République algérienne démocratique et populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université du 20 AOUT 1955 SKIKDA

Faculté des sciences
Département d'informatique

Mémoire de fin d'études en vue de l'obtention du diplôme
Master Académique
Option *: Systèmes Informatiques*

THEME

La recherche De Coucou Pour Le Clustering De Donnée D'incendie En Algérie

Réalisé par :

Ramdane Selma

encadré par :

Dr Ramdane Chafika

Juin 2022

Remerciement

Je tiens avant tout à exprimer ma reconnaissance à Dr Ramdane Chafika pour avoir accepté de m'encadrer dans cette étude. Je la remercie pour son implication, son soutien et ses encouragements tout au long de ce travail.

Je souhaite également remercier Ramdane Asma pour avoir su me faire confiance et m'avoir conseillée tout au long de ces cinqs années.

Merci Dr Mansoul Abdelhak et à Dr Boulenmour Imene d'avoir accepté d'évaluer mon travail au sein du jury de soutenance.

À tous ces intervenants, je présente mes remerciements, mon respect et ma gratitude.

Dédicace

Je dédie ce Modeste travail en Première lieu à ceux qui m'ont

Soutenu durant ma vie...

A mes très chers parents **Khalfaoui Baya** *&* **Ramdane Abdelaziz** *qui ont toujours été là pour moi, Et Pour leur amour et leur sacrifices et encourage Je les aime beaucoup*

A mes chers sœurs **Wisseme, Chafika , Saoussen, Asma & Imene** *, mes frères* **Tahar & Abdenour** *pour leurs encouragements*

Aux anges de ma famille **Lina, Djawed, Amdjed, Aline**

A tous mes amies et Aya & Ghada chers amies de parcours universitaire

A ma Grande famille

A ceux qui m'aime et que j'aime...

RAMDANE Selma

Résumé :

Dans ce travail de master, 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 opté pour l'algorithme de la recherche de coucou.

Nous avons mené une série d'expérimentations sur des jeux de données réels représentant des données de maladies cancéreuses et des données d'incendies de forêts en Algérie dans la région de Bejaia et Sidi Bel-Abbas. L'objectif est d'utiliser des éléments météorologiques et des indices de forêts pour déterminer les conditions d'apparitions d'incendies ou non.

L'évaluation de résultats basée sur des mesures internes et externes a été effectuée

Mots clefs : le clustering de données, la recherche de coucou, métaheuristique.

Abstract

In this master work, we deal with the problem of data clustering. Clustering represents a fundamental task for a large number of different domains. This is a very common approach that helps to better understand the dataset being analyzed. This problem can be modeled as an optimization problem. For its resolution, we opted for the cuckoo search algorithm.

We conducted a series of experiments on real datasets representing cancer disease data and forest fire data in Algeria in the region of Bejaia and Sidi Bel-Abbas. The objective is to use meteorological elements and forest indices to determine the conditions for the appearance of fires or not. The evaluation of results based on internal and external measures has been carried out.

Keywords: data clustering, cuckoo search, metaheustic.

Liste des matières

Chapitre 3 : La recherche de coucou pour le clustering de données

Chapitre 4 : Implémentation de l'approche

Chapitre 5 : Résultats expérimentaux

Liste des tables

Liste des figures

Introduction générale

Introduction générale

Le travail présenté dans ce mémoire s'inscrit dans le vaste champ de clustering de données qui intervient dans de nombreuses activités humaines.

Le clustering de données est le processus de regroupement d'objets semblables basé sur une certaine mesure de similitude (ou dissimilitude) réciproque , il joue un rôle très important dans les applications de la fouille des données telles par exemples l'exploration des données scientifiques , la fouille des textes , les applications de bases de données spatiales , l'analyse des sites Web, le marketing , les diagnostiques médicaux , et bien entendu la bioinformatique , et beaucoup d'autres domaines .

Le problème de la classification en général est de construire une procédure permettant d'associer une classe à un objet, en classification non- supervisée ou clustering les classes (Clusters) possibles et leur nombre ne sont pas connues à l'avance, donc le but est de découvrir des relations intéressantes qui peuvent exister implicitement entre les données et qui permettant de regrouper dans un même groupe (cluster) les objets considérés comme similaires pour constituer les classes (clusters). Dans les années passées, et pour la réalisation de ce but on utilisait des techniques traditionnelles pour le clustering telles que les techniques hiérarchiques et les techniques par partitionnement mais ces derniers ne travaillent que sur un petit sous - ensemble de l'espace de recherche, pour cela d'autres méthodes de clustering ont été proposées, telles que les méthodes basées sur les métaheuristiques qui modélisent le problème de clustering comme un problème d'optimisation.
Dans le cadre de ce travail de master, nous allons opter pour la métaheuristique la recherche de coucou. L'objectif de notre travail est triple. Dans un premier lieu, il s'agit d'explorer le domaine de clustering des données et de voir les différentes techniques qui existent. Dans un second lieu, il s'agit de mener une étude de l'algorithme de la recherche de coucou inspirée du mode de reproduction de certaines espèces de coucous.

En troisième lieu, il s'agit d'appliquer l'algorithme de la recherche de coucou et de voir son apport sur deux types de données. Des données de maladies cancéreuses et des données d'incendies de forêts en Algérie dans la région de Bejaia et Sidi Bel-abbas. L'objectif est d'utiliser des éléments météorologiques et des indices de forets pour déterminer les conditions d'apparitions d'incendies ou non.

A la fin, nous allons présenter une série d'expérimentations, de tests, de résultats et d'analyses.

Le présent mémoire est organisé en cinq chapitres :

Le premier chapitre sera consacré à une présentation de l'état de l'art du clustering de données, ses méthodes et les mesures de validité et ses différentes techniques.

Le deuxième chapitre décrit la méthode de recherche coucou. Nous commençons par décrire le principe de la méthode et ses détails et nous terminons par les domaines d'application.

Le troisième chapitre est dédié à la présentation de la conception de notre approche, dans lequel, nous présentons les diagrammes de flux de données.

Le quatrième chapitre sera consacré à l'implémentation de notre application avec des captures détaillées de l'application, nous présentons les outils de programmation, l'environnement de développement et la démarche d'exécution.

Le cinquième chapitre Ce dernier chapitre est consacré aux différentes expérimentations de l'approche et nous terminons par une analyse de résultats et une discussion.

Finalement, ce mémoire termine par la présentation d'une conclusion générale.

Chapitre 1

Le clustering de données

1.1. Introduction

L'objectif général de la classification est de pouvoir étiqueter des données en leur associant une classe. L'apprentissage automatique se propose de construire automatiquement une telle procédure de classification en se basant sur des exemples, c'est - à - dire sur un ensemble limité de données disponibles. Si les classes possibles sont connues et si les exemples sont fournis avec l'étiquette de leur classe, on parle d'apprentissage supervisé. Et dans le cas où seul des exemples sans l'étiquette sont disponibles et les classes et leurs nombres sont inconnus, on parle d'apprentissage non supervisé.

Dans ce premier chapitre notre étude est basée sur l'une des qualités de la classification non supervisé, qui est le clustering de données, d'abord on commence par les définitions et les concepts nécessaires de clustering, ensuite on présente les différentes techniques principales avec les différents types de mesures de validité des clusters.

1.2. Définition de clustering

Les premiers papiers sur le clustering ont été rédiges dans les années 60.On parlait alors principalement de partage des ressources sur le réseau. Mais c'est au début des années 90, avec les développements des interfaces réseau à haut - debit et des normes logicielles, que le concept de clustering a prit toute son importance. Le clustering consiste à classer des objets en différents groupes. Il s'agit donc de partitionner un ensemble de données en sous - ensembles (des clusters), de telle sorte que les données au sein d'un cluster partagent certaines caractéristiques (telles que la proximité selon une certaine mesure de distance). Les techniques de clustering sont utilisées dans des nombreux domaines tels que la machine Learning, l'analyse de données, le processus d'apprentissage non supervisé dans le domaine de la reconnaissance de formes ou la bioinformatique.[Bouria, kachid, Karboua, 08]

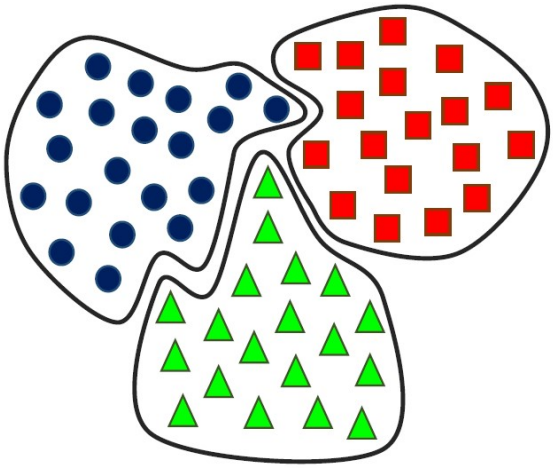


Figure1.1. Clustering.

1.3. Objectifs de Clustering :

Mirkin [B. Mirkin, 17] a identifié une liste d'objectifs de clustering :

- ❖ Structuration : représentation des données comme un ensemble de groupes d'objets similaires (généralement, la structuration est l'objectif principal du clustering).
- ❖ Description : des clusters en fonctions des caractéristiques.
- ❖ Association : la découverte des interrelations entre différents aspects du phénomène.
- ❖ Généralisation : un relevé de données sur les propriétés du phénomène que les données ont des relations avec elles, cet objectif nécessite une analyse multi-étage des objectifs précédents.
- ❖ Visualisation : représentation des structures des clusters comme une image visuelle.

1.4. Définition d'un cluster :

Le cluster n'a pas de définition précise, les plus couramment utilisés sont les suivants [Berkhin,02] [Kumar,00]

A. Définition d'un cluster bien séparé :

Un cluster est un ensemble de points tel que chaque point du cluster est plus proche (ou plus similaire) de tout autre point du cluster que de tout point extérieur au cluster. Un seuil est parfois utilisé pour indiquer que tous les points d'un cluster doivent être suffisamment proches (ou similaires) les uns des autres (Figure 1.2).

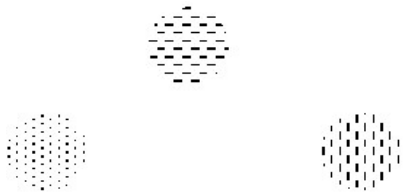


Figure 1.2. Trois clusters bien séparés.

B. Définition de cluster basée 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 tout autre cluster. Le centre d'un cluster est généralement un centroïde, la moyenne de tous les points du cluster, ou un médoïde, le point le plus représentatif d'un cluster; Cependant, de nombreux algorithmes de clustering utilisent cette définition (Figure 1.3).

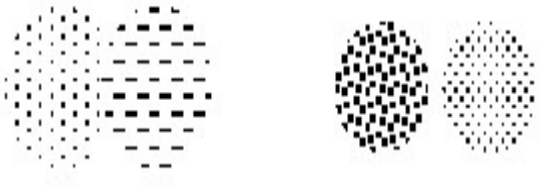


Figure 1.3. Quatre clusters basés sur le centre.

C. Définition d'un cluster contigu (plus proche voisin ou cluster transitif) :

Un cluster est un ensemble de points tel qu'un point d'un cluster est plus proche (ou plus similaire) d'un ou plusieurs points du cluster que tout autre point n'est pas situé dans le cluster (Figure 1.4).

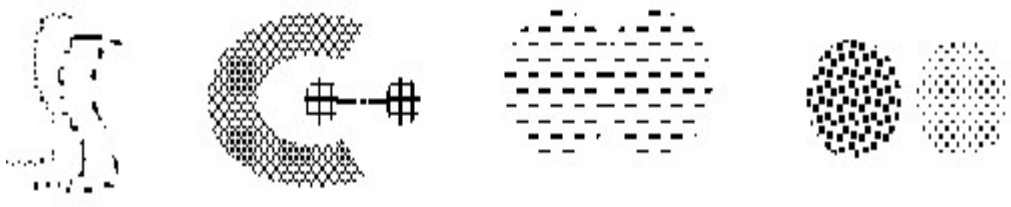


Figure 1.4. Huit clusters contigus.

D. Définition basée sur la densité :

Un cluster est une région dense de points séparés des autres régions à haute densité par des régions à faible densité. Cette définition est souvent utilisée lorsque les groupes sont irréguliers ou imbriqués et lorsqu'il y a du bruit (Figure 1.5).

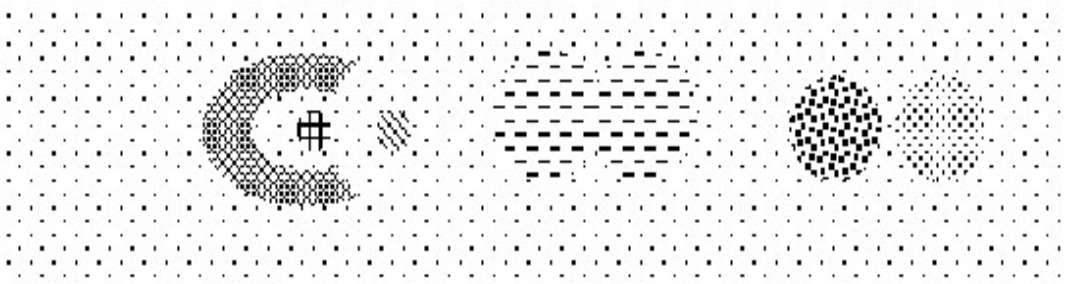


Figure 1.5. Six clusters denses.

E. Définition de Cluster conceptuels :

Se sont des *clusters qui partagent des* propriétés entre eux (Figure 1.6). [Bouria, kachid, Karboua, 08]

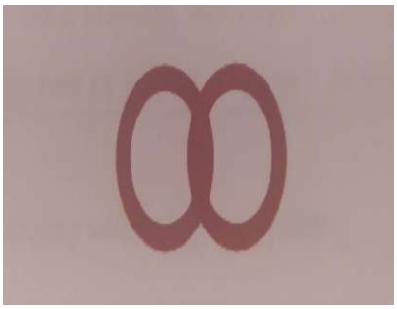


Figure 1.6. Deux cercles superposés.

1.5. Les Concepts De Clustering :

1.5.1. La matrice de données :

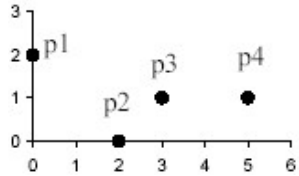
Objets (échantillons, mesures, modèles, événements) sont généralement représentés sous forme de points (vecteurs) dans un espace multidimensionnel, chaque dimension représentant un attribut différent (variable, mesure) qui décrit l'objet. Ainsi, un ensemble d'objets est représenté sous la forme d'une matrice n,×m avec m lignes, une pour chaque objet, et n colonnes, une pour chaque attribut. Ce tableau est appelé matrice de données ou jeu de données.

(La figure 1.7) ci-dessous montre un exemple concret de certains points et leur matrice de données correspondante.

1.5.2. La Matrice De Proximité :

Plusieurs algorithmes de clustering utilisent la matrice de données d'origine, et beaucoup d'autres utilisent une matrice de similarité ou une matrice de dissemblance. Par souci de simplicité, les deux matrices sont appelées matrice de proximité P. Une matrice de voisinage P m contenant toutes les×est une matrice m différences ou similitudes entre les objets considérés. Si pi et pj sont respectivement les ième et jième objets, alors l'entrée dans la ième ligne et la jième colonne de la matrice de proximité est la similarité ou la différence entre pi et pj.

La figure1.7 montre chacun quatre points, leur matrice de données et leur matrice de 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 1.7. Quatre points, leur matrice de données et leur matrice de proximité.

1.5.3. Types de données :

Les variables d'un objet peuvent être continues, discrètes, ou binaires [M. R. Anderberg, 24]:

1. **Les variables continues** : ont une plage infinie et innombrable de valeurs (les variables réelles), par exemple la surface d'un trapèze ou d'un triangle.
2. **Les variables discrètes** : ont généralement une plage finie de valeurs, ou dénombrable infinie (les variables entières) par exemple le nombre de mots dans un article.
3. Les **variables binaires** : sont des variables discrètes qui peuvent prendre seulement deux valeurs, par exemple l'existence d'une chose, elle existe ou non.

Clifford et *Stephenson* [C. a. Stephenson, 25] ont développé les trois catégories précédentes en six autres catégories plus détaillées.

1.5.4. Echelles de données :

Deux grandes échelles existent : les échelles qualitatives (nominale et ordinale) et les échelles quantitatives (intervalle et ratio) [C. Romesburg, 26], [A. K. J. R C. Dubes, 19]:

1. **L'échelle nominale :** est une échelle faible (dénuée de sens), par exemple dans les réponses (oui ou non) ou l'égalité de deux objets ($x=y, x\neq y$), elles peuvent être codées par (0 et 1), les différentes couleurs des roses sont aussi un autre exemple de cette échelle (rouge=0, bleu=1, blanc=2, et ainsi de suite).
2. **L'échelle ordinale :** cette échelle est aussi faible, les nombres n'ont pas de sens sauf s'ils sont en relation avec un autre, par exemple les niveaux de difficulté (facile=1, moyen=2, et difficile=3), la différence entre cette échelle et l'échelle nominale est que les valeurs de cette échelle reflètent un ordre.
3. **L'échelle d'intervalle** : cette échelle donne un sens à la différence entre les objets. Une unité de mesure existe et l'interprétation des nombres dépend de cette unité, par exemple la température mesurée en Celsius (40° est plus grande que 25° par 15°).
4. **L'échelle de rapport :**(ratio) est l'échelle la plus forte dans laquelle les nombres ont un sens absolu, la différence entre cette échelle et la précédente, est que cette échelle possède un zéro

absolu intrinsèquement immanent de la réalité, tandis que le zéro de l'autre échelle est défini arbitrairement. Le poids est un exemple de cette échelle, le 0 kg est la limite minimale qui n'est pas arbitraire (l'absence de tout poids).

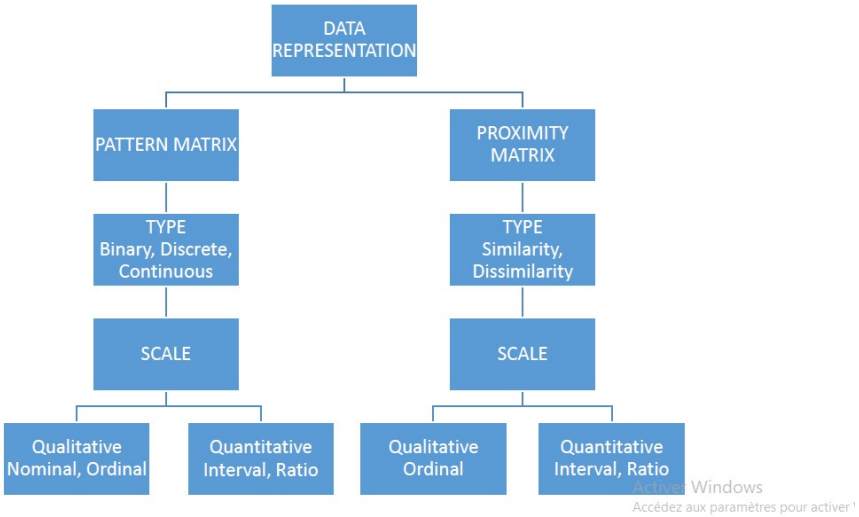


Figure 1.8. Format, types, et échelle de données.

1.5.5. Distance et similarité :

Le concept de similarité ou de dissimilarité est le composant essentiel de n'importe quelle forme du clustering.

Il existe trois concepts de similarité en clustering : [Bouria, kachid, Karboua, 08]

1. **La similarité entre objets** : à maximiser pour deux objets appartenant au même cluster, et à minimiser pour deux objets appartenant à des clusters différents.
2. **la similarité entre un objet et un cluster** : à maximiser pour une bonne cohésion interne des clusters.
3. **la similarité entre clusters** : à minimiser pour une bonne isolation externe des clusters.

La distance est la mesure la plus utilisée parmi les types de mesures de similarité et de dissimilarité qui permet de calculer l'éloignement ou la proximité entre deux points de données x.y.

1.5.5.1. Propriété de la distance :

La positivité : $d_{i,j} = d(x_i,x_j) \geq 0$.

La réflexivité : $d_{i,j} = 0 \Leftrightarrow x_i = x_j$.

L'identité : $d_{i,j}=0$.

La symétrie : $d_{i,j} = d_{j,i}$.

L'inégalité triangulaire : $d_{i,k} + d_{k,j} \geq d_{i,j}$,Tel que $d_{i,j}$ une distance dans R^p entre deux objets x_i,x_j

1.5.5.2. Quelque Fonction de distance :

Les mesures de distance les plus courantes entre deux objets x_i et x_j sont :

Distance Euclidienne $$d(x_i, x_j) = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i - x_j)^2} \quad (1.1)$$

Distance de Hamming $$d(x_i, x_j) = \sum_{i=1}^{n}\sum_{j=1}^{n}\left|x_i - x_j\right| \quad (1.2)$$

Distance de Chebyschev $$d(x_i, x_j) = \max_{\substack{i=1,2,\ldots n \\ j=1,2\ldots n}}\left|x_i - x_j\right| \quad (1.3)$$

Distance de Minkowski $$d(x_i, x_j) = \sqrt[p]{\sum_{i=1,}^{n}\sum_{j=1}^{n}(x_i - x_j)^2}, p > 0 \quad (1.4)$$

Distance de Canberra $$d(x_i, x_j) = \sum_{i=1}^{n}\sum_{j=1}^{n}\frac{\left|x_i - x_j\right|}{x_i + x_j} \quad x_i \text{ et } x_j \text{ sont positifs} \quad (1.5)$$

Séparation Angulaire $$d(x_i, x_j) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}x_i x_j}{\left[\sum_{i=1}^{n}x_i^{\,2}\sum_{j=1}^{n}x_j^{\,2}\right]^{1/2}} \quad (1.6)$$

1.6. Les Différentes Techniques De Clustering :

Plusieurs techniques de clustering sont apparues dans la littérature afin de découvrir des groupes cohésifs. Elles peuvent être classifiées aux types suivants [Ramdane, 06].

1.6.1. Clustering par partitionnement: (Partitional clustering) :

Les techniques par partitionnement créent une décomposition d'un seul niveau des points de données [Kumar, 00], ces méthodes construisent K partitions de données,

où chaque partition représente un cluster. Si k est le nombre désiré de clusters, alors les approches par partitionnement trouvent typiquement tous les k clusters immédiatement.

Il existe différents algorithmes à partitionnement, Nous allons décrire l'algorithme le plus connus : Kmeans [Laetitia, 03].

A. Kmeans :

La technique de clustering de Kmeans [Ramdane, 06] est basée sur la notion du centroide qui est le point de la moyenne ou la médiane d'un groupe de points.
L'algorithme de Kmeans est décrit comme suit :

1. Choisir K points initiaux qui seront les centres de K groupe (centroide).
2. Attribuer chaque point au centroide le plus proche.
3. Recalculer le centroide de chaque groupe.
4. Répéter les étapes 2 et 3 jusqu'à ce que les centroïdes ne changent pas.

Cet algorithme converge toujours vers une solution, qui est typiquement un minimum local, car la variance à l'intérieur d'un cluster ne peut que décroître ou rester stationnaire entre deux étapes successives [Laetitia, 03].
Kmeans a des propriétés importantes tels que : la complexité temporelle et l'efficacité de traitement de grand jeu de données, cependant il souffre de quelque limités et problèmes tels que :

- ❖ le résultat dépend fortement des controides initiaux.
- ❖ le nombre de cluster doit être fixé à l'avance.
- ❖ Il se termine souvent à un optimum local.
- ❖ Il est sensible au bruit.
- ❖ Il ne peut être utilisé que les données numériques.
- ❖ Il ne sait gérer des clusters proches dont les tailles sont très différentes, ni des clusters de forme allongée ou concave.

Une solution pour pallier à ces problèmes est de permettre de diviser ou concaténer les clusters résultants, typiquement un cluster est divisé quand sa variance est au-dessus d'un certain seuil pré spécifié, et deux clusters sont concaténés quand la distance entre leurs centroides est en dessous d'un autre seuil pré spécifié [Laurent, 04].

La figure 1.9 montre les étapes de l'algorithme de Kmeans.

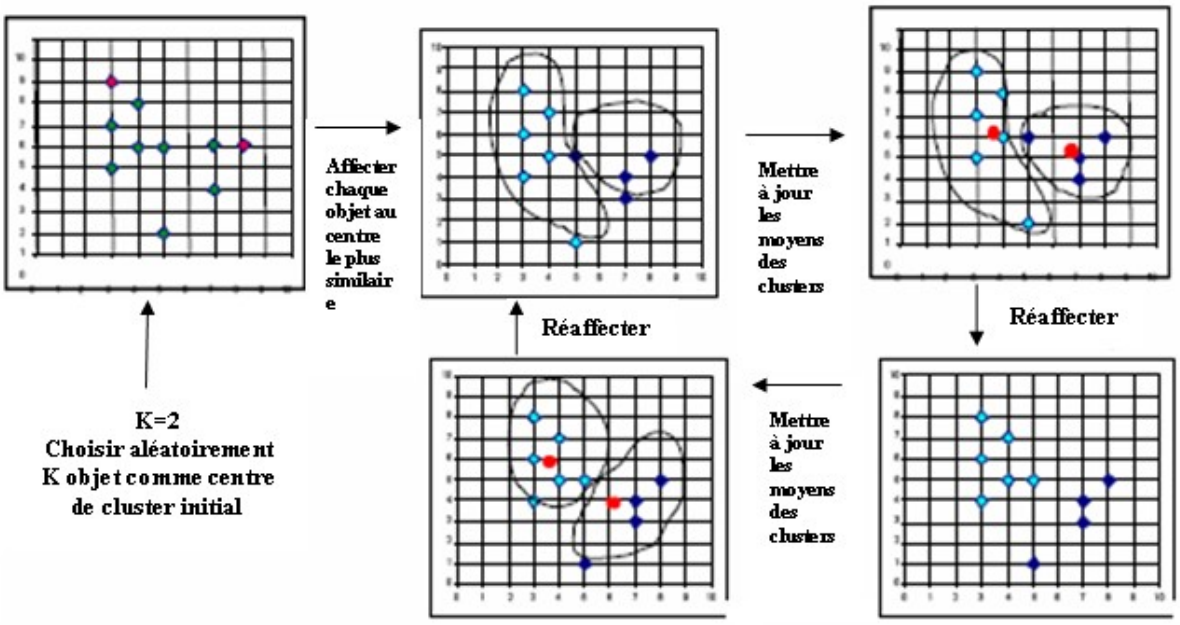


Figure 1.9. Les étapes de l'algorithme de Kmeans.

1.6.2. Clustering hiérarchique (Hierarchical clustering) :

Pour le clustering hiérarchique, le but est de produire une série hiérarchique de clusters nichés, cette hiérarchie est également connu sous le nom de dendrogramme ou, en d'autres termes, un arbre de clusters, la racine de cet arbre est formée par le cluster **X** contenant l'ensemble des points de données, et chaque nœud constitue un cluster $C_i \subset X$ d'autre part les feuilles de l'arbre correspondent aux singletons $\{x_1\}, \ldots\ldots, \{x_n\}$.

Le dendrogramme permet une visualisation de l'organisation des données et du processus de clustering car c'est un arbre inversé qui décrit l'ordre dans lequel des points sont fusionnés (vue ascendante) ou les clusters sont dédoublés (vue de haut en bas), il est possible alors d'obtenir une partition de X en coupant l'arbre a un niveau l donné. Par exemple, le choix de l = 4 dans le dendrogramme de la (figure1.9) renvoie le partitionnement suivant :

$$C = \{\{x_1, x_2, x_3\}, \{x_4, x_5, x_6\}\} \quad (1.7)$$

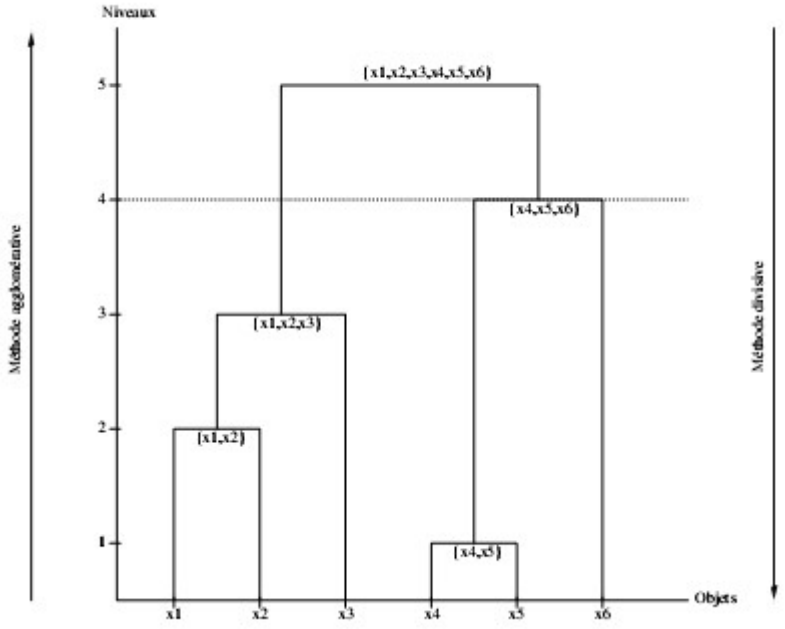


Figure1.10. Exemple d'un dendrogramme.

Ce dernier paramètre l peut être choisi relativement au nombre de clusters désiré ou à l'aide d'une analyse statistique de la qualité des différentes partitions que l'on peut extraire de l'arbre.

On distingue deux approches de base de clustering hiérarchique pour parvenir à un tel arbre hiérarchique : les algorithmes agglomératifs et divisifs.

A. Algorithmes divisifs :

Ces algorithmes commencent par la "racine" qui contient tous les objets, puis procèdent par divisions successives de chaque nœud jusqu'à obtenir des singletons (clusters d'un seul point).

Brièvement l'algorithme est comme suit: [D. P. Mercer, Linacre College, 03]

1. Choisir le cluster contenant la paire d'objets la plus éloignée. C'est le cluster avec le plus grand diamètre.
2. Dans ce cluster, enlever l'objet qui a la plus grande distance moyenne des autres objets. cet objet forme un nouveau cluster de singleton.
3. Pour l'objet h dans le cluster étant dédoublé, calculez la distance moyenne entre ce dernier et le cluster courant; et la distance moyenne entre l'objet et le nouveau cluster. Si la distance au nouveau cluster est inférieure à celle au cluster

courant, déplacez l'objet h à un nouveau cluster. Faites une boucle au-dessus de tous les objets dans le cluster.

4. Si aucun objet ne se déplace, et le nombre courant de clusters est plus grand que k, alors aller à l'étape 1. Sinon arrêter.

B. Algorithmes agglomératifs :

Un regroupement agglomératif construit l'arbre en partant des "feuilles" (singletons) et procède par fusions successives des plus proches clusters jusqu'a obtenir un unique cluster "racine", contenant l'ensemble des objets.

La méthode agglomérative est décrite comme suit: [D. P. Mercer, Linacre College, 03]

1. Considérer chaque objet comme un cluster et calculer la matrice de proximité.
2. Trouver le plus petit élément dans la matrice. Ceci correspond à la paire de clusters qui sont les plus semblables, et fusionnez ces deux clusters (soit i et h).
3. Calculer les distances entre le nouveau cluster formé et les clusters restants.
4. Supprimer la ligne et la colonne du cluster i et recouvrer la ligne et la colonne du cluster h avec les nouvelles valeurs.
5. Si le nombre courant de clusters est plus grand que k, alors aller à l'étape 2 Sinon arrêter.

Le clustering hiérarchique agglomératif a un certain nombre de limitations et problèmes tels que [Kumar, 00]:

- aucune fonction objective n'est optimisée.
- Les décisions de fusionnement sont finales.
- les bonnes décisions de fusionnement locales peuvent ne pas avoir comme de bons résultats globaux.
- Il a des problèmes avec le bruit, forme des clusters non convexes, et une tendance de diviser de grands clusters.

1.6.3. Clustering basés sur la Densité :

Dans ce type de clustering les clusters sont considérés comme des régions en haute densité qui sont séparées par des régions en faible densité. La densité est représentée par le nombre d'objets de données dans le voisinage. C'est pourquoi ces méthodes sont capables de chercher des chusters de forme arbitraire. Donc elles sont utilisées dans la classification de données spatiales ou Il y a deux approches, l'une est basée sur la connectivité de densité, l'autre est basée sur la fonction de

densité, les algorithmes les plus connus de ce type sont DBSCAN (connectivité de densité) et DENCLUE (fonction de densité). [Bouria, kachid, Karboua, 08]

1.6.4. Clustering basé sur les Grilles :

Ce type d'algorithme est conçu pour des données spatiales où on divise l'espace de données en cellules. Une cellule peut être un cube, une région , un hyper rectangle . En fait, elle est un produit cartésien de sous intervalles d'attributs de données. Avec une telle représentation des données, au lieu de faire la classification dans l'espace de données, on la fait dans l'espace spatial en utilisant des informations statistiques des points dans la cellule.[Bouria, kachid, Karboua, 08]

1.7. Métaheuristique pour le clustering :

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].

Résoudre un problème d'optimisation, c'est trouver l'optimum d'une fonction, parmi un nombre fini de choix, souvent très grand, cela se fait par les métaheuristiques qui forment un ensemble de méthodes pour résoudre ces problèmes réputés difficiles. Parmi ces méthodes il existe la méthode de recherche coucou, les algorithmes génétiques, et les algorithmes évolutionnaires qui sont contribué de façon puisant pour résoudre le problème de clustering.

1.8. Technique de validation de clustering :

L'interprétation des résultats d'un clustering est un gros problème, qui conduit à la question : qu'est-ce qu'un bon schéma de clustering ? L'objectif principal de la validation de cluster est d'évaluer le résultat du clustering pour trouver la meilleure partition de jeu de données. Il existe des approches de validité de cluster pour évaluer quantitativement le résultat d'un algorithme de clustering, mais la plupart des évaluations expérimentales d'algorithmes utilisent des ensembles de données 2D pour permettre au lecteur de vérifier visuellement la validité des résultats. Il est clair que la vue de l'ensemble de données est une vérification cruciale des résultats de clustering. Dans le cas de grands ensembles de données multidimensionnelles (par exemple, plus de trois dimensions), il serait difficile de visualiser efficacement l'ensemble de données. De plus, percevoir les clusters avec les outils de visualisation disponibles est une tâche difficile pour les personnes qui ne sont pas habituées aux

espaces à grand nombre de dimensions. Il existe deux types de mesures : les mesures de qualité externes et les mesures de qualité internes [Ramdane, 06].

1.8.1. Mesures externes :

Les mesures de qualité externes utilisent des connaissances externes, elles impliquent la comparaison d'un schéma avec une classification prédéfinie. Ces mesures renvoient ainsi à l'adéquation entre la carte obtenue et une connaissance « externe » des données (carte attendue) [Guillaume, 04]. Les mesures externes que nous allons introduire appliquent directement la connaissance des étiquettes de classe. Ils notent les clusters générés en tenant compte des classes d'appartenance correctes.

- La F-mesure :

  La F Measure est une fonction largement utilisée dans la littérature pour évaluer les algorithmes de clustering. Fmesure adopte les notions de précision et de rappel issues de la recherche documentaire. Il compare la qualité du clustering en considérant les bonnes classes connues pour un jeu de données. Soit C = (C1, C2, . .. ,Ck) un clustering donné et R = (R1,R2,...,Rk′) les classes correctes.Chaque classe Ri contient des points de données Ni, chaque cluster Cj (généré par l'algorithme) est considéré comme l'ensemble des points de données Nj. Nij indique le nombre de points de classe Ri dans le cluster Cj et N indique le nombre total de points dans le jeu de données. Pour chaque classe Ri et un cluster Cj, la précision et le rappel sont définis comme [Ramdane, 06] :

$$\text{Prec}(R_i, C_j) = \frac{N_{ij}}{N_j} \quad et \quad \text{Rep}(R_i, C_j) = \frac{N_{ij}}{N_i} \tag{1.8}$$

  Et la valeur de F-mesure correspondante est :

$$Fmes(R_i, C_j) = \frac{(b^2+1)\cdot \text{Prec}(R_i, C_j)\cdot \text{Rep}(R_i, C_j)}{b^2 \cdot \text{Prec}(R_i, C_j) + \text{Rep}(R_i, C_j)} \tag{1.9}$$

  Où des coefficients égaux de Prec(R_i,C_j) et Rep(R_i,C_j) sont obtenus lorsque b=1. La valeur globale de F-mesure F pour toute la partition est calculée comme

$$F(C) = \sum_{i=1}^{k'} \frac{N_i}{N} \max_{Ci \in C}(Fmes(R_i, C_j)) \tag{1.10}$$

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

❖ La pureté :

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

$$Pur(C_j) = \max_{Ri \in R} \frac{N_{ij}}{N_j} \tag{1.11}$$

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 ensuite calculée comme la pureté moyenne de tous les clusters. Elle est limitée à l'intervalle] 0,1] et peut être maximale [Julia, 03].

1.8.2. Mesures interne :

Les mesures de qualité interne n'utilisent pas de connaissances externes mais uniquement les données d'entrées (matrice de (dis) similarité, descriptions des données etc.) comme référence [Guillaume, 04].

Les mesures qui peuvent être appliquées dans ce type d'évaluation tentent de capturer les deux objectifs de l'analyse de cluster : minimiser la distance intra-cluster (résultant en clusters compacts) et maximiser la distance inter-cluster (résultant en clusters compacts).). . Grappes bien séparées)

❖ La variance intra cluster :

La variance intra-cluster est basée sur le concept de minimisation de la distance intra-cluster, elle est définie selon la distance entre les points d'un même cluster ou selon la distance entre les points et le centroide d'un cluster. Il est donné par :

$$Var(C) = \sum_{Ci \in C} \sum_{p_j \in C_i} d(p_j - \mu_i)^2 \tag{1.12}$$

Où pj désigne un point de données, k désigne le nombre de clusters, μi représente le centre de gravité du cluster Ci, d(.,.) est la fonction de distance

utilisée pour calculer l'écart entre les points de données pj et le centre de gravité μi [Alexandre, 06].

❖ La connectivité :

La mesure de la connectivité des clusters évalue dans quelle mesure les points de données voisins ont été placés dans la même cluster .Il est calculé à l'aide de la formule suivante : $\text{Conn} = \sum_{i=1}^{m}\left(\sum_{l=1}^{L} p_{i,nn_i(l)}\right)$, (1.13)

Où

$$p_{r,s} = \begin{cases} 1/1 & \text{if } \neg\exists c_j : r,s \in c_j \\ 0 & otherwise, \end{cases} \quad (1.14)$$

$nn_i(l)$ est le voisin le plus proche du point de données pi et L est le paramètre qui détermine le nombre de voisins contribuant à la connectivité. La connectivité doit être minimale [Ramdane, 06]

1.9. Conclusion :

Dans ce chapitre, nous avons vu les bases de clustering des données, ses différentes techniques et les critères utilisés pour valider et évaluer ces techniques. Dans le chapitre suivant, nous proposons comme méthode de recherche, l'Algorithme de la recherche de coucou.

Chapitre 2

L'algorithme de la recherche de coucou

2.1. Introduction :

De plus en plus d'algorithmes métaheuristiques modernes inspirés de la nature sont en train d'émerger et deviennent de plus en plus populaires. La puissance de presque toutes les métaheuristiques modernes vient du fait qu'elles imitent les meilleures caractéristiques de la nature, en particulier les systèmes biologiques évolués de la sélection naturelle sur des millions d'années.

Les algorithmes métaheuristiques sont souvent inspirés de la nature, et ils sont maintenant parmi les algorithmes les plus largement utilisés pour l'optimisation. Ils ont de nombreux avantages par rapport aux algorithmes classiques. Les algorithmes métaheuristiques sont très divers, y compris les algorithmes génétiques, le recuit simulé, l'évolution différentielle, les algorithmes de fourmis et d'abeilles, l'optimisation d'essaim de particules, la recherche de coucou et d'autres.

La recherche de coucou en anglais Cuckoo Search (CS) est l'un des derniers algorithmes métaheuristiques inspirés de la nature, développés en 2009 par Xin-She Yang et Suash Deb. CS est basé sur le parasitisme de couvaison de certaines espèces de coucou. De plus, cet algorithme est renforcé par les soi-disant vols de Lévy, plutôt que par de simples randonnées aléatoires isotropes.

Dans ce chapitre, nous allons introduire la recherche de coucou en détail, en commençant par son comportement de reproduction intéressante, nous allons voir également les caractéristiques des voles de Lévy et son algorithme de base ainsi que ses champs d'applications.

2.2. Le coucou dans la nature :

Figure 2.1. Oiseau coucou.

L'habitat est le milieu de vie, de reproduction et de développement d'une population d'une espèce donnée. De même pour les coucous, il constitue une source de nourriture et un lieu de reproduction. Les coucous se produisent dans une grande variété d'habitats. La majorité des espèces survivent dans les forêts et les bois, principalement dans les forêts tropicales à feuilles persistantes. En plus des forêts, certaines espèces de coucous occupent des environnements plus ouverts, ce qui peut inclure même les zones arides comme les déserts [R. Rajabioun, 2011]. A titre d'exemple, le coucou plaintif fréquente une assez grande variété d'habitats tels que les bois ouverts, les forêts secondaires, arbustes et broussailles, les champs cultivés et aussi les jardins, aussi bien en milieu rural que citadin. On peut aussi le voir dans les herbages et les marais. Le coucou gris fréquente les forêts de conifères ou de feuillus, et les zones boisées en général, les espaces boisés ouverts, les lisières de forêts et les clairières, les steppes arborées, les prairies, les marais et les roselières, ainsi que les zones cultivées avec des arbres et des buissons. Le coucou geai fréquente des habitats semi-arides tels que les zones boisées ouvertes (surtout avec des acacias et des arbustes épineux), les contreforts rocheux des collines dans les savanes sèches, et les zones agricoles sèches avec des arbres et des buissons. On le voit souvent voler au-dessus des espaces découverts. Selon la distribution, et particulièrement en Europe, on peut le trouver dans les landes de bruyères avec du chêne-liège et des conifères du genre Pinus pinea. Il fréquente également les bosquets et les plantations d'oliviers.

2.3. Sources d'inspiration :

Comportement de coucou : La nature innove, invente, teste, valide, améliore et diversifie les systèmes vivants depuis des centaines de millions d'années, elle a été toujours une source d'inspiration.

Plusieurs questions préoccupent les biologistes : dans un groupe d'oiseaux, pourquoi le groupe est-il souvent cohérent alors que chaque individu semble autonome? Comment les activités de tous les individus sont elles coordonnées sans supervision? Les éthologistes qui étudient le comportement de groupes d'oiseaux ou d'autres animaux ou même insectes observent que la coopération entre les éléments de groupe est auto-organisée : souvent, elle résulte d'interactions entre les individus. Bien que ces interactions puissent être simples, elles permettent à la collectivité de résoudre des problèmes difficiles.

Dans cette optique, un comportement attirant observé chez les oiseaux coucou a été investigué et exploité et il a mené à l'apparition de l'algorithme de la recherche coucou.

Coucou sont des oiseaux fascinants, non seulement à cause des beaux sons qu'ils peuvent faire, mais aussi en raison de leur stratégie de reproduction agressive. Certaines espèces comme les coucou Ani et Guira pondent leurs œufs dans des nids communautaires, bien qu'ils peuvent retirer les œufs des autres pour augmenter la probabilité d'éclosion de leurs œufs [Payne R. B, Sorenson M. D, and Klitz K, 2005].

Un certain nombre d'espèces s'engagent au parasitisme obligatoire des couvées en pondant leurs œufs dans les nids d'autres oiseaux hôtes (souvent d'autres espèces).

Il existe trois types fondamentaux de parasitisme de couvée: le parasitisme intraspécifique de couvée, l'élevage en coopération et la prise de nid.

Certains oiseaux hôtes peuvent entrer en conflit direct avec les coucous intrus. Si un oiseau hôte découvre que les œufs ne lui appartiennent pas, il jettera ces œufs ou abandonnera simplement son nid et construira un nouveau nid ailleurs.

Quelques espèces de coucou comme Tapera, ont évolué de telle manière que les coucous femelles sont souvent très spécialisés dans la mimique des couleurs et les motifs des œufs de quelques espèces hôtes choisies [Payne R. B, Sorenson M. D, and Klitz K, 2005]. Cela réduit la probabilité que leurs œufs soient abandonnés et augmente ainsi leur reproductivité.

En outre, le moment de la ponte de certaines espèces est également étonnant. Les coucous parasites choisissent souvent un nid où l'oiseau hôte vient de pondre ses propres œufs.

En général, les œufs de coucou éclosent légèrement plus tôt que leurs œufs hôtes. Une fois que le premier poussin coucou est éclos, la première action instinctive qu'il prendra est d'expulser les œufs hôtes en propulsant aveuglément les œufs hors du nid, ce qui augmente la part de nourriture du poussin coucou fournie par son oiseau hôte. Des études montrent également qu'un poussin coucou peut également imiter l'appel des poussins hôtes pour accéder à plus de possibilités d'alimentation (figure 2.2).

Figure 2.2. Un poussin coucou expulse les œufs**.**

2.4. Description de la recherche du coucou (CS) :

Le comportement de parasitisme de couvées chez les coucous est combiné dans l'algorithme de la recherche coucou avec les vols de Lévy pour améliorer la recherche d'un nouveau nid. Les vols de Lévy (figure 3.3), baptisées par le mathématicien français Paul Lévy, représente un modèle des marches aléatoires caractérisées par leurs longueurs de pas qui suivent une distribution de loi de puissance qui s'écrit de la forme suivante :[Bak ,1997].

$$N(s) = s^{-t} \quad (2.1)$$

Dans la nature, les animaux cherchent de la nourriture de manière aléatoire ou quasi-aléatoire. En général, la recherche de la nourriture est effectivement une marche aléatoire parce que le prochain mouvement est basé sur l'emplacement/état actuel et la probabilité de transition à l'emplacement suivant. La direction qu'ils choisissent dépend implicitement d'une probabilité qui peut être modélisée mathématiquement. Différents études ont montré que le comportement de vol de nombreux oiseaux et insectes a les caractéristiques typiques des vols de Lévy [yang et deb, 2010]. D'autre part Reynolds et Frye[Reynolds et Frye,2007] ont montré que les mouches de fruits "Drosophila melanogaster" explorent leur paysage en utilisant une série de trajectoires droites ponctuées par un brusque virage à 90°, ce qui conduit à un modèle de recherche intermittente sans échelle d'un style de vol de Lévy. Ce modèle est communément représenté par de petits pas aléatoires suivis à long terme par de grands sauts [Ouaarab, 2015]. Un tel comportement de vol de Lévy a été appliqué à l'optimisation et les résultats ont montré une capacité prometteuse.

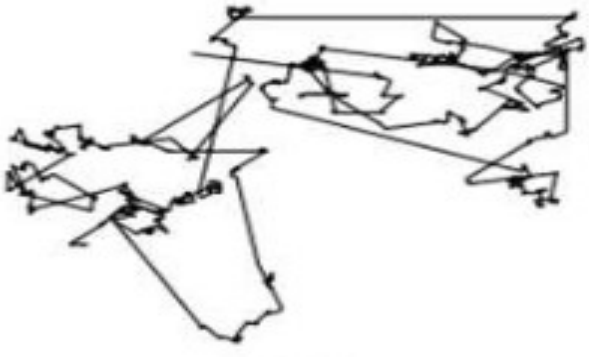


Figure 2.3. Levy flight**.**

2.5. Les principes de bases de l'algorithme de la recherche coucou (CS) :

La métaheuristique de la recherche coucou CS développée par Xin-She Yang et Suash Deb en 2009 prend comme une base les idées suivantes :

- Chaque coucou pond un œuf à la fois et choisit un nid aléatoirement.
- Un bon nid de bonne qualité peut passer vers une nouvelle génération.
- Le nombre de nids hôtes est fixé, et un œuf posé par un coucou peut être découvert par l'oiseau hôte suivant une probabilité $p_\alpha \in [0, 1]$.

Dans ce cas, l'oiseau hôte peut jeter l'œuf ou abandonner le nid et construire un nid complètement nouveau. Pour simplifier, cette dernière hypothèse peut être approchée par la probabilité p_α de n nids d'être remplacés par de nouveaux nids.

L'algorithme de la recherche du coucou se résume autour des règles suivantes :

- Chaque œuf du coucou dans un nid représente une solution.
- Chaque oiseau de coucou pondra un seul œuf à la fois, et choisira sont nid de façon "aléatoire". Donc, chaque individu de la population des coucous a le droit de générer aléatoirement une seule nouvelle solution.
- Les meilleurs nids de meilleure qualité d'œufs nous mèneront vers les nouvelles générations. Ici, on a introduit implicitement la notion d'intensification ou la recherche autour des meilleures solutions.
- Certaines nouvelles solutions doivent être générées par les vols du Lévy autour de la meilleure solution obtenue jusqu'ici. Cela accélérera la recherche locale.
- Le nombre de nids hôtes est fixe, et l'œuf pondu par l'oiseau est découvert par l'hôte avec une probabilité $p_\alpha \in [0, 1]$. Dans ce cas, l'oiseau hôte choisi de se débarrasser de l'œuf, ou d'abandonner le nid et de reconstruire un autre nid quelque part. Pour la simplification, cette dernière hypothèse sera approximée par la fraction pα des n nids qui sont remplacés par des nouveaux (nouvelles solutions aléatoires).
- Une fraction importante de nouvelles solutions doivent êtres générées par randomisation vers des régions lointaines et dont les emplacements doivent être assez loin de la meilleure solution actuelle, ce qui fera que le système ne sera pas pris au piège dans un optimum local.

2.6. Pseudo-code de l'algorithme de la Recherche Coucou par le vol de Lévy :

Les étapes de base de la recherche Coucou (CS) peuvent être résumées par le code pseudo illustré ci-dessou :

Début

fonction objective f (x), x = (x 1, ..., X d) T

Générer la population initiale de n de nids hôtes xi (i = 1, 2, ..., n)

tant que (T <MaxGeneration) ou (critère d'arrêt)

1. Obtenez un coucou (i) aléatoirement par le vol de Lévy
2. évaluer sa qualité/ *fitness Fi*
3. Choisissez un nid parmi n (par exemple, j) aléatoirement
4. *T=T+1*

si (Fi > *Fj*), // les nouvelles solutions de nid j dominent ceux de nid i

Remplacer j par la nouvelle solution;

Fin si

Une fraction (P_a) des nids les plus mauvais est abandonnée et de nouveaux sont construites;

Gardez les meilleures solutions (ou des nids avec des solutions de qualité);

Triez les solutions et trouver ci le meilleur courant

fin tant que

Post-traitement des résultats et la visualisation

Fin

Figure 2.4. L'algorithme de la recherche de coucou.

Un coucou *i* génère une nouvelle solution x^{t+1} via les vols de Lévy, selon l'équation :

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus \text{Lévy}(\lambda), \tag{2.2}$$

où α est la longueur du pas suivant la distribution des vols de Lévy montrée dans l'équation 3.3:

$$\text{Lévy} \sim u = t^{-\lambda}, \quad (1 < \lambda \leq 3), \tag{2.3}$$

Cette équation est stochastique (une marche aléatoire). En général, une marche aléatoire est une chaine de Markov dont l'étape suivante ne dépend que de l'étape actuelle, qui est le premier terme de l'équation, suivi par la probabilité de transition qui est le deuxième terme. Le produit ⊕ représente le produit matriciel. *A* est la longueur maximale du pas qui devrait être liée à l'échelle de l'espace de recherche du problème. Dans ce cas, $\alpha = 1$.

La marche aléatoire par le biais des vols de Lévy est plus efficace dans l'exploration de l'espace de recherche, que les autres marches aléatoires, vu que la taille de son pas est beaucoup plus grande à long terme.

2.7. Efficacité de la recherche coucou (comparaison avec d'autres métaheuristiques):

La faculté d'équilibrer la recherche entre l'exploitation des zones prometteuses et l'exploration de l'espace de recherche à l'échelle globale, est un indice de performance lié à toutes les métaheuristiques. La métaheuristique la plus robuste est celle qui balance parfaitement entre l'intensification et la diversification. CS a un meilleur contrôle d'équilibre de la stratégie de recherche intensive locale et une exploration plus efficace de l'ensemble de l'espace de recherche. Aussi, le nombre réduit de paramètres fait de CS un algorithme moins complexe et donc potentiellement plus générique.

CS utilise un nombre réduit de paramètres de contrôle. En effet, CS utilise deux paramètres, la taille de la population n et la portion des nids abandonnés p_a. En principe, n est fixé et p_a essentiellement contrôle l'élitisme et l'équilibre entre la randomisation et la recherche locale. Le nombre réduit de paramètres rend l'algorithme moins complexe et donc potentiellement plus générique. Tout ceci est expliqué par la facilité de régler les deux paramètres de CS tout en gardant sa robustesse de résoudre une large gamme de problèmes d'optimisation même les problèmes qualifiés NP-difficiles.

La comparaison de CS et de PSO et GA a montré que CS surpasse les performances de ces métaheuristiques. Effectivement, une étude analytique [Yang, 2014] a montré que DE, PSO et SA sont des cas particuliers de la recherche de coucou et il a été aussi montré que la recherche de coucou a une convergence globale.

Des études théoriques sur l'optimisation des essaims de particules ont suggéré que la PSO peut converger rapidement vers la meilleure solution actuelle, mais pas nécessairement les meilleures solutions globales. En fait, certains analystes suggèrent que les équations de mise à jour des PSO ne satisferont pas aux conditions de convergence globale, et donc il n'y a aucune garantie pour la convergence globale. D'autre part, il a prouvé que la recherche du coucou satisfait aux exigences de convergence globale et à ainsi garantit des propriétés. Cela implique que pour l'optimisation multimodale, la PSO peut converger prématurément vers un optimum local, alors que la recherche de coucou peut généralement converger vers l'optimalité globale.

Un autre avantage de la recherche de coucou est que sa recherche globale utilise des vols Lévy, plutôt que de la randonnée aléatoire standard. Comme les vols de Lévy ont une moyenne et une variance infinies, CS peut explorer l'espace de recherche plus efficacement que les algorithmes utilisant des processus gaussiens standard. Cet avantage, conjugué à la fois aux capacités locales et de recherche et à la convergence globale garantie, rend la recherche de coucou très efficace. En effet, diverses études et applications ont démontré que la recherche du coucou est très efficace.

2.8. Applications de la recherche coucou :

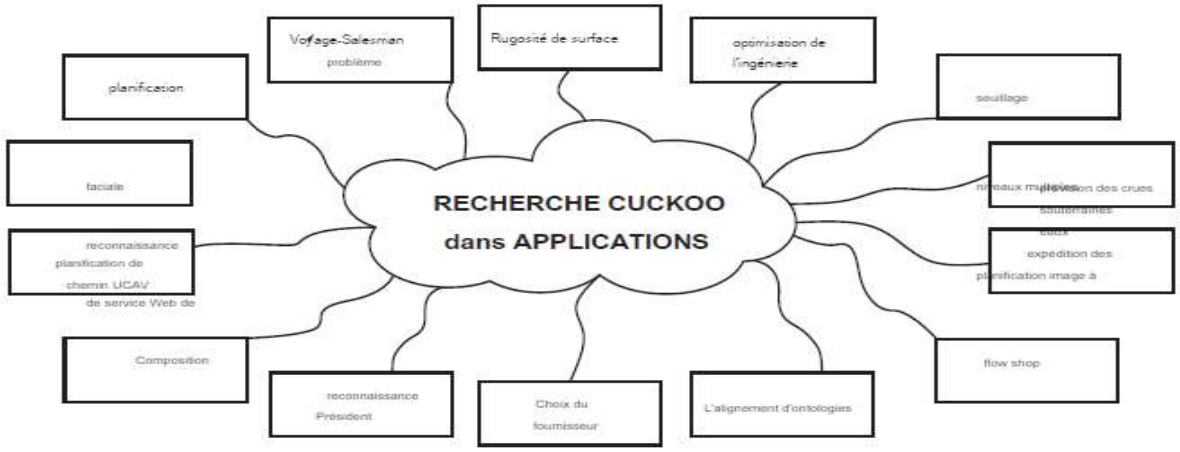


Figure 2.5. Applications de la Recherche de coucou.

Évidemment, l'optimisation de l'ingénierie fait partie des diverses applications. En fait, la
La recherche de coucou et ses variantes ont été appliquées dans presque tous les domaines des sciences, de l'ingénierie et de l'industrie. Certaines des études d'application sont résumées dans le **Tableau 2.1**

Application	Auteur
Seuil d'image multi niveau	Brajevic et al.
Prévision des inondations	Chaowanawatee and Heednacram
Réseaux de capteurs sans fil	Dhivya and Sundarambal
La fusion des données	Dhivya et al.

Cluster dans les réseaux sans fil	Dhivya et al.
Clustering	Goel et al.
Expédition des eaux souterraines	Gupta et al.
Choix du fournisseur	Kanagaraj et al.
Prévision de charge	Kavousi-Fard and Kavousi-Fard
Rugosité de surface	Madic et al.
Planification de magasin de flux	Marichelvam
Remplacement optimal	Mellal et al.
Allocation DG dans le réseau	Moravej and Akhlaghi
Optimisation du filtre Floraison	Natarajan et al.
Réseau neuronal BPNN	Nawi et al.
Problème Voyageur de commerce	Ouaarab et al.
Composition du service Web	Pop et al.
Composition du service Web	Chifu et al.
Correspondance ontologique	Ritze and Paulheim
Reconnaissance des locuteurs	Sood and Kaur
Tests automatisés de logiciels	Srivastava et al.
Optimisation de fabrication	Syberfeldt and Lidberg
Reconnaissance de visage	Tiwari
Formation des modèles neuronaux	Vázquez
Expédition économique non convexe	Vo et al.
Planification des trajets UCAV	Wang et al.

Optimisation des activités	Yang et al.
Sélection des paramètres d'usinage	Yildiz
Planification des travaux en grille	Prakash et al.
Affectation quadratique	Dejam et al.
Problème d'emboîtement de feuilles	Elkeran
Optimisation des requêtes	Joshi and Srivastava
Puzzle des N reines	Sharma and Keswani
Jeux d'ordinateur	Speed

Tableau 2.1. Applications de la recherche coucou.

2.9. Conclusion :

Les algorithmes basés sur l'intelligence d'essaim comme la recherche de coucou sont très efficaces pour résoudre un large éventail de problèmes d'optimisation et ont donc diverses applications en sciences et en ingénierie. Certains algorithmes (par exemple, la recherche de coucou) peuvent avoir une très bonne convergence globale. Cependant, il reste encore des questions difficiles à résoudre dans les études futures. L'un des principaux problèmes est qu'il existe un écart important entre la théorie et la pratique. La plupart des algorithmes métaheuristiques ont de bonnes applications dans la pratique, mais l'analyse mathématique de ces algorithmes manque loin derrière. En fait, en dehors de quelques résultats limités sur la convergence et la stabilité sur les algorithmes tels que l'essaim de particules, les algorithmes génétiques et la recherche de coucou, de nombreux algorithmes n'ont pas d'analyse théorique. Par conséquent, nous pouvons savoir qu'ils peuvent bien fonctionner dans la pratique, nous comprenons difficilement pourquoi il fonctionne et comment les améliorer avec une bonne compréhension de leurs mécanismes de travail. Une autre question importante est que tous les algorithmes métaheuristiques ont des paramètres dépendant de l'algorithme, et les valeurs réelles / le paramétrage de ces paramètres influenceront largement la performance d'un algorithme. Par conséquent, le bon paramétrage lui-même devient un problème d'optimisation. En fait, le paramétrage est un domaine important de la recherche, qui mérite plus d'attention à la recherche. En outre, même avec de très bonnes applications de nombreux algorithmes, la plupart de ces applications concernent les cas où le nombre de variables de conception est inférieur à quelques

centaines. Il serait plus avantageux pour les applications réelles si le nombre de variables peut augmenter à plusieurs milliers ou même à l'ordre de millions.

Nous allons présenter dans le chapitre prochain l'application de la méthode de recherche coucou sur le clustering des données.

Chapitre 3

La Recherche de Coucou Pour Le Clustering De Données

3.1. Introduction :

Après avoir présenté dans le chapitre précédent les notions et les concepts de base de la recherche coucou, nous allons présenter dans ce chapitre son adaptation au problème de clustering.

Nous commençons par la description de cette adaptation, nous passons ensuite à la présentation de l'objectif de notre application ainsi que la mise en évidence de l'architecture générale de notre système et la description de ses modules qui constitue une étape fondamentale qui précède l'implémentation, nous détaillons, également, les différents diagrammes et scénarios à implémenter dans la phase suivante. Ceci permettra de mieux comprendre notre application.

Pour cela nous allons utiliser les diagrammes de flux de données pour la modélisation de notre application.

3.2. Clustering basé sur la Recherche Coucou :

La recherche de Coucou (CS) est l'un des algorithmes metaheuristiques. Cet algorithme fonctionne sur la base de la stratégie de reproduction de l'oiseau coucou.

Notre travail représente un algorithme de clustering de données basé sur l'optimisation par la recherche de coucou. La recherche de coucou est générique et robuste pour de nombreux problèmes d'optimisation et dispose de fonctionnalités intéressantes.

Pour résoudre le problème de clustering de données, l'algorithme de recherche de coucou est adapté pour atteindre les centroïdes des clusters. Pour faire cela, nous supposons que nous avons $\boldsymbol{n}$ objets (données) et chaque objet est défini par $\boldsymbol{m}$ attributs. Dans notre travail, l'objectif principale du CS est de trouver $\boldsymbol{k}$ centroïdes des clusters qui réduisent au maximum la fonction objectif dénotée par **SSE** qui représente une mesure de qualité interne, elle permet de calculer la distance intra cluster (la somme des distances entre les points et leur centroïde de cluster correspondant), elle est définie par l'équation (3.1). Sachant que l'ensemble de données (jeu de données) doit être représenté par une matrice (n,m), tel que $\boldsymbol{n}$ représente le nombre d'objets (le nombre de points de données) et $\boldsymbol{m}$ représente le nombre des attributs (le nombre de dimensions).

$$SSE = \sum_{i=1}^{k} \sum_{j=1}^{n} W_{ij} * \sqrt{\sum_{p=1}^{m} (o_{jp} - c_{ip})^2} \qquad (3.1)$$

Où $\boldsymbol{W_{ij} = 1}$ si l'objet est dans le cluster et **0** sinon, **k** est nombre de cluster, **n** est le nombre d'objet, **m** est le nombre des attributs, et $\boldsymbol{c_{ip}}$ est la valeur de l'attribut numéro **p** du centroïdes de cluster numéro **i** [Ishak et al, 14].

Dans le mécanisme de recherche de coucou, les nids sont des solutions et dans notre travail une solution est un ensemble de centroïdes représenté par un vecteur de dimension **k*m** (où **k** est le nombre de centroïdes des clusters et **m** représentent le nombre d'attributs).

3.3. Les étapes principales de notre système :

L'initialisation de solutions se fait d'une manière Aléatoire. , les étapes principales de notre système sont les suivantes :

- Déterminer un jeu de données à partitionner.
- Appliquer l'algorithme CS sur un jeu de données.
- Comparer les résultats de ces algorithmes en utilisant des mesures d'évaluation internes (la fonction objective SSE) et externes (la F-mesure).
- Evaluer l'indice FE qui représente le nombre d'évaluations de la fonction objectif que les algorithmes effectuent jusqu'à l'obtention de la meilleure valeur de cette fonction.

(La figure 3.1) représente le schéma global de notre application.

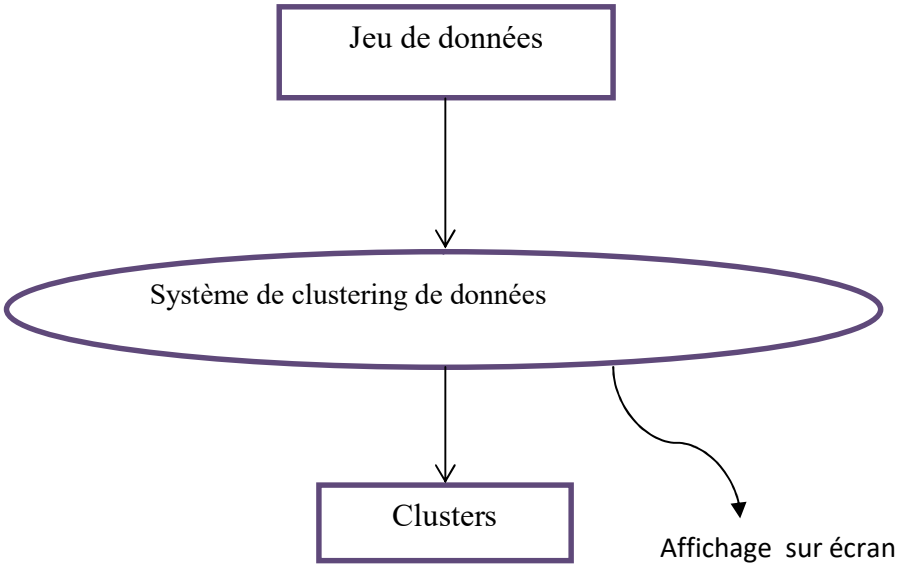


Figure 3.1. Diagramme de flux de données (DFD)-niveau 1-schéma global.

3.4. L'architecture générale de notre système :

L'architecture globale de notre système et le module est illustré dans la figure 3.2 ci-dessous :

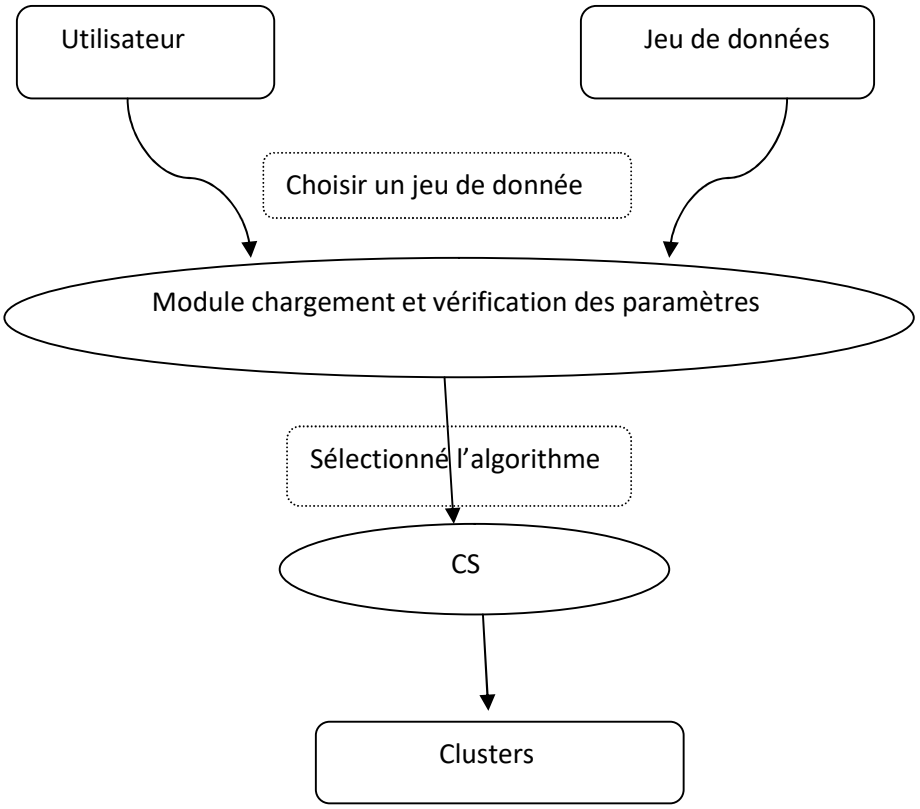


Figure 3.2 Diagramme de flux de données (DFD)-niveau 2 le module

Le système permet à l'utilisateur de :

- Choisir un jeu de données parmi : Wisconsin, Leukemia, Coloncancer, Lungcancer et ForestFires , Algeria Forest Fires (Bejaia) , Algeria Forest Fies (Sidi Bel Abbes).
- Chargement et affichage des paramètres du jeu choisi : nombre de dimensions (nd), nombre de clusters (K) et le nombre de points (nbr).
- Chargement choix e paramètre de l'algorithme : taille de la population, nombre d'itération, probabilité pa , nombre d'exécutions.
- Sélectionnée l'algorithme de CS.
- Affichage des résultats et les sauvegarder dans un fichier texte.

3.4.1. Module chargement et vérification :

Ce module permet de charger un jeu de données sous forme d'une matrice de données et de vérifier la validité des paramètres. Les sorties de ce module seront utilisés par l'un des différents modules de ce système. Le schéma de ce module est résumé dans la figure 3.3.

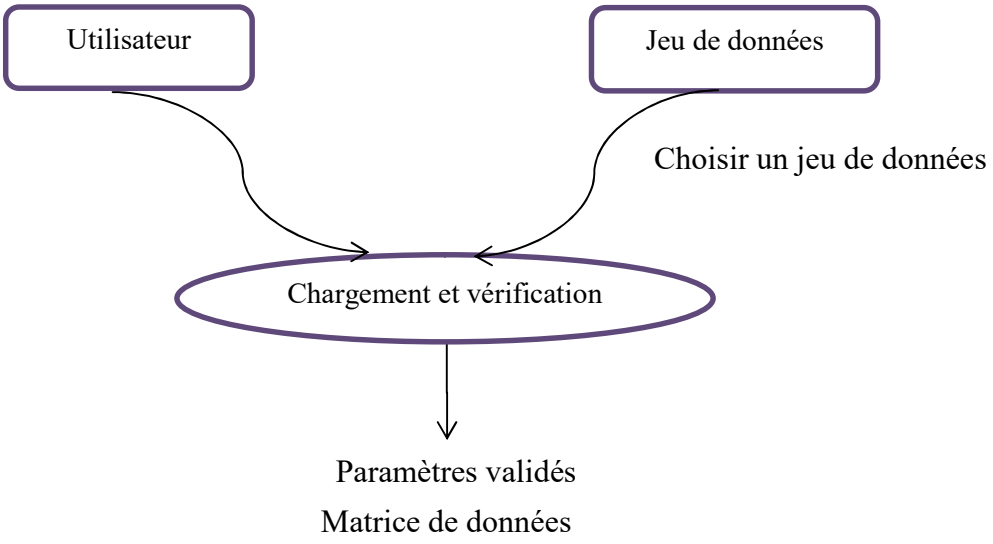


Figure 3.3 : Diagramme de flux de données (DFD)-niveau 2-Module de chargement et vérification.

3.4.2. Module CS (Cuckoo Search) :

Ce module permet d'appliquer l'algorithme CS à un jeu de données, cet algorithme est constitué de plusieurs étapes qui sont :

- Initialisation aléatoire des centroïdes.
- Affectation des points aux clusters.
- Evaluer les solutions en utilisant la fonction objective.
- Trier les solutions actuelles et trouver la meilleure.
- Générer des solutions en utilisant les vols de Lévy.
- Abandonner les mauvaises solutions avec une probabilité $P_{a.}$
- Remplacer la solution courante par une meilleure et calculer l'indice FE.
- Calculer la F-mesure.

La figure 3.4 représente l'algorithme de CS.

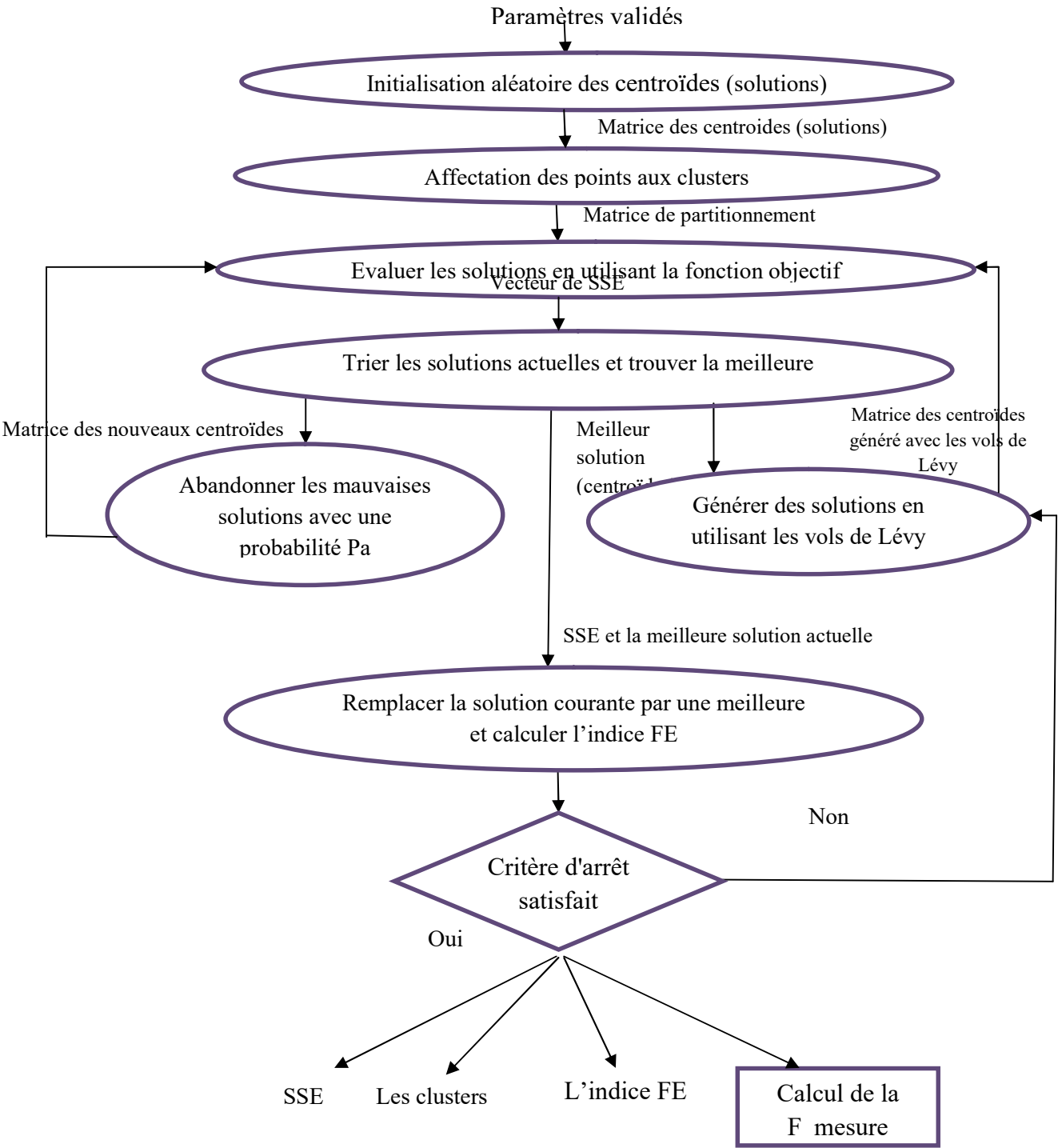


Figure 3.4. Diagramme de flux de données(DFD)- niveau 3- Algorithme CS.

La description des étapes de l'algorithme CS est la suivante :

A. Initialisation aléatoire:

Cette étape permet de choisir aléatoirement les centroïdes à partir d'un jeu de données.

B. Affectation des points aux clusters :

Après avoir initialisé les centroïdes, on affecte les points aux clusters en calculant la distance entre les points et les centroïdes en utilisant la formule suivante :

$$\boldsymbol{d} = \sqrt{\sum_{\boldsymbol{p=1}}^{\boldsymbol{m}} (\boldsymbol{o_{jp}} - \boldsymbol{c_{ip}})^{\boldsymbol{2}}} \qquad (3.2)$$

C. Evaluer les solutions en utilisant la fonction objectif:

Cette étape permet d'évaluer les solutions (centroides) en minimisant la fonction objective SSE représentée dans la formule (3.1).

D. Trier les solutions actuelles et trouver la meilleure :

Après avoir évalué les solutions, on effectue le tri de celle-ci de façon croissante en utilisant ses fonctions objectif, ce qui permet de trouver la meilleure solution.

E. Calcul des centroides avec le vol de Lévy :

En appliquant le vol de Lévy, cette étape permet de générer des solutions (calcul de centroïdes) en utilisant les formules suivantes

$$s = \frac{u}{|v|^{1/\beta}} \qquad (3.3)$$

$$u \sim N(0, \sigma_u^2) \ , \ v \sim N(0, \sigma_v^2) \qquad (3.4)$$

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta)\sin(\pi\beta/2}{\Gamma[(1+\beta)/2]\beta 2^{(\beta-1)/2}} \right\}^{1/\beta} , \ \sigma_v = 1 \qquad (3.5)$$

Où ***S*** représente la longueur du pas par la distribution de vol de Lévy.

F. Abandonner les mauvaises solutions (centroïdes) avec une probabilité P_a :

Cette étape permet d'abandonner les mauvaises solutions avec une probabilité p_a et de construire des nouveaux.

G. Remplacer la solution courante par une meilleure et calculer l'indice FE :

Cette étape permet de remplacer la solution courante par une meilleure en utilisant un test de comparaison entre les valeurs de la fonction objectif courante et l'ancienne. Elle permet également de calculer le nombre d'évaluations (FE) de la fonction objectif que l'algorithme CS effectue pour obtenir la meilleure valeur de la fonction objectif SSE.

H. Critère d'arrêt :

Le critère d'arrêt de cet algorithme est un nombre d'itération limité, si ce dernier atteint la fin des itérations l'algorithme s'arrête sinon il va continuer.

I. Calcul de la F-mesure :

Le calcul de la F-Mesure est constitué d'une seule étape représentée par la figure (3.5)

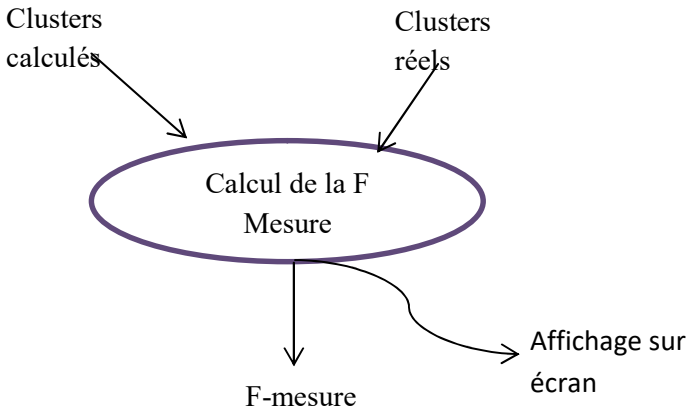


Figure 3.5 : Diagramme de flux de données(DFD)- niveau 4- F-Mesure.

Dans cette étape, on compare les clusters calculés par l'algorithme CS avec les clusters réels importés à partir d'un fichier du jeu de données. Le calcul de la F-Mesure se fait par la formule suivante :

$$\boldsymbol{Fmeasure} = \sum_{i=1}^{k'} \frac{N_i}{N} \max_{C_i \in C}(\boldsymbol{Fmes}(\boldsymbol{R_i}, \boldsymbol{C_j})) \qquad (3.6)$$

On détermine la qualité de clustering selon la valeur de la F-mesure, elle est limitée à l'intervalle [0,1], si cette valeur tend vers 1 le partitionnement est meilleur sinon le partitionnement est mauvais.

3.5. Conclusion :

Nous avons vu dans ce chapitre la description et l'adaptation de la méthode de recherche de coucou pour le clustering de données , ensuite nous avons déterminé l'idée de base de notre approche ,ainsi que les étapes principales et l'architecture générale de notre système , et nous avons également met en évidence la description de module du système en utilisant le diagramme de flux de données.

Dans le prochain chapitre nous allons présenter la partie implémentation de notre approche en déterminant le langage de programmation utilisé, la structure de données utilisées dans notre application ainsi que le déroulement d'une session de travail.

Chapitre 4

Implémentation de l'approche

4.1. Introduction:

Ce chapitre est consacré à l'implémentation de notre application, avec présentation des outils de développement matériels et logiciels. Après la présentation des outils, nous montrons quelques captures d'application illustrant son fonctionnement.

4.2. Environnement de développement :

Pour réaliser un projet, un ensemble d'outils logiciels et matériels sont nécessaires pour pouvoir accomplir le travail demandé.

4.2.1. Environnement matériel :

L'application a été développée sous un environnement matériel caractérisé par :

- **Les outils utilisés** : PC (Intel® Core™i2 Duo, CPU3.00 GHz, 2.00 Go de RAM).

4.2.2. Environnement logiciel :

- **Système d'exploitation :** L'application a été développée et exécutée sous un système d'exploitation Windows 10 Famille 64 bits.
- **Langages de programmation (MATLAB)** : (MATrixLABoratory) est un langage de programmation de quatrième génération et un environnement de programmation interactif pour le calcul scientifique et la visualisation des données, il est développé par la société The MathWorks.

➢ **Le choix du langage MATLAB : Nous avons opté pour le langage MATLAB pour les raisons suivantes :**

❖ C'est un environnement de développement puissant capable d'effectuer de grands calculs.

❖ Puisque nous nous intéressons au clustering de données, ces données sont en fait des matrices de (n*m) dimensions; le langage MATLAB facilite la manipulation de matrice, l'affichage des courbes et des données, la mise en œuvre des algorithmes, et la création des interfaces utilisateurs.

❖ Il offre de bibliothèques riches en commandes, ce qui permet de construire rapidement des applications et simplifie au maximum l'écriture du code.

4.3. Les structures de données :

Pour la représentation de différents composants de notre application, nous avons utilisé :

La notion du tableau : Un tableau est une structure de données de type liste contenant une collection d'éléments stockés en mémoire de façon contigüe et accessibles via un indice; On a utilisé les tableaux dans la programmation car les jeux de données sont stockés dans des

fichiers sous forme de tableaux, en plus la manipulation de tableaux est simple et l'accès à la mémoire est rapide.

A. Les tableaux utilisés : Voici quelques tableaux utilisés dans notre application

- Data [nbr, nd] : c'est la matrice qui contient le jeu de données de nd dimension et nbr points.
- best_Vect [nbr] : la partition trouvée à la fin d'exécution.

4.4. Déroulement d'une session de travail :

4.4.1. Le schéma global de déroulement (voir figure 4.1)

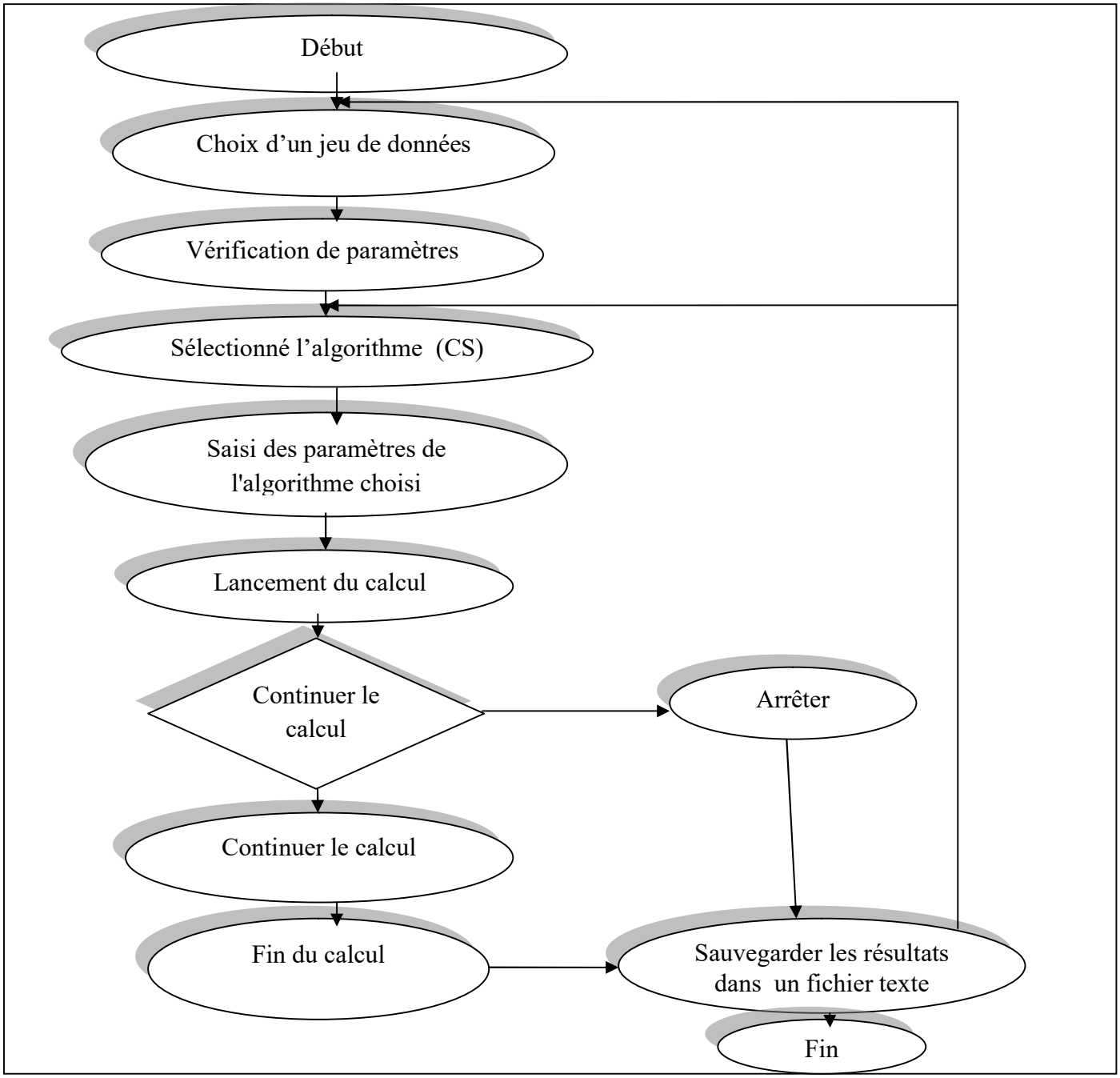


Figure 4.1. Déroulement 'une session de travail.

4.4.2. Présentation de l'application :

Notre application est composée de deux parties principales :

A. La Fenêtre D'accueil :

La fenêtre d'accueil contient les boutons de fonctionnalités principales. Le bouton principal « Fenêtre principale » qui mène à l'interface principale de notre application.

Figure 4.2 : Fenêtre d'accueil.

Le menu de cette fenêtre contient un bouton qui permet à l'utilisateur d'effectuer deux actions :

- ***Fichier*** : lorsqu'on clique sur ce bouton, on peut soit afficher la fenêtre principale ou bien quitter l'application.

B. La Fenêtre Principale : elle est divisée en 6 zones (voir figure 4.3).

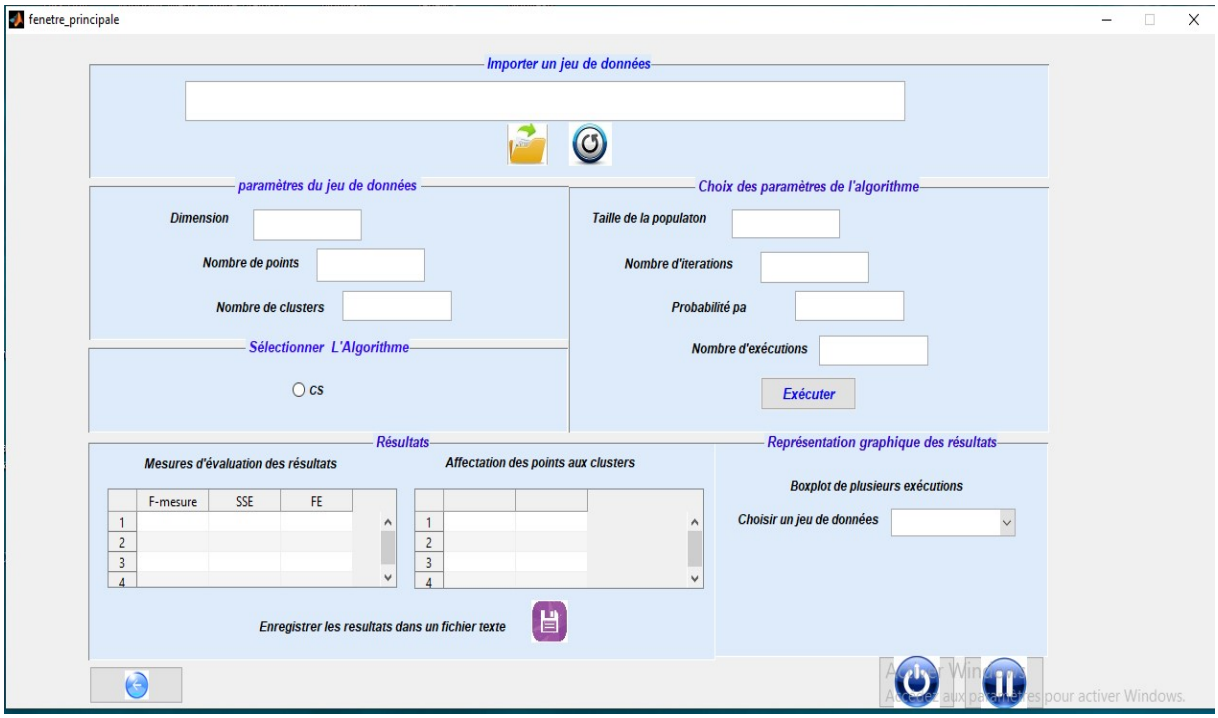


Figure 4.3. La fenêtre principale.

1. La première zone « *Importer un jeu de données* » permet à l'utilisateur de choisir le jeu de données, lorsqu'on clique sur le bout  une boite de dialogue s'affiche sur l'écran pour sélectionner le fichier du jeu de données voulu (voir figure 4.4).

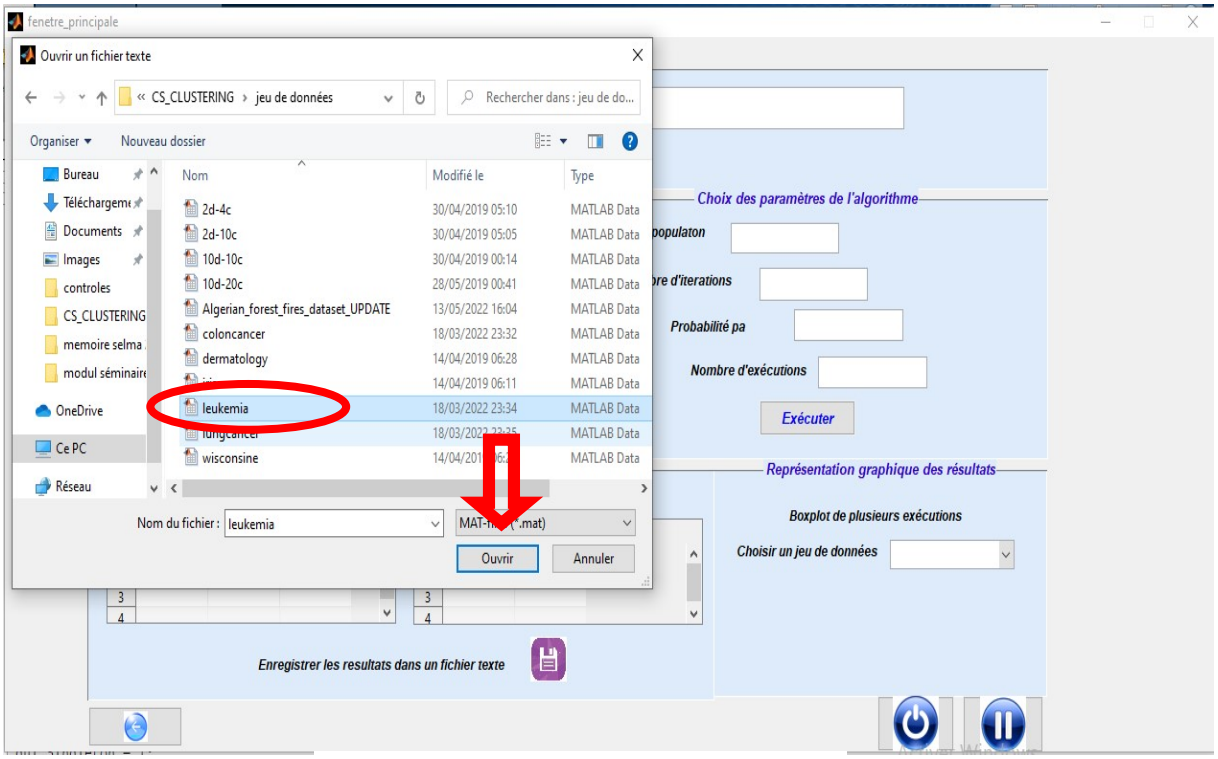


Figure 4.4. Choix du jeu de données.

2. Dans la deuxième zone « *Paramètres du jeu de données* » En cliquant sur le bouton « ouvrir» les paramètres du jeu de données choisi (qui sont constantes), son chemin s'affiche.

- ❖ l'utilisateur peut consulter les paramètres du jeu de données choisi tels que le nombre de clusters **K**, le nombre de points **nbr** et les dimensions **nd**, mais sans effectuer aucune modification.

- ❖ Le bouton recommencer  permet de changer le choix du jeu de données. (voir figure 4.5)

Figure 4.5. Paramètres du jeu de données.

3. La troisième zone « *Sélectionner un algorithme*» : l'utilisateur peut accéder à cette zone pour sélectionner l'algorithme à appliquer. Cette partie permet d'ajouter d'autres algorithmes dans l'avenir.

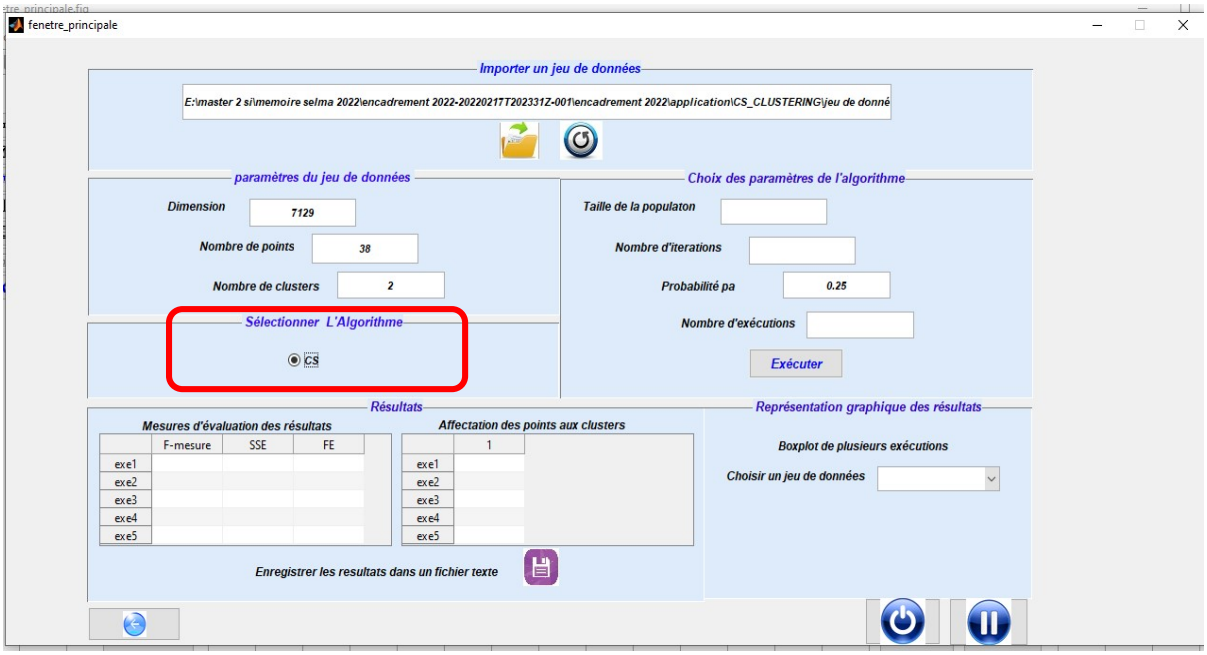


Figure 4.6 : Sélection l'algorithme.

4. La quatrième zone: « *choix des paramètres de l'algorithme* » quant l'utilisateur sélectionne n'importe quel algorithme, les paramètres s'affichent et l'utilisateur aura la possibilité de les modifier.

❖ L'utilisateur peut saisir les paramètres de l'algorithme à exécuter; tel que la taille de populations, le nombre d'itérations, et le nombre d'exécutions de l'algorithme **n**. (voir figure 4.7)

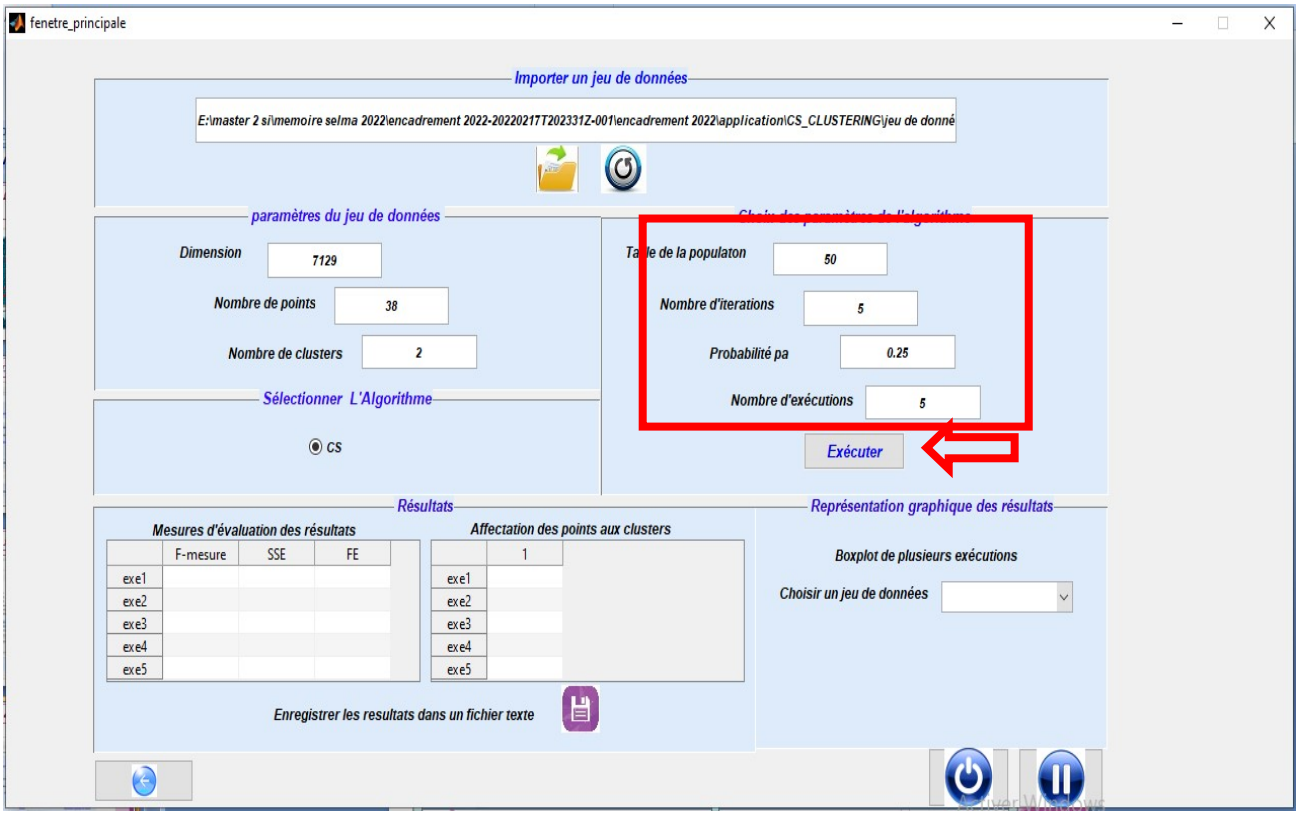


Figure 4.7. Saisie des paramètres de l'algorithme.

- L'utilisateur doit cliquer sur le bouton '**Exécuter** ' pour lancer l'exécution
- Une foi l'exécution d'un algorithme est lancée; une barre de progression s'affiche. Comme montre (la figure 4.8).

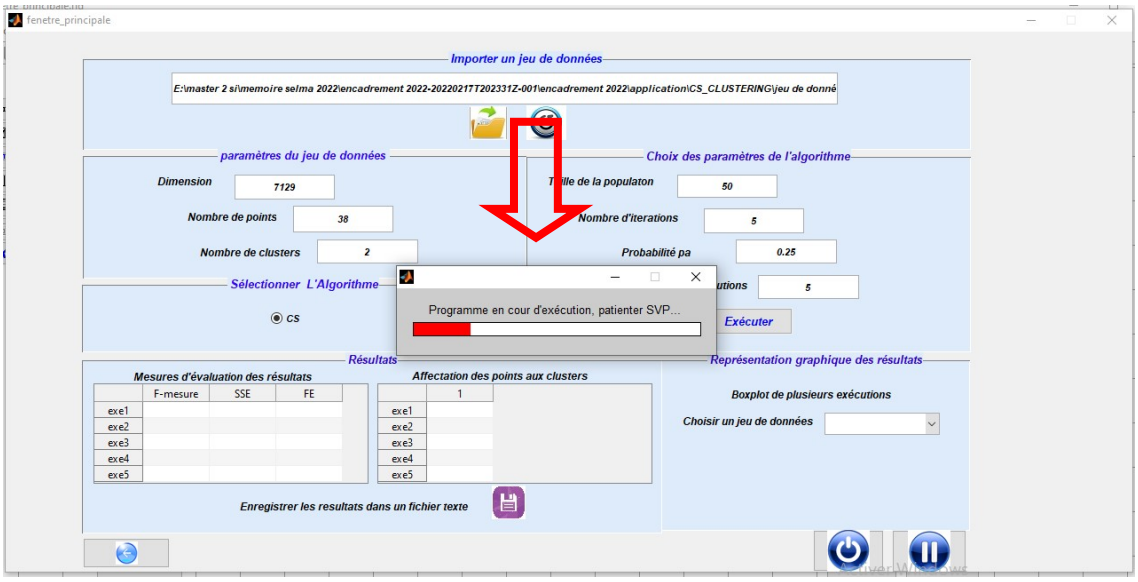


Figure 4.8. Une barre de progression de l'exécution de l'algorithme.

5. La cinquième zone : « *Résultats d'exécution* » : une fois l'exécution est terminée, tous les résultats s'affichent de manière progressive dans les deux tableaux (mesures d'évaluation des résultats, et l'affectation des points aux clusters). (Voir figure 4.9)

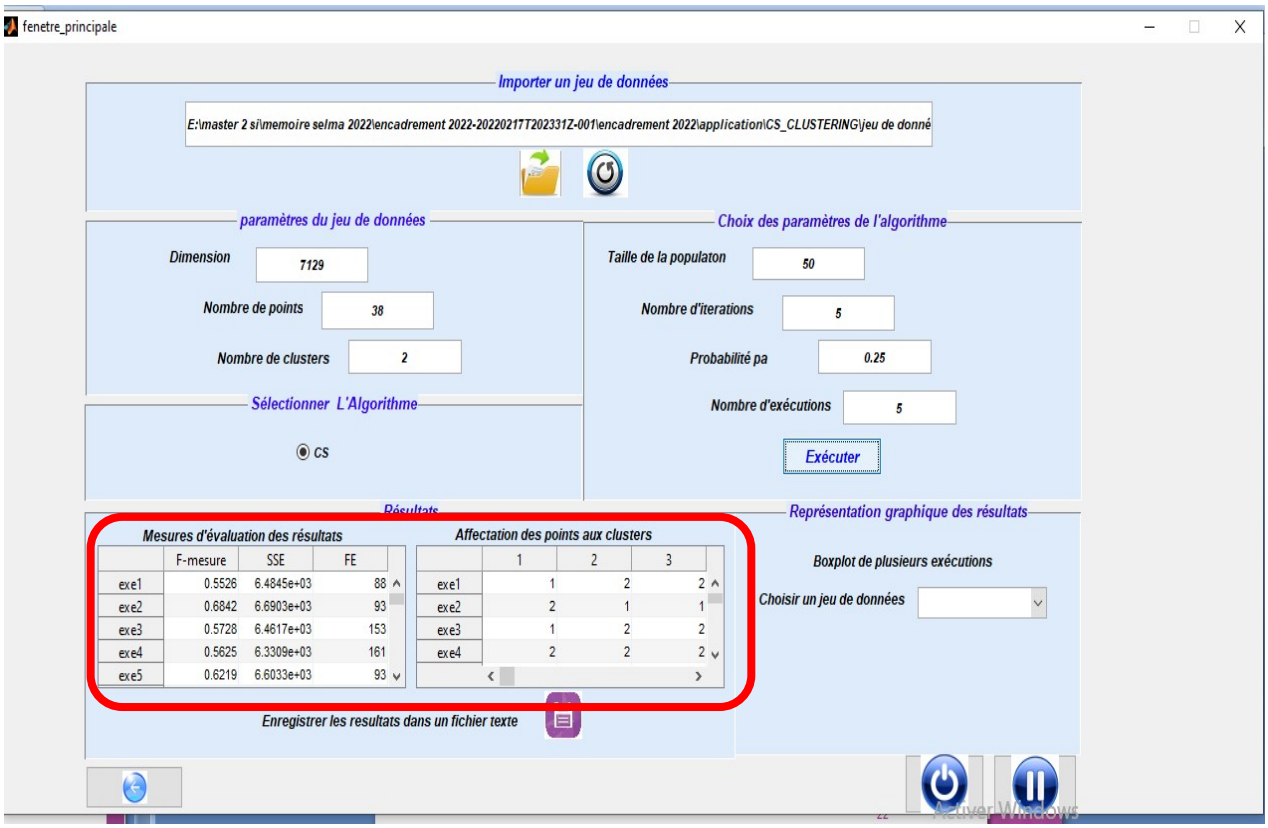


Figure 4.9 : Résultats d'exécution de l'algorithme

- L'utilisateur peut sauvegarder les résultats affichés dans un fichier texte en cliquant sur le bouton  ; un fichier texte sera crée et sauvegarde dans un chemin précis (voir figure 4.10)

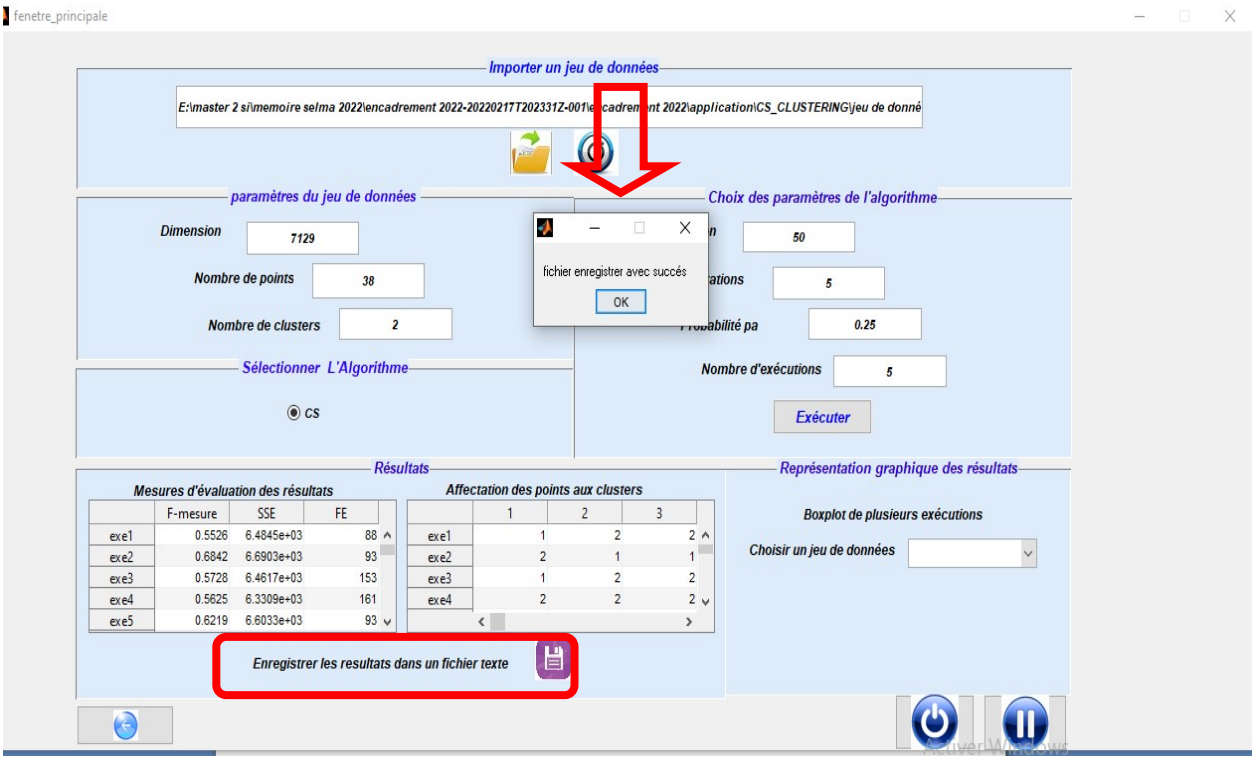


Figure 4.10 : Enregistrement des résultats.

6. La sixième zone « *Représentation graphique des résultats* » : l'utilisateur choisit un jeu de données (Wisconsin, Leukemia, Coloncancer, Lungcancer, Forestfires, Algerian Forest Fires (Bejaia), Algerian Forest Fires (Sidi Bel Abbes)).

❖ afin de visualiser les résultats de l'algorithme. Les résultats sont représentés sous forme de trois boxplots de la F-mesure, la SSE et le nombre FE. (voir figure 4.11)

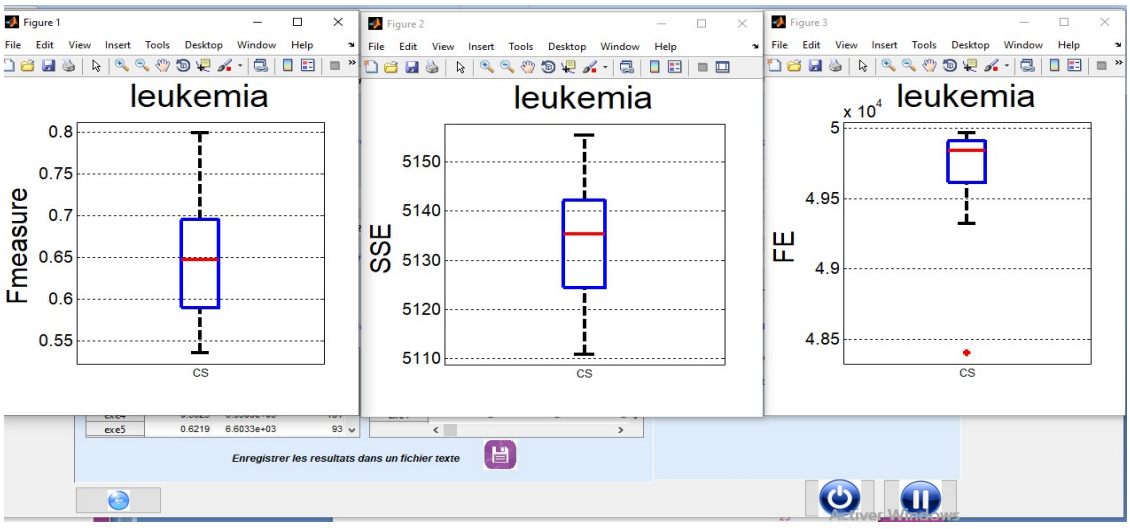


Figure 4.11. Les trois boxplots de la F-mesure, la SSE et l'indice FE.

- ***Autres boutons:*** à partir desquels l'utilisateur peut effectuer les actions suivantes :
  - Arrêter  pour arrêter l'exécution temporairement.
  - Fermer  pour quitter l'application définitivement.
  - Flèche précédente  pour aller à la fenêtre de démarrage (accueil).

4.5. Conclusion :

Dans ce chapitre, nous avons présenté une partie essentielle de notre approche, c'est l'implémentation de notre application. Dans cette partie, on a précisé les outils de développement matériels et logiciels, ainsi que la présentation des outils utilisés et quelques captures d'application illustrant son fonctionnement. Dans le prochain chapitre, nous allons présenter les jeux de données utilisées et les paramètres choisis, ensuite la présentation graphique des résultats en utilisant les boxplots, nous terminons par l'évaluation et l'analyse de ces résultats.

Chapitre 5

Résultats expérimentaux

5.1. Introduction:

Après avoir présenté dans le chapitre précédent la partie implémentation de notre approche, en précisant les outils de développement matériels et logiciels, ainsi que la présentation des outils utilisés et quelques captures d'application illustrant son fonctionnement. Dans ce chapitre, nous présentons les résultats obtenus pour l'algorithme utilisé, nous commençons par présenter les jeux de données utilisées et les paramètres choisis, nous passons ensuite à la présentation graphique des résultats en utilisant les boxplots. A la fin, nous terminons par l'évaluation et l'analyse de ces résultats.

5.2. Tests et résultats :

Nous allons appliquer notre approche sur deux types de données, des données représentant des maladies cancéreuses et d'autre représentant des données d'incendies. Nous présentons les résultats de notre approche et leur présentation graphique en utilisant les boxplots.

5.2.1. Jeux de données :

Nous avons appliqué l'algorithme de coucou à des données réelles définies comme indiqué dans le tableau 5.1, où *m* désigne le nombre total de points de données dans le jeu de données, *Dim* donne la dimensionnalité et *K* est le nombre de groupes. Les jeux de données réels sont pris de l'UCI qui est un répertoire de bases de données d'apprentissage [Blake et Merz, 98].

A. Données représentant des maladies cancéreuses :

❖ **Colon cancer :**

Cette base de données relative au cancer du côlon a été mise à disposition par l'université PRINCETON. Ce jeu de données contient des informations sur 62 tissus et 2000 attributs, les tissus sont répartis en deux classes : tissu normal et tissu tumeur.

Le jeu de données contient l'expression de 2000 gènes à plus haute intensité minimale pour 62 tissus. Les gènes sont placés dans l'ordre décroissant de l'intensité minimale. Chaque attributs de 2000 attributs est une intensité de gènes provenant des ~ 20 paires de fonctions qui correspondent au gène de la puce, obtenue en utilisant un procédé de filtration. Les données sont par ailleurs non traités (par exemple, elles n'ont pas été normalisées par l'intensité moyenne de chaque expérience).

❖ **Wisconsin (Breast Cancer) :**

Wisconsin Breast Cancer est un jeu de données qui contient les informations médicales de 699 cas cliniques relatifs au cancer du sein répartis sur 2 groupes avec 9 attributs. L'objectif est de classer chaque cas en tumeurs bénignes ou malignes [Blake et al, 98].

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.

❖ **Lung cancer:**

Le jeu de données “Lung cancer” (cancer du poumon) est disponible sur le web et a été publié dans [HANG et YANG, 91]. Le jeu de données “Lung cancer” (cancer du poumon) décrit trois types de pathologie du cancer du poumon. Il contient des données manquantes. Le jeu contient 27 instances pour 56 attributs descriptifs. Tous les attributs sont nominaux et prennent leurs valeurs parmi les entiers de 0 à 3 [JOURDAN, 03]

❖ **Leukemia :**

Le jeu de données Leukemia contient des informations médicales de 38 cas relatifs au cancer du sang classé en deux classes avec 7129 attributs.

B. Données représentant des incendies de forêt:

❖ **Algerian Forest Fires (Bejaia) et Algerian Forest Fires (Sidi Bel Abbes) :** Le jeu de données comprend 244 instances qui regroupent les données de deux régions de l'Algérie, à savoir la région de Bejaia située au nord-est de l'Algérie et la région de Sidi Bel-abbes située au nord-ouest de l'Algérie.122 instances pour chaque région. La période de juin 2012 à septembre 2012. L'ensemble de données comprend 11 attributs et 1 attribut de sortie (classe). Les 244 instances ont été classées en classes « incendie » (138 classes) et « non incendie » (106 classes).

Le jeu de données regroupe deux régions de l’Algérie : la région de Sidi Bel abbas située au nord-ouest de l’Algérie et la région de Bejaia située au nord-est de l’Algérie. L’ensemble de données utilisé comprend 244 instances (une combinaison des données recueillies dans Sidi Bel-abbas et Bejaia, avec 122 instances chacune).

Les observations météorologiques pour la période de l'été 2012, de juin à septembre, ont été utilisées puisque le nombre d'incendies est élevé pendant cette période et que l'année 2012 est l'année où le nombre d'incendies enregistré est le plus élevé de 2007 à 2018. L'ensemble de données utilisé comprend les éléments météorologiques critiques qui influent l'occurrence d'incendies de forêt, à savoir la température, l'humidité relative et la vitesse du vent. Les valeurs des attributs numériques sont utilisées pour chercher deux possibilités, à savoir « incendie » et « pas d'incendie ». [Faroudja ABID et al, 2019],

Le jeu de données est décrit par 12 attributs numériques et littéraux :

1. Le jour
2. Le mois de (1 à 12),
3. L'année (2012)

Observations des données météorologiques.

4. Temp : température midi (température max) en degrés Celsius : 22 à 42.
5. RH : (en Anglais Relative Humidity) Humidité relative en %: 21 à 90.
6. Ws : (en Anglais) Vitesse du vent en km/h : 6 à 29.
7. Rain : journée totale en mm : 0 à 16,8.

Composants FWI

8. Indice Fine Fuel Moisture Code (FFMC): 28,6 à 92,5.
9. Duff Indice Duff Moisture Code (DMC): 1,1 à 65,9.
10. Indice Drought Code (DC): 7 à 220,4.
11. Indice Initial Spread Index (ISI): 0 à 18,5.
12. Indice Buildup Index (BUI): 1,1 à 68.
13. IndiceFire Weather Index (FWI): 0 à 31.1.

Fine Fuel Moisture Code (FFMC), Duff Moisture Code (DMC), Drought Code (DC), Initial Spread Index (ISI), Bui Index (BUI) et FWI. Les trois premiers sont liés aux codes de carburant : le FFMC indique la teneur en humidité des déchets de surface et influence l'inflammation et la propagation du feu, tandis que la DMC et le DC représentent la teneur en humidité des couches organiques peu profondes et profondes, qui influent sur l'intensité du feu.

L'ISI est un score qui correspond à la propagation de la vitesse du feu, tandis que le BUI représente la quantité de carburant disponible. L'indice FWI est un indicateur de l'intensité du feu et il combine les deux composantes précédentes. Bien que différentes échelles soient utilisées pour chacun des éléments FWI, des valeurs élevées suggèrent des conditions de combustion plus sévères. De plus, les codes d'humidité du carburant exigent un délai des conditions

météorologiques passées : 16 heures pour le FFMC, 12 jours pour le DMC et 52 jours pour le DC.

❖ **Forest fires :**

Ce jeu de données contient des données sur les feux de forêt du parc naturel de Montesinho, dans la région de Tras-os-Montes au nord-est du Portugal. Ce parc contient une grande diversité de flore et de faune. Insérée dans un climat supra-méditerranéen, la température annuelle moyenne est comprise entre 8 et 12 C. Les données utilisées ont été recueillies de janvier 2000 à décembre 2003.

L'ensemble de données sont décrits par 13 attributs:

Pour plus d'informations, lire [Cortez and Morais, 2007].

1. X : coordonnée spatiale sur l'axe des x dans la carte du parc de Montesinho: 1 à 9
2. Y : Coordonnée spatiale sur l'axe des ordonnées dans la carte du parc de Montesinho : 2 à 9.
3. month : mois de l'année : 'jan' à 'dec'.
4. day : jour de la semaine: 'mon' à ‘sun’.
5. FFMC : Indice FFMC: 18.7 à 96.20.
6. DMC : Indice DMC: 1.1 à 291.3.
7. DC : Indice DC: 7.9 à 860.6.
8. ISI : Indice ISI WI: 0.0 à 56.10.
9. temp : température en degrés Celsius: 2.2 à 33.30.
10. RH : (en Anglais Relative Humidity) Humidité relative en %: 15.0 à 100.
11. wind : (en Anglais wind speed) vitesse du vent en km/h: 0.40 à 9.40.
12. rain : pluie extérieure en mm/m2 : 0.0 à 6.4.
13. area : a superficie brûlée de la forêt (en ha): 0.00 à 1090.84.

Le tableau 5.1 récapitule des informations sur les jeux de données d’incendies.

Jeu de données réels	K	Dim	M
Leukemia	2	7129	38
Coloncancer	2	2000	62
Lungcancer	3	56	27
Wisconsin	2	9	683
Forest Fires	2	13	517
Algerian Forest Fires (Bejaia)	2	13	122
Algerian Forest Fires (Sidi Bel Abbes)	2	13	122

Tableau 5.1. Résumé des jeux de données réels.

5.2.2. L'évaluation :

L'évaluation est faite à différents niveaux. La première évaluation est accomplie avec la mesure externe : la F-mesure [Stein et al, 03]. C'est une fonction souvent utilisée dans la littérature de clustering. Elle compare la qualité du clustering au classes connues et correctes pour un jeu de données. Soit $C = (C_1, C_2,..., C_K)$ le clustering des donnés et soit $R = (R_1,R_2,..., R_K)$ l'ensemble des classes correctes. La F-mesure d'un groupe donné *Ci* par rapport à une classe *Rj* est :

$$\text{Prec}(R_i, C_j) = \frac{N_{ij}}{N_j} \quad et \quad \text{Rep}(R_i, C_j) = \frac{N_{ij}}{N_i} \qquad (5.1)$$

Soit *m* le nombre total de points de données. La F-mesure de l'ensemble de clustering *C* par rapport à *R* est défini par l'équation (5.2). La F-mesure prend ses valeurs dans l'intervalle [0,1] et devraient être maximisée pour un clustering optimal.

La valeur de F-mesure est calculée en utilisant la formule suivante :

$$F(C) = \sum_{i=1}^{k'} \frac{N_i}{N} \max_{Ci \in C}(Fmes(R_i, C_j)) \qquad (5.2)$$

La deuxième évaluation est accomplie avec la fonction objectif interne SSE pour évaluer la qualité de l'optimisation. Elle consiste à calculer la somme de la distance entre des points de données et leurs centroïdes de groupes correspondant:

$$SSE = \sum_{i=1}^{k} \sum_{j=1}^{n} W_{ij} * \sqrt{\sum_{p=1}^{m} (o_{jp} - c_{ip})^2} \qquad (5.3)$$

Où $\boldsymbol{W_{ij} = 1}$ si l'objet est dans le cluster et **0** sinon, **k** est nombre de cluster, **n** est le nombre d'objet, **m** est le nombre des attributs, $\boldsymbol{c_{ip}}$ est la valeur de l'attribut numéro **p** du centroïdes de cluster numéro **i**, et $\boldsymbol{o_{jp}}$ est la valeur de l'attribut numéro **p** de point de jeu de données numéro **i**. Comme objectif, la fonction *SSE* devrait être minimisée.

La troisième évaluation consiste à calculer la valeur FE qui représente le nombre d'évaluations de la fonction objectif que l'algorithme effectue pour obtenir la meilleure valeur de la fonction objectif. Ce nombre d'évaluations de la fonction objectif (FE) donne une idée sur la vitesse et la qualité de la convergence de l'algorithme.

5.2.3. Représentation graphique des résultats :

La représentation des résultats est accomplie en utilisant l'outil statistique "Boxplot", c'est un des outils fourni par MATLAB. Cette représentation a un double intérêt : la simplification de la visualisation des résultats obtenus pour chaque jeu de données et la description détaillée des résultats d'une manière claire, simple et réduite. La figure 5.1 suivante présente les informations essentielles données par un Boxplot.

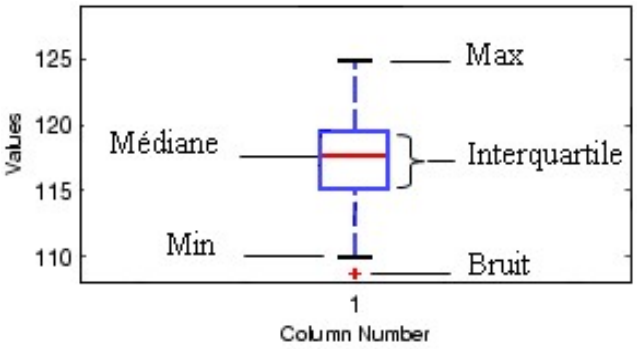


Figure 5.1. Informations données par un Boxplot.

Le fond et le sommet 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 par un plus "+", [Ramdane, 06].

5.3. Les résultats :

Les résultats sont montrés dans les tableaux de 5.2 jusqu'à 5.4 qui montrent les valeurs de la médiane, l'interquartile, le maximum et le minimum des mesures d'évaluations obtenus à travers 50 exécutions des deux variantes d'algorithme CS.

Ces tableaux sont visualisés dans les boxplots des figures 5.2, 5.3 et 5.4.

Jeu de données		***CS***
Wisconsin	Médiane	0.9648
	Interquartile	0
	Max	0.9648

	Min	0.7327
Leukemia	Médiane	0.6466
	Interquartile	0.1058
	Max	0.7989
	Min	0.5353
Coloncancer	Médiane	0.6864
	Interquartile	0
	Max	0.6864
	Min	0.6289
Lungcancer	Médiane	0.6199
	Interquartile	0.1053
	Max	0.7007
	Min	0.4618
Forest Fires	Médiane	0.6083
	Interquartile	0
	Max	0.6674
	Min	0.5339
Algerian Forest Fires (Bejaia)	Médiane	0.7195
	Interquartile	0
	Max	0.7195
	Min	0.6291
Algerian Forest Fires (Sidi Bel Abbes)	Médiane	0.6077
	Interquartile	0
	Max	0.6784
	Min	0.6003

Tableau 5.2. Interquartile, max et min de la F-mesure obtenus pour CS.

Jeu de données		***CS***
Wisconsin	Médiane	2.9644e+03
	Interquartile	7.5000e-04

	Max	2.9644e+03
	Min	2.9644e+03
Leukemia	Médiane	5.1353e+03
	Interquartile	17.8829
	Max	5.1554e+03
	Min	5.1109e+03
Coloncancer	Médiane	4.3870e+06
	Interquartile	3.0119e+04
	Max	4.4530e+06
	Min	4.2947e+06
Lungcancer	Médiane	110.4034
	Interquartile	110.5425
	Max	111.5245
	Min	107.6994
Forestfires	Médiane	5.0039e+04
	Interquartile	0.5098
	Max	5.0040e+04
	Min	5.0039e+04
Algerian Forest Fires (Bejaia)	Médiane	3.8325e+03
	Interquartile	0.0128
	Max	3.8325e+03
	Min	3.8325e+03
Algerian Forest Fires (Sidi Bel Abbes)	Médiane	3.7757e+03
	Interquartile	0.0270
	Max	3.7758e+03
	Min	3.7757e+03

Tableau 5.3. Interquartile, max et min de la fonction objective SSE.

Jeu de données	***CS***

Wisconsin	Médiane	4.9776e+04
	Interquartile	399
	Max	49984
	Min	48304
Leukemia	Médiane	4.9845e+04
	Interquartile	298
	Max	49967
	Min	48403
Coloncancer	Médiane	4.9768e+04
	Interquartile	478
	Max	49993
	Min	48847
Lungcancer	Médiane	4.9854e+04
	Interquartile	224
	Max	49999
	Min	48613
Forest Fires	Médiane	49752
	Interquartile	392
	Max	49999
	Min	49033
Algerian Forest Fires (Bejaia)	Médiane	4.9736e+04
	Interquartile	0.0128
	Max	49997
	Min	47953
Algerian Forest Fires (Sidi Bel Abbes)	Médiane	4.9775e+04
	Interquartile	334
	Max	49991
	Min	47862

Tableau 5.4. Interquartile, max et min de FE obtenues pour CS.

A. Résultats évalués avec la F mesure :

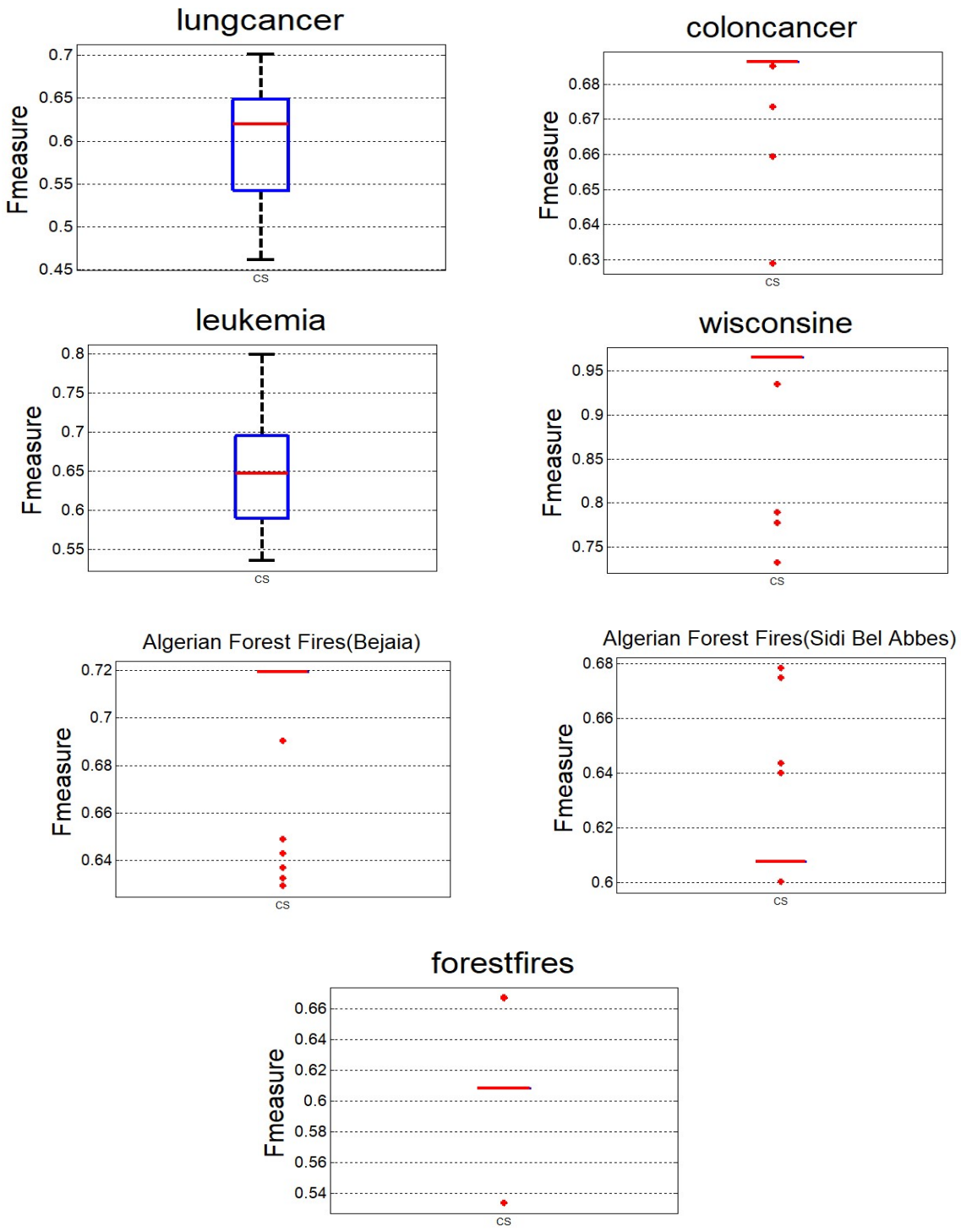


Figure 5.2. Les boxplots des résultats de CS avec la fonction objectif la F-mesure.

B. Résultats évalués avec la SSE :

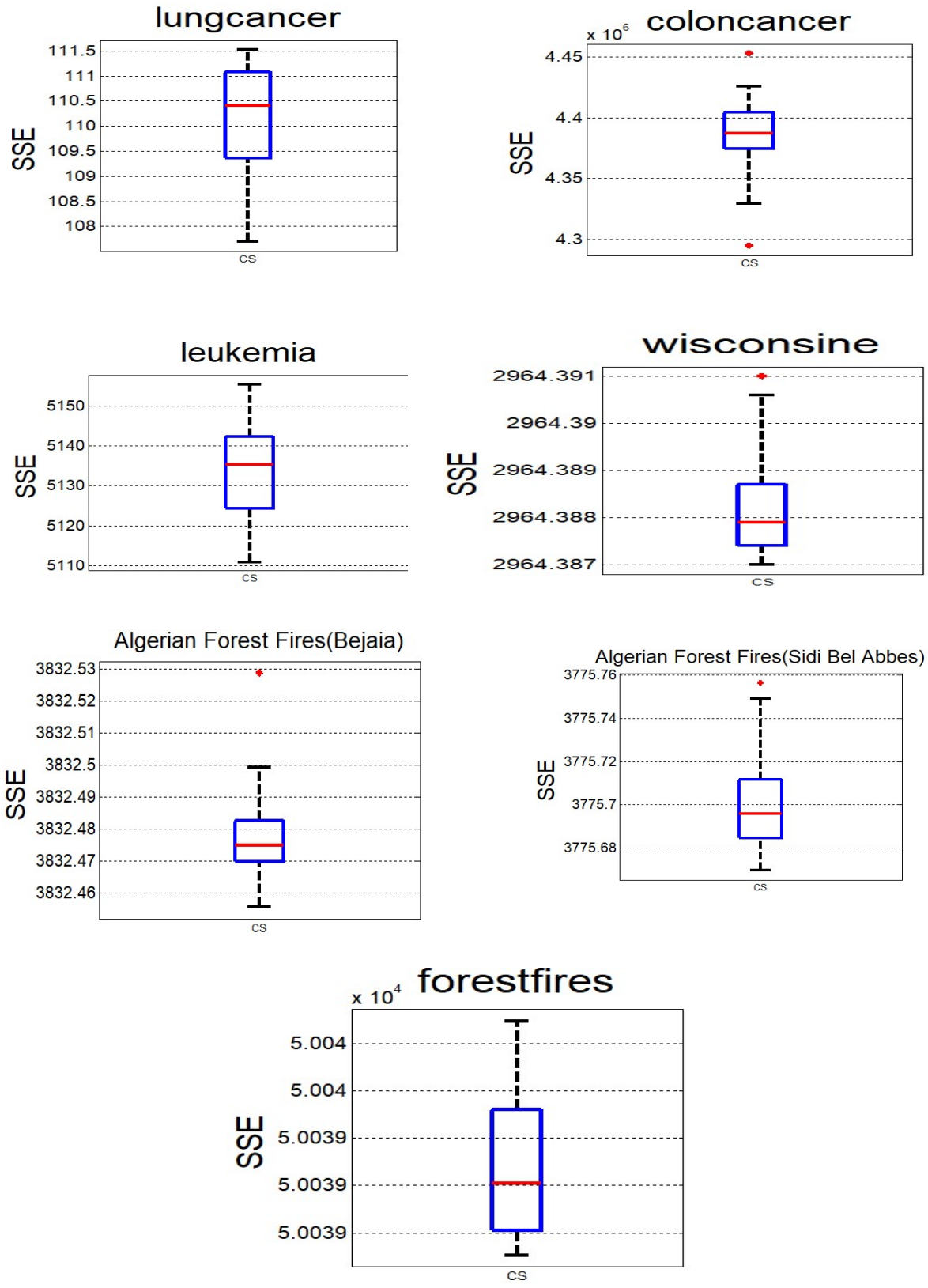


Figure 5.3. Les boxplots des résultats de CS avec la SSE.

C. Résultats évalués avec la FE :

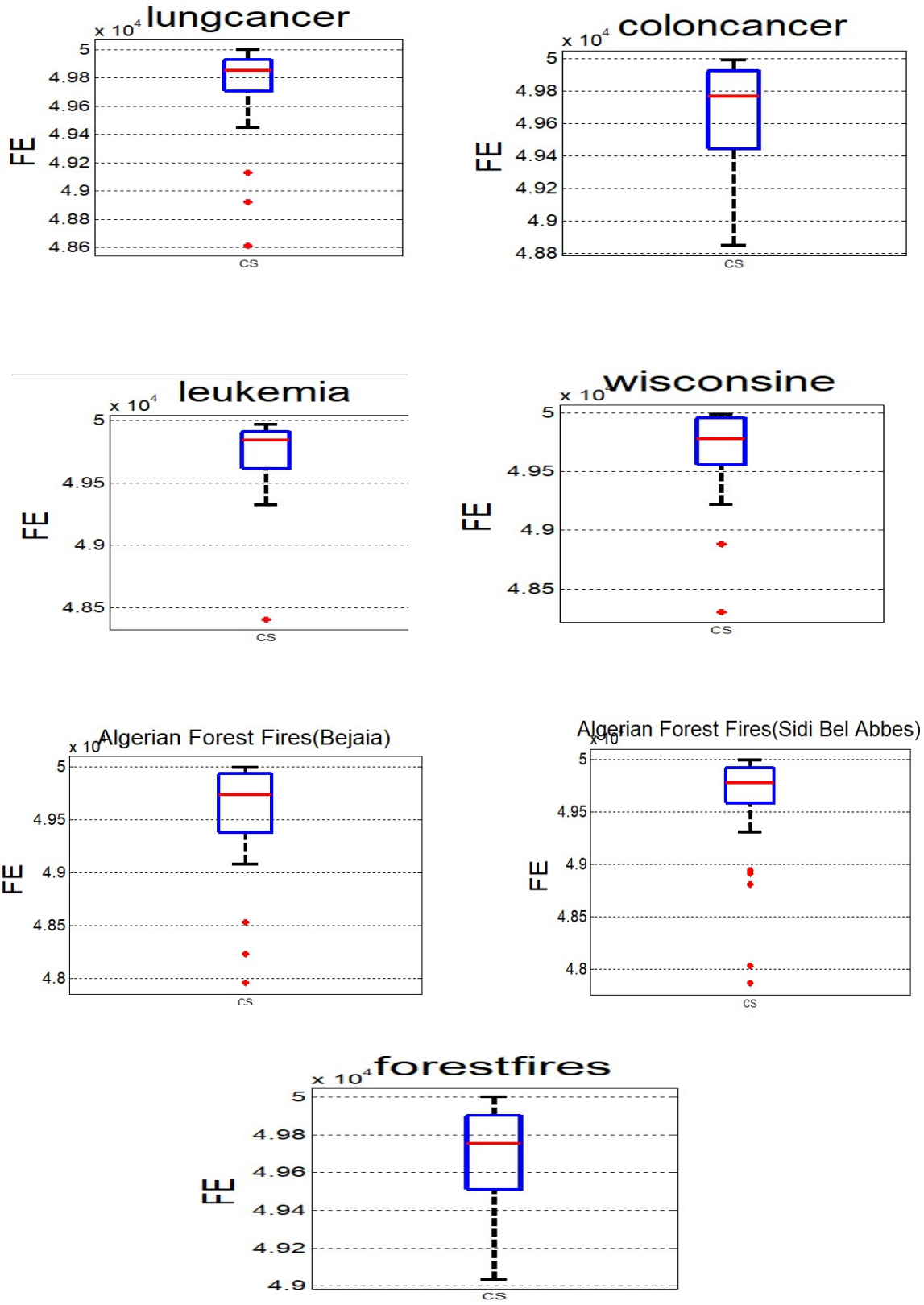


Figure 5.4. Les boxplots des résultats CS avec la FE.

En général, l'algorithme CS trouve des résultats acceptables pour les jeux de données d'incendies de forêt, l'algorithme arrive à détecter l'incendie selon les caractéristiques météorologiques et des indices forêt-météo avec un pourcentage de 72% pour le jeu de données de la région de Béjaia et 61% pour le jeu de données de la région de Sidi Bel Abbes et 61% pour le parc naturel de Montesinho au Portugal.

Pour les jeux de données de maladies cancéreuses, l'algorithme détecte les tumeurs bénignes des tumeurs malignes pour le jeu de données Wisconsin avec un taux de 96%. pour Coloncancer, l'algorithme arrive à un taux de 69% et un taux de 65% pour Leukemia et un taux de 62% pour détecter les deux types de tumeurs pour Lungcancer.

5.4. Conclusion :

Dans ce chapitre, nous avons présenté les tests et les résultats et les expérimentations effectués sur plusieurs jeux de données. L'évaluation de résultats est faite avec plusieurs mesures, A la fin, nous avons présenté une analyse et une discussion de nos résultats.

Conclusion générale

Conclusion générale

Dans ce travail de master, nous avons présenté l'adaptation de la méthode de recherche de coucou au problème de clustering de données. D'abord, nous avons commencé par présenter les notions et les concepts de base du domaine de clustering de données, tels que les définitions de clustering et de cluster, les techniques principales et techniques de validation. Ensuite nous avons passé à étudier les concepts, les principes et le comportement de la méthode de recherche de coucou ainsi que la description détaillée de son algorithme.

A la fin, nous avons passé à l'adaptation de la méthode de la recherche de coucou au clustering en l'appliquant sur des données de maladies cancéreuses et des données d'incendies de forêts.

Nous avons mené une série d'expérimentations, tests et analyse des résultats basée sur des mesures internes et externes.

Plusieurs enrichissement peuvent être apportés à notre application et peuvent être suivi pour la continuité de ce travail, l'utilisation l'algorithme de recherche coucou à plusieurs variantes comme CS basé sur une optimisation multi objective de plusieurs critères.

.

Bibliographies

[Bak, 97]: Bak, P: ″How nature works″. Oxford university press Oxford, 1997.

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

[D. P. Mercer, 2003] : D.P.Mercer, linacre college : " Clustering large Datasets ", Octobre 2003.

[Guillaume, 04] : Guillaume Cleuziou, thèse de doctorat : "Une méthode de classification non-supervisée pour l'apprentissage de règles et la recherche d'information", Décembre 2004.

[Ishak et al, 14]: Ishak B.S, Kamel.N et Bendjeghada.O: Article "A New Algorithm for Data Clustering Based on Cuckoo Search Optimization "Advances in Intelligent Systems and Computing 238, Springer International Publishing Switzerland, 2014.

[Julia, 03]: Julia Handl, mémoire de magistère:" Ant-based methods for tasks of clustering and topographic mapping, improvements, evaluation and comparison with alternative methods", 2003.

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

[Laetitia, 03] : Laetitia Jourdan, thèse de doctorat : "Métaheuristiques pour l'extraction de connaissances application à la génomique", Novembre 2003.

[Laurent, 04] : Laurent Candillier, rapport technique : "La classification non supervisée", Septembre 2004.

[Ouaarab, 15] : Aziz OUAARAB : Thèse de Doctorat ″Résolution de Problèmes d'Optimisation Combinatoire par des Métaheuristiques Inspirées de la Nature : Recherche du Coucou via les Vols de Lévy″ ,2015.

[Payne, Sorenson et Klitz, 05] : Payne. R. B., Sorenson M. D. and Klitz K, «The Cuckoos, Oxford University Press", 2005.

[Rajabioun, 11]:R. Rajabioun. "Cuckoo Optimization Algorithm, Applied Soft Computing", Vol. 11, N° 8,pp. 5508-5518, 2011.

[Ramdane, 06] : Ramdane Chafika, mémoire de magistère :"Le clustering des données : une nouvelle approche évolutionnaire quantique ", Université Mentouri de Constantine, Juin 2006.

[Reynolds et Frye, 07]: Reynolds, A. M. et Frye, M. A.: ″ Free-flight odor tracking in drosophila is consistent with an optimal intermittent scale-free search″. PloS one, 2007.

[Yang et Deb, 10]: Yang X. S. et Deb, S: "Engineering optimisation by cuckoo search". International Journal of Mathematical Modelling and Numerical Optimisation, 2010.

[Yang, 14]: Xin-She Yang, ″ Cuckoo Search and Firefly Algorithm″: Theory and Applications, Studies in Computational Intelligence, Vol. 516 Springer International Publishing, 2014.

[Bouria, kachid, Karboua, 08] : mémoire de master : "Hybridation entre le clustering possibiliste et le clustering évolutionnaire quantique " Université 20 aout skikda 55, 2008.

[B. Mirkin , 17]" Clustering for Data Mining": A data Recovery Approach, National Research University Higher School of Economics, 2005.

[M. R. Anderberg, 24] "Cluster Analysis for Applications", 1973.

[A. K. J. R C. Dubes, 19] "Algorithm for Clustering Data", 1988.

[C. Romesburg, 26] "Cluster Analysis for Researchers", 1984.

[C. a. Stephenson, 25] "An Introduction to Numerical Classification", 1975.

[Paunovic, M. and Schlesinger, M., 2006] "Fundamentals of Electrochemical Deposition. Second Edition", John Wiley & Sons, Inc., Toronto, 81.

[Blake et Merz, 98]: Blake, C.L. and Merz, C.J. (1998) "UCI repository of machine learning databases",1998.

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

[Cortez and Morais, 2007] P. Cortez and A. Morais, "A Data Mining Approach to Predict Forest Fires using Meteorological Data". In J. Neves, M. F. Santos and J. Machado Eds.," New Trends in Artificial Intelligence, Proceedings of the 13th EPIA 2007 - Portuguese Conference on Artificial Intelligence", December, Guimarães, Portugal, pp. 512-523, 2007.

[Hong, Z.Q. and Yang, J.Y, 91]"Optimal Discriminant Plane for a Small Number of Samples and Design Method of Classifier on the Plane", Pattern Recognition, Vol. 24, No. 4, pp. 317-324, 1991.

[Faroudja ABID et al, 2019, "Forest Fire in Algeria using Data Mining Techniques": Case Study of the Decision Tree Algorithm, International Conference on Advanced Intelligent Systems for Sustainable Development (AI2SD 2019) , 08 - 11 July , 2019, Marrakech, Morocco