 2005). Aussi, nous présentons, plus particuèrement, les travaux les plus importants à base de métaheuristiques. Nous terminons ce chapitre par une discussion sur les difficultés non encore résolues du problème de clustering.

Pour compléter la présentation du contexte de notre étude, nous présentons dans le chapitre 3 les techniques d'évaluation du clustering. Nous discutons les critères les plus importants dans le contexte de l'évaluation de la validité du clustering. Les différents indices de validation proposés dans la littérature sont discutés dans ce chapitre.

Les chapitres 4 et 5 sont consacrés à la présentation de nos contributions.

Dans le chapitre 4, nous présentons en détails la première contribution de cette thèse, qui est la proposition d'un nouvel indice de validation du clustering. Afin de montrer l'efficacité de l'indice proposé pour l'évaluation des algorithmes de clustering et la sélection de leurs paramètres appropriés, nous terminons le chapitre par une présentation de l'étude expérimentale menée sur des ensembles de données synthétiques et réels, en utilisant les algorithmes de clustering bien connus NBC et DBSCAN.

Dans le chapitre 5, nous présentons les autres contributions de cette thèse, consistant en la proposition de nouvelles approches de clustering permettant de remédier aux difficultés et lacunes posées par la plupart des approches de clustering existantes dans la littérature. Nous présentons, dans un premier temps, la première contribution qui permet d'automatiser le processus de clustering en utilisant la métaheuristique VNS et l'algorithme NBC. Dans un deuxième temps, nous présentons deux autres contributions qui utilisent l'algorithme ABC et qui sont basées, respectivement, sur les algorithmes NBC et DBSCAN, afin d'automatiser et améliorer la qualité du clustering. Afin de montrer l'efficacité des différentes approches

proposées, une étude expérimentale comparative a été réalisée sur plusieurs ensembles de données avec des algorithmes de clustering bien connus.

Dans la conclusion générale nous synthétisons la problématique étudiée et les contributions apportées au domaine de recherche. Ensuite, nous présentons quelques perspectives que nous comptons explorer pour enrichir le domaine de recherche de notre étude.

Chapitre 1

Concepts du Clustering

Sommaire

1.1. Introduction

Dans ce chapitre, nous présentons le problème de clustering et décrivons brièvement ses concepts de base. Nous discutons par la suite les caractéristiques et la classe des problèmes de clustering et pourquoi le clustering est un problème difficile. Autrement dit, la motivation de ce travail est justifiée, dans ce chapitre.

1.2. Le problème de Clustering

Le clustering (ou regroupement) est l'un des processus essentiels et les plus étudiés pour l'exploration de données et la découverte de connaissances. Il consiste à regrouper un ensemble de données en des groupes (ou clusters en anglais) homogènes de telle sorte que les objets dans un même cluster soient similaires et différents des objets des autres clusters, dans le contexte du problème défini. Le clustering a plusieurs applications dans différents domaines tels que : vision par ordinateur (Belahbib & Souami, 2011), catégorisation des documents (Mecca et al., 2007), bioinformatique (Valafar, 2002), reconnaissance de formes (Mirkin, 2005), biologie (Sneath & Sokal, 1973), traitement d'image (Chou et al., 2004) et sécurité informatique (Barbarà & Jajodia, 2002). La difficulté principale dans le problème de clustering est le fait que les caractéristiques structurales des données multidimensionnelles sont inconnues (c-à-d, la distribution des données en termes de nombre, la densité et la forme de clusters sont inconnues).

1.3. Notions de base du clustering

Dans cette section, nous définissons les notions de bases utiles pour la suite de l'étude du problème de clustering.

Pour fixer le contexte et clarifier la terminologie prolifique, nous considérons un ensemble de données *D* composé de *N* points de données (ou de manière synonyme, objets, instances, cas, modèles, tuples, transactions) $x_i = (x_{i1}, x_{i2}, \dots, x_{im}) \in A$, ou $i = 1, N$ et chaque composant $x_{ij} \in A_j$ est un attribut numérique ou catégoriel (ou de manière synonyme, fonctionnalité, variable, dimension, composant, champ). Une discussion sur les types d'attributs de données est disponible dans (Han & Kamber 2001). Un tel format de données par attribut correspond conceptuellement à une matrice de dimension $N \times m$, utilisé par la majorité des algorithmes de clustering. Le clustering permet de générer un ensemble de k clusters $C = \{C_1, \dots., C_k\}$ tel que $C_i \subset D$ et $\bigcup_{i=1}^{k} C_i = D$.

Trois formalismes de clustering existent : le clustering dur (*hard* ou *crisp-clustering*), le clustering avec recouvrements et le clustering flou (*fuzzy-clustering*).
Le clustering dur consiste à classer chaque objet dans une et une seule classe. Le résultat dans ce cas est une partition ou hiérarchie stricte selon la méthode utilisée.

Le clustering flou consiste à classer chaque objet d'une façon fractionnaire dans les différents clusters (c'est-à-dire qu'il est partagé entre les différents clusters) $\{C_1, C_2, \dots, C_k\}$, en utilisant des fonctions d'appartenance (appelées aussi fonctions d'affectation) $\{f_j\}_{j=1\dots k}$. Ces fonctions permettent de tenir compte des incertitudes et ambiguïtés pouvant intervenir dans l'appartenance d'un objet à une classe. Les algorithmes de clustering flou sont très utilisés dans différents domaines d'applications tels que le regroupement de données textuelles ou la segmentation d'images (Chuai-Aree et al., 2000).

Le clustering avec recouvrements (appelé parfois soft-clustering) consiste à classer chaque objet d'une façon dure à une ou plusieurs classes. Il existe très peu d'approches de clustering avec recouvrements.

La plupart des travaux de synthèse sur le clustering ne concernent que le clustering dur dont l'avantage est de proposer un regroupement non ambigüe, simple et facile à utiliser pour d'autres traitements (Berkhin, 2000; Grabmeier & Rudolph, 2002 ; HÖsel & Walcher, 2000; Cleuziou, 2004).

Dans notre étude, nous nous intéressons à étudier, à travers différentes approches, le premier formalisme qu'est le clustering dur. Les termes « mesure de similarité/dissimilarité » et « distance » sont utilisés de manière interchangeable, tout au long de ce document. Cela est principalement dû au fait que nous nous concentrons sur les données spatiales qui sont sous forme numérique pure.

1.3.1. Définitions

1.3.1.1. Partitions, pseudo-partitions et partitions floues

Définition 1.1. *C* est une partition de *D*, appelée aussi « partition stricte », si et seulement si les propriétés suivantes sont vérifiées :

- $C_i \subset D, \forall\, C_i \in C$,
- $\bigcup_{i=1}^{k} C_i = D$,
- $C_i \cap C_j = \phi$ tel que $i \neq j$.

Définition 1.2. *C* est une pseudo-partition de *D* si et seulement si les propriétés suivantes sont vérifiées :

- $C_i \subset D, \forall\, C_i \in C$,
- $\bigcup_{i=1}^{k} C_i = D$,
- $C_i \subseteq C_j$ si et seulement si $i = j$.

Définition 1.3. C est une partition floue de *D* si et seulement si elle est définie par la donnée de k fonctions : $f_j : D \to [0,1],\ j = 1 \ldots k$, tel que $f_j(x_i)$ représente le degré d'appartenance de l'objet x_i au cluster C_j.

De la même façon, la construction d'une partition ou d'une pseudo-partition peut être formalisée par la donnée de *k* fonctions à valeurs binaires comme suit :

$f_j : D \to \{0,1\},\ j = 1 \ldots k$, tel que $f_j(x_i) = \begin{cases} 1 & si\ x_i \in C_j \\ 0 & sinon \end{cases}$,

À la différence d'une partition, dans une pseudo-partition nous pouvons y avoir des intersections entre clusters mais sans inclusion entre eux.

1.3.1.2. Hiérarchies et pseudo-hiérarchies

Le résultat de certaines méthodes de clustering est un arbre hiérarchique (appelé aussi dendrogramme). Dans ce cas la distinction entre hiérarchies et pseudo-hiérarchies est faite de même qu'entre partitions et pseudo-partitions. Pour plus de détails, voir (Cleuziou, 2004).

1.3.1.3. Centroïdes et médoïdes

Certaines méthodes de clustering utilisent des points représentatifs des clusters (appelés centroïdes ou médoïdes) pour diminuer la complexité et faciliter la manipulation de ces derniers.

Définition 1.4. Le centroide x_* d'un cluster C_l est le point défini par :

$$\forall\, j = 1, \ldots, m\ , x_{*j} = \frac{1}{|C_l|} \sum_{x_i \in C_l} x_{ij}$$

Dans le cas d'attributs quantitatifs (numériques), le centroïde d'un cluster représente la moyenne de tous les points composant le cluster. Généralement, le centroïde d'un cluster ne représente pas forcément un objet du cluster mais il correspond à son centre de gravité. Par contre, dans le cas où certains attributs sont qualitatifs (catégoriels), la définition du centroïde n'est pas applicable et les clusters sont représentés par l'objet le plus représentatif appelé médoïde (ou prototype).

Définition 1.5. Le médoïde d'un cluster C_l est l'objet $x_* \in C_l$ défini par :

$$x_* = \underset{x_i \in C_l}{argmin} \frac{1}{|C_l|} \sum_{x_j \in C_l} d(x_i, x_j)$$

D'après cette définition, le médoïde d'un cluster est un élément du cluster ayant une dissimilarité moyenne minimale avec tous les autres objets du cluster. Autrement dit, le médoïde d'un cluster est l'objet le plus similaire aux autres (en moyenne).

1.3.1.4. Les outliers (les points aberrants)

Les outliers sont des objets non conformes au comportement ou au modèle général des données (objets atypiques). La notion de $DB(m, \delta) - Outlier$ (Distance-Based Outlier) (Knorr & Ng, 1998) est souvent utilisée comme définition formelle d'un outlier. Elle est donnée comme suit :

Définition 1.6. *Un objet $x_* \in D$ est un $DB(m, \delta) - Outlier$ s'il existe un sous ensemble D' de D constitué d'au moins t objets $x'_1, \dots, x'_t$ tel que $\forall\, x'_i \in D', d(x_*, x'_i) > \delta$, où d est une mesure de dissimilarité définie sur D.*

La détection d'outliers est difficile mais très importante puisque ces derniers peuvent faire entrer un biais dans le processus de clustering. Les outliers peuvent être détectés à l'aide de tests statistiques qui supposent une distribution ou un modèle probabiliste pour les données, ou en utilisant des mesures de distance où les objets qui sont éloignés de tout autre groupe sont considérés comme outliers (Han et al., 2012).

1.3.2. Caractéristiques du problème de clustering (difficultés posées)

Le problème de clustering est NP-difficile (Garey & Johnson, 1979). Il est impossible d'énumérer toutes les solutions possibles en un temps raisonnable, étant donné qu'il existe $\frac{1}{k!}\sum_{i=0}^{k}(-1)^{k-1}\binom{k}{i}i^n$ solutions possibles à ce problème (Anderberg, 1973). En plus, du caractère fortement combinatoire du problème, d'autres difficultés se posent telles que : le nombre inconnu de clusters, le traitement de clusters de formes arbitraires, les variations de densité, le traitement de données mixtes, la définition des objectifs de clustering et les contraintes spécifiques.

Généralement, le nombre de clusters dans un ensemble de données n'est pas connu a priori. En plus, un ensemble de données peut avoir des clusters de formes arbitraires avec des variations de densité et possiblement contenir des outliers. Dans ce cas, la définition et le calcul des objectifs de compacité, de séparation et de connectivité deviennent compliqués. Aussi, les attributs d'un ensemble de données peuvent comprendre divers types de données (numériques, ordinales et nominales), et cela affecte le calcul de similarité/dissimilarité entre des paires d'objets de données. Conceptuellement, le clustering vise à obtenir des clusters compacts, connectés et bien séparés. Cependant, il est difficile de quantifier et de combiner ces objectifs et de trouver un seul objectif à optimiser afin de trouver les clusters cibles en fin de compte. Dans certains domaines d'applications de clustering, des contraintes spécifiques, dont certaines sont en contradiction avec la tendance de clustering naturel de l'ensemble de données, sont imposées. Le clustering par contraintes est vu comme un apprentissage non-supervisé dans lequel des contraintes doivent être satisfaites lors du clustering. Des

mécanismes supplémentaires sont donc nécessaires dans l'algorithme de clustering pour leur prise en compte. Enfin, l'analyse d'ensembles de données volumineux exige des temps de traitement longs et une grande capacité en mémoire.

Quelques problèmes difficiles sont discutés en détail comme suit.

- **Le nombre de clusters**

Le nombre de clusters peut être inconnu pour le clustering. Certains algorithmes de clustering de la littérature supposent qu'un nombre fixe de clusters est donné a priori. Cependant, plusieurs domaines d'applications de clustering dans le monde réel n'ont pas cette connaissance préalable, tels que la segmentation d'image, la bioinformatique et les systèmes d'information géographiques. Certaines approches utilisent un indice de validation pour trouver le nombre de clusters. Ces approches exécutent l'algorithme de clustering avec différentes valeurs du nombre de clusters, par la suite sélectionnent celui de la partition ayant la meilleure valeur de l'indice de validation. Cependant, les indices de validation existants ne sont pas valables dans toutes les situations, pour les même raisons données pour la définition des objectifs de clustering.

- **Type de données**

Les ensembles de données peuvent inclure deux types d'attributs : qualitatif ou catégoriel (binaire, nominal ou ordinal) ou quantitatif (continu, discret ou intervalle). Cette variété affecte particulièrement le calcul de la mesure de similarité/dissimilarité. Dans le cas d'attributs quantitatifs (par exemple taille, poids), la dissimilarité est mesurée par des fonctions de distance telles que la distance de Manhattan, la distance Euclidienne, la distance de Mahalanobis et la distance de Minkowski. Dans le cas d'attributs qualitatifs (par exemple, couleur, rang), des mesures de similarité bien connues sont les distances de Jaccard et de Hamming. Dans le cas de présence d'attributs à la fois quantitatif et qualitatifs dans un ensemble de données (c'est-à-dire cas des données mixtes), il est difficile de mesurer la similarité/dissimilarité entre les objets. La distance de Minkowski généralisée dans (Ichino & Yaguchi, 1994) est utilisée dans des ensembles de données mixtes (Jain et al., 1999; Han & Kamber, 2001; Xu & Wunsch, 2005). Afin d'assurer la comparabilité entre les attributs, les attributs sont normalisés par diverses méthodes de mise à l'échelle telles que min-max, z-score et normalisation décimale. Cette mise à l'échelle devient problématique lorsque l'ensemble des données comprend différents types d'attributs car les données binaires dominent généralement les données numériques (Jain et al., 1999). Par conséquent, les types d'attributs et la mesure de dissimilarité utilisée ont des effets importants sur le développement d'un algorithme de clustering.

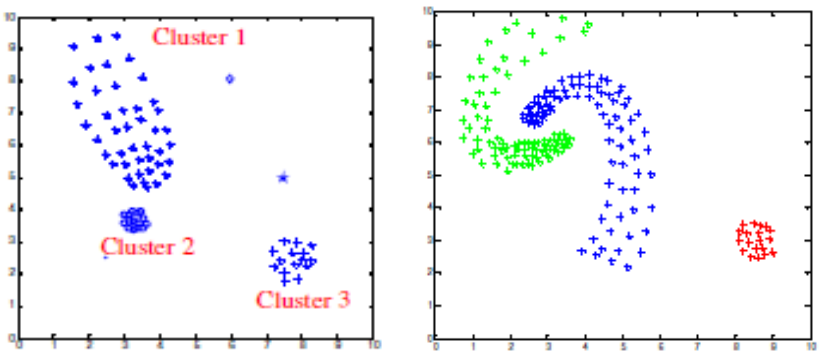
- **Objectifs de clustering et variations de densité**

Un algorithme de clustering doit prendre en compte trois objectifs, à savoir : compacité, séparation et connectivité des clusters. En effet, le clustering consiste à partitionner les données de telle sorte que les objets du même cluster soient similaires (i.e. compacité) et ceux de différents clusters soient dissimilaire (i.e. séparation), en plus chaque objet est affecté au même cluster que ses proches voisins. Toutefois, il est difficile de trouver des mesures qui quantifient et combinent ces objectifs, pour des ensembles de données provenant de divers domaines.

L'objectif de compacité garantit la cohérence des objets de données dans un cluster en termes de similarité et densité intra-clusters. Dans la littérature, les mesures de compacité peuvent être classées en deux catégories : mesures basées sur des objets représentatifs et mesures basées sur les arêtes de graphes. Celles basées sur des objets représentatifs consistent à minimiser la distance/dissimilarité totale entre les objets du cluster et un objet représentatif du cluster (centroïde, médoïde, etc.). Dans ce cas, généralement, le nombre de points représentatifs est identique au nombre de clusters. Les méthodes de clustering qui optimisent un tel objectif sur la base d'une telle mesure sont limitées à des clusters de formes sphériques, c'est-à-dire les clusters de formes arbitraires (allongées, spirales, etc.) ne peuvent pas être obtenus par ces méthodes (Guha et al., 1998; Ester et al., 1996). Les mesures basées sur les arêtes de graphes consistent à minimiser la somme des distances entre les paires d'objets de données dans un cluster ou la longueur maximale des arêtes dans un graphe connecté représentant le cluster. Ces mesures sont plus adéquates pour le traitement des formes arbitraires, par rapport aux mesures basées sur des objets représentatifs.

L'objectif de séparation correspond à la dissimilarité inter-clusters qui quantifie l'occurrence d'un changement de densité entre les clusters. Différents mesures de séparation existent dans la littérature, telles que : les mesures de liaison (single-link, average-link, et complete-link) et la distance entre les points représentatifs des clusters. Single-link calcule la distance entre une paire de clusters comme étant la distance entre les objets les plus proches (les plus similaires) entre eux alors que complete-link calcule la distance entre les points les plus éloignés (les plus différents) entre une paire de clusters. Dans average-link la distance moyenne entre des paires d'objets dans les deux clusters est utilisée.

L'évaluation de la compacité et de la séparation seulement est insuffisante dans les ensembles de données présentant des variations de densité qui est définie comme le nombre de points de données dans le volume unitaire (Ester et al., 1996; Liu et al., 2008). Les deux exemples de la figure 1.1 illustre cette situation où la distance entre deux points quelconques dans la région supérieure du cluster 1 de l'exemple (a), qui est moins dense, est supérieure à la distance minimale entre deux points dans les clusters 1 et 2. En se basant seulement sur ces mesures de distances dans le cas de l'exemple (a), les clusters 1 et 2 sont fusionnés ou le cluster 1 est divisé en plusieurs clusters.



(a) **(b)**

Figure 1.1. Deux exemples de données avec une variation de densité intra- et inter-clusters.

L'objectif de connectivité consiste à mesurer le lien qui existe entre les objets de données qui sont proches selon une mesure de similarité. La mesure de connectivité se concentre sur

les relations de proximité entre les objets de données de sorte que les voisins soient affectés au même cluster. Par exemple, les auteurs dans (Handl & Knowles, 2007) ont calculé l'objectif de connectivité comme le nombre d'objets de données voisins placés dans le même cluster pour un nombre *k* de plus proches voisins. La définition du voisinage est cruciale dans la construction des clusters. L'utilisation de cet objectif seul peut avoir des lacunes, le fait que la définition d'un voisinage pur incluant des objets d'un même cluster attendu est difficile.

Nous pouvons dire qu'aucune mesure générique qui s'adapte à toutes les situations n'est disponible dans la littérature. Les mesures utilisées pour définir les objectifs à utiliser dans le clustering dépendent des caractéristiques de l'ensemble de données (types d'attributs, nombre et formes des clusters, etc.) et le champ d'application. Par exemple, lorsque le nombre de clusters dans l'ensemble de données augmente, l'objectif de compacité s'améliore (ou reste le même) alors que les objectifs de séparation et de connectivité s'aggravent (ou ne s'améliorent pas). Ainsi, il n'est pas possible d'obtenir un clustering de bonne qualité par optimisation de l'un de ces objectifs ou une combinaison de ceux-ci. Pour remédier à ce problème, le clustering multi-objectif prend en compte chaque objectif et assure des compromis entre les objectifs.

- **Clusters ayant des formes arbitraires**

Les clusters de données peuvent être de différentes tailles, formes et densités. La figure 1.2 ci-après montre des exemples de clusters de formes arbitraires. Les clusters de forme arbitraire existent, particulièrement, dans les ensembles de données géo-spatiales (dans les systèmes d'information géographique), la géologie et les sciences de la terre et la segmentation d'images (Ester et al., 1996; Jain et al., 1999).

La détection de clusters de forme arbitraire dépend à la fois de l'objectif de clustering et de la fonction de similarité utilisée. Par exemple, comme nous l'avons déjà mentionné, il n'est pas possible de découvrir le cluster 1 de l'exemple (a) de la figure 1.1 (un cluster allongé avec une variation de densité intra-cluster) en utilisant soit une mesure de compacité basée sur un point représentatif ou sur une arête. L'objectif de connectivité est nécessaire pour découvrir les clusters de formes arbitraires en plus des objectifs de compacité et de séparation.

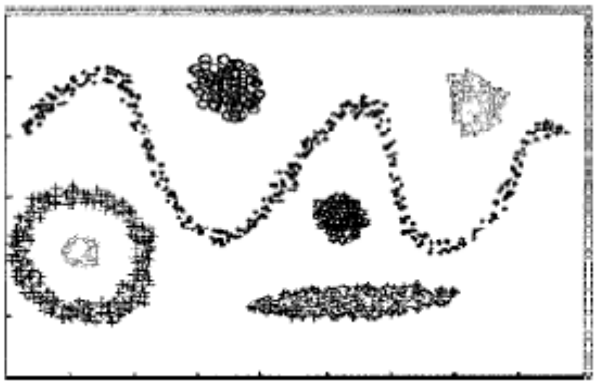


Figure 1.2. Exemples de clusters de formes arbitraires (He & Chen, 2003).

1.4. Les mesures de similarité

Le clustering consiste à grouper des objets similaires. Il est important donc de définir la notion de similarité (ou de façon opposée de dissimilarité) qui est utilisée dans la plupart des algorithmes de clustering. Dans la suite de cette section, nous visons d'abord à définir ces différentes notions (similarité, dissimilaritéou distance) ainsi que leurs différences. Par la suite, nous présentons brièvement quelques mesures de similarité ou de dissimilarité selon la nature de données selon qu'elles soient décrites par des attributs numériques ou symboliques (catégorielles). Plus de détails sur ces mesures sont disponibles dans (Xu & Wuncsh, 2005; Han et al., 2012).

La mesure de dissimilarité peut être soit calculée sur les attributs décrivant les objets, soit donnée directement. Dans cette thèse, nous nous limitons à l'étude des dissimilarités calculées à l'aide des attributs.

1.4.1. Concepts formels de base

Une mesure de similarité (*resp.* dissimilarité) est une application symétrique *s* (*resp. d*) de $D \times D$ dans $\mathbb{R}^+$ telle que $s(x_i, x_j)$ *(resp.* $d(x_i, x_j)$*)* est maximale (*resp.* minimale) et $s(x_i, x_j)$ *(resp.* $d(x_i, x_j)$*)* est d'autant plus élevée (*resp.* plus faible) que les descriptions (valeurs des attributs) des objets x_i, et x_j sont similaires.

La dissimilarité entre deux objets est souvent définie comme une distance. Nous rappelons qu'une distance vérifiée les propriétés suivantes :

- une distance est positive ou nulle : $d(x_i, x_j) \geq 0$,
- une distance est nulle si et seulement si les deux objets comparés sont identiques : $d(x_i, x_j) = 0 \Leftrightarrow x_i = x_j$,
- une distance est symétrique : $d(x_i, x_j) = d(x_j, x_i)$,
- une distance respecte l'inégalité triangulaire : $d(x_i, x_j) \leq d(x_i, x_k) + d(x_k, x_j)$.

Il convient de noter ici que le passage d'une distance à une mesure de similarité est trivial. Nous pouvons par exemple utiliser l'égalité suivante :
$\forall\, x_i, x_j \in D, s(x_i, x_j) = d_{max} - d(x_i, x_j)$, où d_{max} est la distance maximale séparant deux objets de *D*.

1.4.2. Mesures de similarité entre objets à description numérique

Parmi les mesures de distance les plus utilisées, nous avons les distances de Minkowski et la distance de cosinus qui supposent l'indépendance entre les attributs. Dans le cas de corrélation entre les attributs, alors cela reviendrait à donner plus de poids à certains attributs. La distance de Mahalanobis qui prend en compte la covariance des attributs (Duda et al., 2001) permet de corriger le problème d'indépendance entre attributs. La pondération des attributs lors de la préparation des données permet d'éviter la considération davantage des attributs dont les valeurs sont fortement dispersées.

- La distance de Minkowski est la plus connue et la plus utilisée dans le cas d'attributs numériques. Elle est définie par

$$d(x_i, x_j) = \left(\sum_{k=1}^{m} |x_{ik} - x_{jk}|^l \right)^{1/l}$$

Selon la valeur du paramètre l, nous appelons cette distance :

- distance de Manhattan, lorsque $l = 1$,
- distance Euclidienne, lorsque $l = 2$, ou
- distance de Tchebychev, lorsque $l = \infty$.

- La distance du cosinus est une autre distance, particulièrement très utilisée dans le cas de données textuelles. Cette distance correspond au cosinus de l'angle θ formé par les deux vecteurs x_i et x_j. Elle est définie par

$$d(x_i, x_j) = \cos(\theta) = \frac{\langle x_i \,.\, x_j \rangle}{\|x_i\| \|x_j\|},$$

où $\langle . \rangle$ désigne le produit scalaire, et $\|x_i\|$ désigne la norme de x_i : $\|x_i\| = \sqrt{\sum_k x_{ik}^2}$

- La distance de Mahalanobis permet d'éviter l'hypothèse d'indépendance entre attributs. Elle est définie comme suit :

$$d(x_i, x_j) = \sqrt{(x_i - x_j)^T S^{-1} (x_i - x_j)}$$

Où S est la matrice de variance/covariance. Notons que dans le cas où $S = I$ (matrice identité), on obtient la distance Euclidienne.

1.4.3. Mesures de similarité entre objets à description symbolique

Les mesures de similarité décrites précédemment ne conviennent pas dans le cas où les objets sont décrits (de façon totale ou partielle) par des attributs symboliques (qualitatifs). Dans ce cas, pour évaluer la similarité entre deux objets décrits de façon catégorielle, certains indices tel que *Rand* (Rand, 1971) et *Jaccard* (Jaccard, 1912) notés respectivement *R* et *J* procèdent par remplacement de l'espace de description par des attributs (propriétés) binaires (les attributs numériques sont généralement traités par découpage en intervalles) puis un comptage des propriétés partagées ou non par les deux objets. Ces indices sont définis comme suit :

$$R(x_i, x_j) = \frac{a + b}{a + b + c + d}$$

$$J(x_i, x_j) = \frac{a}{a + c + d}$$

Où ,

a désigne le nombre de propriétés partagées par les deux objets,
b désigne le nombre de propriétés qui ne sont pas vérifiées par aucun des deux objets,
c désigne le nombre de propriétés vérifiées par l'objet x_i seulement,
d désigne le nombre de propriétés vérifiées par l'objet x_j seulement,

D'autres indices tels que les indices de Sorensen (Sorensen, 1948) ou de Sokal et Michener (Sokal & Michener, 1958) utilisent aussi ce type de comptage. Ces indices sont plus adéquats dans le cas d'attributs initialement binaires car la redéfinition des attributs (passage vers des attributs binaires) engendra une surreprésentation des attributs ayant de nombreuses modalités. Dans le cas où les objets sont décrits par des attributs de différents types : numériques et symboliques, binaires ou non, d'autres approches existent (Malerba et al., 2001). Par exemple, Martin et Moal (2001) ont proposé de définir un nouveau langage de description $\mathcal{L}$. La similarité entre deux objets x_i et $x_j$est évaluée comme suit :

$$s(x_i, x_j) = \frac{1}{\mathcal{L}} \sum_{p \in \mathcal{L}} \delta_p(x_i, x_j)$$

Où,
p correspond à un terme (propriété) du langage $\mathcal{L}$,

$$\delta_p(x_i, x_j) = \begin{cases} 0, & Si\ la\ propriété\ p\ est\ satisfaite\ par\ un\ seul\ objet\ parmi\ les\ deux. \\ 1, & Sinon. \end{cases}$$

Plusieurs types de propriétés du langage peuvent être définis pour cette mesure :

- Quand il s'agit d'attributs symboliques, les propriétés sont généralement de la forme $attribut_i = valeur_j$, tel que $valeur_j$ désigne une parmi les modalités possibles de l'attribut *i*.
- Pour des attributs numériques, les propriétés sont généralement de la forme $attribut_i = valeur_j$, $attribut_i \leq valeur_j$, $attribut_i \geq valeur_j$ ou $valeur_j \leq attribut_i \leq valeur_k$, tel que $valeur_j$ et $valeur_k$ appartiennent au domaine de définition de l'attribut *i*, ou choisis parmi les valeurs de cet attribut dans l'ensemble des d'objets.
- D'autres formes de propriétés plus compliquées du langage $\mathcal{L}$ peuvent êtres définies :
    - Une expression linéaire combinant plusieurs attributs comme par exemple : $c_0 + c_1 att_1 + \cdots + c_m att_m \geq 0$, où $att_1 \ \ldots \ att_m$ sont tous des attributs numériques.
    - Conjonction de propriétés comme par exemple : $attribut_i \leq valeur_j\ ET\ attribut_m \leq valeur_k$
    - Intégration de connaissances extérieures pour affiner la pertinence du langage $\mathcal{L}$, comme par exemple le terme : $couleur = rouge\ OU\ couleur = rose$ qui permet de mettre en commun des modalités assez proches.

Généralement, la distance euclidienne est la plus utilisée dans le cas où les objets sont décrits par des attributs numériques seulement, et la mesure de Martin et Moal dans les autres cas.

1.5. Conclusion

Dans ce chapitre, nous avons présenté le problème de clustering. Les différents concepts liés au clustering ont été abordés et plus particulièrement, les grandes difficultés posées par ce

problème ont été analysées. Pour compléter la présentation du contexte du clustering, nous présentons dans le chapitre 2, les principales méthodes de clustering, et dans le chapitre 3, les différentes techniques d'évaluation du clustering.

Chapitre 2

Les principales méthodes de clustering

Sommaire

2.1. Introduction

Dans la littérature, plusieurs méthodes de clustering ont été développées pour résoudre le problème de clustering. Dans ce chapitre, nous décrivons les principales méthodes existantes. Les propriétés et capacités de ces méthodes sont passées en revue et leurs différents points de vue et hypothèses sont discutées. En effet, les différentes études de synthèse classent les algorithmes de différentes manières, selon que l'on s'intéresse au résultat de clustering (hiérarchie ou partitionnement, clustering dur ou clustering flou), ou selon la méthode utilisée pour parvenir à ce résultat (utilisation de fonctions probabilistes, graphes, grilles, métaheuristiques, etc.). Par exemple, dans (Jain et al., 1999), les auteurs présentent trois catégories de méthodes de clustering, à savoir : hiérarchiques, par partitionnement et les approches probabilistes, alors que Berkhin (2002) a considéré seulement les approches hiérarchiques et par partitionnement, en incluant les méthodes probabilistes dans cette dernière catégorie. Ceci est dû au fait que les catégories de méthodes se recouvrent (le résultat de certaines méthodes utilisant des modèles probabilistes est un partitionnement). Dans (Fung, 2001), une classification qui sépare les méthodes de clustering paramétriques des méthodes non paramétriques a été proposée. Une catégorisation proche de celle donnée dans

(Jain et al., 1999) a été présentée dans (Cleuziou, 2004; Rokach, 2010), en séparant les méthodes hiérarchiques, par partitionnement, approches basées sur : des fonctions probabilistes, des découpages en grilles, les densités et des descriptions conceptuelles.

Il est difficile de proposer une catégorisation logique des méthodes de clustering parce que la majorité des approches de clustering font appel à la fois aux différentes méthodes rencontrées dans la littérature. Les différentes catégories peuvent se chevaucher de sorte qu'une méthode peut avoir les caractéristiques de plusieurs catégories. Néanmoins, il est utile de présenter une image relativement organisée des méthodes de clustering. En général, la majorité des méthodes de clustering peuvent être classées selon les deux grandes catégories : hiérarchiques et par partitionnement (Barbakh et al., 2009; Xu & Wunsch, 2005; Xu & Wunsch, 2009; Jain et al., 1999). La figure 2.1 ci-après, montre les différents algorithmes de clustering selon une classification inspirée de celle donnée par Xu et Wunsch (2009). Nous présentons dans ce qui suit les deux catégories, à travers différents types d'approches, tout en montrant les avantages et les inconvénients de chacune.

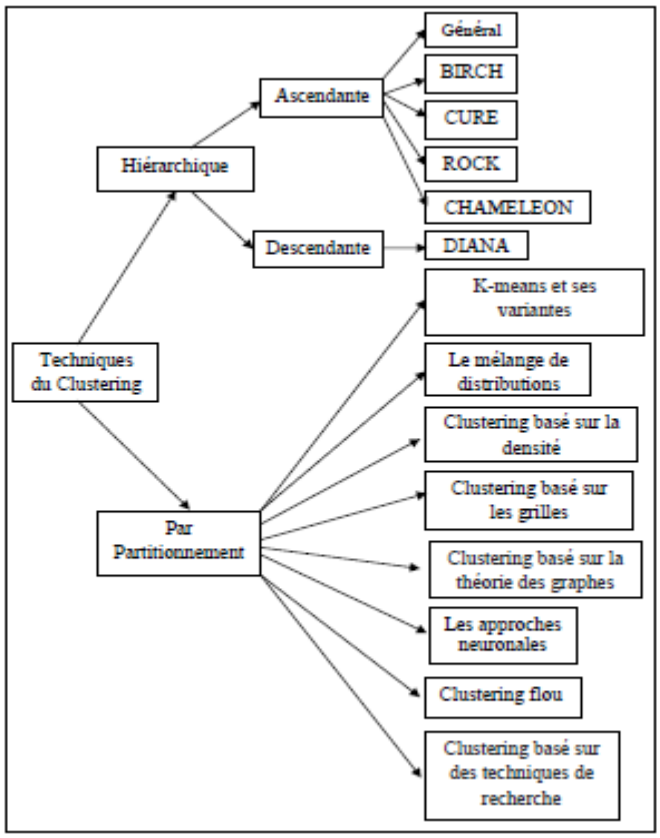


Figure 2.1. Les techniques de clustering (Xu & Wunsch, 2009)

2.2. Le clustering hiérarchique

Les méthodes hiérarchiques permettent une visualisation de l'organisation des données et du processus de clustering par la création d'une décomposition hiérarchique de l'ensemble des données. Elles permettent ainsi de construire un arbre de clusters (dendrogramme), c.-à-d. que le résultat de ces méthodes, comme le montre la figure 2.2 ci-après, est un ensemble de partitions et non seulement une partition unique des objets. Il est possible d'avoir une partition des données, en coupant le dendrogramme au niveau désiré. Par exemple le niveau 4

de l'arbre de la figure 2.2 retourne la partition $C = \{\{x_1, x_2, x_3\}, \{x_4, x_5, x_6\}\}$. Généralement, le niveau de coupure de l'arbre est choisi selon le nombre de clusters attendu ou en faisant une analyse de la qualité des différentes partitions de l'arbre (le processus de construction de l'arbre hiérarchique peut être arrêté lorsque le nombre de clusters voulu est atteint ou lorsqu'un seuil de qualité est atteint).

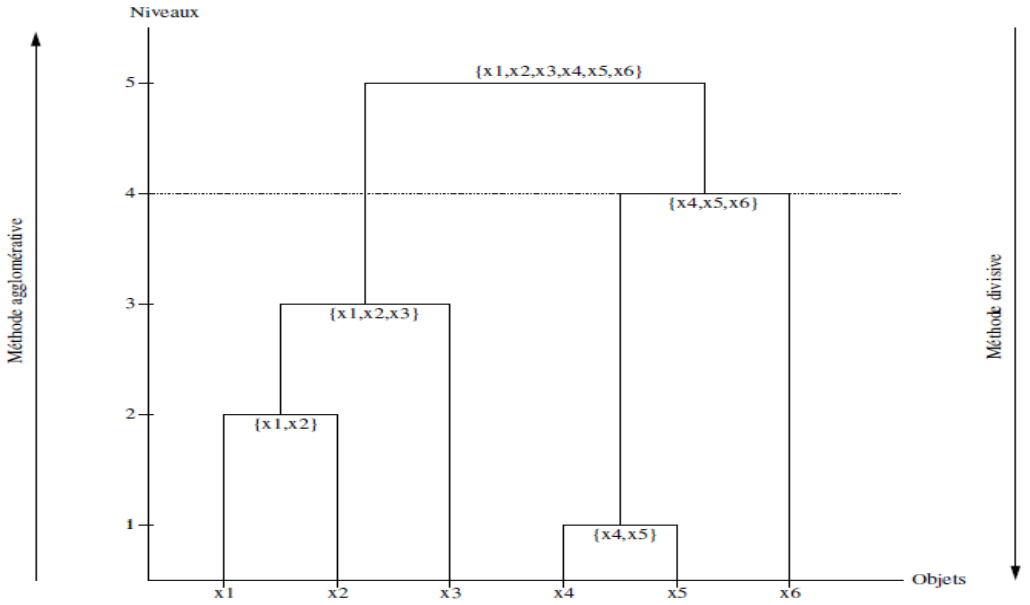


Figure 2.2. Exemple de dendrogramme représentant le clustering hiérarchique des objets de données $\{x_1, x_2,\ x_3,\ x_4, x_5, x_6\}$.

Deux sortes de ces méthodes peuvent être considérées, selon la forme de décomposition : les méthodes ascendantes, dites agglomératives et descendantes, dites divisives. Les méthodes ascendantes construisent l'arbre du bas vers le haut, en commençant à partir des clusters contenant un seul point (singleton) et fusionnent successivement les clusters les plus proches jusqu'à l'obtention d'un seul cluster (racine) ou jusqu'à ce qu'une condition de terminaison soit respectée. D'une façon inverse, les méthodes divisives construisent l'arbre du haut vers le bas en démarrant à partir d'un seul cluster contenant tous les objets et fractionnent successivement le cluster dont les objets sont plus éloignés, jusqu'à l'obtention des singletons ou jusqu'à ce qu'une condition de terminaison soit respectée. Les premiers algorithmes ascendants proposés sont basés sur l'algorithme SAHN (Sequential Agglomerative Hierarchical and Non-overlapping) (Sneath & Sokal, 1973), à savoir : SLINK (Single-LINK), CLINK (Complete-LINK) et ALINK (Average-LINK). Contrairement à ces trois algorithmes qui sont sensibles aux clusters de formes arbitraires et à la présence d'outliers, des algorithmes agglomératifs plus récents tels que BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) (Zhang et al., 1996), CURE (Clustering Using Representatives) (Guha et al., 1998), CHAMELEON (Karypis et al., 1999) et ROCK (RObust Clustering using linKs) (Guha et al., 2000) ne sont plus sensibles. Néanmoins, l'efficacité de ces approches dépend fortement du choix (qui est assez difficile) de l'échantillon représentatif dans le cas de CURE et ROCK et des paramètres de contrôle dans le cas de BIRCH et CHAMELEON.
Il existe beaucoup plus de méthodes ascendantes et elles sont plus utilisées par rapport aux méthodes descendantes vu la complexité très élevée de ces dernières. Dans le cas de fusion de

deux clusters parmi n, on a $\frac{n(n-1)}{2}$ possibilités, par contre dans le cas de division d'un cluster de n objets en deux sous-clusters on a $2^{n-1}-1$ possibilités. Pour éviter l'exploration de toutes les possibilités de division, des techniques ont été envisagées et utilisées dans certains algorithmes divisifs tels que : DIANA (DIvisive ANAlysis) (Kaufman & Rousseeuw, 1990) et PDDP (Principal Direction Divisive Partitioning) (Boley, 1998). Les méthodes hiérarchiques nécessitent donc des mécanismes supplémentaires pour prendre une décision sur le nombre de clusters. De plus, les décisions de fusion/division ne peuvent pas être changées durant les itérations. Ainsi, elles pourraient proposer des solutions sous-optimales (Xu & Wunsch, 2005).

2.3. Le clustering par partitionnement

Les algorithmes de clustering par partitionnement consistent à créer en sortie une seule partition de données, de tel sorte que les objets d'un même cluster soient similaires et ceux de différents clusters soient dissimilaires, contrairement aux approches hiérarchiques qui créent un arbre de partitions. Ces méthodes essayent de découvrir les clusters soit par réallocation itérative des objets entre les clusters, soit par identification direct des clusters comme étant zones très denses. Les algorithmes du premier type sont étudiés dans la sous-section « Les algorithmes de clustering par réallocation des objets autour de centres mobiles ». Les algorithmes de clustering par partitionnement du deuxième type sont étudiés dans la sous-section « Les algorithmes de clustering basés sur la densité ». Ces algorithmes essayent de séparer les régions de données de haute densité des régions de faible densité. Récemment, certains travaux ont exploité davantage différentes métaheuristiques et les ont utilisées pour le clustering. Ces méthodes sont étudiées dans la sous-section « Approches de clustering utilisant les métaheuristiques ». Dans la sous-section « Quelques autres approches de clustering », nous citons d'autres approches de clustering utilisant différents concepts.

2.3.1. Les algorithmes de clustering par réallocation des objets autour de centres mobiles (*k*-means et ses variantes)

Etant donné *n* objets, les algorithmes de ce type permettent de générer *k* clusters ($k \leq n$), tel que chaque cluster doit contenir au moins un objet et chaque objet appartient exactement à un seul cluster. De tels algorithmes créent un partitionnement initial choisi plus au moins aléatoirement. Par la suite, ils utilisent une technique de réallocation itérative permettant de déplacer les objets d'un cluster à un autre autour de centres mobiles. Ils consistent à comparer plusieurs partitions afin de faire sortir la partition qui optimise une fonction caractéristique (un critère de qualité). La partition optimale par rapport à une fonction objectif donnée peut être obtenue en énumérant toutes les partitions possibles. Vu la complexité très élevée d'une énumération exhaustive de toutes les partitions possibles, soit :
$P(n,k) = \frac{1}{k!}\sum_{i=1}^{k}(-1)^{k-i}\binom{i}{k}\, i^n$ partitions possibles, ces algorithmes cherchent des solutions approchées (la partition obtenue correspond le plus souvent à un optimum local, pour une fonction objectif).

Des algorithmes tels que : *k*-means, *k*-medoids, CLARANS, EM font partie de ce type d'algorithmes. L'algorithme le plus classique et le plus connu qui fait partie de cette catégorie

étant l'algorithme des k-moyennes (*k*-means) (Hartigan & Wong, 1979; Hartigan, 1975; MacQueen, 1967). Ce dernier est très utilisé dans divers domaines, à cause de sa simplicité et sa facilité de mise en œuvre. Etant donnée un paramètre *k* qui est le nombre de clusters à générer, cet algorithme consiste à générer une partition qui minimise la fonction de dissimilarité (basée sur la distance) définie comme E = $\sum_{i=1}^{k}\sum_{p\in Ci}|p - mi|^2$ où m_i est le centroide du cluster C_i. Il se résume par les étapes suivantes :

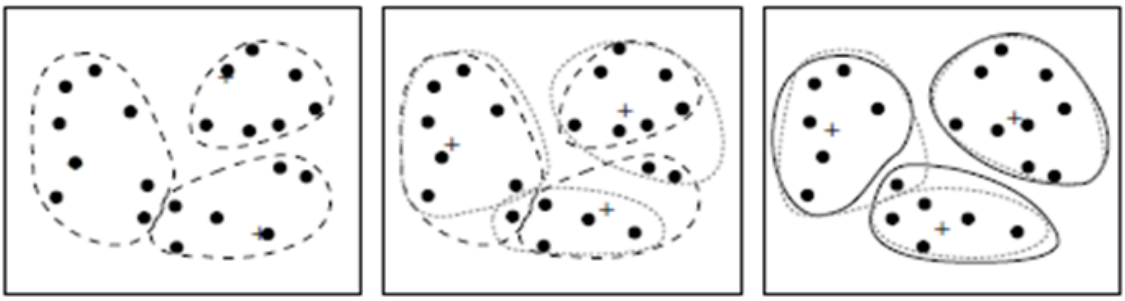
1. choisir aléatoirement *k* objets comme centroïdes des clusters,

Répéter

2. (Re)affecter chaque objet au centroïde le plus proche,
3. Recalculer le centroide pour chaque cluster comme moyenne des objets faisant partie du cluster,

Jusqu'à ce qu'aucun changement ne s'opère sur les clusters ou jusqu'à ce qu'un nombre d'itérations *t* soit atteint.

La procédure de l'algorithme k-means est illustrée dans la figure 2.3.



(a) clustering initial (b) itérations (c) clustering final

Figure 2.3. Clustering d'un ensemble d'objets par l'algorithme *k*-means. Pour (b) adaptation des centres des clusters et réaffectation des objets (la moyenne de chaque cluster est marquée par un +).

Bien que la complexité de cet algorithme est linéaire (*O(nkt)*, où $k \ll n \ll t$), l'utilisateur doit spécifier le nombre de clusters a priori. De plus, la partition obtenue correspond souvent à un optimum local qui dépend du choix de la partition initiale (il est nécessaire donc de faire un bon choix des centres initiaux). Aussi, l'application de *k*-means est limitée sur des objets décrits par des attributs numériques seulement, vu la définition de la moyenne. Un autre problème de cet algorithme est qu'il est sensible aux outliers, car un seul outlier peut influencer la valeur de la moyenne et par conséquent la répartition des objets. Les algorithmes de *k*-médoïdes (Han et al., 2009; Kaufman & Rousseeuw, 1990) permettent de régler certains problèmes de *k*-means, en prenant comme centres des clusters des objets existants réellement (médoïdes) au lieu des moyennes (centroïdes). Ils sont donc applicables à tous type de données et plus robustes aux outliers. Toutefois, les outliers doivent appartenir à une classe. Parmi les algorithmes des *k*-médoïdes, nous citons : PAM (Partition Around Medoids) (Diday, 1975), CLARA (Clustering Large Applications) (Kaufman & Rousseeuw, 1990) et CLARANS (Clustering Large Application RANdomized Search) (NG & HAN, 1994).

Ces algorithmes ont tendance à construire des clusters de formes convexes et ne donnent pas de bons résultats dans certaines situations (clusters de forme arbitraire, présence d'outliers, etc.).

2.3.2. Les algorithmes de clustering basés sur la densité

Les algorithmes de clustering à base de densité sont utilisés notamment pour le clustering de données spatiales qui peuvent présenter des clusters de formes arbitraires. Ils permettent de séparer les régions de haute densité considérées comme étant des ensembles d'objets voisins (clusters) des régions de faible densité. Généralement, les algorithmes de clustering à base de densité se basent sur des concepts tels que le voisinage d'un objet, l'objet noyau, l'accessibilité et la connectivité entre objets. L'idée de base de ces algorithmes est que dans chaque cluster, le voisinage de chaque objet doit contenir un nombre minimal d'objets. Par l'utilisation de la notion de voisinage, ces algorithmes cherchent à découvrir chaque cluster à partir d'un objet central (noyau) par lequel tous les objets de ce dernier sont accessibles (les voisins et leurs voisins). Donc, pour construire un cluster, il faut d'abord trouver un objet noyau, puis y ajouter tous les objets accessibles par ce noyau.

Parmi les algorithmes qui en découlent, nous citons : DBSCAN (Density-Based Spatial Clustering of Applications with Noise) (Ester et al., 1996), OPTICS (Ordering Points To Identify the Clustering Structure) (Ankerst, 1999), NBC (Neighborhood-Based Clustering) (Zhou et al., 2005) et les différentes variantes de DBSCAN, à savoir : GDBSCAN (Generalized DBSCAN) (Sander et al., 1998), VDBSCAN (Liu et al., 2007), DVBSCAN (Ram et al., 2010), DBCLASD (Distribution-Based Clustering of Large Spatial Databases) (Xu et al., 1998) et ST-DBSCAN (Birant & Kut, 2007).

Contrairement aux algorithmes de clustering par réallocation des objets autour de centres mobiles, les algorithmes de clustering à base de densité sont moins sensibles aux outliers et peuvent découvrir des clusters de formes arbitraires. Malgré leurs avantages, la performance de ces algorithmes est affectée par plusieurs paramètres à déterminer a priori, principalement, le nombre minimum de points pour définir le voisinage. Le réglage correct de ces paramètres est crucial et nécessite des efforts supplémentaires. En plus, les premiers travaux sur ces méthodes ont des limites dans la détection des clusters de densité variable. Généralement, ces algorithmes sont appliqués sur des données de faible dimension et d'attributs numériques, connu sous le nom de données spatiales.

Dans ce qui suit, nous présentons brièvement deux algorithmes qui font partie de ce type d'algorithmes, à savoir : DBSCAN (Ester et al., 1996) et NBC (Zhou et al., 2005).

2.3.2.1. L'algorithme DBSCAN (Density Based Spatial Clustering of Applications with Noise)

L'algorithme DBSCAN, proposé initialement par Ester et al. (1996), est destiné à la découverte de clusters de formes arbitraires et la détection d'outliers. Cet algorithme nécessite deux paramètres, à savoir : le rayon de voisinage *Eps* et le nombre minimum de points dans le voisinage (dans un cluster) *MinPts*.

DBSCAN construit tout d'abord un cluster à partir d'un objet noyau *p* sélectionné arbitrairement. Par la suite, ce cluster est étendu en ajoutant tous les autres objets qui sont *d*-accessibles à partir de *p* relativement à *Eps* et *MinPts*. Ensuite, de la même manière, DBSCAN recherche et ajoute tous les objets *d*-accessibles à partir de chaque objet noyau du cluster jusqu'à ce qu'il n'y ait plus d'objets à ajouter au cluster. Le processus de recherche est

répété récursivement, à partir du reste d'objets qui ne font partie d'aucun cluster déjà découvert, jusqu'à ce que tous les clusters soient découverts. Cela peut s'exprimer par le fait qu'il n'y a plus d'objets noyaux. Enfin, les objets qui ne peuvent pas être affectés à aucun cluster sont étiquetés comme des outliers.

Bien que DBSCAN permette de découvrir des clusters de formes arbitraires et convient aux grandes bases de données spatiales, il est difficile de trouver les bonnes valeurs de ses deux paramètres d'entrée. Généralement, l'utilisateur cherche à trouver une valeur approximative pour ces paramètres à l'aide du graphe *k*-dist (Ester et al., 1996). Cependant, comme la plupart des méthodes de clustering basées sur la densité, DBSCAN ne peut pas faire face à la difficulté de détection des clusters de différentes densités et ayant une variation de densité (des densités différentes qui existent au sein du même cluster) (Ertöz et al., 2003; Zhu et al., 2016). En plus, le nombre attendu de clusters ne peut pas être vérifié par l'utilisateur comme dans le cas de l'algorithme *k*-means.

2.3.2.2. L'algorithme NBC (Neighborhood-Based Clustering)

L'algorithme de clustering basé sur le voisinage NBC permet de découvrir des clusters de formes arbitraires et de différentes tailles et densités, en utilisant un seul paramètre d'entrée *k*. Il est basé sur le facteur de densité de voisinage NDF (Neighborhood Density Factor) qui permet de mesurer la densité locale d'un objet. L'idée de base de cet algorithme est que pour chaque objet *p* d'un cluster, le nombre d'objets dont le *k*-voisinage contient *p* ne doit pas être inférieur au nombre d'objets du *k*-voisinage de *p*. En se basant sur la valeur de NDF, un objet est considéré comme un objet noyau du cluster (objet dense) si la valeur de son NDF est supérieure ou égale à 1, ou un outlier si la valeur de son NDF est inférieure à 1.

Pour construire des clusters, NBC commence par le développement d'un cluster dénoté comme "cluster de *p*" à partir d'un objet noyau *p* (*NDF (p)* $\geq$ 1) sélectionné arbitrairement. Ensuite, il ajoute tous les autres objets qui se trouvent dans le *k*-voisinage de *p* *k*NB(*p*) relativement à *k* au cluster de *p*. Puis, de la même façon, il ajoute tous les objets qui sont dans le *k*-voisinage de chaque objets noyau dans le cluster de *p* jusqu'à ce qu'il n'y ait plus d'objets pouvant être ajoutés dans le cluster de *p*. Ce processus est répété récursivement à partir du reste de l'ensemble de données (objets n'appartenant à aucun cluster) jusqu'à ce que tous les clusters soient découverts (il ne reste aucun objet noyau non affecté). Enfin, les objets n'appartenant à aucun cluster sont marqués comme des outliers.

2.3.3. Approches de clustering par métaheuristiques

Les métaheuristiques sont des outils d'optimisation qui ont été utilisées dans plusieurs applications, y compris celles de l'exploration de données. L'objectif de ces méthodes est d'explorer de façon efficiente l'espace de recherche pour trouver des solutions proches de l'optimum pour des problèmes d'optimisation combinatoire (généralement, NP-Difficiles). Récemment, les métaheuristiques ont été bien adaptées pour résoudre le problème de clustering considéré NP-difficile (Banerjee, 2013; Kurada et al., 2013), notamment dans le cas de clusters de formes sphériques. Diverses métaheuristiques ont été appliquées aux problèmes de clustering (Hruschka et al. , 2009; Bagirov et al., 2020, Adil et al., 2020) : la recherche tabou (Sung & Jin, 2000), les algorithmes de recuit simulé (Sun et al., 1994), la méthode de

recherche par voisinages variables (VNS) (Mladenovic & Hansen, 1997), les algorithmes de colonie de fourmis (Jiang et al., 2010), les algorithmes génétiques (Agustín-Blas et al., 2012; Rahman & Islam, 2014; Raposo et al., 2014; Sabau, 2012), l'algorithme de colonie d'abeilles artificielles (Armano & Farmani, 2014; Ranjbar et al., 2015; Zhang et al., 2010) et l'algorithme d'optimisation par essaims de particules (Armano & Farmani, 2016; Cura, 2012 ; Das et al., 2008).

Dans ce qui suit, nous fournissons une revue littéraire de quelques travaux qui appartiennent à cette catégorie et nous discutons les difficultés, non encore résolues, rencontrées par ces derniers.

2.3.3.1. Travaux connexes sur le clustering à base de métaheuristiques

Les métaheuristiques à solution unique appliquées pour résoudre les problèmes de clustering incluent principalement, la recherche tabou, la recherche par voisinages variables et le recuit simulé. Ces méthodes sont plus connues sous le nom de méthodes de recherche locale ou méthodes de trajectoire. Elles démarrent à partir d'une première solution, ensuite au cours des itérations, la solution courante est déplacée dans un voisinage local afin de l'améliorer. Des algorithmes de recherche tabou, de recherche par voisinages variables et de recuit simulé adaptés au problème de clustering sont décrits dans (Sung & Jin, 2000; Al-Sultan, 1995; Al-Sultan & Fedjki, 1997) (Tsai & Chiu, 2006; Orlov et al., 2018) et (Klein & Dubes, 1989; Selim & Alsultan, 1991; Sun et al., 1994; Brown & Entail, 1992), respectivement.

Dans la suite de cette section, nous considérons les métaheuristiques à base de population où un ensemble de solutions initiales est utilisé au lieu d'une seule solution. Elles ont des capacités de recherches locales et globales et elles utilisent un haut niveau d'abstraction, leur permettant d'être adaptées à une large gamme de problèmes différents. Nous discutons les métaheuristiques qui sont fréquemment utilisées pour le clustering comme les algorithmes génétiques (AGs), l'algorithme d'optimisation par essaims de particules (OEP) et l'algorithme de colonie d'abeilles artificielles (ACB). Un aperçu des métaheuristiques appliquées au problème de clustering non revues dans cette section peut être trouvé, par exemple dans (Das et al., 2009; Gong et al., 2007; Handl & Knowles, 2004; Handl & Knowles, 2007; Hruschka, 2009; Rui & Wunsch, 2005; Suresh et al., 2009).

Les AGs ont été appliqués pour résoudre des problèmes de clustering sur des données de différents types, numériques et catégorielles (Rahila et al., 2008). Dans certaines approches de clustering par AGs, le nombre de clusters est utilisé comme paramètre d'entrée, alors que dans d'autres il est déterminé pour chaque solution au cours des itérations.

Pour éviter la sensibilité des AGs au nombre d'itérations, ainsi que la taille de la population, certaines versions des AGs hybrides ont été utilisées. Par exemple, dans (Babu & Murty, 1993) les AGs ont été utilisés pour générer de bons centres initiaux aux clusters, puis l'algorithme *k*-means est appliqué pour générer la partition finale. Une autre approche proposée dans (Rahman & Islam, 2014) consiste à appliquer les AGs pour trouver à la fois le nombre de clusters et les bons centres initiaux de clusters qui peuvent être utilisés ensuite par l'algorithme *k*-means. Kuo, et al., (2010) ont proposé un algorithme hybride basé sur les AGs

et PSO. Aussi, une hybridation des AGs avec une heuristique de recherche locale a été proposée par Chaves et Lorenab (2011).

Dans (Raposo et al., 2014), une approche utilisant un AG a été proposée pour éviter le problème des paramètres d'entrée de l'algorithme *k*-means. L'algorithme utilise un encodage de solution de longueur fixe égale à un nombre maximum de clusters fixé par l'utilisateur. Le nombre de clusters dans une solution peut être déterminé par des valeurs d'activation associées aux centroïdes. La fonction objectif considérée correspond à l'indice de Calinski-Harabasz (CH). L'algorithme proposé ne peut pas détecter les clusters de formes arbitraires et il est sensible aux outliers lors de l'affectation des objets aux centroïdes les plus proches. Rahman et Islam (2014) ont proposé un algorithme de clustering combinant un AG avec *k*-means, où chaque individu est une solution de clustering composée de deux composants ou plus et chaque composant représente un centre de cluster. L'algorithme détermine en premier lieu les meilleures valeurs des composants des individus (c'est-à-dire les centres) grâce à une approche de sélection déterministe. Ensuite, il utilise les opérateurs génétiques et l'algorithme *k*-means pour réajuster les centres initiaux. Cet algorithme utilise aussi une technique de réarrangement pour traiter le cas d'application de l'opération de croisement sur deux chromosomes ayant un nombre différent de gènes. Bien que cet algorithme puisse produire de bons résultats de clustering dans un petit nombre d'itérations que la sélection aléatoire de centres, il est sensible aux outliers et ne peut pas faire face à des ensembles de données de formes arbitraires. Généralement, la combinaison des AGs avec l'algorithme *k*-means évite le problème du minimum local issu de *k*-means et ne nécessite pas le nombre de clusters comme paramètre d'entrée (Hruschka et al., 2009).

Pour le traitement de clusters de formes arbitraires, dans le travail de Sabau (2012), l'algorithme DBSCAN a été combiné avec un AG. Des opérateurs de croisement et de mutation simples ont été utilisés avec un opérateur de réadaptation pour garantir des résultats valides. Bien que les résultats obtenus soient comparables aux techniques de clustering à base de densité existantes, la population initiale est générée de manière aléatoire, ce qui peut nécessiter plusieurs itérations pour obtenir une solution sous-optimale ou parfois de mauvaises solutions. En plus, le processus de réadaptation prend plus de temps pour changer et valider les individus générés après les opérations de croisement et de mutation.

Aussi, la majorité des approches de clustering basées sur PSO se concentrent sur les limites de l'algorithme *k*-means et ses variantes en exploration, et cherchent à se rapprocher le plus possible de l'optimum global. Afin d'augmenter l'efficacité de PSO, les approches proposées utilisent des algorithmes de recherche locale tels que *k*-means (Kao et al., 2008; Ahmadi et al., 2010). En 2009, un algorithme hybride combinant PSO et *k*-harmonic means a été proposé pour le clustering (Yang et al., 2009). Un autre algorithme hybride efficace combinant PSO, ACO et *k*-means a été proposé dans (Niknam & Amiri, 2010). Il existe plusieurs approches de clustering hybrides efficaces et robustes combinant PSO avec d'autres algorithmes évolutionnaires tels que, les algorithme génétiques (Kuo et al., 2012) et l'évolution différentielle (Xu et al., 2010). Dans (Tsai & Kao, 2011), un algorithme de clustering basé sur PSO combiné avec *k*-means a été proposé. Cet algorithme est basé sur la régénération sélective de particules. Dans (Chuang et al., 2011), des opérateurs chaotiques incorporant les propriétés du chaos ont été utilisés pour améliorer la convergence de PSO. Dans ces deux derniers algorithmes, les opérateurs de régénération des particules utilisées permettent une

exploration profonde de l'espace de recherche et assurent ainsi une meilleure convergence par rapport à *k*-means. Cura (2012) a proposé un algorithme basé sur PSO pour le clustering des données avec un nombre inconnu de clusters.

Dans la plupart des approches de clustering basées sur PSO, le nombre de clusters est introduit comme paramètre d'entrée et la fonction objectif est basée sur la distance (Omran et al., 2005; Omran et al., 2002; Jarboui et al., 2007; Kao et al., 2008; Ahmadi et al., 2010). Par conséquent, ces approches sont valables pour des ensembles de données avec des clusters de formes sphériques. Bien qu'il yait quelques approches qui déterminent le nombre de clusters dans l'ensemble de données, elles génèrent également des clusters de formes sphériques car la fonction objectif est une fonction de distance qui maximise la similarité intra-clusters (Veenhuis & Köppen, 2006; Cura, 2012).

Pour remédier au problème de la fonction objectif. Armano et Farmani (2016) ont considéré le clustering comme un problème multi-objectif. Un algorithme d'optimisation par essaim de particules multi-objectif a été utilisé pour le clustering automatique des ensembles de données avec des clusters de formes arbitraires. Deux objectifs utilisant les concepts de connectivité et de cohésion des données ont été définis par les auteurs. Les résultats de cet algorithme sont comparables avec d'autres approches de clustering conventionnelles sur quelques ensembles de données sélectionnés. Guan et al. (2019) ont conçu un algorithme d'optimisation par essaims de particules combiné avec l'algorithme DBSCAN pour corriger les inconvénients de DBSCAN. Cependant, cette approche ne peut pas traiter correctement les ensembles de données ayant des clusters avec des densités différentes.

L'algorithme de colonie d'abeilles artificielles est devenu l'un des paradigmes d'intelligence en essaim les plus intéressants, en raison de sa simplicité et de son nombre limité de paramètres de contrôle. Beaucoup d'algorithmes de clustering basés sur ABC ont été développés. Les auteurs de (Zou et al., 2010) ont proposé un ABC coopératif basé sur le principe de l'algorithme *k*-means, où chaque individu représente une solution avec un nombre de composants égal au nombre de clusters. Chaque composant d'un individu représente un centroïde du cluster et les solutions initiales sont générées de manière aléatoire. Yan et al. (2012) ont proposé un algorithme de colonie d'abeilles artificielles hybride pour le clustering. Une opération de croisement inspirée des AGs est incorporée dans ABC afin d'améliorer sa performance. Cet algorithme a montré une performance supérieure à celle de certains algorithmes tels que PSO, AGs, ABC et *k*-means. Dans (Gong et al., 2016), un ABC optimisé est proposé afin de surmonter le problème de l'optimum local issue de *k*-means (dépendant de l'état initial). Cet algorithme peut améliorer la sélection initiale des centres de clusters sur la base d'un ajustement dynamique. Cet ajustement est utilisé pour améliorer également l'optimisation locale. Dans (Danish et al., 2019), une forme modifiée de l'algorithme ABC standard, appelé GABCS a été appliquée pour le clustering de données. Plusieurs autres algorithmes de clustering basés sur ABC ont été proposés dans la littérature (Ji et al., 2015; Karaboga & Ozturk, 2011; Zhang et al., 2010; Gupta & Kumar, 2014; Kumar et al., 2017).

Bien que l'algorithme ABC soit simple et il a un nombre limité de paramètres, son application au problème de clustering reste limité au traitement du problème du minimum local issu de *k*-means. A notre connaissance, les différentes approches appliquant ABC pour le clustering ne peuvent pas traiter les ensembles de données ayant des clusters de formes arbitraires et de densités différentes. Elles génèrent également des clusters de formes

sphériques vu que la fonction objectif est une fonction de distance qui maximise la similarité intra-cluster.

2.3.3.2. Discussion sur les difficultés

Contrairement à la plupart des algorithmes de clustering traditionnels tels que *k*-means, les approches de clustering par méta-heuristiques effectuent une recherche globale dans l'espace de recherche. Cependant, ils peuvent rencontrer des difficultés dans la recherche de solutions proches de l'optimum, en particulier dans le cas de grands ensembles de données. Aussi, le codage de solution par les différentes approches existantes (Bhuyan et al., 1991; Maulik & Bandyopadhyay, 2000; Murthy & Chowdhury, 1996; Bezdek et al., 1994) peut poser des problèmes, en particulier pour le croisement dans le cas des AGs (une procédure de réadaptation est nécessaire pour assurer la validité des solutions après croisement) (Rahman & Islam, 2014 ; Sabau, 2012). Ces approches peuvent généralement identifier automatiquement le nombre de clusters (José-García & Gómez-Flores, 2016). Cependant, l'initialisation aléatoire des solutions conduit parfois à des solutions locales.

Bien que certaines approches puissent produire de bons résultats de clustering, elles sont sensibles aux outliers et ne peuvent pas faire face aux cas des clusters de formes arbitraires avec des densités variables. En fait, ces approches dépendent des paramètres définis par l'utilisateur et des objectifs de clustering qui sont difficiles à définir. En plus, pour l'évaluation des solutions de clustering, ces approches n'utilisent que les indices de validation de clustering conventionnels. Ces indices sont basés le plus souvent sur des mesures de distance et peuvent échouer, donc, dans le cas de clusters de formes arbitraires et de densité variable. En outre, ne considérer qu'un seul critère peut ne pas être conforme aux formes complexes des clusters. Pour les approches multi-objectif, afin de déterminer une seule solution optimale, des préférences subjectives supplémentaires (définies par un décideur) sont nécessaires. Ces préférences qui permettent de quantifier un compromis entre deux ou plusieurs objectifs conflictuels restent difficiles à définir.

Comme résumé, nous pouvons dire que la plupart des approches de clustering basées sur les métaheuristiques, qui existent dans la littérature, ne peuvent pas traiter les ensembles de données ayant des clusters de formes arbitraires et de différentes densités.

2.3.4. Quelques autres approches de clustering

Nous citons, dans cette section, d'autres approches de clustering adoptant d'autres concepts et approches.

2.3.4.1. Clustering basé sur les grilles

Les méthodes de clustering par grilles partitionnent les données à partir d'un partitionnement de l'espace de représentation de ces données en un ensemble de cellules (Xu & Wunsch, 2009). Ainsi, les clusters générés sont formés par des cellules connectées. La notion de densité est utilisée pour autoriser la construction de clusters moins proches, mais de densités homogènes, et ne pas se limiter à construire uniquement des clusters d'objets très proches. Les clusters sont générés en utilisant les informations de densité dans les grilles de manière hiérarchique. Ainsi, ces méthodes peuvent traiter des ensembles de données

constitués de clusters de formes arbitraires avec des variations de densité. Cependant, elles sont basées sur des hypothèses et un réglage des paramètres utilisateur est nécessaire. L'avantage principal de ces méthodes est le temps de traitement qui est indépendant du nombre d'objets (il dépend uniquement du nombre de cellules). Cependant, le problème qui se pose avec ces méthodes est leur exigene de construction de cellules d'une taille appropriée (problème de granularité). Parmi les algorithmes de clustering basés sur les grilles, nous citons : STING (STatistical INformation Grid-based method, Wang et al., 1997), WaveCluster (Wavelet-Based clustering, Sheikholeslami et al., 1998), GIZMO (Brézellec & Didier, 2001) et CLIQUE (CLustering In QUEst, Agrawal et al., 1998).

2.3.4.2. Clustering basé sur la théorie des graphes

Les méthodes de clustering basées sur les graphes utilisent différents formalismes de la théorie des graphes pour le partitionnement des données. Ces méthodes procèdent par une représentation des données par un graphe de proximité ayant comme sommets l'ensemble des objets et les arêtes reflètent la distance entre les objets (Schaeffer, 2007). Les principaux graphes de proximités sont : Le graphe du plus proche voisin NNG (Nearest Neighbor Graph (Paterson & Yao, 1992)), Arbre couvrant de poids minimal MST (Minimum Spanning Tree (Zahn, 1971)), Le graphe de voisinage relatif RNG (Relative Neighborhood Graph (Toussaint, 1980)), Le graphe de Gabriel GG (Gabriel Graph (Matula & Sokal, 1980)) et La triangulation de Delaunay DT (Delaunay Triangulation (Lee & Schachter, 1980)). Le partitionnement se fait par un découpage de l'ensemble des sommets selon les relations données par les arêtes. D'autres approches cherchent à partitionner le graphe en un nombre *k* fixe de clusters (groupes de sommets) de telle sorte que le nombre d'arêtes inter-clusters soit minimal. Pour éviter la complexité d'une telle approche qui augmente exponentiellement avec le paramètre *k*, la majorité des algorithmes procèdent par un partitionnement récursif en 2 (Pellegrini, 1994). Certaines d'autre approches cherchent à partitionner le graphe directement, sans fixer le nombre de clusters, de façon à obtenir le nombre optimal de clusters satisfaisant un critère donné (par exemple le diamètre maximum des clusters) (Brandenburg et al., 1999). Généralement, ces méthodes sont combinées avec d'autres algorithmes de clustering comme les algorithmes hiérarchiques et ceux basés sur la densité, tout en gardant la propriété principale de connectivité issue de la théorie des graphes. Cependant, la complexité temporelle concernant la construction des graphes est très élevée (par exemple, la construction de graphes de proximité tels que MST a une complexité temporelle de $O(n^2)$). L'algorithme le plus connu qui est basé sur la théorie des graphes est celui proposé par Zahn (1971). Cet algorithme consiste à générer des clusters à partir de la suppression des plus longues arêtes d'un arbre couvrant de poids minimal (MST).

2.3.4.3. Les approches neuronales

Les approches utilisant les réseaux de neurones artificiels sont très utilisées pour le clustering, en raison de l'aspect parallélisation du processus qui permet le traitement de grandes bases de données. Les principales limitations de ces approches sont le paramétrage, c'est-à-dire le nombre de clusters (nœuds dans la couche de sortie) et les paramètres de poids initiaux qui sont requis a priori. L'algorithme le plus connus est la carte auto-organisatrice de Kohonen SOM (Self-Organizing Map) (Kohonen, 1984).

2.3.4.4. Le clustering par mélange de distributions de probabilités

Les approches probabilistes supposent que les données sont tirées indépendamment à partir d'un modèle de mélange de plusieurs distributions de probabilités (un nombre *k* correspondant au nombre de clusters) (Mclachlan & Basford, 1988). Les distributions de probabilités sont dérivées de différents types de fonctions de densité (gaussiennes, multi-variées, etc.), ou en utilisant des paramètres différents pour un même type. La recherche de clusters revient donc à estimer les paramètres de plusieurs modèles (Xu & Wunsch, 2005). La performance de ces approches dépend de l'initialisation des paramètres et la procédure d'estimation pourrait donner des solutions sous-optimales. En plus, leur temps de calcul peut être assez élevé. L'algorithme EM (Expectation-Maximization, Dempster et al., 1977; Mclachlan & Krishnan, 1997) est le plus connu ayant utilisé l'approche générale d'estimation de paramètres par le maximum de vraisemblance (LM).

2.4. Conclusion

Dans ce chapitre, les différentes méthodes de clustering ont été étudiées et discutées en termes d'avantages et d'inconvénients.

D'après cette étude, il apparaît qu'il existe beaucoup de méthodes de clustering par partitionnement qui sont largement utilisées par rapport aux méthodes hiérarchiques. Ceci revient au fait que ces dernières ont une complexité quadratique dans la plupart des cas, alors que les méthodes de clustering par partitionnement ont, généralement, une complexité linéaire. Cependant, la majorité de ces méthodes nécessitent l'introduction de certains paramètres tels que, le nombre de clusters dans le cas de l'algorithme *k*-means ou le rayon de voisinage et le nombre de points dans le voisinage dans le cas de DBSCAN. Bien que divers algorithmes de clustering aient été proposés, la plupart d'entre eux ne peuvent pas traiter des clusters de formes arbitraires avec une densité variable. De plus, ils dépendent de quelques paramètres d'entrée qui sont difficiles à définir. Dans le cadre de cette thèse, nous avons essayé de résoudre ces problèmes en s'appuyant sur les métaheuristiques, pour éviter la difficulté issue de la définition des paramètres et des variations de densité dans les clusters, tout en profitant d'avantage des méthodes de clustering à base de densité, qui permettent le traitement des clusters de formes arbitraires. Aussi, vu les difficultés issues de la définition des objectifs de clustering et les lacunes des différentes techniques de validation, nous avons considéré le clustering comme un problème d'optimisation multi-objectif. Dans le chapitre 5, nous proposons des approches de clustering mono- et multi-objectif qui permettent de traiter le problème posé.

Chapitre 3

Techniques d'évaluation du clustering

Sommaire

3.1. Introduction

La validation des résultats obtenus par les algorithmes de clustering est une partie fondamentale du processus de clustering. L'évaluation des résultats du clustering permet de trouver le partitionnement qui correspond le mieux au jeu de données. Dans ce chapitre, nous discutons des concepts fondamentaux de la validité des clusters, et nous présentons les critères les plus importants dans le contexte de l'évaluation de la validité du clustering. Les différents indices de validation proposés dans la littérature sont discutés.

3.2. Spécification du problème d'évaluation du clustering

L'évaluation des résultats de clustering a un rôle important dans le processus de clustering. Elle permet de répondre à la question : qu'est-ce qu'un bon schéma (ou résultat) de clustering ? La réponse à cette question est une problématique qui a été synthétisée et étudiée dans plusieurs travaux (Halkidi et al., 2002; Halkidi & Vazirgiannis, 2001). En effet, le but des algorithmes de clustering est d'organiser un ensemble d'objets dans des groupes (clusters) similaires selon le contexte du problème étudié. En d'autres termes, il s'agit de maximiser la similarité dans les clusters et la dissimilarité entre les différents clusters. Cependant, l'exécution d'algorithmes de clustering différents génère des solutions (partitions) différentes, et aucun de ces algorithmes ne peut être considéré comme meilleur dans toutes les situations. Aussi, la plupart de ces algorithmes nécessitent l'intervention de l'utilisateur pour définir les valeurs des paramètres d'entrée. Par conséquent, l'application du même algorithme avec des valeurs de paramètres différentes génère des partitions différentes. Ainsi, l'utilisateur est confronté à des difficultés pour sélectionner l'algorithme de clustering le mieux adapté et fixer les valeurs des paramètres d'entrée pour une tâche particulière. Ce problème de choix de la meilleure partition peut être traité en utilisant des indices d'évaluation (de validation) de la

qualité du clustering. Par exemple, dans le cas de l'algorithme *k*-means (Hartigan & Wong, 1979), le nombre de clusters est introduit par l'utilisateur, ce qui signifie que les performances de cet algorithme sont très sensibles au nombre choisi. Ainsi, pour un processus de clustering efficace, nous devons exécuter l'algorithme plusieurs fois avec un nombre différent de clusters pour chaque exécution. Ensuite, nous devons sélectionner la meilleure partition parmi les partitions obtenues, en utilisant un indice de validité de clustering (Cluster Validity Index (CVI)). Le CVI est un indicateur de la qualité d'un algorithme de clustering pour une situation particulière et un indicateur des meilleures valeurs des paramètres d'entrée tel que le nombre correct de clusters dans le cas de l'algorithme *k*-means. Cependant, la notion de « bon » clustering est strictement liée au domaine d'application et à ses exigences spécifiques. Néanmoins, il est généralement admis que la réponse à la validité des résultats du clustering doit être recherchée dans des mesures de séparation entre les clusters et de cohésion au sein des clusters. Ces mesures sont utilisées pour définir les CVIs. Elles sont largement connues sous le nom de critères de validité de clusters. Pour définir ces mesures et évaluer les clusters, nous devons prendre en compte des aspects spécifiques de leur définition vu que les différents algorithmes de clustering se comportent différemment selon:

- Les caractéristiques de l'ensemble de données (géométrie et distribution de densité des clusters).
- Les valeurs des paramètres d'entrée.

L'évaluation du clustering est, donc, une tâche difficile.

3.3. Concepts fondamentaux de la validité des clusters

La procédure d'évaluation des résultats d'un algorithme de clustering est connue sous le terme « validité de clusters ». Généralement, il existe deux approches pour étudier la validité des clusters. La première est basée sur des critères externes qui permettent d'évaluer les résultats d'un algorithme de clustering sur la base d'une classification pré-spécifiée (structure imposée à un ensemble de données et reflète une intuition sur la structure de clustering de l'ensemble de données). La deuxième approche est basée sur des critères internes. Dans ce cas, l'évaluation d'un résultat de clustering se fait sur la base de certains critères à évaluer sur le résultat lui-même sans avoir besoin d'informations préalables. Deux critères sont proposés pour l'évaluation du clustering et la sélection d'un schéma de clustering optimal (Berry & Linoff, 1996) :

- La compacité intra-cluster : ce critère exprime le fait que les objets d'un même cluster doivent être similaires (proches) autant que possible l'un de l'autre. Une mesure usuelle de la compacité est la variance dont une faible valeur est un indicateur de proximité.
- La séparation inter-clusters : ce critère exprime le fait que les objets de différents clusters doivent être dissimilaires autant que possible, c.-à-d. les clusters doivent être largement espacés. Il existe trois approches usuelles pour mesurer la distance entre deux clusters différents. La première (Single linkage) mesure la distance entre les objets les plus proches des clusters. La deuxième (Complete linkage) mesure la distance entre les objets les plus éloignés. Quant à la troisième (Comparison of centroids), elle mesure la distance entre les centres des clusters. Cette dernière a été

largement utilisée dans la formalisation des indices de clustering par rapport aux deux autres en raison de leur inconvénient majeur qui est le coût de calcul élevé.

Les approches de validation basées sur des critères définis sur la base de mesures de distances sont efficaces dans le cas de clusters de formes sphériques (une bonne partition aura, donc, une petite distance intra-cluster et de grandes distances inter-clusters). Cependant, dans des cas où les clusters ont des formes arbitraires et des densités variables, la définition des critères de compacité et de séparation sur la base de mesures de distance seulement ne sera plus efficace. Dans ce cas là, ces mesures doivent évaluer la distribution de densité au sein et entre les clusters.

3.4. Analyse de quelques indices de validation du clustering

Les indices d'évaluation de la qualité du clustering (CVIs) sont classés en deux catégories : internes et externes (Halkidi et al., 2002; Deborah et al., 2010; Hancer & Karaboga, 2017; Wu et al., 2009).

3.4.1. Indices externes

Les indices externes tels que l'indice de Rand RI (Rand Index, Rand, 1971), Rand Ajusté ARI (Adjusted Rand Index, Hubert & Arabie, 1985), F-mesure (Rezaei & Fränti, 2016), Jaccard (Jaccard, 1912) et Variation de l'Information (VI, Arbelaitz et al., 2013; Batagelj & Bren, 1995), évaluent les résultats de clustering (les partitions obtenues) par leur comparaison avec une classification prédéfinie considérée comme correcte. Dans ce cas, une connaissance externe sur les données (partition attendue) est requise pour l'évaluation de l'adéquation des partitions obtenues. Ces indices sont généralement utilisés pour la comparaison et l'évaluation d'algorithmes mais ils n'ont pas d'application réelle, étant donné qu'aucune solution n'est pas a priori disponible pour le clustering qui est une tâche non supervisée.

- **L'indice de Rand** (Rand, 1971) : cet indice évalue les solutions de clustering en se basant sur une classification préexistante. Il prend une valeur dans l'intervalle [0, 1], où 0 indique que les deux partitions (la partition en question et la bonne partition) diffèrent sur toutes les paires d'objets et 1 indique que les deux partitions sont exactement les mêmes. Ainsi, par rapport à la partition correcte, une valeur plus élevée de RI signifie une meilleure solution de clustering.

Etant donné deux partitions C et C' de D à comparer, $C = \{C_1, C_2, \dots, C_k\}$ de k clusters et $C' = \{C_1', C_2', \dots C_k'\}$ de k' clusters, l'indice de Rand (RI) entre les partitions C et C' est défini comme :

$$RI(C, C') = (a + b)/(a + b + c + d)$$

Où

- a est le nombre de paires d'objets dans D qui sont dans le même cluster dans C et dans le même cluster dans C'.
- b est le nombre de paires d'objets dans D qui sont dans des clusters différents dans C et dans des clusters différents dans C'.
- c est le nombre de paires d'objets dans D qui sont dans le même cluster dans C et dans des clusters différents dans C'.

- d est le nombre de paires d'objets dans D qui sont dans des clusters différents dans C et dans le même cluster dans C'.

• **L'indice de Rand ajusté** (Hubert & Arabie, 1985) : l'indice de Rand ajusté (ARI) a été proposé par Hubert et Arabie en (1985) pour éviter que la valeur attendue de l'indice de Rand de deux partitions aléatoires ne prenne une valeur constante. Il suppose que les partitions C et C' à comparer sont sélectionnées au hasard de telle sorte que le nombre d'objets dans les clusters de C et les clusters de C' est fixe. Comme pour l'indice RI, par rapport à la partition correcte, une valeur plus élevée de ARI signifie une meilleure solution de clustering. ARI entre deux partitions C et C' est défini comme suit :

$$ARI(C, C') = (RI - E(RI))/(max(RI) - E(RI))$$

Où, $E(RI)$ est la valeur attendue de RI et $max(RI)$ est la valeur maximale de RI. Ainsi, ARI peut être simplifié comme suit :

$$ARI(C, C') = (a - 2(d + a)(c + a)/(|D|(|D| - 1)))/((d + c + 2a)/2 - 2(d + a)(c + a)/(|D|(|D| - 1)))$$

Où a, b, c et d sont les même variables utilisées pour définir l'indice de Rand.

• **F-measure** (Rezaei & Fränti, 2016) : la F-mesure (FM) adopte les idées de la précision et du rappel de la recherche documentaire (Rijsbergen, 1979). Elle est définie comme la moyenne harmonique de précision et de rappel. FM prend une valeur dans l'intervalle [0, 1]. Une valeur plus élevée de FM signifie une solution de clustering de meilleure qualité.

$$FM = 2 \times precision \times recall/(precision + recall)$$

Où,

$precision = a/(a + c)$ et $recall = a/(a + d)$

Par conséquent, FM peut être simplifiée comme suit :

$$FM = 2a/(2a + c + d)$$

Où a, c et d sont les même variables utilisées pour définir l'indice de Rand.

3.4.2. Indices internes

Les indices de validation internes évaluent les résultats de clustering (les partitions obtenues) sans aucune information préalable. Dans la réalité, aucune solution n'est a priori disponible pour le clustering qui est une tâche non supervisée. Par conséquent, la validation des résultats de clustering est basée seulement sur les indices internes (Liu et al., 2013; Zhou & Xu, 2018) comme : Dunn (Dunn, 1974), Davies–Bouldin (Davies & Bouldin, 1979), Calinski–Harabasz (Calinski & Harabasz, 1974).

Plusieurs indices de validation internes ont été proposés dans la littérature (Halkidi et al., 2002; Arbelaitz et al., 2013; Halkidi et al., 2001). Généralement, ces indices sont définis en utilisant les deux critères de compacité intra-cluster et de séparation inter-clusters qui sont considérés suffisants pour mesurer la qualité d'un clustering.

Les indices internes diffèrent seulement dans la formulation de ces deux concepts de compacité et de séparation. Les chercheurs ont proposé différentes variantes de CVIs conventionnels et une bonne revue peut être trouvée dans (Halkidi et al., 2002; Arbelaitz et al., 2013; Zhang et al., 2014; Yang et al., 2006). Dans ce qui suit, nous donnons la définition de quelques indices connus de cette catégorie tels que : Calinski-Harabasz (CH) (Calinski & Harabasz, 1974), Davies-Bouldin (DB) (Davies & Bouldin, 1979), Silhouette (Sil) (Rousseeuw, 1987) et Dunn (Dunn, 1974).

Soit $C = \{C_1, C_2, \dots, C_k\}$ une partition d'un ensemble de données D de N objets $D = \{x_1, x_2, \dots, x_N\}$ dans l'espace Euclidean R^m, où chaque point $x_i = (x_{i1}, x_{i2}, \dots, x_{im}), i = 1, N; \bigcup_{C_k \in C} C_k = D,\ C_k \cap C_l = \phi, \forall\, k \neq l$.

- Le centroide d'un cluster C_k est son vecteur moyen noté, $\overline{C_k} = \sum_{x_i \in C_k} x_i \,/\, |C_k|$, où $|C_k|$ est le nombre d'objets dans le cluster C_k.
- Le centroide de l'ensemble de données est son vecteur moyen noté, $\overline{D} = \sum_{x_i \in D} x_i / N$.
- La distance Euclidienne entre les points x_i et x_j notée $d(x_i, x_j)$, est définie comme :

$$d(x_i, x_j) = \|x_i - x_j\| = \sqrt{(x_{i1} - x_{j1})^2 + \cdots + (x_{im} - x_{jm})^2}$$

- La distance entre un point x et son $i^{ième}$ plus proche voisin est notée comme $KNN(x, i)$.

- **Silhouette (Sil)** (Rousseeuw, 1987) : cet indice est basé sur la cohésion, mesurée comme distance intra-cluster, et la séparation mesurée comme distance inter-clusters. Il prend une valeur dans l'intervalle [-1, +1], où une valeur plus élevée de cet indice signifie une partition de meilleure qualité. Il est défini comme :

$$sil(C) = \left[\sum_{i=1}^{k} \sum_{x \in C_i} \big(b(x) - a(x)\big)/max(a(x), b(x))\right]/N, \text{ où}$$

$$a(x) = \left(\sum_{y \in C_i, y \neq x} d(x, y)\right)/(|C_i| - 1) \quad \text{et} \quad b(x) = \min_{1 \leq j \leq k, j \neq i} \left(\sum_{y \in C_j} d(x, y)/|C_j|\right)$$

- **Calinski–Harabasz (CH)** (Calinski & Harabasz, 1974) : l'indice CH est égal au rapport entre la distance inter-clusters et la distance intra-cluster. La distance intra-cluster est mesurée comme la distance entre chaque point d'un cluster et son centroïde, tandis que la distance inter-clusters est mesurée comme la distance entre les centroïdes des clusters et le centroïde de l'ensemble de données. Une valeur plus élevée de l'indice CH indique une meilleure partition de données. Cet indice est défini comme suit :

$$CH(C) = \left[\sum_{l=1}^{k} |C_l| d(\overline{C_l}, \overline{D}) \Big/ (k-1)\right] \Big/ \left[\sum_{l=1}^{k} \sum_{x \in C_l} d(x, \overline{C_l}) \Big/ (N-k)\right]$$

- **L'indice Dunn (D)** (Dunn, 1974) : l'indice Dunn est un rapport entre la compacité et la séparation mesurées, respectivement, par la distance au voisin le plus proche et la largeur maximale du cluster. Une valeur plus élevée de cet indice signifie une meilleure partition de donnée. Il est décrit comme :

$$D(C) = \min_{1\le i\le k}\left[\min_{1\le j\le k, j\neq i}\left[\min_{x\in C_i, y\in C_j} d(x,y) \Big/ \max_{1\le l\le k}\left(\max_{x,y\in C_l} d(x,y)\right)\right]\right]$$

- **Davies–Bouldin (DB) (**Davies & Bouldin, 1979) : cet indice est un rapport entre la compacité et la séparation, où la compacité est mesurée par la distance de chaque point au centroïde de son cluster et la séparation est basée sur la distance entre les centroïdes des clusters. Une petite valeur de cet indice suggère une meilleure partition de données. Cet indice est défini comme :

$$DB(C) = \sum_{l=1}^{k} \max_{m=1,\dots,k, l\neq m} [(S(C_l) + S(C_m))/d(\overline{C_l}, \overline{C_m})]/k$$

Où $S(C_l) = \sum_{x\in C_l} d(x, \overline{C_l}) / |C_l|$

- **L'indice S_Dbw (S_Dbw)** (Halkidi & Vazirgiannis, 2001) : l'indice S_Dbw est basé sur la compacité des clusters (variance intra-cluster) et prend également en compte la densité entre les clusters (densité inter-clusters). Une valeur faible de cet indice suggère une meilleure partition de données. Généralement, S_Dbw est capable de sélectionner à la fois l'algorithme et les valeurs de ses paramètres pour un meilleur clustering. Cependant, il ne convient pas dans le cas de clusters de formes arbitraires et une densité variable (Halkidi et al., 2002). Cet indice est défini comme :

$$S\_Dbw(C) = Scat(C) + Dens\_bw(C)$$

Où, $Scat(C)$ est la variance intra-cluster qui mesure la diffusion (distribution) moyenne des clusters. Elle est calculée comme :

$$Scat(C) = \left(\sum_{l=1}^{k} \|\sigma(C_l)\| / \|\sigma(D)\|\right) \Big/ k$$

Où, $Dens\_bw(C)$ est la densité inter-clusters qui mesure la densité moyenne dans la région séparant (entre) les clusters par rapport à la densité des clusters. Elle est calculée comme :

$$Dens\_bw(C) = \left[\sum_{l=1}^{k}\sum_{m=1;m\neq l}^{k} dens(C_l, C_m) \Big/ \max\{dens(C_l), dens(C_m)\}\right] \Big/ k(k-1)$$

où, $dens(C_l) = \sum_{x\in C_l} f(x, \overline{C_l})$, $dens(C_l, C_m) = \sum_{x\in C_l\cup C_m} f(x, (\overline{C_l} + \overline{C_m})/2)$,

et $f(x, C_l) = \begin{cases} 0 & si\ d(x, \overline{C_l}) > stdev(C) \\ 1 & autrement \end{cases}$

Où, l'écart type d'une partition (stdev (C)) est défini comme : $stdev(C) = \sqrt{\sum_{l=1}^{k} \|\sigma(l)\|}\ /k$, et l'écart-type d'un ensemble d'objets D, $\sigma(D) = \sum_{x\in D}(x - \overline{D})^2/|D|$.

De plus, la norme euclidienne est définie comme : $\|x\| = (x^T x)^{1/2}$.

- **Validation de clustering basée sur la densité (Density-Based Clustering Validation (DBCV)) (**Moulavi et al., 2014) : cet indice a été proposé pour évaluer les clusters de forme arbitraire. Il est basé sur la densité de connectivité intra- et inter-clusters. Il prend une valeur dans l'intervalle [-1, +1], où une valeur élevée de DBCV indique une meilleure solution de

clustering. Il est basé sur la distance d'accessibilité mutuelle entre deux points x et y définie comme :

$$d_{mreach}\,(x,y) \;=\; max\,\{a_{pts}coredist(x), a_{pts}coredist(y), d(x,y)\}$$

La distance centrale d'un objet x appartenant au cluster C_i est donnée par :

$$a_{pts}cordist(x) = \left(\sum_{i=2}^{|C_i|}(1/KNN(x,i))^d \Big/ (|C_i|-1)\right)^{-1/d}$$

DBCV est défini comme : $DBCV(C) = \sum_{i=1}^{k}|C_i|V_C(C_i)/|D|$
où,

$$V_C(C_i) = \left[\min_{1\le j\le k, j\neq i}\Big(DSPC\big(C_i,C_j\big)\Big) - DSC(C_i)\right] \Big/ \max\left(\min_{1\le j\le k, j\neq i}\Big(DSPC\big(C_i,C_j\big)\Big), DSC(C_i)\right)$$

$DSPC(C_i, C_j)$ est la distance d'accessibilité minimale entre les nœuds internes des MST_{MRDs} des clusters C_i et C_j.

MST_{MRD} est l'arbre de poids minimum construit en utilisant $a_{pts}coredist$ et en considérant les objets de C_i.

$DSC(C_i)$ est le poids maximal des arrêtes internes dans MST_{MRD} du cluster C_i.

- **L'indice de validation basé sur le centre du cluster et le cluster voisin le plus proche (VCN)** (Zhou & Xu, 2018) : VCN est une simplification de l'indice Silhouette et il est mesuré par la distance de déviation intra- et inter-clusters normalisée. Une valeur plus élevée de cet indice implique une meilleure partition de données. Il est défini comme :

$$VCN(C) = \left[\sum_{m=1}^{k}(bd(m) - wd(m))/\max(bd(m), wd(m))\right] \Big/ k$$

où,

$$bd(m) = \min_{1\le l\le k, l\neq m}\left(\sum_{y\in C_m} d(y,\overline{C_l}\,)/|C_m|\right)$$

et $$wd(m) = (\textstyle\sum_{y\in C_m,} d(y,\overline{c_m}))/(|C_m|)$$

- **L'indice de validation de clustering basé sur la variance (VCVI)** (Zhu & Ma, 2018) : l'indice VCVI est basé sur la connaissance de la variance. Il est défini en se basant sur la distribution spatiale des ensembles de données en tenant compte du point de vue global (inter-clusters) et du point de vue local (intra-cluster). Une petite valeur de cet indice indique une meilleure partition de données. Il est défini comme :

$$VCVI = \left[k \times \sum_{l=1}^{k}(d(\overline{C_l},\overline{D}))^2 + \sum_{l=1}^{k}\sum_{x\in C_l}(d(x,\overline{C_l}))^2\right]/N$$

3.5. Travaux connexes sur les indices de validation internes de clustering

Nous avons parlé, dans la section précédante, des deux catégories d'indices de validation : externes et internes. Les indices de validation externes sont généralement utilisés pour l'évaluation et la comparaison d'algorithmes de clustering. Ils évaluent les résultats de clustering par une comparaison à une classification existante. Par contre, les indices internes mesurent la qualité des résultats en utilisant seulement l'ensemble de données.

Plusieurs indices internes de validation de clustering sont proposés dans la littérature (Halkidi et al., 2002; Arbelaitz et al., 2013; Halkidi et al., 2001). Généralement, ces mesures sont définies comme un rapport entre la compacité intra-cluster et la séparation inter-clusters et ne diffèrent que par la formulation de ces deux concepts de compacité et de séparation. Nous avons déjà présenté, dans la section 3.4.2, des indices bien connus de cette catégorie, tels que l'indice de Calinski-Harabasz (CH) (Calinski & Harabasz, 1974), l'indice de Davies-Bouldin (DB) (Davies & Bouldin, 1979), l'indice Silhouette (Sil) (Rousseeuw, 1987) et l'indice Dunn (Dunn, 1974). De plus, les chercheurs ont proposé différentes variantes de CVIs conventionnelles et une bonne revue peut être trouvée dans (Halkidi et al., 2002; Arbelaitz et al., 2013; Zhang et al., 2014; Yang et al., 2006). Cependant, la plupart de ces CVIs sont sensibles aux ensembles de données ayant des clusters de formes arbitraires, une densité variable et des outliers. Ceci est dû au fait que les concepts de compacité et de séparation sont basés uniquement sur la mesure de la distance et dépendent des centres de clusters. La figure 3.1 présente trois ensembles de données avec des clusters de formes arbitraires dans lesquels ces indices ne parviendront pas à capturer la structure des clusters.

Récemment, Shibing Zhou et Zhenyuan Xu (2018) ont proposé un nouvel indice de validation de clustering basé sur le centre du cluster et le cluster voisin le plus proche (VCN). Cet indice est une simplification de l'indice Silhouette qui prend en compte la déviation de distance intra- et inter-clusters. Comparé à Sil, VCN a une faible complexité temporelle ($O(n)$) (compétitif avec d'autres indices) et il surpasse Sil. Par conséquent, il peut être utilisé comme une variante plus rapide de Sil pour évaluer les résultats de clustering des ensembles de données avec des clusters de formes sphériques. Cependant, il est également basé sur les distances et dépend des centroïdes des clusters et des valeurs moyennes, ce qui le rend inapproprié pour les ensembles de données avec des clusters de différentes formes, différentes densités et incluant éventuellement des outliers.

Rojas-Thomas et al. (2017) ont proposé un nouvel indice interne utilisant des notions de graphes pour capturer les structures des clusters. Chaque cluster est partitionné en sous-clusters. La qualité à l'intérieur des clusters (compacité) et les distances entre eux (séparation) sont évaluées sur la base de l'arbre de poids minimal (MST) généré à l'aide des centroïdes des sous-clusters. La compacité d'un cluster est définie comme le degré de cohésion entre ses sous-clusters. Les résultats expérimentaux montrent que cet indice est efficace pour traiter certains ensembles de données avec des clusters de formes arbitraires, mais il ne peut pas traiter correctement les ensembles de données avec un nombre élevé de clusters tel que l'ensemble de données D31 présenté dans la figure 4.6.

Dans (Yang & Lee, 2004), un graphe de proximité est utilisé pour déterminer les bordures des clusters et ainsi décider si une partition est bonne ou pas. Cependant, cette approche dépend des paramètres d'entrée et elle est inappropriée pour la comparaison des partitions. Une autre approche de validation de clustering basée sur des notions de graphes est proposée dans (Pal & Biswas, 1997). Les auteurs ont défini des indices basés sur des informations extraites des arêtes des arbres de poids minimal représentant les solutions de clustering. L'utilisation des notions de graphes pour définir ces indices, permettrait de capturer la structure des clusters. Cependant, l'utilisation des distances euclidiennes et les centroïdes de clusters pour calculer la compacité et la séparation favorisent également les clusters de formes sphériques. Ces indices sont donc inappropriés pour des clusters de formes arbitraires et de tailles différentes tels que ceux de la figure 3.1.
Dans toutes ces approches (Rojas-Thomas et al., 2017; Yang & Lee, 2004; Pal & Biswas, 1997), les graphes sont générés en utilisant des distances et aucune d'entre elles n'a utilisé un concept de densité pour l'évaluation du clustering à base de densité. Pour résoudre ce problème, les chercheurs ont introduit la notion de densité dans la définition des indices. Par exemple, l'indice S_Dbw proposé dans (Halkidi & Vazirgiannis, 2001) est également basé sur la compacité à l'intérieur des clusters et la séparation entre les clusters en plus d'une mesure de densité. Cependant, le fait que le concept de densité soit basé seulement sur la variance et la distance entre les centroïdes des clusters, il ne peut pas traiter correctement les ensembles de données dans lesquels les centres ne sont pas nécessairement des points représentatifs de clusters de formes arbitraires comme l'ensemble de données en spirale présenté dans la figure 3.1. Pour faire face à ce problème, dans (Halkidi & Vazirgiannis, 2008), les auteurs ont proposé un nouvel indice de validation nommé CDbw. Cet indice est plus approprié pour les clusters de formes arbitraires, en tenant compte des points multi-représentatifs par cluster au lieu d'un centroïde. La distribution spatiale de ces points représentatifs peut capturer la forme arbitraire du cluster, ce qui laisse l'indice CDbw être l'une des mesures les plus adaptées pour la validation du clustering à base de densité. Bien que cet indice ait une meilleure performance dans le traitement efficace des clusters de formes arbitraires (en évaluant la distribution de densité à l'intérieur et entre les clusters) par rapport à d'autres indices, il a quelques limites et inconvénients liés à l'utilisation de points représentatifs multiples, notamment :

- L'utilisation du même nombre de points représentatifs pour tous les clusters est défavorable pour un ensemble de données avec des clusters de formes, densités et tailles différentes comme l'ensemble de données « compound » de la figure 3.1.
- La prédétermination du nombre de points représentatifs est un paramètre influençant significativement les performances de cet indice.
- Même si un nombre approprié de points représentatifs peut être défini, les points représentatifs eux-mêmes peuvent être déterminés par différentes approches et donc l'indice CDbw peut produire des évaluations différentes.
- L'indice CDbw peut être utilisé pour évaluer des ensembles de données avec des clusters bien séparés mais peut échouer dans le cas de clusters connectés par un pont ou des clusters avec une zone de haute densité entre eux (degré de chevauchement élevé entre les clusters), tels que les ensembles de données « compound » et « path-based » de la figure 3.1.

Dans (Žalik & Žalik, 2011), un CVI nommé OS a été proposé pour traiter les ensembles de données comprenant des clusters de densités et tailles différentes. Cet indice ne dépend pas des centroïdes des clusters et des valeurs moyennes, ce qui le rend plus efficace. Il est basé sur une mesure de séparation et une mesure de chevauchement. La mesure de séparation est calculée en utilisant tous les objets de données tandis que la mesure de chevauchement mesure le degré de chevauchement entre les clusters en utilisant quelques objets de données d'un cluster qui sont plus proches d'un ou plusieurs autres clusters. Bien que l'indice OS soit efficace pour l'évaluation des ensembles de données avec des clusters de formes arbitraires, son paramètre exprimant l'importance relative du chevauchement doit être ajusté de manière adaptative.

Les auteurs dans (Moulavi et al., 2014) ont proposé un indice de validation basé sur la densité (Density-Based Clustering Validation index (DBCV)) pour la validation de clusters de formes arbitraires. L'indice DBCV prend en compte la connectivité par densité à l'intérieur des clusters, mesurée comme la zone de densité la plus faible à l'intérieur des clusters et la connectivité par densité entre les clusters, mesurée comme la zone de plus haute densité entre les clusters. La connectivité par densité est basée sur le concept de distance centrale. Bien que cet indice prenne en compte les propriétés de densité et de forme, il ne parvient pas à traiter les ensembles de données avec des clusters qui ne sont pas bien séparés (ensembles de données contenant un pont entre les clusters), tels que les ensembles de données « compound » et « path-based » de la figure 3.1. En plus, le calcul de la connectivité par densité prend beaucoup de temps.

Pour traiter la sensibilité de la mesure de compacité dans le cas d'ensembles de données contenant des clusters de formes arbitraires et des outliers, les auteurs ont proposé dans (Lee et al., 2018) un nouveau CVI basé sur la description des données par un vecteur support (support vector data description (SVDD)). Le SVDD peut trouver des bordures de clusters de formes sphériques. Sur la base du concept SVDD, différentes variantes de CVIs ont été proposées. En fait, les CVIs proposés calculent la compacité des clusters en transformant les données originales en un espace de dimension supérieure. Bien que cette transformation soit efficace et peut produire une compacité précise pour les clusters de formes arbitraires, en particulier dans le cas de données de grande dimension, les variantes de CVIs basées sur SVDD ne peuvent pas traiter les cas de données avec des clusters ayant des formes complexes et des chevauchements importants, tels que les ensembles de données présentés dans la figure 3.1.

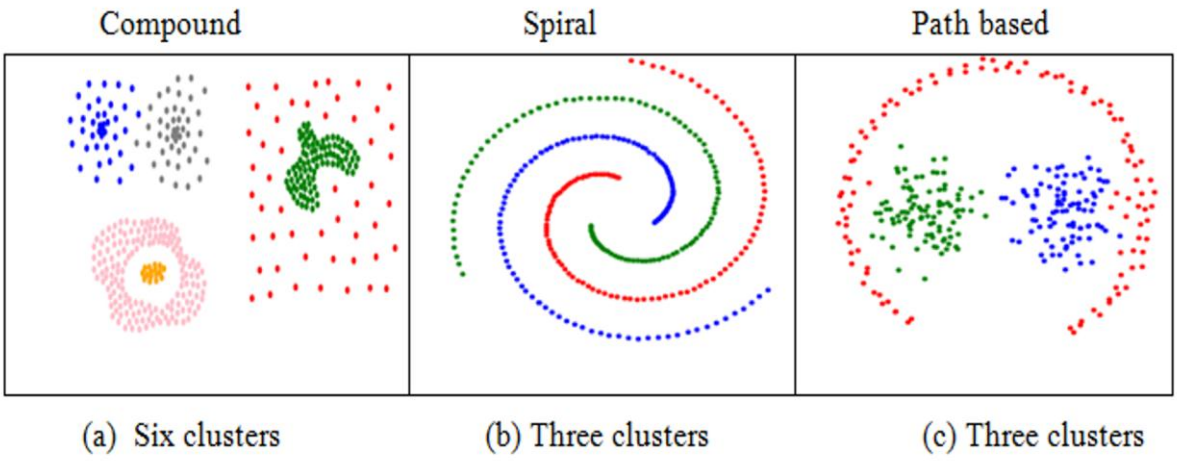


Figure 3.1. Exemples d'ensembles de données avec des clusters de formes arbitraires.

3.6. Conclusion

L'évaluation des résultats du clustering a une considération importante dans le domaine du clustering. Dans ce chapitre, nous avons présenté les concepts de base liées à l'évaluation des algorithmes de clustering. Nous avons, aussi, décrit et discuté en détails plusieurs indices de validation de clustering. La majorité de ces indices de validation sont basés sur la distance et sont valables, donc, pour la validation de clusters de formes sphériques. Bien que des indices basés sur la densité existent, ils présentent des lacunes et ils n'aboutissent pas à des résultats satisfaisants dans le cas des clusters de formes arbitraires et de densité variable.

Comme résumé, nous pouvons dire que la plupart des indices de validité de clustering sont basés sur la distance ou ne sont pas adéquats pour un ensemble de données particulier. Dans le chapitre 4 qui suit, nous proposons un indice de validation de clustering à base de densité qui permet de faire face au cas de clusters de formes arbitraires et de différentes densités.

Chapitre 4

Un indice de validation de clustering basé sur la connectivité et la densité

Sommaire

4.1. Introduction

Dans la littérature, il existe une grande variété d'algorithmes de clustering et chacun pourrait générer des résultats assez différents selon les paramètres d'entrée. En effet, l'utilisateur est toujours confronté à la difficulté de sélectionner le meilleur algorithme de clustering et les meilleures valeurs de ses paramètres d'entrée pour un ensemble de données donné. Ces problèmes peuvent être résolus sur la base d'une analyse de la validité du clustering (Sarma et al., 2013). Par exemple, la performance de l'algorithme *k*-means (Hartigan & Wong, 1979) est très sensible au nombre de clusters utilisé comme paramètre d'entrée. Ainsi, pour un processus de clustering efficace, nous devons exécuter l'algorithme plusieurs fois avec des nombres différents de clusters. Ensuite, la meilleure partition est sélectionnée, parmi les partitions obtenues, à l'aide d'un indice de validité de cluster (cluster validity index (CVI)). Un CVI est un indicateur de qualité d'un algorithme de clustering et de ses paramètres d'entrée (tels que le nombre correct de clusters dans le cas de l'algorithme *k*-means).

Plusieurs indices de validation de clustering ont été proposés pour évaluer les résultats du clustering et trouver la partition adéquate à l'ensemble de données d'entrée. Cependant, ces indices de validation peuvent ne pas atteindre des résultats satisfaisants, en particulier dans le cas de clusters de formes arbitraires. Dans ce chapitre, nous présentons notre contribution qui consiste à proposer un nouvel indice de validation de clustering à base de densité, pour les clusters de formes arbitraires. Nous présentons, tout d'abord, en détails le nouvel indice proposé qui est basé sur les relations de densité et de connectivité extraites sur la base du graphe de proximité de Gabriel. Par la suite, Nous discutons sa définition en termes de critères de cohésion et de séparation. Afin de montrer l'efficacité de l'indice proposé pour l'évaluation des algorithmes de clustering et la sélection de leurs paramètres appropriés, nous terminons le chapitre par une étude expérimentale sur des ensembles de données synthétiques et réels, en utilisant les algorithmes de clustering bien connus NBC et DBSCAN.

4.2. Description de l'indice de validation du clustering basé sur la connectivité et la densité proposé

Dans cette section, nous présentons notre indice de validation de clustering basé sur la connectivité et la densité (connectivity- and density-based cluster validity index (CDBCVI)), proposé particulièrement pour les ensembles de données avec des clusters de formes arbitraires et contenant des outliers. Il vise à évaluer différentes partitions (qui peuvent être obtenues par des algorithmes différents) d'un ensemble de données et à sélectionner la meilleure. Cet indice exploite les notions de connectivité directe et indirecte définies par Gabriel et Sokal dans le graphe de Gabriel (GG) (Gabriel & Sokal, 1969), pour mesurer la cohésion d'un cluster (en termes de densité et de connectivité intra-cluster) et la séparation d'un cluster (en termes de connectivité inter-clusters). L'idée principale du CDBCVI est l'extraction des relations de densité et de connectivité entre un ensemble de points en utilisant des concepts inspirés du GG.

L'utilisation d'une mesure basée sur la distance seulement peut produire une dissociation dans les voisinages et la fusion d'autres régions de densité. Par conséquent, elle ne peut pas évaluer correctement les clusters, en particulier pour les ensembles de données avec des clusters de formes arbitraires et une variation de densité. En fait, notre indice CDBCVI permet de déterminer et de vérifier si les voisins d'un point sont dans le même cluster, en utilisant simultanément des informations de proximité, de connectivité et de densité.

Des graphes de proximité comme l'arbre de poids minimal, triangulation de Delaunay et le graphe de Gabriel (Moulavi et al., 2014; Sung & Jin, 2000; Jiang et al., 2010; Armano & Farmani, 2014; Koontz et al., 1976; Liu et al., 2008; Inkaya et al., 2010) ont été appliqués et sont efficaces dans la définition du voisinage. Bien que le graphe de Gabriel génère des graphes complets, ses concepts sont exploités de manière adaptative pour déterminer les principaux voisins de chaque point.

Dans la suite de ce chapitre, nous utilisons les mêmes notations utilisées dans le chapitre 3, pour définir quelques indices de validation de clustering.

Nous introduisons, tout d'abord, certaines définitions liées au graphe de Gabriel, qui sont utilisées pour la description de l'indice CDBCVI.

Soit $BR(x,r)$ l'ensemble des points à l'intérieur de la balle centrée en un point *x* et de rayon *r*, c'est-à-dire $BR(x,r) = \{y | d(x,y) < r,\ y \neq x\}$.

Définition 4.1. Un GG d'un ensemble D de points exprime une notion de proximité de ces points. Formellement, c'est un graphe G avec un ensemble de sommets D et des arêtes (x,y), tels que : $BR(z,\ d(x,y)/2) \cap D = \phi$, où z est le point du milieu de la ligne reliant les points x et y. Cela signifie qu'une arête (x,y) est insérée dans le GG si et seulement si

$$d(x,y) \leq \min_i \left\{\sqrt{d(x,i)^2 + d(i,y)^2} \mid i \in D\right\}.$$

Soit $BD(x,y,d(x,y))$ l'ensemble des points à l'intérieur de la balle passant par les points x et y et de diamètre $d(x,y)$.

Définition 4.2. Deux points x et y sont directement connectés par une arête du GG, si et seulement si $BD(x,y,\ d(x,y)) \cap D = \phi$, ce qui signifie que :

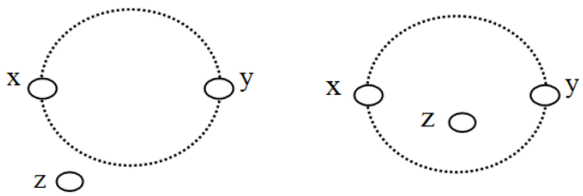
$$d(x,y) \leq Min_i \left\{\sqrt{d(x,i)^2 + d(i,y)^2} \mid i \in D\right\}$$

Définition 4.3. Deux points x et y sont indirectement connectés, si $BD(x,y,\ d(x,y)) \cap D \neq \phi$, c'est-à-dire que la balle passant par les points x et y et de diamètre $d(x,y)$ contient au moins un autre point de D.

Définition 4.4. La densité entre deux points x et y, notée $density(x,y)$, est égale au nombre de points dans la balle passant par les points x et y et de diamètre $d(x,y)$, c'est-à-dire,

$$density(x,y) = |BD(x,y,d(x,y)) \cap D|.$$

La figure 4.1 ci-dessous illustre les définitions des notions de connexion directe, connexion indirecte et de densité.



(a) (b)

Figure 4.1. Deux exemples de connexion directe et indirecte entre deux points. (a) x est directement connecté à y et $density(x,y) = 0$. (b) x est indirectement connecté à y et $density(x,y) = 1$.

Les principales contributions de notre approche de validation sont :

(1) L'utilisation de connexions directes pour déterminer les voisins noyaux (centraux) de chaque point et ainsi vérifier la compacité des clusters en termes de connectivité, c'est-à-dire qu'un cluster est plus compact s'il comprend plus de voisins centraux de ses points.

(2) L'utilisation de la densité des points, mesurée comme la distance entre eux et leurs voisins centraux, pour vérifier la compacité des clusters en termes de densité, c'est-à-dire qu'un cluster est plus compact si les densités de ses points sont plus proches.

(3) L'utilisation de connexions directes des points pour vérifier la séparation des clusters en termes de connectivité, c'est-à-dire que deux clusters sont bien séparés s'il existe plus de connexions directes entre leurs points qui ne sont pas des voisins centraux.

Afin d'évaluer chacune de ces mesures (compacité des clusters en termes de connectivité, compacité des clusters en termes de densité et séparation des clusters), nous déterminons les voisins noyaux de chaque point en utilisant des informations sur les connexions directes et indirectes et aussi la densité. Pour cela, pour chaque point *x*, nous formons un ensemble ordonné S_x composé de tous les points restants dans *D* classés par ordre croissant de leur distance au point x. En suivant les définitions 4.2, 4.3 et 4.4 données précédemment, nous pouvons déterminer le point le plus proche y ayant une connexion indirecte au point x. Donc, $d(x, y)$ est la distance minimale d'une connexion indirecte au point *x*.

L'ensemble des voisins noyaux du point x noté CN_x est l'ensemble des points directement connectés au point *x* avec une densité égale à 0 et une distance à *x* plus courte que *d (x, y)*, c'est-à-dire : $CN_x = \{z \in S_x | density(x, z) = 0 \; et \; d(x, z) < d(x, y)\}$, où y est le premier point ayant une connexion indirecte au point *x*.

Les connexions indirectes du point *x* à d'autres points peuvent être établies à partir de l'ensemble S_x, c'est-à-dire : $INDC_x = \{y | density(x, y) > 0\}$.

De même, les connexions directes du point *x* à d'autres points peuvent être établies à partir de l'ensemble S_x, c'est-à-dire : $DC_x = \{y | density(x, y) = 0\}$.

La figure 4.2 ci-après illustre un exemple de construction des ensembles *CN*, *DC* et *INDC*. Pour cet exemple, l'ensemble ordonné pour le point 1 est : S_1 = {2, 4, 3, 9, 10, 5, 6, 8, 7} avec des densités respectives au point 1 {0, 0, 2, 0, 2, 0, 1, 2, 3}.

Les points 2 et 4 sont connectés directement au point 1 (density(1,2) = density(1,4) = 0) et 3 est le premier point connecté indirectement au point 1 (density(1,3) ≠ 0).
Donc, CN_1 = {2, 4} (d(1,2) < d(1,3) et d(1,4) < d(1,3)).
$INDC_1 = \{3, 10, 6, 8, 7\}$ et $DC_1 = \{2, 4, 9, 5\}$.

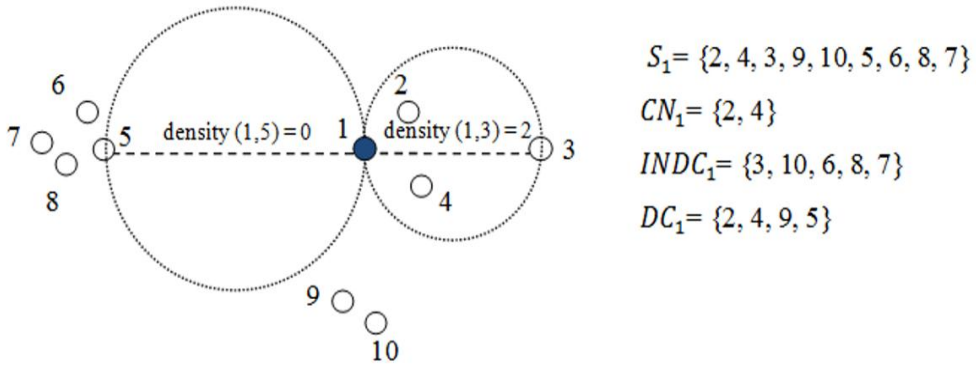


Figure 4.2. Exemple de construction des ensembles CN, DC et $INDC$ d'un point.

4.2.1. Compacité des clusters en termes de connectivité

Un bon partitionnement implique que les objets du même cluster sont bien connectés. Ainsi, les voisins noyaux d'un point doivent faire partie du même cluster que ce point. Nous exploitons cette notion de voisins noyaux pour évaluer la compacité des clusters en termes de connectivité.

Nous définissons la compacité d'une partition *C* en termes de connectivité des clusters par la formule suivante :

$$CompCon(C) = \sum_{i=1}^{k} \sum_{j=1}^{|C_i|} \left(\left|CN_j \cap C_i\right| / \left|CN_j\right|\right)$$

Selon cette équation, la compacité des clusters en termes de connectivité mesure la similarité au sein des clusters en testant si le voisinage noyau (central) d'un point fait partie du même cluster que le point lui-même. $CompCon(C) = |D|$, si tous les voisins noyaux de chaque point appartiennent au même cluster que le point lui-même, sinon $CompCon(C) \in [0, |D|]$.

4.2.2. Compacité des clusters en termes de densité

La compacité des clusters mesure la qualité intrinsèque des clusters. Plus cette mesure est grande, plus la densité des clusters est élevée. Cependant, la mesure de compacité précédemment définie, basée uniquement sur la connectivité, ne prend pas en compte l'homogénéité du voisinage au sein d'un cluster (c'est-à-dire que la densité au sein d'un voisinage ne devrait pas changer considérablement). Toutes les densités des points noyaux dans un cluster devraient être similaires. Par conséquent, les distances entre tous les points dans le même voisinage noyau de chaque point doivent être proches. Pour cette raison, nous définissons une mesure supplémentaire de compacité basée sur la densité de chaque point. La notion de densité prend en compte la variabilité des distances au sein des voisinages noyaux d'un cluster donné. Ainsi, la compacité en termes de densité doit être faible si des points provenant de différents voisinages sont inclus dans les mêmes clusters.
Nous évaluons la densité de chaque point *x* au sein d'un cluster C_i comme :

$$dens(x) = \sum_{j \in CN_x \cap C_i} d(j, x)$$

Ensuite, la définition de la compacité d'une partition *C* en termes de densité de clusters est donnée par la formule suivante :

$$CompDens(C) = \sum_{i=1}^{k} \left[\left[\sum_{j=1}^{|C_i|} \sum_{m=1, m \neq j}^{|C_i|} (min(dens(j), dens(m)) / max(dens(j), dens(m))) \right] \Big/ |C_i| \right]$$

Selon cette définition, la compacité en termes de densité des clusters mesure les changements de densité au sein des clusters. Ainsi, un bon partitionnement implique une valeur élevée de cette mesure, tandis que des valeurs faibles indiquent qu'il existe des zones de haute densité et des zones de faible densité au sein du même cluster.

4.2.3. Séparation des clusters en termes de connectivité

Dans notre travail, pour une meilleure évaluation de la séparation entre des clusters de formes arbitraires et de densités différentes, la séparation de chaque cluster est estimée en fonction de la densité des zones séparant celui-ci aux autres clusters. La densité en termes de nombre de points voisins noyaux provenant d'autres clusters est une indication de proximité des clusters. Ainsi, une bonne séparation d'un cluster est caractérisée par un nombre plus petit de voisins noyaux de ses points qui sont dans d'autres clusters. En fait, nous avons déjà utilisé cette caractéristique dans la définition de la compacité (une bonne compacité d'un cluster est caractérisée par l'existence de nombreux voisins noyaux de ses points dans le cluster). Ici, le terme "séparation" d'un cluster fait référence au nombre de points qui ne sont pas des voisins noyaux et sont directement connectés aux points du cluster considéré. Ainsi, un bon partitionnement se caractérise par une valeur élevée de la mesure de séparation des clusters. Notez que les connexions directes dans d'autres clusters qui ne sont pas des voisins noyaux des points sont des points de la bordure du cluster considéré. Par conséquent, l'utilisation de cette mesure peut capturer efficacement la forme des clusters.

Nous définissons la séparation de chaque cluster par l'équation :

$$Sep(C_l) = \sum_{i=1}^{|C_l|} \sum_{j=1, j\neq l}^{k} \left|C_j \cap (DC_i/CN_i)\right| \Big/ (|DC_i| - |CN_i|)$$

Ainsi, la séparation des clusters est définie par la formule suivante :

$$Separat(C) = \sum_{i=1}^{k} Sep(C_i)\ /k$$

4.2.4. Définition de l'indice CDBCVI

En utilisant les définitions de la compacité et de la séparation des clusters proposées ci-dessus, nous pouvons formuler un nouvel indice de validation d'une solution de clustering *C*, nommé CDBCVI, basé sur la connectivité et la densité, comme suit :

$$CDBCVI(C) = (Coh(C) + Separat(C))/(3|D|)$$

Où, la cohésion des clusters ($Coh(C)$) est définie comme la compacité en termes de connectivité au sein des clusters, avec prise en compte des changements de densité. Elle est définie comme suit :

$$Coh(C) = CompCon(C)\ +\ CompDens(C)$$

Il est facile de vérifier que notre indice proposé prend des valeurs dans l'intervalle [0, 1], où des valeurs élevées indiquent de meilleures solutions de clustering.

Notez que les outliers sont implicitement pris en compte dans notre formulation en tenant compte de la taille du cluster $|C_i|$ et le nombre total de points dans l'ensemble de données considéré (y compris les outliers) donné par $|D|$ dans la définition de l'indice CDBCVI.

4.2.5. Discussion

La définition de CDBCVI prend en compte tous les critères d'un bon clustering (c'est-à-dire la cohésion (compacité en termes de densité et de connectivité) et la séparation), permettant ainsi une évaluation fiable des résultats de clustering. Un résultat de clustering avec des clusters compactes et bien séparés sans aucune variation de densité ou avec une faible variation de densité au sein des clusters donne des valeurs élevées pour les deux termes composant l'indice CDBCVI (c'est-à-dire, cohésion et séparation).

Selon la définition du CDBCVI, les chevauchements de clusters, les différences de densité intra-cluster et inter-clusters peuvent avoir le plus grand impact sur sa performance. En fait, lorsque deux clusters ou plus ne sont pas bien séparés ou se chevauchent, l'indice CDBCVI peut se comporter de deux manières différentes d'un point de vue densité. D'une part, CDBCVI peut préférer la partition dans laquelle les clusters qui se chevauchent sont considérés comme un seul cluster. Cela est dû à la valeur élevée de la compacité en termes de connectivité au sein des clusters par rapport au changement de densité au sein des clusters et à la séparation, car la majorité des voisins noyaux des points sont dans le même cluster et la densité peut varier considérablement au sein du même cluster. D'un autre côté, CDBCVI peut préférer la partition dans laquelle les clusters sont séparés, en raison des valeurs élevées de séparation et de compacité en termes de densité au sein des clusters par rapport à la compacité en termes de connectivité. Cela est dû au fait que la majorité des voisins noyaux des points peuvent ne pas appartenir au même cluster et les clusters peuvent avoir une variation de densité faible. En outre, un bon résultat de clustering d'un ensemble de données avec des variations de densité intra-cluster et inter-clusters donne une faible valeur de compacité en termes de densité au sein des clusters. Selon la définition de CDBCVI, il peut échouer comme les autres CVIs, à produire de bons résultats pour des ensembles de données particuliers ayant des différences de densité à l'intérieur et entre les clusters (tels que les ensembles de données présentés dans la figure 4.3 ci-dessous) et les ensembles de données ayant des clusters qui ne sont pas bien séparés (comme l'ensemble de données présenté sur la figure 4.4).

Figure 4.3. Exemple d'ensembles de données ayant des clusters de différentes formes et des variations de densité intra-cluster et inter-clusters.

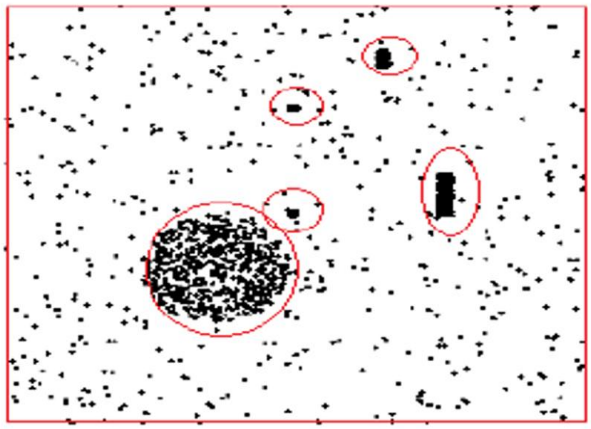


Figure 4.4. Un ensemble de données ayant 6 clusters de formes, tailles et densités différentes, les clusters internes sont beaucoup plus denses que celui externe.

4.3. Étude expérimentale

Dans cette section, nous présentons l'étude comparative menée, afin de tester la performance de notre indice CDBCVI proposé par rapport à des CVIs bien connus. Dans l'étude comparative, nous avons utilisé l'approche habituelle qui consiste à exécuter un algorithme de clustering sur un ensemble de données avec des valeurs différentes de ses paramètres d'entrée conduisant à différentes partitions. L'indice considéré (CVI) est calculé pour chaque partition et la partition qui donne la meilleure valeur du CVI est considérée comme la meilleure obtenue par ce CVI pour l'ensemble de données considéré. Si le nombre de clusters obtenu par un CVI (celui de la meilleure partition) est égal au nombre correct, le CVI est considéré comme efficace. De plus, puisque les CVIs sont utilisés pour déterminer la meilleure partition d'un ensemble de partitions au lieu d'estimer le nombre correct de clusters, nous avons également utilisé la méthodologie utilisée par Moulavi et al. (2014) qui tire pleinement avantage des informations externes. Cette méthodologie évalue la précision des CVIs par une comparaison de leurs résultats (les meilleures partitions trouvées) à ceux trouvés par un indice externe, tel que l'indice ARI. Le meilleur CVI est celui qui a sélectionné la partition la plus similaire à celle obtenue par un indice externe. Dans ce cas, la similarité est mesurée par la corrélation de Pearson.

Dans nos expériences, nous avons utilisé les indices de validation externes F-mesure (Rezaei & Fränti, 2016), RI (Rand, 1971) et ARI (Hubert & Arabie, 1985), qui peuvent sélectionner la meilleure partition définie comme la plus similaire à la partition correcte. Étant donné qu'une partition peut avoir un nombre correct de clusters, mais qu'elle peut présenter un cluster non naturel, la partition sélectionnée par l'indice externe n'est pas toujours celle ayant le vrai nombre de clusters.

4.3.1. Les algorithmes de clustering

Puisque notre objectif principal est la validation des résultats du clustering à base de densité, nous avons utilisé deux algorithmes de clustering bien connus, basés sur la densité, pour la génération de partitions à partir d'ensembles de données, à savoir : DBSCAN (Ester et

al., 1996) et NBC (Zhou et al., 2005). En modifiant les valeurs des paramètres d'entrée de chacun de ces algorithmes, son exécution peut conduire à différentes partitions. Nous avons, donc, utilisé l'algorithme NBC pour générer des partitions par une variation de son paramètre *k*, 20 fois dans l'intervalle [2, 30] (c.-à-d., $k \in \{2, 3, 4, 6, 8, 9, 10, 12, 14, 15, 16, 18, 20, 21, 22, 24, 26, 27, 28, 30\}$). Dans le cas de l'algorithme DBSCAN, nous avons choisi son paramètre *Minpts* dans l'ensemble {3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 15, 16, 18, 20, 21, 24, 25, 27, 28, 30} tandis que les 20 valeurs du paramètre *Eps* (des valeurs différentes) sont sélectionnées aléatoirement pour chaque ensemble de données entre les valeurs minimale et maximale des distances entre les paires d'objets. Nous avons, donc, considéré un scénario dans lequel l'utilisateur n'a aucune idée des valeurs des paramètres à choisir, c'est-à-dire un scénario dans lequel un CVI interne est utile.

4.3.2. Les indices de validation utilisés pour la comparaison

Dans l'étude comparative, nous avons utilisé plusieurs CVIs bien connus qui sont Sil, CH, Dunn, DB, S_Dbw, DBCV, VCN et VCVI. Ces CVIs ont été décrits dans la section 3.4.2 du chapitre 3. Pour chaque ensemble de données, chaque CVI a été calculé pour toutes les partitions générées par les algorithmes DBSCAN et NBC.

4.3.3. Les ensembles de données utilisés

Le fait que l'objectif principal de l'indice CDBCVI proposé est l'évaluation des partitions obtenues à partir d'ensembles de données ayant des clusters de formes arbitraires et de densité variable, nous avons utilisé différents ensembles de données réellement étiquetés : réels et synthétiques obtenus respectivement à partir du référentiel d'apprentissage automatique de l'UCI[1] et du référentiel de clustering de l'Unité de traitement de la parole et de l'image (Franti, 2015). Le tableau 4.1, ci-après, décrit les ensembles de données utilisés dans les expériences. Aussi, pour confirmer l'efficacité de l'indice CDBCVI proposé dans la validation des résultats de clustering à base de densité, nous montrons, dans la figure 4.5, les formes des clusters des ensembles de données bidimensionnels utilisés.

[1]UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/ (2018).

Tableau 4.1. Caractéristiques des ensembles de données utilisés dans les expériences.

Dataset	Nombre d'objets	Dimension	nombre Correct de clusters
Path-based	300	2	3
Spiral	312	2	3
Flame	240	2	2
Jain	373	2	2
Aggregation	788	2	7
Compound	399	2	6
R15	600	2	15
D 31	3100	2	31
Dim128	1024	128	16
Dim512	1024	512	16
Iris	150	4	3
Wine	178	13	3
Breast Tissue	106	9	6
Glass	214	9	6
Musk	476	166	2
Scadi	70	205	7

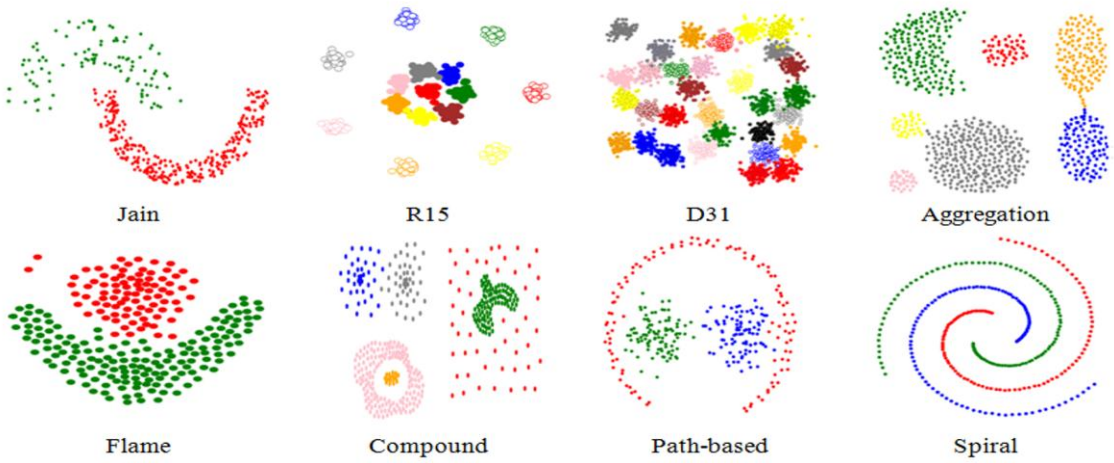


Figure 4.5. Exemples d'ensembles de données bidimensionnels (avec des clusters de formes arbitraires) utilisés dans les expériences.

4.3.4. Description du processus comparatif

Pour des ensembles de données dont la classification est connue (les objets sont étiquetés), le processus comparatif utilisé pour évaluer la performance de l'indice CDBCVI proposé se déroule en quatre étapes comme suit :

Étape 1 : Exécuter les algorithmes NBC et DBSCAN 20 fois chacun pour chaque ensemble de données, en faisant varier leurs paramètres ; calculer les CVIs ainsi que la mesure F, l'indice de Rand et l'indice de Rand ajusté pour chaque partition générée.

Étape 2 : Comparer le nombre optimal de clusters, sélectionné par chaque CVI, avec le nombre réel (correct) de clusters de chaque ensemble de données (le nombre optimal de clusters trouvé par un CVI est celui de la partition ayant la meilleure valeur du CVI). Les CVIs qui peuvent trouver le nombre correct de clusters sont plus efficaces.

Étape 3 : Puisqu'une partition peut avoir le nombre correct de clusters alors qu'elle peut présenter des clusters non naturels ; de plus la meilleure partition doit être la plus similaire à la bonne, mais pas nécessairement celle avec le vrai nombre de clusters, alors la comparaison basée uniquement sur le nombre optimal de clusters trouvé par les CVIs peut ne pas être efficace. Ainsi, les valeurs des indices FM, RI et ARI obtenues par chaque CVI sont comparées (les valeurs de FM, RI et ARI sont celles données par la partition ayant la meilleure valeur du CVI).

Étape 4 : Enfin, pour quantifier la précision des CVIs par rapport aux indices FM, RI et ARI, cette étape consiste à calculer la corrélation entre les vecteurs avec les 40 valeurs de chaque CVI et les 40 valeurs de chaque indice parmi FM, RI et ARI.
Pour chaque ensemble de données, le CVI qui fournit à la fois la meilleure valeur des indices FM, RI et ARI (près de 1), la meilleure corrélation avec les indices FM, RI et ARI et le nombre correct de clusters est le plus efficace.

4.3.5. Résultats et discussion

Les tableaux 4.2, 4.3 et 4.4, ci-après montrent respectivement, le nombre optimal de clusters sélectionné (obtenu) par chaque CVI, les valeurs de FM, RI et ARI de la meilleure partition sélectionnée par chaque CVI et la corrélation entre chaque CVI et les indices externes FM, RI et ARI. Les meilleurs résultats sont en gras.

Tableau 4.2. Le nombre optimal de clusters trouvé par chaque CVI pour seize ensembles de données.

Dataset	Nombre correct de clusters	Nombres de clusters générés par les algorithmes DBSCAN et NBC	Indice								
			CDBCVI	Sil	CH	Dunn	DB	S_Dbw	DBCV	VCN	VCVI
Path-based	3	2, 3, 4, 5, 6, 7, 12, 17, 26, 53	**3**	**3**	2	53	**3**	2	4	2	2
Spiral	3	2, 3, 4, 5, 6, 9, 11, 29	**3**	**3**	2	2	2	2	4	2	2
Flame	2	2, 3, 5, 7, 8, 10, 15, 33	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**
Jain	2	2, 4, 3, 5, 6, 7, 8, 9, 11, 12, 20, 28, 29, 30, 34, 69	**2**	**2**	6	6	6	12	**2**	**2**	6
Aggregation	7	3, 4, 5, 6, 7, 9, 13, 16, 32, 117	**7**	4	4	117	**7**	**7**	**7**	4	**7**
Compound	6	2, 3, 4, 5, 6, 7, 8, 10, 12, 17, 32, 60	7	2	2	5	5	5	2	2	5
R15	15	7, 8, 10, 11, 15, 18, 24, 71, 124	**15**	8	10	124	10	8	8	8	8
D 31	31	28, 29, 30, 34, 36, 66, 111	29	30	30	111	29	30	29	28	28
Dim128	16	4, 16, 18, 20, 30, 44, 46, 60, 73, 80, 92, 107, 132	**16**	**16**	18	107	**16**	107	**16**	4	**16**
Dim512	16	4, 16, 17, 32, 46, 50, 69, 78, 90, 107, 122, 143	**16**	**16**	**16**	90	**16**	107	**16**	4	32
Iris	3	2, 3, 5, 7, 14, 28,	2	2	2	28	2	2	2	2	**3**
Wine	3	2, 3, 4, 5, 6, 7, 9, 11, 15, 21, 34	**3**	2	2	21	5	11	**3**	2	**3**
Breast Tissue	6	3, 4, 5, 6, 8, 11, 14, 21,	3	5	3	21	5	5	11	3	3
Glass	6	2, 3, 4, 5, 7, 9, 13, 23, 41	2	3	2	41	2	2	4	2	2
Musk	2	2, 3, 5, 7, 14, 15, 23, 29, 55, 85	**2**	**2**	**2**	85	**2**	85	**2**	**2**	3
Scadi	7	2, 4, 5, 6, 10, 11	2	5	2	4	6	11	5	2	6

Tableau 4.3. Meilleures valeurs des indices FM, RI et ARI trouvées par neuf CVIs pour seize ensembles de données.

Dataset		Indice								
		CDBCVI	Sil	CH	Dunn	DB	S-Dbw	DBCV	VCN	VCVI
Path-based	ARI	**0.8597**	0.4507	0.662	0.0541	0.4355	0.662	0.3946	0.662	0.662
	RI	**0.9385**	0.7524	0.8466	0.6713	0.7429	0.8466	0.7211	0.8466	0.8466
	FM	0.6862	0.6406	**0.7791**	0.1242	0.6325	**0.7791**	0.6042	**0.7791**	**0.7791**
Spiral	ARI	**1**	0.0537	0.0161	0.0161	0.0161	0.016	0.9156	0.0161	0.015
	RI	**1**	0.5032	0.4105	0.4105	0.4105	0.4857	0.9636	0.4105	0.4081
	FM	**1**	0.4513	0.4761	0.4761	0.4761	0.428	0.9464	0.4761	0.4752
Flame	ARI	**0.9387**	0.152	0.0206	0.0206	0.0206	0.0206	**0.9387**	0.0206	0.0407
	RI	**0.9694**	0.5767	0.5375	0.5375	0.5375	0.5375	**0.9694**	0.5375	0.5455
	FM	**0.9636**	0.5956	0.6752	0.6752	0.6752	0.6752	**0.9636**	0.6752	0.6756
Jain	ARI	**0.9722**	0.8738	0.7901	0.7901	0.7901	0.1218	0.8595	0.9393	0.7901
	RI	**0.9867**	0.721	0.5362	0.5362	0.5362	0.4814	0.6909	0.9709	0.5362
	FM	**0.9891**	0.8055	0.6781	0.6781	0.6781	0.2259	0.7917	0.9714	0.6781
Aggregation	ARI	**0.9965**	0.815	0.815	0.0632	0.9883	0.9887	**0.9965**	0.815	0.967
	RI	**0.9985**	0.893	0.893	0.7865	0.995	0.9952	**0.9985**	0.893	0.9861
	FM	**0.9965**	0.8016	0.8016	0.0904	0.9884	0.9888	**0.9965**	0.8016	0.967
Compound	ARI	**0.9025**	0.4453	0.4366	0.0295	0.0295	0.0295	0.4366	0.4366	0.0295
	RI	**0.9635**	0.7184	0.7068	0.331	0.331	0.331	0.7068	0.7068	0.331
	FM	**0.9241**	0.6306	0.6277	0.4036	04036	0.4036	0.6277	0.6277	0.4036
R15	ARI	**0.9772**	0.2636	0.2962	0.0879	0.2962	0.2636	0.2636	0.2636	0.2636
	RI	**0.9971**	0.7506	0.8163	0.9031	0.8163	0.7506	0.7506	0.7506	0.7506
	FM	**0.9776**	0.3431	0.425	0.1396	0.425	0.3431	0.3431	0.3431	0.3431
D 31	ARI	**0.7555**	0.6971	0.6919	0.421	0.7098	0.6919	**0.7555**	0.7337	0.7337
	RI	**0.9831**	0.9784	0.978	0.9665	0.9793	0.979	**0.9831**	0.981	0.981
	FM	**0.7564**	0.6966	0.691	0.42	0.7079	0.691	**0.7564**	0.7334	0.7334
Dim128	ARI	**0.9733**	**0.9733**	0.1326	0.1004	0.116	0.1004	**0.9733**	0.364	0.116
	RI	**0.9969**	**0.9969**	0.5258	0.4267	0.6284	0.4267	**0.9969**	0.7888	0.6284
	FM	**0.9749**	**0.9749**	0.1274	0.1135	0.1459	0.1135	**0.9749**	0.3684	0.1459
Dim512	ARI	**0.9599**	**0.9599**	0.1423	0.1324	0.1423	0.1322	**0.9599**	0.3168	0.1039
	RI	**0.9954**	**0.9954**	0.5512	0.3781	0.5512	0.3792	**0.9954**	0.7497	0.424
	FM	**0.9624**	**0.9624**	0.1314	0.1137	0.1314	0.1132	**0.9624**	0.3298	0.1179
Iris	ARI	**0.5184**	0.4829	0.4799	0.05	0.4829	0.4829	**0.5184**	0.4961	0.3199
	RI	**0.7618**	0.7525	0.7511	0.6584	0.7525	0.7525	**0.7618**	0.7543	0.6917
	FM	**0.7069**	0.6769	0.675	0.1905	0.6769	0.6769	**0.7069**	0.6898	0.5547
Wine	ARI	**0.4125**	0.3494	0.3494	0.0746	0.2835	0.1797	**0.4125**	0.3494	0.3171
	RI	**0.7257**	0.6743	0.6743	0.668	0.6809	0.6919	**0.7257**	0.6743	0.702
	FM	**0.6272**	0.6121	0.6121	0.1827	0.5226	0.2907	**0.6272**	0.6121	0.5355
Breast Tissue	ARI	**0.2469**	0.2196	0.1659	0.0781	0.2196	0.2247	0.1454	0.2273	0.1659
	RI	0.7178	0.7642	0.6923	**0.8102**	0.7642	0.78	**0.8102**	0.7196	0.6923
	FM	**0.4078**	0.3618	0.336	0.1562	0.3618	0.3571	0.2642	0.3997	0.336
Glass	ARI	0.2483	**0.267**	0.2223	0.0757	0.2223	**0.267**	0.2318	0.2508	0.2223
	RI	0.6295	0.6646	0.6417	**0.7154**	0.6417	0.6646	0.6689	0.634	0.6417
	FM	**0.5027**	0.4879	0.4703	0.1562	0.4703	0.4879	0.4858	0.5026	0.4703
Musk	ARI	0.5709	0.5709	0.5709	0.0193	0.5709	0.0193	0.5709	0.5709	**0.5898**
	RI	0.4965	0.4965	0.4965	0.5037	0.4965	0.5037	0.4965	0.4965	**0.5181**
	FM	**0.4554**	**0.4554**	**0.4554**	0.1727	**0.4554**	0.1727	**0.4554**	**0.4554**	0.4371
Scadi	ARI	0.3748	0.6205	0.3748	**0.6399**	0.3966	0.1356	0.3275	0.3748	0.3966
	RI	0.7039	0.8637	0.7039	**0.867**	0.8041	0.7329	0.7602	0.7039	0.8041
	FM	0.5776	0.7075	0.5776	**0.7268**	0.5057	0.2678	0.4807	0.5776	0.5057

Tableau 4.4. Corrélation entre les CVIs et les indices externes FM, RI et ARI pour seize ensembles de données.

Dataset		Indice								
		CDBCVI	Sil	CH	Dunn	DB	S-Dbw	DBCV	VCN	VCVI
Path-based	ARI	0.3527	0.3139	**0.499**	-0.194	-0.206	0.1459	0.1326	0.4156	0.4904
	RI	0.1558	0.2167	**0.4496**	-0.0472	-0.2089	0.1791	0.0298	0.2895	0.2950
	FM	0.5166	0.4123	0.5277	-0.3412	-0.1133	0.2327	0.2247	0.4652	**0.6884**
Spiral	ARI	**0.7918**	-0.5151	-0.4816	-0.4461	-0.8092	-0.7663	0.5867	-0.6387	-0.4330
	RI	**0.9278**	-0.6378	-0.5962	-0.5245	-0.8836	-0.9230	0.6376	-0.7598	-0.6004
	FM	**0.7692**	-0.4799	-0.3460	-0.4195	-0.8756	-0.9119	0.5870	-0.5497	-0.3809
Flame	ARI	**0.9895**	0.3822	-0.3930	-0.7863	-0.7216	-0.9704	0.7796	0.2612	0.0405
	RI	**0.9892**	0.3983	-0.3517	-0.7659	-0.6994	-0.9676	0.7912	0.2966	0.0713
	FM	**0.9210**	0.4766	-0.1071	-0.6247	-0.5093	-0.9139	0.8390	0.4719	0.3318
Jain	ARI	**0.7645**	0.4128	0.114	-0.3206	-0.3447	-0.6540	0.28	0.7104	0.3970
	RI	**0.7106**	0.3302	-0.0999	-0.5154	-0.5594	-0.5609	0.2832	0.6079	0.3820
	FM	**0.7403**	0.3966	0.0393	-0.4129	-0.4020	-0.6183	0.2832	0.6732	0.4664
Aggregation	ARI	0.7596	0.4985	0.6902	-0.5482	0.4559	0.6939	0.7946	0.4541	**0.8321**
	RI	0.4411	0.2967	0.4017	-0.0865	0.3638	0.4977	0.4647	0.1446	**0.5034**
	FM	0.7621	0.4941	0.6837	-0.5492	0.4604	0.685	0.8037	0.4489	**0.8392**
Compound	ARI	**0.7634**	-0.0112	-0.4967	-0.6874	-0.4399	0.0238	0.7314	-0.1741	0.4700
	RI	**0.8308**	-0.1061	-0.7088	-0.8341	-0.4229	-0.0500	0.6098	-0.2426	0.1762
	FM	**0.7876**	0.0679	-0.3605	-0.5824	-0.4084	0.038	0.7799	-0.0987	0.6435
R15	ARI	**0.9788**	0.0717	0.5118	0.0331	0.4684	0.7999	0.4009	-0.1308	0.4483
	RI	**0.9503**	0.0807	0.4563	0.4758	0.0808	0.6945	0.0286	-0.6006	0.0533
	FM	**0.9787**	0.063	0.4727	0.3496	0.4256	0.7995	0.189	-0.2783	0.3647
D 31	ARI	0.8568	0.8207	**0.9393**	-0.1008	0.9236	-0.0168	0.3514	0.9060	0.8340
	RI	0.5671	**0.8487**	0.7683	0.2952	0.7293	0.1354	0.6119	0.7254	0.4895
	FM	0.2958	0.1404	0.2543	-0.1028	0.2894	-0.0223	0.1870	0.8589	**0.8869**
Dim 128	ARI	**0.9624**	0.7513	-0.1209	-0.7207	0.5382	-0.1453	0.8872	0.4877	0.3318
	RI	0.8641	**0.9668**	0.1356	-0.9236	0.8232	-0.2202	0.7233	0.6926	0.5269
	FM	**0.9691**	0.7544	-0.1277	-0.7210	0.5399	-0.1387	0.8964	0.4904	0.3271
Dim512	ARI	**0.8579**	0.7575	-0.0193	-0.7003	0.4856	-0.1009	0.5946	0.4563	0.3214
	RI	0.6320	**0.9596**	0.2809	-0.9331	0.7669	-0.1968	0.4989	0.7656	0.5420
	FM	**0.8689**	0.7383	-0.0457	-0.6925	0.4562	-0.1119	0.5857	0.4422	0.3102
Iris	ARI	**0.9638**	0.7965	0.9449	-0.8033	0.7045	0.6790	0.7133	0.9450	0.1307
	RI	0.8406	0.7270	**0.8449**	-0.5605	0.6591	0.7079	0.5804	0.8167	-0.0607
	FM	**0.9461**	0.7725	0.9157	-0.8683	0.6749	0.6626	0.7245	0.9317	0.1786
Wine	ARI	0.5318	0.4882	0.4229	-0.4873	0.38	-0.0642	0.3735	**0.6173**	0.4603
	RI	0.2343	0.1662	0.0504	-0.2309	0.0690	-0.0179	0.2028	**0.2571**	0.2182
	FM	0.7199	0.6671	0.6417	-0.6386	0.5900	-0.1023	0.4830	**0.7940**	0.6249
Breast Tissue	ARI	0.7921	0.73	0.4364	-0.9631	0.7558	-0.7332	0.6556	0.7173	**0.8113**
	RI	-0.9660	-0.5745	-0.7221	**0.7179**	-0.5881	0.5218	-0.3631	-0.6884	-0.5629
	FM	**0.8941**	0.7519	0.5591	-0.9880	0.7622	-0.7761	0.6788	0.7729	0.8477
Glass	ARI	**0.9340**	0.8308	0.8169	-0.7289	0.4931	0.8275	0.5954	0.7022	0.8081
	RI	-0.8599	-0.8819	-0.9640	0.5969	-0.6154	**0.9683**	-0.4239	-0.9061	-0.6323
	FM	**0.8568**	0.8153	0.8379	-0.7767	0.4822	-0.8774	0.6165	0.7862	0.8553
Musk	ARI	0.8213	0.1794	0.7608	-0.093	**0.8745**	-0.5286	0.2133	0.7652	0.3169
	RI	0.1013	-0.2868	-0.1457	-0.4709	-0.0795	-0.1956	0.3014	0.1383	**0.4839**
	FM	**0.9170**	-0.028	0.8410	-0.7440	0.7113	-0.8904	0.5063	0.8371	0.7374
Scadi	ARI	0.6391	0.7090	0.3241	0.5097	0.4044	-0.2244	0.5652	0.2557	**0.7382**
	RI	0.1681	0.3948	-0.3143	-0.0173	**0.7289**	-0.1572	0.2693	-0.3612	0.3996
	FM	0.7580	0.7341	0.5257	-0.6516	0.2262	-0.2246	0.6249	0.4540	**0.7945**

Les tableaux 4.2, 4.3 et 4.4 montrent que l'indice CDBCVI proposé surpasse les autres CVIs dans presque tous les ensembles de données, en particulier pour les ensembles de données synthétiques qui contiennent des clusters de formes arbitraires et une densité variable. D'après le tableau 4.2, nous pouvons voir que dans la plupart des cas, CDBCVI a pu déterminer avec succès le nombre correct de clusters par rapport aux autres CVIs, en particulier les cas des clusters avec des formes arbitraires et qui se chevauchent. Notez que le nombre correct de clusters n'a pas été trouvé par tous les CVIs dans le cas des datasets Compound, D31, Breast Tissue, Glass et Scadi, car l'ensemble des partitions généré par les algorithmes NBC et DBSCAN peut ne pas contenir une partition avec le nombre correct de clusters (cas de D31, Glass et Scadi) ou la partition générée qui avait le nombre correct de clusters n'est pas la meilleure, comparée à la partition correcte (Compound, Breast Tissue). Ce problème est clairement visible à partir des tableaux 4.2 et 4.3. Par exemple, dans le cas de l'ensemble de données D31, il n'y a pas de partition parmi les partitions générées qui contient le nombre correct de clusters (31) (voir tableau 4.2). Bien que la meilleure partition générée par les indices Sil, CH et S_Dbw ait le nombre de clusters (30) le plus proche au nombre correct, les meilleures partitions sélectionnées par les indices CDBCVI et DBCV, ayant 29 clusters, ont les meilleures valeurs de FM, RI et ARI (voir tableau 4.3). Pour certains ensembles de données, bien que le nombre de clusters sélectionné par CDBCVI soit correct et identique à celui trouvé par d'autres CVIs, les valeurs de FM, RI et ARI trouvées par CDBCVI sont les meilleures (voir, par exemple, le cas de l'ensemble de données Flame, où tous les CVIs sélectionnent le même nombre de clusters qui est correct, mais les valeurs de chaque indice externe parmi FM, RI et ARI diffèrent considérablement). De plus, dans le cas de certains ensembles de données réels, l'étiquetage (la classification existante) ne maintient pas la structure des données à base de densité. Par exemple, dans l'ensemble de données Iris, l'étiquetage fait référence à trois clusters sphériques. Cependant, deux parmi ces clusters se chevauchent et ne forment donc qu'un seul cluster d'un point de vue basé sur la densité. Dans ce cas, CDBCVI préfère 2 clusters au lieu de 3.

Le tableau 4.3 illustre les valeurs des indices FM, RI et ARI de la meilleure partition sélectionnée par chaque CVI. Ce tableau montre que l'indice CDBCVI surpasse tous les autres CVIs dans la plupart des ensembles de données, en particulier les ensembles de données ayant des clusters de formes arbitraires, comme le cas de Spiral, Path-based et Jain (voir la structure complexe et le chevauchement important entre les clusters dans le cas de l'ensemble de données Path-based).
Pour l'ensemble de données Glass, CDBCVI fournit le meilleur résultat selon FM et un résultat très proche, selon ARI, au meilleur fourni par les indices Sil et S_Dbw. En outre, pour l'ensemble de données Musk, il fournit le meilleur résultat en termes de FM et le résultat le plus proche, en termes d'indice ARI, au meilleur fourni par l'indice VCVI.

Bien que l'indice DBCV ait pu trouver les meilleures valeurs de FM, RI et ARI comme l'indice CDBCVI sur les ensembles de données Flame, Agreggation, D31, Dim128, Dim512, Iris et Wine, il présente de mauvaises performances contre la structure de données complexe, la densité variable et le chevauchement important dans le cas des ensembles de données Path-based, Jain et Compound. Compte tenu des cas où CDBCVI et DBCV produisent le nombre correct de clusters et les meilleures valeurs de FM, RI et ARI, CDBCVI surpasse DBCV en

termes de corrélation, à l'exception du cas de l'ensemble de données « aggregation » dans lequel CDBCVI produit un résultat proche de DBCV.

Le tableau 4.4 montre la corrélation entre chaque CVI et chaque indice externe parmi FM, RI et ARI. Pour la plupart des ensembles de données, CDBCVI surpasse les autres CVIs. Compte tenu des résultats de l'ensemble de données Wine, VCN fournit les meilleurs résultats, suivi de CDBCVI qui fournit les résultats les plus proches des meilleurs. Pour les ensembles de données Path-based et D31, CDBCVI fournit les résultats les plus proches des meilleurs fournis par CH et VCVI en termes de ARI et de FM respectivement.

Afin d'illustrer l'efficacité de chaque CVI pour l'évaluation des partitions, nous visualisons à titre d'exemple, dans la figure 4.6 ci-aprés, les résultats expérimentaux obtenus par les neuf CVIs sur l'ensemble de données Jain. De cette figure, nous pouvons voir que pour l'ensemble de données Jain, par rapport aux autres CVIs, CDBCVI fournit la meilleure corrélation et les meilleures valeurs de FM, RI et ARI, qui sont les valeurs maximales obtenues à partir des 40 partitions générées par les algorithmes NBC et DBSCAN. Par conséquent, CDBCVI reconnaît la meilleure solution qui lui est présentée. Il fait donc une bonne comparaison des solutions, du fait qu'une meilleure valeur de CDBCVI implique considérablement les meilleures valeurs de FM, RI et ARI. En fait, cela vaut également pour d'autres ensembles de données.

Pour donner une impression directe sur les résultats de l'évaluation, nous visualisons, dans la figure 4.7 ci-après, les meilleures valeurs des indices FM, RI et ARI trouvées par chaque CVI sur les ensembles de données sélectionnés (nous avons considéré les 40 exécutions réalisées pour chaque ensemble de données en utilisant deux algorithmes DBSCAN et NBC avec une variation des paramètres). Cette figure confirme la supériorité de l'indice CDBCVI proposé par rapport à tous les autres indices pour l'évaluation des ensembles de données, en particulier les ensembles de données ayant des clusters de formes arbitraires. De plus, la figure 4.8 illustre les résultats globaux de chaque CVI, en termes de nombre optimal de clusters trouvé, les meilleures valeurs de FM, RI et ARI et la meilleure corrélation entre les CVIs et les indices externes FM, RI et ARI. Cette figure montre le pourcentage de succès obtenue par chaque CVI, en prenant en compte tous les ensembles de données. Nous pouvons voir sur cette figure que CDBCVI surpasse considérablement les autres CVIs en termes de nombre correct de clusters trouvé, les meilleures valeurs de FM, RI et ARI obtenues et la meilleure corrélation trouvée.

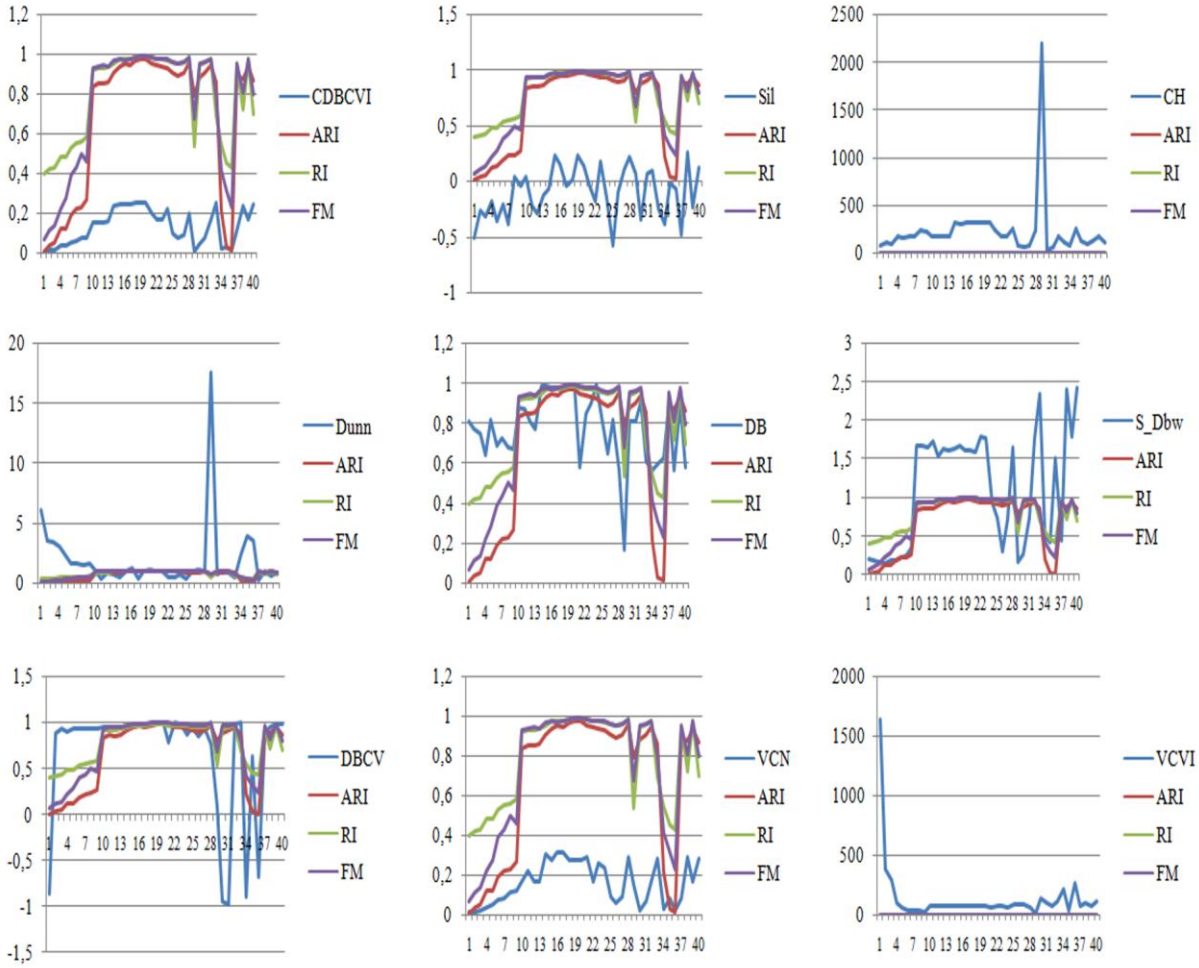


Figure 4.6. Résultats de l'ensemble de données Jain obtenu par les neuf indices de validation sur les différentes partitions (de 1 à 40) générées par les algorithmes DBSCAN et NBC.

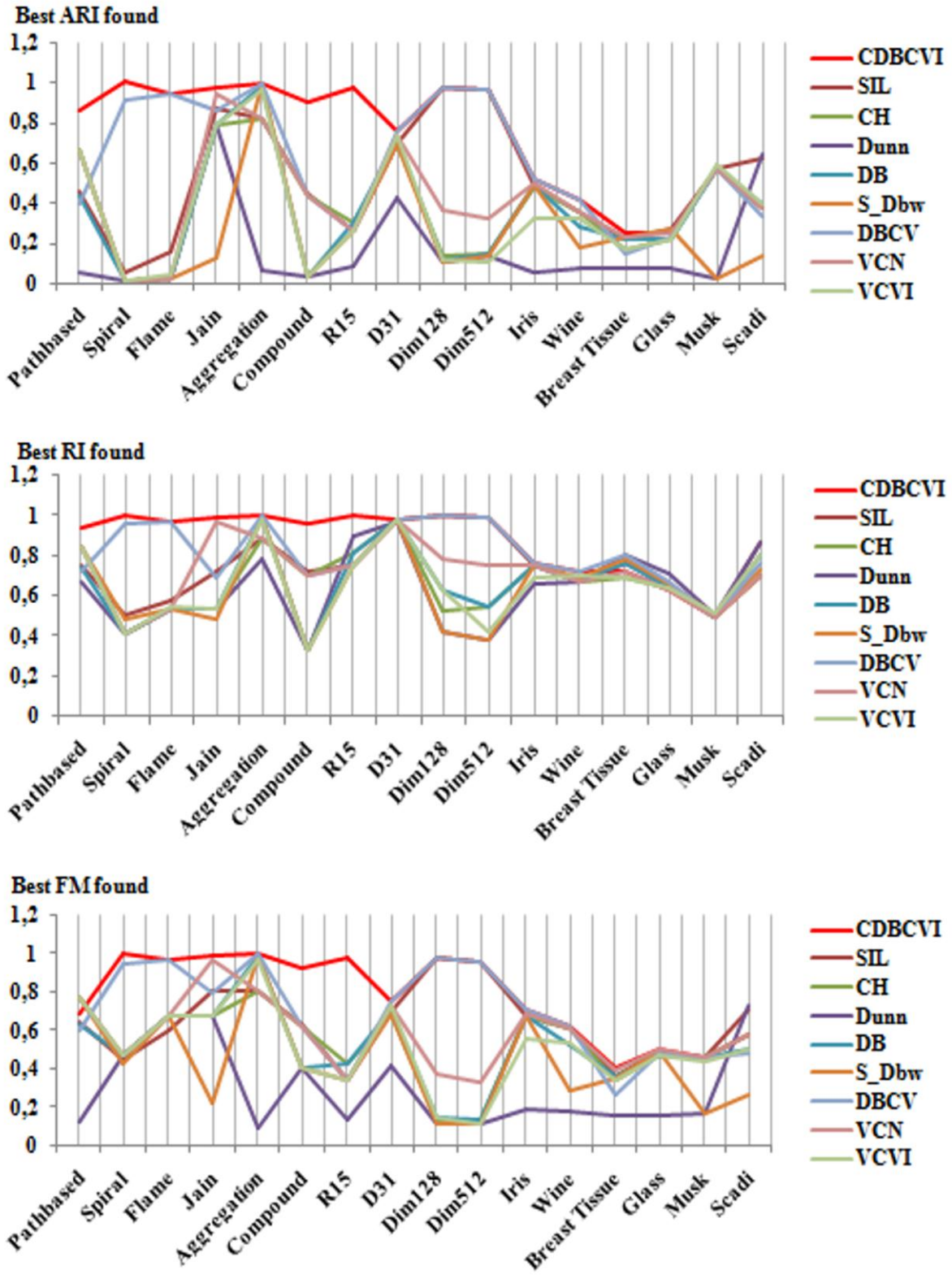


Figure 4.7. Comparaison des indices de validation CDBCVI, Sil, CH, Dunn, DB, S_Dbw, DBCV, VCN et VCVI pour seize ensembles de données.

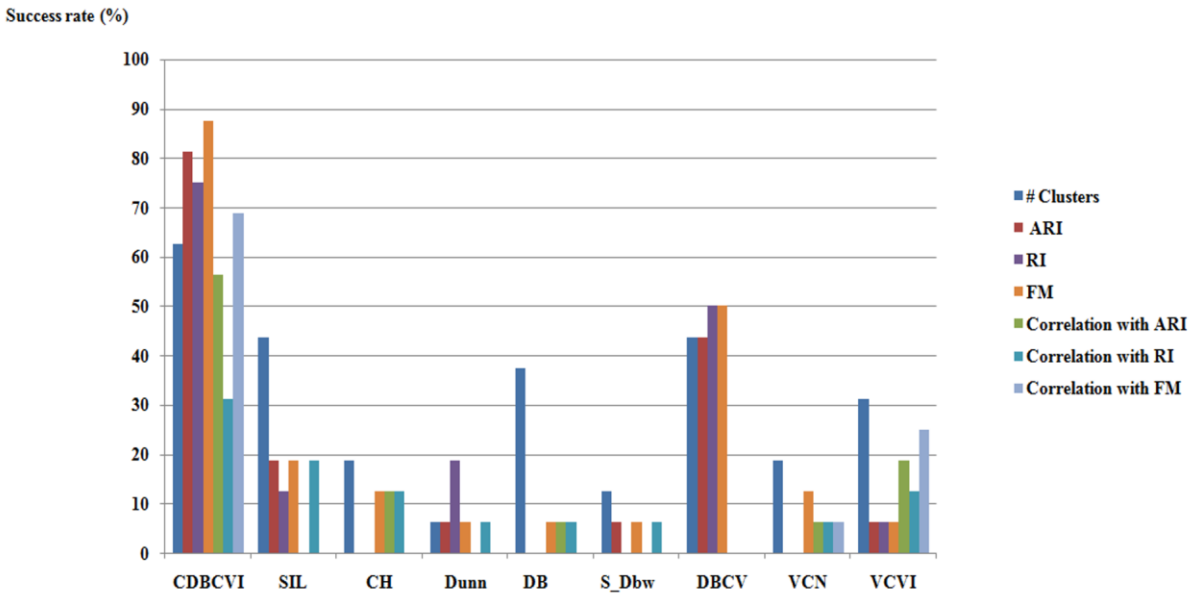


Figure 4.8. Résultats globaux de chaque CVI à partir des expériences réalisées sur seize ensembles de données, en termes du nombre optimal de clusters trouvé, les meilleures valeurs de ARI, RI et FM et la meilleure corrélation avec ARI, RI et FM.

4.4. Conclusion

Dans ce chapitre, nous avons proposé un nouvel indice de validation de clustering basé sur la connectivité et la densité (CDBCVI). L'indice CDBCVI permet de faciliter le choix d'un algorithme de clustering et de ses paramètres d'entrée pour une situation particulière. Contrairement à la plupart des indices proposés dans la littérature pour la validation des clusters globulaires, CDBCVI mesure la compacité de chaque cluster (validité intra-cluster) et la séparation (validité inter-clusters) entre clusters, en utilisant les concepts de connectivité directe et indirecte définis par Gabriel et Sokal dans le graphe de Gabriel (Gabriel & Sokal, 1969). Les nouvelles définitions des mesures de compacité et de séparation permettent à l'indice CDBCVI proposé de traiter correctement les cas de clusters de formes arbitraires et des outliers. Les résultats expérimentaux sur divers ensembles de données prouvent l'efficacité de l'indice CDBCVI proposé pour la validation des résultats de clustering, en particulier dans le cas de structures de données complexes (clusters de formes et de densités différentes et également le cas de densité variable au sein d'un cluster) et de chevauchements importants entre les clusters, où les indices de validation proposés dans la littérature peuvent ne pas atteindre des résultats satisfaisants.

Chapitre 5

Clustering automatique à base de densité utilisant des métaheuristiques

Sommaire

5.1. Introduction

Comme nous avons déjà vu dans le chapitre 2, la plupart des algorithmes de clustering existant dans la littérature, tel que *k*-means, ne peuvent pas traiter correctement les clusters de

formes arbitraires et les outliers, dépendent de paramètres définis par l'utilisateur et souffrent du problème bien connu de minima locaux. Les algorithmes de clustering basés sur la densité peuvent traiter les cas de clusters de formes arbitraires. Cependant, la plupart d'entre eux ne peuvent pas traiter les ensembles de données ayant des clusters de densité variable et dépendants de quelques paramètres définis par l'utilisateur. Aussi, comme nous avons déjà discuté dans le chapitre 2, la majorité des approches de clustering basées sur les métaheuristiques qui ont été proposées récemment, pour surpasser les algorithmes de clustering classiques, présentent des lacunes. En effet, elles ne peuvent pas traiter les ensembles de données ayant des clusters de formes arbitraires et de différentes densités.

Dans ce chapitre, nous présentons nos contributions qui consistent à proposer des approches de clustering permettant de remédier aux différentes difficultés et lacunes posées par la plupart des approches existantes dans la littérature.

Nous présentons, dans un premier temps, notre première contribution qui consiste à utiliser la métaheuristique à solution unique VNS afin de remédier à la difficulté du choix de la valeur du paramètre de l'algorithme NBC et automatiser ainsi le processus de clustering. Dans un deuxième temps, nous présentons deux autres contributions qui consistent à utiliser l'algorithme de colonie d'abeilles artificielles afin d'automatiser et améliorer la qualité du clustering à base de densité. La première contribution parmi ces deux est basée sur l'algorithme NBC, alors que la deuxième est basée sur l'algorithme DBSCAN.

Afin de montrer l'efficacité des différentes approches proposées, nous présentons aussi, dans ce chapitre, l'étude expérimentale menée sur plusieurs ensembles de données. Les comparaisons ont été éffectuées avec des algorithmes de clustering récents et bien connus.

5.2. Recherche par voisinage variable pour le clustering automatique à base de densité

L'algorithme de recherche par voisinage variable (VNS) (Hansen et al., 2008) est devenu, récemment, l'une des métaheuristiques les plus intéressantes, en raison de sa simplicité et de son fonctionnement sans paramètres. Dans cette section, nous proposons une nouvelle approche de clustering basée sur l'heuristique VNS.

Etant donné que l'algorithme NBC, présenté dans le chapitre 2, est très efficace dans le cas de clusters de formes arbitraires et de différentes densités, nous proposons de tirer avantage de cet algorithme et de la version de VNS standard, afin d'éviter le problème du choix des valeurs des paramètres et d'améliorer la qualité des solutions de clustering.

5.2.1. Description de l'algorithme de recherche par voisinage variable de base

VNS est une métaheuristique proposée pour résoudre les problèmes d'optimisation combinatoire. Elle applique explicitement une stratégie de recherche basée sur un changement systématique des structures de voisinage d'une solution. Habituellement, l'ensemble de voisinages $\{N_1, \ldots, N_t, \ldots, N_{max}\}$ doit être prédéfini avant d'exécuter VNS. L'algorithme VNS est décomposé en deux phases : une phase déterministe de recherche locale qui permet de converger vers un optimum local, et une phase stochastique mise en place pour s'en échapper.

La partie stochastique de l'algorithme consiste à générer, à partir de la solution courante *s*, une nouvelle solution *s′* selon un voisinage donné. Cette phase est appelée phase de perturbation (*shaking* en anglais).

Soit N_t l'ensemble des structures de voisinage (t = 1,…, t_{max}), et $N_t(s)$ l'ensemble des solutions au $t^{ième}$ voisinage de *s*. Les étapes de la recherche par voisinage variable de base (BVNS) sont données dans l'algorithme 1. Le critère d'arrêt du VNS est le temps CPU maximum autorisé, le nombre maximal d'itérations atteint ou le nombre maximum d'itérations entre deux améliorations. Le lecteur intéressé peut se référer à (Mladenovic & Hansen, 1997; Hansen & Mladenovic, 2001) pour plus de détails sur VNS.

Algorithme 1. VNS de base

$s_0 \leftarrow$ GenerateInitialSolution, choisir $\{N_t\}$, t = 1,…, t_{max}.
Répéter
$t \leftarrow 1$; // l'index de voisinage
Répéter
$s' \leftarrow$ Shake(s, t) // choix aléatoire à partir de $N_t(s)$;
$s'' \leftarrow$ BestImprovement(s') // Recherche locale par rapport à N_t.
s, $t \leftarrow$ NeighborhoodChange(s, s'', t); // changement de voisinage
Jusqu'à $t = t_{max}$;
Jusqu'à condition de terminaison atteinte
retourner *s*

5.2.2. Description de l'approche de recherche par voisinage variable proposée pour le clustering

L'approche VNS proposée utilise un codage de solution basé sur l'algorithme de clustering NBC présenté dans (Zhou et al., 2005). Cette approche peut identifier des solutions diversifiées lors des itérations à travers les 3 structures de voisinage que nous avons définies.

5.2.2.1. Codage des solutions

Étant donné que l'algorithme NBC utilise le nombre minimum d'objets dans le voisinage comme paramètre d'entrée et la même valeur de ce paramètre peut donner des solutions différentes, dans l'approche VNS proposée, nous proposons à travers cet algorithme un schéma de codage de solutions basé sur la densité.

Comme le montre la figure 5.1 ci-après, nous représentons une solution de clustering par un vecteur composé de N+1 composantes, où N est le nombre d'objets représentatifs de clusters (objets noyaux) et la dernière composante est la valeur du paramètre k qui est le nombre minimum d'objets considéré dans un voisinage. Pour construire une solution durant les différentes itérations, tout d'abord, la valeur du paramètre k est sélectionnée. Ensuite, l'algorithme NBC est appliqué en utilisant la valeur sélectionnée du paramètre k. Le premier objet noyau utilisé pour construire chaque cluster par l'algorithme NBC est ajouté en tant que composant de la solution (individu) codée. Ainsi, comme le montre la figure 5.2, la même valeur de k peut donner des solutions différentes.

Avec ce codage de solutions, nous n'avons pas besoin d'avoir le nombre de clusters et le paramètre *k* de l'algorithme NBC comme paramètres définis par l'utilisateur. Nous pouvons automatiquement trouver le bon nombre de clusters et les meilleurs objets représentatifs, en exploitant l'espace de recherche avec différentes valeurs de *k* et différents objets représentatifs durant les différentes itérations de l'algorithme VNS proposée.

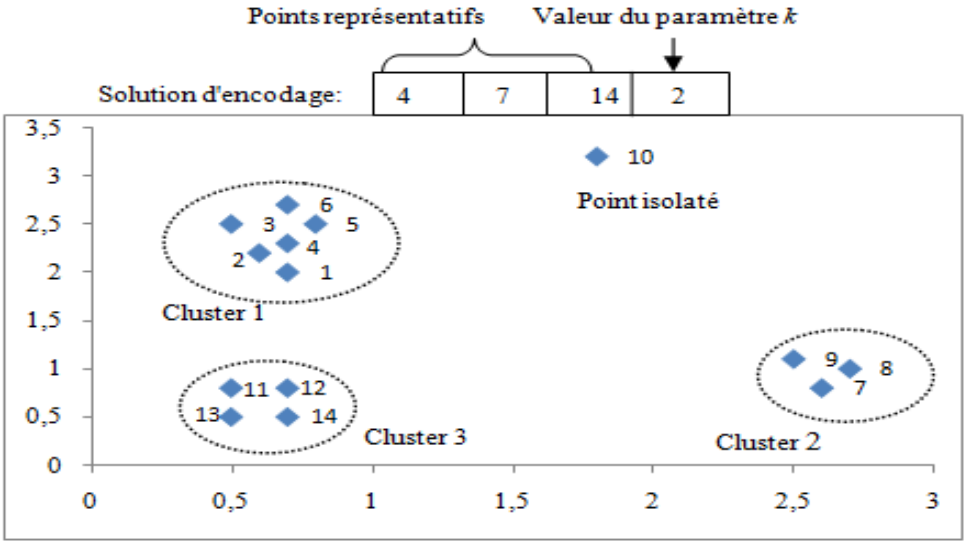


Figure 5.1. Exemple d'un codage de solution générée par l'algorithme NBC.

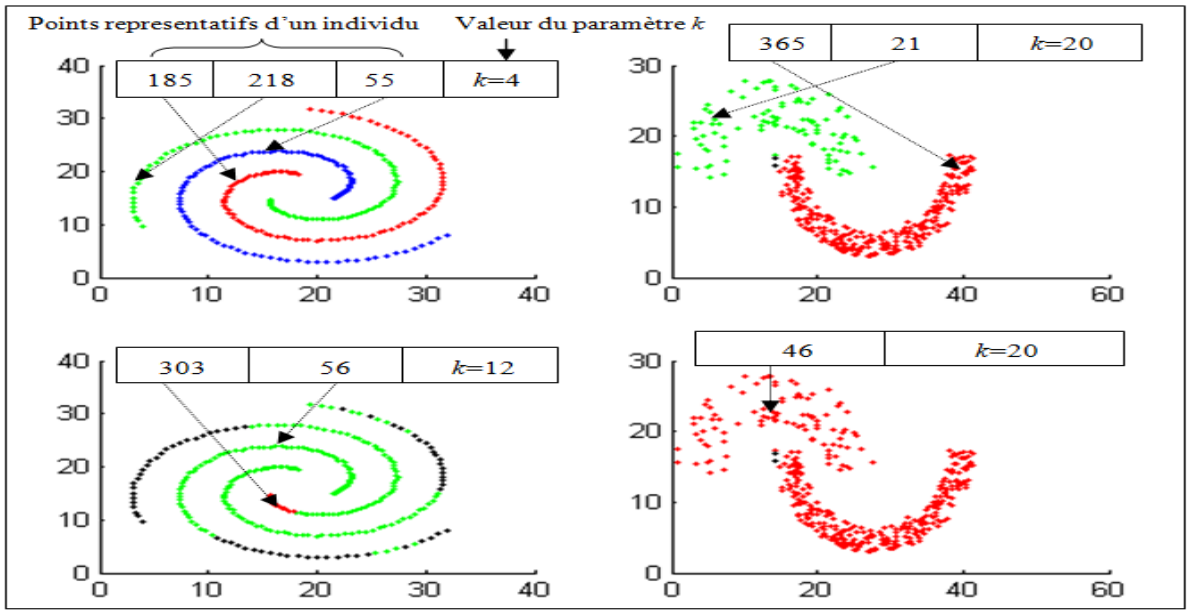


Figure 5.2. Exemples de solutions codées par l'algorithme NBC.

5.2.2.2. Définition des structures de voisinage

Étant donné qu'une solution de clustering est représentée par un vecteur composé de points représentatifs de clusters (points noyaux) et de la valeur du paramètre *k*, ses solutions voisines peuvent être générées en échangeant les points représentatifs, l'ordre de certains points

représentatifs dans le vecteur ou la valeur du paramètre *k*. Le codage proposé permet, donc, de prédéfinir efficacement les structures de voisinage dans VNS. Nous avons défini trois structures de voisinage N_1, N_2 et N_3 dans l'algorithme VNS proposé.

- Une recherche locale par permutation de deux points représentatifs (*a two-point swap local search method*) est utilisée pour trouver toutes les solutions voisines dans N_1.
- Un échange d'un point représentatif (*a representative point exchange local search*) est utilisé pour trouver toutes les solutions de voisinage dans N_2.
- Un échange de la valeur *k* (*a k value exchange local search*) est utilisé pour trouver toutes les solutions de voisinage dans N_3.

La première structure de voisinage N_1 est motivée par le rôle important des positions des composants d'une solution dans l'exploration de l'espace de recherche. Dans cette structure, c.-à-d. la recherche locale par permutation de deux points, deux points représentatifs dans le vecteur de solution sont permutés. Comme la montre la figure 5.3 (a), une solution voisine de la solution courante 5-100-7-20-200 k = 3 est 5-20 -7-100-200 k = 3 si les points 100 et 20 sont échangés.

La deuxième structure N_2 consiste en une recherche locale par échange de points représentatifs. Dans cette structure un point représentatif est sélectionné parmi ceux du vecteur représentant la solution et il est remplacé par un nouveau point sélectionné du même cluster. Tandis que la troisième structure N_3 consiste en une recherche par échange de la valeur de *k*. Dans cette structure, une nouvelle valeur de *k* est sélectionnée aléatoirement, puis les points représentatifs sont générés de nouveau par l'algorithme NBC. La figure 5.3 illustre une solution voisine d'une solution dans chacune des structures de voisinage prédéfinie.

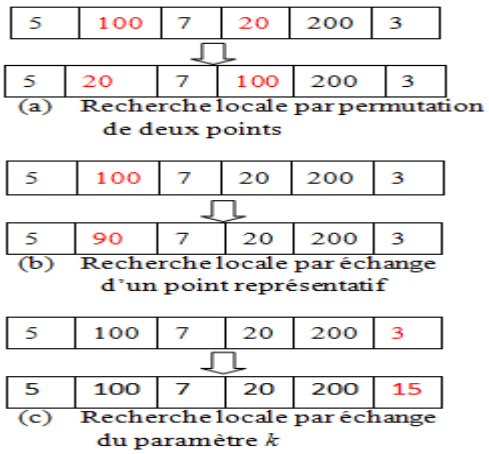


Figure 5.3. Un exemple de solution voisine d'une solution selon les trois structures de voisinage définies.

5.2.2.3. Description du processus de clustering par VNS

Comme le montre la figure 5.4 ci-dessous, le processus de recherche d'une solution de clustering par l'algorithme VNS proposé est décrit comme suit :

En premier lieu, une solution initiale S est générée comme solution courante et l'indice de voisinage t est initialisé à 1. Puis, dans chaque structure de voisinage N_t (t = 1, 2, 3), trois étapes principales sont réalisées, à savoir : perturbation, recherche locale et déplacement. L'étape de perturbation consiste à sélectionner une solution S_s aléatoirement pour perturber le processus de recherche de solution. Ensuite, dans l'étape de recherche locale, toutes les solutions voisines de S_s sont générées conformément à la définition de N_t. Par la suite, la solution optimale locale parmi toutes les solutions voisines S_l est comparée à la solution courante S, dans l'étape de déplacement. Si S_l est meilleur que S, S sera remplacée par S_l et l'algorithme recommence avec t = 1. Sinon, une nouvelle étape de perturbation recommence dans le $(t+1)^{ème}$ voisinage N_{t+1}. Ce processus est répété par VNS jusqu'à ce qu'un critère d'arrêt soit satisfait. Ce critère est testé à la fin du passage par la dernière structure de voisinage N_3, si la solution courante reste la meilleure. Dans notre cas, nous avons défini le critère d'arrêt comme le nombre maximal d'itérations entre deux améliorations égal à 30.

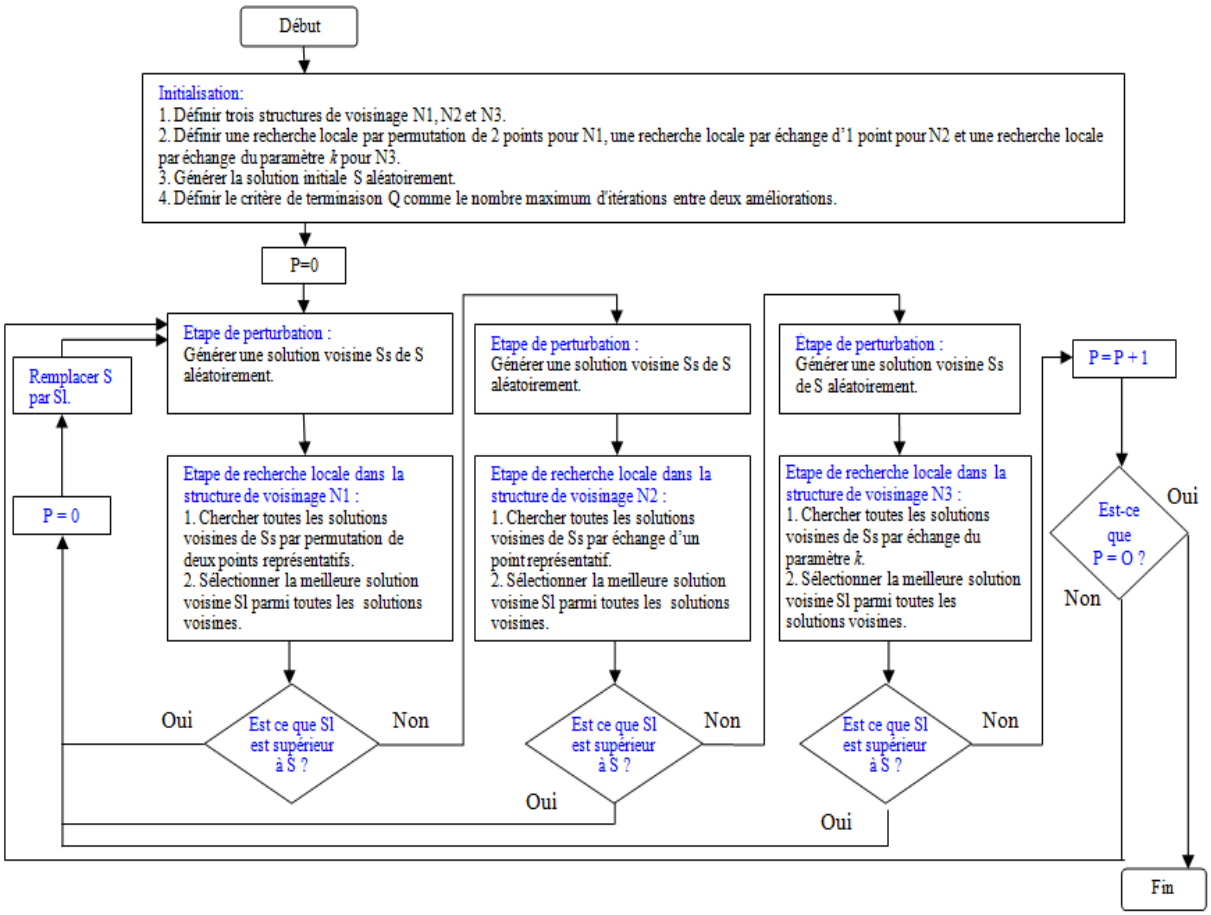


Figure 5.4. Organigramme de l'algorithme VNS proposée pour le clustering à base de densité.

5.3. Développement d'approches à base de colonie d'abeilles artificielles pour le clustering automatique à base de densité

L'algorithme de colonie d'abeilles artificielles (ABC) est une métaheuristique à base de population devenue récemment l'un des paradigmes d'intelligence en essaim les plus intéressants, en raison de son nombre limité de paramètres de contrôle.

Dans cette section, nous proposons des approches de clustering mono- et multi-objectif, en utilisant le paradigme à base de colonie d'abeilles artificielles. L'algorithme ABC est utilisé, dans un premier temps, comme un outil de réglage des paramètres de l'algorithme NBC, tout en améliorant la qualité des résultats de clustering par une exploration globale de l'espace de recherche. Dans un deuxième temps, il sera utilisé pour remédier aux difficultés rencontrées par l'algorithme DBSCAN et améliorer la qualité des résultats. Avant de décrire ces approches, nous présentons quelques travaux dont nous avons tiré avantage et qui ont étendu l'algorithme ABC pour faire face aux différents problèmes d'optimisation multi-objectif.

5.3.1. Travaux liés à l'optimisation multi-objectif par l'algorithme ABC

Ces dernières années, de nombreux chercheurs ont étendu l'algorithme ABC pour traiter divers problèmes d'optimisation multi-objectif (Martín-Moreno & Vega-Rodríguez, 2018; Saif et al., 2019), en particulier le problème de clustering multi-objectif (MOC). Plusieurs études ont démontré que pour le problème MOC, l'algorithme ABC étendu a surpassé l'algorithme ABC standard en termes de convergence, de diversité et de qualité des solutions. Par exemple, dans (Akbari et al., 2012), un algorithme ABC multi-objectif (MOABC) a été proposé. Cet algorithme utilise une archive externe de taille fixe et une méthode de sélection pour enregistrer les meilleures solutions à partir du front Pareto. Les abeilles employées utilisent l'archive externe pour ajuster leurs positions.
Dans (Zhong et al., 2014), un ABC étendu appelé dMOABC a été proposé pour l'optimisation multi-objectif. L'espace de recherche dans l'algorithme dMOABC est divisé en trois colonies : deux colonies de base et une colonie synthétique partageant des informations avec les deux autres. Dans la colonie synthétique, les abeilles employées et les abeilles spectatrices mettent à jour leurs positions en utilisant la meilleure solution extraite d'une archive externe, au lieu d'utiliser un voisin aléatoirement.

Wang et Li (2015) ont proposé un algorithme ABC multi-objectif appelé MABC. Cet algorithme fournit une méthode améliorée pour maintenir les meilleures solutions de la population selon les rangs des solutions non dominées et de la distance de surpopulation. Il adopte de nouveaux modèles de recherche adaptative pour les abeilles employés et spectatrices. Pour les abeilles spectatrices, un nouveau schéma de sélection remplace la sélection par la roulette.

En (2016), Kishor et al. ont proposé un algorithme ABC multi-objectif appelé NSABC basé sur le tri des solutions non dominées. Cet algorithme a été utilisé pour le clustering des données. NSABC utilise une nouvelle approche dans la phase des abeilles employées pour atteindre, à la fois, les objectifs de convergence et de diversité. De plus, il utilise une archive externe pour stocker les meilleures et les plus diverses solutions trouvées au cours du

processus de recherche. Dans (Amarjeet & Chhabra, 2017), un algorithme ABC à deux archives (TA-ABC) a été proposé pour résoudre le problème de regroupement de modules logiciels multi-objectif. TA-ABC utilise un concept à deux archives et un indicateur de qualité évaluant la fonction fitness des sources alimentaires pour la sélection au lieu de la méthode de sélection par dominance de Pareto.

5.3.2. Approche multi-objectif par combinaison des algorithmes ABC et NBC

Dans cette section, nous proposons une nouvelle approche de clustering automatique à base de voisinage en développant un algorithme de colonie d'abeilles artificielles multi-objectif, appelée NBC-MOABC (*an automatic neighborhood-based clustering approach using a multi-objective artificial bee colony* (NBC-MOABC) algorithm). L'approche proposée est basée sur le paradigme de colonie d'abeilles artificielles initialement proposé par Karaboga (2005) combiné avec l'algorithme de clustering NBC proposé dans (Zhou et al., 2005), dans un cadre multi-objectif. Une nouvelle méthode d'initialisation des sources de nourriture et une stratégie de recherche locale déterministe (les équations de recherche de solution sont modifiées dans NBC-MOABC par rapport à l'algorithme ABC standard) sont proposées pour assurer un équilibre entre le processus de diversification et d'intensification dans l'espace de recherche. Aussi, une phase de mutation est intégrée dans l'approche NBC-MOABC pour améliorer la capacité de recherche globale. De plus, deux fonctions objectif sont proposées pour améliorer l'évaluation de la qualité des solutions de clustering, vu que la considération d'un seul critère (objectif) peut ne pas être conforme aux formes complexes des clusters.

NBC-MOABC consiste, tout d'abord, à déterminer un ensemble de solutions non dominées (front de Pareto) avec éventuellement des nombres de clusters différents. Par la suite, un simple décideur est utilisé pour sélectionner la meilleure solution sur la base d'un compromis entre les deux objectifs qui sont conflictuels.

5.3.2.1. Représentation et génération des solutions

Dans NBC-MOABC, nous utilisons le même codage de solution proposé dans la section 5.2.2.1 pour VNS. Pour la construction d'une solution durant les différentes itérations de l'algorithme NBC-MOABC, la valeur du paramètre *k* est générée d'une façon déterministe ou aléatoire.

5.3.2.2. Fonctions objectif basées sur la connectivité par densité intra et inter-clusters

La validation des résultats de clustering est l'un des défis du problème de clustering. La plupart des critères de validation existant quantifient la variance des clusters et la séparation des clusters en fonction des métriques basées sur la distance. Ainsi, ces indices peuvent échouer dans le cas de clusters de formes arbitraires. Dans ce cas, une bonne solution de clustering est celle dans laquelle la zone de densité la plus élevée dans le voisinage des clusters est moins dense que la zone de densité la plus basse à l'intérieur de chaque cluster. Ainsi, un indice de validation interne doit être défini en utilisant des concepts de densité plutôt que de distance, pour l'évaluation du clustering basé sur la densité. Dans (Moulavi et

al., 2014), un indice de validation de clustering (DBCV), basé sur la connectivité par densité relative entre des paires d'objets, est proposé pour évaluer la qualité du clustering. Dans ce contexte, notre idée est d'utiliser les concepts définis pour l'indice DBCV, pour définir deux objectifs contradictoires comme suit :

Soit $O = \{o_1, \ldots, o_n\}$ un ensemble de données contenant n objets dans l'espace d'attributs $\mathbb{R}^d$. Soit *Dist* une matrice $n \times n$ de paires de distances $d(o_p, o_q)$, où $o_p, o_q \in O$, pour une métrique de distance donnée $d(.,.)$.

Soit $KNN(o, i)$ la distance entre l'objet o et son $i^{ème}$ plus proche voisin.

Soit $C = (\{C_i\}, N)$ $1 \leq i \leq l$ une solution de clustering contenant l clusters et un ensemble N (éventuellement vide) d'outliers, pour laquelle n_i est la taille du $i^{ème}$ cluster et n_N est la cardinalité de l'ensemble N des outliers.

Définition 5.1. Distance centrale (ou noyau) d'un objet (*Core distance of an object*)

La distance centrale (inverse de la densité) d'un objet o appartenant au cluster C_i, relativement à tous les autres objets n_i-1 dans C_i, est définie comme :

$$a_{pts}coredist(o) = \left(\sum_{i=2}^{n_i}(1/KNN(o,i))^d/(n_i-1)\right)^{-1/d} \quad (5.1)$$

Définition 5.2. Distance d'accessibilité mutuelle (*Mutual Reachability Distance*)

La distance d'accessibilité mutuelle entre deux objets o_i et o_j dans O est définie comme :

$$d_{mreach}(o_i, o_j) = Max\{a_{\text{pts}}coredist(o_i), aptscoredist(o_j), d(o_i, o_j)\} \quad (5.2)$$

Définition 5.3. Graphe de distance d'accessibilité mutuelle (*Mutual Reachability Distance Graph*)

Le graphe de distance d'accessibilité mutuelle (*MRD*) est un graphe complet ayant les objets de O comme sommets et la distance d'accessibilité mutuelle entre les paires d'objets respectives comme poids de chaque arrête.

Définition 5.4. MST de Distance d'accessibilité mutuelle (*Mutual Reachability Distance MST*)

Soit G un graphe de distance d'accessibilité mutuelle. L'arbre couvrant de poids minimal (MST) de G est appelé $MST_{MRD.}$

L'arbre couvrant de poids minimal est utilisé pour capturer la forme des clusters sur la base de densité. En utilisant les MSTs des clusters, nous pouvons trouver la zone de densité la plus basse à l'intérieur des clusters (compacité) et la zone de densité la plus élevée entre les clusters (séparation).

Ainsi, nous définissons une première fonction objectif de compacité pour tous les l clusters comme suit :

$$f_1 = \sum_{i=1}^{l} DSC(C_i)|C_i|/|O|l \quad (5.3)$$

Où, $DSC(C_i)$ est la densité éparse du cluster C_i (density sparseness of cluster C_i) définie comme le poids maximal parmi les poids des arrêtes internes (toutes les arrêtes sauf celles

avec un sommet final de degré un) de son MST_{MRD} correspondant et $|C_i|$ est le nombre d'objets dans le cluster C_i.

Une valeur faible de $DSC(C_i)$ signifie que, pour la plus grande partie, les objets de données voisins sont dans le même cluster, tandis qu'une valeur élevée indique que les objets de données voisins sont répartis sur des clusters différents. Par conséquent, la fonction f_1 doit être minimisée.

La séparation entre les clusters peut être quantifiée par la fonction objectif f_2 comme suit :

$$f_2 = \sum_{i=1}^{l} \underset{1 \le j \le l, j \ne i}{Min} \left(DSPC(C_i, C_j) \right) |C_i| / |O| l \qquad (5.4)$$

Où, $DSPC(C_i, C_j)$ est la densité de séparation des clusters C_i et C_j (density separation of clusters C_i and C_j) définie comme la distance *MRD* minimale entre les noeuds internes (tous les objets sauf ceux de degré un) des MST_{MRDs} des clusters C_i et C_j.

$\underset{1 \le j \le l, j \ne i}{Min} (DSPC(C_i, C_j))$ est considérée comme la zone de densité la plus élevée entre le cluster C_i et les autres clusters C_j. Par conséquent, la fonction objectif f_2 doit être maximisée.

Notez que f_1 s'améliore (c'est-à-dire diminue) et f_2 se détériore (c'est-à-dire diminue) lorsque le nombre de clusters diminue et vice versa. Ainsi, en utilisant ces deux fonctions comme critères de performance, un meilleur compromis peut être obtenu lorsque le nombre de clusters augmente ou diminue.

5.3.2.3. Processus de prise de décision

Dans la plupart des problèmes d'optimisation multi-objectif, les fonctions objectif sont conflictuelles et aucune solution qui optimise simultanément chaque objectif n'existe. Le résultat de l'optimisation est un ensemble de solutions non dominées appelées solutions de Pareto (une solution *S* est dite non dominée s'il n'existe pas de solution réalisable *S'* telle que toutes les valeurs des fonctions objectif de *S'* soient meilleures que les valeurs des fonctions objectif correspondantes à *S*). Par conséquent, un processus de décision est nécessaire pour atteindre un équilibre entre les objectifs et choisir une seule solution du front de Pareto. Plusieurs méthodes ont été proposées à cette fin (kasprzak & lewis, 2001). Dans notre approche proposée NBC-MOABC, nous utilisons le même décideur que celui utilisé dans (Armano & Farmani, 2016). Ce décideur, comme l'illustre la figure 5.5 ci-après, est basé sur la distance. Il consiste à sélectionner la solution ayant la distance minimale à une solution idéale appelée point utopique (une solution correspondant au meilleur point de chaque objectif sur le front de Pareto).

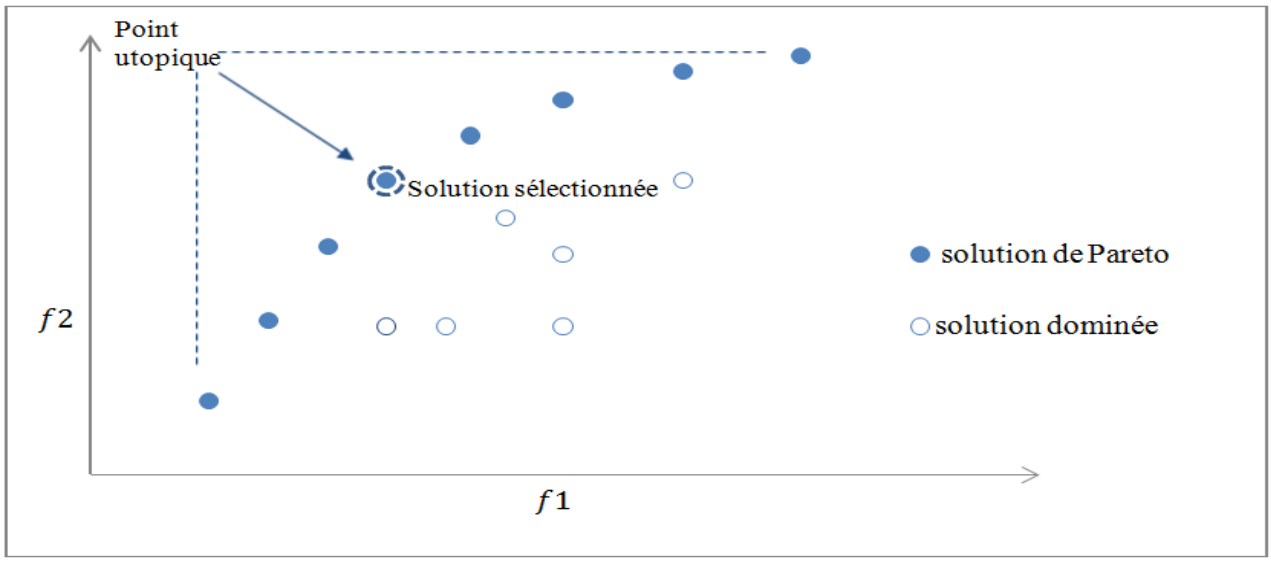


Figure 5.5. Solution finale choisie du front de Pareto comme la plus proche du point utopique (f_1 à minimiser et f_2 à maximiser).

5.3.2.4. Description des étapes principales de l'algorithme NBC-MOABC

L'approche proposée correspond à un algorithme ABC multi-objectif basé sur la dominance de Pareto. Il utilise une archive Pareto de taille fixe pour maintenir les meilleures solutions non dominées trouvées. Par conséquent, l'archive Pareto est mise à jour à la fin de chaque itération de l'algorithme ABC et le point utopique utilisé pour le processus de prise de décision est déterminé. La méthode de mise à jour de l'archive est décrite dans l'algorithme 3 ci-après. La qualité d'un individu (solution de clustering) est mesurée en fonction des valeurs des fonctions objectif retournées et de celles du point utopique. Plus les valeurs des fonctions objectif retournées par un individu sont proches de celles du point utopique, meilleure est la qualité de cet individu. Ainsi, un indicateur de qualité I_q est calculé pour chaque individu (solution) a comme suit :

$$Iq(a) = (f1(a) + f2(a))/2 \quad (5.5)$$

Où $f_1(a)$ et $f_2(a)$ sont les valeurs des fonctions objectif f_1 (compacité) et f_2 (séparation) renvoyées par l'individu a.

Dans les phases des abeilles employées, des abeilles spectatrices et de mutation, une procédure de sélection gourmande est appliquée comme suit. Si la nouvelle solution domine l'ancienne, alors l'ancienne solution est remplacée par la nouvelle, sinon si aucune n'est dominée par l'autre, les valeurs de leurs indicateurs de qualité I_q sont calculées et l'ancienne solution est remplacée si la valeur I_q de la nouvelle est meilleure que celle de l'ancienne par rapport à celle du point utopique.

Dans les phases des abeilles spectatrices et de mutation, une sélection par roulette est utilisée pour choisir un individu en fonction d'une probabilité liée à sa qualité (fitness). La fonction fitness est définie par l'équation suivante :

$$fit(i) = 1/R(i) \quad (5.6)$$

Où, $R(i)$ est le rang de l'individu i dans la population et il est déterminé en fonction de la proximité de sa valeur d'indicateur de qualité avec celle du point utopique.

Le processus de recherche de l'algorithme NBC-MOABC est décrit dans la figure 5.6.

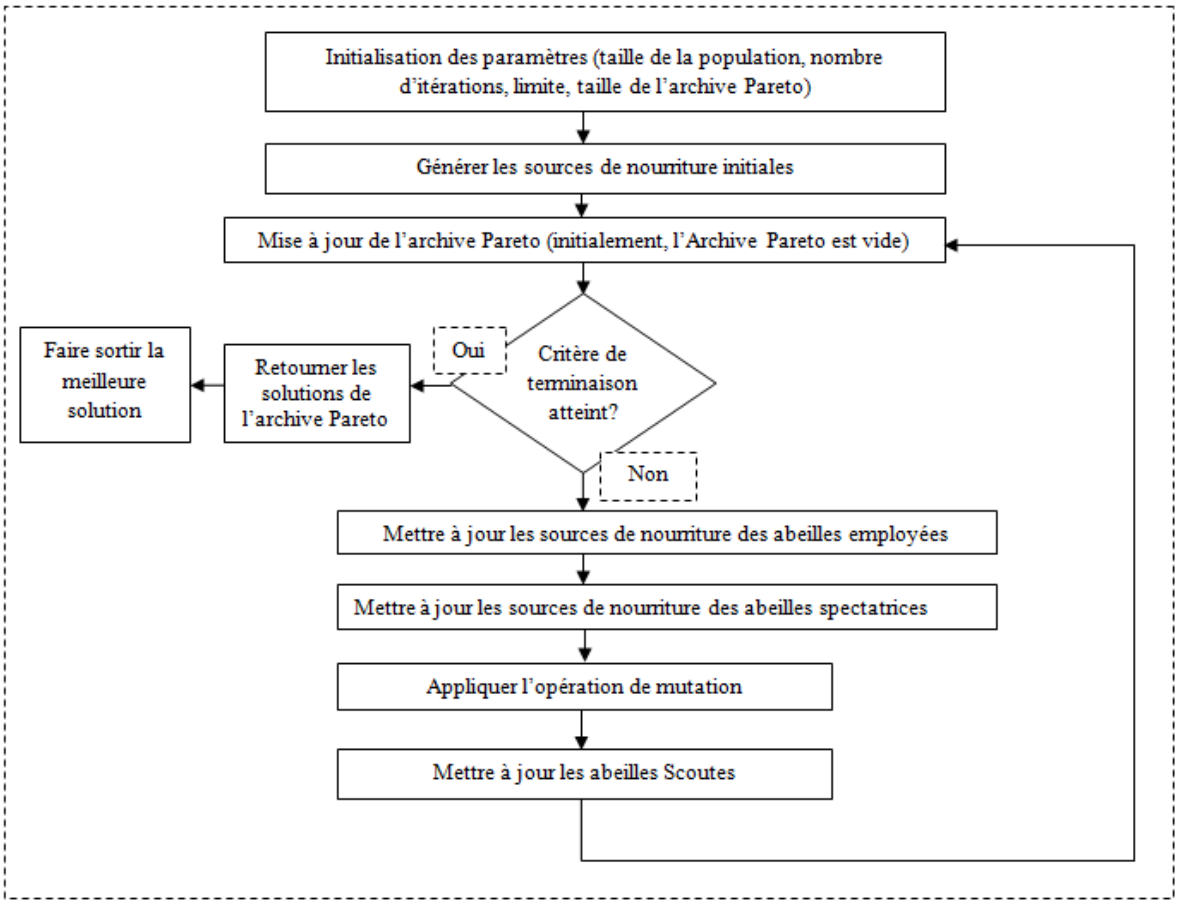


Figure 5.6. L'organigramme de l'algorithme NBC-MOABC.

L'approche NBC-MOABC proposée comprend sept étapes.

Étape 1. Initialisation

La population initiale est composée d'abeilles employées. Selon le schéma de codage et de génération de solutions proposé, les sources initiales (solutions) sont sélectionnées d'une manière déterministe, en utilisant l'algorithme NBC, avec un paramètre associé *k* sélectionné dans l'intervalle [1, 3 × *Spop*] par la formule suivante :

$$val_k(j) = (j + 1) \times 3 - 2 \quad (5.7)$$

Où, *Spop* est la taille de la population, *j* est l'indice de la source de nourriture et $val_k(j)$ est la valeur du paramètre *k* associé à la $j^{ième}$ source.

Cette manière déterministe garantit que les sources de nourriture générées sont diversifiées le plus que possible les unes des autres. Ensuite, les fonctions objectif f_1 et f_2 de chaque source de nourriture sont évaluées. Enfin, les solutions non dominées sont stockées dans l'archive Pareto et le point utopique est déterminé.

Étape 2. Phase des abeilles employées

Dans la phase d'abeilles employées de l'ABC original, une nouvelle source de nourriture est produite au voisinage de la solution courante en modifiant la valeur d'une seule composante du vecteur dimensionnel, ce qui conduit à une faible convergence de l'algorithme (Gong et al., 2016). Pour surmonter ce problème et obtenir de meilleurs résultats de clustering par l'algorithme NBC-MOABC, de nouvelles sources de nourriture sont générées par les abeilles employées pour explorer l'espace de recherche, où la valeur du paramètre *k* d'une solution *j* à la $i^{ème}$ itération est générée dans le voisinage de son ancienne valeur en utilisant la formule suivante :

$$k = val_k(j) + mod((i + 1),\ 3) \tag{5.8}$$

Les nouvelles solutions générées sont comparées aux anciennes en utilisant la procédure de sélection gourmande proposée, après avoir calculé leurs fonctions objectif.

Étape 3. Phase des abeilles spectatrices

Dans la phase des abeilles spectatrices, chaque abeille spectatrice évalue les informations sur le nectar (fitness) tirées de toutes les abeilles employées et choisit une source de nourriture avec une probabilité p_i liée à sa quantité de nectar (fitness) en utilisant la procédure de sélection par roulette (Rahman & Islam, 2014). La probabilité p_i est calculée par la formule suivante :

$$p_i = fit(i)/\sum_{n=1}^{Spop} fit(n) \tag{5.9}$$

Par la suite, chaque abeille spectatrice produit aléatoirement une nouvelle source de nourriture dans le voisinage de celle sélectionnée. Enfin, la procédure de sélection gourmande est appliquée pour choisir la meilleure.

Étape 4. Phase de mutation

Étant donné que l'algorithme NBC peut produire des solutions différentes en utilisant la même valeur de *k*, les positions des composants au sein d'un individu sont importantes. Ainsi, pour une meilleure exploration de l'espace de recherche, une phase de mutation est intégrée dans l'algorithme NBC-MOABC. Dans cette phase, la mutation est appliquée sur quelques sources de nourriture (individus) sélectionnées dans la population. La sélection par roulette est utilisée pour sélectionner un individu pour la mutation. La mutation consiste à échanger les valeurs de deux points représentatifs choisis aléatoirement. Pour mettre à jour les sources de nourriture, la procédure de sélection gourmande est appliquée. La figure 5.7 montre comment la qualité de clustering est améliorée lors de l'application de l'opérateur de mutation (les individus b et c sont les résultats de la mutation effectuée sur l'individu a).

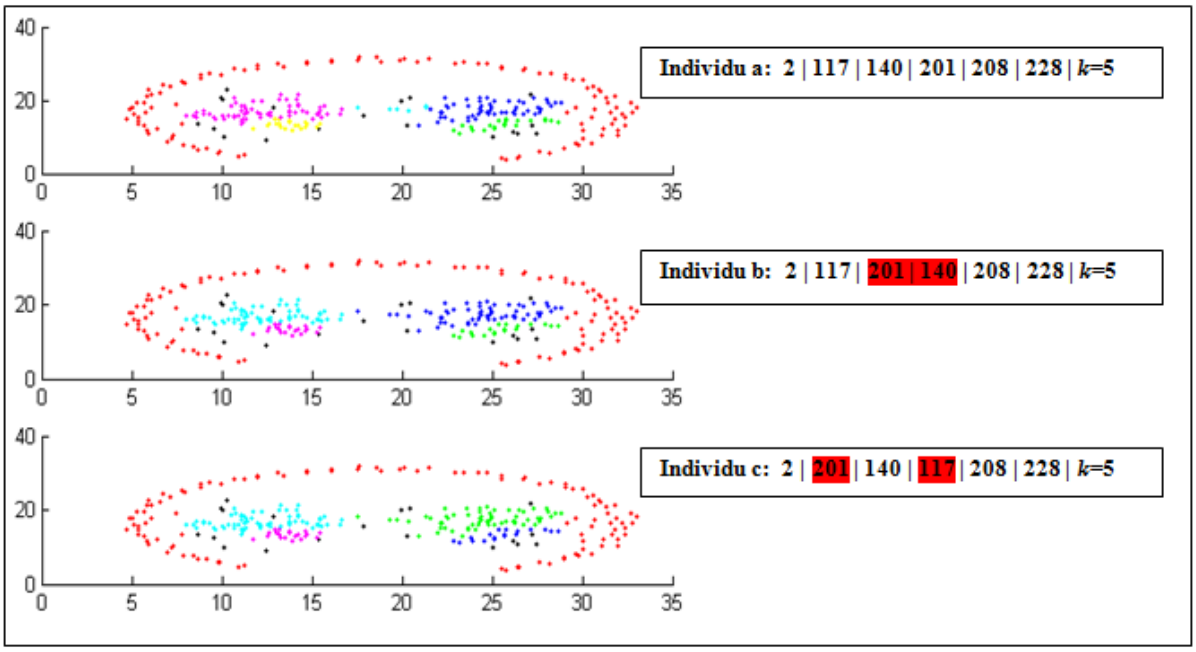


Figure 5.7. Exemple de solution de clustering avant et après l'opération de mutation.

Étape 5. Phase des abeilles scoutes

Si une source de nourriture ne peut pas être améliorée après un nombre prédéfini de cycles (appelé *limite*), alors la source de nourriture est abandonnée. Pour diversifier la recherche, les abeilles scoutes remplacent les sources de nourriture abandonnées par les abeilles employées par d'autres générées d'une façon déterministe à partir des sources de nourriture épuisées. La valeur du paramètre *k* utilisé pour produire une nouvelle solution d'un individu *j* est générée à l'aide de la formule suivante :

$$k = val_k(j) + 3 \times Spop/2 \qquad (5.10)$$

Étape 6. Phase de mise à jour de l'archive Pareto

Dans cette étape, l'archive Pareto est mise à jour en incluant les dernières meilleures solutions non dominées trouvées et en supprimant celles dominées (voir l'algorithme 3). Par conséquent, le point utopique est mis à jour et la qualité (fitness) des solutions est calculée selon l'équation (5.6).

Les étapes 2, 3, 4, 5 et 6 sont répétées jusqu'à ce que le nombre maximal d'itérations soit atteint.

Étape 7. Phase de prise de décision

Dans cette étape, les solutions contenues dans l'archive Pareto sont examinées et la meilleure (la solution la plus proche du point utopique, c'est-à-dire celle ayant le meilleur classement de Pareto par rapport à la valeur de *Iq* du point utopique) est sélectionnée comme solution finale.

Le pseudo-code détaillé de l'algorithme NBC-MOABC est donné par l'algorithme 2 ci-après.

Algorithme 2. Pseudo code de l'algorithme NBC-MOABC.

// Initialisation des paramètres (taille de la population, nombre d'itérations, limite, taille de l'archive Pareto)

//phase d'initialisation
Pour *i = 0* à taille de la population **faire**
Initialiser une source de nourriture (utiliser l'équation (5.7) et appliquer l'algorithme NBC)
Évaluer les fonctions objectif f_1 et f_2 pour chaque source de nourriture
FinPour

// *phase de mise à jour de l'archive Pareto*
Mettre à jour l'archive Pareto (appliquer l'algorithme 3) // *l'archive de Pareto est initialement vide*
Déterminer le point utopique
Mémoriser la meilleure source de nourriture de l'archive Pareto

Répeter

// *Phase* des abeilles employées
Pour *i = 0* à nombre maximum d'abeilles employées **faire**
Générer une nouvelle solution (utiliser l'équation (5.8) et appliquer l'algorithme NBC)
Évaluer l'indicateur de qualité pour la nouvelle solution générée
Appliquer une sélection gourmande.
FinPour

//Calcul des préférences pour les sources de nourriture actuelles
Calculer les probabilités de chaque solution en se basant sur leurs préférences par les abeilles spectatrices en utilisant Eq. (5.9)

// *Phase des abeilles spectatrices*
Pour *i = 0* à nombre maximum d'abeilles spectatrices **faire**
Sélectionner une solution selon la sélection par roulette
Générer une nouvelle solution au hasard dans le voisinage de celle sélectionnée actuellement (appliquer l'algorithme NBC)
Calculer l'indicateur de qualité de la nouvelle solution générée
Appliquer une procédure de sélection gourmande
FinPour

// *Phase de mutation*
Pour *i* = 0 à nombre maximum de sources de nourriture sélectionnées **faire**
Sélectionner un individu de la population actuelle pour la mutation en utilisant la sélection par roulette
Sélectionner aléatoirement deux composants de l'individu sélectionné
Appliquer l'opération de mutation en échangeant les composants sélectionnés
Appliquer une sélection gourmande (entre l'individu sélectionné et son mutant)
FinPour

// *Phase des abeilles scoutes*
Si une source de nourriture ne peut pas être améliorée sur un nombre limité de cycles **alors**
Réinitialisez de manière déterministe la source de nourriture épuisée en utilisant une nouvelle valeur du paramètre *k* selon l'Eq. (5.10)

// *Phase de mise à jour de l'archive Pareto*
Mettre à jour l'archive Pareto (appliquer l'algorithme 3)
Déterminer le point utopique
Mémoriser la meilleure source de nourriture de l'archive Pareto
Jusqu'à (nombre d'itérations atteint)

// *Phase de prise de décision*
Sortie : la meilleure solution de l'archive Pareto

Algorithme 3. Algorithme de mise à jour de l'archive Pareto.

1. Sélectionner les solutions non dominées dans la population comme P'
2. Fusionner l'archive Pareto A avec P' dans A'
3. Supprimer les solutions dominées de A'
4. Déterminer le point utopique et évaluer la qualité (fitness) de chaque solution dans A'
5. **Tantque** |A'| > ArchSize **faire** // *ArchSize est une taille maximale fixe de l'archive Pareto*
   Déterminer x' la mauvaise solution ayant la valeur minimale de fitness ((x' = min (fit))
   Supprimer x' de A'
6. **FinTque**
7. A = A'
8. **Retourner** A

5.3.2.5. Résultats expérimentaux

Afin d'analyser les performances de l'algorithme NBC-MOABC, nous avons tout d'abord effectué un réglage préliminaire des paramètres. Ensuite, nous l'avons comparé à sept algorithmes de clustering bien connus dans la litterature. Les ensembles de données utilisés dans les expériences sont pris des référentiels de clustering de (Franti, 2015). Ce sont des ensembles de données à deux dimensions ayant des clusters de formes arbitraires. Nous avons utilisé comme critères de performance, l'indice de validité externe Rand (RI) (Rand, 1971) et sa forme ajustée (ARI) (Hubert & Arabie, 1985). Ces indices utilisent la bonne solution de clustering (classification existante) pour évaluer la solution de clustering obtenue. Plus la valeur de l'indice RI ou de l'indice ARI est proche de 1, meilleure est la solution de clustering obtenue. Pour prouver l'efficacité des fonctions objectif définies, nous avons évalué l'algorithme NBC-MOABC sous sa forme multi-objectif et sous sa forme mono-objectif (appelée NBC-ABC) en utilisant RI comme critère de qualité (fitness). En plus, un ensemble d'expériences est mené sur l'algorithme NBC-MOABC, en utilisant l'indice de validité interne Silhouette (Rousseeuw, 1997) comme critère de fitness. L'indice silhouette (SI) indique de bons résultats lorsque sa valeur est proche de 1.

5.3.2.5.1. Réglage des paramètres

Pour déterminer le réglage approprié des paramètres de l'algorithme NBC-MOABC, un ensemble étendu d'expériences a été effectué sur des ensembles de données sélectionnés, en utilisant des combinaisons différentes de valeurs des paramètres (taille de la population, nombre d'itérations et le paramètre *limite*). Le tableau 5.1 montre les valeurs sélectionnées des paramètres de NBC-MOABC.

Tableau 5.1. Paramètres de l'algorithme NBC-MOABC.

Paramètre	Description	Valeur séléctionnée
Spop	Taille de la population (égale au nombre d'abeilles employées)	8
Niter	Nombre maximum d'itérations (critère de terminaison)	10
Limite	Nombre de cycles après lequel une solution est remplacée par une nouvelle	5
ArchSize	Taille fixe de l'archive	8

5.3.2.5.2. Comparaisons expérimentales de NBC-MOABC avec d'autres algorithmes

La performance de l'approche de clustering NBC-MOABC (Boudane & Berrichi, 2022) est comparée à celle de sept algorithmes de clustering bien connus ou récents, à savoir : *k*-means (Hartigan & Wong, 1979), single-linkage (Voorhees, 1985), DBSCAN (Ester et al., 1996), NC-closures (Inkaya & özdemirel, 2013), MCPSO présenté dans (Armano & Farmani, 2016), VNS que nous avons proposé et présenté dans la section 5.2.2 de ce chapitre (Boudane & Berrichi, 2017) et l'algorithme NBC (Zhou et al., 2005). La distance euclidienne est utilisée comme mesure de dissimilarité, pour tous les ensembles de données et algorithmes.

Comme nous l'avons déjà mentionné précédemment, nous évaluons la forme mono-objectif de l'algorithme NBC-MOABC qui est appelé dans ce qui suit NBC-ABC. Comme critère de fitness, d'une part l'indice RI est utilisé pour prouver l'efficacité de l'algorithme ABC appliqué dans l'exploration de l'espace de recherche et d'autre part, l'indice SI est utilisé pour prouver l'efficacité et l'importance des fonctions objectif proposées (la forme multi-objectif) dans l'évaluation des solutions.

Les expériences sur *k*-means et single-linkage ont été réalisées sur chaque ensemble de données par une variation du nombre de clusters dans l'intervalle 2% et 10% du nombre de points dans l'ensemble de données. DBSCAN a été exécuté avec toutes les valeurs possibles du paramètre *MinPts* (entre 1 et 15). Les paramètres des algorithmes MCPSO et VNS sont les mêmes que ceux utilisés dans (Armano et Farmani, 2016) et (Boudane et Berrichi, 2017) respectivement. L'algorithme NBC a été exécuté en utilisant les mêmes valeurs du paramètre *k* des solutions générées par l'algorithme NBC-ABC proposé lors de l'utilisation de l'indice RI comme critère de fitness. Pour VNS, les expériences ont été menées en utilisant 30 itérations entre deux améliorations comme critère de terminaison. Nous avons, aussi, utilisé l'indice de validité externe RI comme critère de fitness dans VNS juste pour juger la qualité de la recherche par rapport à NBC-ABC.

Les algorithmes MCPSO, NBC, VNS et NBC-MOABC ont été exécutés 40 fois indépendamment.

Les tableaux 5.2, 5.3, 5.4 et les figures 5.8, 5.9, 5.10, 5.11 et 5.12 illustrent les résultats.

Tableau 5.2. Valeur moyenne et écart-type de RI mesurés sur les sorties des algorithmes.

Datasets	taille	# Clusters	NBC-MOABC	K-Means	Single-linkage	DBSCAN	NC-closures	MCPSO	NBC-ABC en utilisant RI comme critère de fitness	NBC-ABC en utilisant SI comme critère de fitness	VNS	NBC
Pathbased	300	3	0.79 ± 0.010	0.84 ± 0.005	0.85 ± 0.037	0.90 ± 0.005	0.89 ± 0.007	0.94 ± 0.006	0.94 ± 0.000	0.80 ± 0.012	0.83 ± 0.072	0.94 ± 0.002
Spiral	312	3	1.00 ± 0.000	0.79 ± 0.084	0.87 ± 0.002	0.93 ± 0.020	0.81 ± 0.007	0.89 ± 0.003	1.00 ± 0.000	0.51 ± 0.055	0.97 ± 0.054	1.00 ± 0.000
Compound	399	6	0.95 ± 0.002	0.81 ± 0.003	0.75 ± 0.052	0.80 ± 0.016	0.79 ± 0.014	0.92 ± 0.075	0.98 ± 0.000	0.73 ± 0.004	0.95 ± 0.017	0.97 ± 0.010
Jain	373	2	0.98 ± 0.002	0.89 ± 0.012	0.75 ± 0.004	0.96 ± 0.002	0.90 ± 0.006	0.95 ± 0.008	0.98 ± 0.001	0.83 ± 0.157	0.96 ± 0.001	0.87 ± 0.173
Flame	240	2	0.95 ± 0.009	0.83 ± 0.006	0.85 ± 0.003	0.89 ± 0.051	0.87 ± 0.042	0.91 ± 0.021	0.97 ± 0.001	0.95 ± 0.010	0.95 ± 0.007	0.96 ± 0.015
Aggregation	788	7	0.99 ± 0.002	0.94 ± 0.003	0.89 ± 0.014	0.88 ± 0.023	0.85 ± 0.017	0.97 ± 0.028	0.99 ± 0.000	0.96 ± 0.021	0.99 ± 0.002	0.99 ± 0.000
R15	600	15	0.99 ± 0.003	0.91 ± 0.037	0.87 ± 0.009	0.89 ± 0.014	0.90 ± 0.007	0.94 ± 0.042	0.99 ± 0.000	0.87 ± 0.104	0.98 ± 0.015	0.99 ± 0.000

Tableau 5.3. Valeur moyenne et écart-type de ARI mesurés sur les sorties des algorithmes.

Datasets	taille	# Clusters	NBC-MOABC	K-Means	Single-linkage	DBSCAN	NC-closures	MCPSO	NBC-ABC en utilisant RI comme critère de fitness	NBC-ABC en utilisant SI comme critère de fitness	VNS	NBC
Pathbased	300	3	0.54 ± 0.022	0.71 ± 0.002	0.72 ± 0.024	0.78 ± 0.002	0.76 ± 0.005	0.81 ± 0.003	0.88 ± 0.000	0.55 ± 0.026	0.64 ± 0.151	0.87 ± 0.004
Spiral	312	3	1.00 ± 0.000	0.70 ± 0.012	0.79 ± 0.003	0.86 ± 0.041	0.74 ± 0.002	0.82 ± 0.001	1.00 ± 0.000	0.46 ± 0.083	0.94 ± 0.126	1.00 ± 0.000
Compound	399	6	0.87 ± 0.006	0.75 ± 0.022	0.68 ± 0.023	0.74 ± 0.016	0.70 ± 0.009	0.86 ± 0.011	0.95 ± 0.000	0.46 ± 0.005	0.89 ± 0.042	0.93 ± 0.027
Jain	373	2	0.96 ± 0.005	0.41 ± 0.010	0.37 ± 0.002	0.69 ± 0.002	0.50 ± 0.014	0.66 ± 0.006	0.97 ± 0.003	0.66 ± 0.313	0.93 ± 0.047	0.91 ± 0.067
Flame	240	2	0.90 ± 0.020	0.43 ± 0.005	0.47 ± 0.004	0.56 ± 0.031	0.51 ± 0.022	0.73 ± 0.051	0.94 ± 0.005	0.91 ± 0.021	0.91 ± 0.014	0.93 ± 0.032
Aggregation	788	7	0.98 ± 0.004	0.81 ± 0.011	0.80 ± 0.010	0.76 ± 0.067	0.72 ± 0.027	0.87 ± 0.032	0.99 ± 0.000	0.92 ± 0.046	0.98 ± 0.006	0.99 ± 0.001
R15	600	15	0.91 ± 0.030	0.85 ± 0.074	0.77 ± 0.008	0.80 ± 0.027	0.83 ± 0.006	0.88 ± 0.032	0.98 ± 0.005	0.50 ± 0.210	0.89 ± 0.123	0.98 ± 0.005

Comme l'illustre les tableaux 5.2 et 5.3, l'approche proposée (NBC-MOABC) fonctionne mieux que les algorithmes *k*-means, signle-linkage, DBSCAN, NC-closures, MCPSO et VNS en termes de précision et de stabilité dans presque tous les ensembles de données, en particulier dans le cas des ensembles de données Jain, Spiral et Agrégation. Par exemple, dans le cas de l'ensemble de données Spiral, l'indice ARI a une valeur égale à 1, ce qui signifie que les clusters obtenus sont identiques aux vrais (ceux d'une préexistante classification).

Par rapport à l'approche VNS qui utilisait l'indice RI comme critère de fitness et le même schéma de codage que l'approche NBC-MOABC proposée, l'algorithme NBC-ABC utilisant l'indice RI comme critère de fitness montre l'influence significative de la stratégie de recherche déterministe (les valeurs du paramètre *k* des individus sont générés de manière déterministe et stochastique dans l'algorithme NBC-ABC et juste aléatoirement dans l'algorithme VNS) et le mécanisme de mutation intégré, dans l'exploration de l'espace de recherche.

De plus, les résultats de NBC-MOABC sont similaires ou proches de ceux de NBC-ABC avec RI comme critère de fitness, mais meilleurs que NBC-ABC lors de l'utilisation de SI comme critère de fitness (voir les figures 5.11 et 5.12), ce qui signifie que les fonctions objectif proposées sont efficaces pour le clustering des données ayant des formes arbitraires. Les figures 5.8 et 5.9 visualisent les résultats de clustering.

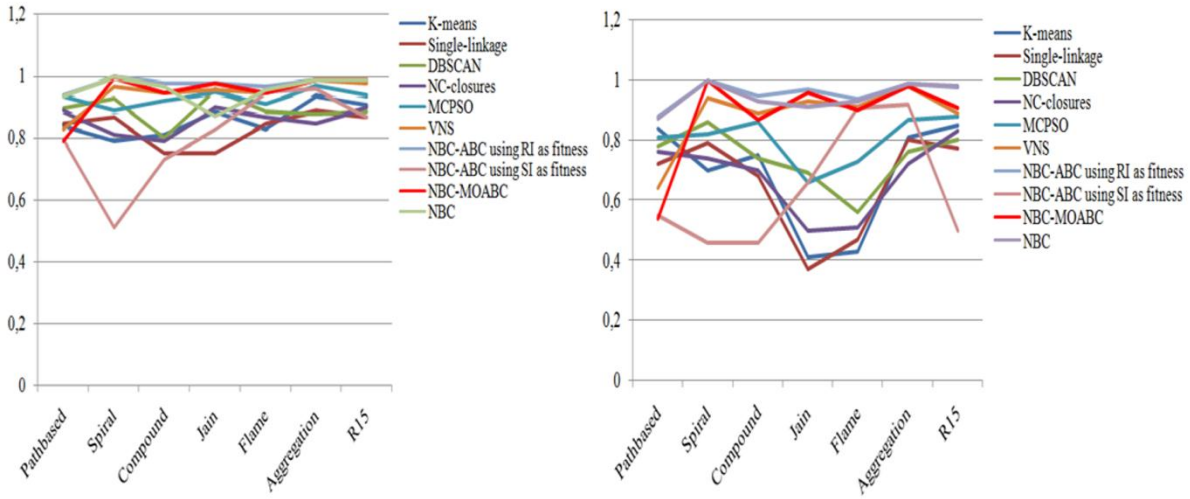


Figure 5.8. Comparaison de la précision moyenne de dix algorithmes sur sept ensembles de données en utilisant RI (côté gauche) et ARI (côté droit) comme critère de performance.

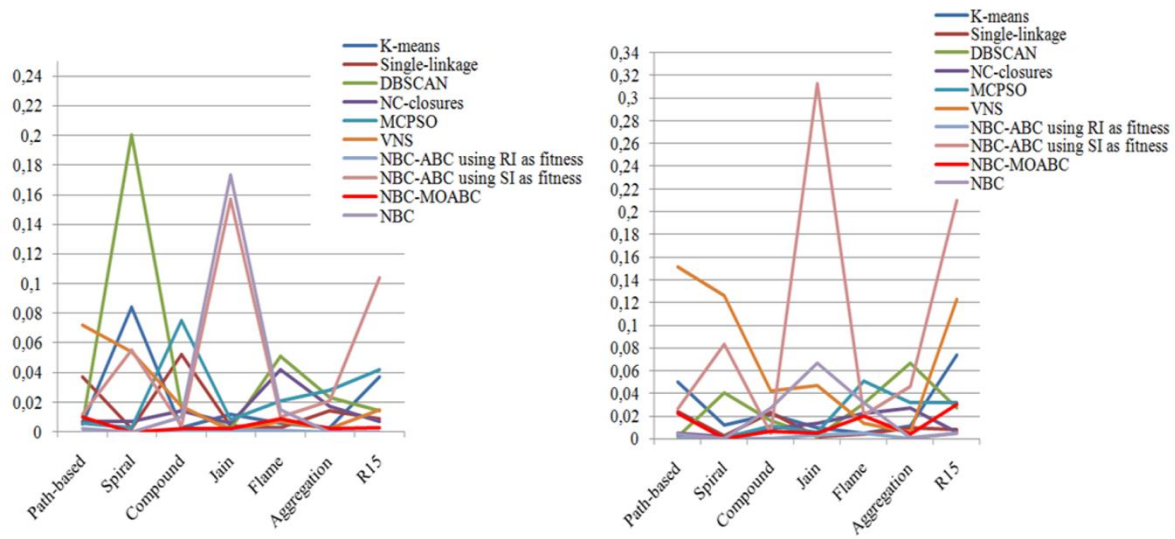


Figure 5.9. Comparaison de l'écart-type de dix algorithmes sur sept ensembles de données en utilisant RI (côté gauche) et ARI (côté droit) comme critère de performance.

A partir des figures 5.8 et 5.9, nous pouvons voir que l'approche NBC-MOABC produit de meilleurs résultats et elle est plus robuste et efficace par rapport à d'autres algorithmes.

Bien que nous ayons exécuté l'algorithme NBC avec les meilleures valeurs du paramètre *k* généré par l'algorithme NBC-ABC lors de l'utilisation de RI comme critère de fitness, nous pouvons remarquer que NBC-ABC obtient de meilleurs résultats que NBC. Cela est dû au fait que la même valeur du paramètre *k* peut générer des solutions différentes selon le choix et l'ordre d'apparition des points représentatifs des clusters dans le codage de la solution.

Comme le montrent les figures 5.11 et 5.12 ci-après, l'algorithme NBC-ABC avec RI comme critère de fitness obtient des valeurs optimales et une très petite déviation, en comparaison avec les meilleurs et les mauvais résultats. Cette déviation peut être interprétée comme une conséquence de la génération aléatoire des points représentatifs des clusters et de la phase de mutation intégrée. Le tableau 5.4, ci-après, illustre cette différence par une présentation des valeurs de l'indice RI obtenues par les algorithmes NBC-ABC avec RI comme critère de fitness et NBC utilisant la même valeur générée du paramètre *k*.

Tableau 5.4. Valeurs de RI obtenues par NBC-ABC avec RI comme critère de fitness et l'algorithme NBC en utilisant les mêmes valeurs générées du paramètre *k*.

Datasets	Valeur du paramètre *k*	NBC-ABC utilisant RI comme critère de fitness	NBC
Pathbased	6	0.9494	0.9452 0.9494
Spiral	4	1	1
Compound	7	0.9833 0.9821 0.9824	0.9833 0.9797 0.9824 0.9821 0.9818 0.9811 0.9505 0.9515 0.9806 0.9514 0.9732 0.9809 0.9811 0.9524
Jain	20 21 18 19	0.9867 0.9867 0.9814 0.9814	0.609 0.9867 0.9814 0.9867 0.9814 0.9761 0.9657 0.609 0.9814 0.9657 0.9761 0.9814
Flame	17 16 27 18 20 7	0.9721 0.9694 0.9723 0.9667 0.9694 0.9642	0.9721 0.9222 0.9694 0.9642 0.9507 0.9614 0.9642
Aggregation	19 27 28 18	0.9985 0.9971 0.9979 0.9979	0.9985 0.9965 0.9975 0.9964 0.9971 0.9964 0.9979
R15	28 29 30	0.9965 0.9975 0.9971 0.9979 0.9993 0.9989 0.9976 0.998 0.9984 0.9972	0.9961 0.9965 0.9962 0.9967 0.9979 0.9971 0.9975 0.998 0.9984 0.9989 0.9976 0.9993

Dans le tableau 5.4, les différentes valeurs de RI des meilleures solutions obtenues par les algorithmes NBC-ABC et NBC sont présentées. Comme prévu, la même valeur de *k* génère des valeurs différentes de RI.

Nous pouvons voir, de la figure 5.10 ci-après, que l'opération de mutation peut améliorer la valeur de l'indice RI.

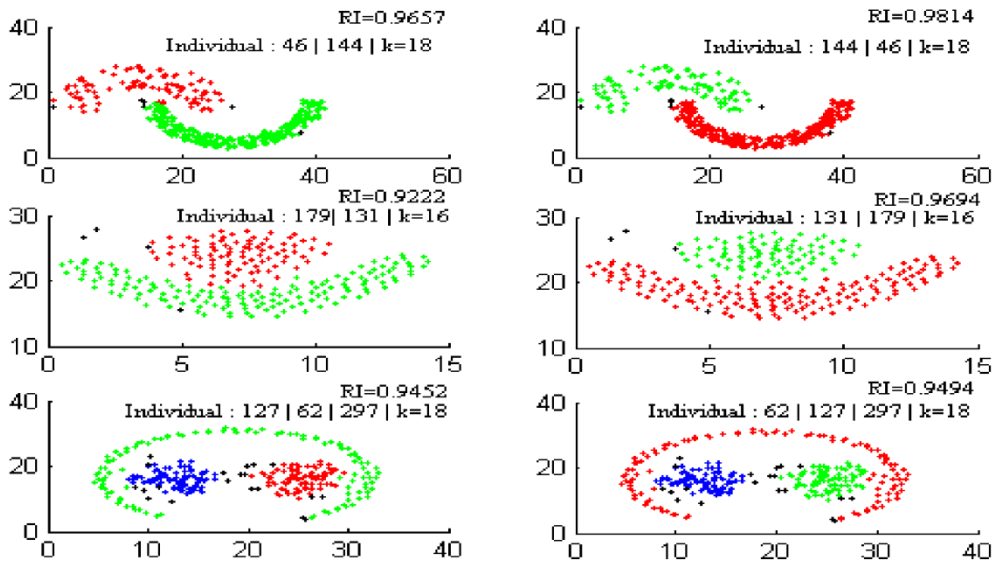


Figure 5.10. Visualisation de quelques solutions de clustering obtenues avant et après mutation de leurs composants. Ensembles de données de haut en bas : Jain, Flame, Pathbased.

Les figures 5.11 et 5.12 montrent, respectivement, les meilleurs et les mauvais résultats de clustering obtenus par les algorithmes NBC, NBC-MOABC et NBC-ABC sur 40 exécutions indépendantes. L'algorithme NBC-ABC est exécuté en utilisant une fois l'indice de Rand et une autre fois l'indice silhouette comme critère de fitness.

Comme on peut observer, les résultats de NBC-ABC lors de l'utilisation de l'indice RI comme critère de fitness sont meilleurs que ceux de NBC, en particulier dans le cas des ensembles de données Jain et Flame. Les résultats de l'algorithme NBC-MOABC sont significativement meilleurs que ceux du NBC-ABC lors de l'utilisation de l'indice silhouette comme critère de fitness (NBC-MOABC a un très petit écart par rapport aux meilleurs et mauvais résultats), en particulier dans le cas des ensembles de données Jain et Spiral. Ce qui prouve la capacité d'évaluation des fonctions objectif définies pour le clustering multi-objectif des ensembles de données ayant des clusters de formes arbitraires et bien séparés.

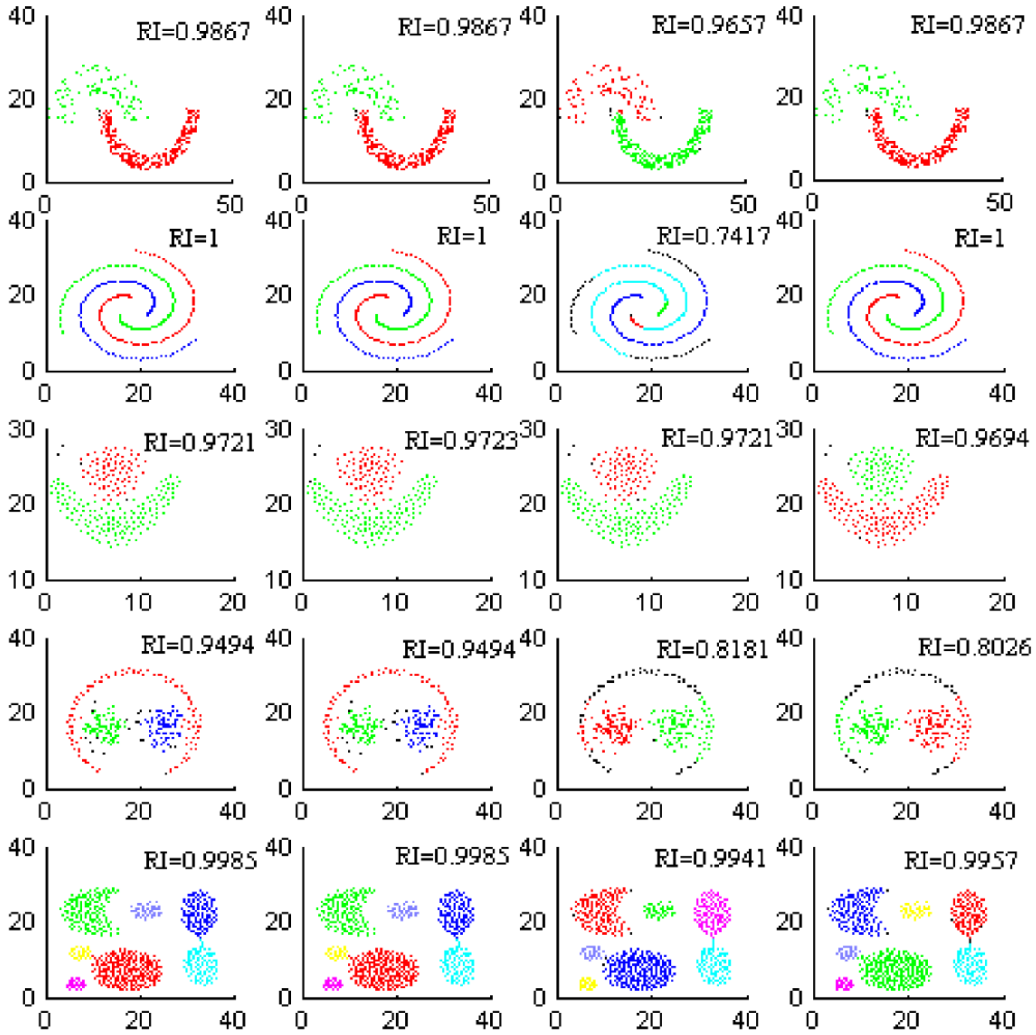


Figure 5.11. Visualisation des meilleures solutions de clustering obtenues par les algorithmes, de gauche à droite, NBC, NBC-ABC avec RI comme critère de fitness, NBC-ABC avec SI comme critère de fitness et NBC-MOABC. Ensembles de données de haut en bas : Jain, Spiral, Flame, Pathbased, Aggregation. (Dans tous les cas, les points noirs sont des outliers).

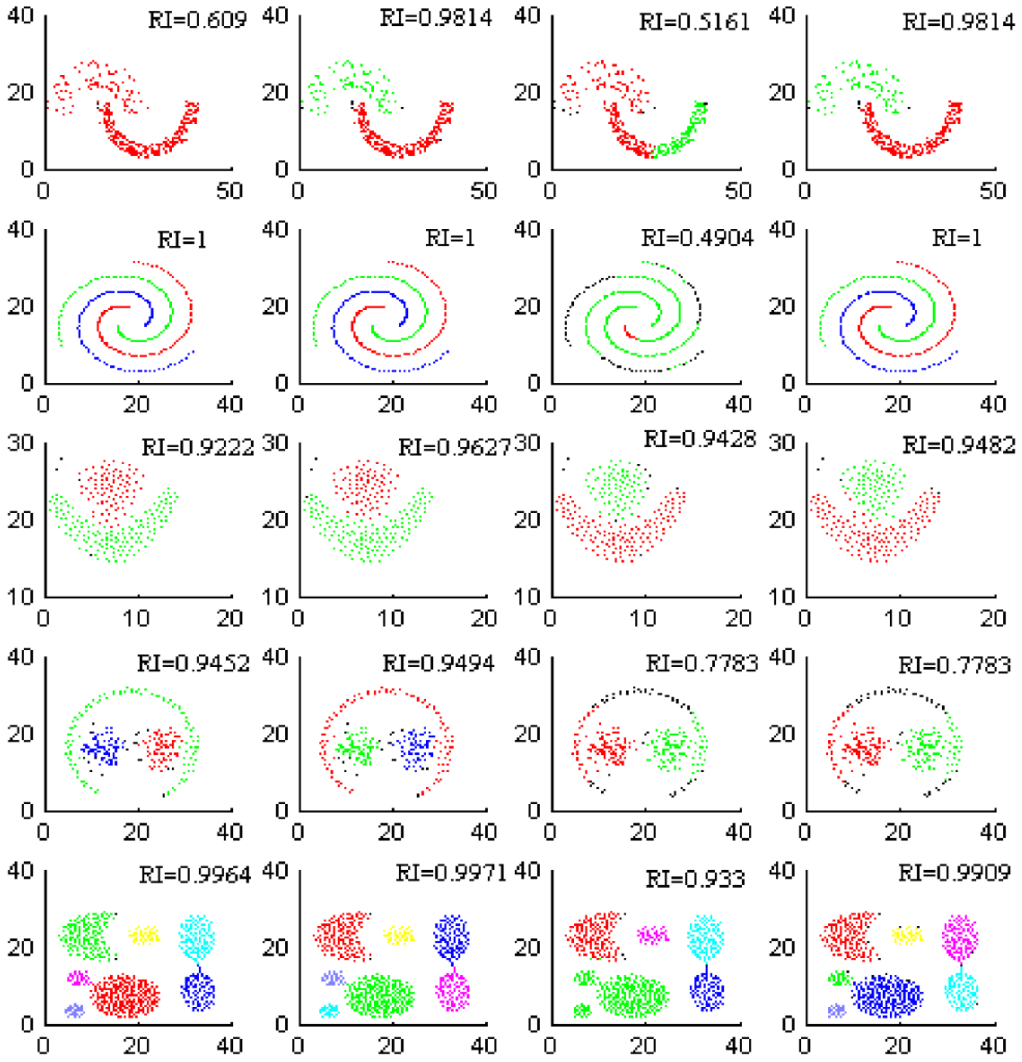


Figure 5.12. Visualisation des mauvaises solutions de clustering obtenues par les algorithmes, de gauche à droite, NBC, NBC-ABC avec RI comme critère de fitness, NBC-ABC avec SI comme critère de fitness et NBC-MOABC. Ensembles de données de haut en bas : Jain, Spiral, Flame, Pathbased, Aggregation. (Dans tous les cas, les points noirs sont des outliers).

5.3.3. Clustering multi-objectif par combinaison des algorithmes ABC et DBSCAN

Dans cette section, nous proposons une autre approche de clustering automatique à base de densité en utilisant un algorithme de colonie d'abeilles artificielles multi-objectif combiné avec DBSCAN (*Density-based clustering approach using Multi-objective ABC (DCMABC) algorithm*). À la différence de l'algorithme NBC-MOABC, l'approche DCMABC que nous proposons est basée sur le paradigme de l'ABC et l'algorithme de clustering DBSCAN proposé dans (Ester et al., 1996). DBSCAN est l'un des algorithmes basés sur la densité les plus connus. Nous rappelons qu'il utilise deux paramètres définis par l'utilisateur, le rayon de voisinage (*Eps*) et le nombre minimum d'objets dans un voisinage (*Minpts*). Cependant, premièrement, comme la plupart des méthodes de clustering basées sur la densité, DBSCAN ne peut pas faire face à la difficulté de recherche des clusters avec des densités très variables (Ertöz et al., 2003; Zhu et al., 2016). En effet, il est difficile de définir les valeurs de ses paramètres d'entrée qui sont fixes. Deuxièmement, le nombre attendu de clusters ne peut pas être vérifié par l'utilisateur comme dans le cas de l'algorithme de clustering par partitionnement bien connu *k*-means (Hartigan & Wong, 1979).

Certaines méthodes d'optimisation ont déjà été appliquées pour améliorer les performances de DBSCAN. Par exemple, dans (Sabau, 2012), l'algorithme DBSCAN (Ester et al., 1996) a été combiné avec un algorithme génétique pour surmonter le problème de réglage des valeurs des paramètres posé par l'algorithme DBSCAN. Des opérateurs de croisement et de mutation simples ont été utilisés avec un opérateur de réadaptation qui garantit des résultats valides. Cependant, la population initiale est générée d'une manière aléatoire, le processus de réadaptation prend plus de temps pour changer et valider les individus générés, et l'indice de validité interne utilisé comme fonction de fitness n'est pas approprié pour le clustering basé sur la densité, ce qui nécessite plusieurs itérations pour parvenir à une solution proche de l'optimum. Dans (Karami & Johansson, 2014), les auteurs ont appliqué l'évolution différentielle (DE) pour optimiser les paramètres d'entrée de DBSCAN. Cependant, cette approche utilise une fonction de fitness qui n'est applicable qu'en apprentissage supervisé nécessitant des connaissances sur l'ensemble de données cible. Récemment, dans (Guan et al., 2019), une nouvelle approche appelée PODCC a été proposée pour le clustering basé sur la densité et la classification. PSO a été utilisé, dans l'approche PODCC, comme un outil de paramétrage pour DBSCAN. PODCC a utilisé certaines fonctions objectif qui lui ont permis de fonctionner en termes d'apprentissage supervisé et non supervisé. Cependant, les fonctions de fitness proposées pour l'apprentissage sont définies sur la base d'indices de validation de clustering qui peuvent ne pas bien fonctionner dans le cas de clusters de formes arbitraires.

Mentionnons que ces méthodes sont utilisées comme outils de réglage des paramètres pour l'algorithme DBSCAN et que les fonctions de fitness utilisées sont basées sur des indices de validité internes traditionnels. Ainsi, ces méthodes peuvent échouer en cas de clustering basé sur la densité et d'ensembles de données ayant des clusters de formes arbitraires (Xie et al., 2019; Rojas-Thomas et al., 2017; Liu et al., 2013; Arbelaitz et al., 2013). Par conséquent, la difficulté de détection des clusters avec des densités très variables est toujours posée.

Ainsi, nous proposons l'approche DCMABC pour surpasser les inconvénients de l'algorithme DBSCAN. Afin de surmonter le premier problème posé par DBSCAN (la difficulté de recherche des clusters avec des densités très variables), nous utilisons l'algorithme ABC afin d'identifier les meilleurs paramètres pour un clustering basé sur la densité à travers une recherche globale dans tout l'espace des paramètres. Dans notre approche, ABC est utilisé comme un outil de paramétrage, pour le processus d'extension de clusters de DBSCAN (pour chaque cluster au lieu de tous les clusters). Le fait que la plupart des indices de validation internes sont définis pour les méthodes basées sur les centroïdes et peuvent ne pas fonctionner correctement pour le clustering basé sur la densité et les clusters de formes arbitraires, nous proposons plusieurs fonctions objectif permettant de trouver une solution suffisamment bonne. Une parmi ces fonctions consiste à éviter le second problème issu de DBSCAN (le nombre attendu de clusters ne peut pas être vérifié par l'utilisateur) en contrôlant le nombre de clusters. Ce dernier ne peut pas être contrôlé par l'utilisateur dans le cas de l'algorithme DBSCAN.

Notons que DCMABC diffère de NBC-MOABC dans le codage et la génération de solutions ainsi que les fonctions objectif utilisées.

5.3.3.1. Codage et génération des solutions

Afin de traiter le problème de clusters avec des densités différentes, posé par l'algorithme DBSCAN, nous proposons d'utiliser une paire de valeurs différentes des paramètres *Eps* et *Minpts*, pour étendre chaque cluster. Un individu, qui signifie une solution possible, est une partition de l'ensemble de données en clusters représentés par des objets noyaux associés à des paires de valeurs de paramètres. Les individus sont générés durant les différentes itérations d'une manière déterministe ou aléatoire, en utilisant la fonction d'extension d'un cluster (Expand-cluster) de l'algorithme DBSCAN. Comme l'illustre la figure 5.13 nous représentons un individu (solution) par un vecteur de $3N$ éléments, où N est le nombre d'objets noyaux sélectionnés pour représenter les clusters. Une solution de clustering est générée comme suit :

Tout d'abord, une paire de valeurs des paramètres (*Eps, Minpts*) est sélectionnée et un objet noyau relativement à la paire de valeurs des paramètres sélectionnée est choisi parmi les objets non encore affectés, comme représentatif d'un cluster. Ensuite, la fonction Expand-cluster de l'algorithme DBSCAN est appliquée pour construire un cluster. Ce processus est répété récursivement à partir des objets n'appartenant à aucun cluster découvert, jusqu'à ce que tous les clusters soient découverts. Ainsi, un individu est représenté par le vecteur composé des premiers objets noyaux et de leurs paires des valeurs des paramètres associées qui sont utilisées pour construire les clusters par la fonction Expand-cluster.

Ce schéma de codage n'a pas besoin d'avoir le nombre de clusters comme paramètre défini à l'avance par l'utilisateur. Le bon nombre de clusters et les meilleurs objets représentatifs peuvent être trouvés automatiquement, en exploitant l'espace de recherche avec différentes paires des valeurs des paramètres (*Eps, Minpts*) durant les différentes itérations de notre approche DCMABC proposée.

Rep 1	Eps 1	MinPts 1	Rep 2	Eps 2	MinPts 2	Rep 3	Eps 3	MinPts 3
22	1	3	14	0.7	4	12	0.4	5

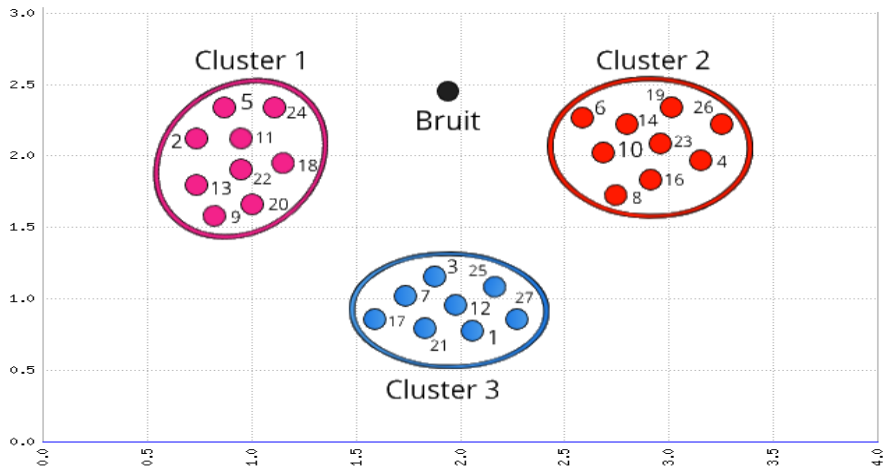


Figure 5.13. Exemple de codage de solution utilisé dans l'approche DCMABC.

5.3.3.2. Fonctions objectif

Bien que certains indices de validité interne aient été proposés pour l'évaluation des résultats de clustering basés sur la densité (Moulavi et al., 2014; Boudane & Berrichi, 2020), ces indices présentent certains inconvénients qui limitent leur efficacité, en particulier dans le cas d'ensembles de données contenant des clusters de formes arbitraires et de densité variable, des clusters non bien séparés (qui se chevauchent) et des outliers. Ainsi, pour l'évaluation des résultats du clustering basé sur la densité, notre idée est d'utiliser les concepts de connectivité directe et indirecte définis par Gabriel et Sokal dans le graphe de Gabriel (Gabriel & Sokal, 1969) (concepts déjà utilisés dans le chapitre 4 pour la définition de l'indice CDBCVI), pour définir des objectifs contradictoires comme suit :

➢ Pour mesurer la similarité au sein des clusters,

- Nous utilisons la définition de la compacité d'une partition C en termes de connectivité des clusters (donnée dans le chapitre 4) par la formule suivante :

$$CompCon(C) = \sum_{i=1}^{k} \sum_{j=1}^{|C_i|} \left(|CN_j \cap C_i| / |CN_j|\right) \quad (5.11)$$

Cette fonction f_1 permet de tester si le voisinage noyau (central) d'un objet fait partie du même cluster que l'objet lui-même. Ainsi, cette fonction est à maximiser.

- Nous utilisons la définition de la compacité d'une partition C en termes de densité de clusters qui est donnée par la formule suivante :

$$CompDens(C) = \sum_{i=1}^{k} \left[\left[\sum_{j=1}^{|C_i|} \sum_{m=1, m\neq j}^{|C_i|} \left(min(dens(j), dens(m)) / max(dens(j), dens(m))\right) \right] / |C_i| \right] \quad (5.12)$$

Où, $dens(x) = \sum_{j \in CN_x \cap C_i} d(j, x)$

Cette fonction f_2 permet de mesurer les changements de densité au sein des clusters. Ainsi, un bon partitionnement implique une valeur élevée de cette fonction.

- Nous définissons, aussi, une nouvelle fonction qui consiste à minimiser le nombre d'objets directement connectés et qui ne sont pas des noyaux dans les clusters comme suit :

$$SeparatD(C) = \sum_{i=1}^{k} SepD(C_i) \quad (5.13)$$

Où, $SepD(C_l) = \sum_{i=1}^{|C_l|} |C_l \cap (DC_i/CN_i)|/(|DC_i| - |CN_i|)$

Cette fonction f_3 permet d'assurer que les objets qui font partie du voisinage d'un objet font partie du même cluster que l'objet lui-même. Ainsi cette fonction est à minimiser car un bon partitionnement se caractérise par une valeur basse de cette mesure de séparation des clusters.

➢ Pour mesurer la séparation entre clusters :

- Nous utilisons la formule, déjà donnée dans le chapitre 4, comme suit :

$$Separat(C) = \sum_{i=1}^{k} Sep(C_i)\,/k \quad (5.14)$$

Où, $Sep(C_l) = \sum_{i=1}^{|C_l|} \sum_{j=1, j\neq l}^{k} \left|C_j \cap (DC_i/CN_i)\right|/(|DC_i| - |CN_i|)$

Un bon partitionnement se caractérise par une valeur élevée de cette mesure de séparation des clusters. Ainsi, cette fonction f_4 est à maximiser.

- Nous définissons, aussi, une nouvelle fonction qui consiste à minimiser le nombre d'objets directement connectés entre clusters.

$$SeparatD(C) = \sum_{i=1}^{k} SepD(C_i)\,/n \quad (5.15)$$

Où, $SepD(C_l) = \sum_{i=1}^{|C_l|} \sum_{j=1, j\neq l}^{k} \left|C_j \cap DC_i\right|/|DC_i|$

Cette fonction f_5 permet d'assurer que le voisinage d'un objet fait partie du même cluster que l'objet lui-même. Ainsi, cette fonction est à minimiser, car un bon partitionnement se caractérise par une valeur basse de cette mesure de séparation des clusters.

➢ En plus des fonctions précédentes, nous définissons une autre fonction objectif qui permet de contrôler le nombre de clusters comme suit :

$$f_6 = abs\ (k - K)/K \quad (5.16)$$

Où, *k* est le nombre de clusters généré et *K* est le nombre de clusters réel (paramètre introduit par l'utilisateur).

Dans certains cas réels, la valeur *K* de clusters est connue a priori, mais il ne peut pas être utilisé pour guider le processus de clustering dans DBSCAN standard. Donc, cette dernière fonction est utilisée pour surmonter l'inconvénient de DBSCAN, à savoir le nombre de clusters ne peut pas être contrôlé par les utilisateurs.

Ainsi, un bon partitionnement se caractérise par une valeur basse de cette fonction (f_6 est à minimiser).

5.3.3.3. Description de l'algorithme DCMABC

Le processus de recherche de l'algorithme DCMABC est le même que celui de l'algorithme NBC-MOABC décrit déjà dans la figure 5.6. Comme NBC-MOABC, DCMABC utilise une archive Pareto de taille fixe pour maintenir les meilleures solutions non dominées trouvées. Cette archive est mise à jour à la fin de chaque itération de l'algorithme ABC. La méthode de mise à jour de l'archive est la même que celle décrite dans l'algorithme 3. Le processus de prise de décision est aussi le même, c.-à-d. que la qualité d'un individu (solution de clustering) est mesurée en fonction de ses valeurs des fonctions objectif retournées et de celles du point utopique. Plus les valeurs des fonctions objectif retournées par un individu sont proches de celles du point utopique, meilleure est la qualité de cet individu.

DCMABC diffère de NBC-MOABC essentiellement dans le codage (décrit dans la section 5.3.3.1) et la génération de solutions au cours des itérations. La génération des solutions, dans les différentes phases de l'algorithme DCMABC, se fait d'une manière aléatoire et déterministe. L'espace de recherche de solutions est divisé en 2 sous espaces. Dans le premier, les mêmes valeurs des paramètres *Eps* et *Minpts* sont utilisées pour générer tous les clusters d'une solution, alors que, dans le deuxième des valeurs différentes des paramètres sont utilisées pour générer chaque cluster.

Les phases de DCMABC qui sont adaptées par rapport à NBC-MOABC sont données comme suit :

- **Phase d'initialisation**

Selon le schéma de codage et de génération de solutions proposé, les sources initiales (solutions) sont sélectionnées en utilisant la fonction d'extension de clusters de l'algorithme DBSCAN dans les deux sous-espaces de recherche de solutions, avec des paramètres associés *Eps* et *Minpts* sélectionnés comme suit :

- Pour la première moitié de la population, les mêmes valeurs de *Eps* et *Minpts* sont utilisées, (sélectionnés aléatoirement dans les intervalles]0, Eps_{max}] et [2, $Minpts_{max}$] respectivement) pour tous les clusters. Autrement dit, DBSCAN est utilisé pour générer la moitié de la population.
- Pour la deuxième moitié de la population, une paire de valeurs différentes de Eps et de Minpts, sélectionnés (d'une façon aléatoire pour certains individus et déterministe pour d'autres) dans les intervalles]0, Eps_{max}] et [2, $Minpts_{max}$] respectivement, est utilisée pour chaque cluster.

- **Phase des abeilles employées**

Dans cette phase, de nouvelles sources de nourriture sont générées par les abeilles employées pour explorer l'espace de recherche, où les valeurs des paramètres *Eps et Minpts* des clusters d'une solution sont générées dans le voisinage de ses anciennes valeurs.

- **Phase des abeilles spectatrices**

Chaque abeille spectatrice produit aléatoirement une nouvelle source de nourriture dans le voisinage de celle sélectionnée.

- **Phase de mutation**

La mutation consiste à échanger les valeurs des paramètres *Eps* et *Minpts* de deux points représentatifs choisis aléatoirement.

- **Phase des abeilles scoutes**

Dans cette phase, les abeilles scoutes remplacent les sources de nourriture abandonnées par les abeilles employées par d'autres générées d'une façon déterministe à partir des sources de nourriture épuisées. Les valeurs des paramètres *Eps et Minpts* utilisés pour produire une nouvelle solution d'un individu sont générées dans le voisinage des anciennes valeurs.

5.3.3.4. Résultats expérimentaux

Dans cette section, nous présentons les résultats obtenus par un ensemble d'expériences pour évaluer la performance de l'approche DCMABC. Nous avons utilisé les même datasets utilisés pour tester l'approche NBC-MOABC. Les comparaisons ont été effectuées en utilisant les indices de validité externe RI et ARI. Nous avons déterminé, tout d'abord, un réglage approprié des paramètres de l'algorithme DCMABC, en utilisant comme fonction de fitness l'indice ARI. Un ensemble étendu d'expériences a été effectué sur les ensembles de données sélectionnés, en utilisant des combinaisons différentes des valeurs des paramètres. Nous avons pu voir que DCMABC converge vers un état stable après environ 50 itérations. Dans certains cas, DCMABC a nécessité moins de 20 itérations. Ainsi, les valeurs des paramètres pour les expériences sont fixés comme suit : taille de la population 20, nombre maximum d'itérations 50, la taille de l'archive Pareto est fixée égale à la taille de la population 20 et le paramètre *limit*e 5. Pour la recherche de solutions les deux paramètres *Eps* et *Minpts* de l'algorithme DBSCAN sont pris dans les intervalles]0, Eps_{max}] et [2, $Minpts_{max}$], telque Eps_{max} et $Minpts_{max}$ sont fixés à 15.

Nous évaluons l'efficacité de l'algorithme DCMABC par rapport à NBC-MOABC et VNS dans l'exploration de l'espace de recherche, en utilisant comme critère de fitness l'indice RI. Le tableau 5.5 ci-après montre les résultats obtenus.

Tableau 5.5. Valeurs moyennes et écart-types des indices RI et ARI mesurés sur les sorties des algorithmes DCMABC, NBC-MOABC et VNS, en utilisant l'indice RI comme critère de fitness.

Datasets	Taille	# Cluster	DCMABC		NBC-MOABC		VNS	
			RI	ARI	RI	ARI	RI	ARI
Pathbased	300	3	0.97 ± 0.008	0.93 ± 0.019	0.94 ± 0.000	0.88 ± 0.000	0.83 ± 0.072	0.64 ± 0.151
Spiral	312	3	1.00 ± 0.000	1.00 ± 0.000	1.00 ± 0.000	1.00 ± 0.000	0.97 ± 0.054	0.94 ± 0.126
Compound	399	6	0.99 ± 0.002	0.99 ± 0.007	0.98 ± 0.000	0.95 ± 0.000	0.95 ± 0.017	0.89 ± 0.042
Jain	373	2	1.00 ± 0.000	1.00 ± 0.000	0.98 ± 0.001	0.97 ± 0.003	0.96 ± 0.001	0.93 ± 0.047
Flame	240	2	0.97 ± 0.004	0.95 ± 0.009	0.97 ± 0.001	0.94 ± 0.005	0.95 ± 0.007	0.91 ± 0.014
Aggregation	788	7	0.99 ± 0.001	0.99± 0.003	0.99 ± 0.000	0.99 ± 0.000	0.99 ± 0.002	0.98 ± 0.006
R15	600	15	0.99 ± 0.000	0.98± 0.007	0.99 ± 0.000	0.98 ± 0.005	0.98 ± 0.015	0.89 ± 0.123

Comme le montre le tableau 5.5 ci-dessus, la capacité de recherche de solutions de l'algorithme DCMABC est meilleure que celle des algorithmes NBC-MOABC et VNS dans presque tous les ensembles de données, en particulier dans le cas des ensembles de données Path-based, Compound et Jain. Par exemple, dans le cas de l'ensemble de données Jain qui présente une variation de densité, l'indice ARI a une valeur égale à 1 dans le cas de DCMABC, ce qui signifie que les clusters obtenus sont identiques aux vrais. Par rapport à l'approche VNS et NBC-MOABC qui utilisaient le même codage de solution, l'algorithme DCMABC montre l'influence significative de la stratégie de recherche de solutions qui est basée sur un changement aléatoire et déterministe des valeurs des paramètres de l'algorithme DBSCAN (utilisation d'une valeur différente pour chaque cluster au lieu d'une même valeur pour tous les clusters) dans l'exploration de l'espace de recherche.

Pour comparer la performance de DCMABC avec *k*-means, single-linkage, DBSCAN, NC-closures, MCPSO et NBC-MOABC, en termes de précision, les valeurs moyennes des indices RI et ARI des résultats de clustering sont données pour chaque ensemble de données, dans les tableaux 5.6 et 5.7, respectivement. Nous avons utilisé comme fonctions objectif, à titre d'exemple, les 5 fonctions objectif f_1, f_2, f_3, f_4, f_5 définies par les équations 5.11, 5.12, 5.13, 5.14 et 5.15, respectivement. Aussi, pour permettre de contrôler le nombre de clusters dans le cas où le nombre de clusters réel est disponible, nous avons utilisé la fonction objectif f_6 pour DCMABC.

Tableau 5.6. Valeurs moyennes et écart-types de RI mesurés sur les sorties des algorithmes.

Datasets	Taille	# Cluster	NBC-MOABC	K-Means	Single-linkage	DBSCAN	NC-closures	MCPSO	DCMABC avec *f1, f2, f3, f4, f5* comme fonctions objectif	DCMABC avec $f_1, f_2, f_3, f_4, f_5, f_6$ comme fonctions objectif
Pathbased	300	3	0.79 ± 0.010	0.84 ± 0.005	0.85 ± 0.037	0.90 ± 0.005	0.89 ± 0.007	0.94 ± 0.006	0.78 ± 0.064	0.88 ± 0.068
Spiral	312	3	1.00 ± 0.000	0.79 ± 0.084	0.87 ± 0.002	0.93 ± 0.020	0.81 ± 0.007	0.89 ± 0.003	0.93 ± 0.027	0.97 ± 0.029
Compound	399	6	0.95 ± 0.002	0.81 ± 0.003	0.75 ± 0.052	0.80 ± 0.016	0.79 ± 0.014	0.92 ± 0.075	0.87 ± 0.073	0.95 ± 0.028
Jain	373	2	0.98 ± 0.002	0.89 ± 0.012	0.75 ± 0.004	0.96 ± 0.002	0.90 ± 0.006	0.95 ± 0.008	0.915 ± 0.106	0.94 ± 0.106
Flame	240	2	0.95 ± 0.009	0.83 ± 0.006	0.85 ± 0.003	0.89 ± 0.051	0.87 ± 0.042	0.91 ± 0.021	0.93 ± 0.035	0.96 ± 0.019
Aggregation	788	7	0.99 ± 0.002	0.94 ± 0.003	0.89 ± 0.014	0.88 ± 0.023	0.85 ± 0.017	0.97 ± 0.028	0.94 ±0.019	0.99 ± 0.000
R15	600	15	0.99 ± 0.003	0.91 ± 0.037	0.87 ± 0.009	0.89 ± 0.014	0.90 ± 0.007	0.94 ± 0.042	0.96± 0.024	0.98 ± 0.025

Tableau 5.7. Valeurs moyennes et écart-types de ARI mesurés sur les sorties des algorithmes.

Datasets	Taille	# Cluster	NBC-MOABC	K-Means	Single-linkage	DBSCAN	NC-closures	MCPSO	DCMABC avec $f_1, f_2, f_3, f_4\ f_5$ comme fonctions objectif	DCMABC avec $f_1, f_2, f_3, f_4, f_5, f_6$ comme fonctions objectif
Pathbased	300	3	0.54 ± 0.022	0.71 ± 0.002	0.72 ± 0.024	0.78 ± 0.002	0.76 ± 0.005	0.81 ± 0.003	0.55 ± 0.094	0.74 ± 0.156
Spiral	312	3	1.00 ± 0.000	0.70 ± 0.012	0.79 ± 0.003	0.86 ± 0.041	0.74 ± 0.002	0.82 ± 0.001	0.84 ± 0.060	0.95 ± 0.070
Compound	399	6	0.87 ± 0.006	0.75 ± 0.022	0.68 ± 0.023	0.74 ± 0.016	0.70 ± 0.009	0.86 ± 0.011	0.71 ± 0.131	0.87 ± 0.081
Jain	373	2	0.96 ± 0.005	0.41 ± 0.010	0.37 ± 0.002	0.69 ± 0.002	0.50 ± 0.014	0.66 ± 0.006	0.90± 0.088	0.96 ± 0.047
Flame	240	2	0.90 ± 0.020	0.43 ± 0.005	0.47 ± 0.004	0.56 ± 0.031	0.51 ± 0.022	0.73 ± 0.051	0.87 ± 0.070	0.92 ± 0.038
Aggregation	788	7	0.98 ± 0.004	0.81 ± 0.011	0.80 ± 0.010	0.76 ± 0.067	0.72 ± 0.027	0.87 ± 0.032	0.87 ± 0.039	0.99 ± 0.001
R15	600	15	0.91 ± 0.030	0.85 ± 0.074	0.77 ± 0.008	0.80 ± 0.027	0.83 ± 0.006	0.88 ± 0.032	0.80± 0.150	0.88 ± 0.157

Comme le montre les tableaux 5.6 et 5.7, l'approche DCMABC présente une meilleure performance par rapport aux algorithmes *k*-means, signle-linkage, DBSCAN, NC-closures et MCPSO dans presque tous les ensembles de données, en particulier dans le cas des ensembles de données Jain, Flame et Spiral.

Bien que DCMABC soit meilleure dans l'exploration de l'espace de recherche par rapport à NBC-MOABC (voir tableau 5.5), DCMABC a donné des résultats similaires ou proches dans la plupart des cas. Ceci est dû à l'influence significative des différentes fonctions objectif que nous avons défini, dans l'évaluation des solutions de clustering (individus) durant les différentes itérations. Afin de tester l'efficacité des différentes fonctions objectif que nous avons défini, d'autres expériences peuvent être effectuées en utilisant des combinaisons différentes. De plus, dans le cas où le nombre de clusters est disponible, l'utilisation de la fonction objectif f_6 permet d'améliorer considérablement les résultats.

Le tableau 5.8 ci-après présente les valeurs des indices RI et ARI des meilleurs résultats de clustering trouvés par les deux approches NBC-MOABC et DCMABC sans et avec contrôle du nombre de clusters.

Tableau 5.8. Valeurs de RI et ARI mesurés sur les meilleures sorties des algorithmes NBC-MOABC et DCMABC sans et avec utilisation de la fonction objectif f_6 (c-à-d, sans et avec contrôle du nombre de clusters).

Datasets	Taille	# Cluster	NBC-MOABC sans contrôle du nombre de clusters		DCMABC sans contrôle du nombre de clusters		NBC-MOABC avec contrôle du nombre de clusters		DCMABC avec contrôle du nombre de clusters	
			RI	ARI	RI	ARI	RI	ARI	RI	ARI
Pathbased	300	3	0.8026	0.5657	0.8784	0.7035	0.9452	0.8732	0.9628	0.916
Spiral	312	3	1	1	0.9765	0.946	1	1	1	1
Compound	399	6	0.9529	0.878	0.9158	0.7909	0.9694	0.9036	0.9826	0.9538
Jain	373	2	0.9867	0.9722	1	1	0.9867	0.9722	1	1
Flame	240	2	0.9694	0.9387	0.9752	0.9501	0.9587	0.9283	0.9858	0.9715
Aggregation	788	7	0.9957	0.99	0.9671	0.9261	0.9952	0.9887	0.999	0.9976
R15	600	15	0.9965	0.9714	0.997	0.9752	0.9989	0.991	0.9971	0.9774

5.4. Conclusion

Nous avons présenté, dans ce chapitre, une approche de clustering basée sur la métaheuristique à solution unique VNS et deux autres approches basées sur la métaheuristique à base de population ABC. Ces approches ont permis de pallier aux problèmes rencontrés par la plupart des algorithmes de clustering existants (en particulier ceux basés sur la densité) et traiter correctement les ensembles de données ayant des clusters de formes arbitraires.

Ces approches utilisent un schéma de codage de solution basé sur la densité. Pour l'évaluation des solutions de clustering au cours des itérations, la prise en compte d'une seule fonction objectif peut ne pas être conforme aux ensembles de données avec des outliers et aux formes complexes des clusters. Par conséquent, nous avons proposé et utilisé plusieurs

fonctions objectif basées sur des concepts de densité, pour améliorer le processus d'évaluation.

Une étude comparative par rapport à plusieurs approches existant dans la littérature a montré l'efficacité des différentes approches proposées sur plusieurs ensembles de données. Nos contributions permettent, ainsi, d'automatiser et d'améliorer la qualité du clustering et surpassent les algorithmes de clustering classiques qui ne peuvent pas traiter les ensembles de données ayant des clusters de formes arbitraires et une densité variable. De plus, nos approches sont plus facile à utiliser car elles ne nécessitent pas de connaissances a priori sur l'ensemble de données et aucun paramètre défini par l'utilisateur comme le nombre de clusters ou le seuil de densité.

Conclusion générale et perspectives

L'objectif principal des travaux que nous avons réalisés dans le cadre de cette thèse est de traiter le problème de clustering dans des ensembles de données avec un nombre inconnu de clusters ayant des formes arbitraires et qui présentent des variations de densité et des outliers. Ces spécifications peuvent être observées dans les ensembles de données spatiales tels que les systèmes d'information géographique et les applications biomédicales. En la présence de toutes ces spécifications, la motivation principale de nos travaux a été d'automatiser le processus de clustering afin d'éviter les difficultés rencontrées par l'utilisateur dans la détermination des meilleures valeurs des paramètres d'entrée dans le cas des algorithmes de clustering basés sur la densité et d'améliorer la qualité des résultats de clustering.

Pour atteindre notre objectif, nous avons proposé plusieurs solutions.

La première solution qui est la première contribution à cette thèse consiste à proposer un indice de validation de clustering, pour faciliter le choix de l'algorithme de clustering et de ses paramètres appropriés pour une situation particulière. Nous avons défini, donc, un nouvel indice de validation de clustering basé sur la connectivité et la densité (CDBCVI). Cet indice permet de faire face au cas de clusters de formes arbitraires et de différentes densités. Il facilite ainsi l'évaluation des algorithmes de clustering et la sélection de leurs paramètres appropriés. Contrairement à la plupart des indices proposés dans la littérature pour la validation des clusters globulaires, qui sont basés sur la distance, CDBCVI est basé sur la connectivité et la densité. Il mesure la compacité de chaque cluster (validité intra-cluster) et la séparation (validité inter-clusters) entre clusters, en utilisant les concepts de connectivité directe et indirecte définis par Gabriel et Sokal dans le graphe de Gabriel (Gabriel & Sokal, 1969). Les nouvelles définitions des mesures de compacité et de séparation permettent à l'indice CDBCVI de traiter correctement les cas de clusters de formes arbitraires et d'outliers. Nous avons testé la performance de CDBCVI sur plusieurs ensembles de données synthétiques et réels, en particulier ceux ayant des clusters avec des formes arbitraires et des variations de densité, en utilisant les algorithmes de clustering bien connus NBC et DBSCAN. Les résultats expérimentaux ont montré l'efficacité de l'indice CDBCVI pour la validation des résultats de clustering, particulièrement dans le cas de structures de données complexes et de chevauchements importants entre les clusters où les indices de validation proposés dans la littérature peuvent ne pas aboutir à des résultats satisfaisants.

La deuxième contribution consiste à proposer des approches de clustering mono- et multi-objectif qui permettent de réaliser un clustering avec toutes les spécifications citées précédemment. Étant donné que les algorithmes NBC et DBSCAN sont très efficaces dans le cas de clusters de formes arbitraires et de différentes densités, nous avons tiré avantage de ces algorithmes et des deux métaheuristiques VNS et ABC, afin d'automatiser le processus de

clustering et améliorer la qualité des résultats (tout en évitant la difficulté du choix des valeurs des paramètres qui a un grand impact sur l'efficacité des algorithmes de clustering). Trois approches de clustering qui utilisent un schéma de codage de solutions basé sur la densité ont été proposées. La première consiste à utiliser la métaheuristique à solution unique VNS afin de remédier à la difficulté du choix de la valeur du paramètre unique de l'algorithme NBC et automatiser ainsi le processus de clustering. La deuxième consiste à utiliser l'algorithme ABC afin d'automatiser et améliorer la qualité du clustering de l'algorithme NBC. Quant à la troisième, elle consiste à utiliser l'algorithme ABC afin d'automatiser et améliorer la qualité du clustering en s'inspirant de la procédure d'extension de clusters de l'algorithme DBSCAN.
Nous avons défini et utilisé plusieurs fonctions objectif basées sur des concepts de densité, pour améliorer le processus d'évaluation des solutions de clustering au cours des itérations, vu que la prise en compte d'une seule fonction objectif peut ne pas être conforme aux ensembles de données ayant des clusters de formes complexes et des outliers.
Une étude comparative des approches développées (en utilisant certaines fonctions objectif proposées) par rapport à plusieurs approches existantes (bien connues) a montré l'efficacité et la supériorité des différentes approches proposées sur plusieurs ensembles de données présentant des clusters de formes arbitraires et une densité variable. De plus, les approches proposées sont plus facile à utiliser par rapport à celles existant dans la littérature car elles ne nécessitent pas de connaissances a priori sur l'ensemble de données et aucun paramètre utilisateur, comme le nombre de clusters ou le seuil de densité.

En résumé, le contenu de cette thèse apporte plusieurs contributions à la littérature de recherche. Tout d'abord, la revue de la littérature, présentée dans les trois premiers chapitres, a rassemblée et discutée les difficultés posées par le problème de clustering ainsi que les avantages et les lacunes des principales méthodes existantes, spécialement celles basées sur les méthaheuristiques. Elle pourrait être une source pour les recherches futures, étudiant le problème de clustering. Deuxièmement, nous avons proposé un nouvel indice de validation de clustering, pour faciliter le choix de l'algorithme de clustering et de ses paramètres appropriés pour une situation particulière. Il permet de faire face au cas de clusters de formes arbitraires et de différentes densités. Enfin, nos contributions, proposées dans les deux derniers chapitres, permettent d'automatiser et améliorer la qualité du clustering, particulièrement, dans le cas des ensembles de données ayant des clusters de formes arbitraires et une densité variable.

Comme futurs travaux,

- Nous prévoyons de développer davantage l'indice CDBCVI proposé en incluant d'autres mesures pour calculer la cohésion et la séparation des clusters voire en utilisant d'autres façons pour leurs définitions.
- Une autre question intéressante à approfondir est le fait que, dans des situations particulières, nous pouvons souhaiter trouver des clusters à une échelle spécifique (par exemple, en ignorant les pics de densité inférieurs à une valeur prédéfinie). Ainsi, nous suggérons de tirer pleinement partie des algorithmes de clustering DBSCAN et NBC pour discuter des stratégies de définition de l'échelle en tant que paramètres de l'indice CDBCVI proposé.

- Nous prévoyons de définir de nouvelles fonctions objectif permettant une meilleure évaluation des solutions de clustering en cas de clusters non clairement séparés (chevauchements entre clusters) ou non concentriques.
- Nous envisageons d'effectuer d'autres expériences sur les différentes approches proposées en utilisant des combinaisons différentes de fonctions objectif (déjà proposées et nouvelles). Aussi, nous envisageons d'effectuer de nouvelles expériences, en appliquant les approches proposées, sur certains domaines de recherches tels que la bio-informatique et la segmentation d'images.
- Appliquer l'algorithme ABC pour résoudre d'autres tâches du datamining telles que la classification supervisé.
- Une direction de recherche future intéressante peut être le clustering par contraintes. En règle générale, les contraintes représentent les informations spécifiques au problème et les préférences d'un expert de domaine dans un problème de clustering, et leur prise en compte permettent l'efficacité du clustering. Ainsi, il est possible d'incorporer des mécanismes de gestion des contraintes aux approches proposées.

Bibliographie

Agrawal, R., Gehrke, J., Gunopulos, D., & Raghavan, P. (1998). Automatic subspace clustering of high dimensional data for data mining applications. In Proceedings of Int. Conf. Management of Data ACM-SIGMOD, pp. 94-105. Seattle.

Agustín-Blas, L.E., Salcedo-Sanz, S., Jiménez-Fernández, S., Carro-Calvo, L., Del Ser, J., & Portilla-Figueras, J. A. (2012). A new grouping genetic algorithm for clustering problems. Expert Systems with Applications. 39, 9695-9703.

Ahmadi, A., Karray, F., & Kamel, M.S. (2010). Flocking based approach for data clustering. Natural Computing. 9 (3), 767-791.

Akbari, R., Hedayatzadeh, R., Ziarati, K., & Hassanizadeh, B. (2012). A multi-objective artificial bee colony algorithm. Swarm and Evolutionary Computation. 2, 39-52.

Almeida, H., Neto, D.G., Jr, W.M., & Zaki, M.J. (2012). Towards a Better Quality Metric for Graph Cluster Evaluation. Journal of Information and Data Management. 3 (3), 378-393.

Al-Sultan, K.S. (1995). A tabu search approach to the clustering problem. Pattern Recognition. 28(9), 1443-1451.

Al-Sultan, K.S., & Fedjki, C.A. (1997). A tabu search-based algorithm for the fuzzy clustering problem. Pattern Recognition. 30(12), 2023-2030.

Amarjeet, P., & Chhabra, J.K. (2017). TA-ABC : Two-Archive Artificial Bee Colony for Multi-objective Software Module Clustering Problem. Journal of Intelligent Systems. 27(4), 619-641. Doi : 10.1515/jisys-2016-0253.

Anderberg, M.R. (1973). Cluster Analysis for Applications. Academic Press, New York.

Ankerst, M., Breunig, M., Kriegel, H. P., & Sander, J. (1999). OPTICS : Ordering Points To Identify the Clustering Structure. In : Proceedings of Int. Conf. Management of Data ACM-SIGMOD, pp. 49-60. Philadelphia, Pennsylvania, USA.

Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J. M., & Perona, I. (2013). An extensive comparative study of cluster validity indices. Pattern Recognition. 46, 243-256.

Armano, G., & Farmani, M.R. (2014). Clustering Analysis with Combination of Artificial Bee Colony Algorithm and k-Means Technique. International Journal of Computer Theory and Engineering. 6(2), 141-145. Doi : 10.7763/IJCTE.2014.V6.852.

Armano, G., & Farmani, M.R. (2016). Multiobjective clustering analysis using particle swarm optimization. Expert Systems With Applications. 55, 184-193.

Babu, G.P., & Murty, M.N. (1993). A near optimal initial seed value selection in the k-meanws algorithm using a genetic algorithm. Pattern Recognition Letters. 14(10), 763-769.

Bagirov, A.M, Karmitsa, N., & Taheri, S. (2020). Metaheuristic Clustering Algorithms. Partitional Clustering via Nonsmooth Optimization.165-183.

Bai, Q. (2010). Analysis of particle swarm optimization algorithm. Comput. Inf. Sci. 3(1), 180-184.

Banerjee, A. (2013). Evolutionary Algorithms for Robust Density-Based Data Clustering. Computational Mathematics. Doi : 10.1155/2013/931019.

Barbakh, W., Wu, Y. & Fyfe, C. (2009). Review of Clustering Algorithms. In : Non-Standard Parameter Adaptation for Exploratory Data Analysis. 7-28. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-04005-4-2

Barbarà, D., & Jajodia, S. (2002). Applications of Data Mining in Computer Security. Kluwer Academic Publishers, Boston, MA, pp. 78-99.

Batagelj, V., & Bren, M. (1995). Comparing resemblance measures. Journal of Classification. 12 (1995) 73-90.

Belahbib, F., & Souami, F. (2011). Genetic algorithm clustering for color image quantization. 3rd European Workshop on Visual Information Processing (EUVIP), pp. 83-87.

Berkhin, P. (2002). Survey Of Clustering Data Mining Techniques. Rapport technique, San Jose, CA, Accrue Software.

Berry, M.J.A. & Linoff, G. (1996). Data Mining Techniques For Marketing, Sales and Customer Support. John Wiley & Sons, Inc., USA.

Bezdek J.C., Boggavarapu, S., Hall, L.O., & Bensaid, A. (1994). Genetic algorithm guided clustering. IEEE Congress on Evolutionary Computation, CEC. 34-40.

Bhuyan, N.J., Raghavan, V.V., Venkatesh, K.E. (1991). Genetic algorithms for clustering with an ordered representation. In Proceedings of the Fourth International Conference on Genetic Algorithms, pp. 408-415.

Birant, D., & Kut, A. (2007). ST-DBSCAN : An Algorithm for Clustering Spatial-temporal data. Data and Knowledge Engineering. 208-221.

Boley, D. (1998). Principal Direction Divisive Partitioning. Data Mining and Knowledge Discovery. 2, (4), 325-344.

Brown, D.E., & Entail, C.L. (1992). A practical application of simulated annealing to the clustering problem. Pattern Recognition. 25, 401-412.

Boudane, F., & Berrichi, A. (2017). Variable neighborhood search for automatic density-based clustering. In Proceeding of the 2017 International Conference on Mathematics and Information Technology (ICMIT 2017*)*, pp.141-147. Doi : 10.1109/MATHIT.2017.8259708.

Boudane, F., & Berrichi, A. (2020). Gabriel graph-based connectivity and density for internal validity of clustering. Progress in Artificial Intelligence. 9, 221-238. https://doi.org/10.1007/s13748-020-00209-z

Boudane, F., & Berrichi, A. (2022). Multi-Objective Artificial Bee Colony Algorithm for Parameter-Free Neighborhood-Based Clustering. International Journal of Swarm Intelligence Research. 13(2), Article 1.

Brandenburg, F., Edachery, J., & Sen, A. (1999). Graph clustering using distance-k cliques. Lecture Notes in Computer Sciences. 1731, 98-106.

Brézellec, P., & Didier, G. (2001). GIZMO : un algorithme de grille cherchant des clusters homogènes. In : Conférence francophone d'Apprentissage (CAp'2001), pp. 101-116. Grenoble, France.

Calinski, T., & Harabasz, J. (1974). A dendrite method for cluster analysis. Communications in Statistics. 3, 1-27.

Chaves, A.A., & Lorenab, L.A.N. (2011). Hybrid evolutionary algorithm for the capacitated centered clustering problem. Expert Systems with Application. 38(5), 5013-5018.

Chou, C.H., Su, M.C., & Lai, E. (2004). A new cluster validity measure and its application to image compression. Pattern Analysis and Applications 7, 205-220.

Chowdhury, A.K.M.R., Mollah, M.E., & Rahman, M.A. (2010). An efficient Mehtod for subjectively choosing parameter k automatically in VDBSCAN. Proceedings of ICCAE 2010 IEEE, 1, pp. 38-41.

Chuai-Aree, S., Lursinsap, C., Sophatsathit, P., & Siripant, S. (2000). Fuzzy C-Mean : A statistical feature classification of text and image segmentation method. In : Proceeding of Intern. Conf. on Intelligent Technology, pp. 279-284. Assumption University Bangkok, Thailand.

Chuang, L.Y., Hsiao, C.J., & Yang, C.H. (2011). Chaotic particle swarm optimization for data clustering. Expert Systems with Applications. 38(12), 14555-14563.

Cleuziou, G. (2004). Une méthode de classification non-supervisée pour l'apprentissage de règles et la recherche d'information. Thèse de doctorat. Université d'Orléans, 2004. HAL Id : tel-00084828. https://tel.archives-ouvertes.fr/tel-00084828

Cowgill, M.C., Harvey, R.J., & Watson, L.T. (1999). A genetic algorithm approach to cluster analysis. Computers & Mathematics with Application. 37, 99-108.

Cura, T. (2012). A particle swarm optimization approach to clustering. Expert Systems with Applications. 39(1), 1582-1588.

Danish, Z., Shah, H., Tairan, N., Gazali, R., & Badshah, A. (2019). Global Artificial Bee Colony Search Algorithm for Data Clustering. International Journal of Swarm Intelligence Research, 10(2).Doi : 10.4018/IJSIR.2019040104.

Das, S., Abraham, A., & Konar, A. (2009). Clustering using multi-objective differential evolution algorithms. In : Metaheuristic Clustering. Studies in Computational Intelligence, 178, pp. 213-238. Springer, Berlin/Heidelberg.

Das, S., Abraham, A., & Konar, A. (2008). Automatic kernel clustering with a Multi-Elitist Particle Swarm Optimization Algorithm. Pattern Recognition Letters. 29(5), 688-699.

Davies, D.L., & Bouldin, D.W. (1979). A clustering separation measure. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1, 224-227.

Deborah, L. J., Baskaran R., & Kannan, A. (2010). A survey on internal validity measure for cluster validation. International Journal of Computer Science & Engineering Survey. 1(2), 85-102.

Dempster, A.P., Laird, N.M., & Rubin, D.B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of Royal Statistical Society B. 39, 1-38.

Diday, E. (1975). La méthode des nuées dynamiques. Revue de Statistique Appliquée. XIX (2), 19-34.

Duda, R.O., Hart, P.E., & Stork, D.G. (2001). Unsupervised learning and clustering. Pattern classification, page 571.

Dunn, J.C. (1974). Well separated clusters and optimal fuzzy partitions. Journal of Cybernetics. 4 (1), 95-104.

Eberhart, R.C. & Shi, Y. (2001). Particle swarm optimization : Developments, applications and resources. In : Proceedings of the 2001 Congress on Evolutionary Computation, Seoul, Korea, pp. 81-86.

Ertöz, L., Steinbach, M., & Kumar, V. (2003). Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data., in : SDM, pp. 47-58.

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In : Proceedings of KDD, pp. 226-231.

Falkenauer, E. (1998). Genetic Algorithms and Grouping Problems. Wiley, New York.

Franti, P. (2015). Speech and image processing unit, clustering datasets. School of Computing, University of Eastern Finland.

Fung, G. (2001). A comprehensive overview of basic clustering algorithms. Technical Report. Department of Computer Sciences, University of Wisconsin–Madison.

Gabriel, K.R., & Sokal, R.R. (1969). New statistical approach to geographic variation analysis. Syst. Zool. 18 (3), 259-278.

Garey, M.R., & Johnson, D.S. (1979). Computers and Intractability : A Guide to the Theory of NP-Completeness, W. H. Freeman.

Gen, M., & Cheng, R. (1997). Genetic Algorithms and Engineering Design. Wiley, New York.

Goldberg, D. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley.

Gong, A., Gao, Y., Ma, X., Gong, W., Li, H., & Gao, Z. (2016). An Optimized Artificial Bee Colony Algorithm for Clustering. International Journal of Control and Automation. 9(4), 107-116.

Gong, M. Zhang, L., Jiao, L., & Gou, S. (2007). Solving multi-objective clustering using animmune inspired algorithm. In : Proceedings of IEEE Conference on Evolutionary Computation, pp. 15-22.

Grabmeier, J., & Rudolph, A. (2002). Techniques of Cluster Algorithms in Data Mining. Data Mining and Knowledge Discovery. 6, pp. 303-360.

Guan, C., Yuen, K.K.F, & Coenen, F. (2019). Particle swarm Optimized Density-based Clustering and Classification : Supervised and unsupervised learning approaches. Swarm and Evolutionary Computation. 44, 876-896.

Guha, S., Rastogi, R., & Shim, K. (2000). ROCK : A Robust Clustering Algorithm for Categorical Attributes. Information Systems. 25(5), 345-366.

Guha, S., Rastogi, R., & Shim, K. (1998). CURE : an efficient clustering algorithm for large databases. In : proceedings of ACM SIGMOD International Conference on Management of Data.73-84. Seattle, Washington.

Gupta, T., & Kumar, D. (2014). Optimization of Clustering Problem Using Population Based Artificial Bee Colony Algorithm : A Review. International Journal of Advanced Research in Computer Science and Software Engineering. 4(4), 491-502.

Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2002). Clustering Validity Checking Methods : Part II. Newsletter ACM SIG MOD Record. 31 (3), 19-27. Doi : 10.1145/601858.601862

Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2001). On Clustering Validation Techniques. Journal of Intelligent Information Systems. 17 (2/3), 107-145.

Halkidi, M., & Vazirgiannis, M. (2001). Clustering validity assessment : finding the optimal partitioning of a data set. In Proceedings of the First IEEE International Conference on Data Mining (ICDM'01), pp. 187-194, California, USA.

Halkidi, M., & Vazirgiannis, M. (2008). A density-based cluster validity approach using multi-representatives. Pattern Recognition Letters. 29, 773-786.

Han, J. & Kamber, M. (2001). Data Mining : Concepts and Techniques. Morgan Kaufman, Massachusetts.

Han, J., Kamber, M., & Peng, J. (2012). Data Mining Concepts and Techniques, third ed.

Han, J., Lee, J-G., & Kamber, M. (2009). An Overview of Clustering Methods in Geographic Data Analysis. Geographic Data Mining and Knowledge Discovery, Second Edition. pp 149-188.

Hancer, E., & Karaboga, D. (2017). A comprehensive survey of traditional, merge-split and evolutionary approaches proposed for determination of cluster number. Swarm and Evolutionary Computation. 32, 49-67.

Handl, J. & Knowles, J. (2007). An evolutionary approach to multiobjective clustering. IEEE Transactions on Evolutionary Computation. 11 (1), 56-76.

Handl, J., & Knowles, J. (2004). Evolutionary multi-objective clustering. In : Proceedings of 8th International Conference on Parallel Problem Solving from Nature, pp. 1081-1091.

Hansen, P., & Mladenovic, N. (2001). Variable neighborhood search : principles and applications. European Journal of Operations Research. 130. 449-467.

Hansen, P., Mladenovic, N., & Pérez, J.A.M. (2008). Variable neighborhood search : methods and applications. 4OR : A Quarterly Journal of Operations Research. 6, 319-360.

Hartigan, J. (1975). Clustering Algorithms. John Wiley & Sons, New York, NY.

Hartigan, J.A., & Wong, M.A. (1979). A K-Means clustering algorithm, Applied Statistics. 28, 100-108.

He, Y., & Chen, L. (2003). A novel nonparametric clustering algorithm for discovering arbitrary shaped clusters. In : Proceedings of the 2003 Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia, 1826-1830.

Hruschka, E. R., Campello, R. J. G. B., Freitas, A. A., & De Carvalho, A. C. P. L. F. (2009). A survey of evolutionary algorithms for clustering. IEEE Transactions on Systems, Man, and Cybernet., Part C : Applications and Reviews, 39, 133-155.

Hubert, L., & Arabie, P. (1985). Comparing partitions. Journal of classification. 2 (1), 193-218.

Ichino, M. & Yaguchi, H. (1994). Generalized Minkowski metrics for mixed feature-type data analysis. IEEE Transactions on Systems, Man, and Cybernetics. 24 (4), 698-708.

Inkaya, T., Kayalıgil, S., Özdemirel, N.E. (2010). A new density-based clustering approach in graph theoretic context. International Journal of Computer Science and Information Technology. 5 (2), 117-135.

Inkaya, K., & özdemirel.(2013). A neighborhood construction algorithm for the clustering problem. Technical Report, Middle East Technical University, Ankara, Turkey.

Jaccard, P. (1912). The distribution of the flora in the alpine zone. New Phytol. 11, 37-50.

HÖsel, V., & Walcher, S. (2000). Clustering Techniques : A Brief Survey. Rapport Technique, 62-71.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering : a review. ACM Computing Surveys. 31(3), 264-323.

Jarboui, B., Cheikh, M., Siarry, P., & Rebai, A. (2007). Combinatorial particle swarm optimization for partitional clustering problem. Applied Mathematics and Computation. 192 (2), 337-345.

Ji, J., Pang, W., Zheng, Y., Wang, Z., & Ma, Z. (2015). A novel artificial bee colony based clustering algorithm for categorical data. PLoS ONE. 10(5), 1-17.

Jiang, H., Yi, S., Li, J., Yang, F., & Hu, X. (2010). Ant clustering algorithm with Kharmonic means clustering. Expert Systems with Applications. 37(12), 8679-8684.

Kao, Y.T., Zahara, E., & Kao, I.W. (2008). A hybridized approach to data clustering. Expert Systems with Applications. 34 (3), 1754-1762.

Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. Technical Report. No.06. Computer Engineering Department, Engineering Faculty, Erciyes University.

Karaboga, D., & Ozturk, C. (2011). A novel clustering approach : artificial bee colony algorithm. Applied Soft Computing. 11, 652-657.

Karami, A. & Johansson, R. (2014). Choosing dbscan parameters automatically using differential Evolution. International Journal of Computer Applications. 91(7), 1-11.

Karypis, G., Han, E.-H., &Kumar , V. (1999). CHAMELEON : Hierarchical Clustering Using Dynamic Modeling. Computer. 32(8), 68-75.

Kasprzak, E. M., & Lewis, K. E. (2001). Pareto analysis in multiobjective optimization using the collinearity theorem and scaling method. Structural and Multidisciplinary Optimization. 22 (3), 208-218.

Kaufman, L., & Rousseeuw, P. J. (1990). Finding Groups in Data. An Introduction to Cluster Analysis. John Wiley & Sons, Inc, New York, NY.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. IEEE Int. Conf. Neural Networks, Perth, WA, Australia, vol 4, pp.1942-1948.

Kishor, A., Singh, P.K., & Prakash, J. (2016). NSABC : Non-dominated sorting based multi-objective artificial bee colony algorithm and its application in data clustering. Neurocomputing, 216 (2016), 514-533.

Klein, R.W., & Dubes, R.C. (1989). Experiments in projection and clustering by simulated annealing. Pattern Recognition. 22, 213-220.

Knorr, E. M., & Ng, R. T. (1998). Algorithms for Mining Distance-Based Outliers in Large Datasets. In : Proceedings of the 24rd International Conference on Very Large Data Bases. pp. 392-403. New York City, USA.

Kohonen, T. (1984). Self-Organization and Associative Memory. Springer.

Koontz, W.L.G., Narendra, P.M., & Fukunaga, K. (1976). A graph-theoretic approach to non parametric cluster analysis. IEEE Transactions on Computers. 25, 936-944.

Kumar, A., Kumar, D., & Jarial, S.K. (2017). A Review on Artificial Bee Colony Algorithms and Their Applications to Data Clustering. Cybernetics and Information Technologies, 17(3), 3-28. Doi : 10.1515/cait-2017-0027.

Kuo, R.J, & Lin, L.M. (2010). Application of a hybrid of genetic algorithm and particle swarm optimization algorithm for order clustering. Decision Support Systems. 49(4), 451-462.

Kuo, R.J, Syu, Y.J, Chen, Z.Y, & Tien, F.C. (2012). Integration of particle swarm optimization and genetic algorithm for dynamic clustering. Information Sciences. 195, 124-140.

Kurada, R.R., Pavan, K.K., & Rao, A.V.D. (2013). A Preliminary Survey On Optimized Multiobjective Metaheuristic For Data Clustering Using Evolutionary Approaches. International Journal of Computer Science and Information Technology. 5(5), 57–77. Doi : 10.5121/ijcsit.2013.5504

Lee, S., Jeong, Y., Kim, J., & Jeong, M.K. (2018). A New Clustering Validity Index for Arbitrary Shape of Clusters. Pattern Recognition Letters. 112, 263-269. Doi : https://doi.org/10.1016/j.patrec.2018.08.005

Lee, D.T., & Schachter, B.J. (1980). Two Algorithms for Constructing a Delaunay Triangulation. International Journal of Computer and Information Sciences. 9, 219-242.

Rokach, L. (2010). A survey of Clustering Algorithms, Data Mining and Knowledge Discovery Handbook, 2nded, pp 269-298.

Liu, Y., Li, Z. , Xiong, H., Gao, X., Wu, J. & Wu, S. (2013). Understanding and enhancement of internal clustering validation measures. IEEE Transactions on Cybernetics. 43(3), 982-994.

Liu, D., Nosovskiy, G. V., & Sourina, O. (2008). Effective clustering and boundary detection algorithm based on Delaunay triangulation. Pattern Recognition Letters, 29 (9), 1261-1273.

Liu, Y., Wu, X., & Shen, Y. (2011). Automatic clustering using genetic algorithms. Applied Mathematics and Computation. 218(4), 1267-1279.

Liu, P., Zhou, D., & Wu, N. (2007). Varied Density Based Spatial Clustering of Application with Noise. In proceedings of IEEE Conference ICSSSM 2007, pp. 528-531.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In : Proceedings of the Fifth Berkeley Symposium on Mathematical statistics and probability. pp. 281-297. Berkeley.

Malerba, D., Esposito, F., Gioviale, V., & Tamma, V. (2001). Comparing Dissimilarity Measures for Symbolic Data Analysis. In : Int. Conferences on Exchange of Technologies and Know-How and New Techniques & Technologies for Statistics. Crete, Greece.

Martin, L., & Moal, F. (2001). A Language-Based Similarity Measure. In : 12th European Conference on Machine Learning ECML. pp. 336-347. Freiburg, Germany.

Martín-Moreno, R., & Vega-Rodríguez, M.A. (2018). Multi-Objective Artificial Bee Colony algorithm applied to the bi-objective orienteering problem. Knowledge-Based Systems, 154, 93-101.

Matheus, C.J., Chan, P.K., & Piatetsky-Shapiro, G. (1993). Systems for Knowledge Discovery in Databases. IEEE Transactions on Knowledge and Data Engineering. 5(6), 903-913.

Matula, D.W., & Sokal, R.R. (1980). Properties of Gabriel graphs relevant to geographic variation research and clustering of points in the plane. Geographical Analysis. 12(3), 205-222.

Maulik, U., Bandyopadhyay, S. (2000). Genetic algorithm-based clustering technique. Pattern Recognition. 33(9), 1455-1465.

MCLACHLAN, G., & BASFORD, K. (1988). Mixture Models : Inference and Applications to Clustering. Marcel Dekker, New York, NY.

MCLACHLAN, G., & KRISHNAN, T. (1997). The EM Algorithm and Extensions. John Wiley & Sons, New York, NY.

Mecca, G., Raunich, S., & Pappalardo, A. (2007). A New Algorithm for Clustering Search Results. Data and Knowledge Engineering. 62, 504-522.

Mishra, G., & Mohanty, S.K. (2019). A fast hybrid clustering technique based on local nearest neighbor using minimum spanning tree. Expert Systems with Applications, 132 (2019). 28-43.

Mirkin, B. (2005). Clustering for Data Mining : A Data Recovery Approach. Chapman & Hall / CRC, BocaRaton, Florida.

Mladenovic, N., & Hansen, P. (1997). Variable neighborhood search. Computers & Operations Research. 24, 1097-1100.

Moulavi, D., Jaskowiak, P.A., Campello, R.J.G.B., Zimek, A., & Sander, J. (2014). Density-Based Clustering Validation. In : Proceedings of the 14th SIAM International Conference on Data Mining (SDM), Philadelphia, PA.

Murthy, C. A., & Chowdhury, N. (1996). In Search of Optimal Clusters using Genetic Algorithms. Pattern Recognition Letters. 17, 825-832.

NG, R., & HAN, J. (1994). Efficient and effective clustering methods for spatial data mining. In Proceedings of the 20th Conference on VLDB. pp. 144-155, Santiago, Chile.

Niknam, T., & Amiri, B. (2010). An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis. Applied Soft Computing. 10,183-197.

Omran, M., Engelbrecht, A.P., & Salman, A. (2005). Particle swarm optimization method for image clustering. International Journal of Pattern Recognition and Artificial Intelligence. 19 (3), 297-321.

Omran, M., Salman, A., & Engelbrecht, A.P. (2002). Image classification using particle swarm optimization. In : Proceedings of the Fourth Asia-Pacific Conferenceon Simulated Evolution and Learning, Singapore.

Orlov, V.I., Kazakovtsev, L.A., Rozhnov, I.P., Popov, N.A., & Fedosov, V.V. (2018). Variable neighborhood search algorithm for k-means clustering. IOP Conf. Series : Materials Science and Engineering, 450(2018) : 022035. Doi : 10.1088/1757-899X/450/2/022035.

Pal, N., & Biswas, J. (1997). Cluster validation using graph theoretic concepts. Pattern Recognition. 30 (6), 847-857.

Paterson, M.S., & Yao, F.F. (1992). On Nearest Neighbor Graphs.Automata, Languages and Programming. 623, 416-426.

Pellegrini, F. (1994). Static mapping by dual recursive bipartitioning of process and architecture graphs. IEEE.486-493.

Poli, R., Kennedy, J. & Blackwell, T. (2007). Particle swarm optimization an overview. Swarm intelligence. 1(1), 33-57.

Rahman, M.A., & Islam, M.Z. (2014). A hybrid clustering technique combining a novel genetic algorithm with K-Means. Knowledge-Based Systems, 71, 345-365.

Ram, A., Jalal, S., Jalal, A. S., & Kumar, M. (2010). A density Based Algorithm for Discovery Density Varied cluster in Large spatial Databases. International Journal of Computer Application. 3(6), 1-4.

Rand, W. M.(1971). Objective criteria for the evaluation of clustering methods. Journal of the American Statistical association. 66 (336), 846-850.

Ranjbar, M., Azami, M., & Rostammi, A.S. (2015). Fuzzy Artificial Bee Colony for Clustering. Journal of Agricultural Science and Engineering. 1(2), 46-53.

Raposo, C., Antunes, C.H., & Barreto, J.P. (2014). Automatic Clustering using a Genetic Algorithm with New Solution Encoding and Operators. Computational Science and Its Applications (ICCSA), Springer. 2, 92-103.

Rezaei, M., & Fränti, P. (2016). Set matching measures for external cluster validity. IEEE Transactions on Knowledge and Data Engineering. 28(8), 2173-2186.

Rijsbergen, C. (1979). Van Information retrieval. Butterworths, Second edition.

Rojas-Thomas, J.C., Santos, M., & Mora, M. (2017). New internal index for clustering validation based on graphs. Expert Systems with Applications. 86, 334-349.

Rousseeuw, P. (1987). Silhouettes : a graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics. 20, 53-65.

Rui, X., & Wunsch, D. (2005). Survey of clustering algorithms. IEEE Trans. Neural Netw. 16(3), 645-678.

Sabau, A.S. (2012). Variable Density Based Genetic Clustering. 14th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, pp 200-206. Doi : 10.1109/SYNASC.2012.31

Selim, S.Z., & Alsultan, K. (1991). A simulated annealing algorithm for the clustering problem. Pattern Recognition Letters. 24(10), 1003-1008.

Saif, U., Guan, Z., Zhang, L., Zhang, F., Wang, B., & Mirza, J. (2019). Multi-objective artificial bee colony algorithm for order oriented simultaneous sequencing and balancing of multi-mixed model assembly line. Journal of Intelligent Manufacturing. 30(3), 1195-1220.

Sander, J., Ester, M., Kriegel, H. P., & Xu, X. (1998). Density-Based Clustering in Spatial Databases : The Algorithm GDBSCAN and its Applications. Data Mining and Knowledge Discovery. 2(1998), 169-194.

Sarma, T.H., Viswanath, P., & Reddy, B.E. (2013). Speeding-up the kernel K-means clustering method : A prototype based hybrid approach. Pattern Recognition Letters. 34, 564-573.

Schaeffer, S. E. (2007). Survey Graph clustering. Computer Science Review. 1 (2007) 27-64.

Sheikh, R.H., Raghuwanshi, M. M., & Jaiswal, A.N. (2008). Genetic Algorithm Based Clustering : A Survey. First International Conference on Emerging Trends in Engineering and Technology, IEEE.DOI 10.1109/ICETET.2008.48.

Sheikholeslami, G., Chatterjee , S., & Zhang, A. (1998). WaveCluster : A Multi- Resolution Clustering Approach for Very Large Spatial Databases. In : Proc. 24th Int. Conf. Very Large Data Bases, VLDB, pp. 428-439. New York City, USA.

Shi, B., Han, L., & Yan, H. (2018). Adaptive clustering algorithm based on *k*NN and density. Pattern Recognition Letters.104, 37-44.

Sneath, P. H. A., & Sokal, R. R. (1973). Numerical Taxonomy - The Principles and Practice of Numerical Classification. San Francisco, W. H. Freeman and Compagny.

Sokal, R. R., & Michener, C. D. (1958). A Statistical Method for Evaluating Systematic Relationships. University of Kansas Science Bulletin. 38, 1409-1438.

Sorensen, T. (1948). A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and Its Application to Analyses of the Vegetation on Danish Commons. Biologiske Skrifter. 5, 1-34.

Sun, L.X., Xie, Y.L., Song, X.H., Wang, J.H., & Yu, R.Q. (1994). Cluster analysis by simulated annealing. Computers & Chemistry. 18(2), 103-108.

Sung, C.S., & Jin, H.W. (2000). A tabu-search-based heuristic for clustering. Pattern Recognition. 33, 849-858.

Suresh, K., Kundu, D., Ghosh, S., Das, S., & Abraham, A. (2009). Data clustering using multi-objective differential evolution algorithms. Fund. Inform. 97(4), 381-403.

Toussaint, G.T. (1980). Algorithms for computing relative neighborhood graph. Electronics Letters. 6(22) p. 860.

Tsai, C.-Y., &Chiu, C.-C. (2006). A VNS-based hierarchical clustering method. In Proceedings of the 5th WSEAS International Conference on Computational Intelligence, Man-Machine Systems and Cybernetics (CIMMACS'06), pp. 268-275.

Tsai, C., & Kao, I. (2011). Particle swarm optimization with selective particle regeneration for data clustering. Expert Systems with Applications. 38, 6565-6576.

Urquhart, R. (1982). Graph theoretical clustering based on limited neighbourhood sets. Pattern Recognition. 15 (3), 173-187.

Valafar, F. (2002). Pattern Recognition Techniques in Microarray Data Analysis : A Survey. Annals of New York Academy of Sciences. 980, 41-64.

Veenhuis, C., & Köppen, M. (2006). Data swarm clustering. Studies in Computational Intelligence. 34, 221-241.

Voorhees, E.M. (1985). The effectiveness and efficiency of agglomerative hierarchic clustering in document retrieval. Technical Report. Cornell University, Ithaca, N. Y.

Wang, Y., & Li,Y. (2015). Multi-objective Artificial Bee Colony algorithm. In 2015 International Conference on Computational Intelligence and Communication Networks.

Wang, W., Yang, J., & Muntz, R. R. (1997). STING : A Statistical Information Grid Approach to Spatial Data Mining. In : Twenty-Third International Conference on Very Large Data Bases, pp. 186-195. Athens, Greece.

Wu, J., Chen, J., Xiong, H., & Xie, M. (2009). External validation measures for K-means clustering : A data distribution perspective, Expert Systems with Applications. 36, 6050-6061.

Xie, J., Xiong, Z.-Y., Dai, Q.-Z., Wang, X.-X., & Zhang, Y.-F.(2019). A new internal index based on density core for clustering validation. Information Sciences. 506 (2020), 346-365.

Xu, X., Ester, M., Kriegal, H.-P, & Sabder, J. (1998). A Distribution Based Clustering Algorithm for Mining in Large Spatial Databases. In : 14th International Conference on Data Engineering, pp. 324-331. Orlando, FL, . *ICDE-98.*

Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. IEEE Transactions on Neural Networks. 16 (3), 645-678.

Xu, R., & Wunsch, D. C. (2009). Clustering. IEEE Press John Wiley and sons, 2nd ed.

Xu, R., Xu, J., & Wunsch, D.C. (2010). Clustering with differential evolution particle swarm optimization. IEEE Congress on Evolutionary Computation. Barcelona, Spain, pp. 1-8.

Yan, X. Zhu, Y., Zou, W., & Wang, L. (2012). A new approach for data clustering using hybrid artificial bee colony algorithm. Neurocomputing. 97, 241-250.

Yang, J., & Lee, I. (2004). Cluster validity through graph based boundary analysis. In : IKE, pp. 204-210.

Yang, X. Song, Q., & Cao, A. (2006). A new cluster validity for data clustering. Neural Processing Letters. 23(3), 325-344.

Yang, F., Sun, T., & Zhang, C. (2009). An efficient hybrid data clustering method based on K-harmonic means and Particle Swarm Optimization. Expert Systems with Applications. 36, 9847-9852.

Zahn, C. T. (1971). Graph theoretical methods for detecting and describing gestalt clusters. IEEE Transactions on Computers. 20(1), 68-86.

Zaidi, F., & Melançon, G. (2010). Evaluating the Quality of Clustering Algorithms using Cluster Path Lengths, 10th Industrial Conference (ICDM), pp. 42-56, Berlin, Germany.

Žalik, K.R., & Žalik, B. (2011). Validity index for clusters of different sizes and densities. Pattern Recognition Letters. 32, 221-234.

Zhang, D., Ji, M., Yang, J., Zhang, Y., & Xie, F. (2014). A novel cluster validity index for fuzzy clustering based on bipartite modularity. Fuzzy Sets and Systems. 253, 122-137.

Zhang, C., Ouyang, D., & Ning, J. (2010). An artificial bee colony approach for clustering. Expert Systems with Applications. 37(7), 4761-4767.

Zhang, T, Ramakrishnan, R, & Livny, M. (1996). BIRCH : An efficient data clustering method for very large databases. In proceeding of the ACM SIGMOD Conference on Management of Data, pp. 103-114.

Zhong, Y.B., Xiang, Y., & Liu, H.L. (2014). A multi-objective artificial bee colony algorithm based on division of the searching space. Applied Intelligence. 41(4), 987-1011.

Zhou, S., & Xu, Z. (2018). A novel internal validity index based on the cluster centre and the nearest neighbor cluster. Applied Soft Computing Journal. 71, 78-88. https://doi.org/10.1016/j.asoc.2018.06.033.

Zhou, S., Zhao, Y., Guan, J., & Huang, J. (2005). NBC : A Neighborhood-Based Clustering Algorithm. In : Proceedings of the 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), pp. 361-371.

Zhu, E., & Ma, R. (2018). An effective partitional clustering algorithm based on new clustering validity index. Applied Soft Computing. 71, 608-62.

Zhu, Y., Ting, K.M., & Carman M.J. (2016). Density-ratio based clustering for discovering clusters with varying densities. Pattern Recognition.
DOI : http://dx.doi.org/10.1016/j.patcog.2016.07.007

Zhu, S., Xu, L., & Goodman, E.D. (2019). Evolutionary multi-objective automatic clustering enhanced with quality metrics and ensemble strategy. Knowledge-Based Systems. https://doi.org/10.1016/j.knosys.2019.105018.

Zou, W., Zhu, Y., Chen, H., & Sui, X. (2010). A Clustering Approach Using Cooperative Artificial Bee Colony Algorithm. Discrete Dynamics in Nature and Society. Article ID 459796. https://doi.org/10.1155/2010/459796