République Algérienne Démocratique et Populaire
Ministère de L'enseignement Supérieur et de
la Recherche Scientifique

Université Mentouri de Constantine
Faculté des Sciences de l'Ingénieur
Département d'Informatique

N° d'ordre :
Série :

Mémoire

Présenté en vue de l'obtention du diplôme de Magistère en Informatique

Option : Information & Computation

Thème

Le clustering des données : une nouvelle approche évolutionnaire quantique

Présenté par :

RAMDANE CHAFIKA

Dirigé par :

Dr. S. MESHOUL

Soutenu le : / / 2006

Devant le jury d'examen composé de :

Dr. K. KHOLLADI	Maître de Conférences, Université de Constantine.	Président.
Dr. S. CHIKHI	Maître de Conférences, Université de Constantine.	Examinateur.
Dr. F. BELLALA	Maître de Conférences, Université de Constantine.	Examinateur.
Dr. S. MESHOUL	Maître de Conférences, Université de Constantine.	Rapporteur.

Remerciements

J'adresse ma gratitude sincère à mon Dieu, à chaque fois que je fait face à une difficulté, je prie Dieu de m'aider et il est toujours là me protége et me sauve.

Je voudrais exprimer mes vifs remerciements à mon encadreur *Dr Souhem Meshoul* pour son aide, sa patience et son encouragement. Sans oublier *Pr Batouche* responsable de l'équipe "Vision et Infographie" pour m'avoir accueilli dans son équipe et donné les moyens de faire ce travail dans de très bonnes conditions.

Je remercie tous les membres du jury qui ont accepté d'évaluer mon travail.

Je remercie tout le personnel du laboratoire LIRE, mes collègues et toutes les personnes qui m'ont aidé.

Un grand merci à mes parents pour tous ce qu'ils ont fait pour moi, mes sueurs et frères Wisseme, Saoussene, Asma, Mohamed Tahar, Abdenour et Selma.

Je tiens à remercier toute personne ayant contribué de près ou de loin à la réalisation de ce travail.

Liste des matières

Introduction générale

Ces dernières années les volumes de données de toutes sortes croît de plus en plus. Avec tant de données disponibles, il est nécessaire de développer des algorithmes qui peuvent extraire des informations significatives à partir de ces vastes volumes. La recherche des pépites utiles d'information parmi des quantités énormes de données est devenue connue comme le champ de fouille de données. Ce champ interdisciplinaire tire ces techniques et taches de plusieurs autres domaines. Une tache en particulier, le clustering est un processus d'analyse exploratoire qui divise les données en un ensemble de clusters, ces clusters découverts peuvent être employés pour expliquer des caractéristiques de la distribution de données sous-jacente. Le clustering permet de réduire les données en un ensemble de représentants moins nombreux permettant une représentation simplifiée des données initiales. Ainsi, le clustering est une méthode de réduction des données. Il s'agit d'une démarche très courante qui permet de mieux comprendre l'ensemble analysé. Le problème de clustering a été adressé dans de nombreux contextes et par de nombreux chercheurs dans beaucoup de disciplines, cela reflète son large appel et utilité comme une des étapes dans l'analyse de données exploratoire. Ces applications sont nombreuses, en statistique, traitement d'image, intelligence artificielle, reconnaissance des formes, l'analyse du Web, le marketing, le diagnostic médical, la biologie, et beaucoup d'autres. Pour toutes ces raisons, il constitue une tache très importante.

Le clustering est reconnu comme une tache de challenge, dont la difficulté est causée par un manque de définition unique et précise d'un cluster. Mais ce manque n'est pas un inconvénient, il a l'avantage de donner naissance à de multiples facettes du problème basées sur plusieurs définitions différentes d'un cluster. Dans la littérature, une multitude de techniques de clustering est développée. Les techniques peuvent se diffère dans leurs principes, propriétés, paramètres et formes générales du partitionnement généré. La catégorisation de ces techniques peuvent être réalisée selon plusieurs aspects : la mesure de proximité utilisée entre les données , la théorie ou les concepts fondamentales sur lesquels se basent les techniques , la nature des données manipulées et beaucoup d'autres critères. En réalité, la catégorisation de ces techniques n'est ni directe, ni canonique car les catégories se chevauchent. Les quatre catégories principales de techniques de clustering disponibles dans la littérature sont : les techniques par partitionnement, les techniques hiérarchiques, les

techniques basées sur la densité et les techniques basées sur les grilles [Berkhin,02]. Les techniques hiérarchiques construisent une hiérarchie de partitions, représentées comme un dendrogramme dans lequel chaque partition est nichée dans la partition du niveau suivant dans la hiérarchie. Les techniques par partitionnement génèrent une seule partition, avec un nombre spécifié ou estimé de clusters. Les techniques basées sur la densité manipulent les clusters comme des régions denses tandis que les techniques basées sur les grilles partitionnent l'espace de données à un nombre fini de cellules, les cellules denses sont connectées pour former les clusters.

Le nombre de façons de classer n points dans k clusters est approximatif à $k^n/k!$ [Bilmes et al,97]. Par conséquent, même avec un nombre fixe k, l'espace de recherche pour le problème de clustering s'accroît exponentiellement. Par exemple, si nous prenons n=25 et k=5, il y a 2 436 684 974 110 751 façons de répartir les 25 points dans 5 clusters [Anderberg ,73]. Il est donc évident qu'un parcours exhaustif des solutions est impossible même pour des jeux de données de taille moyenne. En effet, le problème de clustering est connu pour être NP-difficile.

Les techniques traditionnelles ne travaillent que sur un petit sous-ensemble de l'espace de recherche, elles obtiennent donc, en général, des optima locaux et rarement globaux. Pour cela une autre formulation du problème de clustering a été proposée, il peut être modélisé comme un problème d'optimisation. Les métaheuristiques d'optimisation, ayant déjà fait leurs preuves pour la résolution de problèmes combinatoires de grandes tailles, elles semblent intéressantes pour se dégager de ces optima locaux et trouver de façon plus fréquente les optima globaux.

A nos jours, le clustering est un sujet de recherches actives. Les chercheurs minent dans d'autres domaines, creusent dans la nature, s'inspirent des insectes, tentent de trouver d'autres modèles. Leurs motivations sont d'une part de tester de nouveaux algorithmes sur le problème de clustering et de connaître leurs apports, et d'autre part, de proposer de nouvelles sources d'inspiration, car le problème de clustering se rencontre souvent dans la nature.

Dans cette optique, il existe une branche qui s'inspire plus spécialement des principes issus du monde de l'infiniment petit, ce monde miraculeux des atomes et des phénomènes quantiques qui les régissent. Plus précisément, les algorithmes évolutionnaires quantiques QEAs [Han et al,02] constituent un nouveau champ de recherche qui combine les algorithmes évolutionnaires classiques et les principes de l'informatique quantique. Ce dernier est un domaine émergent fondant le calcul sur des principes issus de la mécanique quantique, tente d'apporter des solutions novatrices à différents types de problèmes informatiques non encore

résolus. Effectivement, les principes quantiques tels que la superposition, l'enchevêtrement, l'interférence, et autres ont doté l'informatique quantique une capacité de traitement et de stockage exponentielles au sein d'une machine quantique. Cependant, la construction de machines quantiques est toujours en état de recherche. En plus, l'écriture d'un algorithme quantique pur est une tache dure. Cela est témoigné par l'existence de peu d'algorithmes quantiques purs mais qui ont prouvé leur supériorité à leurs homologues classiques, comme l'algorithme de factorisation des nombres de Shor [Shor, 94]. Cet algorithme a une complexité polynomiale or le meilleur algorithme classique a une complexité exponentielle. Un autre algorithme très important est l'algorithme de Grover [Grover, 96] pour la recherche d'une donnée dans une liste désordonnée de données. Cet algorithme trouve l'élément recherché dans un temps quadratique. Cela a incité les chercheurs à combiner les algorithmes classiques et les principes de l'informatique quantique afin de tirer profit de cette puissance quantique. Contrairement aux algorithmes quantiques purs, les algorithmes inspirés du quantique ne nécessitent pas la présence des machines quantiques. Les algorithmes inspirés du quantique se sont montrés efficaces pour de nombreux problèmes d'optimisation combinatoire.

Dans le cadre de ce travail, nous nous sommes attachés à montrer l'adéquation des algorithmes évolutionnaires quantiques au problème de clustering des données. Pour réaliser cet objectif, il nous a fallu en premier lieu de faire une recherche exploratoire des techniques et algorithmes de clustering existants et de mener une étude de synthèse analytique et comparative pour pouvoir extraire tout ce que peut constituer un algorithme de clustering.

En second lieu, c'est le développement d'une nouvelle approche évolutionnaire quantique pour le clustering des données qui repose sur une représentation quantique appropriée et une dynamique évolutionnaire quantique adaptée. La particularité de cette approche provient de plusieurs aspects. Tout d'abord, la nature quantique de la représentation fournit une condensation d'un grand nombre d'individus au sein d'un seul individu quantique. En plus, cette représentation est considérée comme ouverte, vu qu'elle s'ajuste bien à d'autres extensions prometteuses. L'initialisation de la population n'est pas faite aléatoirement, mais elle est fondée sur une nouvelle fonction bien puissante. Pour mettre en œuvre l'approche, un effort particulier a été effectué dans le choix et la conception des opérateurs quantiques et de la fonction objective optimisée. La dynamique quantique réduit la taille de la population et le nombre d'itérations nécessaire. Comme support de notre étude, nous avons testé notre approche sur des problèmes de références réels et d'autres synthétiques qui ont prouvé de bon résultats.

Ce mémoire s'articule en quatre chapitres.

- Le premier chapitre est une introduction au domaine d'extraction de connaissances à partir de données, dans le quel on a visé le problème de clustering des données et le rôle important qu'il joue pour extraire des connaissances.
- Le deuxième chapitre présente une étude de synthèse des différents techniques et algorithmes de clustering des données.
- Le troisième chapitre présente une introduction à l'informatique quantique et aux algorithmes évolutionnaires quantiques.
- Le dernier chapitre est consacré à l'approche proposée. Il présente son principe, la méthodologie de résolution sous-jacente ainsi que les résultats de l'évaluation de ses performances.

Nous terminerons ce mémoire par différentes perspectives de recherche qui nous semblent intéressantes pour continuer ce travail.

Chapitre 1

Extraction de connaissances et Clustering

"We are drowning in information, but starving for knowledge."
— John Naisbett

1.1 Introduction

La quantité d'informations que détiennent et échangent les opérateurs et acteurs dans le vie quotidienne est colossale et son volume augmente de plus en plus notamment avec le développement technologique des dispositifs de stockage et la baisse des coûts. Certaines entreprises doivent gérer des millions de transactions par jour, stockées dans des bases de données de plusieurs tera-octets ; de plus on estime que la quantité de données collectées par les entreprises double tous les 9 mois [Goethals et al,03]. Au fond de cette masse gigantesque de données qui ne cesse d'accroître se cachent des connaissances parfois d'importance stratégique qu'il faut impérativement les découvrir automatiquement afin de les intégrer dans le processus décisionnel de l'entreprise. En effet, bien que très riche en données, le monde est pauvre en connaissances. Les techniques d'analyse statistiques traditionnelles ne traitent que les données de taille limitée et son utilisation exige un bagage statistique important de la part de l'utilisateur, ainsi par manque ou insuffisance d'outils permettant d'extraire les connaissances utiles à partir des données brutes, les bases de données sont inexploitables, et les décisions sont plus souvent prises sur la base d'intuitions que de connaissances valables.

Dans cette optique, on a vu émerger un nouveau domaine d'étude qui s'inscrit parfaitement dans ce créneau, il s'agit de l'Extraction de connaissances à partir de données (ECD), une discipline émergente et multifocale, rassemblant les travaux des chercheurs en statistiques, intelligence artificielle, apprentissage automatique, reconnaissance de formes, bases de données, visualisation des données, linguistique et beaucoup d'autres (figure1.1). L'ECD est un processus qui englobe le stockage et la préparation des données, l'analyse de celles-ci par différentes techniques, et enfin, l'interprétation et l'évaluation des connaissances acquises. Ici, le principe est, idéalement, à partir de données dont on ne sait rien et sur lesquelles on ne fait aucune hypothèse, d'obtenir des informations pertinentes, et à partir de celle-ci de découvrir de la connaissance.

Aujourd'hui, le gouffre qui sépare données et connaissances est peu à peu comblé par le développement des techniques de fouille de données. Ce terme réfère à une étape du

processus d'ECD, mais sûrement la plus importante. Cette étape permet d'obtenir des informations à partir de données réarrangées et préparées. Par contre, il est nécessaire de passer par une étape d'évaluation avec l'aide d'un expert du domaine afin de relever la pertinence de ces informations et de leur éventuel apport pour la connaissance de l'entreprise.

Il existe plusieurs taches de fouille de données, dont le clustering est la tache la plus largement utilisée. Il est dans un cadre d'apprentissage non supervisé, qui a pour but de d'obtenir des informations sans aucune connaissance préalable, au contraire de l'apprentissage supervisé. Il a plusieurs possibilités de combinaison avec d'autres taches, en pre- ou en post-traitement. En effet, il peut résumer l'information afin de la transmettre à une autre tache et ainsi mieux analyser les données. Grouper des objets ensemble selon certaines caractéristiques, afin de dissocier différents ensembles est un processus de clustering. Ce processus naturel à l'homme permet, à partir d'un certain nombre de données et de règles, de diviser un ensemble d'objets en différentes classes, distinctes et homogènes. Le principe de clustering est d'imiter ce mécanisme par une machine. Pour ceci, il faut développer des méthodes qui s'appuient sur les données à proprement parlé, sans aucune connaissance autre. Le clustering a de multiples applications, et dans des domaines très diverses. En économie, le clustering peut aider les analystes à découvrir des groupes distincts dans leur base clientèle, et à caractériser ces groupes de clients, en se basant sur des habitudes de consommations. En biologie, on peut l'utiliser pour dériver des taxinomies de plantes et d'animaux, pour catégoriser des gènes avec une ou plusieurs fonctionnalités similaires, pour mieux comprendre les structures propres aux populations. Le clustering peut tout aussi bien aider dans l'identification des zones de paysage similaire, utilisée dans l'observation de la terre, et dans l'identification de groupes de détenteurs de police d'assurance automobile ayant un coût moyen d'indemnisation élevé, ou bien dans la reconnaissance de groupes d'habitation dans une ville, selon le type, la valeur et la localisation géographique. Il est possible aussi de classifier des documents sur le Web, pour obtenir de l'information utile. Sa fonction dans un processus de fouille de données sera de découvrir la distribution des données, d'observer les caractéristiques de chaque groupe et de se focaliser pourquoi pas sur un ou plusieurs ensembles particuliers d'objets pour de prochaines analyses. Parallèlement, il peut servir comme une étape de préparation de données, pour d'autres taches. Ainsi, ce type d'analyse est devenu un domaine hautement actif dans la recherche de type fouille de données.

Dans ce chapitre nous allons présenter une petite introduction à l'extraction de connaissances à partir de données en citant son processus et en insistant sur l'étape importante

de fouille de données. Ensuite, nous allons focaliser une tache de cette étape de fouille de données, qui est la tache de clustering des données. Nous allons situer le clustering en présentant son importance, son domaine large d'applications ainsi que les problèmes qu'il rencontre.

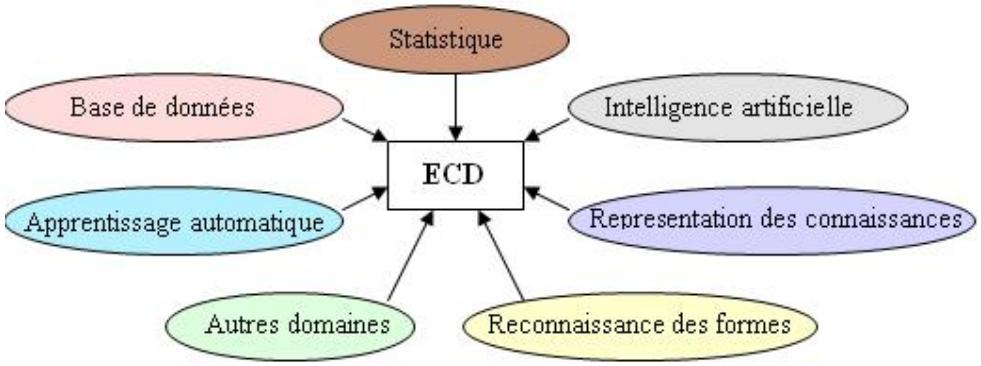


Figure 1.1. ECD à la confluence de nombreux domaines

1.2 Définitions de l'extraction de connaissances à partir de données (ECD)

- **Définitions 1**

L'ECD est un processus non trivial qui consiste à identifier, dans les données, des modèles nouveaux, valides, potentiellement utiles et surtout compréhensibles et utilisables. [Fayyad et al, 96a].

- **Définitions 2**

L'ECD est un processus itératif et interactif d'analyse d'un grand ensemble de données brutes afin d'en extraire des connaissances exploitables par un utilisateur-analyste qui y joue un rôle central [Zighed et al,01].

D'après ces deux définitions, l'utilisateur fait partie intégrante du processus. L'interactivité est liée aux différents choix que l'utilisateur est amené à effectuer. L'itérativité est liée au fait que l'ECD soit composée de plusieurs phases et l'utilisateur peut décider de revenir en arrière à tout moment si les résultats ne lui conviennent pas.

1.3 Le processus de l'ECD

Le processus de l'ECD est un processus interactif et itératif, impliquant de nombreuses étapes avec beaucoup de décisions faites par l'utilisateur [Fayyad et al, 96b]. Les phases constituant le processus de l'ECD sont :

- La compréhension du domaine d'application
- La création du jeu de données cible
- Le nettoyage et le prétraitement des données
- La réduction et la transformation des données
- La fouille de données
- L'interprétation et l'évaluation
- L'intégration de la connaissance

La figure 1.2 récapitule ces différentes phases ainsi que les enchaînement possibles entre elles. Cette séparation est théorique. En pratique, ce n'est pas toujours le cas. En effet, dans de nombreux systèmes, certains de ces étapes sont fusionnées [Lefébure et al, 01] [Courtier, 05].

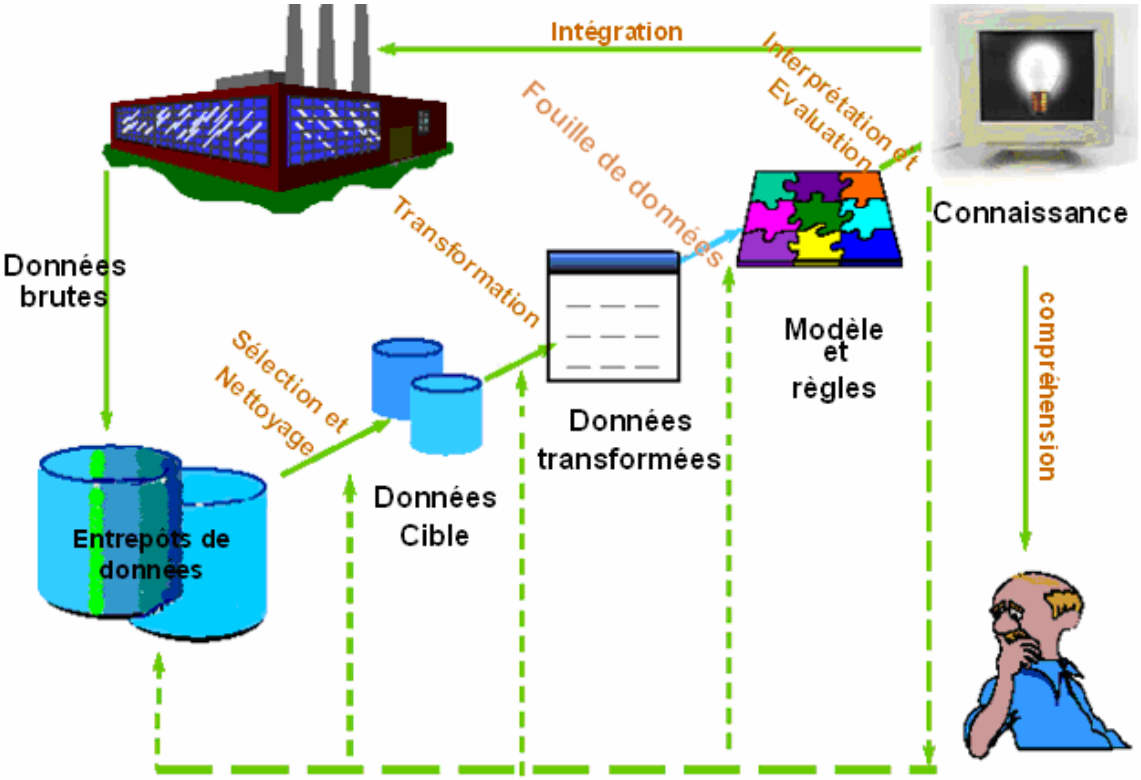


Figure 1.2. Processus d'extraction de connaissances

Ici nous citons les étapes de base :

1.3.1 Compréhension du domaine d'application

Cette première phase est celle où l'on expose le problème et où l'on définit les objectifs, le résultat attendu ainsi que les moyens de mesurer le succès du processus de l'ECD. Il s'agit de comprendre le contexte de la recherche en vue de donner une signification logique aux variables. Dans cette phase introductive, il est intéressant de recueillir les intuitions et la connaissances des experts afin d'orienter le processus de découverte ou tout simplement pour identifier les variables les plus pertinentes susceptibles d'expliquer les phénomènes analysés [Lefébure et al, 01].

1.3.2 Création du jeu de données cible

Il s'agit dans cette phase de déterminer la structure générale des données ainsi que les règles utilisées pour les constituer. Il faut identifier les informations exploitables et vérifier leur qualité. La recherche d'une sélection optimale des données est un point central dans le processus d'ECD. Cette sélection nécessite souvent l'aide d'expert du domaine pour déterminer les attributs les plus aptes à décrire la problématique. De tels experts sont capables d'indiquer les variables qui ont une influence sur le problème à résoudre. Il est important, dans cette phase, de prendre connaissance d'éléments du contexte qui permettent de construire une représentation préliminaire du problème. Par apport à une approche classique de type système expert, on ne demande pas à l'expert d'organiser son processus d'analyse mais de lister ce qui, selon lui, a une importance.
Si les experts ne sont pas disponibles, une recherche des facteurs les plus déterminants est entreprise par d'autres techniques d'analyses [Lefébure et al, 01].

1.3.3 Nettoyage et prétraitement des données

Il faut éviter le piège GIGO (Garbage In, Garbage Out) dans lequel les erreurs en entrée entraînent des erreurs en sortie, cela est tout à fait applicable à l'extraction de connaissances, où la qualité du modèle final dépend grandement de la qualité des données. Il est donc particulièrement important de gérer leur manque de fiabilité. Il y a quelques types de problèmes de qualité de données : certaines données peuvent être absentes et gêner ainsi l'analyse. Il faut donc définir des règles pour gérer ou pour remplacer ces données manquantes. De nombreuses solutions sont proposées et plusieurs techniques sont possibles [Han et al, 06], [Zighed et al,02] :

- Ignorer tout simplement le tuple dont une valeur manque.
- Remplir les valeurs manquantes manuellement.
- Utiliser une constante globale pour remplir les valeurs manquantes.
- Remplir la valeur manquante par la moyenne ou la médiane des valeurs de la variable.
- Estimer ces valeurs manquantes par des méthodes d'induction comme la régression, les réseaux de neurones simples ou multicouches, ou les graphes d'induction.

Un bruit est une erreur ou une variation aléatoire dans une variable dû à une défaillance des instruments de collection des données ou à des problèmes de saisie et de transmissions de données [Han et al, 06]. Pour le traitement des données aberrantes, il faut d'abord repérer ces dernières au moyen de quelques méthodes dont la plus usuelle consiste à définir un espace entre la moyenne et un certain nombre d'écart types, puis à exclure ou à plafonner toute les valeurs se trouvant à l'extérieur de cet intervalle [Lefébure et al, 01].

1.3.4 Réduction et transformation des données

La réduction des données vise à obtenir une représentation réduite du jeu de données qui a un volume beaucoup plus petit mais produit les mêmes ou presque les mêmes résultats analytiques [Han et al, 06]. Elle s'effectue sur des données qui sont déjà sous forme tabulaire. Il s'agit ensuite de définir un filtre qui permet de sélectionner un sous-ensemble de lignes ou de colonnes .L'objectif est soit de réduire le nombre de données soit de sélectionner les lignes ou colonnes les plus pertinentes par rapport aux préoccupations de l'utilisateur [Zighed et al, 02]. Les techniques mises en œuvre dans ce but relèvent des méthodes statistiques d'échantillonnage, de sélection d'instances ou de sélection d'attributs. Cette sélection peut également s'effectuer selon des conditions exprimées par l'utilisateur. Les techniques couramment utilisées sont :

- Méthodes de test de corrélation telle que le test de Khi2 pour éliminer les variables indépendantes du phénomène à expliquer.

- Analyse en composante principale (ACP)

  Etant donnée N vecteurs de données de K dimensions, l'ACP consiste à trouver C vecteurs orthogonaux qui peuvent être les meilleurs employés pour représenter des données (C<=K).

La transformation des données consiste à transformer un attribut A en une autre variable A' qui serait, selon les objectifs de l'étude, plus appropriée. Différentes méthodes sont pratiquées comme la discrétisation et la normalisation des données:

La discrétisation consiste à transformer des attributs continus en découpant le domaine de valeurs de ces attributs en intervalles afin d'obtenir des attributs qualitatifs. Il existe à cet effet une pléthore de méthodes de discrétisation : supervisées ou non, à intervalles de tailles identiques, ou à intervalles à effectifs constants.

La normalisation permet de réduire l'échelle de grandeur de variables. Il existe plusieurs manières pour normaliser les données, par exemple, on peut centrer par rapport à la moyenne et réduire par l'écart type les valeurs des variables continues. Ce traitement leur confère certaines propriétés mathématiques intéressantes lors de la mise en œuvre de méthodes d'analyse des données multidimensionnelles [Zighed et al, 02].

1.3.5 Fouille de données

La fouille de données est au cœur du processus d'ECD, il se réfère à une série d'activités comme le choix du type de la tache de fouille de données; la sélection de la technique de fouille de données; le choix de l'algorithme de fouille de données; et l'extraction des modèles. D'abord, le type de la tache de fouille de données doit être choisi. Les taches de la fouille de données qui peuvent être distinguées sont le clustering, la classification, la régression, l'analyse des associations ,le résumé des données et la détection de déviation. Basée sur la tache choisie pour l'application, une technique de fouille de données appropriée est alors choisie. Une fois qu'une technique de fouille de données est choisie, l'étape suivante est de choisir un algorithme particulier de la technique de fouille de données choisie. Le choix d'un algorithme de fouille de données inclut une méthode pour chercher les modèles dans les données, cette décision d'algorithme et de paramètres appropriés doit apparier la technique particulière de fouille de données à l'objectif global de l'ECD [Guo, 01]. D'après [Wolpert et al, 97], tout le problème de fouille de données réside dans le choix de la technique adéquate à un problème donné. Il est également possible de combiner plusieurs techniques pour essayer d'obtenir une solution optimale globale.

1.3.6 Interprétation et évaluation

Cette étape d'interprétation et d'évaluation des modèles découverts inclut le filtrage d'information à être présentée en enlevant les modèles redondants ou non pertinents, en visualisant les modèles utiles, et en les traduisant en termes compréhensibles par des

utilisateurs. Dans l'interprétation de résultats, on détermine et résout des conflits potentiels avec la connaissance précédemment trouvée ou on décide de refaire n'importe laquelle des étapes précédentes. La connaissance extraite est également évaluée en terme de son utilité à un décideur et à l'objectif de l'application. La connaissance extraite est par la suite employée pour supporter la prise de décision humaine telle que la prédiction et pour expliquer des phénomènes observés.

1.3.7 Intégration de la connaissance

La connaissance ne sert à rien tant qu'elle n'est pas convertie en décision puis en action. Cette phase d'intégration de la connaissance consiste à implanter le modèle ou les résultats dans les systèmes informatiques ou dans les processus de l'entreprise. Elle est donc essentielle, puisqu'il s'agit de la transition du domaine des études au domaine opérationnel.

1.4 Fouille de données : taches et techniques

Les deux buts primaires de la fouille de données tendent dans la pratique à être la prédiction et la description [Fayyad et al, 96b]. La prédiction implique l'utilisation de quelques variables ou valeurs de données pour prévoir les valeurs inconnues ou futures des variables d'intérêt et la description focalise la découverte des modèles interprétables par les humains décrivant les données. Ces deux buts de prédiction et de description peuvent être réalisés en utilisant une variété de taches de fouille de données dont on peut distinguer entre le clustering, l'analyse des associations, le résumé de données et la détection des déviations qui génèrent des modèles descriptifs et la classification et la régression qui génèrent des modèles prédictifs.

1.4.1 Classification

La classification se fait naturellement depuis déjà bien longtemps pour comprendre et communiquer notre vision du monde (par exemple les espèces animales, minérales ou végétales). La classification emploie des classes de références prédéfinies pour ordonner les objets dans une collection de données, elle consiste à examiner des caractéristiques d'un objet de données nouvellement présenté afin de l'affecter à une classe de l'ensemble de classes de références, elle s'inscrit dans le cadre de la découverte de connaissances supervisée. Le fonctionnement de la classification se décompose en deux phases. La premiere est la phase d'apprentissage où les approches de classification utilisent normalement un jeu d'apprentissage où tous les objets sont déjà associés aux classes de références connues. L'algorithme de classification apprend du jeu d'apprentissage et construit un modèle. La

seconde phase est la phase de classification proprement dite, dans laquelle le modèle est employé pour classifier de nouveaux objets [Zaïane, 99].

Il existe différentes techniques parmi lesquelles on trouve les arbres de décisions, la classification bayésienne, les réseaux de neurones, les k-plus proches voisins, les algorithmes génétiques..etc. Chacune de ces techniques possède des atouts et des limites. Il faut préalablement analyser la problématique pour choisir la technique dont les critères sont les mieux adaptés [Courtier, 05].

1.4.2 Analyse des associations et de motifs séquentiels

La recherche de règles d'associations est liée aux travaux d' Agrawal et al qui ont été amenés à travailler sur une application de vente de produits dans les supermarchés [Agrawal et al, 1994]. Cette application est également appelée "analyse du panier de la ménagère" et elle est à l'origine des règles d'associations. Il s'agit d'obtenir des corrélations de type "Si Condition alors Résultat". Pour ce problème, chaque panier n'est significatif que pour un client en fonction de ses besoins et de ses envies, mais si le supermarché s'intéresse à tous les paniers simultanément, des informations utiles peuvent être trouvées et exploitées . Tous les clients sont différents et achètent des produits différents, en qualités différentes et à des périodes différentes. L'analyse du panier de la ménagère étudie qui sont les clients et pourquoi ils effectuent tel ou tel type d'achat. Elle permet d'étudier quels produits tendent à être achetés en même temps et lesquels seront les mieux adaptés à une campagne promotionnelle. Bien que cette méthode soit initialement prévue pour le secteur de la distribution, elle peut être appliquée à d'autres domaines.

La recherche de motifs séquentiels est basée sur le même principe que la recherche de règles d'association en ajoutant toutefois une idée de temps "Si Condition alors Résultat dans X intervalle de temps".

Le principal intérêt de cette méthode est que les résultats sont clairs et facilement interprétables pour un expert car ils sont exprimés sous la forme d'implication mais le nombre de règles d'association peut varier de plusieurs dizaines de milliers à plusieurs millions [Courtier, 05]. Le problème de la pertinence et de l'utilité des règles demeure un problème majeur. Il est lié au nombre de règles extraites, qui en général, est très important, et à la présence de règles redondantes [Zaki, 04].

Le premier algorithme d'extraction de règles d'association est l'algorithme Apriori [Agrawal ,93], [Agrawal ,94], plusieurs algorithmes optimisant l'efficacité de l'algorithme Apriori ont été proposées. Des métaheuristiques ont été appliquées pour accomplir cette tache tel que

l'algorithme génétique de recherche de règle d'associations ASGARD décrit dans [Jourdan, 03].

1.4.3 Le résumé de données

Le résumé de données est une tache qui consiste à trouver une description compacte pour un ensemble de données. Le résumé peut être accompli en représentant les données au moyen d'indicateurs statistiques (l'écart type, la variance, la médiane..etc) [Fayyad et al, 96,1]. D'autres méthodes plus sophistiquées impliquent la dérivation des règles récapitulatives ont été proposés dans [Agrawal et al, 96]. Le résumé de données peut se concrétiser selon un large éventail de formes, mais intuitivement, les processus de résumé de données tentent de produire des règles valides sur les données en général, ou un sous-ensemble précis d'entres-elles. Une telle règle pourrait par exemple s'exprimer ainsi : "Les personnes dans la trentaine et dont le revenu est confortable sont souscripteurs de crédit immobilier".

Un autre type de méthodes basées sur la compression sémantique existe. Il s'agit ici d'utiliser la nature sémantique des données à résumer pour en fournir une représentation plus concise. Les techniques mises en jeu cherchent donc à fournir une représentation en intention d'un ensemble d'observations. Le travail décrit dans [Saint-Paul, 05] utilise cette compression sémantique ainsi que la théorie des sous-ensembles flous. Les résumés produits proposent une description d'un sous-ensemble de la base initiale au moyen d'un ensemble de descripteurs linguistiques flous.

1.4.4 La détection des déviations

Cette tache focalise la découverte des déviations intéressantes. Une "déviation" caractérise le fait qu'un sous-ensemble d'éléments de la même catégorie présente un comportement différent de l'ensemble de sa catégorie. La détection de déviation est habituellement effectuée après la segmentation de la base de données pour déterminer si les déviations représentent des données bruyantes ou un accident peu commun[Kodratoff, 98].

Parmi les approches proposées pour la détection des déviations, il existe celles qui utilisent quelque forme d'analyse basée sur des règles. L'analyse basée sur les règles se fonde sur des ensembles de règles prédéfinies qui sont fournies par un administrateur, automatiquement créé par le système, ou tous les deux. Les systèmes experts sont la forme la plus commune d'approches de détection des déviations basées sur les règles.

Les systèmes basés sur les règles souffrent d'une incapacité de détecter des déviations qui peuvent arriver au cours d'une période prolongée de temps.

Les réseaux de neurones artificiels sont aussi utilisés dans des systèmes de détection des déviations du comportement typique et la détermination de la similarité des événements à ceux qui sont indicatifs d'une attaque [Guo, 01].

1.4.5 Régression

La régression consiste à prévoir le comportement d'une variable par rapport à ses attitudes passées. Les deux taches de classification et régression sont utilisées pour la prédiction. La distinction entre les deux est que la variable de sortie de la classification est catégorique, tandis que celle de régression est numérique et continue. Les mêmes techniques peuvent être utilisées dans la régression et la classification comme par exemple les réseaux de neurones [Guo, 01].

1.4.6 Clustering

Le clustering consiste à trouver des groupes homogènes au sein d'une population. Il s'agit de regrouper les données ayant un haut degré de similarité au sein de groupes ou clusters. Contrairement à la classification supervisée, le clustering est une tâche d'apprentissage "non supervisée" car on ne dispose d'aucune autre information préalable que la description des données. Cette tache de clustering consiste la thématique centrale de ce mémoire. Elle sera présentée plus en détails dans le chapitre 2.

1.5 Le clustering pour la fouille de données

Le clustering est une division de données en groupes d'objets similaires. La représentation des données par quelques clusters perd nécessairement certains détails fins, mais réalise la simplification. Il modélise les données par ses clusters. La modélisation de données met le clustering dans une perspective historique enracinée dans les mathématiques, les statistiques et l'analyse numérique. D'une perspective d'apprentissage, les clusters correspondent aux connaissances et modèles cachés, la recherche de clusters est un apprentissage unsupervisé. D'une perspective pratique, il joue un rôle exceptionnel dans des applications de fouille de données telles que l'exploration de données scientifique, la fouille de texte, les applications de base de données spatiales, l'analyse du Web, le marketing, le diagnostic médical, la biologie , et beaucoup d'autres [Berkhin,02] . Il a plusieurs possibilités de combinaison avec d'autres taches, en pre- ou en post-traitremnt. En effet, il peut résumer l'information afin de la transmettre à une autre tache et ainsi mieux analyser les données. Le clustering est un

processus naturel à l'homme, son automatisation exige le développement des algorithmes qui s'appuient soit sur les données à proprement parlé, sans aucune connaissance autre, soit sur les données et sur un savoir acquis préalablement (automatiquement ou grâce à un expert du domaine) [Jollois,03].

Les caractéristiques désirées d'un algorithme de clustering dépendent du problème particulier à étudier. La liste suivante représente des problèmes à surmonter et des caractéristiques désirées et des exigences à satisfaire pour le clustering dans le domaine de la fouille de données.

1.5.1 Problèmes typiques et caractéristiques désirables

Ces caractéristiques incluent [Kumar, 00] :

- La scalabilité (scalability):

Les techniques de clustering pour les grands jeux de données doivent prendre en compte les deux facteurs temps et espace. Plusieurs bases de données peuvent contenir des millions d'enregistrements, et ainsi, n'importe quel algorithme de clustering utilisé devrait avoir une complexité temporelle linéaire ou presque linéaire pour manipuler de tels grands jeux de données (Même les algorithmes qui ont la complexité de $O(m^2)$ ne sont pas pratiques pour des grands jeux de données).

- Indépendance de l'ordre de données en entrée :

Quelques algorithmes de clustering dépendent de l'ordre de données en entrée, c.-à-d., si l'ordre dans lequel les points de données sont traités change, alors les clusters résultants peuvent changer. C'est désagréable puisqu'il met en question la validité des clusters qui ont été découverts.

- Moyens efficaces pour détecter et traiter le bruit ou points isolées :

Un point bruit ou simplement un point atypique peut déformer un algorithme de clustering. En appliquant des tests qui déterminent si un point particulier appartient vraiment à un cluster donné, quelques algorithmes peuvent détecter les bruits et supprimer ou éliminer leurs effets négatifs. Ce traitement peut arriver ou bien tandis que le processus clustering a lieu ou bien comme une étape de post traitement.

Cependant, dans quelques cas, des points ne peuvent pas être rejetés, ils doivent être regroupés. Dans de tels cas, il est important de s'assurer que ces points ne déforment pas le processus de clustering pour la majorité des points.

Ø Moyens efficaces pour évaluer la validité des clusters qui sont produit :

Il est commun pour des algorithmes de clustering de produire des clusters qui ne sont pas de "bons" clusters quand ils sont évalués plus tard.

Ø Interprétation facile de résultats :

Beaucoup de méthodes de clustering produisent des descriptions de cluster qui sont juste des listes des points appartenant à chaque cluster. Tels résultats sont souvent difficile à interpréter. Une description d'un cluster comme une région peut être beaucoup plus compréhensible qu'une liste de points. Ceci peut prendre la forme d'un point de centre avec un rayon.

En outre, le clustering de données est parfois précédé par une transformation de l'espace initial de données, souvent à un espace avec un nombre réduit de dimensions. Ceci peut être utile pour trouver des clusters mais il peut rendre les résultats très difficiles à interpréter.

Ø La capacité de manipuler des distances dans des espaces à grand nombre de dimensions.

Ø Robustesse en présence de caractéristiques différentes de données et de cluster:

Une technique robuste de clustering doit tenir compte des caractéristiques des données et des clusters suivantes :

- la dimensionnalité
- les bruits
- la distribution statistique : certaines données ont une distribution gaussienne (normale) ou uniforme, et autres.
- la forme de cluster : les clusters peuvent prendre des formes différentes par exemple, rectangulaires, globulaires, convexes ou irrégulièrement formés.
- la taille de cluster : certaines méthodes de clustering, par exemple, Kmeans ne fonctionnent pas bien en présence de clusters de tailles différents.
- la densité de cluster : certaines méthodes de clustering, par exemple, Kmeans ne fonctionnent pas bien en présence de clusters de densités différents.

- la séparation de cluster : dans certains cas les clusters sont bien séparés, mais dans beaucoup d'autres cas les clusters peuvent se toucher ou se superposer.
- type d'espace de données, par exemple, euclidien ou non-euclidean : quelques techniques de clustering calculent les moyennes, ou emploient d'autres opérations vectorielles qui ont un sens seulement dans l'espace euclidien.
- beaucoup de types et de types mixtes d'attributs, par exemple, (temporel, continu, catégorique) : un mélange de types d'attribut est habituellement manipulé par une fonction de proximité qui peut combiner tous les différents attributs d'une façon "intelligente".

Beaucoup de techniques de clustering ont des suppositions sur des caractéristiques de données et de clusters et elles fonctionnent mal si ces suppositions sont violées. Dans de tel cas la technique de clustering peut manquer des clusters, fragmenter des clusters, fusionner des clusters ou juste produire de mauvais clusters

Ø La capacité d'estimer tous les paramètres :

(Par exemple, le nombre de clusters, la taille des clusters, ou la densité des clusters). Beaucoup d'algorithmes de clustering prennent le nombre de clusters comme un paramètre. Cela peut être une particularité utile dans de certains cas, comme par exemple l'utilisation de clustering pour la compression. Il est possible de déterminer empiriquement le meilleur nombre de clusters mais cela augmente la quantité de calcul requise.

Ø La Capacité de fonctionner d'une manière incrémentale :

Dans certains cas, les données utilisées pour le clustering initial peuvent changer à travers le temps. Si l'algorithme de clustering peut manipuler l'ajout de nouvelles données ou la suppression des anciennes données, d'une manière incrémentale, alors cela est beaucoup plus efficace que la re-exécution de l'algorithme sur le nouveau jeu de données.

1.5.2 Applications du clustering à la fouille de données

La fouille de données peut être appliquée aux bases de données relationnelles, transactionnelles, spatiales et également aux grands stocks de données non structurées telles que le World Wide Web. Il existe beaucoup d'applications et systèmes de fouille de données en service aujourd'hui.

La fouille de données, comme le clustering, est une activité exploratoire, ainsi des méthodes de clustering sont bien appropriées pour la fouille de données. Le clustering est souvent une étape initiale importante de plusieurs processus de fouille de données [Jain et al, 99].
Certaines des approches de fouille de données qui utilisent le clustering sont les suivant :

- La segmentation :

Des méthodes de clustering sont utilisées dans la fouille de données pour segmenter des bases de données en groupes homogènes. Cela peut servir les buts de compression de données en travaillant avec des groupes plutôt que d'objet de données individuels, ou d'identifier les caractéristiques des sous populations qui peuvent être visées pour des buts spécifiques. Le clustering est également utilisé pour segmenter des images et en extraire des informations utiles .Comme dans [Faber et al, 94] où l'algorithme de clustering Kmeans a été employé pour grouper des pixels dans des images satellitaires de la terre (Landsat images). Chaque pixel a initialement 7 valeurs de différentes bandes de satellites, y compris l'infrarouge. Il est difficile pour les humains d'assimiler et analyser ces 7 valeurs sans aide. Les pixels avec les 7 attributs de valeurs sont groupés dans 256 clusters, et à chaque pixel est assigné la valeur du centroid du cluster. L'image peut alors être affichée avec l'information spatiale intacte. Les visualisateurs humains peuvent regarder une image simple et identifier une région d'intérêt (par exemple, une route ou une forêt) et l'étiqueter comme un concept. Le système identifie alors des pixels dans le même cluster comme une instance d'un concept.

- La visualisation

Les clusters dans de grandes bases de données peuvent être employés pour la visualisation, afin d'aider les analystes humains à identifier les groupes et les sous-groupes ayant des caractéristiques semblables.
WinViz [Lee, 96] est un outil de fouille de données de visualisation dans lequel les clusters tirés peuvent être exportés comme des nouveaux attributs qui peuvent alors être caractérisés par le système. Par exemple, les céréales de petit déjeuner sont groupées selon des calories, la protéine, la graisse, le sodium, la fibre, l'hydrate de carbone, le sucre, le potassium et les vitamines contenu dans une portion. En voyant les clusters résultants, l'utilisateur peut exporter les clusters à WinViz comme des attributs. Le système montre qu'un des clusters est caractérisé par un haut contenu de potassium et l'analyste humain reconnaît les individus dans

le cluster comme appartenant à la famille de céréale de "son", ce qui mène à une généralisation que "des céréales de son sont riches de potassium".

- La fouille du web et la fouille du texte

Le WWW continue à se développer à une cadence étonnante comme une porte de l'information et comme un moyen pour la conduite des affaires. La fouille du web est l'extraction de connaissance intéressante et utile et de l'information implicite d'artefacts ou l'activité liée au WWW [Cooley, 00]. La fouille du texte est aussi l'analogue de la fouille de données mais pour le traitement de données textuelles.

Beaucoup de systèmes et applications de Web exigent souvent l'utilisation d'un algorithme de clustering. Par exemple, on peut définir un site portail comme une division hiérarchique d'un ensemble de documents.

Les outils de recherche disponibles actuellement offrent des possibilités de recherche basées essentiellement sur des mots clés. Cette formulation de requête limite les moteurs de recherche et les réponses qu'ils apportent sont généralement peu précises, même pour des requêtes bien détaillées. Les sites portails peuvent êtres considérés comme une bonne alternative. Ce sont des outils efficaces lorsque l'utilisateur désire une information d'un certain type ou d'un certain sujet.

La conception de chaque outil de recherche ou site portail doit commencer par la collecte de documents à indexer. Il faut ensuite extraire des pièces d'information à partir des documents trouvés (titres, mots clés, etc.) et enfin, présenter le site portail avec une classification hiérarchique de ces documents basée sur la similarité entre les textes. D'une manière plus générale un site portail peut être vu comme une classification hiérarchique d'un ensemble de documents en catégories et sous-catégories, de sorte que chaque sous-catégorie soit la plus similaire possible à sa catégorie et la plus dissimilaire possible aux autres.

Anttree et ces variantes [Azzag et al, 04] [Azzag et al, 06] sont des exemples d'algorithmes de clustering qui construisent de manière automatique une hiérarchie tout en classant de façon arborescente des pages web.

Un autre algorithme de clustering Ωmeans inspiré de Kmeans [Christophe,04] permet de classer un grand nombre de documents. Il a été intégré dans un système de navigation basé sur l'idée que certains systèmes de recherche d'information intègrent une méthode de clustering pour présenter les éléments par ordre de pertinence de clusters ou bien pour permettre à l'utilisateur de choisir parmi une liste de clusters, sous forme d'ensembles de mots, pour une expansion de la requête.

- La bioinformatique

Plusieurs solutions de la fouille de données ont été présentées pour la bioinformatique. Le clustering a reçu une attention significative dans ce domaine [Bellaachia et al, 02]. Au cours des années il a été utilisé dans beaucoup de secteurs s'étendant de l'analyse d'information clinique, la phylogénie, la génomique, la protéomique [Zhao et al,]. Par exemple, des algorithmes de clustering appliqués aux données d'expressions de gènes peuvent être employés pour identifier les gènes co-réglés et fournir une empreinte digitale génétique pour différentes maladies [Homayouni et al, 05]. Des algorithmes de clustering appliqués sur une base de données des protéines connues peuvent être employés pour organiser automatiquement les différentes protéines en familles de lien étroit et de lien distant et identifier les sous séquences qui sont la plupart du temps préservés à travers des protéines [Krasnogor et al, 04].

1.6 Conclusion

Ces dernières années les volumes de données de toutes sortes croît de plus en plus. Avec tant de données disponibles, il est nécessaire de développer des algorithmes qui peuvent extraire des informations significatives à partir de vastes mémoires. La recherche des pépites utiles d'information parmi des quantités énormes de données est devenue connue comme le champ de fouille de données. Ce champ interdisciplinaire tire ces techniques et taches de plusieurs domaines. Une tache en particulier, le clustering est un processus d'analyse exploratoire qui divise un jeu de données en un ensemble de clusters, ces clusters découverts peuvent être employés pour expliquer des caractéristiques de la distribution de données sous-jacente, et servir comme une base pour d'autres tâches de fouille de données.

Le clustering a été couronnée de succès dans un grand nombre d'applications d'extraction de connaissances mais il reste encore une tache de défi qui a beaucoup de problème à surmonter.

Dans le chapitre suivant, nous allons présenter un état de l'art général sur les différents techniques et algorithmes de clustering existants, nous allons également présenter l'application des méthodes métaheuristiques au clustering.

Chapitre 2

Le Clustering des données

Knowledge is of two kinds: we know a subject ourselves,
or we know where we can find information about it.
— Samuel Johnson, 1775

2.1 Introduction

Le clustering est une tâche qui a été pratiquée par les humains pendant des milliers d'années et il a été entièrement automatisé en dernières quelques décennies grâce aux avancements des technologies de calcul. Le clustering est généralement défini comme la tâche de trouver des groupes naturels dans un ensemble de données multidimensionnelles. Le regroupement est fait tel que les données dans le même cluster sont plus similaires que celles dans des clusters différents. C'est un concept intuitif mais difficile à réaliser dans la pratique. Cette difficulté est causée par un manque de définition unique et précise d'un cluster dû à un manque d'informations préalables sur les distributions des données.

Dans la littérature, plusieurs définitions d'un cluster, de ce que constitue un cluster et la relation entre les clusters existent. Sur cette base, une multitude de techniques de clustering est développée. Les techniques peuvent se diffère dans leurs principes, propriétés, paramètres et formes générales du partitionnement généré. La catégorisation de ces techniques peuvent être réalisée selon plusieurs aspects : la mesure de proximité utilisée entre les données , la théorie ou les concepts fondamentales sur lesquels se basent les techniques , la nature des données manipulées et beaucoup d'autres critères. En réalité, la catégorisation de ces techniques n'est ni directe, ni canonique car les catégories se chevauchent.

En générale, un algorithme de clustering contient les étapes suivantes :

- Le choix d'une mesure de proximité appropriée au domaine des données. Cette mesure doit assurer que tous les attributs des données contribuent également au calcul de la mesure de proximité et il n'y a aucun attribut qui domine d'autres.
- La définition d'un critère de clustering qui peut être exprimé par l'intermédiaire d'une fonction objective ou d'un autre type de règles.
- Une représentation simple et compacte de l'ensemble de données et l'ensemble des clusters.
- Le clustering : cette étape peut être réalisée de nombreuses façons, selon la technique adoptée.

- La validation des résultats : la qualité des résultats d'un algorithme de clustering est vérifiée en employant des mesures de validité de clusters. La qualité de chaque cluster doit être jugée non seulement par l'algorithme de clustering (la fonction objective) qui l'a produit, mais aussi selon un critère d'évaluation externe [Law et al,04].

Dans ce chapitre nous commençons par présenter les techniques principales de clustering qui sont des fois référées dans la littérature comme classiques. En second lieu, nous présentons un autre cadre dans lequel le clustering peut être mis, c'est le cadre d'optimisation. En effet, le problème de clustering peut être modélisé comme un problème d'optimisation où l'espace de recherche grandit exponentiellement et ne peut pas être parcouru exhaustivement même pour des problèmes de taille moyenne. Le problème de clustering est connu pour être NP-difficile et l'utilisation des métaheuristiques d'optimisation inspirées de la biologie offre beaucoup d'avantages à ce problème. Dans cet état de l'art, nous présentons les algorithmes les plus utilisés, les plus connus, et ceux qui exhibent des idées différentes et intéressantes. A la fin, nous terminons par donner différents types de mesures de validité de clusters.

2.2 Quelques concepts nécessaires

2.2.1 La matrice de données (Jeu de données)

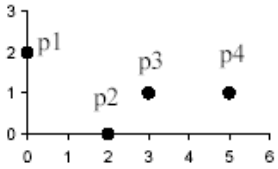
Les objets (échantillons, mesures, modèles, événements) sont habituellement représentés comme des points (vecteurs) dans un espace multidimensionnel, où chaque dimension représente un attribut distinct (variable, mesure) décrivant l'objet.

Ainsi, un ensemble d'objets est représenté comme une matrice $m \times n$, avec m lignes , une pour chaque objet et n colonnes , une pour chaque attribut. Cette matrice est appelée matrice de données ou jeu de données. La figure 2.1, ci-dessous, fournit un exemple concret de quelques points et leur matrice de données correspondante.

2.2.2 La matrice de proximité

Plusieurs algorithmes de clustering utilisent la matrice de données originale et beaucoup d'autres emploient une matrice de similarité, ou une matrice de dissimilarité. Pour la convenance, les deux matrices sont généralement mentionnées comme une matrice de proximité, P. Une matrice de proximité, P, est une matrice $m \times m$ contenant toutes les dissimilarités ou les similarités entre les objets considérés. Si p_i et p_j sont le $i^{ème}$ et le $j^{ème}$ objets, respectivement, alors l'entrée à la i^{emme} ligne et la j^{emme} colonne de la matrice de proximité est la similarité, ou la dissimilarité, entre p_i et p_j.

Les figures 2.1 montre, respectivement, quatre points, leur matrice de données et leur matrice proximité correspondante.



point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

	p1	p2	p3	p4
p1	0.000	2.828	3.162	5.099
p2	2.828	0.000	1.414	3.162
p3	3.162	1.414	0.000	2.000
p4	5.099	3.162	2.000	0.000

Figure 2.1. Quatre points, leur matrice de données et leur matrice de proximité.

2.2.3 Définitions d'un Cluster

La définition de ce que constitue un cluster n'est pas bien défini et le terme, cluster n'a pas de définition précise [Kumar,00] [Berkhin,02]. Cependant, plusieurs définitions d'un cluster sont généralement utilisées.

A. Définition de cluster bien séparé: Un cluster est un ensemble de points tel que n'importe quel point dans le cluster est plus proche (ou plus similaire) de chaque autre point dans le cluster que de n'importe quel point qui n'est pas dans le cluster. Parfois un seuil est employé pour spécifier que tous les points dans un cluster doivent être suffisamment proches (ou similaires) l'un de l'autre (figure 2.2).

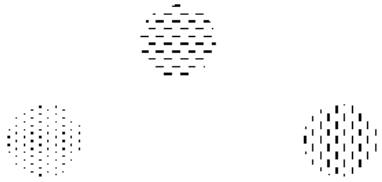


Figure 2.2. Trois clusters bien séparés

Cependant, dans beaucoup de jeux de données, un point sur le bord d'un cluster peut être plus proche (ou plus similaire) de quelques points dans un autre cluster que de points dans son propre cluster. Par conséquent, beaucoup d'algorithmes de clustering utilisent la définition suivante.

B. Définition de cluster basé sur le centre: Un cluster est un ensemble de points tel qu'un point dans un cluster est plus proche (plus similaire) du " centre" de ce cluster, que du centre de n'importe quel autre cluster. Le centre d'un cluster est souvent un centroïde, la moyenne de tous les points dans le cluster, ou un médoïde, le point le plus représentatif d'un cluster (figure 2.3).

Figure 2.3. Quatre clusters basés sur le centre

C. **Définition de Cluster contiguë** (le voisin le plus proche ou le clustering transitif): Un cluster est un ensemble de points tel qu'un point dans un cluster est plus proche (ou plus similaire) d'un ou de plusieurs autres points dans le cluster que de n'importe quel point qui n'est pas dans le cluster (figure 2.4).

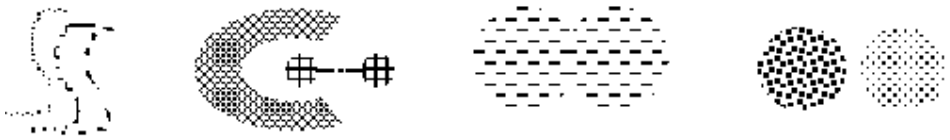


Figure 2.4. Huit clusters contigus

D. **Définition basée sur la densité** : Un cluster est une région dense de points, qui est séparée des autres régions de haute densité par des régions de basse densité. Cette définition est souvent utilisée quand les clusters sont irréguliers ou entrelacés et quand les bruits sont présents (figure 2.5).

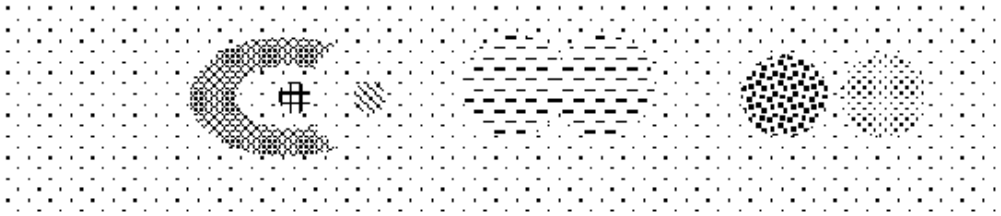


Figure 2.5. Six clusters denses

2.2.4 Types et échelles de données

La mesure de proximité et le type de clustering utilisé dépendent des types et échelles des attributs de données [Jain et al,88] . Les trois types d'attributs sont montrés dans le tableau 2.1, tandis que les différentes échelles de données sont montrées dans le tableau 2.2.

Binaire	deux valeurs, vrai ou faux
Discret	un nombre fini de valeurs ou les entiers
Continu	un nombre infini de valeurs ou les réels

Tableau2.1. Les différents types d'attributs.

Qualitative	Nominal	les valeurs sont juste des noms différents. par exemple : les codes postaux, les couleurs,le sexe.
	Ordinal	les valeurs reflètent un ordre, rien plus. par exemple : bon, meilleur, mieux ou couleurs ordonnées par le spectre.
Quantitative	Intervalle	la différence entre les valeurs est significative par exemple, l'intervalle de température.
	Ratio	rapport entre deux grandeurs. par exemple : les quantités monétaires, comme le salaire et le bénéfice et beaucoup de quantités physiques comme courant électrique, pression, etc.

Tableau2.2. Les différentes échelles de données.

2.2.5 Distance et similarité

Le concept de similarité ou de dissimilarité est le composant essentiel de n'importe quelle forme du clustering qui nous aide à naviguer dans l'espace de données pour former des clusters [Pedrycz,05].

En calculant la similarité, nous pouvons sentir et articuler à quel point deux points sont proches, et sur la base de cette proximité, nous pouvons, les assignent au même cluster.

Formellement, la similarité *d(x,y)* entre *x* et *y* est considéré comme une fonction à deux arguments satisfaisant les conditions suivantes :

$d(x, y) \geq 0$

$d(x, x) = 0$

$d(x, y) = d(y, x)$

La distance est la mesure la plus utilisé parmi les types de mesures de similarité et de dissimilarité, elle exige la satisfaction de l'inégalité triangulaire c'est-à-dire, pour n'importe quel points *x*, *y* et *z*, nous avons : $d(x, y) + d(y, z) \geq d(x, z)$

Le tableau suivant présente quelques fonctions de distance. [Pedrycz,05]

Fonction de distance	Formule
Distance Euclidienne	$d(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$
Distance de Hamming	$d(x,y) = \sum_{i=1}^{n} \lvert x_i - y_i \rvert$
Distance de chebyschev	$d(x,y) = \max_{i=1,2,\ldots,n} \lvert x_i - y_i \rvert$
Distance de Minkowski	$d(x,y) = \sqrt[p]{\sum_{i=1}^{n} (x_i - y_i)^p}, p > 0$

Distance de Canberra	$d(x,y)=\sum_{i=1}^{n}\frac{\lvert x_i-y_i\rvert}{x_i+y_i}$, x_i et y_i sont positifs
Séparation angulaire	$d(x,y)=\frac{\sum_{i=1}^{n}x_i y_i}{\left[\sum_{i=1}^{n}x_i^2\sum_{n=1}^{n}y_i^2\right]^{1/2}}$

Tableau 2.3. Fonctions de distance entre deux points *x* et *y*.

2.3 Techniques principales de Clustering

Au cours des années, beaucoup de différentes techniques de clustering ont été proposées dans la littérature. La catégorisation de ces techniques n'est ni directe, ni canonique, en réalité, les catégories se chevauchent [Berkhin,02] .Les algorithmes de clustering peuvent être classifiés selon [Halkidi et al,01a] :

- Le type des données en entrée.
- Les critères de clustering définissant la similarité entre les points de données.
- La théorie et les concepts fondamentaux sur lesquels les techniques de clustering sont basés (par exemple, la théorie floue, les statistiques).

Ainsi, selon la méthode adoptée pour définir des clusters, les algorithmes peuvent être classifiés aux types suivants :

- Clustering par partitionnement (Partitional clustering)
- Clustering hiérarchique (Hierarchical clustering)
- Clustering basé sur la densité (Density-based clustering)
- Clustering basé sur les grilles (Grid-based clustering)

Pour chacune de ces catégories, il y a beaucoup de sous-types et beaucoup d'algorithmes différents pour trouver des clusters.

2.3.1 Clustering par partitionnement

Les techniques par partitionnement créent un partitionnement des points de données, d'un seul niveau. Si *k* est le nombre désiré de clusters, alors les approches par partitionnement trouvent typiquement tous les *k* clusters immédiatement.

Les techniques par partitionnement sont divisées en deux sous-catégories principales [Kotsiantis et al, 01], les algorithmes basés sur les centroïdes et les algorithmes basés sur les médoïdes. Nous allons décrire les deux algorithmes les plus connus : Kmeans et Kmedoid.

Ces deux techniques sont basées sur l'idée qu'un point de centre peut représenter un cluster. Pour Kmeans on emploie la notion du centroïde qui est le point de la moyenne ou la médiane d'un groupe de points. Pour Kmedoid on utilise la notion d'un médoïde qui est le point le plus représentatif (central) d'un groupe de points.

A. Kmeans

La technique de clustering de Kmeans [Forgy,65][Macqueen,67] est très simple, son algorithme de base est décrit comme suit :

1. Choisir *k* points comme centroïdes initiaux.
2. Assigner tous les points au centroïde le plus proche.
3. Recalculer le centroïde de chaque cluster.
4. Répétez les étapes 2 et 3 jusqu'à ce que les centroïdes ne changent pas.

Ce procédé converge toujours à une solution, bien que la solution soit typiquement un minimum local.

La complexité spatiale de Kmeans est *O(mn)* où *m* est le nombre de points et *n* est le nombre d'attributs, tandis que sa complexité temporelle est *O(I*k*m*n)* où I est le nombre d'itérations exigées pour la convergence. Kmeans est efficace et simple tant que le nombre de clusters est significativement moins que le nombre de points *m*.

Le choix des centroïdes initiaux appropriés est l'étape clef de l'algorithme de Kmeans. Il est facile de choisir les centroïdes initiaux aléatoirement mais l'inconvénient est que la partition finale dépend de la partition initiale.

Le grand avantage de l'algorithme de Kmeans est sa complexité temporelle, qui le rend efficace dans le traitement de grands jeux de données, mais il a un certain nombre de limitations et de problèmes tels que [Berkhin,02] :

- Le résultat dépend fortement de la conjecture initiale de centroïdes.
- Il se termine souvent à un optimum local,
- Le processus est sensible aux bruits.
- Seulement des attributs numériques sont couverts.
- Il tend à trouver des clusters sphériques de tailles égales.

Plusieurs variantes de l'algorithme de Kmeans existent dans la littérature. Certains d'entre eux essayent de choisir une bonne partition initiale pour que l'algorithme trouve la valeur du minimum global. Une autre variante consiste à fractionner ou fusionner les clusters résultants. Typiquement un cluster est fractionné quand sa variance est au-dessus d'un seuil pré spécifié et deux clusters sont fusionnés quand la distance entre leur centroïdes est au-dessous d'un

autre seuil pré spécifié [Jain et al, 99] . L'algorithme le plus connu qui utilise cette technique de fusion et de fractionnement de clusters c'est ISODATA, qui a été est utilisé particulièrement dans le traitement d'image [Jain et al,88].

B. Kmedoid

Dans l'approche Kmedoid, un cluster est représenté par un de ses points. Ce point représentatif est appelé médoïde, c'est un point qui est le plus placé au centre en tenant en compte quelques mesures, comme par exemple, la distance.

De nouveau, l'algorithme est conceptuellement simple. Il est décrit comme suit [Kumar, 00]:

1. Choisir *k* points initiaux. Ces points sont les médoïdes candidats qui sont destinés à être les points les plus centraux de leurs clusters.
2. Considérer l'effet de remplacer un des points choisis (médoïdes) avec un des points non choisis. Conceptuellement, ceci est fait de la façon suivante :
   On calcule la distance entre chaque point non choisi et le médoïde candidat le plus proche et on calcule la somme de toutes les distances, cette somme représente le "coût" de la configuration actuelle. Tous les échanges possibles d'un point non choisi par un autre choisi sont considérés, et le coût de chaque configuration est calculé.
3. Choisir la configuration avec le coût le plus bas. Si c'est une nouvelle configuration, alors répéter l'étape 2.
4. Sinon, associer chaque point non choisi au point choisi le plus proche (médoïde) et arrêter.

Le $i^{ème}$ médoïde est calculé en utilisant $\sum_{j=1}^{n_i} P_{ij}$ où P_{ij}, est la proximité entre le $i^{ème}$ médoïde et le $j^{ème}$ point dans le cluster. Pour des similarité (dissimilarité) on veut que cette somme soit le plus possible grande (petite).

Cette approche n'est pas limitée aux espaces euclidiens. En outre, l'utilisation de médoïdes pour définir des clusters rend cette méthode résistante contre les bruits dans les données mais la complexité temporelle est $O(k\ (m - k)^2)$ où m est le nombre de points du jeu de données [Halkidi et al,01a] . La dégradation est faite dans les étapes 2 et 3 de l'algorithme, puisque la découverte d'un meilleur médoïde exige l'essai de tous les points qui ne sont pas médoïdes. Cela est très coûteux en temps de calcul.

Plusieurs algorithmes basés sur la notion de médoïde existent comme PAM (Partitioning Around Medoids) qui est un algorithme Kmedoid qui essaye de grouper un ensemble de *m*

points en *k* clusters en exécutant les étapes décrites ci-dessus. CLARA (Clustering LARge Applications) est une adaptation de PAM pour manipuler de grands jeux de données. CLARANS est né de deux algorithmes de clustering PAM et CLARA. CLARANS est un des premiers algorithmes de clustering qui a été développé précisément pour le datamining spatial [Ng et al,94].

2.3.2 Clustering hiérarchique

Le clustering hiérarchique construit une hiérarchie de clusters, ou, en d'autres termes, un arbre de clusters, également connu sous le nom de dendrogramme. Ce dendrogramme décrit l'ordre dans lequel les points sont fusionnés (vue de bas en haut) ou les clusters sont fractionnés (vue de haut en bas). La figure 2.6 représente sept points p_1, p_2, p_3, p_4, p_5, p_6 et p_7 dans trois clusters et le dendrogramme correspondant. Le dendrogramme peut être coupé à différents niveaux pour donner les différents clusters des données [Jain et al, 99].

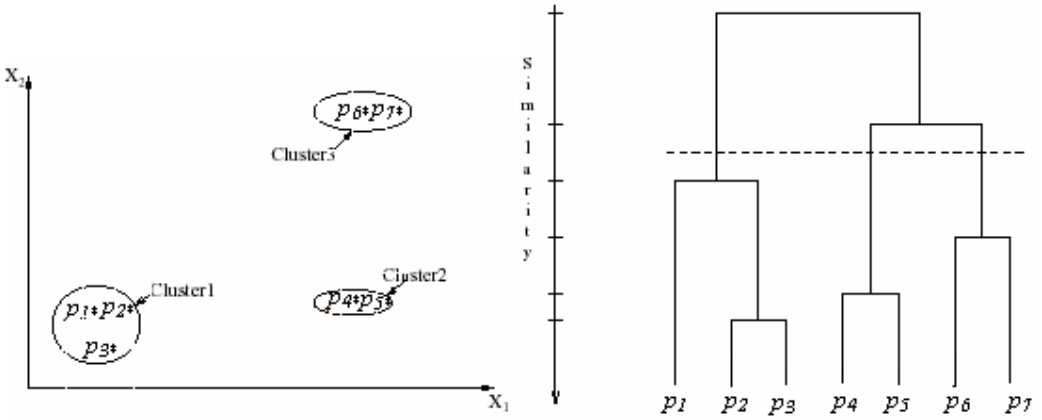


Figure 2.6 : clustering de sept points et le dendrogramme correspondant.

Il y a deux approches de base pour générer un clustering hiérarchique :

- **Agglomérative** : commence par des clusters d'un seul point (singleton) et fusionne récursivement deux ou plusieurs clusters les plus appropriées.
- **Divisive :** commence par un cluster de tous les points de données et fractionne récursivement le cluster le plus approprié.

Les techniques divisives sont moins communes, et on va se concentrer sur les techniques agglomératives.

A. Techniques Divisives

L'algorithme divisive le plus simple et le plus utilisé est donné par les étapes suivantes [Wilson et al,90] :

1. Calculer l'arbre de couverture minimum (MST ou Minimum Spanning Tree) du graphe de proximité.
2. Créer un nouveau cluster en enlevant le lien correspondant à la plus grande distance.
3. Répéter l'étape 2 jusqu'à ce que seulement les clusters de singleton restent.

Cette approche est la version divisive de la technique agglomérative Single link que nous verrons dans la section suivante. Pour la recherche de l'arbre de couverture minimum (MST) pour un graphe *G* donné, on peut citer de façon non exhaustive les algorithmes connus de Kruskal [Kruskal, 56] et de Prim [Prim, 57]. L'algorithme de Prim est donné par :

1. Débuter l'arbre minimum avec les deux sommets les plus proches.
2. Déterminer le sommet qui est le plus proche de n'importe quel sommet de l'arbre minimum.
3. Ajouter ce sommet à l'arbre minimum.
4. Répéter 2 et 3 tant qu'il reste des sommets non connectés à l'arbre minimum.

B. Techniques agglomératives

Plusieurs techniques hiérarchiques agglomératives peuvent être exprimées par l'algorithme suivant, qui est connu comme algorithme de Lance-Williams [Lance et al,67] :

1. Calculer la matrice de proximité.
2. Fusionner les deux clusters les plus proches (les plus similaires).
3. Mettre à jour la matrice de proximité pour refléter la proximité entre le nouveau cluster et les clusters originaux.
4. Répéter les étapes 3 et 4 jusqu'à ce que seulement un seul cluster reste.

L'étape clé de cet algorithme est le calcul de la proximité entre deux cluster et c'est à ce niveau où les diverses techniques hiérarchiques agglomératives diffèrent. La formule de Lance-Williams pour calculer la proximité entre les clusters *Q* et *R*, où *R* est formé en fusionnant les clusters *A* et *B*, est donnée par:

$$d(R, Q) = a_A\, d(A, Q) + a_B\, d(B, Q) + b\, d(A, Q) + g\, |d(A, Q) - d(B, Q)|$$

Cette formule indique qu'après la fusion des clusters *A* et *B* pour former le cluster *R*, la distance du nouveau cluster, *R*, à un cluster existant, *Q* , est une fonction linéaire des distances entre *Q* et les deux clusters originaux *A* et *B*.

Le réglage des paramètres a_A , a_B , b *et* g selon le tableau 2.4 implique un certain nombre d'algorithmes de clustering.

L'algorithme	$a_{A,}(a_B)$	b	g
Single link	1/2	0	-1/2
Complete link	1/2	0	1/2
Average link	$\frac{n_A}{n_A + n_B}$	$-\frac{n_A n_B}{(n_A + n_B)^2}$	*0*

Tableau 2.4. Valeurs des paramètres dans la formule de Lance Williams et les algorithmes de clustering agglomératif résultants, n_A, n_B et n_C dénotent le nombre de points dans les clusters correspondants

Plusieurs algorithmes dérive de l'algorithme de Lance-Williams mais nous allons décrire dans ce qui suit les plus connus.

Ø Single link

Pour la version Single link du clustering hiérarchique, la proximité de deux clusters est définie par la distance minimale entre n'importe quels deux points dans les clusters différents. Cette distance minimale entre les points appartenant aux clusters *A* et *B* est calculée avec la formule : $d(A,B) = \min_{x \in A, x \in B} d(x,y)$ [Pedrycz,05] .

Cette technique est bonne pour la manipulation de formes non elliptiques, mais elle est sensible aux bruits. Le Single link est une des méthodes les plus utilisées [Jain et al, 99]. La figure 2.7 donne un échantillon de matrice de similarité pour cinq points (p_1 – p_5) et le dendrogramme qui montre les séries des fusions qui résultent de l'utilisation de la technique de Single link.

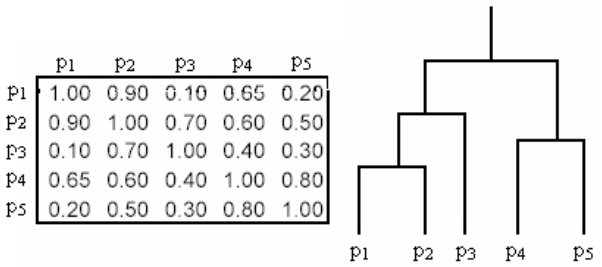
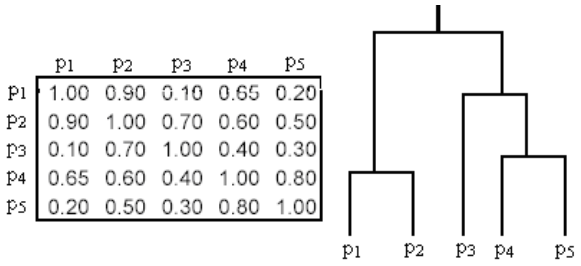


Figure 2.7. Exemple d'une matrice de similarité et le dendrogramme correspondant à l'application de Single link

Ø Complete link

Pour la version de Complete link du clustering hiérarchique, la proximité de deux clusters est définie par la distance maximale entre n'importe quels deux points dans les clusters différents. Cette distance maximale entre les points appartenant aux clusters *A* et *B* est calculée avec la formule : $d(A,B) = \max_{x \in A, x \in B} d(x,y)$ [Pedrycz,05] .

Le Complete link est moins susceptible au bruit, mais il peut fractionner de grands clusters, et il a des problèmes avec les formes convexes. La figure 2.8 donne un échantillon de matrice de similarité pour cinq points ($p_1 - p_5$) et le dendrogramme qui montre les séries des fusions qui résultent de l'utilisation de la technique de Complete link.



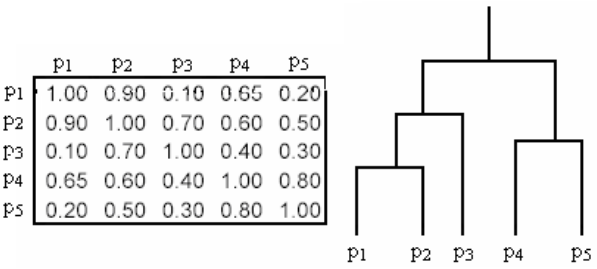
	p1	p2	p3	p4	p5
p1	1.00	0.90	0.10	0.65	0.20
p2	0.90	1.00	0.70	0.60	0.50
p3	0.10	0.70	1.00	0.40	0.30
p4	0.65	0.60	0.40	1.00	0.80
p5	0.20	0.50	0.30	0.80	1.00

Figure 2.8 Exemple d'une matrice de similarité et le dendrogramme correspondant à l'application de Complete link

Ø Average link

Pour la version de Average link du clustering hiérarchique, la distance de deux clusters est définie par la moyenne des distances entre toutes les paires de points dans les clusters différents. C'est une approche intermédiaire entre Single link et Avearge link .Ceci est exprimé par l'équation suivante: $d(A,B) = \frac{1}{n_A n_B} \sum_{x \in A, y \in B} d(x,y)$, où n_A et n_B représente la taille des clusters *A* et *B* respectivement [Pedrycz,05].

La figure 2.9 donne un échantillon de matrice de similarité et le dendrogramme montre les séries des fusions qui résultent de l'utilisation de l'approche de Avearge link. Le clustering hiérarchique dans ce cas simple est le même comme produit par Single link.



	p1	p2	p3	p4	p5
p1	1.00	0.90	0.10	0.65	0.20
p2	0.90	1.00	0.70	0.60	0.50
p3	0.10	0.70	1.00	0.40	0.30
p4	0.65	0.60	0.40	1.00	0.80
p5	0.20	0.50	0.30	0.80	1.00

Figure 2.9. Exemple d'une matrice de similarité et le dendrogramme correspondant à l'application de Average link

En générale, les techniques aglomératives tendent à fractionner les grands clusters. Elle tendent à prendre de bonnes décisions locales concernant la fusion de deux clusters mais une fois qu'une décision est prise elle ne sera pas changée, elle est finale et le problème c'est que les bonnes décisions de fusion locales peuvent ne pas aboutir aux bons résultats globaux [Kumar, 00] .

Plusieurs algorithmes hiérarchiques existent comme CURE [Guha et al,98] qui est un algorithme agglomératif qui emploie une variété de différentes techniques. Il représente un cluster en employant plusieurs points " représentatifs " du cluster. Ces points captureront la géométrie et la forme du cluster. Le premier point représentatif est choisi pour être le point le plus loin du centre du cluster, tandis que les points restants sont choisis de sorte qu'ils soient les plus éloignés de tout des points précédemment choisis. Une fois que les points représentatifs sont choisis, ils sont rétrécis vers le centre par un facteur, α.

La distance utilisée entre deux clusters est la distance minimale entre n'importe quels deux points représentatifs (après qu'ils sont rétrécis vers leurs centres respectifs). Cet arrangement est l'équivalent de Average link si $\alpha = 0$, et il est presque le même que le Single link si $\alpha = 1$.

2.3.3 Clustering basé sur la densité

Les algorithmes basés sur la densité considèrent les clusters comme des régions de points denses dans l'espace de données qui sont séparées par des régions de faible densité [Halkidi et al,01a] . Un des algorithmes les plus bien connus de cette catégorie est le DBSCAN [Ester et al,96]. L'idée principale de DBSCAN est celle pour chaque point dans un cluster, le voisinage d'un rayon donné *e* doit contenir au moins un nombre minimum de point *MinPts*, c-à-d la cardinalité du voisinage doit excéder un seuil. DBSCAN est basé sur les concepts de point noyau, point bordure et point bruit, qui sont aussi basés sur des notions d'accessibilité et de connectivité.

Un point noyau est un point avec un voisinage consistant de plus de *MinPts* points. Le concept d'un point *y* qui densité-accessible d'un point noyau *x* est défini comme une séquence finie entre *x* et *y* tel que chaque point successeur appartient au voisinage de son prédécesseur.

Le concept de densité-connectivité est défini tel que deux points *x, y* sont dits densité-connectés s'ils sont densité-accessibles d'un même point noyau.

Un cluster basé sur la densité est maintenant défini comme un ensemble de points densité-connectés .Un cluster contient non seulement des points noyau mais également des points qui ne satisfont pas la condition de point de noyau. Ces points sont les points bordure du cluster. Le bruit est l'ensemble de points non contenus dans n'importe quel cluster.

L'algorithme de DBSCAN est comme suit:

1. Choisir un point arbitraire *x*.

2. Trouver tous les points qui sont densité-accessibles de *x.* Si *x* est un point noyau alors nous avons formé un cluster. Si *x* est un point bordure alors aucun point n'est densité-accessible de *x.*
3. Répétition pour le prochain point dans les données.

L'inconvénient majeur est que les clusters doivent avoir un minimum de points ***MinPts.*** Ceci rend impossible pour DBSCAN de trouver les très petits clusters dans de grands jeux de données.

Plusieurs versions DBSCAN ont été proposées. Une version incrémentale de DBSCAN est présentée dans [M.Ester et al ,98] .Il a été prouvé que cet algorithme incrémental donne le même résultat que DBSCAN. Un autre algorithme de clustering GDBSCAN généralisant l'algorithme DBSCAN est présenté dans [Sander et al,98] .GDBSCAN peut grouper des points selon les deux, leurs attributs numériques et catégoriques. En outre, DBCLASD élimine le besoin de paramètres MinPts et ε [Xu et al ,98] .OPTICS est présenté dans [Ankerst et al ,99], il généralise DBSCAN en créant un ordre des points qui permet l'extraction de clusters avec des valeurs arbitraires de *e*.

2.3.4 Clustering basé sur les grilles

Ces algorithmes commencent par partitionner l'espace de données à un nombre fini de cellules et accomplir ensuite des opérations exigées sur l'espace partitionné. Les cellules qui contiennent plus qu'un certain nombre de points sont traitées comme denses et les cellules denses sont connectées pour former les clusters [Kotsiantis et al,01].

La forme de base d'un algorithme basé sur les grilles est la suivante [Steinbach et al,01]:

1. Diviser l'espace sur lequel les données s'étendent en cellules rectangulaires, par exemple, on divise l'intervalle de valeurs de chaque dimension en cellules de taille égales. La figure 2.10 montre un exemple de grille de deux dimensions.
2. Rejeter les cellules de grille à densité basse. Ceci assume une définition de cluster basée sur la densité, c'est-à-dire, que les régions à haute densité représentent des clusters, alors que les régions à basse densité représentent le bruit. Cette supposition est souvent bonne, bien que les approches basées sur la densité puissent avoir des difficultés quand il y a des clusters de différentes densités.
3. Combiner des cellules adjacentes à haute densité pour former des clusters.

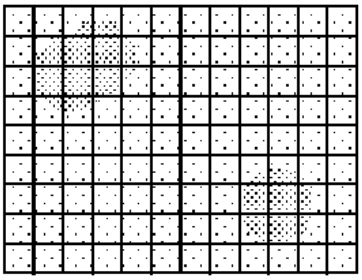


Figure 2.10. Grille bidimensionnelle pour la détection de clusters

Il y a un certain nombre de problèmes évidents. Les grilles sont carrées ou rectangulaires et elle ne convient pas nécessairement à la forme des clusters. Ceci peut être traité en augmentant le nombre de cellules de la grille, mais cela augmente le temps de calcul. En outre, ces méthodes ne fonctionnent pas bien pour des dimensions moyennes ou élevées.

Certains des algorithmes basés sur les grilles sont : Sting qui est basé sur des méthodes statistique [Wang et al,97] et WaveCluster [Sheikholeslami et al,98] , cette dernière technique de clustering interprète les données originales comme un signal bidimensionnel et applique ensuite des techniques de traitement des signaux pour représenter les données originales dans un nouvel espace où l'identification de cluster est plus directe. Plus spécifiquement, WaveCluster définit une grille bidimensionnelle uniforme sur les données et représente les points de données dans chaque cellule de la grille par un nombre de points. Ainsi, une collection de points de données bidimensionnels devient une image, c.-à-d., un ensemble de pixel, et le problème de clustering devient un problème de segmentation d'image.

2.4 Métaheuristiques pour le Clustering

Le problème de clustering de données est identifié comme une des problématiques majeures en extraction des connaissances à partir de données. La popularité, la complexité et toutes ces variantes du problème de clustering de données ont donné naissance à une multitude de méthodes de résolution. Ces méthodes peuvent à la fois faire appel à des principes heuristiques ou encore mathématiques. Parmi celles-ci, il existe une branche qui s'inspire plus spécialement de principes issus de la biologie. Les motivations des chercheurs sont d'une part de tester de nouveaux algorithmes sur le problème de clustering et de connaître leurs apports. Mais elles sont aussi de proposer de nouvelles sources d'inspiration, car le problème de clustering se rencontre souvent chez les animaux et dans les systèmes biologiques.

Le problème de clustering peut être modélisé comme un problème d'optimisation où l'espace de recherche grandit exponentiellement et ne peut pas être parcouru exhaustivement même

pour des problèmes de taille moyenne. En effet, le problème de clustering est connu pour être NP-difficile [Bilmes et al, 97].

Nous nous intéressons dans cette partie aux algorithmes de clustering qui utilisent des métaheuristiques d'optimisation inspirées de la biologie et aux avantages qu'ils apportent pour ce problème. Parmi les algorithmes actuellement utilisés, on peut citer par exemple les algorithmes génétiques, algorithmes à base de populations de fourmis artificielles, algorithmes d'essaims de particules et les systèmes immunitaires artificiels.

2.4.1 Clustering par algorithmes évolutionnaires

Les algorithmes évolutionnaires sont des algorithmes inspirés de l'intelligence de la nature. Ils sont basés sur la théorie de l'évolution et de la sélection naturelle élaborée par Charles Darwin**,** qui annonce que dans un environnement, seules les espèces les mieux adaptées survivent au cours du temps, les autres sont condamnées à disparaître, et au sein de chaque espèce, le renouvellement des populations est essentiellement dû aux meilleurs individus de l'espèce.

Les algorithmes génétiques représentent une catégorie des algorithmes évolutionnaires. Les algorithmes génétiques simulent le processus d'évolution d'une population. A partir d'une population de solutions du problème représentant des individus, on applique des opérateurs simulant les interventions sur le génome telle que le croisement , la mutation et la sélection pour arriver à une population de solutions de mieux en mieux adaptée au problème. Cette adaptation est évaluée grâce à une fonction objective (fitness) [Jourdan, 03].

Un algorithme génétique doit coder ces individus, le codage dépend de la nature des variables du problème à coder. Parmi les principaux mécanismes de codage que les algorithmes génétiques utilisent on peut citer le codage binaire et le codage réel.

L'opérateur de sélection est un processus qui sélectionne ou choisit des individus à partir d'une population courante pour un but de construire une nouvelle population qui contient en général des individus qui semblent plus proche de la solution optimale du problème.

Le principe de l'opérateur de croisement utilisé par l'algorithme génétique est le même que le principe du croisement biologique, il permet la création de nouveaux individus selon un processus fort et simple. Il consiste a sélectionner deux individus aléatoirement (parents), choisir un point identique pour les deux individus pour les coupées et intervertit les deux premières portions ou les derniers des deux individus coupés. Donc un opérateur de croisement permet l'échange d'information entre les individus. Le résultat de cet opérateur est la construction de deux nouveaux individus (fils).

L'opérateur de mutation modifie aléatoirement, avec une certaine probabilité, la valeur d'un bit ou de plusieurs bits d'un individu [Draa,04].

La fonction d'évaluation quantifie la qualité de chaque individu par rapport au problème. Elle est utilisée pour sélectionner les individus pour la reproduction. Les individus ayant une bonne qualité ont plus de chance d'être sélectionnés pour la reproduction et donc plus de chance que la population suivante hérite de leur matériel génétique. La fonction d'évaluation produit la pression qui permet de faire évoluer la population de l'algorithme génétique vers des individus de meilleure qualité [Jourdan, 03]. La figure 2.11 montre le fonctionnement itératif simplifié d'un algorithme génétique.

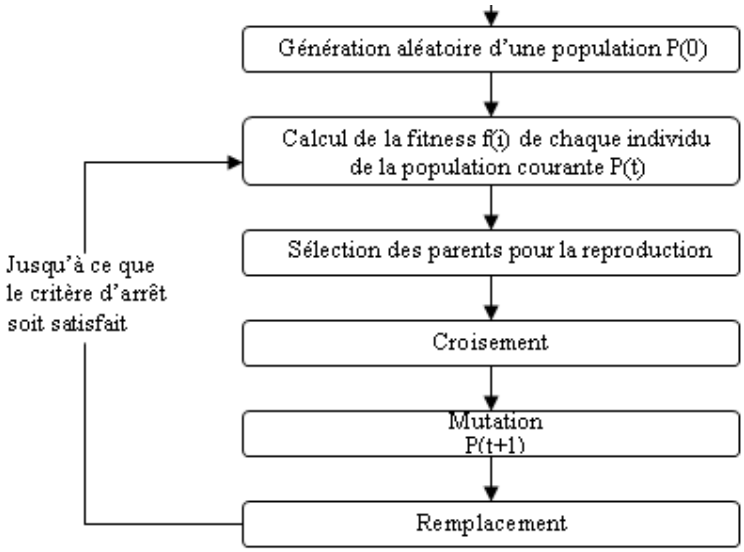


Figure 2.11. Fonctionnement général d'un algorithme génétique de base

Il existe de nombreux algorithmes de clustering par algorithmes génétiques. Les premiers travaux sont dus à [Raghavan et al, 79].Le nombre de clusters est fixé à l'avance et la représentation de longueur *N* associe un cluster à chaque donnée, comme dans la figure 2.12.

donnée	d_1	d_2	….	d_n
clusters	c_2	c_k	…..	c_1

Figure 2.12. Un individu représenté comme un vecteur

Le croisement à un point échange des étiquettes de clusters entre deux individus. Cet opérateur peut donc faire disparaître des clusters. La mutation peut faire apparaître de nouveaux clusters. La fonction d'évaluation consiste à minimiser une erreur quadratique.

Dans [Maulik et al,00], [Babu et al,94], les auteurs proposent de représenter un individu comme vecteur de centroïdes de clusters. Pour un jeu de données de M attributs et K clusters désirés, un individu est donc représenté par un vecteur de longueur M×K (figure 2.13).

c_1				c_2				…..	c_K			
0.8	10.5	…..	90.8	47.0	51.3	…..	12.3	…..	56.0	1.5	…..	41.6

Figure 2.13.Un individu représenté comme un vecteur de centroïdes.

L'approche proposée dans [Babu et al,94], est une approche hybride qui utilise l'algorithme génétique pour trouver de bons centroïdes de clusters initiaux et l'algorithme Kmeans pour trouver la partition finale.

Le codage introduit dans [Bezdek et al ,94] consiste à représenter un individu par une matrice booléenne MB de N lignes et K colonnes où N représente le nombre de points de données et K le nombre de clusters. Un élément de la matrice MB(i,j) prend la valeur 1 si le point de donnée d_i appartient au cluster j, 0 sinon. Un individu est donc de la forme présentée dans la figure 2.14:

Données/Clusters	c_1	c_2	…..	c_K
d_1	1	0	…..	0
d_2	0	1	…..	0
…..	…..	…..	…..	…..
d_N	0	0	…..	1

Figure 2.14. Un individu représenté comme une matrice booléenne

Dans cette représentation, l'opérateur de croisement est défini cette fois en 2D. Un point très important à noter dans cette représentation est la possibilité de la généraliser au clustering flou qui améliore l'approche en évitant le problème d'apparition de plusieurs 1 sur une même ligne. Dans le clustering flou, les valeurs ne sont plus booléennes mais représentent des degrés d'appartenance.

Dans [Hall et al ,99], Hall et al proposent quand à eux de coder un individu par une matrice de M lignes et K colonnes où M représente les attributs et K le nombre de clusters à trouver. Ainsi, chaque ligne représente un centroïde et ses coordonnées. Un individu est donc de la forme présentée dans la figure 2.15:

Données/Clusters	c_1	c_2	…..	c_K
dim_1	0.8	47.0	…..	56.0
dim_2	10.5	51.3	…..	1.5
…..	…..	…..	…..	…..
dim_M	90.8	12.3	…..	41.6

Figure 2.15. Un individu représenté comme une matrice de centroïdes.

La fonction d'évaluation utilisée est la distance intra cluster. Le croisement appliqué est un croisement à deux points réalisé sur chaque centroïde de parents.

Krishma et al. [Krishna et al ,99] ont proposé un algorithme génétique hybride reprenant le codage présenté dans la figure 2.12, leur algorithme GKA utilise Kmeans comme un opérateur. L'opérateur de mutation qu'ils ont défini est un opérateur basé sur la distance. L'algorithme GKA partitionne un jeu de données en un nombre de clusters connus en minimisant la variance totale inter cluster.

Dans [Greene, 03] a été développé un algorithme génétique effectuant un clustering hiérarchique présentée sous la forme d'un arbre de centroïdes. Cet algorithme est restreint aux données numériques mais ne fait pas d'hypothèses sur le nombre de clusters.

2.4.2 Clustering par fourmis artificielles

Les fourmis réelles ont inspiré les chercheurs en informatique dans de nombreux domaines. Cela se justifie particulièrement quand on connaît la richesse comportementale de ces insectes. Plusieurs comportements observés chez les fourmis peuvent être directement mis en relation avec le problème du clustering [Azzag et al ,04b], l'exemple du tri collectif du couvain ou de la constitution de cimetières sont les plus marquants. Certains travaux expérimentaux montrent que certaines espèces de fourmis sont capables d'organiser spatialement divers éléments du couvain : les œufs, les larves et les nymphes.

Deneubourg apparaît comme un pionnier dans le domaine du tri d'objets par des fourmis artificielles. Dans [Deneubourg et al,90] il propose avec ses collègues les principes suivants : des fourmis artificielles se déplacent sur un plan. Les objets à rassembler sont répartis sur ce plan. Une fourmi ne dispose que d'une perception locale de ces objets et ne communique pas avec les autres. Au lieu de cela, la configuration des objets sur le sol va influencer leurs actions.

Lorsqu'une fourmi rencontre un objet, la probabilité qu'elle en ramasse est d'autant plus grande que cet élément est isolé.

Lorsqu'une fourmi transporte un objet, elle le dépose avec une probabilité d'autant plus grande que la densité d'objets du même type dans le voisinage est grande.

Ces règles relativement simples font qu'il apparaît des regroupements d'objets, et elles permettent ainsi de trier ces objets.

Cet algorithme a été à l'origine présenté pour des tâches en robotique. Lumer et Faieta ont modifié l'algorithme pour être applicable aux données numériques [Lumer et al ,94] ,

l'algorithme utilise une mesure de similarité entre les données (sous la forme d'une distance euclidienne). Les données sont initialement réparties aléatoirement sur une grille 2D. Chaque fourmi est située dans une case de cette grille et ne perçoit que les données situées dans son voisinage. Les probabilités vues précédemment ont été améliorées, elles dépendent de la moyenne des similarités entre une donnée portée par une fourmi et les données situées dans son voisinage. Une donnée sur la grille est ramassée avec une probabilité d'autant plus grande qu'elle est peu similaire aux données voisines. De la même manière, une donnée portée par une fourmi est plus facilement déposée dans une région comportant des données qui lui sont similaires.

En se basant sur ces travaux, une extension a été présentée dans [Monmarché,99], c'est l'algorithme Antclass où les fourmis peuvent empiler les objets les uns sur les autres dans une même case d'une grille qui est maintenant toroïdale afin que les fourmis passent d'un bord à l'autre. La taille de la grille est déterminée automatiquement. Lorsque les fourmis rencontrent un tas d'objets, elles peuvent se saisir de l'objet le plus dissimilaire. Antclass a présenté des erreurs, ce qui a poussé les auteurs à l'hybrider avec l'algorithme Kmeans. Cette hybridation consiste à utiliser la séquence d'algorithmes suivante : Antclass Kmeans, Antclass Kmeans.

Un autre algorithme Antclut a été proposé [Labroche et al ,02], il est inspiré du système chimique d'identification des fourmis. Dans ce système, les interactions continues entre les fourmis, leurs environnement et comportement mènent à la construction d'une odeur coloniale. Cette odeur construite par les fourmis sert à identifier qui fait partie du groupe et qui doit être rejeté. De la même manière, l'algorithme Antclut associe une donnée du jeu de données à une fourmi dont l'odeur est déterminée par les valeurs prises par les attributs décrivant cette donnée. Les fourmis effectuent des rencontres aléatoires et décident d'appartenir au même groupe ou non. A la fin, les fourmis artificielles qui partagent une odeur similaire sont groupées dans le même groupe, qui fournit le clustering attendue.

Par analogie au clustering hiérarchique, l'algorithme Anttree [Azzag et al,03] modélise le principe d'auto-assemblage observé chez une colonie de fourmis où des fourmis se fixent progressivement à un support fixe (racine de l'arbre) puis successivement aux fourmis déjà fixées afin de construire des structures vivantes. Les fourmis artificielles de Anttree vont de manière similaire construire un arbre. Chaque fourmi représente une donnée. Les déplacements et les assemblages des fourmis sur cet arbre dépendent de la similarité entre les données.

Des améliorations ont été apportées à Anttree pour l'adapter à la construction d'un arbre de documents permettant de générer automatiquement des sites portails [Azzag et al,06] .

2.4.3 Clustering par essaim de particules

Les algorithmes d'optimisation par essaim de particules, en anglais particule swarm optimization (PSO) ont été introduits en 1995 par Kennedy et Eberhart [Kennedy et al ,95]. Ces algorithmes sont inspirés des essaims d'insectes (ou des bancs de poissons ou des nuées d'oiseaux) et de leurs mouvements coordonnés. En effet, tout comme ces animaux se déplacent en groupe pour trouver de la nourriture ou éviter les prédateurs, les algorithmes à essaims de particules recherchent des solutions pour un problème d'optimisation. Les individus de l'algorithme sont appelés particules et la population est appelée essaim.

Le but de l'algorithme de PSO est de trouver la position de particule qui résulte à la meilleure évaluation d'une fonction fitness (objective) donnée.

Chaque particule représente une position dans un espace multidimensionnel, et elle circule dans cet espace de recherche, en ajustant sa position par apport à la meilleure position de la particule trouvée et la meilleure position dans le voisinage de cette particule.

Chaque particule a un ensemble d'attributs: vitesse courante, position actuelle, la meilleure position découverte par la particule et, la meilleure position découverte par la particule et ses voisins. La position d'une particule est la somme de sa position précédente et sa vitesse actuelle. La vitesse est ainsi calculée basée sur trois contributions: une fraction de la vitesse précédente, le composant cognitif qui est une fonction de la distance de la particule de sa meilleure position personnelle, et le composant social qui est une fonction de la distance de la particule de la meilleure particule trouvée.

Les algorithmes à essaim de particules ont été utilisés pour réaliser différentes tâches d'extraction de connaissances. Dans le cadre de clustering, Merwe et Engelbrecht ont proposé deux approches basées sur les essaims de particules [Merwe et al ,03].

Une particule représente un vecteur de K centroïdes de clusters, où K est le nombre de clusters et il doit être spécifié par l'utilisateur. Par conséquent, un essaim représente un certain nombre de vecteurs candidats pour un jeu de données. L'algorithme de PSO commence avec une population de vecteurs choisis aléatoirement, ensuite pour chaque vecteur, on assigne les points de données au centroïde de cluster le plus proche en terme de distance Euclidienne. La fitness de chaque vecteur est mesurée par l'erreur de quantification. L'évolution de la population est accomplie en ajustant les centroïdes de chaque vecteur par apport au meilleur vecteur de centroïde trouvé et le meilleur vecteur dans le voisinage de ce vecteur.

La deuxième approche consiste à améliorer l'algorithme PSO par le remplacement d'un individu de la population initiale par le résultat de l'algorithme Kmeans.

Dans [Xiao et al,03], les auteurs utilisent une méthode hybride basée sur les essaims de particules et les cartes organisatrices (Self-Organizing Maps) pour réaliser un clustering des gènes dans des expérimentations d'expression génique.

Les algorithmes à essaim de particules ont été utilisés aussi pour le clustering des images, l'algorithme cherche les centroids d'un nombre de clusters spécifié par l'utilisateur, chaque cluster regroupe les primitives d'images similaires [Omran,05].

2.4.4 Clustering par système immunitaire artificiel

Un système immunitaire artificiel est un système informatique basé sur des métaphores du système immunitaire naturel [Timmis,01]. Un système immunitaire est un complexe des cellules, des molécules et des organes qui visent à protéger le corps contre l'infection. En présence d'infections, les antigènes, les substances capables de stimuler une réponse immunitaire, sont produits. Le système immunitaire produit habituellement un groupe de B-cellules, qui sécrètent des anticorps. Ces anticorps peuvent identifier et s'attacher aux antigènes et finalement les tuer. L'affinité entre un antigène et un anticorps décrit la force de l'interaction d'antigène-anticorps, également référée comme affinité entre l'antigène et la B-cellule. Plus l'affinité entre un anticorps et un antigène est grande, plus l'anticorps peut s'attacher à l'antigène étroitement.

Le système immunitaire produit aléatoirement beaucoup de B-cellules. Les B-cellules avec une affinité élevée aux antigènes sont clonées. Ces cellules clonées peuvent facilement identifier et s'attacher aux antigènes et elles s'appellent ainsi les cellules mémoire. Ce processus de clonage qui produit des cellules mémoire s'appelle la sélection clonale.

Les cellules mémoire ont une plus longue vie que les B-cellules normales et elles sont ainsi utiles quand une infection semblable se produit à un temps futur. Les B-cellules qui ont une affinité faible aux antigènes sont directement éliminées ou mutées. Le processus de mutation reforme les anticorps attachés à la surface des B-cellules pour obtenir une affinité aux antigènes comparativement plus élevée. Ce processus qui augmente l'affinité s'appelle la maturation d'affinité.

La théorie de réseau immunitaire indique que le système immunitaire implique non seulement l'interaction d'anticorps et d'antigènes, mais aussi l'interaction d'anticorps avec d'autres anticorps. Les cellules peuvent se connecter l'une à l'autre pour former un réseau représentant une image interne des antigènes originaux. Le réseau peut répondre positivement ou négativement. Une réponse positive aurait comme conséquence la prolifération de cellules,

l'activation et la sécrétion d'anticorps. Une réponse négative mènerait à la suppression de réseau.

En ce qui concerne le clustering, le premier algorithme immunitaire est l'algorithme Ainet [Castro et al ,00], dans lequel chaque point de données est traité comme un antigène et représenté par un vecteur.

L'algorithme évolue une population des anticorps basés sur la théorie de réseau immunitaire, la sélection clonale et la maturation d'affinité. Ces anticorps forment un réseau, qui peut représenter les antigènes d'une manière compressée.

La procédure d'évolution des anticorps pour représenter des antigènes commence par la génération d'un ensemble d'anticorps d'une manière aléatoire, ensuite les étapes suivantes sont affranchies :

- Calcul d'affinité : Calculer l'affinité entre l'antigène et chaque anticorps, ce calcul est basé sur le distance Euclidienne entre les vecteurs qui représentent l'antigène et l'anticorps.
- Sélection clonale : sélectionner un sous-ensemble d'anticorps avec l'affinité la plus élevée et cloner les.
- Maturation d'affinité : muter chaque anticorps avec un antigène avec un taux inversement proportionnel à son affinité. En conséquence, les anticorps ayant les affinités les plus élevées subissent les mutations les plus petites.
- Suppression de réseau: Éliminez les anticorps redondants.

Ces étapes sont faites pour chaque antigène. L'algorithme continu à évoluer la population d'anticorps par refaire ces étapes durant un nombre fixe d'itération. À la fin l'algorithme arrive à placer des anticorps qui agissent comme des détecteurs d'antigènes de manière judicieuse et en nombre adapté aux données.

D'autres modèles plus complexes existent. Ainsi dans [Nasaroui et al, 02], l'algorithme utilise plusieurs niveaux de cellules et d'interaction et améliore l'algorithme une fonctions d'appartenance floue.

2.5 Autres types de Clustering

2.5.1 Clustering multiobjectif

Le clustering multiobjectif est un nouveau terme qui a vu le jour en 2004. Les méthodes existantes essayent explicitement ou implicitement, simultanément ou séparément d'optimiser

plusieurs fonctions objectives. Le but de clustering multiobjectif est de trouver des clusters dans des jeux de données en appliquant une de ces deux méthodes :

- L'optimisation de plusieurs fonctions objectives simultanément et d'une manière explicite, ces fonctions objectives représentent des mesures de qualité de clusters.
- La combinaison au sein d'un seul algorithme, plusieurs algorithmes de clustering correspondant aux différentes fonctions objectives. Dans ce cas l'optimisation multiobjective est faite d'une manière implicite.

La première méthode est incarnée dans les travaux de Handl et Knowles. Ils ont proposé MOEA [Handl et al,04] leur premier algorithme de clustering évolutionnaire multiobjectif. Il est basé sur un autre algorithme d'optimisation multiobjective PESA-II. MOEA optimise deux mesures de qualité de clusters : la déviation totale et la connectivité qui reflète le concept de compacité et de connectivité respectivement. MOEA exige que le nombre de clusters soit connu. Le problème majeur de MOEA est qu'une seule exécution fournit plusieurs solutions. Ce problème est surmonté dans la deuxième version MOCK [Handl et al,05c] par le choix d'une seule solution parmi les solution du front de pareto. Ce choix est basé sur le Gap statistic qui est une méthode statistique pour déterminer le nombre de clusters du jeu de données. A l'étape initiale, MOCK agit sur des solutions issus des deux algorithmes Kmeans et MST, puis l'évolution est faite en se basant sur l'algorithme PESA-II. . La sélection de la solution finale est basée sur une méthode inspirée de Gap statistic.

Une autre amélioration de MOCK est proposée dans [Handl et al,05a] , les modifications ont touché l'étape d'initialisation et quelques paramètres ainsi que l'enrichissement des jeux de données utilisés. Une autre version de MOCK autour des médoïdes à été proposée dans [Handl et al,05d].

Dans le cas de MOCK, l'application de l'optimisation multiobjective pour le clustering s'écroule quand des critères opposés sont optimisés simultanément comme les concepts de séparation spatiale entre des clusters et la connectivité [Handl et al,05c] . Cet inconvénient peut être vu comme un avantage dans l'approche proposé par Law, Topchy et Jain [Law et al,04]. Leur algorithme de clustering multiobjectif est un processus de deux étapes. Il inclut la détection des clusters par un ensemble de fonctions objectives candidates ainsi que leurs intégration dans une partition cible. La première étape emploie des algorithmes de clustering multiples qui peuvent être basés sur des techniques de résolution conflictuelles et génèrent un ensemble de clusters différents. La deuxième étape évalue la qualité de chaque cluster en utilisant la stabilité de cluster, cela est fait par introduire des variations dans la solution de clustering sous certaines perturbations des données. La perturbation est faite par

l'échantillonnage des données. Les clusters stables sont habituellement préférables, parce que si les mêmes clusters sont formés indépendamment des changements mineurs du jeu de données, les clusters sont robustes et par conséquent fiable. La sélection de clusters est accomplie par une simple heuristique.

Au sein de cet algorithme multiobjectif, une combinaison d'algorithmes a été inclut. Chacun de ces algorithmes possède des paramètres. Les valeurs de ces paramètres sont réglées dans des intervalles mais le choix de ces intervalles n'est pas évident, vu que ce réglage diffère d'un jeu de données à un autre. En outre, l'utilisation de plusieurs versions de chaque algorithme correspondant à plusieurs valeurs de paramètres peut conduire à un temps de calcul considérable.

2.5.2 Clustering flou

Les méthodes conventionnelles de clustering exact (hard clustering) limitent chaque point du jeu de données à exactement un seul cluster. Depuis que Zadeh [Zadeh,65] a proposé la théorie des ensembles flous qui a produit l'idée de l'appartenance partielle décrite par la fonction d'appartenance, le clustering flou a été largement étudié et appliqué dans une variété de secteurs. Le clustering flou qui emploie des techniques floues pour grouper des données, considère qu'un point peut être assigné à plus d'un cluster. Ce type d'algorithmes manipule l'incertitude des données réelles.

Dans la littérature sur le clustering flou, l'algorithme de clustering Cmeans flou ou FCM proposé par Dunn [Dunn,74] et étendu par Bezdek [Bezdek,81] est la méthode la plus connue et la plus utilisée.

Il est parfois appelé le Kmeans flou car il est une version flou du Kmeans qui est basé sur une extension flou de la fonction objective optimisé par Kmeans [Omran,05]. FCM essaye de trouver le point le plus caractéristique dans chaque cluster, qui peut être considéré comme un centre du cluster et, ensuite, le degré d'appartenance de chaque point aux clusters.

D'autres améliorations de FCM existent dans la littérature mais ils sont basés sur la théorie possibilistique comme le Cmeans possibilistique [Krishnapuram et al,96] et l'algorithme PCA [Yang et al,05].

2.6 Techniques de validation de Clustering

L'objectif principal de la validation de clusters est d'évaluer le résultat de clustering afin de trouver le meilleur partitionnement du jeu de données. Il existe des approches de validité de cluster pour évaluer quantitativement le résultat d'un algorithme de clustering [Halkidi et

al,01] mais dans la plupart des évaluations expérimentales des algorithmes, des jeux de données 2D sont employés pour que le lecteur soit capable de vérifier visuellement la validité des résultats (c-à-d, à quel point l'algorithme de clustering a découvert les clusters du jeu de données). Il est clair que la visualisation du jeu de données est une vérification cruciale des résultats de clustering. Dans le cas de grands jeux de données multidimensionnels (par exemple plus de trois dimensions) la visualisation efficace du jeu de données serait difficile.
De plus la perception de clusters à l'aide des outils de visualisation disponibles est une tâche difficile pour les gens qui ne sont pas habitués aux espaces d'un grand nombre de dimensions [Halkidi et al,02].
Il est commun de distinguer entre l'évaluation intrinsèque et extrinsèque. Les mesures de qualité externes emploient la connaissance externe. Beaucoup de ces derniers comparent le clustering à une autre partition. Les mesures de qualité internes n'emploient aucune connaissance externe, mais elles sont basées sur ce qui est disponible pour l'algorithme de clustering [Handl, 03].
Il existe deux critères qui ont été largement considérés suffisants pour mesurer la qualité du partitionnement de données [Halkidi et al,01b].
La compacité, les membres de chaque cluster doivent être proche l'un de l'autre autant que possible. Une mesure commune de compacité est la variance, qu'on doit minimiser.
La séparation, les clusters eux-mêmes doivent être largement espacée. La distance euclidienne entre les centroïdes de clusters donne une indication de la séparation de clusters.

2.6.1 Mesures externes

Le premier groupe de fonctions d'évaluation analytiques disponibles pour l'analyse de clusters est le groupe de fonctions conçues pour des problèmes de références pour lesquels le bon nombre de cluster et la classification correcte pour chaque point de données sont connus. L'évaluation devient beaucoup plus juste et sérieuse dans ces cas, puisque les propriétés désirées du partitionnement (qui conforme avec un certain degré à la définition de problème) peuvent être négligées, et nous pouvons seulement se concentrer sur la validité des assignements aux clusters obtenus [Handl, 03].
Les mesures externes que nous allons présenter, appliquent directement la connaissance des étiquettes de classes. Elles évaluent les clusters générés en prenant en compte les classes d'appartenances correctes.

- La F-mesure

La F-mesure est une fonction utilisée souvent dans la littérature pour évaluer les algorithmes de clustering. La F-mesure adopte les idées de la précision et du rappel de la recherche documentaire [Rijsbergen, 79]. Elle compare la qualité de clustering en tenant compte des classes correctes connues pour un jeu de données. Soit $C = (C_1, C_2, \ldots, C_k)$ un clustering donné et $R = (R_1, R_2, \ldots, R_{k'})$ les classes correctes.

Chaque classe R_i contient N_i points de données, chaque cluster C_j (généré par l'algorithme) est considéré comme l'ensemble de N_j points de données. N_{ij} donne le nombre de points de la classe R_i dans le cluster C_j et N donne le nombre total des points du jeu de données. Pour chaque classe R_i et un cluster C_j, la précision et le rappel sont alors défini comme :

$$\text{Prec}(R_i, C_j) = \frac{N_{ij}}{N_j} \quad et \quad \text{Rep}(R_i, C_j) = \frac{N_{ij}}{N_i}$$

Et la valeur de F-mesure correspondante est :

$$Fmes(R_i, C_j) = \frac{(b^2 + 1) \cdot \text{Prec}(R_i, C_j) \cdot \text{Rep}(R_i, C_j)}{b^2 \cdot \text{Prec}(R_i, C_j) + \text{Rep}(R_i, C_j)}$$

Où des coefficients égaux de $\text{Prec}(R_i, C_j)$ et $\text{Rep}(R_i, C_j)$ sont obtenu si b=1. La valeur globale de F-mesure F pour toute la partition est calculée comme

$$F(C) = \sum_{i=1}^{k'} \frac{N_i}{N} \max_{Ci \in C}(Fmes(R_i, C_j))$$

Elle est limitée à l'intervalle [0,1] et devrait être maximale.

- La pureté :

La pureté de cluster C_j $\hat{I}$ C est définie comme le pourcentage du type de données prédominant selon la classe réelle connue R_i $\hat{I}$ R, qui est :

$$Pur(C_j) = \max_{Ri \in R} \frac{N_{ij}}{N_j}$$

où N_j est la taille du cluster C_j et N_{ij} est le nombre des points de données de la classe R_i dans ce cluster. La pureté P(C) d'une partition entière est alors calculée comme la pureté moyenne de tous les clusters. Elle est limité à l'intervalle] 0,1] et devrait être maximale [Handl, 03].

- L'entropie

En outre, le degré relatif d'aspect aléatoire du partitionnement peut être évalué en utilisant la notion d'entropie de cluster. C'est une mesure plus complète que la pureté, car elle tient compte de la distribution de toutes les classes dans chaque cluster. L'entropie d'un cluster est :

$$Entr(C_j) = -\frac{1}{\log(N)} \sum_{Ri \in R} \frac{N_{ij}}{N_j} \log(\frac{N_{ij}}{N_j})$$

et, encore, l'entropie globale E(C) est calculée en faisant la moyenne des entropies de clusters. L'entropie de cluster est limitée à l'intervalle [0,1] devrait être minimale [Handl, 03].

- L'indice Rand

D'autres méthodes qui sont généralement appliquées pour comparer le résultat du clustering obtenu à la structure de cluster connue sont des indices basés sur des statistiques relative aux assignements de clusters, qui peut généralement être appliquée également pour évaluer la conformité entre deux partitions du même jeu de données.

Étant donné les partitions U et V, les quantités a, b, c et d sont calculées pour toutes les paires possibles de points de données p_i et p_j et leurs assignements de cluster respectives $C_U(p_i)$, $C_U(p_j)$,$C_V(p_i)$ et $C_V(p_j)$,où

$$a = \left|\left\{p_i, p_j \middle| C_U(p_i) = C_U(p_j) \wedge C_V(p_i) = C_V(p_j)\right\}\right|$$
$$b = \left|\left\{p_i, p_j \middle| C_U(p_i) = C_U(p_j) \wedge C_V(p_i) \neq C_V(p_j)\right\}\right|$$
$$c = \left|\left\{p_i, p_j \middle| C_U(p_i) \neq C_U(p_j) \wedge C_V(p_i) = C_V(p_j)\right\}\right|$$
$$d = \left|\left\{p_i, p_j \middle| C_U(p_i) \neq C_U(p_j) \wedge C_V(p_i) \neq C_V(p_j)\right\}\right|$$

Par conséquent, *a* et *d* maintiennent les correspondances entre les deux partitions, tandis que *b* et *c* calculent des déviations claires. L'indice le plus connu est l'indice Rand [Rand,71], qui est défini comme :

$$R(U,V) = \frac{a+d}{a+b+c+d}$$

Évidemment, *R* est limité à l'intervalle [0,1]. Une valeur de 1 est seulement obtenue pour une correspondance parfaite entre les assignements de clusters des partitions *U* et *V*, et les plus petites valeurs montrent un grand écart entre les deux solutions.

Quelques variations de cet indice existent, comme l'indice Rand ajusté [Hubert et al,85], qui présente une normalisation afin de rapporter des valeurs près de 0 pour des partitions aléatoires.

2.6.2 Mesures internes

Si on aborde des jeux de données dont la structure réelle est inconnue, l'évaluation des résultats de clustering devient beaucoup plus compliquée.

Les mesures qui peuvent être appliquées dans ces cas essayent de capturer les deux objectifs d'analyse de cluster : la minimisation de la distance intra cluster (qui résulte aux clusters

compacts) et la maximisation de la distance inter-cluster (qui résulte aux clusters bien-séparés).

- La variance intra cluster

La variance intra cluster optimisée implicitement par l'algorithme Kmeans, est basée sur le concept de la minimisation de la distance intra clusters. Elle est donnée par :

$$Var(C) = \sum_{Ci \in C} \sum_{p_j \in C_i} d(p_j - \mu_i)^2$$

Où p_j dénote un point de données, k dénote le nombre de clusters, μ_i représente le centroïde de cluster C_i, $d(.,.)$ est la fonction de distance utilisée pour calculer la déviation entre le point de données p_j et le centroïde μ_i

La limite inférieure de la variance qu'on peut obtenir dépend des données et le nombre de clusters employé, mais elle peut au meilleur être égal au zéro [Lioyd, 82].

- L'indice SD

L'indice SD [Halkidi et al,00] est une combinaison linéaire entre deux mesures de distance intra clusters et de distance inter clusters, il est donné par :

$$SD(C) = a\ Scat(C) + Dis(C)$$

où a est un terme de pondération qui fait un compromis entre l'importance relative de la dispersion moyenne des clusters $Scat(C)$ et la séparation totale entre les clusters $Dis(C)$, où

$$Scat(C) = \frac{1}{k}\sum_{i=1}^{k} N_k \frac{|s(m_i)|}{|s(C)|}, \qquad Dis(C) = \frac{D_{\max}}{D_{\min}} \sum_{i=1}^{k} \left(\sum_{j=1}^{k} d(m_i, m_j) \right)^{-1}$$

La l^{emme} dimension de $s(m_i)$ est $s_{m_i}^{l}$ et la l^{emme} dimension de $s(C)$ est s_C^{l}, ils sont donnés par les formules suivantes :

$$s_{m_i}^{l} = \sum_{j=1}^{Ni} (p_j^l - m_i^l)^2 \Big/ N_i \quad , \quad s_C^{l} = \sum_{j=1}^{N} (p_j^l - X^l)^2 \Big/ N$$

où X^l est la l^{emme} dimension de $X = \frac{1}{N}\sum_{j=1}^{N} p_j$

$\frac{D_{\max}}{D_{\min}}$ Dénote la proportion entre le maximum et le minimum de la distance entre les centroïdes de clusters. k est le nombre de clusters , N_i est le nombre de points de données assignées au cluster C_i, N est le nombre totale de points de données μ_i est le centroïde de cluster C_i.

- La famille des indices de *Dunn*

L'indice de validité de clusters proposé par *Dunn* [Dunn ,74] essaye d'identifier des clusters compacts et bien séparés. L'indice est défini pour un nombre spécifique de clusters. Il est donné par :

$$D_k = \min_{i=1,.....,k} \left\{ \min_{j=i+1,.....k} \left(\frac{d(C_i, C_j)}{\max_{l=1,.....,k} diam(C_l)} \right) \right\}$$

Où $d(C_i,C_j)$ est la fonction de dissimilarité entre deux clusters C_i et C_j défini comme :

$$d(C_i, C_j) = \min_{p \in Ci, q \in Cj} d(p,q)$$

Et *diam(C)* est le diamètre d'un cluster, qui peut être considéré comme une mesure de dispersion des clusters. Le diamètre d'un cluster *C* peut être défini comme suit :

$$diam(C) = \max_{p,q \in C} d(p,q)$$

Si le jeu de données contient des clusters compacts et bien séparés, la distance entre les clusters sera grande et le diamètre des clusters sera petit. Ainsi, basé sur la définition d'indice de *Dunn*, nous pouvons conclure que les grandes valeurs de l'indice indiquent la présence de clusters compacts et bien séparés.

Les implications de l'indice de *Dunn* sont le temps considérable requis pour son calcul, et la sensibilité à la présence de bruit dans les jeux de données, puisque ceux-ci vont probable augmenter les valeurs de *diam(C)*.

L'indice comme *Dunn* (index like *Dunn*) proposé dans [Pal et al, 97] est plus robuste à la présence de bruit. Il est basé sur le concept du MST (minimum spanning tree), il est donné par l'équation :

$$D_k = \min_{i=1,.....,k} \left\{ \min_{j=i+1,.....k} \left(\frac{d(C_i, C_j)}{\max_{l=1,.....,k} diam_l^{MST}} \right) \right\}$$

Il existe beaucoup d'indices basés sur l'indice de *Dunn*.

- L'indice de Davies-Bouldin

L'indice *DB* [Davies et al,79] est une fonction basée sur la minimisation du rapport des dispersions intra-clusters S_i et de la séparation inter-clusters introduite L_i par Davis et Bouldin. L'indice *DB* est donné par :

$$DB = \frac{1}{k} \cdot \sum_{i=1}^{k} L_i, \text{ avec}$$

$$L_i = \max_{\substack{j=1,\ldots,k, \\ i \neq j}} L_{ij} \quad et \quad L_{ij} = \frac{(S(C_i) + S(C_j))}{d(C_i, C_j)}$$

Une mesure typique de dispersion d'un cluster C_i est $S(C_i) = \frac{1}{|c_i|} \sum_{p \in ci} \|p - c_i\|$, et la mesure typique de distance entre clustres est la distance entre les centroïdes , $d(\mu_i, \mu_j)$. Donc, la proportion est petite si les clusters sont compacts et éloignés les uns des autres. Par conséquence, l'indice de Davies-Bouldin aura une petite valeur quand le clustering sera de bonne qualité.

- La connectivité

La mesure de connectivité de cluster évalue le degré auquel des points de données voisins ont été placés dans le même cluster. Elle est calculée par la formule suivante :

$$\text{Conn} = \sum_{i=1}^{m} \left(\sum_{l=1}^{L} p_{i,nn_i(l)} \right), \quad \text{où} \quad p_{r,s} = \begin{cases} 1/l & \text{if } \neg\exists c_j : r, s \in c_j \\ 0 & otherwise, \end{cases}$$

$nn_i(l)$ est l^{eme} le plus proche voisin du point de données p_i et L est le paramètre déterminant le nombre des voisins qui contribuent à la connectivité. La connectivité devrait être minimale [Handl et al,05c].

2.7 Conclusion

Nous avons vu dans ce chapitre les notions de base du clustering des données. Nous avons également présenté un nombre important de techniques de clustering, leurs points de forces et de faiblesses ainsi que les critères utilisés pour valider et évaluer ces techniques. La synthèse de ces techniques montre la difficulté du problème de clustering. En effet, il reste beaucoup de défis qui affrontent cette tache.

Les métaheuristiques, ayant déjà fait leurs preuves pour la résolution de problèmes combinatoires de grandes tailles, et elles semblent intéressantes pour le clustering. Dans ce sens, nous allons proposé une nouvelle approche dite évolutionnaire quantique pour traiter le problème de clustering des données. Il s'agit des algorithmes évolutionnaires enrichis par des concepts quantiques. Donc une présentation de l'informatique quantique qui a inspiré ces principes de la mécanique quantique paraît être nécessaire. Dans le prochain chapitre nous allons présenter les notions et les principes de base de l'informatique quantique ainsi que la combinaison entre des algorithmes classiques et quantiques et les avantages qui en résultent.

Chapitre 3

Les principes de bases de l'informatique quantique

"En 2020, quelques atomes pour un bit."
— Bertuli Roland

3.1 Introduction

Deux des grands courants scientifiques du vingtième siècle, la mécanique quantique et l'informatique se sont récemment rencontrés pour étudier ensemble dans quelle mesure des propriétés parfois étranges, que la mécanique quantique avait distinguées dans le comportement des particules élémentaires, peuvent être exploitées à des fins de représentation et de traitement de l'information. L'informatique quantique est née de cette rencontre interdisciplinaire, elle a immédiatement suggéré que des problèmes hors de portée de l'informatique classique pourraient être traités en exploitant ce paradigme informationnel radicalement non classique. Elle a ouvert des perspectives scientifiques et technologiques lointaines, certes, mais immenses.

Les chercheurs dans ce domaine travaillent pour la construction des ordinateurs quantiques qui sont plus rapides et plus efficaces que leurs homologues classiques, et ceci en utilisant les principes tirés des phénomènes quantiques tels que la superposition d'états, l'enchevêtrement et l'interférence.

En informatique classique, l'information est codée avec des bits qui sont soit dans l'état 0, soit dans l'état 1. Ces bits sont organisés en registres, et l'état d'un registre, qui code une valeur utilisée lors d'un calcul, n'est autre que la suite des états des bits qui composent ce registre. Au cours d'un calcul, l'état des registres pourra être transformé, pourra être recopié dans d'autres registres, puis l'état de l'un de ces registres sera lu à la fin pour produire un résultat. En informatique quantique, les bits seront donc des bits quantiques, des qubits : ils pourront être soit dans l'état 0, soit dans l'état 1, mais aussi, privilège du monde quantique, dans une superposition des deux, 0 et 1 à la fois. L'état d'un registre de n qubits pourra être une superposition d'un ensemble quelconque des 2^n valeurs possibles sur n bits alors qu'un registre de n bits classiques ne peut contenir, à chaque instant, qu'une seule de ces valeurs. Conséquence : comme les calculs transformeront l'état de tels registres, toute opération effectuée lors d'un calcul quantique pourra agir simultanément sur 2^n valeurs différentes. Ceci apporte un parallélisme massif.

Profitant de ces capacités, quelques algorithmes ont vu le jour, ils essayent de démontrer l'efficacité de l'informatique quantique pour la résolution des problèmes dont la solution classique est coûteuse en temps et en espace. Cependant, l'écriture d'un algorithme quantique est encore une tâche dure et les idées de l'informatique quantique ne sont pas encore exploitables vu le manque d'une machine quantique puissante. Mais les recherches ne sont pas bloquées par cette non disponibilité de machines quantiques vu que l'informatique quantique peut servir de base à de nouvelles méthodes qui aident à résoudre des problèmes en optimisation et en intelligence artificielle et beaucoup autres domaines. Plusieurs chercheurs s'intéressent à la combinaison entre des algorithmes classiques et les principes de l'informatique quantique. L'avantage est que ces algorithmes hybrides ne nécessitent pas la présence d'une machine quantique mais ils tirent parti de cette puissance quantique sur un ordinateur classique. Un exemple basé sur cette idée est le cas des algorithmes évolutionnaires quantiques [Han et al,02].

Dans ce chapitre, nous avons commencé par présenter la mécanique quantique, puis nous avons passé aux principes de bases de l'informatique quantique issus de la mécanique quantique. A la fin, la combinaison entre le classique et le quantique est élucidée par la présentation des algorithmes inspirés du quantique.

3.2 Mécanique quantique

Avec la théorie de la relativité, la mécanique quantique aura été la théorie scientifique la plus révolutionnaire du vingtième siècle. Elle nous permet d'accéder au monde de l'infiniment petit peuplé d'atomes, de photons, de neutrinos, de quarks et autres particules aux noms exotiques. C'est un monde bizarre et déroutant qui semble défier la logique et le bon sens. Pourtant, la théorie quantique a fait ses preuves, puisqu'elle est à l'origine des progrès technologiques.

La mécanique quantique est une théorie dans le sens mathématique: elle est régi par un ensemble d'axiomes. Les conséquences des axiomes décrivent le comportement de systèmes quantiques.

Les phénomènes quantiques mécaniques sont difficiles à comprendre puisque la plupart de nos expériences quotidiennes ne sont pas applicables. Les particules subatomiques agissent très différemment des objets dans le monde quotidien. Les particules peuvent être dans plusieurs endroits en temps. En outre deux particules bien séparées peuvent avoir des destins enchevêtrés, et l'observation d'une des particules causera la disparition de ce comportement remarquable. La mécanique quantique décrit cela et d'autres phénomènes physiques jaillissent extraordinairement [Grosshans, 02].

Une expérience simple telle que l'expérience de la polarisation des photons illustre certains des aspects clefs de la mécanique quantique nécessaire pour le calcul quantique.

3.2.1 Expérience de la polarisation des photons

Les seules particules que l'être humain peut observer directement sont les photons, pour cette raison, on a choisi d'utiliser une source puissante de lumière comme par exemple un pointeur laser et trois polaroïdes (filtres de polarisation) pour établir une expérience simple dite de polarisation des photons [Rieffel et al,00]. Elle consiste à orienter un faisceau de lumière vers un écran de projection et croiser ce faisceau avec des filtres placés entre ce dernier et l'écran de projection. Les filtres *A*, *B* et *C* sont polarisés horizontalement, à 45 degré et verticalement, respectivement. L'expérience passe par les 3 étapes suivantes :

Etape 1 : On commence par insérer le filtre *A* et on suppose que la lumière entrante est aléatoirement polarisée. Il résulte que l'intensité de lumière sortante aura la moitié de l'intensité de la lumière entrante. Les photons sortants sont maintenant tous horizontalement polarisés (figure 3.1).

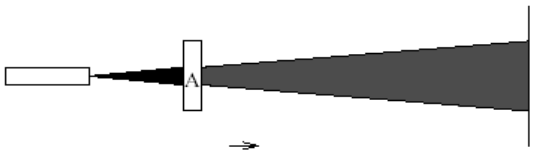


Figure 3.1. L'insertion du filtre A.

On ne peut pas expliquer la fonction du filtre *A* comme une simple passoire qui laisse passer seulement les photons qui s'avèrent être déjà horizontalement polarisés. Si c'était le cas, peu de photons entrants aléatoirement polarisés seraient horizontalement polarisés, ainsi nous attendrions une atténuation beaucoup plus grande de la lumière qui passe par le filtre.

Etape 2 : Quand le filtre *C* est inséré, l'intensité de la sortie baisse à zéro. Aucun des photons horizontalement polarisés ne peut passer par le filtre vertical. Un modèle de passoire pourrait expliquer ce comportement (figure 3.2).

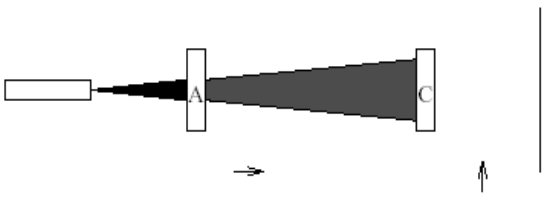


Figure 3.2. L'ajout du filtre *C*

Etape 3 : Finalement, après l'insertion du filtre *B* entre *A* et *C*, une petite quantité de lumière sera visible sur l'écran, exactement un huitième de la quantité originale de la lumière (figure 3.3).

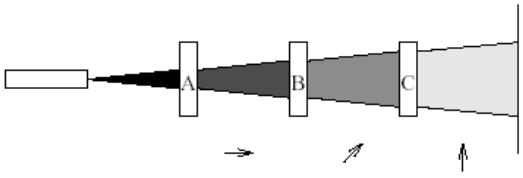


Figure 3.3. L'ajout du filtre *B*

L'intuition suggère que l'addition d'un filtre doit seulement être capable de diminuer le nombre de photons qui passe mais ce qui est déroulé est le contraire et de nouveau le modèle de passoire ne s'applique pas, on est devant un paradoxe.

Explication :

On peut modélisé l'état de la polarisation d'un photon par un vecteur dirigé, et n'importe quelle polarisation arbitraire peut être exprimée comme une combinaison linéaire :

$a\left|\uparrow\right\rangle + b\left|\rightarrow\right\rangle$ des deux vecteurs de base :

$\left|\uparrow\right\rangle$: Polarisation verticale.

$\left|\rightarrow\right\rangle$: Polarisation horizontale.

Le vecteur représentant l'état de la polarisation sera un vecteur unitaire qui vérifie la formule $|a|^2+|b|^2 = 1$, parce que nous ne sommes intéressés que par la direction de la polarisation. En général, l'état de la polarisation d'un photon peut être exprimée comme : $a\left|\uparrow\right\rangle + b\left|\rightarrow\right\rangle$.Où a et b sont deux nombres complexes tels que $|a|^2+|b|^2 = 1$. Le choix d'une base pour cette représentation est complètement arbitraire : n'importe quels deux vecteurs unitaires orthogonaux feront une base, par exemple : $\left|\,\mathrm{j}\right\rangle \mathrm{et} \left|\mathrm{k}\right\rangle$ ou $\left|\downarrow\right\rangle \mathrm{et} \left|\rightarrow\right\rangle$.

Le principe de la mesure de la mécanique quantique énonce que n'importe quel dispositif mesurant un système bidimensionnel a une base orthonormée associée. La mesure d'un état est la projection de cet état à l'un des vecteurs de la base du dispositif de mesure associé. La probabilité qu'un état $\left|\psi\right\rangle$est mesuré selon un vecteur de la base est égale au carré de l'amplitude du composant de la projection de cet état original sur le vecteur de la base.

Par exemple, étant donné un dispositif pour mesurer la polarisation de photons avec la base associée $\{\left|\uparrow\right\rangle,\left|\rightarrow\right\rangle\}$, L'état $\left|\psi\right\rangle = a\ \left|\uparrow\right\rangle + b\ \left|\rightarrow\right\rangle$ est mesuré selon $\left|\uparrow\right\rangle$ avec une probabilité $|\,a\,|^2$ et selon $\left|\rightarrow\right\rangle$ avec une probabilité $|\,b\,|^2$(figure3.4).

Il existe différents dispositifs de mesure qui peuvent avoir des bases associées différentes, et les mesures employant ces dispositifs auront des résultats différents. Les mesures sont toujours faites selon une base orthonormée.

En outre, la mesure d'un état quantique change cet état. C'est-à-dire si la mesure de l'état quantique $|\psi\rangle = a\ |\uparrow\rangle + b\ |\rightarrow\rangle$ résulte en $|\rightarrow\rangle$, alors l'état $|\psi\rangle$ est changé à $|\rightarrow\rangle$, et une deuxième mesure relativement à la même base va donner $|\rightarrow\rangle$ avec la probabilité 1.Ce que résulte c'est que la mesure change un état, et il n'est pas possible de déterminer ce qu'était l'état original.

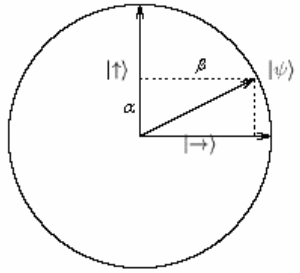


Figure 3.4. Mesure : Une projection sur la base

La mécanique quantique peut expliquer l'expérience de polarisation comme suit :

Un filtre mesure l'état quantique d'un photon selon la base constituée du vecteur correspondant à la polarisation du filtre avec un autre vecteur orthogonal à sa polarisation.

Les photons qui, après avoir été mesurés par le filtre, correspondent à la polarisation du filtre sont laissés passer à travers le filtre. Les autres ayant une polarisation perpendiculaire à celle du filtre sont reflétés. Par exemple, le filtre *A* mesure la polarisation de photon selon le vecteur de base $|\rightarrow\rangle$ correspondant à sa polarisation. Les photons qui passent par le filtre *A* ont tous la polarisation $|\rightarrow\rangle$. Ceux qui sont reflétés par le filtre ont tous la polarisation $|\uparrow\rangle$.

En supposant que la source lumineuse produit des photons avec une polarisation aléatoire, le filtre *A* horizontalement polarisé mesurera 50% de tous les photons. Ces photons traverseront le filtre et leur état sera $|\rightarrow\rangle$.

Finalement, le filtre *B* mesure l'état quantique d'un photon selon la base $\{|k\rangle, |j\rangle\}$, puisque le filtre est polarisé à 45 degré. Cette base est équivalente à $\{1/\sqrt{2}\ (|\uparrow\rangle + |\rightarrow\rangle),\ 1/\sqrt{2}\ (|\uparrow\rangle - |\rightarrow\rangle)\}$ car $|\rightarrow\rangle = 1/\sqrt{2}\ (|k\rangle - |j\rangle)$ et $|\uparrow\rangle = 1/\sqrt{2}\ (|k\rangle + |j\rangle)$.

Les photons passant par *A* avec état $|\rightarrow\rangle$ seront mesurés par *B* comme $|k\rangle$ avec une probabilité de 1/2 et donc 50 % des photons passant par *A* passeront par *B* et seront dans

l'état $|k\rangle$. De la même manière les photons à l'état $|k\rangle$ seront mesurés par le filtre *C* comme $|\uparrow\rangle$ avec une probabilité de 1/2. Ainsi seulement un huitième des photons originaux réussissent à passer par les filtres *A, B* et *C* (figure3.5).

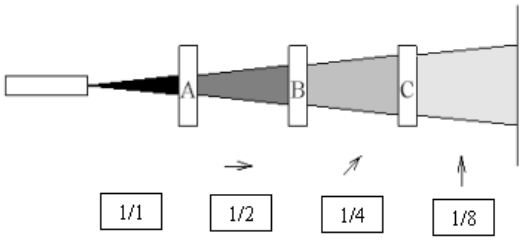


Figure 3.5. Le pourcentage de la lumière après chaque ajout des trois filtres.

3.2.2 Les quatre postulats de la mécanique quantique

La mécanique quantique est mathématiquement très bien définie, elle constitue une plateforme puissante qui définit ce qui peut et ne peut pas se produire dans les systèmes mécaniques quantiques, elle est fondée sur les quatre postulats suivants [Poulin,01] :

- L'état : on peut caractériser un système quantique d'une manière complète par un vecteur d'état dans un espace de Hilbert.
- L'évolution : l'état d'un système change avec le temps.
- La mesure : la mesure est une description de ce que peut être observée. Elle change le vecteur d'état du système à un nouveau vecteur d'état. Seulement certains ensembles de mesures peuvent être faits à n'importe quel moment.
- La composition des systèmes : l'espace d'états d'un système composé est le produit tensoriel des espaces d'états des systèmes constituants.

3.2.3 Espaces d'états de Hilbert, notation de Dirac et produit tensoriel

Un état est une description complète d'un système physique. En mécanique quantique, un état est un rayon dans un espace de Hilbert. Soit *H* un espace de Hilbert de dimension *n* et soit un vecteur de *H* exprimé en terme de ses composantes selon une base orthonormée arbitraire.

$$\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}$$

Suivant la notation de Dirac, ce vecteur est dénoté par un ket $|y\rangle$ tel que [Schoeb,99] :

$$|y\rangle = \begin{pmatrix} a_1 \\ a_2 \\ \mathbb{M} \\ a_n \end{pmatrix}$$

Si on réfère à l'expérience de la polarisation, les deux états de polarisation horizontale et verticale sont représentés en utilisant la notation de Dirac par $|\rightarrow\rangle$ et $|\uparrow\rangle$ respectivement, la même chose est faite pour $|\mathrm{k}\rangle$ et $|\mathrm{j}\rangle$ représentant les polarisations diagonales. Dans un système informatique quantique, les deux états discrets peuvent être exprimés par $|0\rangle$ et $|1\rangle$.

Un système quantique consistant de deux ou plusieurs états quantiques est le produit tensoriel des états séparés dans un certain ordre fixé. Par exemple, le produit tensoriel de deux kets $|y\rangle$ et $|f\rangle$ dans un espace de Hilbert de dimension 2 est un ket dans un espace de Hilbert de dimension 2×2 dont les composantes se calculent comme suit [Schoeb,99] :

$$|y\rangle \otimes |f\rangle = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \otimes \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} a_1 & b_1 \\ a_1 & b_2 \\ a_2 & b_1 \\ a_2 & b_2 \end{pmatrix}$$

Supposons que nous avons deux photons, *P1* et *P2*, où *P1* a l'état $|P1\rangle$ et *P2* a l'état $|P2\rangle$. Nous pouvons exprimer l'état du système commun en tant que $|P1 \otimes P2\rangle$, ou nous pouvons l'exprimer comme $|P1\ P2\rangle$.

3.3 Informatique quantique

L'informatique quantique est un domaine jeune qui, fondant le calcul sur des propriétés issues de la mécanique quantiques, tente d'apporter des solutions novatrices à différents types de problèmes informatiques. L'information quantique proprement dite ne naît en effet qu'au début des années 1980, époque où les physiciens utilisent quotidiennement des ordinateurs et considèrent l'information comme une quantité concrète, mesurable en bits et en octets [Tapp, 99] .

Le physicien Feynmann découvre en 1982 [Feynmann, 82] que certains systèmes quantiques sont exponentiellement difficiles à simuler sur un ordinateur classique, mais qu'il est néanmoins possible de les simuler en un temps raisonnable avec d'autres systèmes quantiques. La difficulté pour un système classique de simuler un système quantique vient de la taille de l'espace de Hilbert de ce dernier : en effet, un système de *n* spins (ou *n* qubits) est

décrit par un vecteur dans un espace de Hilbert de dimension 2^n, alors qu'un registre de *n* bits classiques évolue dans un espace de dimension *n* seulement. Cette différence induit une sorte de parallélisme massif du système quantique, celui-ci étant en quelque sorte dans tous les états à la fois au cours de son évolution. Un ordinateur quantique est un système qui exploite ce parallélisme [Deutsch, 85] pour résoudre un problème, que ce soit la simulation d'un autre système quantique ou un problème classique.

3.3.1 Bit quantique (qubit)

Les systèmes quantiques à deux états, comme la polarisation d'un photon unique, le spin 1/2 d'un électron et l'atome à deux niveaux , ont été utilisés très tôt comme systèmes modèles en mécanique quantique. Ce sont en effet les plus simples des systèmes quantiques : leur état est représenté par un vecteur dans un espace de Hilbert de dimension 2 seulement, ce qui ne les empêche pas de manifester tous les comportements paradoxaux de la mécanique quantique [Grosshans,02].

L'information quantique a rebaptisé ces systèmes quantiques qubits, ou bits quantiques, par analogie au bit classique, qui constitue l'unité de base de l'information ; il peut prendre les deux valeurs possibles {0, 1}. Ce terme qubit a été introduit par Shumacher. Comme sa contrepartie classique, un qubit peut prendre deux valeurs dénotées $|0\rangle$ et $|1\rangle$ .Un qubit vit donc dans un espace d'Hilbert à deux dimensions et peut être dans une superposition de ces deux états orthogonaux [Blais, 03]. Il est convenable de dénoter $\{|0\rangle, |1\rangle\}$ une base orthonormée pour un tel espace. Dans cette base, appelée souvent base standard, l'état quantique le plus général peut être exprimé comme :

$$a|0\rangle + b|1\rangle,$$

où *a* et *b* sont des nombres complexes tels que $|a|^2 + |b|^2 = 1$. Un qubit est n'importe quel état de cette forme. Comme on a vu à l'expérience de la polarisation des photons, on peut faire une mesure qui projette le qubit dans la base standard. Nous allons alors obtenir le résultat $|0\rangle$ avec probabilité $|a|^2$, et le résultat $|1\rangle$ avec probabilité $|b|^2$. De plus, excepté dans les cas où $a = 0$ ou $b = 0$, la mesure change de façon irrévocable l'état initial. Si la valeur du qubit est inconnue au départ, aucune mesure ne permettra de déterminer les valeurs de *a* et *b*. Cependant, après la mesure, le qubit a été préparé dans un état connu : soit $|0\rangle$ ou $|1\rangle$ [Schoeb,99].

3.3.2 Registre quantique

Le registre quantique est une notion très importante en informatique quantique. Le registre quantique est un système quantique constitué d'un ensemble de qubits. Nous pouvons décrire l'état d'un système de *n* qubits par [Poulin,01]:

$$|y\rangle = \sum_{i=0}^{2^n-1} a_i |i\rangle$$

où chaque nombre complexe a_i, représente l'amplitude de l'état de base $|i\rangle$.

Les amplitude a_i satisfont la propriété : $\sum_{i=0}^{2^n-1} |a_i|^2 = 1$

Un registre quantique peut exister dans le mélange de tous ses états permis simultanément, c'est ce qu'on appelle la superposition, par conséquent un registre quantique composé de *n* qubits peut être dans 2^n états en même temps. En revanche un registre classique composé de *n* bits peut être dans un seul état parmi ses 2^n états. Cela signifie que l'on peut stocker une quantité exponentielle d'information dans un registre quantique. Ici nous voyons certaines des premières allusions qu'un ordinateur quantique peut être exponentiellement plus puissant qu'un ordinateur classique [Bhalla et al,02].

Comme on le constate à partir du tableau 3.1, le nombre de bits classiques requis pour décrire correctement l'information quantique est énorme. Par exemple, pour 300 qubits (un nombre bien petit comparativement aux milliards de transistors dans un PC moderne) le nombre de bits classiques requis est plus grand que le nombre d'atomes dans l'univers visible ! Ceci signifie que si l'on prenait toute la matière contenue dans l'univers visible, nous n'aurions toujours pas assez de ressources pour décrire complètement l'état de ces 300 qubits! Évidemment, ceci implique que la simulation classique de simplement 300 qubits, sans mentionner un nombre plus grand de qubits, est sans espoir.

Nombre de qubits	**Nombre de bits classiques requis pour une description complète**
10	1024
20	1 048 580
30	1 0 73 470 000
300	Plus que le nombre d'atomes dans l'univers visible !

Tableau 3.1. Nombre de bits classiques requis pour une description complète d'un registre quantique.

3.3.3 Les principes de l'informatique quantique

L'informatique quantique hérite de la mécanique quantique ses principes de base suivants : la superposition d'états, l'interférence, l'enchevêtrement, le non déterminisme et la non clonage. Certaines de ces caractéristiques sont très déroutantes et sans équivalents dans la vie courante.

A. La superposition

Un qubit peut être dans l'état 0 comme il peut être dans l'état 1, et il peut être également dans les deux états en même temps. Par conséquent un registre quantique de *n* qubits peut être dans une superposition de 2^n état. Ce principe de superposition est la clé de la force d'un ordinateur quantique, il offre des capacités de traitement et de stockage exponentielles. Ce principe est celui qui conduit aux effets quantiques les plus déroutants: l'interférence et le non déterminisme [Bhalla et al,02].

B. L'interférence

Les particules subatomiques possèdent une caractéristique de dualité onde-corpuscule. Une particule subatomique peut se comporter à la fois comme une onde et un corpuscule. En fait, elle n'est ni l'un ni l'autre : c'est une entité beaucoup plus abstraite qui, selon les situations, donne l'impression de se comporter soit comme une onde, soit comme un corpuscule.

L'interaction des particules subatomiques avec leur environnement brouille très rapidement les superpositions des ondes, ce brouillage consiste à annuler et renforcer des ondes produisant ainsi l'effet d'interférence quantique [Nielsen,00]. Donc le phénomène d'interférence est applicable dans la mécanique quantique. Il a comme rôle d'augmenter (interférence constructive) ou diminuer (interférence destructive) l'amplitude d'un état et par conséquence sa probabilité d'être observé. L'interférence est très importante en calcul quantique. Elle permet d'augmenter la probabilité d'avoir les résultats espérés.

C. L'enchevêtrement

L'espace des états d'un système quantique composé de sous-systèmes et le produit tensoriel des espaces des états des sous-systèmes qui le composent : c'est là que réside une différence essentielle entre le monde classique et le monde quantique.

Considérons en effet un système classique composé de deux sous-systèmes : l'espace des états du système global est le produit cartésien des espaces d'états de chacun de ses sous-systèmes, et tout état du système global est un élément de ce produit, c'est-à-dire un couple

formé par un état de l'un des sous-systèmes et un état de l'autre. Par contre, grâce au phénomène de superposition des états quantiques, l'état global d'un système quantique composé de deux sous-systèmes ne sera pas, en général, un tel couple, mais une combinaison linéaire de tels couples. Les situations les plus générales engendrées de cette façon sont alors celles où l'état d'un sous-système n'existe plus que corrélé à celui de l'autre, des situations où seul le système global a un état propre. Ces situations sont désignées par le terme d'enchevêtrement : un système quantique est dans un état enchevêtré si son état n'est pas séparable en un produit direct d'états de systèmes plus simples.

Les états enchevêtrés les plus simples sont ceux des systèmes composés de deux qubits.

Pour la représentation, le traitement et la communication de l'information, l'enchevêtrement est la ressource intrinsèquement quantique. Il intervient de façon cruciale en algorithmique, il facilite la communication classique. Mais on est encore loin d'avoir une compréhension satisfaisante de l'enchevêtrement. Par exemple, il est nécessaire de pouvoir comparer des états enchevêtrés : certains états sont en effet plus enchevêtrés que d'autres. Dans le cas le plus simple, celui de deux qubits enchevêtrés, il existe ainsi plusieurs mesures d'enchevêtrement, dont la pertinence dépend des situations auxquelles on les applique. Pour des systèmes à plus de deux qubits, bien peu de choses de cette sorte sont connues [Le Bellac ,03].

D. Le non déterminisme

Contrairement à la physique classique, la mécanique quantique n'est pas déterministe, c'est-à-dire que les mêmes causes ne produisent pas nécessairement les mêmes résultats. Dans l'expérience de polarisation des photons et d'une manière aléatoire quelques photons apparaissent polarisés tandis que d'autres n'apparaissent pas du tout. Cette imprévisibilité n'est pas un manque de la connaissance. Ce n'est pas que nous manquons de la compréhension complète de l'état des photons. Le comportement aléatoire est vraiment une partie de la nature. Nous ne pouvons pas, même en principe, prévoir lequel des photons apparaîtront et qui ne sera pas [Nielsen,00] . De ce fait, on ne peut pas connaître la valeur d'un qubit avant la mesure, mais on peut augmenter la probabilité d'avoir un état en augmentant son amplitude.

E. Le non clonage

Un état quantique est constitué de plusieurs paramètres, par exemple la position et la vitesse d'une particule. Selon le principe d'incertitude de Heisenberg : On ne peut pas connaître précisément à la fois la position et la vitesse d'une particule .En fait, ce principe va beaucoup plus loin: en général, une particule ne possède pas de position et de vitesse bien définies et si l'on mesure avec précision un paramètre (la vitesse par exemple), l'état quantique de la

particule est perturbé. Ceci entraîne aussi le fait qu'on ne peut cloner une particule, car on ne peut jamais connaître complètement son état quantique. Cette impossibilité de copier parfaitement un état quantique est souvent désignée sous le nom de théorème de non-clonage [Grosshans,02].

3.3.4 La mesure quantique

La mémoire d'un ordinateur classique est faite de bits. Chaque bit porte soit un 1 soit un 0. La machine calcule en manipulant ces bits. Un calculateur quantique travaille sur un jeu de qubits ou registre quantique. Un qubit peut porter soit un 1 soit un 0, soit une superposition d'un 1 et d'un 0, par exemple un qubit porte une distribution de *phase*, angle qui pour 0° lui fait prendre la valeur 1, pour 90° la valeur 0, et entre les deux la superposition d'états dans les proportions du $\sin^2$ et du $\cos^2$ de la phase). Le calculateur quantique calcule en manipulant ces distributions ou superpositions. Le résultat du calcul est l'une des valeurs superposées dans le registre quantique. Comment extraire ce résultat ? En effectuant ce que les physiciens appellent une mesure de ce registre quantique: selon les lois de la physique quantique, la mesure pourra produire l'une quelconque des valeurs superposées dans le registre quantique, chacune avec une certaine probabilité et, en même temps, elle réduira la superposition que contenait le registre à une seule valeur, celle qui aura été choisie.

En retournant à l'exemple précédent, l'interrogation ou la mesure d'un qubit dont la phase θ n'est pas de 0° ou de 90° donne la réponse 0 avec la probabilité $\sin^2 \theta$, et la réponse 1 avec la probabilité $\cos^2 \theta$ (voir figure 3.6).

Comme la mécanique quantique est indéterministe le résultat de la mesure est probabiliste. En outre, la mesure de deux registres contenaient la même superposition ne produira forcément un résultat identique [Rieffel et al,00].

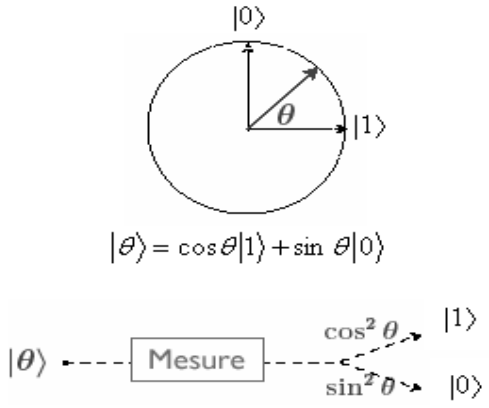


Figure 3.6 Mesure quantique

3.3.5 Calcul quantique et opération logique quantique

Le calcul quantique n'est rien de plus que l'évolution unitaire contrôlée d'un système quantique.

$$|y(t)\rangle = U(t)|y(0)\rangle,$$

où $|y(0)\rangle$ est l'état initial de l'ordinateur et $|y(t)\rangle$ l'état final qui, une fois mesuré, sera la réponse au calcul. L'opérateur unitaire d'évolution $U(t)$ représente le programme quantique : la dynamique du système est choisie de façon à correspondre au calcul à effectuer.
Une opération logique quantique est un opérateur d'évolution unitaire [Blais, 03]. Ce dernier peut être représenté par des portes quantiques. Ces portes servent à construire toutes sortes de circuits quantiques [Schoeb,99] .

3.3.6 Portes et circuits quantiques

Pour l'information classique, la possibilité de réaliser les portes logiques Not et Nand sur un ensemble de bits classiques permet de réaliser n'importe quel calcul classique. On dit donc que {Not,Nand} forment un ensemble universel (complet) pour le calcul classique . Pour le calcul quantique, un ensemble universel doit aussi être défini. Comme dans le cas classique, il existe plusieurs ensembles universels quantiques.
Un ensemble particulièrement utile est formé par une porte non-triviale à deux qubits et toutes les portes à un qubit [Blais, 03]. Dans le présent contexte, non triviale signifie que la porte peut créer de l'enchevêtrement. Les portes à un qubit peuvent être représenté par :

- Les matrices de Pauli

Une opération quelconque sur un qubit est décrite par une matrice unitaire 2×2, et peut être développée sur la base constituée de l'identité et des matrices de Pauli [Lévi,04] :

$$\begin{array}{ll} I: |0\rangle \to |0\rangle & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ \quad |1\rangle \to |1\rangle & \\ X: |0\rangle \to |0\rangle & \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\ \quad |1\rangle \to |1\rangle & \\ Y: |0\rangle \to -|1\rangle & \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \\ \quad |1\rangle \to |0\rangle & \end{array}$$

$$Z: \begin{matrix} |0\rangle \rightarrow & |0\rangle \\ |1\rangle \rightarrow - & |1\rangle \end{matrix} \quad \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Les noms de ces transformations sont conventionnels : *I* : est la transformation identité. *X* **:** est la négation. *Z*: est la transformation de décalage de phase. *Y = Z* X* est une combinaison des deux.

- La transformation de Hadamard

Une porte à un qubit souvent utilisée est la transformation de Hadamard H. Dans la base de calcul, cette opération a l'action et la représentation matricielle suivante [Blais, 03]:

$$\mathrm{H}: \begin{cases} |0\rangle \\ |1\rangle \end{cases} \rightarrow \frac{1}{\sqrt{2}} \begin{cases} |0\rangle + |1\rangle \\ |0\rangle - |1\rangle \end{cases} \qquad \mathrm{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

La transformation d'Hadamard permet notamment de passer de la base propre de $(|0\rangle, |1\rangle)$ à celle de $((|0\rangle + |1\rangle)/\sqrt{2}, (|0\rangle - |1\rangle)/\sqrt{2})$ et réciproquement [Lévi,04] .

Pour l'opération à deux qubits, la seule contrainte est qu'elle permette de créer de l'enchevêtrement. Une opération standard ayant cette capacité est le Non contrôlé.

- La porte Non contrôlé

Pour pouvoir construire n'importe quel opérateur, il faut ajouter aux opérations sur un qubit une porte qui agit sur au moins deux qubits simultanément. La porte la plus pratique est le Non contrôlé (CNOT), qui agit sur deux qubits : l'un est le qubit de contrôle, l'autre la cible. Si le contrôle est dans l´état $|0\rangle$ on ne fait rien, et s'il est dans l'état $|1\rangle$ le qubit cible est inversé. Dans la base naturelle $|00\rangle, |01\rangle, |10\rangle, |11\rangle$, l'action de la porte CNOT est décrite par [Lévi,04] :

$$CNOT: \begin{matrix} |00\rangle & \rightarrow & |00\rangle \\ |01\rangle & \rightarrow & |01\rangle \\ |10\rangle & \rightarrow & |11\rangle \\ |11\rangle & \rightarrow & |10\rangle \end{matrix} \quad \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

La combinaison des opérations logiques forme des circuits plus complexes.

3.3.7 Les algorithmes quantiques

L'exploitation des profits fournit par l'informatique quantique peut être accomplie par le développement des algorithmes quantiques améliorant la résolution de différents problèmes.

Un algorithme quantique peut se décrire comme une suite d'opérateurs unitaires élémentaires, appelés aussi portes quantiques, suivie de mesures quantiques [Lévi,04] .La puissance des algorithmes quantiques vient du parallélisme quantique [Rieffel et al,00]. Ce parallélisme résulte de la capacité du registre de mémoire quantique d'exister dans une superposition des états de base, et puisque le nombre d'états possibles est 2^n où n est le nombre de qubits dans le registre quantique, on peut exécuter dans une opération sur un ordinateur quantique ce qui prendrait un nombre exponentiel d'opérations sur l'ordinateur classique [Bhalla et al,02].

Il existe pour l'instant deux algorithmes principaux tirant partie des spécificités du calcul quantique. L'algorithme de Shor [Shor, 94] permet une amélioration exponentielle pour la factorisation de grands nombres, tandis que celui de Grover [Grover,96] accélère de façon quadratique la recherche d'un mot dans une liste.

La description d'un algorithme quantique n'est pas une tâche facile. Cette difficulté est due en partie à notre manque d'intuition pour des choses aussi étranges que superpositions d'états et enchevêtrement. Une autre difficulté avec les algorithmes quantiques est l'extraction de l'information. Un ordinateur quantique peut calculer simultanément un nombre exponentiel de résultats grâce au parallélisme quantique mais, à la fin, on ne peut en mesurer qu'un seul [Blais, 03]. La recherche de nouveaux algorithmes passe par l'identification de certains types de problèmes pouvant a priori bénéficier d'une résolution accélérée par un ordinateur quantique. Ce domaine de recherche constitue un défi considérable pour les chercheurs en algorithmique quantique.

3.3.8 Les ordinateurs quantiques : rêve ou réalité

Dans un ordinateur quantique le support physique traitant l'information obéit aux lois de la physique quantique. Les bits deviennent des qubits et sont constitués de systèmes à deux niveaux. Un qubit peut être dans une superposition cohérente de ses deux états. Un registre, constitué d'un ensemble de qubits, peut également être dans une superposition cohérente de différents états. En d'autres termes le nombre écrit dans un registre peut prendre plusieurs valeurs à la fois. La manipulation d'un tel registre dans un ordinateur quantique permet l'exploration simultanée de situations correspondant aux différentes valeurs du registre. Les informations appropriées extraites du calcul tirent alors parti d'effets d'interférence. Le résultat obtenu dépendra alors des différents chemins suivis par les différentes valeurs du registre. Les superpositions quantiques peuvent être non séparables, c'est-à-dire que seul l'état de l'ensemble du registre est connu sans que l'on puisse déterminer l'état d'un seul qubit. La quantité d'information contenue dans ces états non séparables (ou enchevêtrés) est

exponentiellement plus grande que dans un système classique de même taille. Ce ``parallélisme'' quantique permet donc l'exploration d'un espace bien plus grand pour un nombre d'opérations donné.

La réalisation pratique d'un tel ordinateur doit cependant faire face à des difficultés expérimentales et théoriques énormes et non encore résolues. La difficulté principale est d'arriver à préserver la cohérence quantique pendant toute la durée du calcul. Cela signifie que les qubits, tout en étant couplés entre eux, doivent être complètement découplés du monde extérieur. Plus le nombre de qubits est grand, plus la probabilité que l'un des qubits interagisse avec l'environnement est grande, et donc plus la probabilité que le calcul échoue est grande. Or le nombre de qubits nécessaire à la réalisation d'une tâche utile (plusieurs milliers pour factoriser un nombre de 200 chiffres) est pour l'instant bien au-delà du record actuel de l'enchevêtrement de 7 qubits.

Une découverte essentielle pour tenter de résoudre le problème de décohérence qui est l'obstacle majeur à la réalisation d'un ordinateur quantique, est l'existence de code correcteur d'erreurs quantiques. Le principe utilisé est de coder l'état d'un qubit de manière redondante sur une superposition judicieusement choisie de plusieurs qubits. L'altération d'un des qubits de la superposition est alors détectée et corrigée sans extraire d'information sur l'état du qubit initial, ce qui préserve la cohérence quantique du registre. Par contre, les codes correcteurs d'erreurs actuels ne sont efficaces que lorsque les taux d'erreurs sont très faibles (typiquement inférieurs à 10^{-5}) [Gautier et al, 01].

3.4 Algorithmes inspirés du quantique

Entre rêve et réalité, les algorithmes quantiques purs confrontent beaucoup de problèmes partant de la non disponibilité de machines quantiques à la difficulté de conception. Les défis à relever sont considérables, tant du point de vue théorique que du point de vue technologique.

En parallèle aux recherches dans le domaine de calcul quantique pur, d'autres recherches sont conduites dans le sens de combinaison des algorithmes classiques avec les principes de l'informatique quantique. L'avantage des algorithmes inspirés du quantique est qu'ils ne nécessitent pas d'être exécuté sur un ordinateur quantique.

L'hybridation des algorithmes évolutionnaires et calcul quantique se ramène aux travaux de Han et Kim. Ils ont conçu des algorithmes pour résoudre des problèmes d'optimisation combinatoire. Ils ont proposé en 2000, leur premier algorithme génétique inspiré du quantique pour résoudre le problème du sac à dos [Han et al, 00]. Cet algorithme a présenté de bons

résultats par apport aux algorithmes génétiques classiques. La chose qui les a encouragé à proposer d'autres algorithmes évolutionnaires inspiré du quantique [Han et al,02] [Han et al,04]. Le cercle de problèmes d'optimisation combinatoire résolus par algorithmes évolutionnaires quantiques a grandi en comprenant d'autres problèmes comme le problème d'allocation de disque [Han et al,03] , la détection de face [Jang et al,04] et beaucoup d'autres.

3.4.1 Principe d'algorithme évolutionnaire quantique

Les algorithmes évolutionnaires sont principalement une méthode d'optimisation et de recherche stochastique basée sur les principes de l'évolution biologique naturel. Les algorithmes évolutionnaires fonctionnent sur une population de solutions potentielles, appliquant le principe de survie du plus convenable pour produire successivement les meilleures approximations à une solution. À chaque génération de l'algorithme évolutionnaire, un nouvel ensemble d'approximations est créé par le processus de sélection selon la fonction fitness et la reproduction employant des opérateurs de variation. Ce processus peut mener à l'évolution des populations des individus qui sont mieux adaptés à leur environnement que les individus dont ils ont été créés, juste comme dans l'adaptation naturelle. Les algorithmes évolutionnaires sont caractérisés par la représentation de l'individu, la fonction d'évaluation des individus, et la dynamique de population comme les opérateurs de variation, la sélection de parent , la méthode de concurrence de survie, la taille de population, la reproduction, etc. Pour avoir un bon équilibre entre l'exploration et l'exploitation, ces composants doivent être conçus correctement [Jourdan, 03].

Les algorithmes évolutionnaires présentent quelques limitations comme la grande taille de la population utilisée pour trouver la meilleure solution, ce qui augmente le temps de calcul.

Récemment des chercheurs ont investigué des concepts quantiques pour combattre ces limitations afin d'explorer l'espace de recherche avec le petit nombre d'individus et exploiter les solutions dans une durée de temps courte. Une nouvelle classes d'algorithmes est donc née : les algorithmes quantiques évolutionnaires ou en anglais Quantum evolutionary algorithm (QEA).

Un algorithme quantique évolutionnaire (QEA) est un algorithme issu de l'hybridation des algorithmes évolutionnaires avec des concepts et principes de l'informatique quantique, comme le bit quantique, la superposition des états, la mesure, l'interférence, etc. Comme un algorithme évolutionnaire, l'algorithme évolutionnaire quantique est également caractérisé par la représentation de l'individu, la fonction d'évaluation, et la dynamique de population.

Cependant, au lieu d'une représentation binaire, numérique, ou symbolique, l'algorithme évolutionnaire quantique emploie une représentation quantique basée sur les qubits. Des portes quantiques sont aussi définies comme des opérateurs de variation de l'algorithme évolutionnaire quantique, pour conduire les individus vers de meilleures solutions [Han et al,02] .

3.4.2 Représentation quantique des individus

QEA emploie une nouvelle représentation quantique basée sur le concept de qubit. Un qubit est la plus petite unité d'information, avec une paire de nombres complexes (a,b). L'état d'un qubit peut être représenté par :

$$|y\rangle = \begin{pmatrix} a \\ b \end{pmatrix} \text{ où } |a|^2 + |b|^2 = 1$$

$|a|^2$ donne la probabilité que le qubit sera trouvé dans l'état " 0 " et $|b|^2$ donne la probabilité que le qubit sera trouvé dans l'état " 1 ". Un qubit peut être dans l'état " 1 ", dans l'état " 0 ", ou dans une superposition linéaire des deux. Un individu peut être représenté par un registre de m qubits définit par :

$$\begin{pmatrix} a_1 & a_2 & \mathbf{L} & a_m \\ b_1 & b_2 & \mathbf{L} & b_m \end{pmatrix} \text{ où } |a_i|^2 + |b_i|^2 = 1,\ i = 1,2,\mathbf{K}\, m.$$

La représentation quantique basée sur les qubits a l'avantage qu'elle peut représenter une superposition linéaire des états d'une manière probabilistique. La représentation quantique a une meilleure caractéristique de produire la diversité dans la population que toutes les autres représentations. Par exemple, un système de trois qubit avec trois paires d'amplitudes représenté comme suit :

$$\begin{pmatrix} \frac{1}{2} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{\sqrt{3}}{2} & \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}$$

Alors les états du système peuvent être représentés par :

$$\frac{1}{4}|000\rangle + \frac{1}{4}|001\rangle - \frac{1}{4}|010\rangle - \frac{1}{4}|011\rangle + \frac{\sqrt{3}}{4}|100\rangle + \frac{\sqrt{3}}{4}|101\rangle - \frac{\sqrt{3}}{4}|110\rangle - \frac{\sqrt{3}}{4}|111\rangle$$

Le résultat ci-dessus signifie que les probabilités de représenter les états : $|000\rangle$, $|001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle, |111\rangle$ sont : $\frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{3}{16}, \frac{3}{16}, \frac{3}{16}, \frac{3}{16}$

respectivement. Par conséquent, ce système de 3 qubits contient l'information de 8 états. En

représentation classique, ces 8 états nécessitent pour être représentés 8 registres classiques de 3 bits. Ce qui signifie clairement que la représentation quantique est une représentation condensée qui permet de diversifier la population et de réduire l'espace de stockage.

3.4.2 Structure générale d'un algorithm évolutionnaire quantique

L'algorithme évolutionnaire quantique (QEA) maintient une population quantique, sur laquelle il effectue des opérations telle que l'interférence, la mesure, l'évaluation, la sélection et la migration globale et locale. La structure générale d'un algorithme quantique évolutionnaire peut être décrite dans la figure 3.7 suivante :

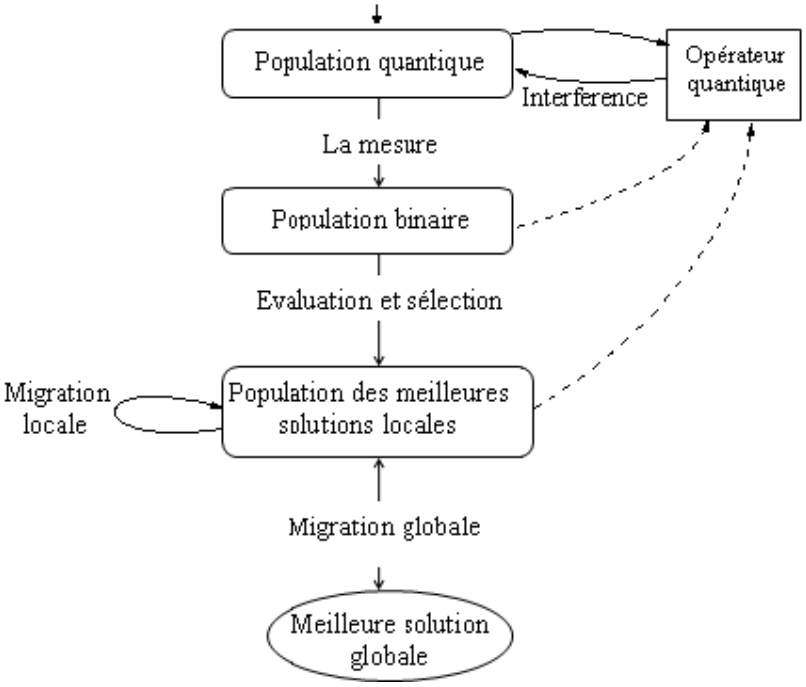


Figure 3.7. Structure générale d'un algorithme quantique évolutionnaire

3.4.3 La mesure

Cette opération permet de générer un individu binaire par l'observation des états quantiques d'un individu quantique. L'observation est faite par sélectionner pour chaque bit et selon les probabilités, une valeur 0 ou 1 parmi la superposition des états. Donc, on aura une solution binaire parmi toutes les solutions présentes dans la superposition. Mais contrairement à la théorie quantique pure, cette mesure ne détruit pas la superposition. Cela a l'avantage de préserver la superposition pour les itérations suivantes sachant qu'on opère sur des machines classiques (figure 3.8).

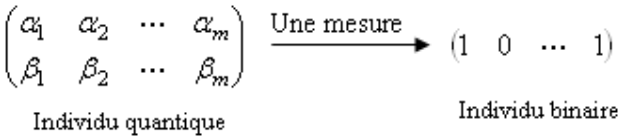


Figure 3.8. Extraction d'un individu binaire à partir d'un individu quantique par une opération de mesure.

3.4.4 Interférence et opérateur quantique

L'interférence introduit des changements sur les individus quantiques afin de diversifier et évoluer la population quantique. L'interférence est de deux sortes, constructive et destructive, l'interférence constructive permet d'augmenter la probabilité d'obtenir un état tandis que l'interférence destructive permet de diminuer la probabilité d'obtenir un état. D'une autre façon, l'interférence augmente la chance d'une solution d'être mesurée par l'opération de mesure. L'interférence constructive consiste essentiellement à déplacer l'état de chaque qubit dans la direction de la valeur du bit correspondant dans la meilleure solution en cours. Cela permet d'intensifier la recherche autour de la meilleure solution en cours. L'interférence destructive consiste à déplacer l'état de chaque qubit loin de la valeur du bit correspondant dans la meilleure solution en cours. Cela permet d'explorer d'autres solutions afin d'échapper aux minimums locaux.

Cette opération d'interférence peut être accomplie en utilisant des portes quantiques. Le choix ou la conception d'une porte quantique doit être comforme au problème à résoudre.

Dans [Han et al,02], une porte quantique de rotation a été utilisée, afin d'augmenter la probabilité de la bonne solution :

$$U(\Delta q_i)=\begin{bmatrix}\cos(\Delta q_i) & -\sin(\Delta q_i)\\ \sin(\Delta q_i) & \cos(\Delta q_i)\end{bmatrix}$$

Où : $\Delta q_i, i=1,2,...,m$ est un angle de rotation de chaque qubit vers l'état 0 ou 1 dépendant de son signe. Δq_i est un paramètre empirique dépendant du problème à résoudre (figure 3.9).

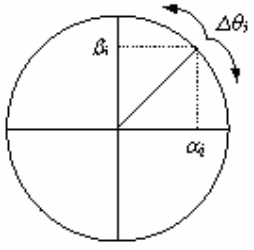


Figure 3.9. Interférence quantique basée sur la rotation.

D'autres portes quantiques que nous avons déjà présentés peuvent être utilisées, comme la porte de négation, le non contrôlé et la porte de Hadamard. La porte de négation change la

probabilité de l'état 1 (ou 0) à l'état 0 (ou 1) .Elle peut être employé pour échapper à un optimum local. Dans le non contrôlé, un des deux bits doit être un bit de contrôle. Si le bit de contrôle est à 1, l'opération de négation est appliquée au deuxième bit. Le non contrôlé peut être employé pour les problèmes où il y a une grande dépendance entre deux bits. La porte de Hadamard convient aux algorithmes qui emploient l'information de phase du qubit aussi bien que l'information d'amplitude [Han et al,02].

3.4.5 Migration globale et locale

La migration est une opération qui s'effectue au sein de la population binaire, cette population est divisée en groupes, pour chaque groupe on sélectionne sa meilleure solution. La migration locale consiste à copier la meilleure solution d'un groupe dans les autres solutions du groupe, tandis que la migration globale consiste à sélectionne la meilleure solution globale de toute la population et la copier dans les solutions de population. L'opération de migration peut induire une variation des probabilités des individus quantiques.

3.5 Conclusion

L'informatique quantique est un sujet très riche et extrêmement stimulant grâce aux capacités de traitement et de stockage qu'il donne. Son gain considérable en temps et en calcul a conduit à l'apparition des approches hybrides. Une hybridation entre les algorithmes évolutionnaires classiques et l'informatique quantique a donné naissance aux algorithmes évolutionnaires quantiques qui ont montré leur efficacité dans plusieurs problèmes combinatoires. Cela nous a encouragé à proposer une nouvelle approche basée sur les algorithmes évolutionnaires quantiques pour le célèbre problème de clustering des données qui est une démarche très courante permettant de mieux comprendre l'ensemble des données analysé.

Chapitre 4

Une approche évolutionnaire quantique pour le Clustering des données

"You will recognize your own path when you come upon it,
because you will suddenly have all the energy and imagination you will ever need"
— Jerry Gillies

4.1 Introduction

La découverte des clusters présents dans les données est un but de longue date. Cette tache a vu la naissance d'une grande variante de méthodes essayant de la résoudre mais sa difficulté l'a qualifiée comme une tache de défi. A nos jours, cette tache de clustering est un sujet de recherches actives. Les chercheurs minent dans d'autres domaines, creusent dans la nature, s'inspirent des insectes, tentent de trouver d'autres modèles pour le clustering. Dans ce sens, nous somme intéressés par le monde de l'infiniment petit, ce monde miraculeux des atomes et des phénomènes quantiques qui les régissent. Plus précisément, nous somme intéressés par le calcul quantique qui est un nouveau paradigme émergeant en informatique qui suscite un intérêt de plus en plus croissant en raison des nouveaux concepts qu'il suggère pour stocker et traiter des données.

Le travail décrit dans ce chapitre est la première tentative d'aborder le problème de clustering de données en utilisant le paradigme évolutionnaire quantique. Nous montrons comment il peut être exprimé comme une tâche d'optimisation et résolue efficacement en utilisant une nouvelle méthode basée sur une représentation quantique pour coder l'espace de recherche et une stratégie de recherche évolutionnaire quantique pour optimiser une mesure de qualité de cluster afin de trouver un bon partitionnement du jeu de données.

Nous avons proposé deux approches QEAC et QEAC2. La deuxième approche est un enrichissement et une amélioration de la première.

Dans ce chapitre, nous avons commencé par formuler le problème de clustering. Nous avons passé ensuite à décrire QEAC et les modifications et les ajouts qui nous ont conduit à QEAC2. Une partie importante de ce chapitre est destinée à l'évaluation. Nous avons décortiqué et analysé les résultats sur des jeux de données réels et synthétiques afin de démontrer et d'extraire des propriétés intéressantes de notre approche.

4.2 Formulation du problème

D'une façon informelle, le clustering est le processus qui permet d'identifier des groupes (clusters) homogènes au sein d'un ensemble de données multidimensionnelles. Le regroupement est fait de telle manière que les données d'un même cluster sont les plus similaires que possible les unes des autres au sens d'un certain critère de similarité et les données appartenant à des clusters différents sont les plus dissimilaires que possible.

Formellement, le problème de clustering peut être défini comme un problème d'optimisation : Étant donné un jeu de données *S*, une mesure de distance *d(i,j)* pour *i* et *j* dans *S* et une fonction objective *f(C,d(.,.))* , nous cherchons une partition $C=\{c_1, c_2,\ldots, c_k\}$ de *S* où chaque c_i représente un cluster et *k* le nombre de clusters censé être spécifié par l'utilisateur. La partition devrait satisfaire les conditions suivantes:

1. $\forall i \quad c_i \neq f$
2. $\forall i, j \quad c_i \cap c_j = f$
3. $\bigcup_i c_i = S$
4. *min* *f(C,d(.,.))*

L'espace de recherche est ainsi l'espace de toutes les partitions potentielles.

4.3 Complexité du problème de clustering

Le nombre de façons de classer *n* objets dans *k* clusters est donné par Liu [Liu,68] :

$$R(k,n) = \frac{1}{k!}\sum_{i=0}^{k}(-1)^i \binom{k}{i}(k-i)^n$$

Par conséquent, même avec un nombre fixe *k*, l'espace de recherche pour le problème de clustering s'accroît exponentiellement. Par exemple, si nous prenons *n*=25 et *k*=5, il y a 2 436 684 974 110 751 façons de répartir les 25 objets dans 5 clusters [Anderberg ,73]. Il est donc évident qu'un parcours exhaustif des solutions est impossible même pour des jeux de données de taille moyenne. En effet, le problème de clustering est connu pour être NP-complet dans plusieurs de ses définitions.

Les méthodes traditionnelles ne travaillent que sur un petit sous-ensemble de l'espace de recherche (le sous-ensemble étant défini par le nombre de clusters, le critère de clustering et la méthode de clustering). Ces méthodes traditionnelles obtiennent donc, en général, des optima locaux et rarement globaux.

Les métaheuristiques, ayant déjà fait leurs preuves pour la résolution de problèmes combinatoires de grandes tailles, elles semblent intéressantes pour se dégager de ces optima locaux et trouver de façon plus fréquente les optima globaux [Jourdan, 03].

4.4 Une approche évolutionnaire quantique pour le Clustering des données QEAC

Dans cette section, nous décrivons comment des concepts de calcul quantique ont été employés pour accomplir le clustering des données. Deux particularités principales caractérisent l'approche proposée: La représentation quantique de l'espace de recherche et la dynamique évolutionnaire quantique. Étant donné un jeu de données à partitionner, l'idée principale consiste à l'optimisation d'une mesure de qualité de cluster afin de trouver une partition du jeu de données. Durant le processus d'optimisation, des opérations quantiques sont appliquées sur des individus quantiques. La population quantique évolue à travers des générations jusqu'à ce qu'un critère d'arrêt soit satisfait.

4.4.1 Représentation quantique d'une partition

Une partition peut être codée sous forme d'une matrice binaire dénotée par *BM* où chaque ligne représente un cluster et chaque colonne indique un point de données qui est un élément du jeu de données. La valeur d'un élément x_{ij} de cette matrice est mise à 1 pour indiquer que le point de données p_j correspondant à la colonne j appartient au cluster c_i et 0 autrement (voir figure 4.1).

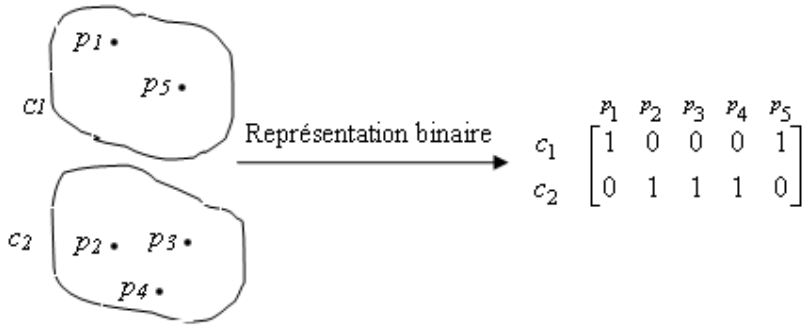


Figure 4.1 Une représentation binaire de deux clusters et cinq points

De ce codage binaire, une représentation quantique peut être facilement tirée. En effet, elle consiste à une matrice quantique dénotée par *QM* (voir figure 4.2) identique dans sa structure à la matrice binaire *BM*, mais différente d'elle dans deux sens:

Premièrement, chaque élément q_{ij} de *QM* est en fait un qubit $\begin{pmatrix} a_{ij} \\ b_{ij} \end{pmatrix}$ où a_{ij} et b_{ij} sont les amplitudes de probabilités satisfaisant la propriété *:* $|\alpha_{ij}|^2+|\beta_{ij}|^2 = 1$. La valeur $|\beta_{ij}|^2$ est interprétée comme la probabilité d'assigner un point de données p_j au cluster c_i et la valeur $|\alpha_{ij}|^2$ est la probabilité du cas inverse. De cette manière, la représentation quantique d'un cluster est un registre quantique contenant la superposition de toutes les combinaisons possibles de points de données dans le cluster. Par conséquent la deuxième différence principale de ce codage est sa capacité de représenter toutes les partitions potentielles au lieu d'une seule. C'est en fait une représentation probabiliste de toutes les configurations d'assignements de points de données aux clusters.

$$\begin{array}{c} \\ c_1 \\ \mathbb{M} \\ c_k \end{array} \begin{array}{c} \begin{array}{ccccc} p_1 & & \mathbf{L} & & p_m \end{array} \\ \begin{bmatrix} \begin{pmatrix} a_{11} \\ b_{11} \end{pmatrix} & \begin{pmatrix} a_{12} \\ b_{12} \end{pmatrix} & \mathbf{L} & \begin{pmatrix} a_{1m} \\ b_{1m} \end{pmatrix} \\ & \mathbb{M} & & \\ & \mathbb{M} & & \\ \begin{pmatrix} a_{k1} \\ b_{k1} \end{pmatrix} & \begin{pmatrix} a_{k2} \\ b_{k2} \end{pmatrix} & \mathbf{L} & \begin{pmatrix} a_{km} \\ b_{km} \end{pmatrix} \end{bmatrix} \end{array}$$

Figure.4.2 Représentation quantique de partitions potentielles : *m* est le nombre de points de données et *k* est le nombre de clusters.

4.4.2 Principe de l'approche proposée : QEAC

Pour une raison de clarté, nous décrivons d'abord la structure générale de l'approche proposée, ensuite nous soulignons chacune de ses étapes principales. Comme nous n'opérons pas sur un ordinateur quantique et afin de maintenir la diversité des individus, nous employons une population de *n* individus quantiques où son i^{eme} individu est dénoté par QM_i et une population de *n* individus binaires où son i^{eme} individu est dénoté par BM_i. La meilleure partition binaire locale trouvée par le i^{eme} individu est dénotée par B_i et la meilleure partition binaire globale trouvée à la fin du processus d'évolution est dénotée par *Pbest*. En commençant avec une population initiale, le processus consiste à évoluer cette population en appliquant quelques opérateurs quantiques de base : la mesure et l'interférence. La figure 4.3 inspirée de [Han et al,02] schématise la structure de l'approche proposée et l'algorithme est donné dans ce qui suit:

INPUT Jeu de données *S*
Begin
t←0
i←0
Repeat
Initialiser (QM_i);
i←i+1
Until (i=*n*)
Repeat
i←0
Repeat
BM_i ← Mesure (QM_i);
Réparer (BM_i);
Evaluer (BM_i);
B_i ← meilleure solution entre BM_i et B_i;
QM_i ←Interférence (QM_i, B_i);
i←i+1 ;
Until (i=*n*)
t←t+1 ;
Until (t=nombre maximal de générations)
Pbest← meilleure solution de tous les B_i
OUTPUT *Pbest*

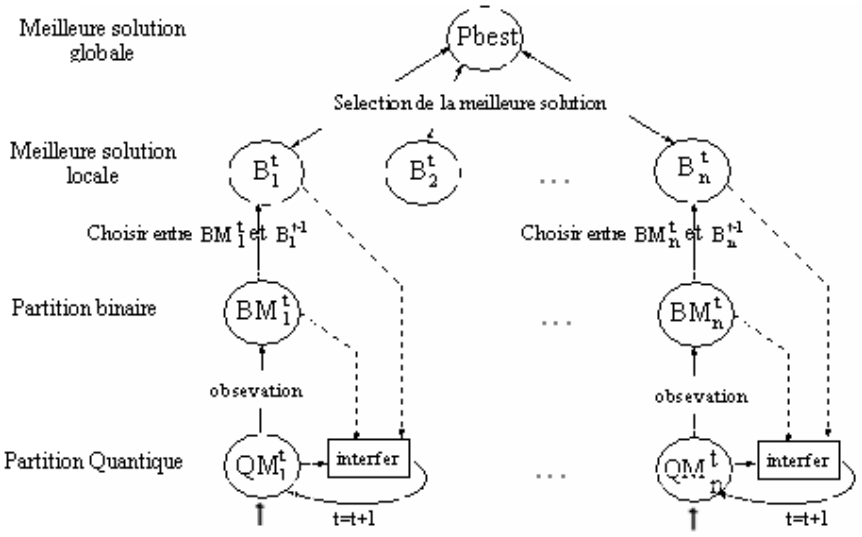


Figure3.4 : Structure de QEAC

4.4.3 Fonction objective

Afin d'évaluer la qualité d'une partition donnée, nous avons choisi une mesure de compacité souvent utilisée qui est la variance intra cluster. Elle consiste à calculer la somme de la distance carrée entre les points de données et leur centroïdes de clusters correspondants :

$$Var = \sum_{i=1}^{k} \sum_{p_j \in c_i} d(p_j - \mu_i)^2 \qquad (1)$$

où p_j dénote un point de données, k dénote le nombre de clusters, μ_i représente le centroïde de cluster c_i, $d(.,.)$ est la distance Euclidienne. L'objectif est donc de minimiser cette mesure.
Maintenant nous commençons à décrire chaque procédure mentionnée dans l'algorithme.

4.4.4 L'étape d'initialisation

L'initialisation est constituée essentiellement de deux étapes :

1. La génération d'une partition binaire initiale et valide.
2. La génération d'une partition quantique initiale à partir de la partition binaire trouvée à l'étape 1.

QEAC commence par sélectionner k points de données comme des centroïdes de clusters initiaux et assigner chaque point de données au centroïde le plus proche. Cela permet de générer une partition binaire possible. Cela est répété 10 fois pour générer 10 partitions, ensuite nous choisissons parmi ces 10 partitions la meilleure partition qui minimise la variance intra cluster. Durant ces étapes, chaque partition contenant des clusters vides est rejetée et elle est remplacée par une autre. Cela est répété autant de fois qu'on génère une partition contenant des clusters vides. Le but est d'assurer la génération d'une partition initiale valide.
Une fois que la partition binaire initiale est générée, nous procédons à la génération de la partition quantique initiale, nous devons définir une fonction qui calcule α_{ij} et β_{ij} de chaque qubit q_{ij}. Une fonction possible basée sur la distance entre les centroïdes des clusters et les points de données est définie comme suit :

$$a_{ij} = \cos(arccotg(d(p_j, m_i))) \qquad (2)$$

$$b_{ij} = \sin(arccotg(d(p_j, m_i))) \qquad (3)$$

L'interprétation géométrique de la fonction choisie est montrée dans figure 4.4 où la distance entre le point de donnée p_j et le centroïde μ_i du cluster c_i est supposée égale à la cotangente de l'angle θ_{ij} ayant α_{ij} sa projection sur l'axe des cosinus et β_{ij} sa projection sur l'axe des sinus.

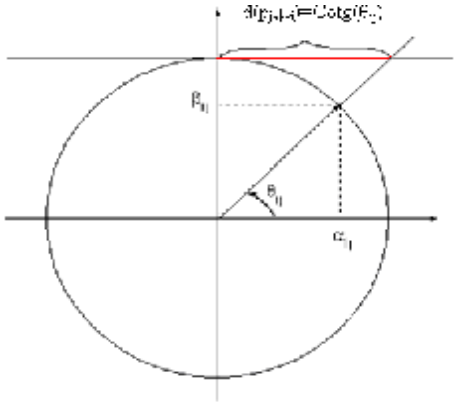


Figure 4.4 Interprétation géométrique de la fonction calculant α_{ij} β_{ij} initiaux

4.4.5 Mesure de la population quantique

La mesure est l'opération qui permet l'observation des états quantiques afin d'extraire une solution parmi toutes celles présentes dans la superposition sans détruire toutes les autres configurations. Le résultat de cette opération est une matrice binaire représentant une partition potentielle. Elle est générée par parcourir la partition quantique colonne par colonne et chercher la valeur maximale de $|\beta_{ij}|^2$de chaque colonne. Une fois que ce maximum est trouvé, nous mettons à 1 l'élément correspondant x_{ij} de la matrice binaire et à 0 le reste des éléments de la colonne j. La figure 4.5 donne un exemple d'observation de partition quantique qui résulte en une binaire.

Cette observation assure que chaque point de donnés peut être assigné à un seul cluster, mais elle n'assure pas la non apparition de clusters vides, pour cette raison nous avons introduit la procédure Réparer.

$$\begin{matrix} 0.1432 & 0.3711 & 0.6887 & 0.9087 \\ 0.9897 & 0.9286 & 0.7250 & 0.4175 \\ 0.0608 & 0.8489 & 0.3310 & 0.8510 \\ 0.9982 & 0.5286 & 0.9436 & 0.5252 \end{matrix} \rightarrow \begin{matrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \end{matrix}$$

Figure 4.5 Exemple de l'observation de *QM* avec 4 points de données et 2 clusters.

4.4.6 L'étape Réparer

Lorsque l'observation de la partition quantique QM^t à la génération t donne la partition binaire BM^t contenant un cluster vide, BM^t sera remplacée par une nouvelle partition binaire générée comme suit:

1. Remplacer chaque cluster vide de l'ensemble de centroïdes $\{\mu_i^t\}$ de QM^t par un point de donnée aléatoirement choisi.
2. Recalculer QM^t en utilisant (2) et (3) avec $d(p_j, \mu_i^t)$.

3. $BM^t \leftarrow$ Mesure (QM^t).

Ces étapes sont répétées autant de fois que la procédure Mesure génère une partition binaire comprenant des clusters vides.

4.4.7 L'étape d'interférence

Dans cette étape, la partition quantique est mise à jour en appliquant un opérateur unitaire quantique qui réalise une rotation avec un angle $\Delta\theta_{ij}$ conçu comme une fonction de α_{ij} , β_{ij} et la valeur binaire correspondante b_{ij} dans la meilleure partition (voir figure 4.6). Chaque élément q_{ij} de la partition quantique est mis à jour suivant ces étapes:

1. Déterminer $\Delta\theta_{ij}$ avec la table de consultation (voir tableau 4.1).
2. Calculer les nouvelles valeurs α'_{ij} , β'_{ij} en utilisant:

$$\begin{bmatrix} \alpha'_{ij} \\ \beta'_{ij} \end{bmatrix} = \begin{bmatrix} \cos\left(\Delta\theta_{ij} \times s(\alpha_{ij},\beta_{ij})\right) & -\sin\left(\Delta\theta_{ij} \times s(\alpha_{ij},\beta_{ij})\right) \\ \sin\left(\Delta\theta_{ij} \times s(\alpha_{ij},\beta_{ij})\right) & \cos\left(\Delta\theta_{ij} \times s(\alpha_{ij},\beta_{ij})\right) \end{bmatrix} \begin{bmatrix} \alpha_{ij} \\ \beta_{ij} \end{bmatrix} \quad (4)$$

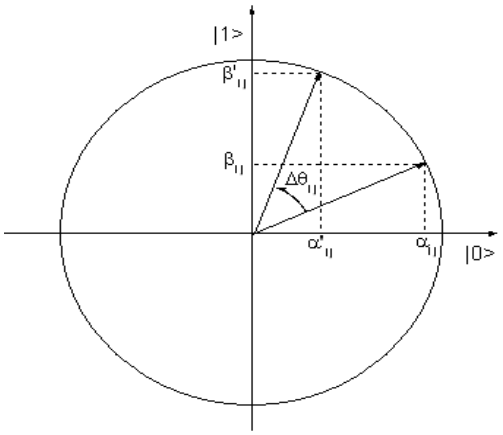


Figure 4.6 Interférence quantique

x_{ij}	b_{ij}	$f(BM) \leq f(B)$	$\Delta\theta_{ij}$	$s(\alpha_{ij},\beta_{ij})$	
0	0	faux	0.05	+1	-1
0	0	vrai	-0.01	+1	-1
0	1	faux	0.05	+1	-1
0	1	vrai	0.01	+1	-1
1	0	faux	-0.05	+1	-1
1	0	vrai	-0.01	+1	-1
1	1	faux	-0.05	+1	-1
1	1	vrai	0.01	+1	-1

Tableau 4.1. Table de consultation *de* $\Delta\theta_{ij}$,où $s(\alpha_{ij},\beta_{ij})$ est le signe *de* $\Delta\theta_{ij}$ et b_{ij} , x_{ij} sont les bits de la meilleure partition *B* et la partition binaire *BM respectivement* . f représente la variance intra cluster.

La valeur de l'angle de rotation $\Delta\theta_{ij}$ est choisie facilement par un raisonnement intuitif. Quand la condition ***f(BM)£f(B)*** est satisfaite $\Delta\theta_{ij}$ prend de très petite valeur pour mettre à jour la solution binaire ***BM*** autour de la meilleure solution ***B***. Cela permet la recherche de nouvelles solutions dans le voisinage de la meilleure solution, mais quand la condition n'est pas satisfaite $\Delta\theta_{ij}$ prend de petite valeurs, mais plus grandes que celles prises dans le cas précédent. Cela permet d'explorer d'autres solutions et pour échapper au minimum local.

4.5 Une deuxième approche évolutionnaire quantique pour le Clustering des données QEAC2

L'interaction entre les individus de la population du premier algorithme QEAC est inexistante. L'ajout d'une interaction entre les individus diversifie la population et réduit le nombre d'individus exigé et il contribue également à la minimisation de la fonction objective. Par conséquent, l'interaction améliore la performance de l'algorithme en offrant un gain en temps et en espace. Pour cette raison nous avons introduit des opérations de migration locale et globale. Nous avons constaté aussi que la fonction que nous avons développé à l'étape d'initialisation, et qui extrait les probabilités a_{ij} et b_{ij} à partir de la distance, contribue à la minimisation de la variance intra cluster, la chose qui nous a encouragé à exploiter cette fonction dans une opération appelée la régénération. D'autres modifications ont été faites comme le changement de paramètres de l'interférence et l'extension de l'ensemble des jeux de données sur lesquelles nous avons évaluer notre deuxième algorithme QEAC2. Dans ce qui suit nous présentons la nouvelle structure de algorithme QEAC2 ainsi que les modifications effectuées et les opération ajoutées.

4.5.1 Principe de l'algorithme QEAC2

Nous employons une population de n individus quantiques où son i^{eme} individu est dénoté par QM_i et une population de n individus binaires où son i^{eme} individu est dénoté par BM_i. La meilleure partition binaire trouvée par le i^{eme} individu est dénotée par B_i et la meilleure partition binaire globale trouvée à chaque période de générations est dénotée par ***Bglob***. La population binaire est divisée en ng groupes, dont chacun contient nd individus et la meilleure partition binaire locale trouvée par le j^{eme} groupe est dénotée par $Bgroup_j$. En commençant avec une population initiale, le processus consiste à évoluer cette population en appliquant quelques opérateurs quantiques de base : la mesure, l'interférence, la régénération et la migration globale et locale. La figure 4.7 inspirée de [Han et al,02] décrit la structure de l'approche proposée que nous pouvons décrire par l'algorithme suivant :

INPUT Jeu de données *S*
Begin
t←1
i←1
Repeat
Initialiser (QM_i);
i←i+1
Until (i>*n*)
Repeat
i←1
Repeat
$BM_i \leftarrow$ Mesure(QM_i);
Réparer (BM_i);
Evaluer (BM_i);
$B_i \leftarrow$ meilleure solution entre BM_i et B_i;
$QM_i \leftarrow$Interférence (QM_i, B_i);
i←i+1 ;
Until (i>*n*)
If (t mod *periode* = 0) /* migration globale */
Bglob← meilleure solution de tous les B_i;
Migrer *Bglob* à tous les B_i;
Else /* migration locale */
j=1;
Repeat
$Bgroup_j \leftarrow$meilleure solution de *nd* B_i du groupe j
Migrer $Bgroup_j$ à *nd* B_i du groupe j
j=j+1;
Until (*j*> *ng*)
End if
i=1;
Repeat
$QM_i \leftarrow$régénération (BM_i, *prob*);
i=i+1;
Until (*i*> *n*)
t←t+1 ;
Until (t = nombre maximal de générations)
OUTPUT *Bglob*

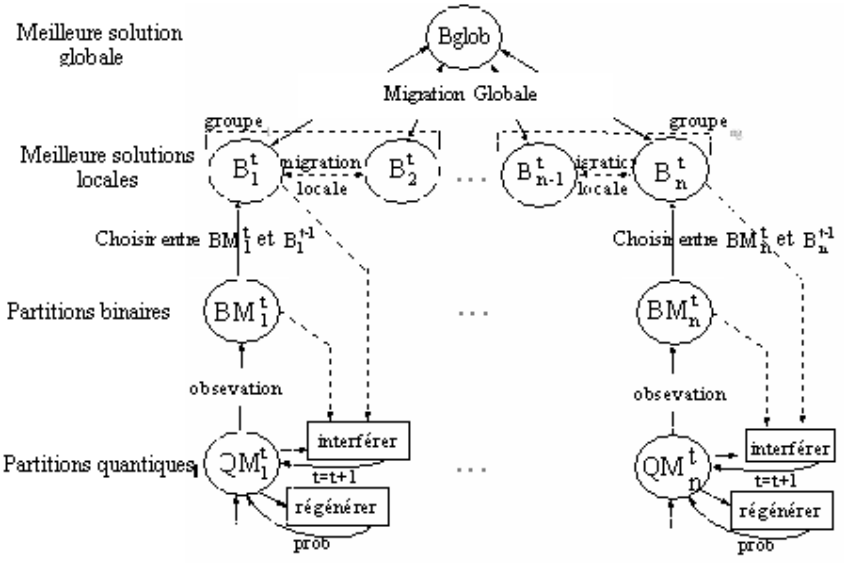


Figure 4.7 Structure de QEAC2

4.5.2 L'étape d'interférence

Le raisonnement sur lequel est basée l'interférence est maintenant changé de cette manière : Quand les deux bits x_{ij} et b_{ij} sont similaires, $\Delta\theta_{ij}$ essaye d'inverser x_{ij} d'une manière graduelle pour explorer d'autres solutions, mais quand ils sont dissimilaires et quand la condition $f(BM)£f(B)$ est satisfaite , $\Delta\theta_{ij}$ aide à explorer d'autres solutions loin de la meilleure solution et quand la condition n'est pas satisfaite $\Delta\theta_{ij}$ donne la possibilité de s'approcher de la meilleure solution en explorant son voisinage . Cela permet d'introduire des perturbations.

x_{ij}	b_{ij}	$f(BM)\leq f(B)$	$\Delta\theta_{ij}$	$s(\alpha_{ij},\beta_{ij})$	
0	0	Faux	0.0045	+1	-1
0	0	vrai	0.0045	+1	-1
0	1	faux	0.025	+1	-1
0	1	vrai	-0.010	+1	-1
1	0	faux	-0.025	+1	-1
1	0	vrai	0.010	+1	-1
1	1	faux	-0.0045	+1	-1
1	1	vrai	-0.0045	+1	-1

Tableau 4.2 Table de consultation *de* $\Delta\theta_{ij}$,où $s(\alpha_{ij},\beta_{ij})$ est le signe *de* $\Delta\theta_{ij}$,et b_{ij} , x_{ij} sont les bits de la meilleure partition *B* et la partition binaire *BM respectivement*. *f* représente la variance intra cluster.

4.5.3 Etape de régénération

Les paramètres choisis pour l'interférence pour explorer des solutions loin de la meilleure solution introduisent des perturbations sur α_{ij} et β_{ij} de la solution quantique et puisque ces probabilités reflètent la distance entre les point de données et les centroïdes des clusters, l'étape de régénération est introduite pour garder cette relation. Elle consiste à recalculer QM^t

à la génération t en utilisant (2) et (3) avec $d(p_j, \mu_i^t)$, où $\{\mu_i^t\}$ est l'ensemble des centroïdes des clusters de la partition binaire BM^t. la régénération est faite avec une probabilité ***prob.***

4.5.5 Migration global et locale

La migration est définie comme le processus de copier la meilleure solution globale ou locale dans toutes les B_i ou des groupes de B_i respectivement. La migration globale est accomplie en sélectionnant la meilleure solution *Bglob* de toutes les solutions B_i et les remplacer par *Bglob,* cela est fait périodiquement. La migration locale est effectuée par le remplacement des solutions B_i de chaque groupe par la meilleure d'entre elles *Bgroup*. L'opération de migration induit une variation des probabilités des solutions quantiques.

4.6 Résultats expérimentaux

Pour évaluer les performances de QEAC et QEAC2, des mesures d'évaluations et des jeux de données ont été utilisés.

4.6.1 Evaluation

Parmi les mesures d'évaluations présentées dans le chapitre2, nous avons sélectionné deux mesures de différents types. La première évaluation est faite avec la mesure externe F-mesure. C'est une fonction utilisée souvent dans la littérature de clustering. Elle compare la qualité de clustering aux classes correctes connues pour un jeu de données. F-mesure prend des valeurs dans l'intervalle [0,1] et devrait être maximale. La deuxième évaluation est faite avec une mesure interne qui est la variance intra cluster. Elle est souvent utilisée pour évaluer la performance des algorithmes de clustering. Cette deuxième évaluation montre à quel point nos algorithmes ont réussit à minimiser la variance intra clusters.

4.6.2 Jeux de données

Nous avons testé l'algorithme proposé QEAC sur plusieurs jeux de données, incluant des jeux de données synthétiques que nous avons généré et des jeux de données du monde réel. Pour testé QEAC2, nous avons enrichit l'ensemble des jeux de données par d'autres construit avec un générateur gaussien de clusters. Cet ensemble de jeux de données est divisé selon leurs sources en trois groupes:

A. Données synthétiques 2D

Nous avons généré deux jeux de données bidimensionnels, Dataset1 et Dataset2 en utilisant des distributions normales $N(\vec{m}, \vec{s})$. Le nombre de clusters, les tailles des différents clusters,

le vecteur de la moyenne $\vec{m}$ et le vecteur de l'écart type $\vec{s}$ sont fixés manuellement.

Dataset1 présente des clusters de forme sphérique tandis que Dataset2 présente des clusters de forme allongée (voir figure 4.8). Les distributions normales permettant de les générer sont décrites dans le tableau 4.3 . Toutes les informations sur ces deux jeux de données sont présentées dans le tableau 4.4.

Jeux de donnée	Source
Dataset1	$N([10,0],[2,2]), N([0,10],[2,2]), N([10,10],[2,2]), N([10,10],[2,2])$
Dataset2	$N([0,1],[1,0.1]), N([0,0],[1,0.1])$

Tableau 4.3 Sources des jeux de données Dataset1 et Dataset1

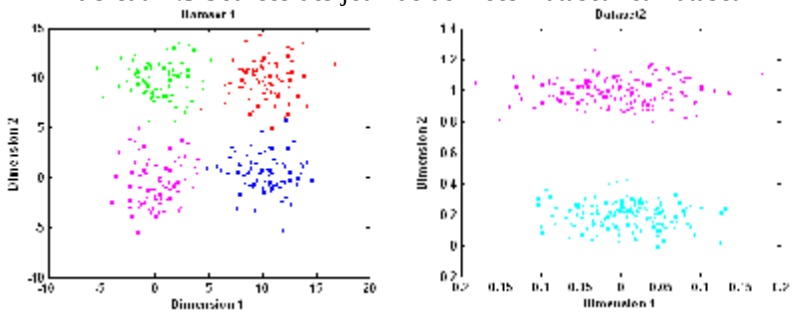


Figure 4.8 Distribution des jeux de données synthétiques 2D.

B. Données synthétiques xdyc

Les jeux de données dénotés xdyc, où x indique la dimensionnalité des données et y donne le nombre de clusters, sont extraits des jeux de données disponibles dans [Handl et al, 05b]. Ils sont obtenus avec un générateur gaussien de clusters développé par Handl et Knowles [Handl et al, 05a]. Ce générateur génère des clusters de forme sphérique et allongée. Afin d'introduire une variété sur les jeux de données synthétiques sur lesquels nous testons nos algorithmes, nous avons introduit des perturbations sur les jeux de données 2d10c, 10d4c, 10d10c en supprimant quelques points d'une manière aléatoire. Cela a les avantages de produire des clusters ayant des densités différentes d'une manière significative et d'attribuer une forme arbitraire à quelques clusters et perturber la forme sphérique ou allongée pour d'autres clusters et encore produire des clusters de différentes tailles. Ces avantages sont visualisés dans les figures 4.9, 4.10 et 4.11. Plus précisément, la distribution du jeu de données 2d10c schématisée dans la figure 4.9 présente ces avantages d'une manière très claire. Ils sont aussi très clairs dans le jeu de données 10d10c. Les figures 4.10 et 4.11 présentent la distribution des jeux de données 10d10c et 10d4c respectivement. Cela est fait au moyen de la projection des points de données de chaque jeu de données selon la dimension n°1 et les dimensions de 2 jusqu'à 9. Ce qui donne 8 projections parmi 36 projections possibles. Toutes ces projections pour chaque jeu de données sont représentées dans l'annexe C.

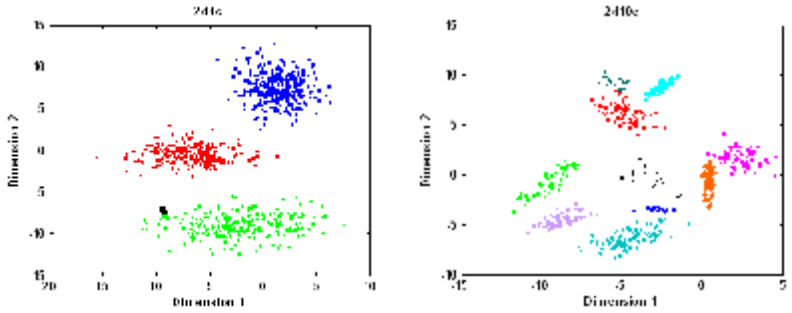


Figure 4.9 Jeux de données synthétiques 2dyc.

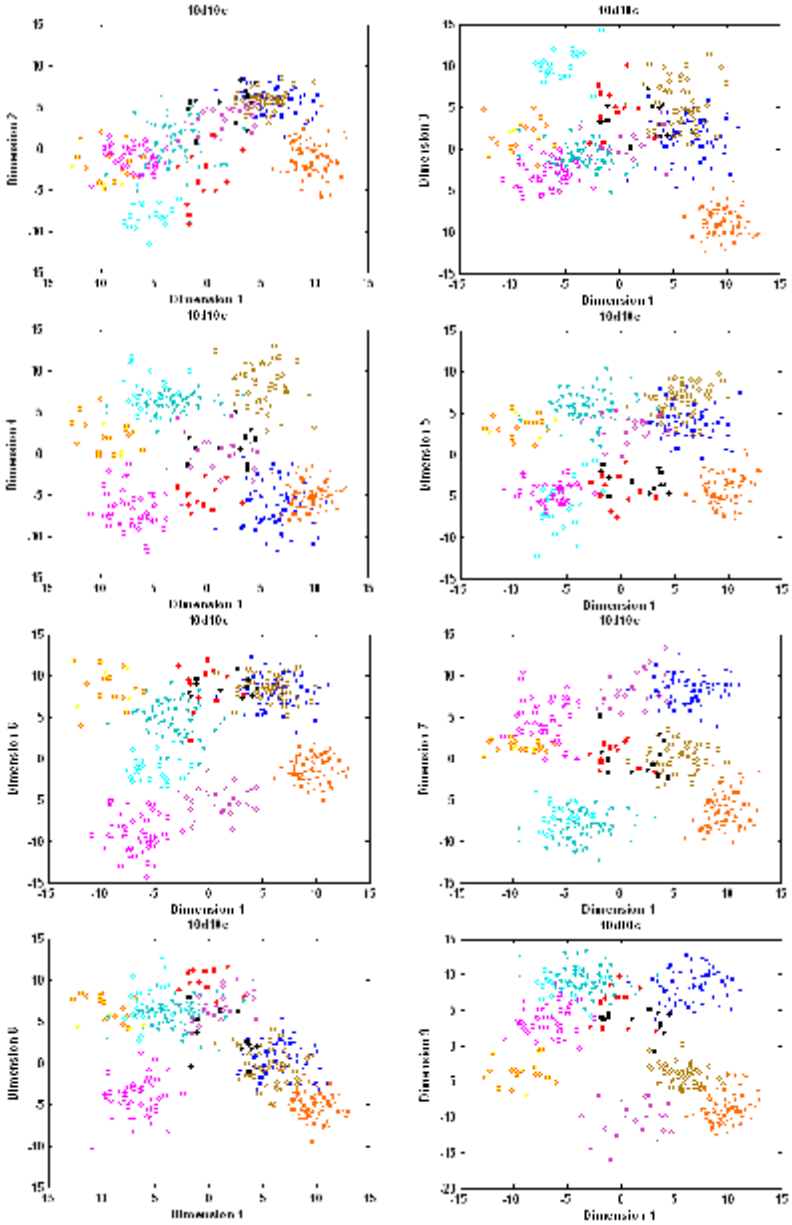


Figure 4.10 Représentation de la distribution du Jeu de données 10d10c par le biais des projections des points selon la dimension 1 et les dimensions de 2 jusqu'à 9.

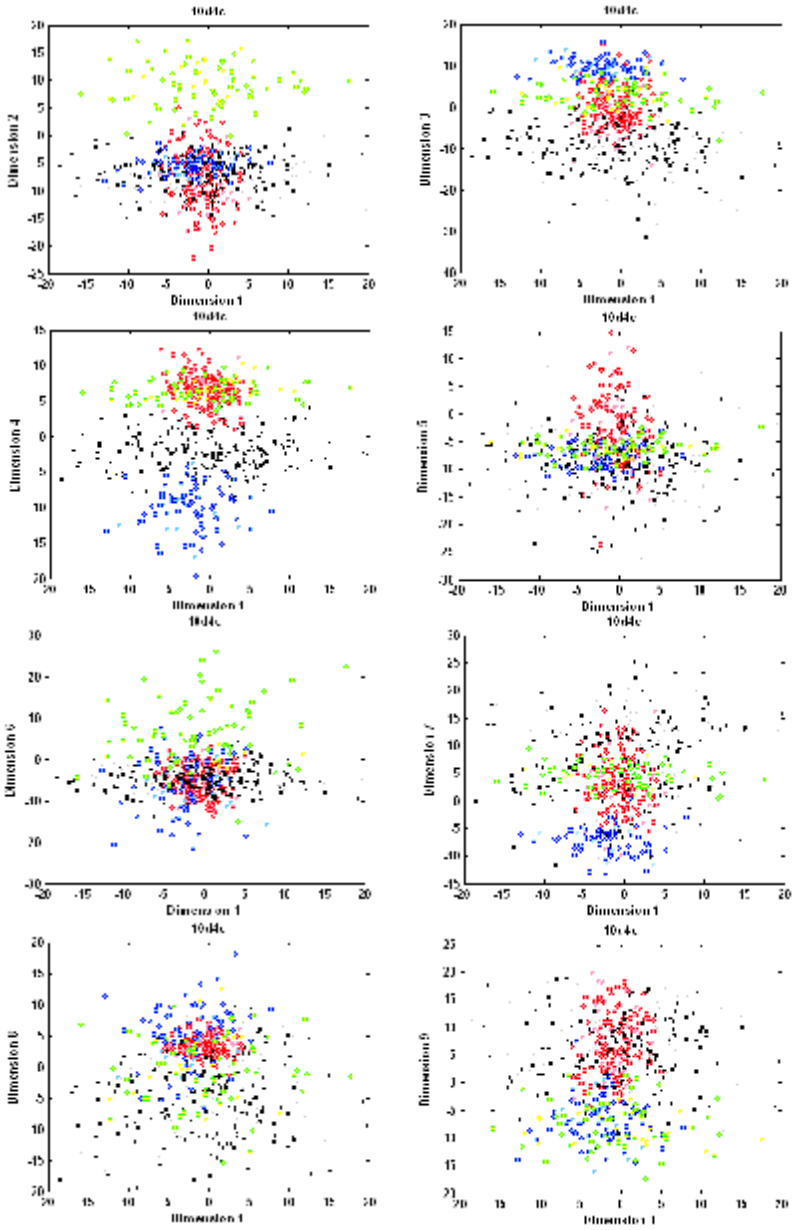


Figure 4.11 Représentation de la distribution du Jeu de données 10d4c par le biais des projections des points selon la dimension 1 et les dimensions de 2 jusqu'à 9.

C. Données réelles

Des jeux de données réels issus de l'UCI Machine Learning Repository [Blake et al,98] sont utilisés. Ils sont récapitulés dans le tableau 4.4. Ces données ne sont pas nécessairement conçues pour des méthodes unsupervisées (clustering), et sont souvent employées comme des benchmarks pour des techniques supervisées. Par conséquent, la performance sur ces données est parfois basse pour tous les algorithmes, car la structure de cluster ne s'accorde pas nécessairement avec les étiquettes de classes. Néanmoins, nous incluons ces données parce

qu'ils exposent une grande diversité dans la dimension, le nombre, la forme et la densité de clusters etc., et nous permettent d'être confiants à n'importe quelles conclusions générales que nous pourrions tirer des données synthétiques.

Une description détaillée de chaque jeu de données réel utilisé est disponible dans l'annexe A. Certains jeux de données comportent des données manquantes, dans ce cas nous avons adopté la stratégie qui est souvent utilisée, elle consiste à éloigner les points de données contenant les valeurs manquantes. Ce choix est justifié par le pourcentage faible des données manquantes dans les jeux de données utilisés.

	Nom	*k*	*m*	n_i	*Type*	*Dim*
jeux de données synthétiques	Dataset1	4	300	75,75,75,75	réel	2
	Dataset2	2	300	150,150	réel	2
	2d4c	4	876	219,244,312,101	réel	2
	2d10c	10	520	67,15,19,53,83,64,65,68,68,18	réel	2
	10d4c	4	504	151,78,201,74	réel	10
	10d10c	10	436	18,83,57,26,67,50,12,72,39,12	réel	10
jeux de données réels	Iris	3	150	50,50,50	réel	4
	Dermatology	6	366	112,61,72,49,52,20	entier	33
	Soybean	4	47	10,10,10,17	entier	35
	Wisconsin	2	699	458,241	entier	9
	Thyroid	3	215	150,35,30	réel	5
	Zoo	7	101	41,20,5,13,4,8,10	boolien	16

Tableau 4.4 Résumé des jeux de données employés, où *m* est le nombre total de points de données dans le jeu de données, n_i est le nombre de points de données appartenant au cluster *i*, *dim* donne la dimensionnalité et *k* le nombre de clusters

4.6.3 Paramètres utilisés

Les valeurs affectées aux paramètres de l'algorithme QEAC sont représentées dans le tableau 4.5, et ceux de l'algorithme QEAC2 sont représentés dans le tableau 4.6. L'évaluation de QEAC est faites à travers 1000 exécutions, et celle de QEAC2 est faites à travers 500 exécutions. Plusieurs versions de chaque algorithme ont été testées afin d'analyser le rôle de chaque opération incorporée dans l'algorithme. Pour QEAC, les deux versions QEAC(100) et QEAC(1) utilisent 100 et 1 individus respectivement. Pour QEAC2, les trois versions QEAC2(6) et QEAC2(1) et QEAC2_itrf(1) utilisent 6, 1 et 1 individus respectivement. QEAC2_itrf(1) montre l'effet de l'interférence toute seule, QEAC2(1) montre l'effet de l'interférence et la régénération ensemble. La comparaison entre QEAC2(1) et QEAC2_itrf(1) montre l'apport de la régénération .QEAC2(6) est la version complète et la comparaison entre QEAC2(6) et QEAC2(1) élucide le rôle de la migration globale et locale. La comparaison entre QEAC(1) et QEAC2_itrf(1) reflète l'effet des deux versions de l'interférence.

Paramètre	Valeur	
	QEAC(100)	QEAC(1)
Nombre de générations	100	100
Taille de la population	100	1

Tableau 4.5 Paramètres de QEAC

Paramètre	Valeur		
	QEAC2(6)	QEAC2(1)	QEAC2_itrf(1)
Nombre de générations	1000	1000	1000
Taille de la population	6	1	1
Nombre de groupes (*ng*)	2	0	0
Taille d'un groupe (*nd*)	3	0	0
Période de migration globale (*période*)	20	0	0
Probabilité de régénération (*prob*)	0.25	0.25	0

Tableau 4.6 Paramètres de QEAC2

4.6.4 Résultas

Les résultats sont évalués en utilisant la F-mesure et la variance intra cluster. La comparaison est faite avec l'algorithme le plus connu Kmeans que nous avons choisi pour la raison qu'il optimise implicitement la variance intra cluster et que notre approche l'optimise explicitement. Kmeans est décrit en détail dans le chapitre 2.

La représentation des résultats est accomplie en utilisant l'outil statistique "Boxplot", c'est un des outils fournit par MATLAB. Cette représentation a un double intérêt : La simplification de la visualisation des résultats obtenus pour chaque jeu de données et la représentation de plus de détail descriptif des résultats d'une manière claire, simple et réduite. La figure 4.12 montre l'essentiel des informations données par un Boxplot. Une description détaillée est disponible dans [Weisstein,99].

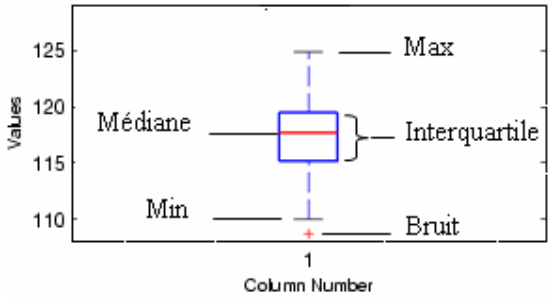
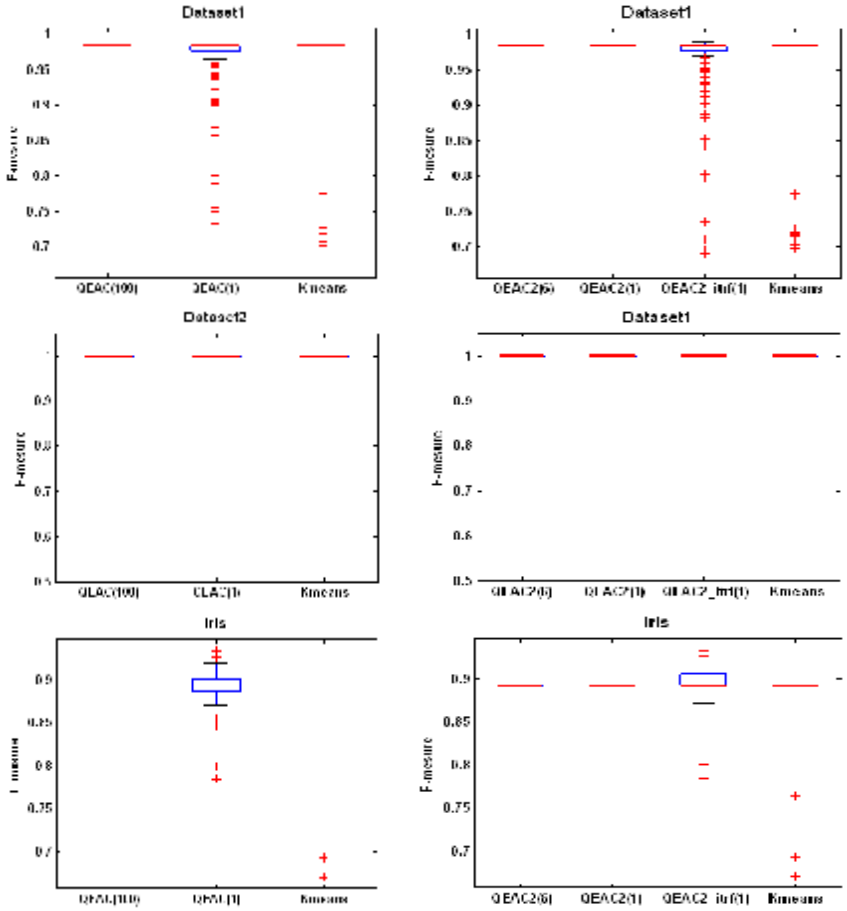


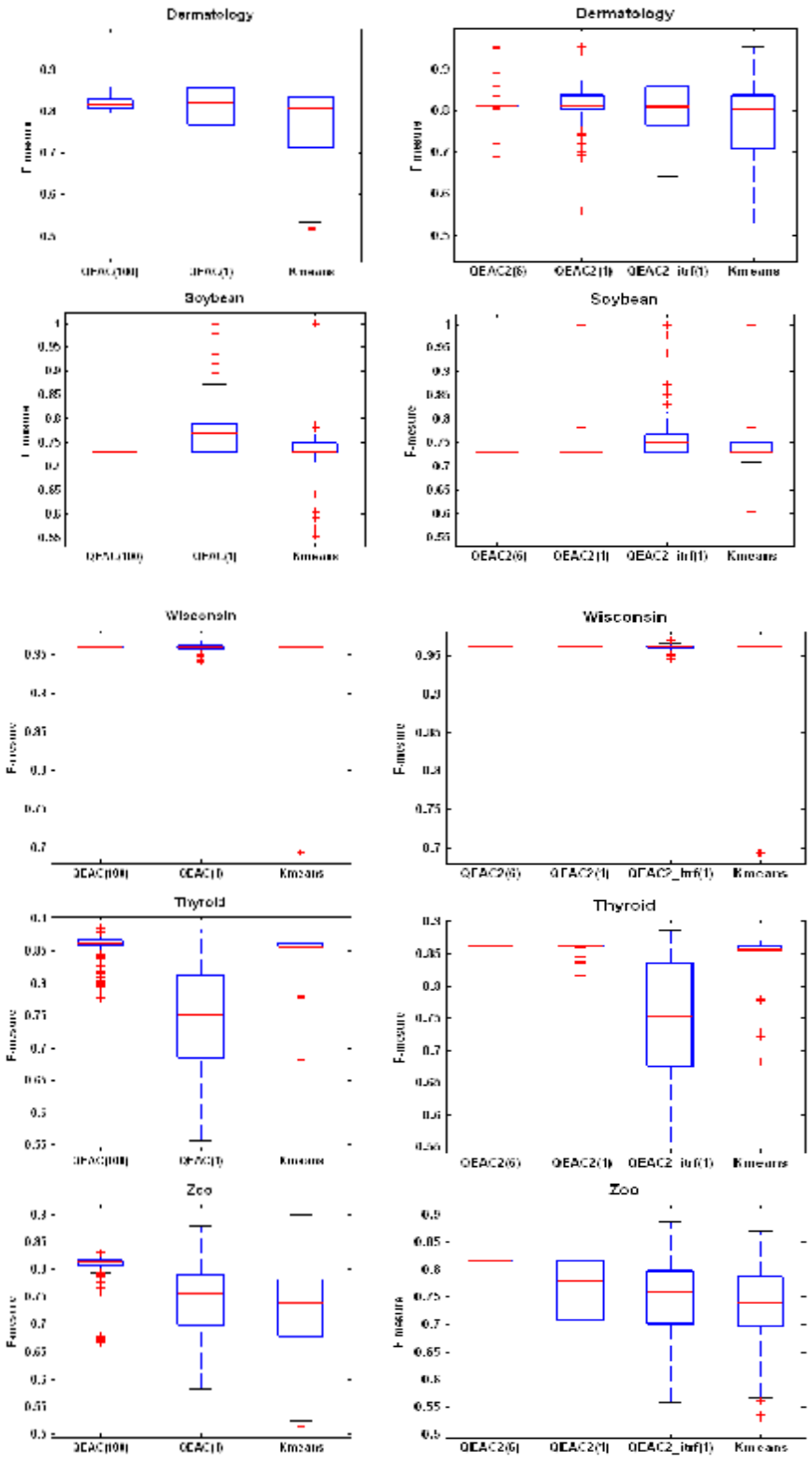
Figure 4.12 Informations données par un Boxplot.

Le fond et le sommet et de la boîte en bleu sont les 25ème et 75ème centiles de l'échantillon. La distance entre le sommet et le fond de la boîte est l'interquartile.

La ligne en rouge au milieu de la boîte est la médiane d'échantillon. Les lignes s'étendant au-dessus et au-dessous de la boîte montrent l'ampleur du reste de l'échantillon (à moins qu'il y ait des bruits). Supposant qu'il n'y a pas de bruits, le maximum de l'échantillon est la ligne supérieure. Le minimum de l'échantillon est la ligne inférieure. Par défaut, un bruit est une valeur qui est plus de 1.5 fois la valeur d'interquartile loin du sommet ou du fond de la boîte. Un bruit est représenté pas un plus "+".

Les boxplots des résultats évalués avec la F-mesure sont schématisés dans la figure 4.13. La figure 4.14 montre la deuxième évaluation basée sur la variance intra cluster. Toutes les valeurs des médianes, des interquartiles, des maximums et des minimums de la F-mesure et la variance intra cluster représentées par les boxplots ont été tabulées. Elles peuvent être trouvées dans l'annexe B.




Dermatology
Dermatology
Soybean
Soybean
Wisconsin
Wisconsin
Thyroid
Thyroid
Zoo
Zoo
F-measure
QEAC2(6)
QEAC2(1)
QEAC2_inf(1)
Kmeans

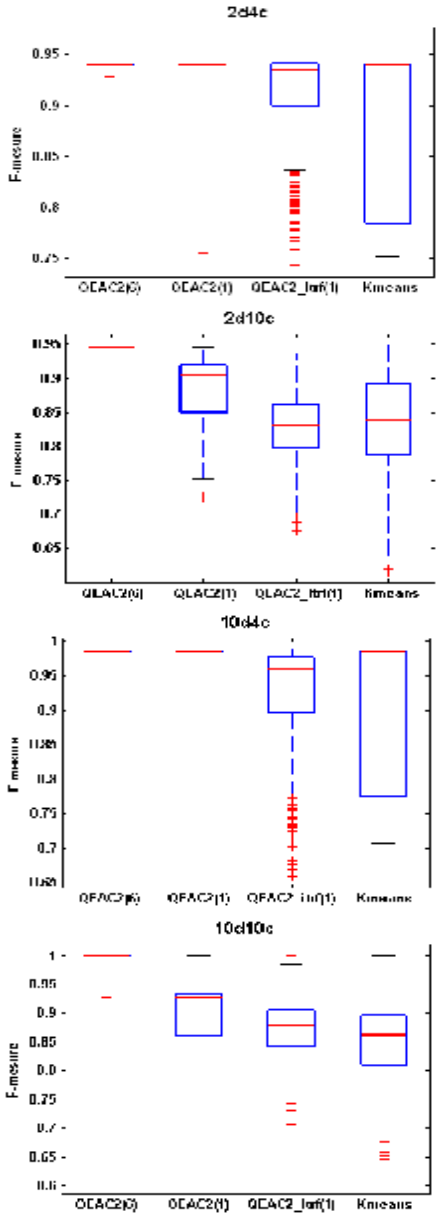
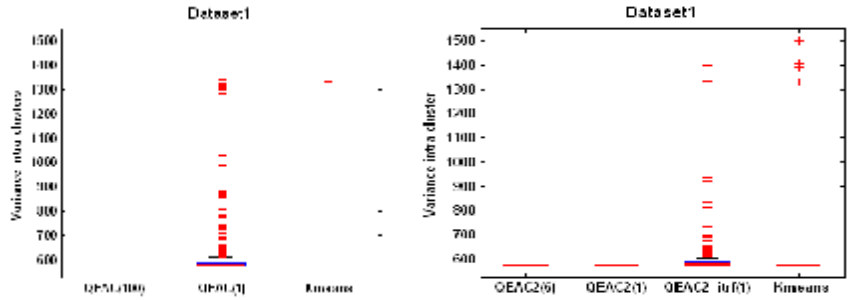
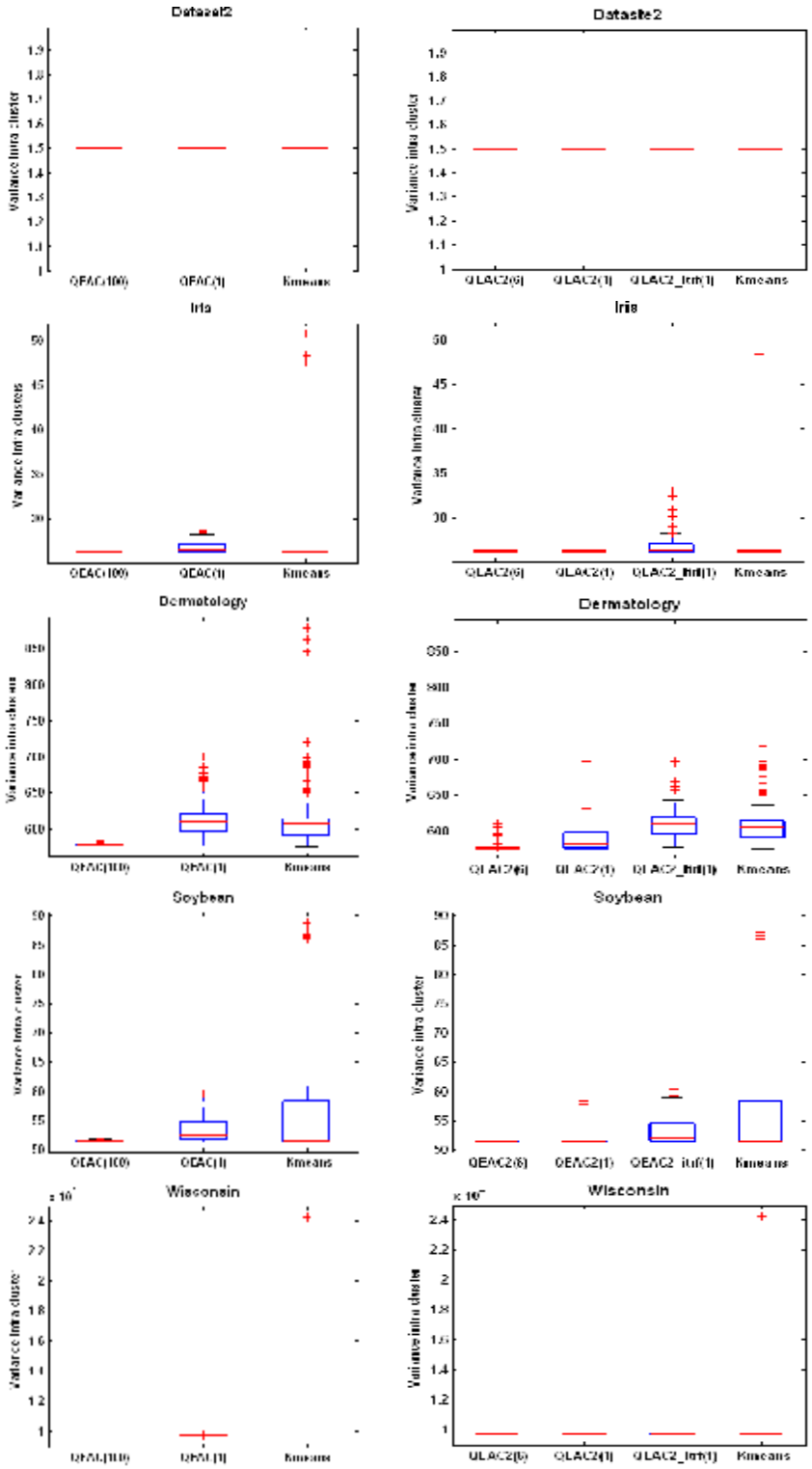
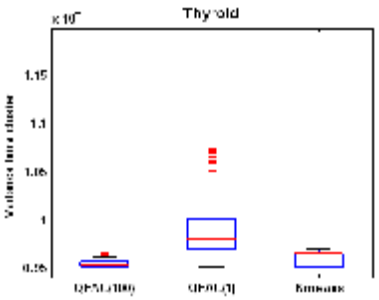
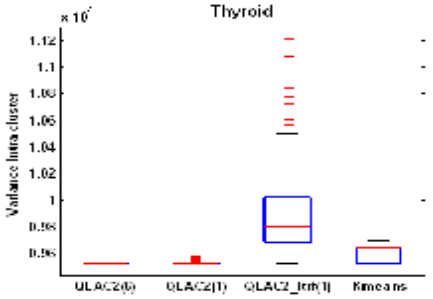


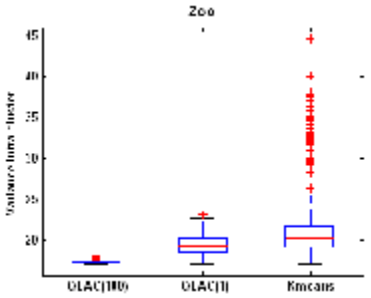
Figure 4.13 Les boxplots des résultats évalués avec la F-mesure

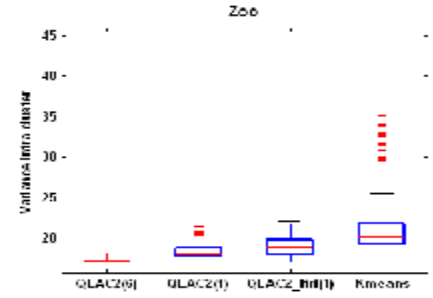


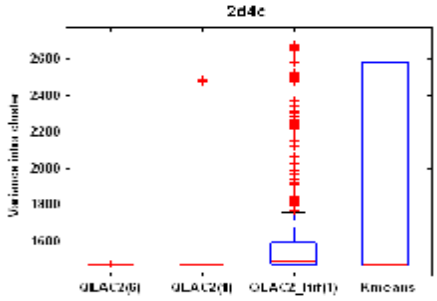

Dataset2
Variance intra cluster
QEAC(100)
QEAC(1)
Kmeans
Dataset2
Variance intra cluster
QLAC2(6)
QLAC2(1)
QLAC2_Init(1)
Kmeans
Iris
Variance intra clusters
QEAC(100)
QEAC(1)
Kmeans
Iris
Variance intra cluster
QLAC2(6)
QLAC2(1)
QLAC2_Init(1)
Kmeans
Dermatology
QEAC(100)
QEAC(1)
Kmeans
Dermatology
QLAC2(6)
QLAC2(1)
QLAC2_Init(1)
Kmeans
Soybean
QEAC(100)
QEAC(1)
Kmeans
Soybean
QEAC2(6)
QEAC2(1)
QEAC2_Init(1)
Kmeans
Wisconsin
QEAC(100)
QEAC(1)
Kmeans
Wisconsin
QLAC2(6)
QLAC2(1)
QLAC2_Init(1)
Kmeans

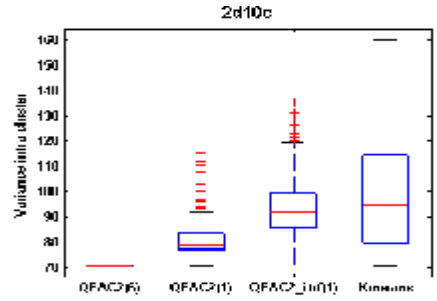

Thyroid
Kmeans

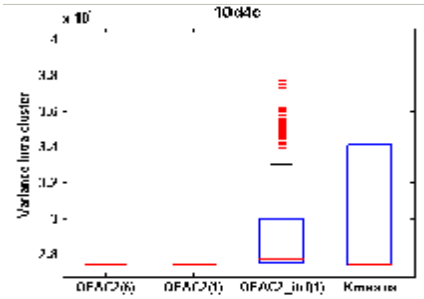

Thyroid
Variance intra cluster
Kmeans


Zoo
Kmeans


Zoo
Variance intra cluster
Kmeans


2d4c
Variance intra cluster
Kmeans


2d10c
Variance intra cluster
Kmeans


10d4c
Variance intra cluster
Kmeans

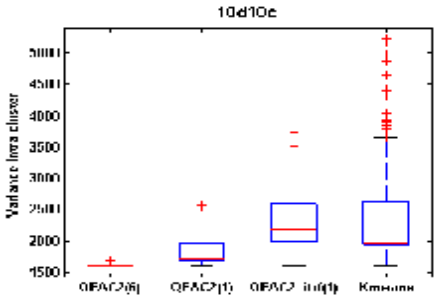


Figure 4.14 Les boxplots des résultats évalués avec la variance intra clusters

A. Résultas de l'algorithme QEAC

Le tableau 4.7 montre les valeurs de la médiane et de l'interquartile de la F-mesure obtenues à travers 1000 exécutions de chaque algorithme, l'interquartile indique l'intervalle de valeurs auxquelles les algorithmes convergent. Le tableau 4.8 montre les valeurs de la médiane et de l'interquartile de la variance intra cluster.

D'après la figure 4.13 et la figure 4.14, les résultats montrent que QEAC(100) est significativement meilleur que QEAC(1) et Kmeans pour tous les jeux de données en terme de F-mesure et variance intra cluster.

En terme de F-mesure et pour Dermatology, Soybean et Zoo, QEAC(1) est meilleur que Kmeans d'une manière remarquable, tandis que pour Dataset1, iris et Wisconsin, QEAC(1) et Kmeans ont la même médiane. Pour ces derniers jeux de données ainsi que Soybean, l'intervalle d'interquartile sur lequel s'étalent les solutions trouvées par QEAC(1) est un peu large mais ne contient pas de mauvaises solutions, cela peut être vu comme un avantage, il peut être exploité pour étendre l'algorithme au clustering multiobjectif. Dans ce cas, cette variété de solutions trouvées contribuent à la diversification des solutions du front de Pareto , ce qui implique l'amélioration de la qualité de clustering multiobjectif basée sur l'optimisation multiobjective.

En terme de variance intra cluster, QEAC(1) est meilleur que Kmeans dans le cas de Zoo et Dataset1. Pour Iris, Dermatology, Soybean et Wisconsin, la variance de Kmeans est légèrement meilleur que QEAC(1).

Kmeans fournit quelques très mauvaises solutions qui sont considérés comme des bruits (voir figure 4.13, 4.14), la chose qui n'existe pas dans le cas de QEAC.

Pour Thyroid, QEAC(1) n'est pas bon.

Pour Dataset2, tous les algorithmes réussissent à trouver l'optimum global.

De ces résultats, plusieurs points essentiels sont tirés, ils ont conduit au développement du deuxième algorithme QEAC2.

- QEAC(1) est basé principalement sur l'interférence. Les résultats de QEAC(1) montre que les valeurs choisit pour les paramètres de l'interférence ne sont pas les meilleurs et elles peuvent être améliorées pour se bénéficier davantage de l'interférence. Ce point est exploité au sein de l'algorithme QEAC2.
- L'interaction entre les 100 individus de QEAC(100) est inexistante, mais malgré ça QEAC(100) donne de bon résultats. Cela implique que le renforcement de l'interaction avec d'autres opérateurs quantiques améliore l'algorithme tout en réduisant le nombre des individus. Ce renforcement est incarné par l'ajout des opérations de migration globale et locale ainsi que l'opération la régénération.

Jeux de données	QEAC(100)	QEAC(1)	Kmeans
Dataset1	0.9833(0.0000)	0.9833(0.0099)	0.9833(0.0000)
Dataset2	1.0000(0.0000)	1.0000(0.0000)	1.0000(0.0000)
Iris	0.8918(0.0000)	0.8918(0.0142)	0.8918(0.0000)
Dermatology	0.8174(0.0210)	0.8164(0.0882)	0.8036(0.1261)
Soybean	0.7281(0.0000)	0.7483(0.0600)	0.7281(0.0210)
Wisconsin	0.9603(0.0000)	0.9603(0.0031)	0.9603(0.0000)
Thyroid	0.8620(0.0040)	0.7561(0.1186)	0.8549(0.0071)
Zoo	0.8144(0.0089)	0.7620(0.0859)	0.7318(0.1000)

Tableau 4.7 Médiane et interquartile de F-mesure obtenus pour QEAC(100), QEAC(1) et Kmeans

Jeux de données	QEAC(100)	QEAC(1)	Kmeans
Dataset1	574.7736(0.000)	576.1542 (14.8064)	574.7736(0.000)
Dataset2	1.4965 (0.000)	1.4965 (0.000)	1.4965 (0.000)
Iris	26.3136(0.000)	26.4622(0.8074)	26.3136(0.000)
Dermatology	578.3475(1.0723)	609.5857(24.1979)	606.2421(22.8621)
Soybean	51.4909(0.1305)	52.4044(2.8384)	51.4909(6.6947)
Wisconsin	9661.5869 (0.000)	9663.1513 (8.2215)	9661.5869 (0.000)
Thyroid	9532.8(40.2614)	9786.8(309.7084)	9646.4 (126.3484)
Zoo	17.1911(0.1531)	19.2886 (1.7007)	20.1962 (2.6473)

Tableau 4.8 Médiane et interquartile de la varaince intra cluster obtenus pour QEAC(100), QEAC(1) et Kmeans

B. Résultas de l'algorithme QEAC2

Le tableau 4.9 montre les valeurs de la médiane et de l'interquartile de F-mesure obtenues à travers 500 exécutions de chaque algorithme. Le tableau 4.10 montre les valeurs de la médiane et de l'interquartile de la variance intra cluster.

Selon les figures 4.13 et 4.14, les résultats montrent que QEAC2(6) et QEAC2(1) sont significativement meilleurs que Kmeans pour tous les jeux de données en terme de F-mesure et variance intra cluster.

Pour 2d10c, 10d10c, Zoo et Dermatology, la comparaison entre QEAC2(1) et QEAC2(6) montre que ce dernier améliore la F-mesure et la variance intra clusters d'une manière remarquable. QEAC2(6) et QEAC2(1) donnent la même valeur de la médiane pour Dataset2, 2d4c, 10d4c, Iris, Soybean, Wisconsin, Thyroid mais QEAC2(6) réduit la génération de solutions considérées comme des bruits. On peut dire que QEAC2(6) réussit à optimiser la variance intra cluster mieux que QEAC2(1). La comparaison entre QEAC2(6) et QEAC2(1) expose l'effet de la migration globale et locale.

QEAC2_itrf(1) reflète l'effet de l'interférence. QEAC2_itrf(1) donne un ensemble de solutions diversifié. Cette diversité représentée par l'interquartile témoigne la nature de l'interférence qui tend à être destructive (voir figure 4.13 et 4.14).Cette destructivité joue un rôle très important avec la régénération. Les deux jouent un rôle complémentaire, l'interférence introduit des perturbations et la régénération agit dans le sens inverse, elle minimise la variance intra cluster.

QEAC2(1) donne l'effet de l'interférence et la régénération ensemble. La comparaison entre QEAC2(1) et QEAC2_itrf(1) jette la lumière sur la complémentarité entre l'interférence et régénération.

L'ajout des jeux de données xdyc a pour but de montrer des propriétés intéressantes de QEAC2. Si nous retournons aux figures 4.9, 4.10 et 4.11 qui représentent les distributions des jeux de données 2d4c, 2d10c, 10d4c et 10d10c, nous pouvons remarquer une différence de densité de clusters, une différence de taille de clusters, quelques clusters de formes arbitraires et des formes perturbées de clusters sphériques et allongées. Ces propriétés sont très claires dans les jeux de données 2d10c et 10d10c. Il sont moins clairs dans les jeux de données 2d4c et 10d4c. Nous retournons maintenant aux résultat (tableau 4.9, 4.10), pour 2d10c et 10d10c, QEAC2(1) améliore les résultat de Kmeans et QEAC2(6) améliore les résultat de QEAC2(1) d'une manière significative. Pour 10d10c, QEAC2(6) réussit à trouver l'optimum globale.

Pour 2d4c et 10d4c, QEAC2(6) et QEAC2(1) améliore les résultats de Kmeans mais d'une manière moins significative que les jeux de données précédents.

Nous pouvons dire à partir de ces jeux de données synthétiques et même des jeux de données réels qui présentent une grande variété de propriétés de clusters que notre algorithme QEAC2 peut détecter des clusters de différentes densités, de différentes tailles et de différentes formes.

D'un autre coté, les valeurs d'interquartile du QEAC2(6) sont toutes égales à zéro pour tous les jeux de données(voir tableau 4.9 et 4.10). Cela peut refléter la non dépendance entre partition finale et la partition initiale.

Jeux de données	QEAC2(6)	QEAC2(1)	QEAC2_itrf(1)	Kmeans
Dataset1	0.9833(0.0000)	0.9833(0.0000)	0.9833(0.0066)	0.9833(0.0000)
Dataset2	1.0000(0.0000)	1.0000(0.0000)	1.0000(0.0000)	1.0000(0.0000)
2d4c	0.9399(0.0000)	0.9399(0.0000)	0.9345(0.0420)	0.9399(0.1558)
2d10c	0.9451(0.0000)	0.9054(0.0692)	0.8321(0.0634)	0.8355(0.1046)
10d4c	0.9841(0.0000)	0.9841(0.0000)	0.9602(0.0812)	0.9841(0.2056)
10d10c	1.0000(0.0000)	0.9270(0.0715)	0.7532(0.0618)	0.8629(0.1005)
Iris	0.8918(0.0000)	0.8918(0.0000)	0.8918(0.0135)	0.8918(0.0000)
Dermatology	0.8107(0.0000)	0.8107(0.0319)	0.8086(0.0928)	0.8036(0.1261)
Soybean	0.7281(0.0000)	0.7281(0.0000)	0.7483(0.0399)	0.7281(0.0210)
Wisconsin	0.9603(0.0000)	0.9603(0.0000)	0.9603(0.0031)	0.9603(0.0000)
Thyroid	0.8620(0.0000)	0.8620(0.0000)	0.7521(0.1612)	0.8549(0.0071)
Zoo	0.8164(0.0000)	0.7775(0.1088)	0.7583(0.0963)	0.7318(0.1000)

Tableau 4.9. Médiane et interquartile de F-mesure obtenues pour QEAC2(6), QEAC2(1) , QEAC2_itrf(1) et Kmeans

Jeux de Données	QEAC2(6)	QEAC2(1)	QEAC2_itrf(1)	Kmeans
Dataset1	574.7736 (0.000)	574.7736 (0.000)	575.7629 (12.3922)	574.7736 (0.000)
Dataset2	1.4965 (0.000)	1.4965 (0.000)	1.4965 (0.0000)	1.4965 (0.000)
2d4c	1467.0372 (0.000)	1467.0372 (0.000)	1483.4882 (118.1172)	1467.0372 (1115.5)
2d10c	70.8330 (0.000)	78.9347 (6.3199)	91.5807 (13.6648)	94.6520 (34.7470)
10d4c	27403.6099 (0.000)	27403.6099 (0.000)	27767.07355 (2545.6)	27403.6099 (6733.2)
10d10c	1610.0554 (0.000)	1702.7707 (251.5430)	2167.3988 (588.7767)	1960.6921 (688.1683)
Iris	26.3136 (0.000)	26.3136 (0.000)	26.4112 (0.8010)	26.3136 (0.000)
Dermatology	576.6163 (0.4200)	582.7106 (20.6060)	609.9999 (22.1420)	606.4401 (22.8304)
Soybean	51.4909 (0.000)	51.4909 (0.000)	52.0386 (3.0253)	51.4909 (6.6947)
Wisconsin	9661.5869 (0.000)	9661.5869 (0.000)	9663.1513 (7.1688)	9661.5869 (0.000)
Thyroid	9520.0502 (0.000)	9520.0502 (0.000)	9798.4397 (329.5270)	9646.3986 (126.3484)
Zoo	17.1006 (0.000)	17.9505 (1.0353)	18.84565 (1.6819)	20.1190 (2.5939)

Tableau 4.10. Médiane et interquartile de la varaince intra cluster obtenues pour QEAC2(6), QEAC2(1) et Kmeans

C. Comparaison entre QEAC et QEAC2

Les deux QEAC(1) et QEAC2_itrf(1) représentent des interférences qui tendent à être destructives. Leurs résultats sont presque les même, avec une légère différence.

Pour tous les jeux de données, QEAC2(1) réussit à améliorer les résultats mieux que QEAC(1). Cela montre que l'opération de régénération ainsi que l'interférence ont un effet énorme que l'interférence toute seule.

De même pour QEAC2(6) et QEAC(100), on est arrivé à réduire le nombre des individus de la population et améliorer les résultats en même temps. Cela monte l'effet considérable de l'interaction ajouté entre les individus.

4.7 Le rôle de nos algorithmes dans un processus d'extraction de connaissance à partir de données

Dans le cadre d'extraction de connaissances à partir de données, et pour expliquer le rôle de nos algorithmes nous allons se focaliser sur les jeux de données réels où l'application de nos algorithmes de clustering a un sens. L'utilisation des jeux de données synthétiques c'était pour démonter des propriétés intéressantes de nos algorithmes.

Pour les jeux de données réels, les étapes de compréhension du domaine d'application, la définition des objectifs, la sélection des attributs les plus aptes à décrire la problématique, la création du jeu de données cible et parfois la transformation des données sont déjà faites.

L'application de nos algorithmes de clustering sur le jeu de données Dermatology a l'objectif de partitionner des données sur des patients en 6 clusters. Chaque cluster représente un groupe de patients affectés par un type parmi les 6 types de la maladie Erythemato-squamous. Le clustering ici joue un rôle dans le diagnostic différentiel des maladies Erythemato-squamous qui est un problème réel en dermatologie. La difficulté ici est que les 6 maladies partagent les signes cliniques de l'erythema et la graduation avec des différences légères. Chaque patient est décrit par des attributs qui représente des signes cliniques et histopathologiques. Nos algorithmes utilisent ces signes cliniques et histopathologiques pour en extraire des informations utiles sur les patients affectés par la même maladie.

L'application de notre algorithme de clustering sur le jeu de données Thyroid permet d'extraire des informations à partir des données sur des glandes thyroïdes des patients. Ces informations extraites représentent les 3 clusters auxquels appartiennent les glandes. Chaque cluster contient des glandes ayant le même fonctionnement. Un cluster contient des glandes

ayant un fonctionnement normal. Un autre cluster contient des glandes ayant un hypo fonctionnement et le dernier cluster contient des glandes ayant un hyper fonctionnement.

L'application de notre algorithme de clustering sur le jeu de données Wisconsin extrait des informations à partir des cas cliniques relatifs au cancer du sein. Ces informations extraites sont les 2 clusters auxquels appartiennent les cas cliniques. Chaque cluster représente un type du cancer, bénin ou malin.

Un autre domaine sur lequel nous avons appliqué notre algorithme, c'est le domaine des plantes, il est représenté par le jeu de données Iris. Le clustering de fleurs d'Iris permet de découvrir 3 clusters. Chaque cluster regroupe des fleurs du même type. Les 3 types qui existe sont : Iris Setosa, Iris Versicolor et Iris Virginica.

Le jeu de données Soybean contient des données sur la plante Soya. Le clustering de ce jeu de données permet d'extraire des informations sur les 4 types de soya.

Concernant le jeu de données Zoo et comme son nom l'indique, il contient des données sur des animaux. A chaque animale est attribué 16 caractéristiques. Parmi ces caractéristiques, on trouve si l'animal est couvert de poil ou non , s'il est couvert de plumes ou nom, s'il donne des œufs ou du lait ,s'il est aquatique, s'il est prédateur, s'il a des dents, s'il a une colonne vertébrale, s'il est venimeux .. etc. Ces attributs sont booliens. L'algorithme de clustering extrait à partir de ces caractéristiques des informations utiles. Il génère 7 clusters, où chaque cluster contient les animaux de la même catégorie.

Dans un processus d'extraction de connaissances à partir de données, il est nécessaire de passer par une étape d'évaluation avec l'aide d'un expert du domaine afin de relever la pertinence des informations extraites. Pour cette raison, nous avons utilisé une mesure d'évaluation externe qui est la F-mesure. L'intérêt de cette évaluation externe est que la mesure compare le résultat de l'algorithme de clustering avec les classes connus pour chaque jeu de données. Ces classes sont fournit par les experts du domaine de chaque jeu de données.

Dans le cas ou les classes du jeu de données ne sont pas disponibles, nous pouvons recourir à une évaluation interne. Dans notre cas, nous avons proposé d'utiliser la variance intra cluster pour couvrir les cas où notre algorithme s'applique sur des jeux de données sans classes connus.

4.8 Conclusion et travaux futurs

Le but de ce travail est de montrer que des concepts quantiques pourraient être employés pour le clustering de données. Les deux approches proposées se caractérisent par plusieurs particularités. En premier lieu, la représentation quantique de l'espace de recherche

permettant de couvrir toutes les partitions potentielles d'un jeu de données. En plus, cette représentation possède la caractéristique d'appartenance partielle d'un point de donnée aux clusters, ce qui facilite énormément l'exploitation de la théorie des sous ensemble flous. En second lieu, dans la littérature sur les algorithmes évolutionnaires de clustering, le terme évolutionnaire est toujours lié aux opérations de croisement ou mutation. Nous venons de casser ce lien traditionnel en utilisant une stratégie de recherche basée sur une dynamique évolutionnaire quantique. Nous avons employé des opérations quantiques telles que la mesure, l'interférence, la migration locale et globale et une nouvelle opération de régénération.

Les résultats obtenus montrent non seulement la faisabilité des approches mais également leur efficacité et leur capacité à trouver des partitions de bonne qualité.

Par ailleurs, cette approche offre une plateforme où d'autres mesures de qualité de cluster basées sur d'autres aspects différents comme la séparation spatiale et la connectivité peuvent être investiguées. Nous pouvons également étendre notre algorithme au clustering multiobjectif en optimisant plusieurs critères.

D'un autre coté, la fonction objective choisie qui est la variance intra cluster ne détecte pas les clusters de forme de trait ou de cercle, mais il existe une méthode de transformation de clusters de cette forme à une autre forme condensée qui peut être détectée par la variance intra cluster. Cette méthode possède un paramètre à régler autre que le nombre de clusters. Nous pouvons introduire des concepts quantiques pour faire cette transformation sans paramètres et étendre notre algorithme.

Le clustering évolutionnaire quantique proposé est classifié comme une méthode de clustering exact qui limite chaque point du jeu de données à exactement un seul cluster, cela résulte de l'opération de mesure qui génère des partitions binaires. L'idée d'appartenance partielle décrite dans le clustering flou et le clustering possibiliste peut être facilement développé dans le clustering évolutionnaire quantique. Cette extension permet d'exploiter l'avantage du clustering flou qui est la détection de clusters enchevêtrés.

De plus, les données sont parfois transformées avant d'être groupées particulièrement quand différents attributs sont mesurés sur différentes échelles. L'idée d'incorporer l'étape de standardisation dans l'algorithme de clustering en employant la fonction de l'opération de régénération semble être prometteuse.

Conclusion générale

Au cours de ce travail de magistère, nous avons proposé une nouvelle approche pour résoudre le célèbre problème du clustering des données.

Dans un premier temps, il nous a fallu appréhender ce problème en réalisant une étude analytique et comparative entre les différentes méthodes de clustering existantes. Nous avons également insisté sur les métaheuristiques qui résolvent ce problème comme un problème d'optimisation. Dans un second temps, nous avons décidé d'aborder ce travail à l'aide des algorithmes évolutionnaires quantiques en insistant sur l'importance de leur nature quantique. Nous avons donc proposé deux approches QEAC et QEAC2, dont la deuxième est une extension et un enrichissement de la première. En effet, les deux approches traitent le problème de clustering des données comme un problème d'optimisation dans l'espace de partitions. Le codage de chaque partition est basé sur une représentation quantique dont le premier avantage est coder au sein d'une solution toutes partitions possibles correspondant aux différentes affectations de points de données aux clusters, le deuxième avantage est la possibilité d'appartenance partielle d'un point de donnée aux clusters, ce qui facilite énormément l'exploitation de la théorie des sous ensemble flous.

L'autre caractéristique des approches développées est l'utilisation d'une dynamique évolutionnaire quantique permettant une stratégie efficace pour la recherche de la bonne partition. Cette stratégie repose sur un ensemble complémentaire d'opérations quantiques permettant de faire un compromis entre l'exploration et l'exploitation de l'espace de partitions. L'opération d'interférence est de deux types constructive ou destructive. Dans le contexte de notre travail, nous avons combiné entre les deux aspects en jouant sur les paramètres, cela a conduit à une interférence qui n'est pas destructive mais tend à être destructive. La deuxième opération est la migration globale et locale, dont le but est d'ajouter l'interaction entre les solutions de la population, cette migration existe seulement au niveau de la deuxième approche QEAC2 .Sa présence permet de réduire la taille de la population de solutions d'une manière significative par apport à la première approche QEAC. La troisième opération est dite de régénération, elle est basée sur une nouvelle fonction que nous avons développé pour l'étape d'initialisation de l'approche QEAC. Mais son importance et sa capacité à aider à optimiser la fonction objective nous ont poussé à l'intégrer comme une opération quantique dans la deuxième approche QEAC2, où son effet est considérable. Nos

expérimentations nous ont montré la robustesse de nos approches sur divers jeux de données réels provenant de l'UCI repository of machine learning databases [Blake et al,98] et d'autres jeux de données synthétiques générés avec un générateur de clusters gaussien [Handl et al,05a] provenant de [Handl et al,05b] et d'autres jeux de données synthétiques que nous avons crée en se basant sur des distributions normales. Nos résultats montre que nos approches donnent un meilleur partitionnement de données par apport à la méthode Kmeans qui optimise la même fonction objective optimisée par nos approches. Contrairement à Kmeans, notre approche détecte des clusters de différentes densités, de différentes tailles et de différentes formes. Elle a réussit à combattre le problème de dépendance de la partition finale de la partition initiale.

Plusieurs voies de recherches prometteuses peuvent être suivies pour une continuité de ce travail. Tout d'abord, le clustering évolutionnaire quantique proposé est classifié comme une méthode de clustering exact qui limite chaque point du jeu de données à exactement un seul cluster, mais la nature quantique de la représentation adoptée simplifie énormément la possibilité d'appartenance partielle décrite dans le clustering flou et le clustering possibiliste qui a l'avantage de détecter les clusters enchevêtrés.

D'un coté, autres mesures de qualité de cluster basées sur d'autres aspects différents comme la séparation spatiale et la connectivité peuvent être investiguées. Nous pouvons étendre notre approche au clustering multiobjectif basé sur une optimisation multiobjective de plusieurs critères. Mais le problème majeur de cette méthode qui reste encore non résolu est l'optimisation de plusieurs critères opposés. Nous pensons que la nature quantique d'un algorithme évolutionnaire quantique dont une seule exécution peut générer plusieurs solutions comme il est fait par un algorithme multiobjectif peut être exploité pour combattre ce problème.

D'un autre coté, la fonction objective choisie qui est la variance intra cluster ne détecte pas les clusters de forme de trait ou de cercle, mais il existe une méthode de transformation de clusters de cette forme à une autre forme condensée qui peut être détectée par la variance intra cluster. Cette méthode possède un paramètre à régler autre que le nombre de clusters. Nous pouvons introduire des concepts quantiques pour faire cette transformation sans paramètres et étendre notre algorithme.

De plus, les données sont parfois transformées avant d'être groupées particulièrement quand différents attributs sont mesurés sur différentes échelles. L'idée d'incorporer l'étape de standardisation dans l'algorithme de clustering en employant la fonction de l'opération de régénération semble être prometteuse.

En outre, l'adaptation de notre approche à des applications précises est possible. Nous pouvons commencer avec le clustering des documents, dont la fonction de distance la plus appropriée est la distance angulaire calculée a base de cosinus de l'angle qui sépare les deux vecteurs de données. Cette distance convient bien avec notre approche évolutionnaire quantique. Nous voulons également appliquer notre approche sur de jeux de données de grands volumes comme ceux issus du domaine de la bioinformatique où le clustering peut être introduit pour identifier des groupes de gènes ayant des expressions similaires et identifier des groupes de protéines ayant des structure ou séquences similaires. Plusieurs domaines d'applications sont possibles à investiguer vu que le problème de clustering vient de la nature humaine.

Annexe A

Descriptif des jeux de données réels pour le clustering

Cette annexe présente les principaux jeux de données sur lesquels ont été testés les algorithmes QEAC et QEAC2 présentés dans le chapitre 4. Ils ont été choisis parmi les jeux de données disponible dans l'UCI Machine Learning Repository [Blake et al, 98] pour démontrer l'adaptabilité des algorithmes aux différents types de données et aux différentes tailles des problèmes.

A.1 Iris

C'est une jeu de données sur la plante Iris dont la source sont les travaux de R.A. Fisher "The use of multiple measurements in taxonomic problems" Annual Eugenics, 7, Part II, 179-188 (1936). Ces données sont souvent utilisées en classification. Il y a 3 classes d'Iris à découvrir : Iris Setosa, Iris Versicolor et Iris Virginica. Le jeu de données contient 150 instances réparties à égalité dans chaque classe (50 par classe). Il y a quatre attributs numériques :

1. sepal length (longueur du sépale) en cm,
2. sepal width (largeur du sépale) en cm,
3. petal length (longueur du pétale) en cm
4. petal width (largeur du pétale) en cm.

Le tableau A.1 [Jourdon, 03] donne des indications sur les données. La figure A.1 [Clech, 04] montre la distribution du jeu de données par l'intermédiaire des projections des points de données selon tous les 16 paires de dimensions.

	Min	**Max**	**Mean**	**SD**	**Class correlation**
sepal length	4.3	7.9	5.84	0.83	0.7826
sepal width	2.0	4.4	3.05	0.43	-0.4194
petal length	1.0	6.9	3.76	1.76	0.9490
petal width	0.1	2.5	1.20	0.76	0.9565

Tableau A.1 – Statistiques descriptives du jeu de données Iris

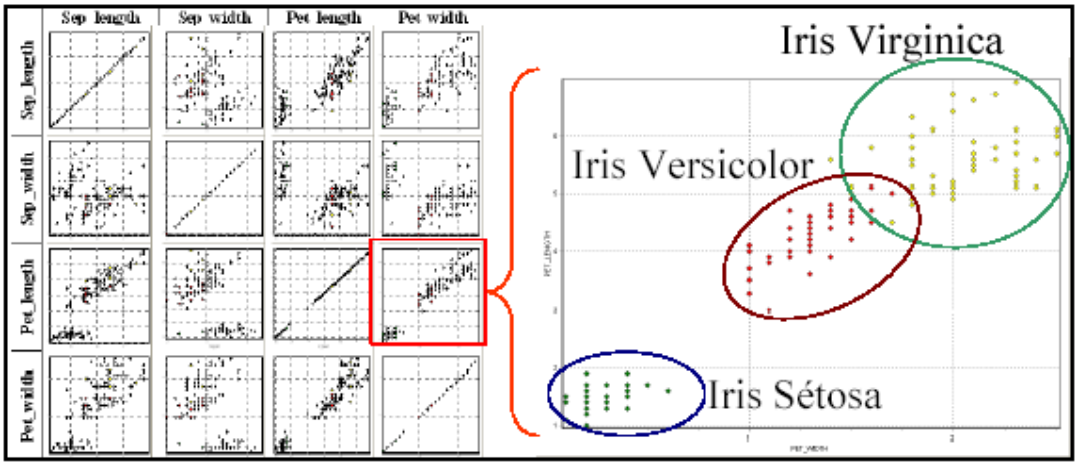


Figure A.1. Distribution du jeu de données iris

A.2 Dermatology

Le but ici est de déterminer le type de maladie Eryhemato-Squamous. Le diagnostic différentiel des maladies Erythemato-squamous est un problème réel en dermatologie. Elles toutes partagent les particularités cliniques de l'Erythema et de la graduation, avec des différences très petites. Les maladies dans ce groupe sont la psoriasis, la dermatite seboreic, planus de lichen, rosea de pityriasis, la dermatite cronic, et la pityriasis rubra pilaris. La difficulté pour le diagnostic différentiel est qu'une maladie peut montrer les particularités d'une autre maladie à l'étape de début et peut avoir les particularités caractéristiques aux étapes suivantes. Des patients ont été la première fois évalués médicalement avec 12 particularités. Après, des échantillons de peau ont été pris pour l'évaluation de 22 particularités histopathologiques. Les valeurs des particularités histopathologiques sont déterminées par une analyse des échantillons sous un microscope. Par conséquent, le jeu de données Dermatology contient les informations sur 366 patients. Leurs cas cliniques et histopathologiques sont décrits avec 33 attributs. Le nombre de classes de maladies est 6.

Attributs cliniques :

1: erythema
2: scaling
3: definite borders
4: itching
5: koebner phenomenon
6: polygonal papules
7: follicular papules
8: oral mucosal involvement
9: knee and elbow involvement
10: scalp involvement
11: family history, (0 or 1)

Attributs histopathologiques

12: melanin incontinence
13: eosinophils in the infiltrate
14: PNL infiltrate
15: fibrosis of the papillary dermis
16: exocytosis
17: acanthosis
18: hyperkeratosis
19: parakeratosis
20: clubbing of the rete ridges
21: elongation of the rete ridges
22: thinning of the suprapapillary epidermis
23: spongiform pustule
24: munro microabcess
25: focal hypergranulosis
26: disappearance of the granular layer
27: vacuolisation and damage of basal layer
28: spongiosis
29: saw-tooth appearance of retes
30: follicular horn plug
31: perifollicular parakeratosis

32: inflammatory monoluclear inflitrate
33: band-like infiltrate

Le jeu de données Dermatology contient 8 valeurs manquantes dans l'attribut age.

A.3 Breast Cancer Wisconsin (Cancer)

Le jeu de données Wisconsin Breast Cancer Database (Cancer) contient les informations médicales de 699 cas cliniques relatifs au cancer du sein classés comme bénin ou malin où 65.5% des cas sont bénins et 34.5% sont malins. Les cas cliniques sont décrits par 9 attributs numériques.

1. Clump Thickness
2. Uniformity of Cell Size
3. Uniformity of Cell Shape
4. Shape Marginal Adhesion
5. Single Epithelial Cell Size
6. Bare Nuclei
7. Bland Chromatin
8. Normal Nucleoli
9. Mitoses

Le jeu de données contient 16 valeurs manquantes.

A.4 Soybean

C'est un petit jeu de données qui est un sous-ensemble de la base de données de soja originale, dont l'origine sont les travaux de Michalski, R.S. "Learning by being told and learning fromexamples: an experimental comparison of the two methodes of knowledge acquisition in the context of developing an expert system for soybean desease diagnoiss", International Journal of Policy Analysis and Information Systems, 1980, 4(2), 125-161.
Soybean contient 47 instances décrites par 35 attributs numériques sans valeurs manquantes. Il est divisé en 4 classes.
Les attribut:

1. date
2. plant-stand
3. precip
4. temp
5. hail
6. crop-hist
7. area-damaged
8. severity
9. seed-tmt
10. germination
11. plant-growth
12. leaves
13. leafspots-halo
14. leafspots-marg
15. leafspot-size
16. leaf-shread
17. leaf-malf
18. leaf-mild
19. stem

20. lodging
21. stem-cankers
22. canker-lesion
23. fruiting-bodies
24. external decay
25. mycelium
26. int-discolor
27. sclerotia
28. fruit-pods
29. fruit spots
30. seed
31. mold-growth
32. seed-discolor
33. seed-size
34. shriveling
35. roots

A.5 Thyroid

Le jeu de données Thyroid contient des informations sur 215 glandes thyroïdes. Le regroupement de ces glandes essaye de prévoir si la thyroïde d'un patient appartient à la classe euthyroidism , à l'hypothyroïdisme ou à l'hyperthyroïdisme. Chaque glande est décrite par 5 attributs numériques :

1. T3-resin uptake test. (A percentage)
2. Total Serum thyroxin as measured by the isotopic displacement method.
3. Total serum triiodothyronine as measured by radioimmuno assay.
4. basal thyroid-stimulating hormone (TSH) as measured by radioimmuno assay.
5. Maximal absolute difference of TSH value after injection of 200 micro grams of thyrotropin-releasing hormone as compared to the basal value.

A.6 Zoo

Le jeu de données Zoo contient des informations sur des animaux décrits par 15 attributs booliens et 1 numérique.

1. hair Boolean
2. feathers Boolean
3. eggs Boolean
4. milk Boolean
5. airborne Boolean
6. aquatic Boolean
7. predator Boolean
8. toothed Boolean
9. backbone Boolean
10. breathes Boolean
11. venomous Boolean
12. fins Boolean
13. legs Numeric (set of values: {0,2,4,5,6,8})
14. tail Boolean
15. domestic Boolean
16. catsize Boolean

Annexe B

Les résultats détaillés de QEAC2 et QEAC

B.1. Les valeurs Max, Médiane, Min et interquartile de la F-mesure obtenus pour QEAC(100), QEAC(1) et Kmeans

Jeu de données	F-mesure	QEAC(100)	QEAC(1)	Kmeans
Dataset1	max	0.9833	0.9933	0.9833
	médiane	0.9833	0.9833	0.9833
	min	0.9833	0.9665	0.9833
	interquartile	0.0000	0.0099	0.0000
Dataset2	max	1.0000	1.0000	1.0000
	médiane	1.0000	1.0000	1.0000
	min	1.0000	1.0000	1.0000
	interquartile	0.0000	0.0000	0.0000
Iris	max	0.8918	0.9200	0.8918
	médiane	0.8918	0.8918	0.8918
	min	0.8918	0.8696	0.8918
	interquartile	0.0000	0.0142	0.0000
Dermatology	max	0.8589	0.9728	0.9534
	médiane	0.8174	0.8164	0.8036
	min	0.7951	0.6580	0.5335
	interquartile	0.0210	0.0882	0.1261
Soybean	max	0.7281	0.8734	0.7668
	médiane	0.7281	0.7483	0.7281
	min	0.7281	0.7281	0.7097
	interquartile	0.0000	0.0600	0.0210
Wisconsin	max	0.9603	0.9677	0.9603
	médiane	0.9603	0.9603	0.9603
	min	0.9603	0.9512	0.9603
	interquartile	0.0000	0.0031	0.0000
Thyroid	max	0.8778	0.8843	0.8620
	médiane	0.8620	0.7561	0.8549
	min	0.8461	0.5576	0.8549
	interquartile	0.0040	0.1186	0.0071
Zoo	max	0.8283	0.8780	0.8992
	médiane	0.8144	0.7620	0.7318
	min	0.7939	0.5825	0.5250
	interquartile	0.0089	0.0859	0.1000

B.2. Les valeurs Max, Médiane, Min et interquartile de la Variance intra cluster obtenus pour QEAC(100), QEAC(1) et Kmeans

Jeux de données	Variance intra cluster	QEAC(100)	QEAC(1)	Kmeans
Dataset1	max	574.7736	610.7144	574.7736
	médiane	574.7736	576.1542	574.7736
	min	574.7736	576.1542	574.7736
	interquartile	0.0000	14.8064	0.0000
Dataset2	max	1.4965	1.4965	1.4965
	médiane	1.4965	1.4965	1.4965
	min	1.4965	1.4965	1.4965
	interquartile	0.0000	0.0000	0.0000
Iris	max	26.3136	28.2894	26.3136
	médiane	26.3136	26.4622	26.3136
	min	26.3136	26.3136	26.3136
	interquartile	0.0000	0.8074	0.0000
Dermatology	max	580.3489	656.4649	636.4322
	médiane	578.3475	609.5857	606.2421
	min	576.729	577.2125	576.6163
	interquartile	1.0723	24.1979	22.8621
Soybean	max	51.8042	58.7898	61.6148
	médiane	51.4909	52.4044	51.4909
	min	51.4909	51.4909	51.4909
	interquartile	0.1305	2.8384	6.6947
Wisconsin	max	9.6616e+003	9669.8084	9.6616e+003
	médiane	9.6616e+003	9.6632e+003	9.6616e+003
	min	9.6616e+003	9661.5869	9.6616e+003
	interquartile	0.000	8.2215	0.000
Thyroid	max	9.6631e+003	1.1872e+004	9.7006e+003
	médiane	9.5328e+003	9.7868e+003	9.6464e+003
	min	9.5201e+003	9.5201e+003	9.5201e+003
	interquartile	40.2614	309.7084	126.3484
Zoo	max	17.5452	22.7003	25.5497
	médiane	17.1911	19.2886	20.1962
	min	17.1006	17.1006	17.1006
	interquartile	0.1531	1.7007	2.6473

B.3. Les valeurs Max, Médiane, Min et interquartile de la F-mesure obtenus pour QEAC2(6), QEAC2(1), QEAC2_itrf(1) et Kmeans

Jeux de données	F-mesure	QEAC2(6)	QEAC2(1)	QEAC2_itrf(1)	Kmeans
Dataset1	max	0.9833	0.9833	0.9900	0.9833
	médiane	0.9833	0.9833	0.9833	0.9833
	min	0.9833	0.9833	0.9698	0.9833
	interquartile	0.0000	0.0000	0.0066	0.0000
Dataset2	max	1.0000	1.0000	1.0000	1.0000
	médiane	1.0000	1.0000	1.0000	1.0000
	min	1.0000	1.0000	1.0000	1.0000
	interquartile	0.0000	0.0000	0.0000	0.0000

2d4c	max	0.9399	0.9399	0.9648	0.9399
	médiane	0.9399	0.9399	0.9345	0.9399
	min	0.9399	0.9399	0.8370	0.7512
	interquartile	0.0000	0.0000	0.0420	0.1558
2d10c	max	0.9451	0.9467	0.9365	0.9484
	médiane	0.9451	0.9054	0.8321	0.8355
	min	0.9451	0.7529	0.7035	0.6290
	interquartile	0.0000	0.0692	0.0634	0.1046
10d4c	max	0.9841	0.9841	0.9881	0.9841
	médiane	0.9841	0.9841	0.9602	0.9841
	min	0.9841	0.9841	0.7735	0.7134
	interquartile	0.0000	0.0000	0.0812	0.2056
10d10c	max	1.0000	1.0000	0.9845	1.0000
	médiane	1.0000	0.9270	0.8777	0.8629
	min	1.0000	0.7666	0.7532	0.6835
	interquartile	0.0000	0.0715	0.0618	0.1005
Iris	max	0.8918	0.8918	0.9200	0.8918
	médiane	0.8918	0.8918	0.8918	0.8918
	min	0.8918	0.8918	0.8718	0.8918
	interquartile	0.0000	0.0000	0.0135	0.0000
Dermatology	max	0.8107	0.8725	0.9755	0.9534
	médiane	0.8107	0.8107	0.8086	0.8036
	min	0.8107	0.761	0.6408	0.5281
	interquartile	0.0000	0.0319	0.0928	0.1261
Soybean	max	0.7281	0.7281	0.8156	0.7668
	médiane	0.7281	0.7281	0.7483	0.7281
	min	0.7281	0.7281	0.7281	0.7097
	interquartile	0.0000	0.0000	0.0399	0.0210
Wisconsin	max	0.9603	0.9603	0.9662	0.9603
	médiane	0.9603	0.9603	0.9603	0.9603
	min	0.9603	0.9603	0.9543	0.9603
	interquartile	0.0000	0.0000	0.0031	0.0000
Thyroid	max	0.8620	0.8620	0.8863	0.8708
	médiane	0.8620	0.8620	0.7521	0.8549
	min	0.8620	0.8620	0.5576	0.8549
	interquartile	0.0000	0.0000	0.1612	0.0071
Zoo	max	0.8164	0.8803	0.8874	0.8685
	médiane	0.8164	0.7775	0.7583	0.7318
	min	0.8164	0.5767	0.5585	0.5658
	interquartile	0.0000	0.1088	0.0963	0.1000

B.4. Les valeurs Max, Médiane, Min et interquartile de la Variance intra cluster obtenus pour QEAC2(6), QEAC2(1), QEAC2_itrf(1) et Kmeans

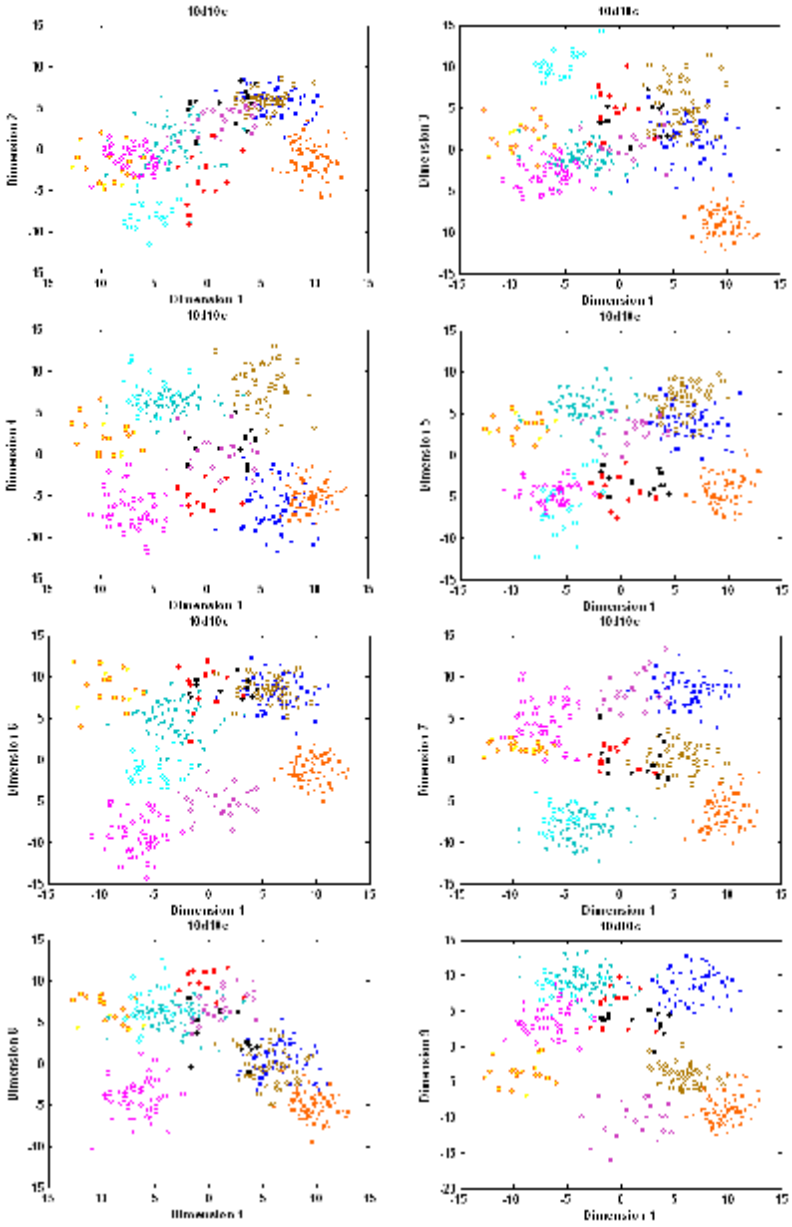
Jeux de données	Variance intra cluster	QEAC2(6)	QEAC2(1)	QEAC2_itrf(1)	Kmeans
Dataset1	max	574.7736	574.7736	604.3685	574.7736
	médiane	574.7736	574.7736	575.7629	574.7736
	min	574.7736	574.7736	574.7736	574.7736

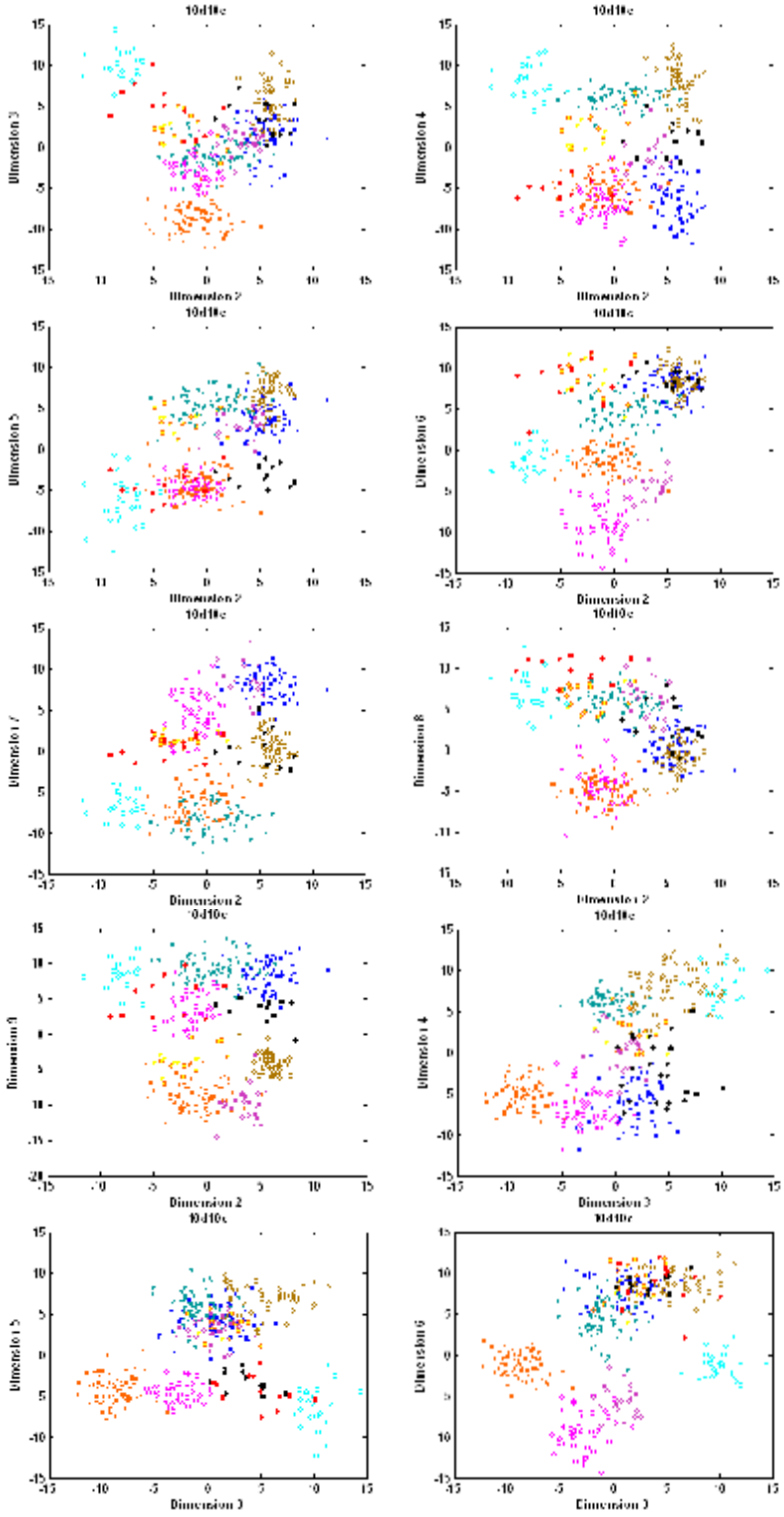
	interquartile	0.0000	0.0000	12.3922	0.0000
Dataset2	max	1.4965	1.4965	1.4965	1.4965
	médiane	1.4965	1.4965	1.4965	1.4965
	min	1.4965	1.4965	1.4965	1.4965
	interquartile	0.0000	0.0000	0.0000	0.0000
2d4c	max	1.4670e+003	1.4670e+003	1753.6506	2707.8615
	médiane	1.4670e+003	1.4670e+003	1483.4882	1467.0372
	min	1.4670e+003	1.4670e+003	1467.0372	1467.0372
	interquartile	0.0000	0.0000	118.1172	1.1155e+003
2d10c	max	70.8330	91.9683	119.5482	159.795
	médiane	70.8330	78.9347	91.5807	94.6520
	min	70.8330	70.833	71.1367	70.833
	interquartile	0.0000	6.3199	13.6648	34.7470
10d4c	max	27403.6099	27403.6099	33084.7118	40045.9667
	médiane	27403.6099	27403.6099	27767.07355	27403.6099
	min	27403.6099	27403.6099	27403.6099	27403.6099
	interquartile	0.0000	0.0000	2.5456e+003	6.7332e+003
10d10c	max	1.6101e+003	1973.4378	3426.9336	3650.719
	médiane	1.6101e+003	1.7028e+003	2167.3988	1.9607e+003
	min	1.6101e+003	1610.0554	1610.0554	1610.0554
	interquartile	0.0000	251.5430	588.7767	688.1683
Iris	max	26.3136	26.3136	28.2761	26.3136
	médiane	26.3136	26.3136	26.4112	26.3136
	min	26.3136	26.3136	26.3136	26.3136
	interquartile	0.0000	0.0000	0.8010	0.0000
Dermatology	max	577.0363	614.262	642.9661	636.4322
	médiane	576.6163	582.7106	609.9999	606.4401
	min	576.6163	576.6163	577.3983	576.6163
	interquartile	0.4200	20.6060	22.1420	22.8304
Soybean	max	51.4909	51.4909	59.0221	61.6148
	médiane	51.4909	51.4909	52.0386	51.4909
	min	51.4909	51.4909	51.4909	51.4909
	interquartile	0.0000	0.0000	3.0253	6.6947
Wisconsin	max	9661.5869	9661.5869	9668.7557	9661.5869
	médiane	9.6616e+003	9.6616e+003	9663.1513	9.6616e+003
	min	9661.5869	9661.5869	9661.5869	9661.5869
	interquartile	0.0000	0.0000	7.1688	0.0000
Thyroid	max	9.5201e+003	9.5201e+003	10494.611	9700.6359
	médiane	9.5201e+003	9.5201e+003	9798.4397	9.6464e+003
	min	9.5201e+003	9.5201e+003	9520.0502	9520.0502
	interquartile	0.0000	0.0000	329.5270	126.3484
Zoo	max	17.1006	20.2655	22.0893	25.5497
	médiane	17.1006	17.9505	18.84565	20.1190
	min	17.1006	17.1006	17.1006	17.1006
	interquartile	0.0000	1.0353	1.6819	2.5939

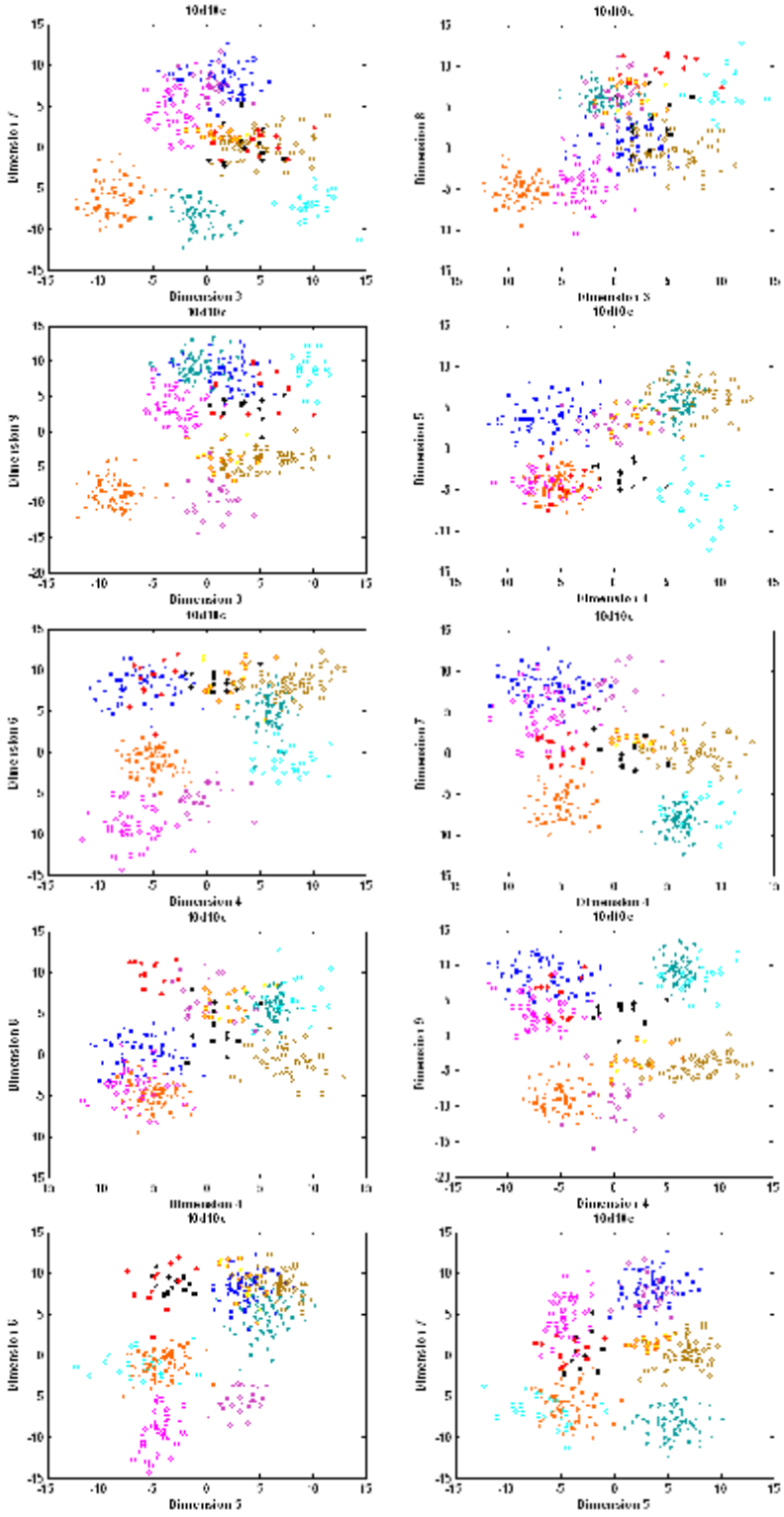
Annexe B

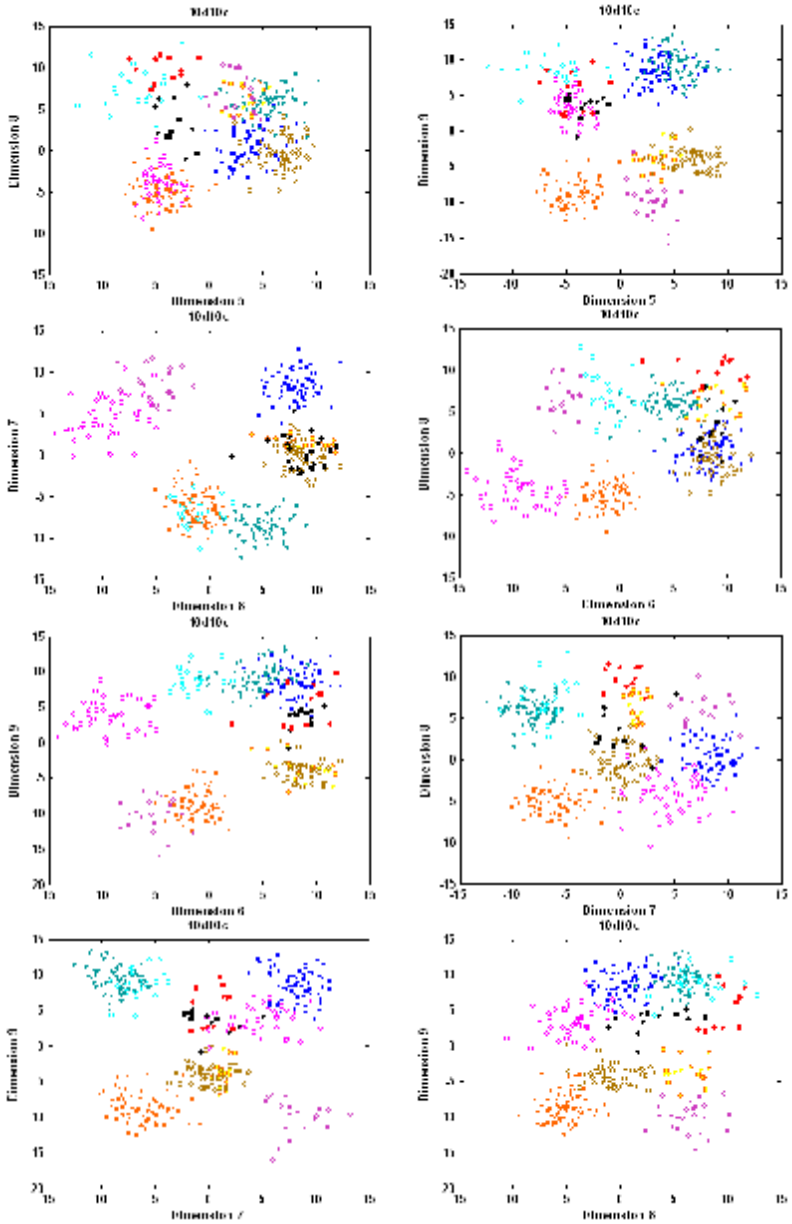
Distributions des deux jeux de données synthétiques 10d10c et 10d4c

C.1 Distributions du jeu de données 10d10c

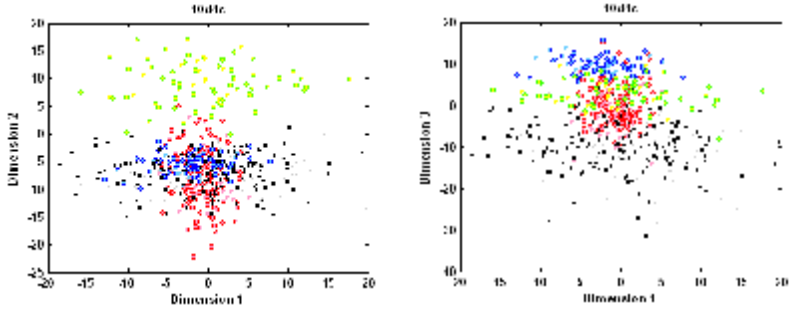


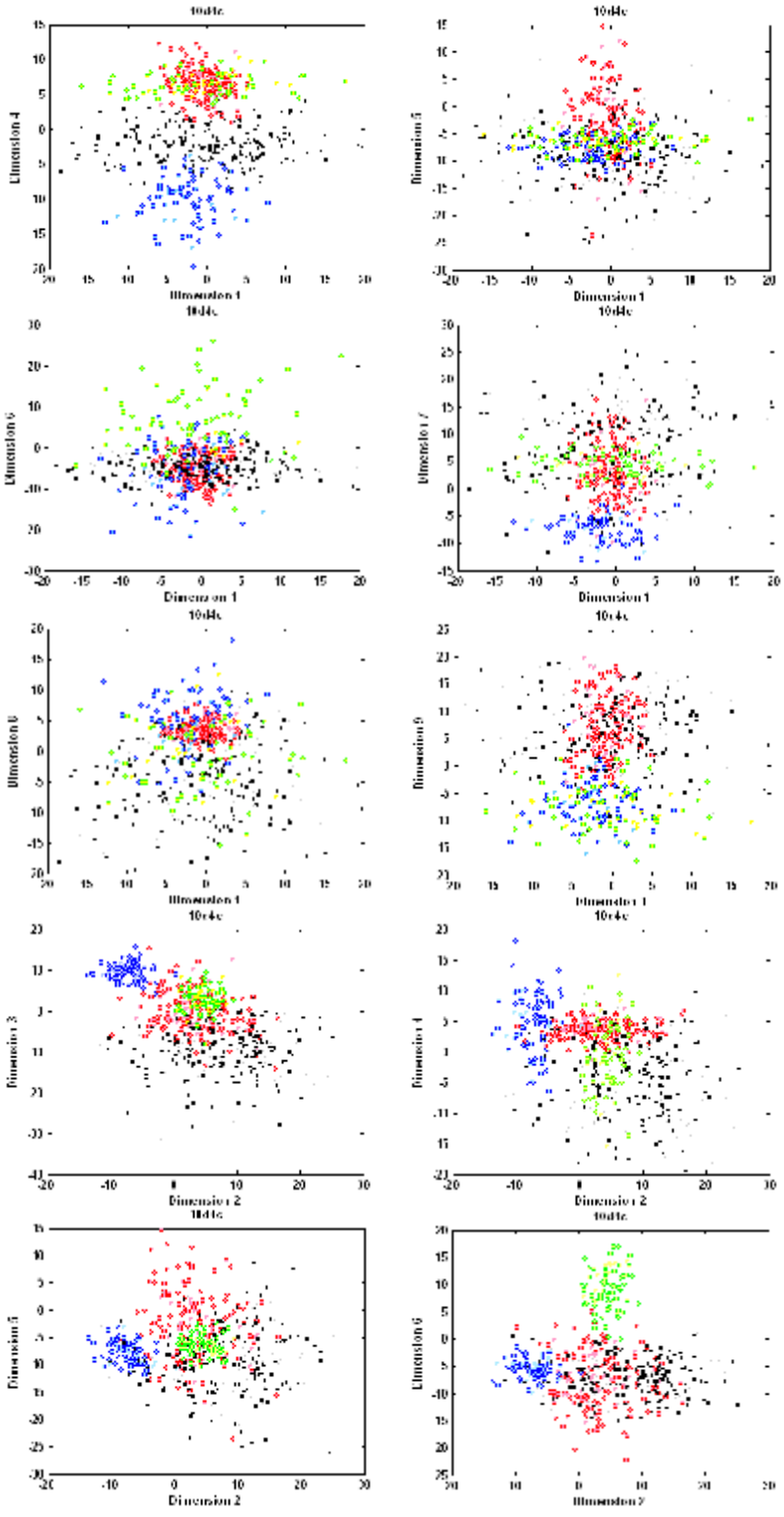

10d10c
Dimension 3
Dimension 2
10d10c
Dimension 4
Dimension 2
10d10c
Dimension 5
Dimension 2
10d10c
Dimension 6
Dimension 2
10d10c
Dimension 7
Dimension 2
10d10c
Dimension 8
Dimension 2
10d10c
Dimension 9
Dimension 2
10d10c
Dimension 4
Dimension 3
10d10c
Dimension 5
Dimension 3
10d10c
Dimension 6
Dimension 3

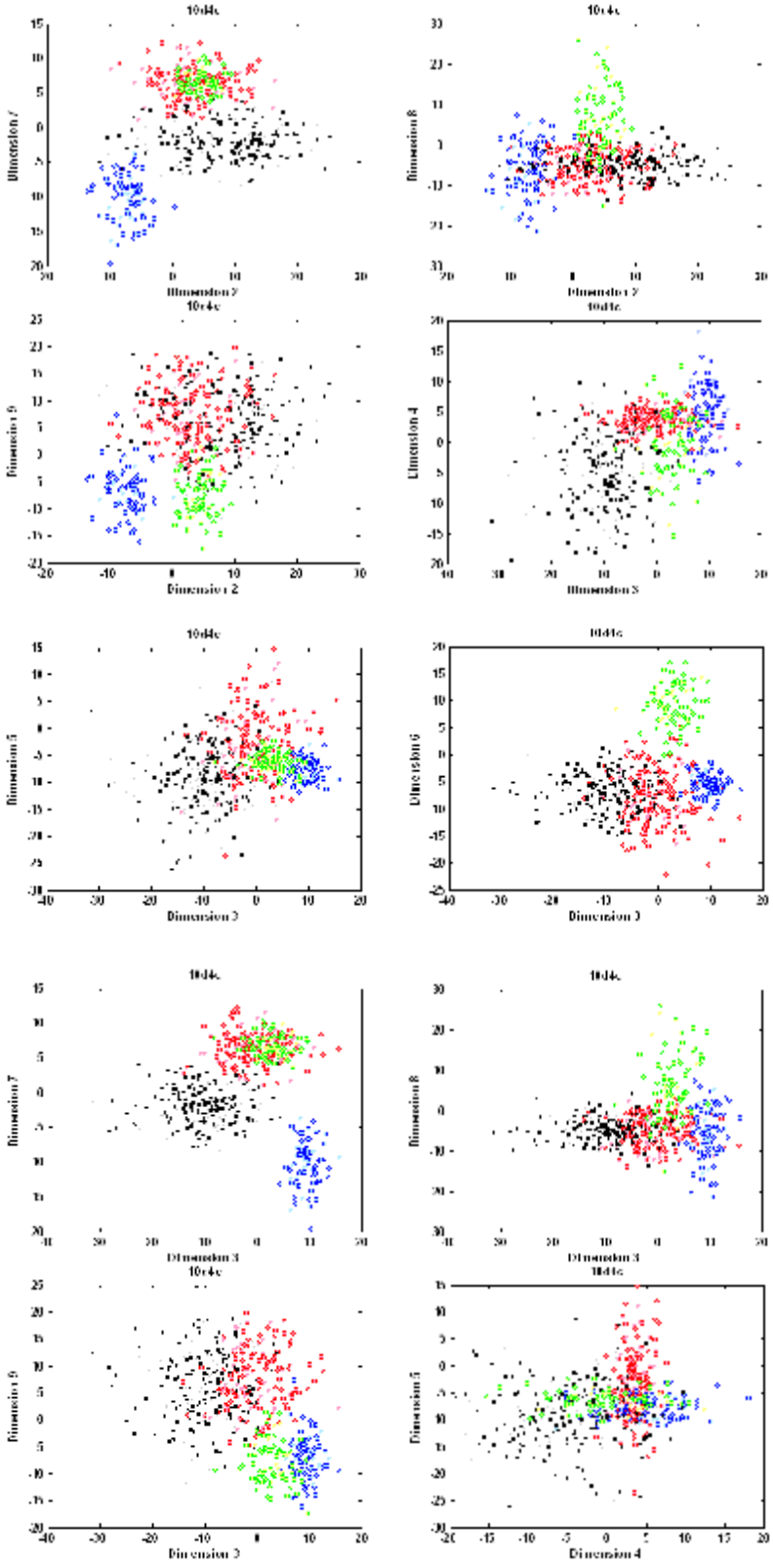

10d10c
Dimension 7
Dimension 3
10d10c
Dimension 8
Dimension 8
10d10c
Dimension 9
Dimension 3
10d10c
Dimension 5
Dimension 1
10d10c
Dimension 6
Dimension 4
10d10c
Dimension 7
Dimension 1
10d10c
Dimension 8
Dimension 4
10d10c
Dimension 9
Dimension 4
10d10c
Dimension 5
Dimension 5
10d10c
Dimension 7
Dimension 5

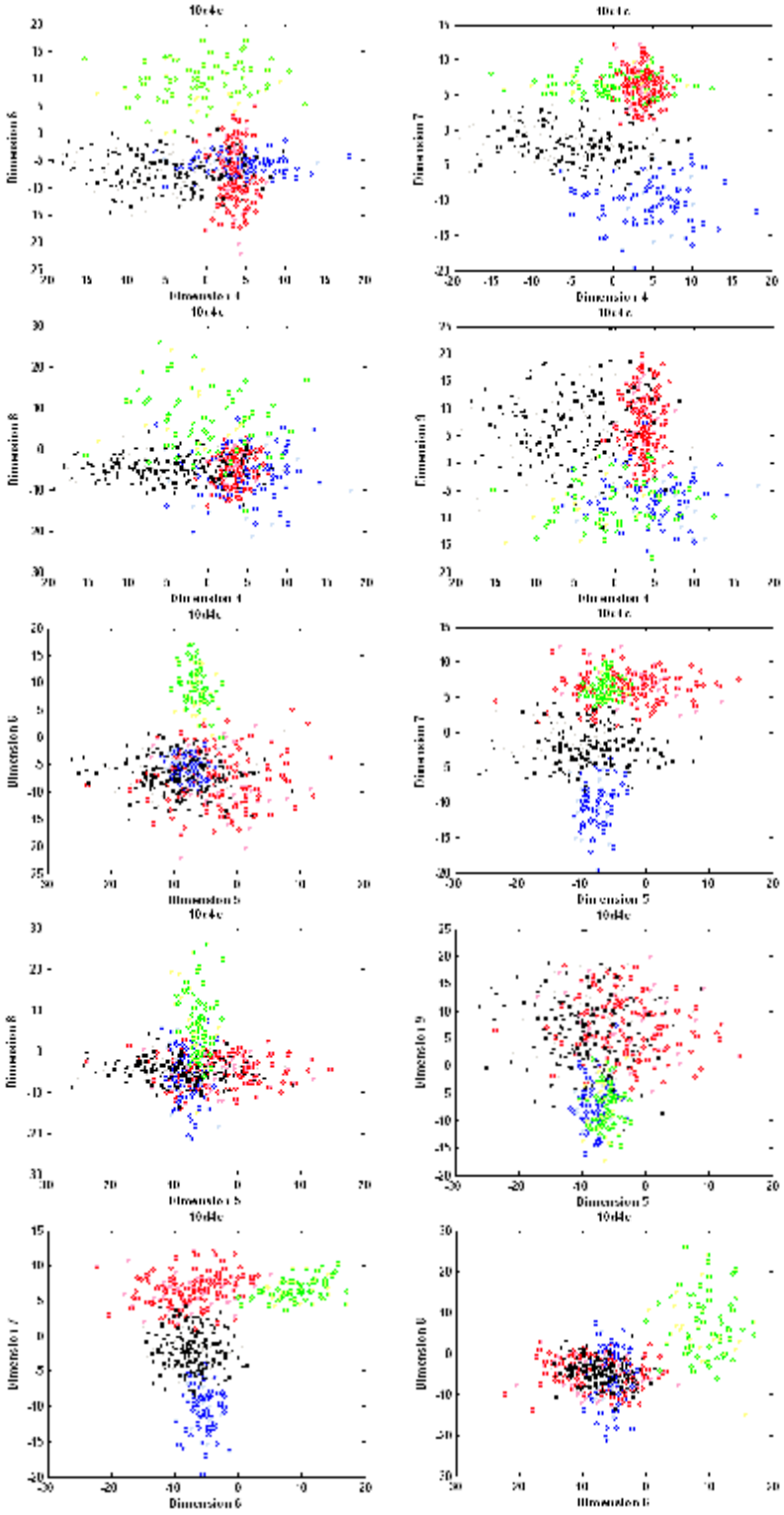


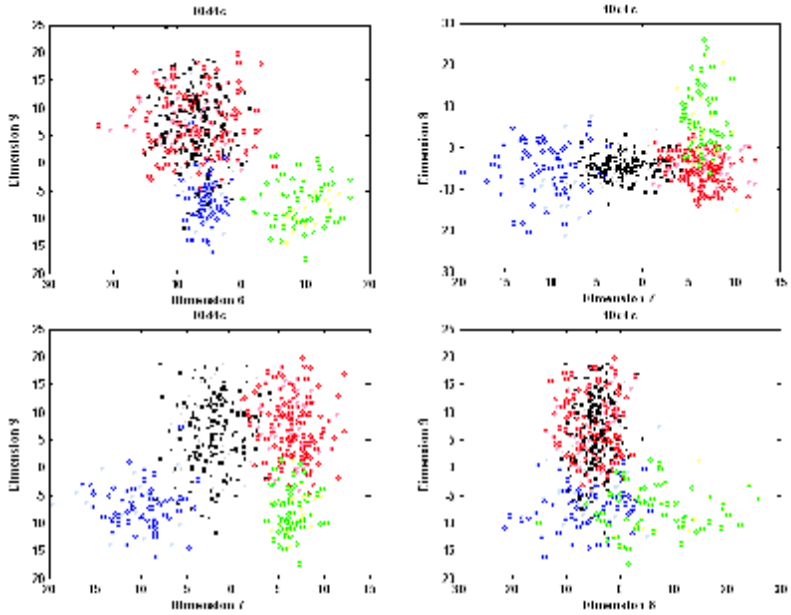
C.1 Distributions du jeu de données 10d4c




10d4c
Dimension 1
Dimension 4
10d4c
Dimension 1
Dimension 5
10d4c
Dimension 1
Dimension 6
10d4c
Dimension 1
10d4c
Dimension 1
Dimension 8
10d4c
Dimension 1
10d4c
Dimension 2
Dimension 3
10d4c
Dimension 2
Dimension 4
10d4c
Dimension 2
Dimension 5
10d4c
Dimension 2






10d1c
Dimension 9
Dimension 6
10d1c
Dimension 9
Dimension 7
10d1c
Dimension 9
Dimension 7
10d1c
Dimension 9
Dimension 8

Bibliographie

[Agrawal ,93]: R. Agrawal, T. Imielinski, and A. "Swami. Mining association rules between sets of items in large databases". In Proceedings of the ACMSIGMOD Conference on Management of Data, Washington, USA, pages 207–216, 1993.

[Agrawal et al ,94]: R. Agrawal and R. Srikant. "Fast algorithms for mining association rules". In *20`eme Conference on Very Large Databases, Santiago, Chili*, pages 487-499, Septembre 1994.

[Agrawal et al, 96]: R.Agrawal, H. Mannila, R. Srikant, H. Toivonen, et A.I.Verkamo. "Fast Discovery of Association Rules", In U. Fayyad, G. Piatetsky-Shaprio, P. Smyth and R.Uthurusamy (eds.), *Advances in Knowledge Discovery and Data Mining*, MIT Press,Cambridge, MA., 307-328, 1996.

[Anderberg,73]: M. R. Anderberg. Cluster Analysis For Applications. Academic Press, New York,1973.

[Ankerst et al ,99]: M.Ankerst , M.Breunig , H.P. Kriegel , J.Sander. "OPTICS: Ordering Points to Identify the Clustering Structure". In the Proc. ACM SIGMOD'99Int. Conf. on Management of Data,1999.

[Azzag et al, 03] :N. Azzag, H. Monmarché, M. Slimane, G. Venturini, et C. Guinot. "Anttree : a new model for clustering with artificial ants". In IEEE Congress on Evolutionary Computation, vol. 1, pp. 2642-2647, 2003.

[Azzag et al ,04b] : H. Azzag, F. Picarougne, C. Guinot, G. Venturini , 2004 , "Un survol des algorithmes biomimétiques pour la classification". Classification et fouille de donnée, p. 13-24, RNTI-C-1, Cépaduès, 2004.

[Azzag et al,04] : H. Azzag, C. Guinot, G. Venturini . "Classification automatique de documents : application au web". 11eme Rencontres de la Société Francophone de Classification (SFC), pp. 91-94, Bordeaux, France, 2004.

[Azzag et al,06] : H. Azzag, C. Guinot, G. Venturini . "Classification hiérarchique et visualisation de pages Web". Des 6èmes journées francophones d'Extraction et Gestion des Connaissances, Lille, pp. 5-16, 2006.

[Babu et al,94]: G. P. Babu et M. N. Murty. "Connectionist approach for clustering". In Proceedings International Conference on Neural Networks, pp.4661–4666, 1994.

[Bellaachia et al,02] : A. Bellaachia , D. Portnoy, Y. Chen et A.G. Elkahloun. "E-CAST: A Data Mining Algorithm For Gene Expression Data", Workshop on Data Mining in Bioinformatics ,pp 49-54,2002.

[Berkhin,02] : P. Berkhin, "Survey of Clustering Data Mining Techniques", Technical report, Accrue software, San Jose,California, 2002.

[Bezdek et al ,94]: J.C. Bezdek, S. Boggavarapu, L. Hall et A. Bensaid. "Genetic algorithm guided clustering". In Proceedings of the First IEEE Conference on Evolutionary Computation, pages 34-39, 1994.

[Bezdek,81] :J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.

[Bhalla et al,02] : A.Bhalla, K.Eguro,M.Hayward. "Quantum Computing, Shor's Algorithm and parallelism", rapport technique, 2002.

[Bilmes et al,97]: J.Bilmes, A.Vahdat, W.Hsu et E.J.Im. "Empirical observations of probabilistic heuristics for the clustering problem". Technical Report TR-97-018, International Computer Science Institute, University of California, Berkeley, CA, 1997.

[Blais, 03]: A.Blais. "Algorithmes et architectures pour ordinateurs quantiques supraconducteurs". Ann. Phys. Fr. 28 , No 5, 2003.

[Blake et al,98]: C. L. Blake and C. J. Merz. UCI repository of machine learning databases,1998.http://www.ics.uci.edu/~mlearn/Machine-Learning.html

[Castro et al ,00]:L. N. de Castro, F.J. Von Zuben. "An Evolutionary Immune Network for Data Clustering". In Proc. of the IEEE SBRN (Brazilian Symposium on Artificial Neural Networks), pp. 84-89, 2000.

[Clech,04] : J.Clech. "Contribution méthodologique à la fouille de données complexes". Thèse de doctorat, université Lumière Lyon2, 2004.

[Cooley, 00]: R.Cooley, "Web Usage Mining: Discovery and Application of Interesting patterns from Web Data", Thèse de doctorat, Université de Minnesota, 2000.

[Courtier, 05]: O. Courtier. " Contribution à la fouille de données : règles d'association et interactivité au sein d'un processus d'extraction de connaissances dans les données ", Thèse de Doctorat, Université d'Artois, 2005.

[Christophe,04] : C. Christophe. "SearchXQ : une méthode d'aide à la navigation fondée sur Ωmeans, algorithme de classification non-supervisée. Application sur un corpus juridique Français".Thèse de Doctorat,Ecole de Mine de Paris collège doctoral, 2004.

[Davies et al,79]:Davies, D.L. and Bouldin, D.W. A Cluster Separation Measure. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1(2), 224–227, 1979.

[Deneubourg et al,90] : J.L. Deneubourg, S. Goss, N.R. Franks, A. Sendova Franks, C. Detrain, et L. Chretien. "The dynamics of collective sorting: robot-like ant and ant-like robots". In Proceedings of the First International Conference on Simulation of Adaptive Behavior, pp 356–365, 1990.

[Deutsch,85] : David Deutsch. "Quantum theory, the church-turing principle and the universal quantum computer". Proceedings of the Royal Society of London A, 400 :97–117, 1985.

[Draa,04] :A. Draa " Une nouvelle approche pour le recalage des images multimodales, basée sur l'informatique quantique et les algorithmes évolutionnaires", Mémoire de magistère en Informatique, Département d'informatique Université Mentouri Constantine. Année 2004.

[Dunn,74] :J.C. Dunn, A fuzzy relative of the ISODATA process and its use in detecting compact, well-separated clusters, J. Cybern. 3, pp 32–57, 1974.

[Ester et al,96]: M.Ester, H.P.Kriegel, J.Sander, et X. Xu. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In Proceeding of 2nd Int. Conference on Knowledge Discovery and Data Mining, pp. 226–231, 1996.

[Faber et al, 94]: V.Faber, J. C. Hochberg, P. M .Kelly, T. R Thomas, et J. M White, "Concept extraction: A data-mining technique". Los Alamos Science *22*, pp 122–149, 1994.

[Fayyad et al , 96b]: U. Fayyad, G. Piatetsky-Shapiro, et P. Smyth. "The KDD process for Extracting Useful Knowledge from Volumes of Data ". In Communications of the ACM, vol 39 n°11: pp 27-34, 1996.

[Fayyad et al, 96a]: U. Fayyad, G. Piatetsky-Shapiro, et P. Smyth. "From data mining to knowledge discovery in databases". In AI Magazine ,pp. 37-54,1996.

[Feynmann,82] : R.Feynmann. "Simulating physics with computers". International Journal of Theoretical Physics, 21(6–7) pp. 467–488, 1982.

[Forgy,65]: E.Forgy. "Cluster analysis of multivariate data: Efficiency versus interpretability of classification". Biometric society meeting,1965.

[Gautier et al,01] : J.Gautier, C. Joachim, J.P. Poizat, D. Vuillaume et S. Raud ."La microélectronique du futur aux etats-unis électronique moléculaire et organique, calcul quantique, ordinateur ADN , nouvelles architectures",rapport technique,2001.

[Goethals et al,03] :B. Goethals et M.J.Zaki. "FIMI'03 :Workshop on Frequent Itemset Mining Implementations". In FIMI'03 Workshop on Frequent Itemset Mining Implementations,2003.

[Greene, 03] :W.A. Greene. "Unsupervised hierarchical clustering via a genetic algorithm". In IEEE Press, editor, Proceedings of the 2003 Congress on Evolutionary Computation, pages 998-1005, 2003.

[Grosshans,02] :F. Grosshans. "Communication et cryptographie quantiques avec des variables continues ".Thèse de Doctorat, université Paris XI UFR scientifique d'Orsay, 2002.

[Grover,96] : L.K. Grover, "A fast quantum mechanical algorithm for database search", Proceedings ,28th Annual Symposium on the theory of Computing, pp.212-219, 1996.

[Guha et al,98]: S.Guha, R. Rastogi, et K. Shim . "CURE: An Efficient Clustering Algorithm for Large Databases". In Proceedings of the ACM SIGMOD Conference, 1998.

[Guo,03]:L.Guo."Applying Data Mining Techniques in Property/Casualty Insurance", In the Forum of the Casualty Actuarial Society ,pp. 1-25,2003.

[Halkidi et al,00]:M. Halkidi, M. Vazirgiannis, I. Batistakis. "Quality scheme assessment in the clustering process", In the Proceedings of the Fourth European Conference on Principles of data mining and Knowlegde discovery, Vol1910 of LNCS, pp 265-267.Springer-Verlag, 2000.

[Halkidi et al,01a] :M. Halkidi, Y. Batistakis, M. Vazirgiannis, "On Clustering Validation Techniques", Intelligent Information Systems Journal, Kluwer Pulishers, vol.17, n°2-3, pp. 107-145, 2001.

[Halkidi et al,01b]: M. Halkidi ,M.Vazirgiannis. "Clustering Validity Assessment: Finding the optimal partitioning of a data set ",in the Proceedings of ICDM Conference ,California, USA, 2001.

[Halkidi et al,02]:M. Halkidi, Y. Batistakis, M. Vazirgiannis. "Cluster Validity Methods: Part II", SIGMOD Record, 2002.

[Hall et al ,99]: L. O. Hall, I. B. Oezyurt, and J. C. Bezdek. "Clustering with a genetically optimized approach". In Proceedings of the IEEE Transactions on evolutionary Computation, vol 3 n°2, pp.103-112, 1999.

[Han et al, 06]J.Han et M. Kamber, "Data Mining: Concepts and Techniques", Slides for Textbook, Chapter 3: Data Preprocessing, Février1, 2006. Web site : http://www.cs.sfu.ca

[Han et al,00] :K.H.Han et J.H.Kim, "Genetic Quantum Algorithm and its Application to Combinatorial Optimization Problem," In Proceedings of the 2000 Congress on Evolutionary Computation, IEEE Press, pp. 1354-1360, July 2000.

[Han et al,02] :K.H. Han and J.H. Kim, "Quantum-Inspired Evolutionary Algorithm for a Class of Combinatorial Optimization", IEEE Transactions on Evolutionary Computation, vol. 6, no. 6, pp.580-593, December 2002.

[Han et al,03] : K.H.Han, J.Y.Hwang, K.H.Han, J.H. Kim et K.H..Park, "A Quantum-inspired Evolutionary Computing Algorithm for Disk Allocation Method," IEICE Transactions on Information and Systems, IEICE Press, Vol. E86-D, No. 3, pp. 645-649, March 2003.

[Han et al,04]:K.H. Han and J.H. Kim, "Quantum-inspired Evolutionary Algorithms with a New Termination Criterion, Hε Gate, and Two Phase Scheme," IEEE Transactions on Evolutionary Computation, IEEE Press, Vol. 8, No. 2, pp. 156-169 , 2004.

[Handl et al,04]:J.Handl et J.Knowles. "Evolutionary multiobjective clustering". In the Proceedings of the Eighth International Conference on Parallel Problem Solving from Nature (PPSN VIII), pp 1081-1091. LNCS 3242, 2004.

[Handl et al,05a]:J. Handl and J. Knowles. "Improvements to the scalability of multiobjective clustering". In the Proceedings of the Congress on Evolutionary Computation, Vol 3, pp. 2372-2379, 2005.

[Handl et al,05b]:J. Handl and J. Knowles. Cluster generators: synthetic data for the evaluation of clustering algorithms, 2005. http://www.whatcounter.com

[Handl et al,05c]: J.Handl et J.Knowles. "Exploiting the trade-off -- the benefits of multiple objectives in data clustering". In the Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization, pp 547-560, LNCS 3410, 2005.

[Handl et al,05d]: J.Handl et J.Knowles. "Multiobjective clustering around medoids". In the Proceedings of the Congress on Evolutionary Computation (CEC 2005) Copyright IEEE,. Vol.1, pp 632-639, 2005.

[Handl, 03]: J Handl, Ant-based methods for tasks of clustering and topographic mapping: extensions, analysis and comparison with alternative techniques. Masters Thesis, Université d' Erlangen-Nurnberg, Erlangen, Germany, 2003.

[Homayouni et al,05] : R.Homayouni, K.Heinrich, L. Wei et M.W.Berry. "Gene clustering by Latent Semantic Indexing of MEDLINE abstracts ", Bioinformatics, Vol. 21, no. 1, pp 104–115, 2005.

[Hubert et al,85]: L.Hubert et P.Arabie. Comparing partitions. Journal of classification, Vol 2 ,pp 193-218, 1985.

[Jain et al, 99] : A.K. Jain, M.N. Murty et P.J. Flynn. "Data Clustering: A Review", ACM Computing Surveys, Vol. 31, No. 3,pp 264- 323 , 1999.

[Jain et al,88] : A. K. Jain et R.C. Dubes . "Algorithms for Clustering Data", Prentice Hall advanced reference series,1988.

[Jang et al,04] :J.S.Jang, K.H. Han et J.H.Kim, "Evolutionary algorithm-based face verification," Pattern Recognition Letters, Elsevier B. V., Vol. 25, No. 16, pp. 1857-1865, 2004.

[Jollois,03]:F,X,Jollois."Contribution de la classification automatique à la fouille de données". Thèse de Doctorat,Université de Metz,2003.

[Jourdan, 03] : L. Jourdan. "Métaheuristiques pour l'extraction de connaissances : application à la génomique ", Thèse de Doctorat , Université des sciences et technologies de Lille, 2003.

[Kennedy et al ,95]: J. Kennedy et R.C. Eberhart. "Particle swarm optimization". In the Proceedings of IEEE International Conference on Neural Networks, pp 1942–1948, 1995.

[Kodratoff, 98] : Y.Kodratoff. "Techniques et outils de l'extraction de connaissances à partir des données", Signaux n°92

[Kotsiantis et al,04]:S.B. Kotsiantis, P. E. Pintelas, "Recent Advances in Clustering: A Brief Survey", WSEAS Transactions on Information Science and Applications, Vol 1(1), pp.73–81,2004.

[Krasnogor et al, 04] : N. Krasnogor, et D. A. Pelta. "Measuring the similarity of protein structures by means of the universal similarity metric", Bioinformatics, Vol. 20 no. 7, pp1015–1021,2004.

[Krishna et al ,99] :K. Krishna and M. Murty. "Genetic k-means algorithm". In Proceedings of the IEEE Transactions on Systems, Man, and Cybernetics - PartB : Cybernetics, vol .29,n°.3 ,pp 433-439, 1999.

[Krishnapuram et al,96]:R. Krishnapuram, J. Keller, "The possibilistic C-Means algorithm: insights and recommendations". IEEE Trans. on Fuzzy Systems, 4, pp 385-393, 1996.

[Kruskal, 56] : J. Kruskal. "On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem". In the Proceedings of the American Mathematical Society, vol.7, pp.48-50, 1956.

[Kumar, 00]: V. Kumar, "An Introduction to Cluster Analysis for Data Mining".Technical report, C.S. Dept. Univ. Minnesota, 2000.

[Labroche et al ,02]: N. Labroche, F.J. Richard, N. Monmarché, A. Lenoir, G. Venturini . "Modelling of the chemical recognition system of ants". In the International Workshop on Self-Organization and Evolution of Social Behaviour, pp 283-292, 2002.

[Lance et al,67]:G.Lance et W.Williams. "A general theory of classification sorting strategies". Computer Journal, 9, 373-386, 1967.

[Law et al,04]:M. H. C. Law, A. P. Topchy and A. K. Jain, "Multiobjective Data Clustering",In the proceeding of IEEE Computer Society Conference on Computer Vision and Pattern Recognition,Vol 2, pp 424-430**,** 2004.

[Le Bellac ,03]: M. Le Bellac, "Introduction à l'informatique quantique", Cours donné à l'Ecole Supérieure de Sciences Informatiques (ESSI), Octobre 2003.

[Lee et al,97]: D. H. Lee et M. H. Kim, "Database summarization using fuzzy ISA hierarchies", IEEE Transactions on Systems Man and Cybernetics. Part B-Cybernetics, vol. 27, pp. 68-78, 1997.

[Lee, 96]:H.Y .Lee et H.L. Ong. "Visualization support for data mining". IEEE Expert 11, pp 69-75, 1996.

[Lefébure et al, 01]: R.Lefébure et G.Venturi. " Data mining Gestion de la relation client Personalisation de sites web ", Editions Eyrolles, 2001.

[Lévi,04] : B. Lévi."Simulation de systèmes quantiques sur un ordinateur quantique réaliste". Thèse de Doctorat, Université Paris 7 – Denis Diderot UFR de physique,2004.

[Lioyd,82] : S.Lioyd.Least squares quantization in PCM.IEEE Transactions on Information Theory,Vol 28 n2,pp 127-138, 1982.

[Liu, 68]: G. L. Liu. Introduction to combinatorial Mathematics. McGraw Hill, 1968.

[Lumer et al ,94] : E.D. Lumer et B. Faieta. "Diversity and adaptation in populations of clustering ants". In the Proceedings of the Third International Conference on Simulation of Adaptive Behaviour, pages 501–508, 1994.

[M.Ester et al ,98] :M.Ester, H.P.Kriegel , J.Sander,M.Wimmer et X. Xu . "Incremental Clustering for Mining in a Data Warehousing Environment". In the Proceedings of the 24th VLDB Conference New York, USA, 1998.

[Macqueen,67]: J.B.Macqueen. "Some methods for classification and analysis of multivariate observation", In the proceeding of 5th Symposium on math, statistics and probability, pp. 281-297, 1967.

[Maulik et al,00]: U. Maulik and S. Bandyopadhyay. "Genetic algorithm-based clustering technique". Pattern Recognition, vol.33 pp.1455–1465, 2000.

[Merwe et al ,03]:D.V.D. Merwe et A.Engelbrecht. "Data clustering using particle swarm optimization". In the Proceedings of IEEE Congress on Evolutionary Computation ,Canbella, Australia. pp. 215-220, 2003 .

[Monmarché,99]: N. Monmarché . "On data clustering with artificial ants". In the Workshop on Data Mining with Evolutionary Algorithms, pp 23-26, 1999.

[Nasaroui et al, 02] : O. Nasaroui, D. Dasgupta, et F. Gonzalez. The fuzzy artificial immune system: Motivations, basic concepts, and application to clustering and web profiling. In Proceedings of the IEEE International Conference on Fuzzy Systems at WCCI, pp 711–716, 2002.

[Ng et al,94]:R.Ng and J.Han,. "Efficient and Effective Clustering Method for Spatial Data Mining". In the Proceedings of International Conference on Very Large Data Bases (VLDB'94), pp. 144-155,1994.

[Nielsen,00] :M.A. Nielsen et I. L. Chuang, “Quantum Communication and Quantum Information”, Cambridge University Press, 2000.

[Omran,05]: M.G.H.Omran. "Partical Swarm optimisation methods for pattern recognition and image processing". Thèse de Doctorat, Université de Pretoria, 2005.

[Pedrycz,05]:W. Pedrycz. "Clustering and Fuzzy Clustering". chapitre1, In Knowledge Based Clustering, Wiley & Sons, Hoboken , 2005.

[Poulin,01]:D. Poulin. "Classicalité du calcul quantique".Thèse de Maître en science en Physique.Université de Montréal, 2001.

[Prim,57]: R. C. Prim, "Shortest Connection Networks and some Generalizations". Bell Systems Technology Journal 36, pages 1389-1401, 1957.

[Raghavan et al, 79]: V.V. Raghavan et K. Birchard. "A clustering strategy based on a formalism of the reproductive process in natural systems". In the Proceedings of the Second International Conference on Information Storage and Retrieval, pp 10–22,1979.

[Rand,71]: W.Rand.Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association, Vol 66 n336, pp 846-850, 1971.

[Rieffel et al,00]:E. G. Rieffel and W. Polak. "An Introduction to Quantum Computing for Non-Physicists" . Los Alamos ArXiv e-print, 1998. Également publié dans ACM Computing Surveys, Vol. 32(3), pp. 300 - 335, 2000.

[Rijsbergen,79]: C. van Rijsbergen. Information retrieval, second edition. Butterworths, 1979.

[Saint-Paul,05]:R. Saint-Paul. "Une architecture pour le résumé en ligne de données relationnelles et ses applications ",Université de Nantes,2005.

[Sander et al,98] :J.Sander , M.Ester, H.P. Kriegel, X. Xu. "Density-Based Clustering in Spatial Data sets: The Algorithm GDBSCAN and Its Applications". Data Mining and Knowledge Discovery 2, Kluwer Academic Publishers ,169–194 ,1998.

[Schoeb,99] : A. Schoeb. "Analyse et comparaison de protocoles de purification de l'intrication quantique". Thèse de Maître en science en informatique, Université de Montréal, 1999.

[Sheikholeslami et al,98] :G. Sheikholeslami, S.Chatterjee et A. Zhang . "Wavecluster: A multi-resolution clustering approach for very large spatial databases". In the Proceedings of the 24th VLDB Conference, 1998.

[Shor, 94]: P W. Shor, "Algorithms for quantum computation: Discrete log and factoring", In Proceedings of the 35th Annual Symposium on Foundations of Computer Science pp. 124–134, 1994.

[Steinbach et al,03]: M. Steinbach, L. Ertöz, et V. Kumar . "Challenges of Clustering High Dimensional Data", In L. T. Wille, editor, New Vistas in Statistical Physics– Applications in Econophysics, Bioinformatics, and Pattern Recognition. Springer-Verlag, 2003.

[Tapp,99] : A. Tapp,"Informatique Quantique : Algorithmes et Complexité de la Communication".Thèse de Doctorat, Universitb de Montréal,1999.

[Timmis,01]: J. Timmis and T. Knight, "Artificial Immune Systems: Using the Immune System as Inspiration for Data Mining".In Data Mining: A Heuristic Approach. Abbas, H, Sarker, R and Newton, C, Idea Group Publishing, 2001.

[Wang et al,97]:W.Wang, J.Yang, et R.Muntz . "STING: A Ststistical Information Grid Approach to Spatial Data Mining". In Proceedings of the 23rd Conference on VLDB, 186-195, 1997.

[Weisstein,99] : E.W.Weisstein. Box-and-whisker plot. From MathWorld—A Wolfram Web Resource,1999. http://mathworld.wolfram.com/Box-and-WhiskerPlot.html.

[Wilson et al,90]: R.J.Wilson et J.J.Watkins. "Graphs : an introductory approach : a first course in discrete mathematics. John Wily and Sons, New York, NY,1990.

[Wolpert et al,97]:D.Wolpert et W.Macready. "No free lunch theorem for optimization". In IEEE Trans.Evol.Computer,pp 67-82,1997.

[Xiao et al,03]:X.Xiao, E.R.Dow, R.C.Eberhart, Z.Ben Miled, et R.J.Oppelt . "Gene clustering using self-organizing maps and particle swarm optimization". In the Proceedings of Second IEEE International Workshop on High Performance Computational Biology, Nice, France. 2003

[Xu et al ,98] : X. Xu ,M.Ester ,H.P. Kriegel et J.Sander. "A Nonparametric Clustering Algorithm for Knowledge Discovery in Large Spatial Datasets". In Proc. IEEE Int. Conf. on Data Engineering, IEEE Computer Society Press, 1998.

[Yang et al,05]:M.S.Yang et K.L.Wu . "Unsupervised possibilistic clustering". In the journal of pattern recognition society, vol 36. pp 5-21 , 2005.

[Zadeh,65]: L.A. Zadeh. Fuzzy sets, Inf. Control 8 ,pp 338–353, 1965.

[Zaïane,99] : O. R. Zaïane. "Chapter I: Introduction to Data Mining".In CMPUT690 Principles of Knowledge Discovery in Databases, 1999.

[Zaki, 04] :M.Zaki. "Mining non-redundant association rules", In international journal of data mining and knowledge discovery, pp: 223-248, 2004.

[Zhao et al,03]:Y.Zhao and G. Karypis ,"Clustering in Life Sciences". Book chapter of "Functional Genomics: Methods in Molecular Biology", Michael Brownstein and Arkady Khodursky (editors). Humana Press, 2003, (ISBN 1588292916).

[Zighed et al, 01]:D.A. Zighed, Y.Kodratoff, A.Napoli. In Bulletin AFIA'01, 2001.

[Zighed et al,02]:D. A. Zighed et R. Rakotomalala. "Extraction des connaissances à partir des données (ECD)", Techniques de l'ingénieur, HA, 2002.

Abstract

In this work, we deal with the problem of data clustering. The clustering represents a fundamental task for a great number of different fields. It is a very current method which allows a better understanding of the analyzed data set. This problem can be modelled as a problem of optimization. For its resolution, we have developed a new quantum evolutionary approach based on a quantum representation to code the space of search and quantum evolutionary strategy of search to optimize a measure of quality of cluster in order to find a good partitioning of data set. The particularity of our approach is the reduced size of the population and the reasonable number of iterations to find the best partitioning of data due to quantum operations involved in the search, which are based on principles of quantum computing as the superposition of the states, the interference and others. Another particularity lies in the initialization step where new important function is used. The other characteristic is the capacity of this approach to being extended in a simple way. Results on synthetic and real data sets are very promising and show the capacity and the efficiency of the approach to be identified valid clusters of various densities, various sizes and various forms as well as the independence between the final partition found and the initial partition.

Liste des tableaux

Liste des figures

Résumé

Dans ce travail de magistère, nous traitons le problème de clustering de données. Le clustering représente une tâche fondamentale pour un grand nombre de domaines différents. Il s'agit d'une démarche très courante qui permet de mieux comprendre l'ensemble de données analysé. Ce problème peut être modélisé comme un problème d'optimisation. Pour sa résolution, nous avons développé une nouvelle approche évolutionnaire quantique. Cette approche repose sur une représentation quantique pour coder l'espace de recherche et une stratégie de recherche évolutionnaire quantique pour optimiser une mesure de qualité de cluster afin de trouver un bon partitionnement du jeu de données. La particularité de cette approche est la taille réduite de la population et le nombre raisonnable d'itérations pour trouver le meilleur partitionnement de données grâce aux opérations quantiques impliquées dans la recherche, qui sont basées sur des principes de l'informatique quantique comme la superposition des états, l'interférence et autres. Une autre particularité réside dans la phase d'initialisation de l'approche basée sur une nouvelle fonction importante. L'autre caractéristique est la capacité de cette approche à être étendue d'une manière simple.

Les résultats sur des jeux de données synthétiques et réelles sont très prometteurs et montrent la capacité et l'efficacité de l'approche à identifier des clusters valides de différentes densités, de différentes tailles et de différentes formes ainsi que l'indépendance entre la partition finale trouvée et la partition initiale.