



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

**UNIVERSITE ABOU-BEKR BELKAID – TLEMCCEN**

# THÈSE LMD

Présentée à :

FACULTE DES SCIENCES – DEPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

**DOCTORAT**

Spécialité: *Apprentissage artificielle et ingénierie de connaissances*

Par :

**Mr BELOUFA FAYSSAL**

Sur le thème

---

## **Conception d'un classifieur à base des règles floues**

---

Soutenue publiquement le 17/12/2016 à Tlemcen devant le jury composé de :

Mr Abdelhafid BESSAID	Professeur	Université de Tlemcen	Président
Mr Mohammed Amine CHIKH	Professeur	Université de Tlemcen	Directeur de thèse
Mr Ahmed LE HIRECH	Professeur	Université de Sidi Bel Abbès	Examineur
Mr Abdelatif RAHMOUN	Professeur	Université de Sidi Bel Abbès	Examineur
Mr Badr BENMAMMAR	MCA	Université de Tlemcen	Examineur
Mr Abdelkader BENYETTOU	Professeur	Université d'Oran	Examineur

*Laboratoire Génie Biomédical (GBM)*

# Résumé

L'interprétabilité représente l'élément le plus important derrière l'implémentation des classifieurs flous pour des problèmes d'application médicale. L'expert devrait être en mesure de comprendre le classifieur et d'évaluer ses résultats. Cette thèse présente deux nouvelles approches pour construire un classifieur flou interprétable. La première approche introduit une modification au niveau de l'algorithme de la colonie d'abeilles artificielles (ABC) dans lequel un opérateur de croisement mixte ( $BLX-\alpha$ ) de l'algorithme génétique est ajouté afin d'améliorer la diversité de l'algorithme standard ABC. Dans la deuxième approche, nous avons intégrés dans l'algorithme d'optimisation par essaim particulaires (OEP) le même opérateur de mutation utilisé dans l'approche précédente, en vue d'améliorer la diversité des solutions dans l'algorithme standard OEP. Les expérimentations sont réalisées sur deux bases de données médicales bien connues, le cancer du sein et le diabète. Les résultats obtenus avec les approches proposées sont très prometteurs par rapport aux précédentes recherches dans la littérature, pour les mêmes problèmes.

*Mots clés* : Algorithme d'optimisation, opérateur de croisement, interprétabilité, classifieur flou.

# Abstract

Interpretability represents the most important driving force behind the implementation of fuzzy classifiers for medical application problems. The expert should be able to understand the classifier and to evaluate its results. This thesis presents two novel approaches for Building an interpretable Fuzzy classifier. The first approach introduces a modified Artificial Bee Colony (ABC) algorithm in which a blended crossover operator (BLX- $\alpha$ ) of genetic algorithm is added in order to enhance the diversity of ABC. In the second approach, we have blended the standard Particle Swarm Optimization (PSO) algorithm with the same mutation operator of Genetic algorithm used in the first approach aiming at improving the diversity of PSO. Experiments are performed on two well-known medical database breast cancer and diabetes available in UCI machine learning repository. The obtained results in our approaches are very promising when compared with the previous research in the literature for the same problems.

*Keywords:* Optimization algorithm, crossover operator, Interpretability, Fuzzy classifier.

# REMERCIEMENTS

Si cette thèse a pu voir le jour, c'est certainement grâce à *ALLAH* le tout puissant qui m'a donné le privilège et la chance d'étudier et de suivre le chemin de la science et de la connaissance, et aussi au soutien et à l'aide de plusieurs personnes. Je profite de cet espace pour les remercier tous.

Au terme de ce travail de longue haleine, je tiens à remercier tous celles et ceux qui m'ont aidé de près ou de loin à mener à bien ma recherche, tous celles ou ceux qui étaient disponibles et ont participé à rendre plus humaine, plus riche et enrichissante cette longue expérience. Mes remerciements iront aussi bien à mes enseignants qu'au cadre administratif de notre département.

Je remercie tout d'abord mon directeur de thèse Professeur CHIKH Mohammed Amine, qui, tout au long de mon parcours, m'a appuyé et m'a encouragé à donner le meilleur de moi-même. Je lui exprime toute ma gratitude pour avoir réussi à stimuler continuellement mes capacités et mon potentiel. Il a su être réconfortant et généreux sans jamais laisser de côté la rigueur ni la précision qui incombent à la recherche.

Je tiens également à remercier Dr EL HABIB DAHO Mostafa pour ses conseils et son soutien moral et scientifique qui m'ont permis de mener à bien ce projet de thèse. Il a été toujours une source inépuisable d'idées, de savoir et d'encouragement.

Mes vifs remerciements sont aussi adressés à tous les membres du jury.

Je remercie également tous mes collègues de l'université de Tlemcen, Dr ATTAR Ahmed, Mr BAROUDI Yacine et Mr KHOVANE Amine pour leur collaboration et présence.

Enfin, je n'oublierais pas d'exprimer toute ma reconnaissance à certains membres de ma famille (ma chère et adorable mère, mon exceptionnel père, mes sœurs et mes frères) et à mes ami(e)s proches, qui m'ont soutenus, ne serait-ce que par le cœur, et qui ont su être présent(e)s quand j'ai eu besoin d'eux.

## *Dédicaces*

Par la grâce d'allah je dédie ce travail . . .

A ma douce Maman, ma conseillère, et amie fidèle, qui m'a assistée dans les moments difficiles, et qui a su me prêter l'attention, l'affection et l'amour qui m'ont permis d'écrire cette thèse dans de bonnes conditions. Tu représentes pour moi le symbole de la bonté par excellence. Tu n'as pas cessé de me soutenir, de m'encourager et de prier pour moi durant toutes les années de mes études, tu as toujours été présente à mes côtés pour me consoler quand il le fallait.

A mon très cher Père, les mots ne me suffisent pas pour exprimer l'attachement, l'amour l'affection et la profonde tendresse que je te porte pour ton soutien et tes encouragements. Tu as toujours répondu présent pour moi, tes actes étaient la lumière dans l'obscur des moments difficiles, et m'ont toujours donné la force de poursuivre mes études.

A mes chères sœurs que j'adore le plus dans ce monde Asma et Téma avec lesquelles j'ai partagé des moments difficiles, beaucoup d'émotion, de complicité dans la vie quotidienne, mes sentiments restent éternels.

Ainsi qu'à mes frères Farouk, Samir, Zakaria auprès de qui, j'ai souvent trouvé réconfort et soutien.

A mes beaux-frères Fethi et Fethallah.

A mes belle-sœurs Hadjer et Ikram.

A ma nièce Bouchra et mes neveux Khalil, Mehdi et Islem.

A Mlle BOUDERBANE Yasmina qui m'a tant aidé merci pour ton soutien moral et tes conseils.

A mes amis les plus proche Bedro, Hafid et Mohamed . . .

A toute la famille ;

Merci à Vous Tous.

## Table de matières

Résumé.....	i
Abstract.....	ii
Remerciment.....	iii
Dédicace.....	iv
Table de matières.....	v
Liste des Figures .....	ix
Liste des Tableaux.....	x

## Introduction Générale

Contexte général.....	1
Problématique.....	3
Contribution .....	4
Organisation du manuscrit.....	6

## Chapitre 1 : Apprentissage Automatique et classification

<b>1 Introduction.....</b>	<b>8</b>
<b>2 Intelligence Artificielle en médecine .....</b>	<b>8</b>
2.1 Systèmes d'Aide à la Décision (SADs) .....	8
2.2 Les Systèmes d'Aide au Diagnostic Médical (SADMs).....	9
<b>3 Apprentissage automatique.....</b>	<b>12</b>
3.1 L'apprentissage non supervisé .....	13
3.2 Apprentissage semi-supervisé.....	14
3.3 Apprentissage supervisé .....	15
3.3.1 La phase d'apprentissage.....	15
3.3.2 La phase de Test .....	15
<b>4 Classification automatique.....</b>	<b>17</b>
4.1 Définition.....	17
4.2 Problème de classification.....	18

<b>5 Evaluation des systèmes de classification .....</b>	<b>19</b>
5.1 Méthode HoldOut.....	19
5.2 Matrice de confusion.....	20
<b>6 Les algorithmes de classification .....</b>	<b>22</b>
6.1 K plus proches voisins (KNN K-nearst neighbours).....	22
6.2 Les arbres de décision.....	23
6.3 Machines à vecteurs supports (SVM).....	26
6.4 Approche des réseaux bayésiens .....	28
6.5 Le Système d'inférence flou .....	29
6.5.1 Notions de base et connaissances préliminaires .....	30
6.5.1.1 Notion d'ensemble et de sous ensemble flou .....	30
6.5.1.2 Les variables linguistiques .....	32
6.5.1.3 Règles linguistiques.....	33
6.5.1.4 Extension des opérations usuelles .....	34
6.5.2 Structure générale d'un système d'inférence flou.....	35
6.5.2.1 La base de connaissances.....	35
6.5.2.2 Fuzzification ou quantification floue.....	35
6.5.2.3 L'inférence floue .....	36
6.5.2.4 Défuzzification.....	38
<b>7 Principes d'optimisation des modèles flous .....</b>	<b>40</b>
<b>8 Conception des classifieurs flous dans la littérature .....</b>	<b>42</b>
<b>9 Conclusion .....</b>	<b>44</b>

## Chapitre 2: Optimisation des classifieurs

<b>1 Introduction.....</b>	<b>45</b>
<b>2 Concept d'optimisation.....</b>	<b>45</b>
<b>3 Les étapes du processus d'optimisation .....</b>	<b>46</b>
<b>4 Les méthodes d'optimisation.....</b>	<b>47</b>
<b>5 Les méta-heuristiques.....</b>	<b>47</b>
<b>6 Classification des méta-heuristiques.....</b>	<b>50</b>
6.1 Méthodes de trajectoire.....	51
6.1.1 Recuit Simulé .....	51

6.1.2 Recherche Tabou.....	52
6.2 Méthodes basées sur la population.....	53
6.2.1 Les algorithmes évolutionnaires.....	54
6.2.2 L'Intelligence en essaim.....	54
6.2.2.1 Caractéristiques générales de l'intelligence en essaim.....	56
6.2.3 La Colonie d'Abeilles Artificielles (ABC).....	57
6.2.3.1 Comportement de butinage.....	59
6.2.3.2 Principe fondamental de l'ABC.....	62
6.2.4 L'Optimisation par Essaim Particulaire (OEP).....	66
6.2.4.1 Les éléments de l'OEP.....	67
6.2.4.2 Principe fondamental de l'OEP.....	68
<b>7 Conclusion.....</b>	<b>71</b>

## **Chapitre 3: Résultats et discussion**

<b>1 Introduction.....</b>	<b>72</b>
<b>2 La maladie du cancer du sein.....</b>	<b>72</b>
2.1 Définition.....	72
2.2 La base de données du cancer du sein.....	73
2.3 Travaux de littérature sur la maladie du cancer du sein.....	75
<b>3 La maladie du diabète.....</b>	<b>76</b>
3.1 Définition.....	76
3.2 Base de données de diabètes.....	76
3.3 Traitement des valeurs manquantes.....	77
3.4 Travaux de littérature sur la maladie du diabète.....	77
<b>4 L'approche adaptée pour l'optimisation du modèle flou.....</b>	<b>78</b>
4.1 Le modèle des règles floues adapté.....	80
4.2 Codification de règles floues.....	81
4.3 L'initialisation de la Population.....	83
4.4 Fonction d'évaluation.....	83
<b>5 Modèle 1: conception d'un classifieur flou par l'algorithme de colonie d'abeilles artificielles (MABC).....</b>	<b>84</b>
5.1 Opérateur de croisement mixte : BLX- $\alpha$ .....	84

## Table de matières

5.2 Principe de l'algorithme ABC modifié (MABC) :.....	85
5.3 Présentation de l'algorithme ABC modifié (MABC).....	86
5.4 Résultats et discussion.....	87
5.4.1 Résultats des tests.....	88
5.4.2 La connaissance extraite par l'algorithme ABC modifié .....	88
5.4.3 Comparaison entre L'algorithme ABC et l'algorithme ABC modifié.....	93
<b>6 Modèle 2: conception d'un classifieur flou par l'algorithme d'optimisation par essaim particulaire (MPSO) :.....</b>	<b>95</b>
6.1 Principe de l'algorithme OEP modifié (MPSO) :.....	96
6.2 Présentation de l'algorithme d'OEP modifié .....	96
6.3 Résultats et discussion.....	97
<b>7 Comparaison de l'approche proposée avec les travaux de la littérature</b>	<b>100</b>
<b>8 Conclusion .....</b>	<b>102</b>
<b>Conclusion Générale.....</b>	<b>103</b>
<b>Bibliographie .....</b>	<b>105</b>

## Liste des Figures

<i>Figure 1.1: les composants d'un SADM</i> .....	12
<i>Figure 1.2: Apprentissage supervisé</i> .....	17
<i>Figure 1.3: Matrice de confusion</i> .....	20
<i>Figure 1. 4: Arbre de décision</i> .....	24
<i>Figure 1.5: problème linéaire</i> .....	26
<i>Figure 1.6: problème non-linéaire</i> .....	27
<i>Figure 1.7: les fonctions d'appartenances les plus utilisées.</i> .....	32
<i>Figure 1.8: variable linguistique.</i> .....	33
<i>Figure 1.9: Structure d'un système d'inférence flou</i> .....	35
<i>Figure 2.1: le processus d'optimisation</i> .....	46
<i>Figure 2 2: les éléments de base pour le comportement de butinage</i> .....	59
<i>Figure 2.3: La danse en rond qu'effectue l'abeille en fonction De la direction de la source de nourriture.</i> .....	60
<i>Figure 2.4: la danse frétilante</i> .....	61
<i>Figure 2.5: comportement des abeilles</i> .....	61
<i>Figure 2.6: Topologies GBEST (à gauche) et LBEST (à droite)</i> .....	67
<i>Figure 2.7: principe du déplacement d'une particule</i> .....	68
<i>Figure 3.1: les deux approches d'optimisation du classifieur flou par les méta-heuristiques</i> .	78
<i>Figure 3.2: Identification d'un classifieur flou par le processus d'optimisation des méta-heuristiques</i> .....	79
<i>Figure 3.3: Fonction d'appartenance de type trapèze</i> .....	81
<i>Figure 3.4: Fonction d'appartenance pour chaque variable d'entrée</i> .....	81
<i>Figure 3.5: La structure de codage de la solution</i> .....	83
<i>Figure 3.6: La convergence de l'algorithme ABC et l'algorithme ABC modifié pour la base de données diabète (sous-ensemble 2)</i> .....	94
<i>Figure 3.7: la Convergence du MPSO et OEP (cancer du sein (sous-ensemble 2))</i> .....	100
<i>Figure 3.8: la Convergence du MPSO et OEP (Diabètes (sous-ensemble 2))</i> .....	100

# Liste des Tableaux

<i>Tableau 2.1: L'ordre chronologique de l'élaboration de certaines méta-heuristiques populaires</i> .....	49
<i>Tableau 2.2: Classification des méta-heuristiques</i> .....	51
<i>Tableau 2.3: L'Algorithme Recuit Simulé</i> .....	52
<i>Tableau 2.4: L'Algorithme Recherche Tabou</i> .....	53
<i>Tableau 2.5: L'Algorithme génétique</i> .....	54
<i>Tableau 2.6: L'Algorithme de La Colonie d'Abeilles Artificielles (ABC)</i> .....	63
<i>Tableau 2.7: L'Algorithme de L'Optimisation par Essaim Particulaire (OEP)</i> .....	69
<i>Tableau 3.1: Attributs des données de la base de cancer de sein</i> .....	74
<i>Tableau 3.2: Attributs des données de la base Pima</i> .....	77
<i>Tableau 3.3: Algorithme d'initialisation basée sur l'opposition</i> .....	83
<i>Tableau 3.4: Algorithme d'opérateur de croisement mixte (BLX-<math>\alpha</math>)</i> .....	85
<i>Tableau 3.5: L'Algorithme de La Colonie d'Abeilles Artificielles modifié (MABC)</i> .....	86
<i>Tableau 3.6: Spécification des paramètres dans l'algorithme proposé</i> .....	87
<i>Tableau 3.7: Les résultats obtenus pour la base PIMA en utilisant 10 fois-validation croisée</i> .88	
<i>Tableau 3.8: les règles floues obtenues par le deuxième sous-ensemble de données pour la base PIMA</i> .....	89
<i>Tableau 3.9: représentation des points modaux.</i> .....	91
<i>Tableau 3.10: Matrices de confusion de l'algorithme ABC modifié</i> .....	91
<i>Tableau 3.11: Les résultats de la comparaison entre ABC et MABC</i> .....	93
<i>Tableau 3.12: Les résultats obtenus pour la base breast cancer en utilisant 10 fois-validation croisée</i> .....	95
<i>Tableau 3.13: L'Algorithme de L'Optimisation par Essaim Particulaire modifié (MPSO)</i> .....	96
<i>Tableau 3.14: spécification des paramètres dans l'algorithme MPSO</i> .....	98
<i>Tableau 3.15: les résultats obtenus par l'algorithme MPSO</i> .....	98
<i>Tableau 3.16: Les résultats de la comparaison entre PSO et MPSO</i> .....	99
<i>Tableau 3.17: taux de classification obtenu par notre classifieur et d'autres classifieurs de la littérature (base de données diabète)</i> .....	101
<i>Tableau 3.18: taux de classification obtenu par notre classifieur et d'autres classifieurs de la littérature (base de données cancer du sein)</i> .....	101

# Introduction Générale

## Contenu

---

Contexte général .....	1
Problématique.....	3
Contribution.....	4
Organisation du manuscrit.....	6

---

### Contexte général

Les experts du domaine médical sont intéressés par l'utilisation des systèmes d'aide au diagnostic médical, le besoin actuel est orienté vers les logiciels qui s'inspirent de l'intelligence humaine comme : le raisonnement, la prise de décision, l'apprentissage (par expérience) et bien d'autres. Durant les dernières années, la révolution numérique a permis le stockage d'une grande masse de données. Les bases de données médicales contiennent une grande quantité de données liée aux maladies et aux pathologies qui devraient être utilisées pour modéliser et améliorer les processus du diagnostic. Ces données forment une collection de valeurs brutes qui ne permettent pas d'autres informations d'établissement des décisions interprétables. Les techniques utilisées en intelligence artificielle ont encouragées les chercheurs pour développer des systèmes d'aide aux diagnostics intelligents (SAD). Ils seront en mesure de réaliser un diagnostic d'une manière automatique.

Les applications des SAD dans le domaine médical visent à aider les médecins dans la prise de décision clinique. Les médecins sont enclins à faire des erreurs de diagnostic, à cause de la fatigue et de la grande complexité des problèmes médicaux. Un médecin a besoin de se rappeler et d'utiliser une grande quantité de connaissances pour prendre une décision comme : les présentations de la maladie, les paramètres de diagnostic, des combinaisons de médicaments, des corrélations entre les maladies.... Etc. Cependant, les capacités cognitives du médecin sont limitées dues à des facteurs comme le multitâche, le raisonnement et la capacité limitée de la mémoire. Par conséquent, les médecins sont incapables de traiter une grande quantité de données et prendre les bonnes décisions tout le temps sans l'aide des systèmes automatisés.

L'un des principaux objectifs de la recherche en informatique biomédicale est l'élaboration des systèmes d'aide au diagnostic médical (SADM). Le développement de ces systèmes et l'avancement dans la recherche médicale s'accompagnent de l'utilisation de moyens informatiques de plus en plus puissants

visant à faciliter la tâche du médecin. Ainsi, le développement des systèmes à base de connaissances a permis le rapprochement entre les médecins, les ingénieurs biomédicaux et patients.

Ces systèmes ne sont pas faits pour remplacer les spécialistes ou les médecins, mais ils sont développés pour aider les praticiens dans le diagnostic et la prévision de l'état du patient en se basant sur certaines règles ou «expérience». Les premiers systèmes médicaux intelligents tels que MYCIN, PIP, Internist-I [1] ont démontré leurs performances par rapport à la pratique manuelle dans le domaine de l'aide au diagnostic médical.

L'objectif principal de cette thèse est de proposer une méthode transparente basée sur les techniques de l'Intelligence Artificielle (IA) pour l'automatisation et la reproduction du processus humain de prise de décision dans le but de réduire le coût, le temps, et l'erreur médicale.

Le raisonnement en intelligence artificielle concerne l'ensemble des techniques permettant la manipulation des connaissances déjà acquises afin de produire de nouvelles connaissances. L'intelligence artificielle n'est pas un nouveau concept, mais elle est considérée comme une nouvelle technologie en informatique qui tente de rendre les machines plus intelligentes. Elle a été appliquée dans plusieurs domaines comme l'éducation, le commerce, la médecine, ... etc. Une étape de base pour tout comportement intelligent est l'apprentissage. La plupart des chercheurs admettent aujourd'hui qu'il n'y a pas d'intelligence sans apprentissage. Par conséquent, l'apprentissage automatique [2, 3, 4, 5] est l'une des principales branches de l'intelligence artificielle et en effet, il est l'un des domaines les plus utilisées et qui fournit plusieurs outils indispensables pour une analyse intelligente des données.

# Problématique

L'apprentissage automatique est actuellement bien adapté pour l'analyse des données médicales. Il existe de nombreuses méthodes qui se basent sur l'intelligence artificielle et qui peuvent participer à la mise en place des systèmes d'aide au diagnostic médical. Ces systèmes sont d'un intérêt majeur pour les experts de la santé, mais ils souffrent de l'absence de l'interprétabilité dans leurs résultats.

Il devient donc nécessaire de créer des systèmes qui, en plus de proposer des prédictions précises, fournissent des explications compréhensibles pour l'expert du domaine. Sachant que le but n'est pas uniquement la modélisation du processus réel du raisonnement humain, mais aussi de fournir la connaissance derrière une décision de manière à favoriser la compréhension humaine. Comme beaucoup d'autres activités humaines, l'établissement d'un diagnostic implique l'utilisation des informations de natures imprécises, ambiguës et parfois incomplètes. Les systèmes développés devraient être en mesure de faire face à des situations pareilles, et encore fournir des prévisions précises et des explications utiles et nécessaires.

L'une des méthodes de l'apprentissage automatique qui traite l'incertitude dans les connaissances et simule le raisonnement humain pour les données incomplètes est l'approche floue. Cette approche a principalement deux caractéristiques qui permettent d'évaluer sa qualité:

- **L'interprétabilité** - Elle se réfère à la capacité du modèle flou à exprimer le comportement du système d'une manière compréhensible. Il s'agit d'une propriété subjective qui dépend de plusieurs facteurs, principalement la structure du modèle, le nombre de variables d'entrée, le nombre de règles floues, le nombre de termes linguistiques, et la forme des ensembles flous.
- **Précision** - Elle se réfère à la capacité du modèle flou à représenter approximativement le système modélisé. le terme approximation est

également utilisé pour exprimer la précision. Plus l'approximation au modèle est élevée, plus sa précision est augmentée. La proximité représente la similitude entre les réponses du système réel et le modèle flou.

Le but principal qui motive notre recherche est le développement d'un modèle intelligent basé sur la théorie floue dans le but de proposer un classifieur interprétable des données médicales.

### Contribution

La modélisation du système représente l'action et l'effet qui simplifient un système réel ou complexe dans le but de faciliter sa compréhension. Grâce à cette modélisation, le système réel peut être expliqué, contrôlé, simulé, prédit, et même amélioré. Le développement des modèles fiables et compréhensibles est l'objectif principal de la modélisation des systèmes.

L'approche proposée dans cette thèse combine la logique floue et les méthodes évolutionnaires dans le but de produire automatiquement un classifieur précis et interprétable. Elle est basée sur deux éléments:

- un système qui est capable de fournir à la fois la précision et **L'interprétabilité**
- un algorithme d'apprentissage supervisé qui permet la construction d'un tel système à partir des données disponibles.

L'une des motivations pour le choix des systèmes flous est leur transparence. Les systèmes flous peuvent fournir un bon schéma pour représenter la connaissance d'une manière qui s'approche du raisonnement humain. Les Systèmes flous présentent certaines caractéristiques qui les rendent adéquats pour résoudre un problème d'aide au diagnostic médical :

- une base de connaissances composée d'un ensemble de règles floues (exprimer d'une manière naturelle pour expliquer les processus de

décision) qui conceptualisent la connaissance du domaine (la base de règles BR), et les fonctions d'appartenance qui définissent les ensembles flous associés aux termes linguistiques utilisés dans les règles floues (la base de données -BD);

- une procédure d'inférence qui utilise la connaissance stockée pour formuler un mappage à partir des données d'entrée vers la sortie du modèle.

La construction des modèles flous à partir des données variées et complexes, exige l'identification de nombreux paramètres. Le problème général consiste à déterminer le nombre adéquat de règles, le placement optimal des fonctions d'appartenance sur l'espace d'entrée et la détermination des conclusions des règles, une fois la structure choisie. Dans beaucoup de cas la structure et les paramètres des systèmes flous sont déterminés empiriquement en choisissant a priori le nombre des fonctions d'appartenance et leur placement et en prenant toutes les combinaisons possibles. Cela peut conduire à une explosion combinatoire des exemples.

Afin de faire face à ce problème nous proposons dans le cadre de cette thèse une méthode évolutionnaire. Les techniques évolutionnaires sont souvent inspirées des systèmes biologiques afin de construire des systèmes dotés d'intelligence et possèdent l'efficacité des systèmes naturels. Les algorithmes évolutionnaires sont largement utilisés pour rechercher des solutions adéquates dans des espaces complexes qui résistent à des solutions analytiques. Il existe plusieurs modèles qui s'inspirent de la biologie tels que les réseaux de neurones, les algorithmes génétiques, les algorithmes de colonies de fourmis, les algorithmes d'optimisation par essais particuliers, les systèmes immunitaires artificiels, etc. Plus précisément, nous utilisons deux techniques évolutives comme outils d'apprentissage pour l'obtention d'un classifieur flou interprétable et précis à partir de données numériques : L'algorithme de la colonie d'abeilles artificiel et l'algorithme d'optimisation par essaim particulier.

Le travail que nous vous présentons dans cette thèse s'inscrit dans le contexte d'aide au diagnostic médicale. Nous donnons un intérêt plus particulier à l'apprentissage automatique structurel et paramétrique en utilisant les méthodes évolutionnaires, dans le but d'augmenter les performances du classifieur flou sur le plan précision et interprétabilité. La technique proposée est très efficace dans la conception des systèmes interprétables de haute précision et à résoudre des problèmes difficiles, en particulier la modélisation des processus d'aide au diagnostic médical.

Les deux principales contributions de ce travail sont comme suit :

- Augmenter les performances d'un classifieur flou des données médicales.
- Doter le classifieur des données d'une base de connaissances optimisées par des techniques évolutionnaires.

## Organisation du manuscrit

Afin de fournir au lecteur une vue d'ensemble du contenu de ce travail de thèse, nous décrivons brièvement le contenu de chacun de ses chapitres:

- Introduction générale présente le contexte général, la problématique et les contributions réalisées.
- Chapitre 1 : Est consacré à la présentation de l'intelligence artificielle et principalement l'apprentissage automatique supervisé. nous présentons aussi une synthèse sur les méthodes de classification les plus populaires utilisées. l'approche floue pour la classification supervisée est présentée en détail.
- Chapitre 2 : Présente le concept de l'optimisation. L'algorithme d'optimisation par essaim particulaire (OEP) et l'algorithme de colonie d'abeille artificiel ainsi que leur comportement social de recherche de nourriture.

## **Introduction Générale**

- Chapitre 3 : Est consacré à l'interprétation et les discussions des résultats, tout en décrivant les implémentations faites avec les résultats expérimentaux obtenus.
- En dernier lieu, une conclusion générale et des perspectives de ce travail de doctorat seront présentées.

# Chapitre 1

## Apprentissage Automatique et classification

---

### Contenu

---

1 Introduction .....	8
2 Intelligence Artificielle en médecine .....	8
2.1 Systèmes d'Aide à la Décision (SADs) .....	8
2.2 Les Systèmes d'Aide au Diagnostic Médical (SADMs).....	9
3 Apprentissage automatique .....	12
3.1 L'apprentissage non supervisé .....	13
3.2 Apprentissage semi-supervisé .....	14
3.3 Apprentissage supervisé .....	15
4 Classification automatique .....	17
4.1 Définition.....	17
4.2 Problème de classification.....	18
5 Evaluation des systèmes de classification .....	19
5.1 Méthode HoldOut.....	19
5.2 Matrice de confusion.....	20
6 Les algorithmes de classification .....	22
6.1 K plus proches voisins (KNN K-nearest neighbours) .....	22
6.2 Les arbres de décision.....	23
6.3 Machines à vecteurs supports (SVM) .....	26
6.4 Approche des réseaux bayésiens .....	28
6.5 Le Système d'inférence flou .....	29
7 Principes d'optimisation des modèles flous .....	40
8 Conception des classifieurs flous dans la littérature.....	42
9 Conclusion.....	44

---

# 1 Introduction

L'intelligence artificielle (IA) est l'un des domaines les plus récents de la science et de l'ingénierie. Les travaux sur ce sujet ont commencé réellement après la seconde guerre mondiale, et le nom lui-même a été inventé en 1956. IA comprend actuellement une grande variété de sous-domaines, comme jeux d'échecs, démonstration des théorèmes mathématiques, conduire une voiture dans une rue bondée, diagnostic des maladies...

L'apprentissage automatique est l'une des extensions de l'IA. Il fait référence au développement, à l'analyse et à l'implémentation des méthodes qui permettent à une machine d'évoluer grâce à un processus d'apprentissage [6], et ainsi de remplir des tâches difficiles ou impossible par des moyens algorithmiques plus classiques. Dans ce chapitre nous allons présenter le concept de l'apprentissage automatique supervisé, les différents types d'apprentissage automatique, les techniques utilisées pour évaluer les résultats de classifieurs et les algorithmes de classification couramment utilisés dans la littérature.

## 2 Intelligence Artificielle en médecine

### 2.1 Systèmes d'Aide à la Décision (SADs)

La prise des décisions concernant des systèmes complexes est souvent basée sur les capacités cognitives des êtres humains. Même si les interactions individuelles entre des variables d'un système peuvent être bien comprises, de prédire comment le système réagira à une manipulation est souvent difficile.

Dans de nombreuses situations la qualité des décisions est importante, l'aide des convenances du jugement humain et de prise de décision ont été une préoccupation majeure de la science à travers l'histoire. Les disciplines telles que la statistique, et les opérations de recherche développent diverses méthodes pour faciliter la prise de décision. Plus récemment, ces méthodes sont souvent renforcées par une variété de techniques issues de l'intelligence artificielle, qui a été mise en œuvre sous la forme de programmes informatiques, soit comme des

## Chapitre 1 : Apprentissage Automatique et Classification

outils libres ou comme des environnements informatiques intégrés. Ces environnements sont souvent appelés Systèmes d'Aide à la Décision (SADs).

Le concept des SADs est extrêmement large, et ses définitions varient, selon le point de vue de l'auteur. Bernard [7] a proposé la définition suivante :

*" The decision support is the activity that is supported on models clearly explained but not necessarily completely formalized, helps get answers to the questions asked by a intervener in a decision-making process ... "*

Pour éviter l'exclusion de tous types de SADs existants, nous allons les définir approximativement comme des systèmes interactifs basés sur machine qui aide les utilisateurs dans le jugement. Un autre nom parfois utilisé comme synonyme de SADs est systèmes à base de connaissances, qui se réfère à leur tentative de formaliser la connaissance du domaine afin qu'elle soit susceptible au mécanisme de raisonnement.

Les SADs concernent plusieurs domaines, y compris business, l'ingénierie, la médecine... Ils sont particulièrement utiles dans des situations où la quantité d'information disponible est grande pour l'intuition et la prise de décision humaine et dans lequel la précision et l'optimalité sont importantes.

Les SADs peuvent aider aussi les insuffisances cognitives humaines en intégrant diverses sources d'information, et fournir un accès intelligent aux connaissances pertinentes afin de d'aider le processus de prise de décisions. Ils peuvent aussi utiliser des méthodes d'intelligence artificielle pour répondre de manière heuristique à des problèmes inabordables. Une bonne application de ces systèmes augmente la productivité, l'efficacité et donne un avantage comparatif.

### 2.2 Les Systèmes d'Aide au Diagnostic Médical (SADMs)

L'utilisation d'outils informatiques et l'Intelligence artificielle pour l'aide au diagnostic médical a été débutée depuis un demi-siècle [8], Les SADMs sont des applications importantes pour la reconnaissance et l'analyse des données médicales, visant à aider les médecins pour prendre des décisions de

## Chapitre 1 : Apprentissage Automatique et Classification

diagnostiques [9]. Ils tendent à fournir aux cliniciens des informations utiles après avoir décrit la situation clinique du patient, afin de les aider à améliorer la qualité des soins. Les SADMs représentent un moteur d'induction qui apprend les caractéristiques de décision des maladies et peut ensuite être utilisé pour diagnostiquer des futurs patients avec des états pathologiques différentes.

Plusieurs définitions, du système d'aide au diagnostic, ont été proposées dans la littérature.

Sim et al. [10][11] ont proposé la définition suivante :

*« Software designed to be a direct aid to clinical decision-making, in which the characteristics of an individual patient are matched to a computerized clinical knowledge base and patient specific assessments or recommendations are then presented to the clinician or the patient for a decision ».*

Kawamoto et al [12] ont défini le système d'aide à la décision médicale comme suit :

*« We defined a clinical decision support system as any electronic or non-electronic system designed to aid directly in clinical decision making, in which characteristics of individual patients are used to generate patient-specific assessments or recommendations that are then presented to clinicians for consideration ».*

Dans [13] les auteurs proposent une autre définition :

*« Computer program whose purpose is to provide physicians with timely and useful information describing the clinical situation of the patient and appropriate knowledge of this situation, properly filtered and presented to improve the quality of care and patients' health ».*

Plusieurs algorithmes ont été développés pour construire des SADMs dans plusieurs spécialités médicales. Nous distinguons deux types de méthodes issues de l'intelligence artificielle : les méthodes symboliques (ou la procédure de classification produite peut être écrite sous forme de règles) et les méthodes non

## Chapitre 1 : Apprentissage Automatique et Classification

symboliques ou adaptatives (ou la procédure de classification produite est de type «boîte noire»)

Le principal inconvénient des méthodes non symboliques ceux qui ne sont pas basées sur un langage naturel est qu'ils ne fournissent aucune explication de la façon dont leur résultat d'inférence a été acquis. Au contraire, les méthodes symboliques ont comblé cette lacune en soutenant des descriptions linguistiques sous forme de règles, nous citons parmi ces méthodes les Systèmes flous. Les systèmes flous sont basés sur la théorie des ensembles flous et génèrent des règles linguistiques interprétables qui prennent la forme de Si-Alors. Ces règles représentent la connaissance dans les SADMs qui peut être facilement comprise par les médecins.

Au cours de ces dernières années, nous observons une utilisation croissante de la théorie des ensembles flous dans le domaine du diagnostic médical. Steimann [14] observe et analyse la croissance rapide de la recherche dans ce domaine. Évidemment, la théorie des ensembles flous répond efficacement à l'incertitude, qui représente un grand problème dans le domaine médical. Zadeh, l'instigateur et le créateur de la théorie des ensembles flous, a déclaré en 1969 que

***" ... les ensembles flous sont très susceptibles d'être appliqués dans le domaine du diagnostic médical ..." [15].***

Cela a été effectivement le début d'une série d'applications importantes, qui visent à contribuer au diagnostic dans le domaine médical.

Selon Ledley et Lusted [16], les SADMs ont généralement deux composants : une base de connaissances médicale et un moteur d'inférence. Les connaissances médicales sont composées de symptômes des maladies, les relations entre les symptômes et les relations entre les maladies. Le moteur d'inférence interprète essentiellement les données des patients en utilisant les connaissances médicales pour prendre une décision. La figure 1.1 montre les composants logiques d'un SADM.

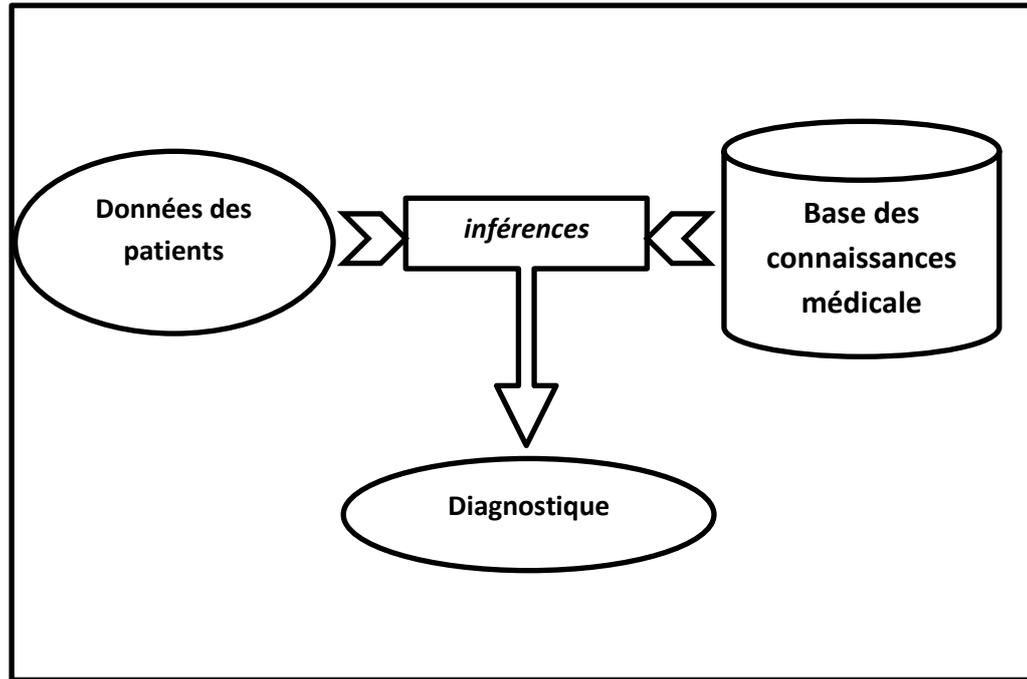


Figure 1.1: les composants d'un SADM

### 3 Apprentissage automatique

En terme général, l'apprentissage automatique concerne les programmes informatiques qui améliorent leurs performances dans certaines tâches grâce à des expériences [4], à savoir, les programmes qui sont capables d'apprendre.

Les domaines d'application de l'apprentissage automatique sont très nombreux, ils représentent une nouvelle discipline en plein développement. Dans des tâches bien déterminées, nous sommes capables de définir un ensemble d'exemples de données (couples d'entrées/sorties) mais pas de relations explicites et rapides entre les données. L'objectif principale de l'apprentissage automatique est de rendre les machines intelligentes, capables d'évoluer automatiquement grâce à l'expérience [4]. Afin d'atteindre cette objectif, l'apprentissage automatique rassemble un ensemble de techniques permettant de concevoir des modèles proches de la réalité à partir d'un ensemble d'exemples de données en entrée, soit en ajustant un modèle, soit en le générant complètement.

Actuellement, l'apprentissage automatique permet de réaliser plusieurs tâches différentes :

## Chapitre 1 : Apprentissage Automatique et Classification

- Analyse et traitement automatique des images.
- Interprétation et Reconnaissance de formes et de la parole humaine.
- Recherche d'information (moteur internet...).
- Découvrir les relations entre les données dans les grandes bases de données
- Faciliter l'exploitation des connaissances qui ne puissent pas être explicitées par le cerveau humain.
- aider les experts humains dans la prise de décisions complexes (ex :aide aux diagnostics médicaux).
- Etc...

Toute composante d'un système peut être améliorée par l'apprentissage automatique à partir d'une série de données. Les techniques utilisées pour leur amélioration, dépendent de quatre facteurs principaux:

- Quel composant doit être amélioré.
- Quelle connaissance préalable du système possède déjà.
- Quelle représentation est utilisée pour les données et le composant.
- Quel genre de réactions est disponible pour apprendre.

L'apprentissage automatique offre une grande variété d'approches qui peuvent être choisies pour capturer des tendances cachées à partir des données d'apprentissage [17]. Nous classons ces techniques en trois principales catégories :

- Apprentissage supervisé,
- Apprentissage non supervisé,
- Apprentissage semi-supervisé.

### 3.1 L'apprentissage non supervisé

Il est aussi appelé apprentissage sans enseignant ou l'apprentissage par corrélation. Ce type d'apprentissage est utilisé dans les cas où nous avons une base d'apprentissage dont les exemples de données ne sont pas étiquetés à l'avance. L'apprentissage non-supervisé consiste à regrouper les différentes formes en classes sur la base d'un critère de similarité choisi dans la technique.

## Chapitre 1 : Apprentissage Automatique et Classification

Ce type d'apprentissage permet la construction automatique des classes sans aucune intervention. Cependant, cette approche nécessite une bonne estimation du nombre de classes [18].

### ➤ *Estimation de densité*

Dans un problème d'estimation de densité, on cherche à modéliser convenablement la distribution des données. L'estimateur obtenu doit pouvoir donner une bonne estimation de la densité de probabilité à un point de test issu de la même distribution (inconnue) que les données d'apprentissage.

### ➤ *Partitionnement*

Un algorithme de partitionnement divise l'espace d'entrée en un certain nombre de "classes" en se basant sur un ensemble de données d'apprentissage ne contenant aucune information sur la classe de sortie. Les critères utilisés pour décider si deux points devraient appartenir à la même classe ou à des classes différentes sont spécifiques à chaque algorithme, mais sont très souvent liés à une mesure de distance entre points.

### ➤ *Réduction de dimensionnalité*

Le but d'un algorithme de réduction de dimensionnalité est de parvenir à "résumer" l'information présente dans les coordonnées d'un point en haute dimension par un nombre plus réduit de caractéristiques. Le but espéré est de préserver l'information "importante", de la mettre en évidence en la dissociant du bruit, et possiblement de révéler une structure sous-jacente qui ne serait pas immédiatement apparente dans les données d'origine en haute dimension.

## 3.2 Apprentissage semi-supervisé

L'Apprentissage semi-supervisé est un paradigme de l'apprentissage automatique concerné par l'étude de la façon dont les ordinateurs et les systèmes naturels tels que les humains apprennent, en présence des exemples de données étiquetées et non étiquetées.

L'Apprentissage semi-supervisé a un grand intérêt car il peut utiliser des exemples de données non étiquetées pour améliorer les tâches d'apprentissage supervisé lorsque les données étiquetées sont rares ou coûteux dans le but de comprendre comment la combinaison de données étiquetées et non étiquetées peuvent modifier le comportement de l'apprentissage [19].

### 3.3 Apprentissage supervisé

L'apprentissage supervisé est tout simplement une formalisation de l'idée d'apprendre à partir d'exemples. L'objectif de ce paradigme consiste à déterminer des règles typiques qui peuvent être utilisées pour différencier entre les données de différentes classes. Les approches de l'apprentissage supervisé utilisent un ensemble d'exemples de données pour construire un modèle, où chaque exemple de données a une étiquette de classe déterminée. D'où le travail d'apprentissage supervisé consiste à analyser les ressemblances entre les formes d'une même classe et les dissemblances entre les formes de classes différentes pour déduire les meilleures séparations possibles afin qu'il puisse identifier des nouveaux exemples avec une grande précision. [20].

Dans l'apprentissage supervisé, la construction et la génération d'un modèle passe par deux phases : *la phase d'apprentissage et la phase de test.*

#### 3.3.1 La phase d'apprentissage

La phase d'apprentissage représente l'étape de construction et de génération d'un modèle grâce à un algorithme qui prend en entrée un ensemble d'exemples de données qui contient toutes les informations nécessaires pour caractériser un problème donné.

#### 3.3.2 La phase de Test

Dans la phase de test le modèle généré doit être capable de prédire l'étiquette d'un nouvel exemple en fonction de leurs valeurs d'entrées.

## Chapitre 1 : Apprentissage Automatique et Classification

Généralement, l'ensemble d'exemples de données est divisé en deux sous-ensembles, un sous-ensemble de données d'apprentissage utilisé dans la *phase d'apprentissage* et un sous-ensemble de données de test utilisé dans la *phase de test* pour évaluer l'efficacité du modèle. L'évaluation dans ce cas dépend des données qui se trouvent dans les deux sous-ensembles, et donc elle peut être très différente en fonction de la façon de diviser les données. Afin d'éviter ce problème, une technique basée sur un échantillonnage statistique de l'ensemble de données est proposée. Cette technique est appelée validation croisée [21].

L'objectif principal de cette méthode est de s'assurer que chaque exemple de l'ensemble de données a la même chance d'apparaître dans les deux phases. La méthode de validation croisée est utile lorsque le nombre d'exemples de données n'est pas vraiment important. Le principe de fonctionnement de la méthode est résumé dans les points suivants:

- L'ensemble d'exemples de données est divisé en  $K$  sous-ensembles.
- La phase d'apprentissage utilise  $K-1$  sous-ensembles.
- La phase de test est effectuée en utilisant le sous-ensemble restant.
- La phase d'apprentissage et la phase de test sont répétées  $K$  fois jusqu'à ce que tous les sous-ensembles soient utilisés dans les deux phases.

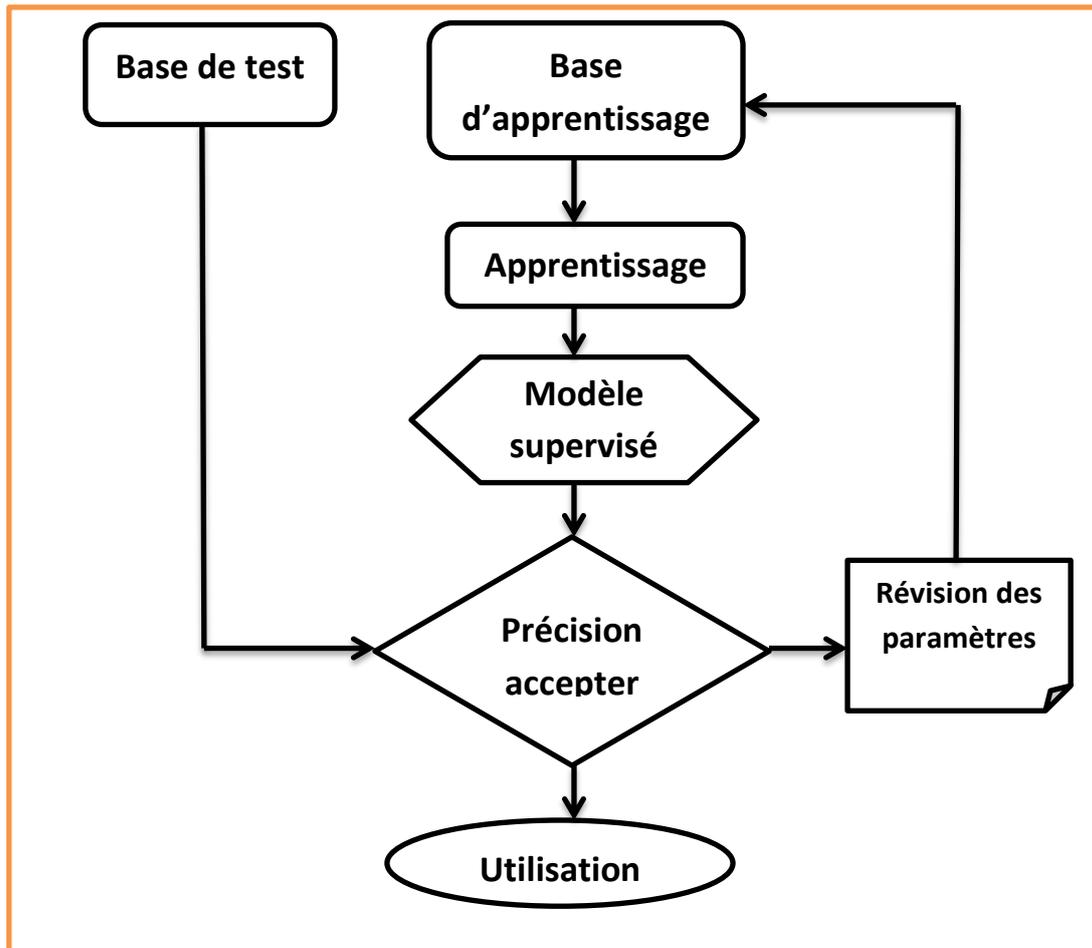


Figure 1.2: Apprentissage supervisé

## 4 Classification automatique

Plusieurs algorithmes sont utilisés dans l'apprentissage d'un classifieur. L'objectif de ces algorithmes est d'apprendre comment classer de nouveaux exemples de données correctement. La capacité de réussir cet objectif est appelé « capacité de généralisation ».

### 4.1 Définition

La classification est l'une des tâches de prise de décisions les plus fréquemment réalisées par les humains. Beaucoup de problèmes dans la science, l'industrie, la médecine ..., etc, peuvent être traités comme des problèmes de classification. Un problème se produit lorsqu'un objet doit être affecté à un groupe ou une catégorie prédéfinie en fonction des caractéristiques observées relatives à cet objet. Afin de

résoudre ce problème, la classification facilite le regroupement d'une population d'effectif important en plusieurs classes de telle sorte que les individus soient les plus semblables possible et que les classes soient les plus distinctes possibles à l'aide de différentes méthodes et approches dites intelligentes.

### 4.2 Problème de classification

Face à un problème particulier, soit  $X$  un ensemble d'échantillons composé des observations (ou autrement appelées, objets, instances, individus, etc). Cet ensemble est appelé l'espace d'exemples. Le nombre de dimensions,  $d$ , de cet espace est égal au nombre d'attributs  $A = \{a_1, a_2, \dots, a_d\}$  qui caractérisent chaque échantillon. Pour chaque attribut,  $a_i$  une valeur est attribuée  $x \in X_i$  ou  $X_i$  est l'ensemble de toutes les valeurs possibles pour l'attribut  $a_i$ .

Les échantillons peuvent être attribuées à une catégorie ou classe,  $y$ , qui divise l'espace d'exemple en différents groupes d'instances. Soit  $Y$  l'ensemble de toutes les classes valides des échantillons  $I$ . Un classifieur,  $C$ , est une application ou une fonction, à partir d'échantillons,  $X$ , vers les classes prédites,  $Y$  [22]:

$$C: X \rightarrow Y$$

Nous formulons maintenant le problème de classification; étant donné un exemple non classé,  $x \in X$  attribué à une classe  $y \in Y$ . Étant donné un classifieur,  $C$ , ceci peut être exprimé comme:

$$C(x) = y$$

Un échantillon,  $x$ , avec sa classe,  $y$ , est définie comme un échantillon classé. Un ensemble d'exemples d'échantillons classés est un ensemble,  $T$ , de paires, de telle sorte que chaque paire est constituée d'instance  $x \in X$  avec sa classe,  $y \in Y$  ordonné par l'indice de l'instance  $j$ :

$$T = \{(x_j, y_j) | x_j \in X, y_j \in Y, j = 1..n\}$$

### 5 Evaluation des systèmes de classification

L'objectif de notre analyse repose sur l'évaluation des classifieurs et des algorithmes d'apprentissage. L'évaluation des performances d'un système de classification est essentielle, car elle peut être utilisée dans l'apprentissage pour optimiser les valeurs des paramètres du classifieur, elle permet aussi de montrer à quel point un algorithme ou un classifieur est performant et elle pourrait aussi faciliter la sélection des algorithmes ou des classifieurs les plus appropriés pour un problème particulier. Il est soutenu que l'évaluation présente la clé pour une estimation plus fiable de la qualité de généralisation d'un classifieur. Des questions importantes généralement soulevées sont de savoir comment évaluer correctement les algorithmes d'apprentissage et comment évaluer la performance prédictive des classifieurs générés par ces algorithmes [23, 24, 25, 26, 27].

De nombreuses méthodes et techniques d'évaluation ont été proposées et nous soutenons qu'il est nécessaire de rassembler et d'évaluer ces méthodes afin d'être capable d'améliorer la qualité et la performance des algorithmes d'apprentissage et des classifieurs. Dans la section suivante, nous allons présenter un aperçu des différents critères d'évaluation utilisés dans notre thèse :

#### 5.1 Méthode HoldOut

Dans cette méthode [28], une fonction d'approximation est utilisée comme un estimateur pour généraliser la performance du modèle. Le modèle choisi est celui qui maximise la fonction d'approximation pour tout l'espace d'exemple. La fonction d'approximation utilise le Taux d'Erreur (TE) ou le Taux de Classification (TC) ( $TE=100-TC$ ) pour mesurer sa performance. TC est une mesure qui est simplement calculée comme suit:

$$TC = \frac{\text{Nombre d'exemple de données bien classés}}{\text{Nombre total d'exemple}}$$

Lors de l'évaluation, un classifieur a la possibilité de mal classer un exemple de donné. Les mesures précédentes ne donnent aucune information sur la nature de

cette erreur. Dans la plus part des cas d'application, il est très important de connaître cette nature des erreurs commises. Par exemple dans un système d'aide au diagnostic médical, un exemple de donnée qui représente un malade sain est considéré comme non sain est beaucoup plus dangereux qu'un exemple de donnée qui représente un malade non sain qui le considère comme sain.

L'un des outils les plus connus pour évaluer la performance d'un classifieur et indiquer la nature d'erreur est *La matrice de confusion*. L'accent est mis sur la capacité prédictive d'un classifieur plutôt que la vitesse nécessaire du modèle pour effectuer la classification.

### 5.2 Matrice de confusion

La matrice de confusion est une présentation tabulaire spécifique qui contient des informations sur les classifications réelles et prédites effectuées par un système de classification. Les performances de ces systèmes sont généralement évaluées en utilisant les données de la matrice. Chaque colonne de la matrice représente les instances d'une classe prédite tandis que chaque ligne représente les instances dans une classe réelle (ou vice-versa). [29] Le nom vient du fait qu'elle rend facile de voir si le système est une source de confusion entre deux classes. La figure suivante montre la matrice de confusion pour un classifieur à deux classes.

		Classes réels	
		Positives	Négatives
Classes prédites	Positives	VP Vrais positive	FP Faux positive
	Négatives	FN Faux négative	VN Vrais négative

Figure 1.3: Matrice de confusion

## Chapitre 1 : Apprentissage Automatique et Classification

Sachant que, par exemple, dans le problème de diagnostic médical :

- **VP** : représente le nombre de personnes malades classé malades.
- **FP** : représente le nombre de personnes non malades classé malades.
- **VN** : représente le nombre de personnes malades classé non malades.
- **FN** : représente le nombre de personnes non malades classé non malades.

À partir de la matrice de confusion, nous pouvons calculer des mesures statistiques comme la sensibilité et la spécificité.

- **La sensibilité :**

La sensibilité indique la probabilité que le test diagnostiquera correctement un cas, ou la probabilité qu'un cas donné sera identifié par le test. Par exemple, la sensibilité représente la proportion de personnes vraiment atteintes de la maladie, dans une population ciblée, qui sont identifiées par le test de dépistage comme étant atteintes de la maladie (c'est-à-dire qu'elles ont des résultats élevés). Ceci peut être exprimé comme suit:

$$\text{Sensibilité (se)} = \frac{\text{Vrai positives (VP)}}{\text{Vrai positives (VP)} + \text{Faux négatives (FN)}}$$

- **La spécificité :**

Un test spécifique en est un qui ne détecte que la maladie en cause, ainsi il a une portée restreinte, ce qui explique le terme « spécifique ». La **spécificité** représente la proportion de personnes *sans* la maladie qui a des résultats peu élevés sur le test de dépistage : la probabilité que le test identifiera correctement une personne n'étant pas atteinte de la maladie. Ceci peut être exprimé comme suit:

$$\text{Spécificité (sp)} = \frac{\text{Vrai négatives (VN)}}{\text{Vrai négatives (VN)} + \text{Faux positives (FP)}}$$

# 6 Les algorithmes de classification

Dans cette thèse, nous allons principalement considérer le problème de la classification supervisée. Dans ce qui suit, les principaux algorithmes de classification supervisée qui sont proposés dans la littérature seront présentés brièvement.

## 6.1 K plus proches voisins (KNN K-nearest neighbours)

L'algorithme K plus proches voisins est une méthode de classification basée sur les exemples d'apprentissage les plus proches dans l'espace de caractéristiques. Cette méthode devrait être parmi les premiers choix pour une étude de classification quand il y a peu ou pas de connaissance préalable de la distribution des données. K Plus proche voisin a été développé à partir de la nécessité d'effectuer une analyse discriminante lorsque les estimations paramétriques fiables des densités de probabilité sont inconnues ou difficiles à déterminer.

Étant donné un exemple de données de test  $X$  et un ensemble de données d'apprentissage  $\{x_i, y_i\}$ , la tâche du classifieur est de prédire l'étiquette de classe de  $X$  sur les classes prédéfinies. L'algorithme de classification KNN essaie de trouver les K plus proches voisins de  $X$  et utilise un vote majoritaire pour déterminer l'étiquette de classe de  $X$ . Sans connaissance préalable, le classifieur KNN applique habituellement une distance euclidienne comme la mesure de la distance. Cependant, cette méthode simple et facile à mettre en œuvre, elle peut encore donner des résultats compétitifs même par rapport à des méthodes d'apprentissage de la machine la plus sophistiquée.

La performance d'un classifieur KNN est principalement déterminée par le choix de K, ainsi que la mesure de la distance appliquée [30]. Cependant, il a été montré dans [31] que, lorsque les points ne sont pas répartis uniformément, la prédétermination de la valeur de K devient difficile. Généralement, les plus grandes valeurs de K sont plus immunisées du bruit. Par conséquent, en choisissant le même K (optimal) devient presque impossible pour des applications différentes.

## Chapitre 1 : Apprentissage Automatique et Classification

Comme indiqué précédemment, nous considérons chaque caractéristique de l'ensemble de données d'apprentissage comme une dimension dans l'espace, et les coordonnées de cette caractéristique sont représentées dans cette dimension, afin d'obtenir un ensemble de points dans l'espace. L'algorithme rassemble les points les plus semblables dans l'ensemble de données d'apprentissage et le choix d'une classe est basé sur les K points de données les plus proches de la nouvelle instance.

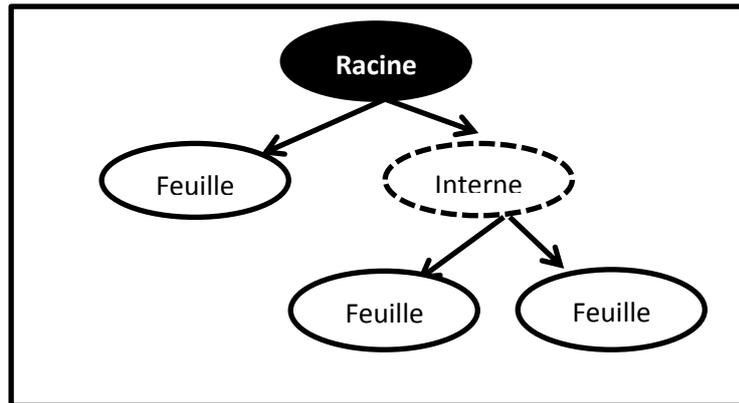
L'algorithme (comme décrit dans [32]) peut être résumé comme suit:

- Un entier positif  $k$  est spécifié, avec un nouvel échantillon.
- Nous sélectionnons les K exemples de données d'apprentissage qui sont les plus proches du nouvel échantillon.
- Nous trouvons la classe la plus commune de ces exemples.
- La classe la plus commune c'est la classe qui va être affectée au nouvel échantillon.

### 6.2 Les arbres de décision

Les arbres de décision sont utilisés avec succès dans beaucoup de domaines divers tels que la reconnaissance de caractères, la télédétection, diagnostic médical, systèmes experts, reconnaissance de la parole..., la caractéristique la plus importante de l'arbre de décision est leur capacité à décomposer un processus complexe en une collection de décisions simples, offrant ainsi une solution qui est souvent plus facile à interpréter.

L'arbre de décision se compose de nœuds qui forment un arbre de racine, ce qui signifie qu'il est un arbre dirigé avec un nœud appelé "root" qui n'a pas de bords entrants. Tous les autres nœuds ont exactement une arête entrante. Un nœud avec des bords sortants est appelé un nœud interne ou nœud de test. Tous les autres nœuds sont appelés feuilles (aussi connus comme nœuds terminaux ou décision).



*Figure1. 4: Arbre de décision*

Dans un arbre de décision, chaque nœud interne divise l'espace d'exemples en deux ou plusieurs sous-espaces selon une certaine fonction discrète de l'ensemble de données des attributs d'entrées. Dans le cas le plus simple et le plus fréquent, chaque test considère un attribut unique, de telle sorte que l'espace est divisé par exemple en fonction de la valeur de l'attribut. Dans le cas des attributs numériques, la condition porte sur un ensemble. Chaque feuille est affectée à une classe représentant la valeur la plus appropriée. Alternativement, la feuille peut détenir un vecteur de probabilité indiquant la probabilité de l'attribut cible ayant une certaine valeur. Les instances sont classées par leur navigation à partir de la racine de l'arbre jusqu'à la feuille, en fonction du résultat des tests au long de parcours. Les arbres de décision utilisés dans l'extraction de données sont de deux types principaux:

- **L'arbre de décision pour la classification** : est utilisé lorsque le résultat est une classe dans laquelle appartient la donnée.
- **L'arbre de décision pour la régression** : est utilisé lorsque le résultat prédit peut être considéré comme un nombre réel (par exemple la durée du séjour d'un patient dans un hôpital).

Les arbres utilisés pour la régression et les arbres utilisés pour la classification ont quelques similitudes mais aussi des différences, telles que la procédure utilisée pour déterminer l'endroit de séparation. Il existe de nombreux algorithmes spécifiques aux arbres de décision, les plus connus sont: ID3 (itératif Dichotomiser 3) [33], C4.5 (successeur de ID3) [35], CART (Arbre de Classification et de

## Chapitre 1 : Apprentissage Automatique et Classification

Régression) [34]. En général, les algorithmes de la construction des arbres de décision sont basés sur le choix d'une variable à chaque étape qui permet une bonne division pour l'ensemble des éléments [36]. Ce choix dépend de la métrique (logarithmique ou entropie de Shannon, Gini impureté, variance de la régression, ...) utilisée pour mesurer le meilleur score de l'attribut.

Dans ce qui suit nous citons quelques avantages et quelques inconvénients d'arbres de décision.

Les avantages d'arbres de décision comprennent :

- Simple à comprendre et à interpréter. Les arbres peuvent être visualisés.
- Nécessite un minimum de préparation de données. D'autres techniques nécessitent souvent la normalisation des données, des variables indicatrices doivent être créées et des valeurs vides doivent être enlevées. Notez cependant que ce module ne supporte pas les valeurs manquantes.
- Le coût d'utilisation de l'arbre (ie, la prédiction des données) est logarithmique selon le nombre de points de données utilisés pour former l'arbre.
- Capable de traiter à la fois des données quantitatives et qualitatives. D'autres techniques sont généralement spécialisées dans l'analyse des ensembles de données qui ont un seul type de variable.
- Capable de gérer des problèmes multi-sorties.
- Utilise un modèle de boîte blanche. Si une situation donnée est observable dans un modèle, l'explication de la condition est facile par la logique booléenne
- Possible de valider un modèle utilisant des tests statistiques.
- Fonctionne bien même si ses hypothèses sont peu violées par le vrai modèle à partir duquel les données ont été générées.

Les inconvénients d'arbres de décision comprennent:

- L'apprentissage des arbres de décision peut créer des arbres plus complexes qui ne généralisent pas bien les données. Ceci est appelé sur-

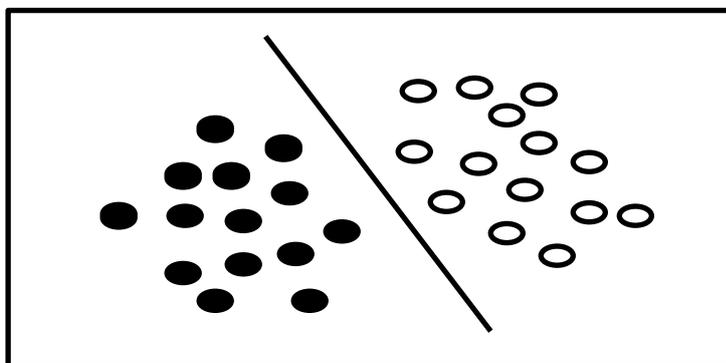
apprentissage. Des mécanismes tels que la taille fixant le nombre minimal d'échantillons requis pour un nœud de feuille et le réglage de la profondeur maximale de l'arbre sont nécessaires pour éviter ce problème.

- Les arbres de décision peuvent être instables parce que les petites variations dans les données pourraient générer un arbre complètement différent.
- Il y a des concepts qui sont difficile à apprendre parce que les arbres de décision ne les expriment pas facilement, comme XOR, la parité ou les problèmes de multiplexeurs.

### 6.3 Machines à vecteurs supports (SVM)

Les machines à vecteurs supports sont basées sur le concept de plans de décisions qui définissent les limites de la décision. Elles ont été introduites en 1995 par Cortes et Vapnik [37], [38]. SVM prend en charge les tâches de régression et de classification et peut gérer des variables quantitatives et qualitatives. Un plan de décision est celui qui sépare entre un ensemble d'objets appartenant à une classe. Un exemple schématique est représenté dans la figure 1.5.

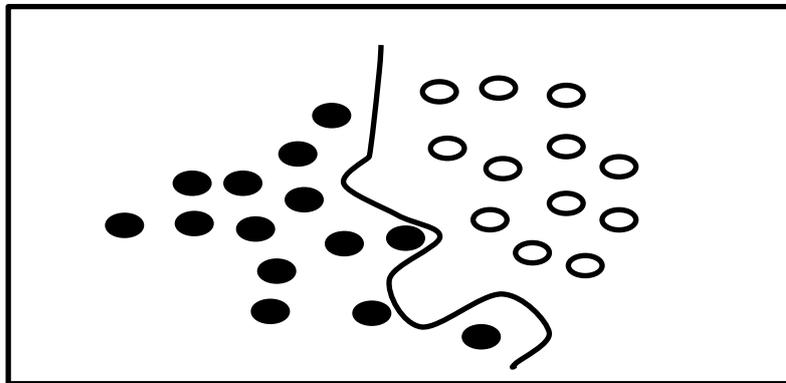
Dans cet exemple, les objets appartiennent soit à la classe verte ou rouge. La ligne de séparation définit une limite sur le côté droit de tous les objets qui sont dans la classe verte et à gauche tous les objets qui sont dans la classe rouge. Tout nouvel objet tombant vers la droite est classé comme objet vert (et pour classer comme objet rouge il devrait tomber à la gauche de la ligne de séparation).



*Figure1.5: problème linéaire*

## Chapitre 1 : Apprentissage Automatique et Classification

Ce qui précède est un exemple classique d'un classifieur linéaire, à savoir, un classifieur qui sépare un ensemble d'objets dans leurs groupes respectifs (vert et rouge dans ce cas) avec une ligne séparatrice. La plupart des tâches de classification ne sont pas aussi simple que cela, et souvent des structures plus complexes sont nécessaires afin de faire une séparation optimale. Cette situation est représentée dans la figure 1.6. Par rapport au schéma précédent (figure 1.5), il est clair que la séparation complète des objets verts et rouges exigerait une courbe (qui est plus complexe qu'une ligne), cela représente un cas d'un classifieur non-linéaire.



*Figure 1.6: problème non-linéaire*

Les Tâches de classification basées sur l'élaboration des lignes de séparation pour distinguer entre les objets de différentes classes sont connues comme classifieurs hyper-plans. Les machines à vecteurs supports sont particulièrement adaptées pour traiter ces tâches. Le SVM repose sur deux concepts: la marge maximale et la fonction du noyau (kernel function). La marge est la distance entre la limite de séparation et les exemples les plus proches appelés vecteurs de support. Dans un problème linéairement séparable, le SVM trouve un séparateur qui maximise la marge. Dans le cas d'un problème non linéaire une fonction du noyau est utilisée pour projeter les données dans un espace de grande dimension où elles sont linéairement séparables. Les avantages des machines à vecteurs de support sont:

- Efficaces dans des espaces de grande dimension.
- Encore efficaces dans les cas où le nombre de dimensions est supérieur au nombre d'échantillons.

## Chapitre 1 : Apprentissage Automatique et Classification

- Utilise un sous-ensemble de points de formation dans la fonction de décision (appelé vecteurs de support).
- Différentes fonctions du noyau peuvent être spécifiées pour la fonction de décision. il est également possible de spécifier des fonctions de noyaux personnalisées.

Les inconvénients de machines à vecteurs de support comprennent:

- Si le nombre d'attributs est beaucoup plus important que le nombre d'échantillons, le procédé est susceptible de donner des performances médiocres.
- le choix empirique d'une fonction de noyau approprié pour le problème.
- un temps de calcul couteux qui augmente selon le nombre de données à traiter

### 6.4 Approche des réseaux bayésiens

Le classifieur naïf bayésien est l'une des méthodes les plus simples en apprentissage supervisé basée sur le théorème de Bayes. L'algorithme de Bayes naïf est un algorithme intuitif qui utilise les probabilités pour faire une prédiction. Cette approche est utilisée généralement dans le cas d'un problème de modélisation prédictive probabiliste.

L'idée derrière un classifieur bayésien est que, si un agent connaît la classe, il peut prédire les valeurs des autres caractéristiques. S'il ne connaît pas la classe, la règle de Bayes peut être utilisée pour prédire la classe. Dans un classifieur bayésien, l'agent d'apprentissage construit un modèle probabiliste des caractéristiques et utilise ce modèle pour prédire la classification d'un nouvel exemple.

Les réseaux bayésiens simplifient le calcul des probabilités en supposant que la probabilité de chaque attribut appartenant à une classe donnée est indépendante de tous les autres attributs. La probabilité de la valeur d'un attribut pour une valeur de classe donnée est appelée la probabilité conditionnelle. La multiplication des probabilités conditionnelles simultanément de chaque attribut, donne la

probabilité d'une instance de données appartenant à cette classe. Pour faire une prédiction, nous pouvons calculer les probabilités des instances appartenant à chaque classe et sélectionner la valeur de classe avec la probabilité la plus élevée.

En général, le classifieur bayésien naïf fonctionne bien quand l'hypothèse d'indépendance est appropriée, lorsque la classe est un bon prédicteur des autres caractéristiques et les autres caractéristiques sont indépendantes de la classe donnée.

### 6.5 Le Système d'inférence flou

Dans Le monde réel, il est trop compliqué pour obtenir des descriptions précises et transparentes, donc les approximations doivent être mises en place afin d'obtenir un modèle raisonnable et interprétable. Par exemple l'homme perçoit, raisonne, imagine et décide à partir d'un modèle ou d'une représentation. Sa pensée n'est pas binaire, elle est basée sur un langage naturel. Aujourd'hui, la connaissance humaine devient de plus en plus importante. Nous avons besoin de formuler une théorie de la connaissance humaine d'une manière systématique que les modèles mathématiques et la mettre dans l'ingénierie des systèmes. Pour réaliser cette tâche, la question qui se pose est de savoir comment formuler la connaissance humaine.

L'idée est d'utiliser une technique qui permet de connaître l'imprécision de la pensée humaine et de l'exprimer avec des outils mathématiques appropriés. La méthode la plus connue qui répond à ce besoin est la logique floue.

La logique floue a été introduite par Zadeh en 1965 [39, 40]. Elle représente un paradigme de calcul qui fournit un outil mathématique basé sur la théorie des ensembles flous pour manipuler l'information d'une manière qui ressemble à des processus de raisonnement humain.

La théorie des ensembles flous généralise la théorie des ensembles classiques, cela signifie que le degré d'appartenance d'un objet à un ensemble n'est pas limité à l'entier 0 et 1, mais il peut prendre n'importe quelle valeur dans  $[0,1]$ . En élaborant

## Chapitre 1 : Apprentissage Automatique et Classification

la notion de sous-ensembles flous et les relations floues, nous pouvons définir des systèmes d'inférence floue (SIFs). Les SIFs sont des systèmes à base de règles dans lesquels une entrée est d'abord fuzzifiée puis traitée par un moteur d'inférence qui extrait des connaissances sous la forme de règles floues contenues dans une base de règles puis à la fin la sortie est defuzzifiée.

Aujourd'hui, les applications des systèmes flous sont très largement utilisées dans plusieurs domaines : la commande, traitement du signal, les communications, la fabrication de circuits intégrés, les systèmes experts pour les entreprises, la médecine, la psychologie, etc.

Pour une meilleure compréhension du fonctionnement de ces systèmes, nous présentons brièvement quelques notions de base.

### 6.5.1 Notions de base et connaissances préliminaires

#### 6.5.1.1 Notion d'ensemble et de sous ensemble flou

La théorie des ensembles est une théorie fondamentale dans les mathématiques, elle a été créée par le mathématicien allemand Georg Cantor où il a défini les ensembles comme des collections d'objets, de notre intuition ou de notre intellect, bien spécifiés et tous différents [41]

Dans son article " Fuzzy Sets " [42], L.A. Zadeh, le fondateur de la théorie des ensembles flous, commence par la définition de collection d'objets. Il donne comme exemple la collection des " nombres réels beaucoup plus grands que 1 ". Ce n'est pas un ensemble au même sens du terme qu'un ensemble de Cantor.

Cette collection est définie imprécisément et un ensemble flou peut très bien exprimer cette imprécision. Zadeh définit un ensemble flou comme : " *une collection telle que l'appartenance d'un élément quelconque à cette ensemble peut prendre toutes les valeurs entre 0 et 1* " [42].

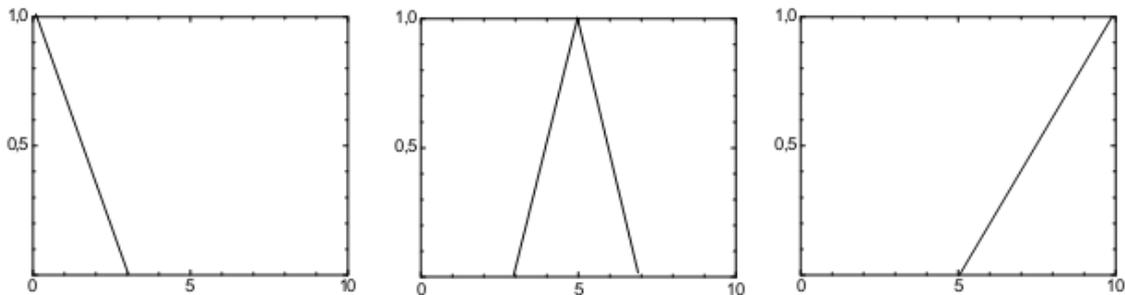
Par analogie avec les ensembles usuels, les ensembles flous permettent de définir des conditions (ou prédicats) graduelles pouvant intervenir dans des requêtes. Par même nature, un prédicat flou induit de la gradualité puisqu'il permet une

## Chapitre 1 : Apprentissage Automatique et Classification

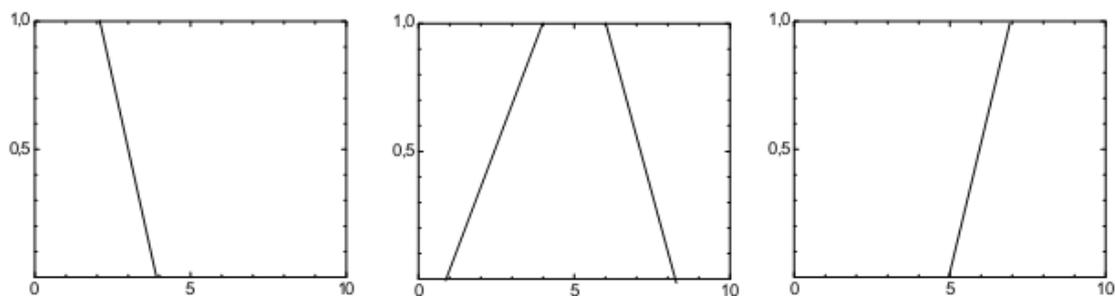
transition souple entre la pleine satisfaction et le rejet total. Par suite, chacun des résultats délivré par un tel prédicat est gradué et ordonné par rapport aux autres (sur l'échelle  $[0,1]$ ).

Les prédicats flous élémentaires correspondent aux adjectifs de la langue naturelle tels que " grand", "moyen " ou bien " jeune " qu'on appelle des variables linguistiques, et sont modélisées par une fonction (de forme trapézoïdale, triangulaire ou gaussienne. . .) d'un ensemble de domaines, cette fonction s'appelle fonction d'appartenance.

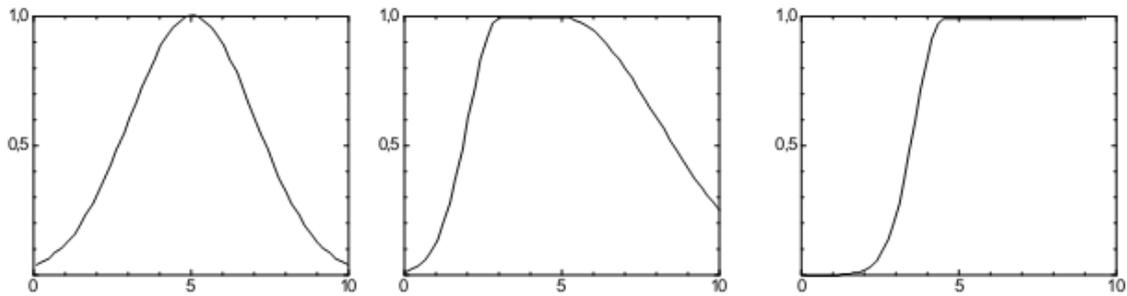
Nous pouvons résumer tous cela en disant que : les ensembles flous sont les mesures qui permettent de représenter et de manipuler des données imprécises et incertaines, et donc de rendre compte du concept de la vie courante, en reflétant notre mode de perception, qui n'était pas pris en compte jusque-là par les approches classiques. En logique classique, le fait d'appartenir à une classe exclut en général le fait d'appartenir à une classe différente. L'intérêt de l'approche floue vient du fait qu'un élément peut appartenir simultanément à plusieurs classes, à des degrés divers.



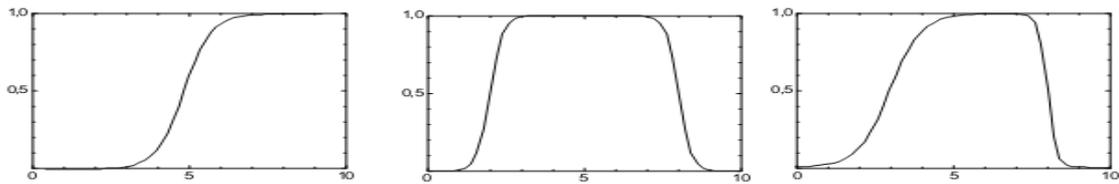
(a) Fonctions d'appartenance triangulaires



(b) Fonctions d'appartenance trapézoïdales



(c) Fonctions d'appartenance Gaussiennes



(d) Fonctions d'appartenance sigmoïdes

*Figure 1.7: les fonctions d'appartenances les plus utilisées.*

### ➤ Appartenance graduelle à un ensemble

La logique floue permet de caractériser une appartenance graduelle à un sous ensemble, appelé sous ensemble flou. On peut dire en cela que la théorie des sous-ensembles flous et la logique floue sont une généralisation de la théorie des ensembles classiques et de la logique booléenne, mais fondées sur un nouvel opérateur d'appartenance qui est multi-value au lieu d'être binaire.

#### 6.5.1.2 Les variables linguistiques

La description d'une certaine situation, d'un phénomène ou d'un procédé contient en général des expressions floues comme :

- *Quelque, beaucoup, souvent,*
- *Chaud, froid, rapide, lent*
- *Grand, Moyen, petit*
- *etc.*

Les expressions de ce genre forment les valeurs des variables linguistiques de la logique floue. D'où une variable linguistique représente un état dans le système à

régler ou une variable de réglage dans un contrôleur flou. Sa valeur est définie dans des termes linguistiques d'un langage naturel ou artificiel.

➤ **Description des variables linguistiques par des fonctions d'appartenance**

Chaque sous ensemble flou peut être représenté par sa fonction d'appartenance. Si l'ensemble de référence est un ensemble fini, les valeurs d'appartenance sont des valeurs discrètes définies dans  $[0,1]$ . Si l'ensemble de référence est fini, on peut représenter ces valeurs comme des fonctions d'appartenance continues.

En général, la forme de fonctions d'appartenance dépend de l'application, et peut être monotone (croissante ou décroissante), triangulaire, trapézoïdale ou en forme de cloche (gaussienne)

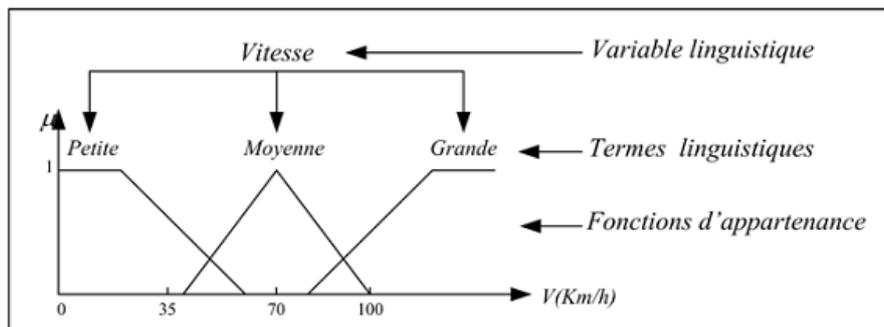


Figure 1.8: variable linguistique.

### 6.5.1.3 Règles linguistiques

L'idée principale des systèmes basés sur la logique floue est d'exprimer la connaissance humaine sous la forme de règles linguistiques de forme **Si... alors...**. Chaque règle a deux parties :

- Partie antécédente (prémisse ou condition), exprimée par **Si...**
- Partie conséquente (conclusion), exprimée par **alors...**

La partie antécédente est la description de l'état du système. La partie conséquente exprime l'action que l'opérateur qui contrôle le système doit exécuter. Chaque

règle floue est basée sur l'implication floue. Zadeh [43] a été le premier qui a introduit la notion de règle floue sous la forme

***Si x est A, alors y est B.***

La forme générale d'une règle *Si... Alors...* est :

***Si (Un ensemble de conditions est satisfait) Alors***

***(Un ensemble de conséquences peut être exécuté).***

### 6.5.1.4 Extension des opérations usuelles

Les opérations de l'algèbre relationnelle peuvent être étendues de façon mutuelle aux relations floues d'une part en les considérant comme des ensembles flous (pour ce qui est des opérations ensemblistes), et d'autre part en introduisant des prédicats graduels au lieu des seuls prédicats booléens. Il existe de nombreuses variantes dans ces opérateurs. Cependant, les plus répandues sont ceux dites " de Zadeh " décrites ci-dessous. En considérant deux relations  $A$  et  $B$  définies sur le même ensemble d'attributs  $X$ , et le degré de vérité d'une proposition  $A$  sera noté  $\mu(A)$ :

- **Union :**

$$\mu_{A \cup B}(x) = \text{Max}(\mu_A(x), \mu_B(x))$$

- **Intersection :**

$$\mu_{A \cap B}(x) = \text{Min}(\mu_A(x), \mu_B(x))$$

- **Complémentaire :**

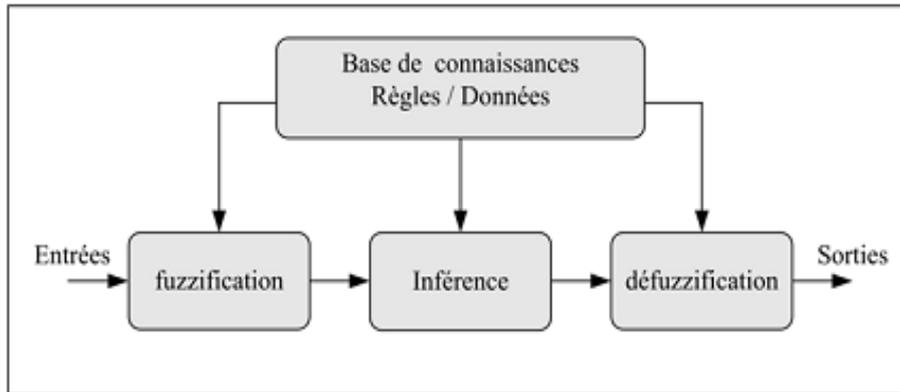
$$\mu_{\bar{A}} = 1 - \mu_A$$

- **Différence :**

$$\mu_{A-B}(x) = \text{Min}(\mu_A(x), 1 - \mu_B(x))$$

### 6.5.2 Structure générale d'un système d'inférence flou

Le fonctionnement interne d'un système d'inférence flou repose sur la structure présentée par la figure suivante qui inclut quatre blocs:



*Figure 1.9: Structure d'un système d'inférence flou*

- La base de connaissances
- La fuzzification des variables d'entrée, avec éventuellement un prétraitement de l'information.
- L'inférence à partir d'une base de connaissances.
- La défuzzification, avec éventuellement un post-traitement de l'information.

#### **6.5.2.1 La base de connaissances**

Elle contient les définitions des fonctions d'appartenance (formes et paramètres) associées aux variables d'entrée/sortie ainsi que l'ensemble des règles floues.

#### **6.5.2.2 Fuzzification ou quantification floue**

Les opérateurs utilisés dans la commande floue agissent sur des sous-ensembles flous. Par conséquent, il est nécessaire de transformer les variables non floues provenant du monde extérieur en des sous-ensembles flous. Pour se faire, on utilise un opérateur dit de fuzzification qui associe à une mesure de la variable  $x_0$  une fonction d'appartenance particulière  $\mu_{x_0}(x)$ . Cette première étape consiste à la conversion analogique/digitale, ainsi que du traitement des grandeurs mesurées

et de leurs transformations en variables linguistiques avec la définition des fonctions d'appartenances.

La fuzzification détermine le degré d'appartenance de chaque variable d'entrée à chaque état. Celui-ci est déterminé à l'aide des fonctions d'appartenance définies dans le système. Ces fonctions d'appartenances, qui utilisent en général entre trois et sept états flous pour chaque variable, possèdent différentes formes : en triangles, en cloche (gaussienne) ou encore en trapèze.

Donc la fuzification consiste à :

- mesurer des variables d'entrées ;
- traiter des grandeurs mesurées ;
- Attribuer un ensemble de degrés d'appartenance à chaque valeur d'entrée.

### 6.5.2.3 L'inférence floue

Le mécanisme d'inférence consiste d'une part à calculer le degré de vérité des différentes règles du système et d'autre part à associer à chacune de ces règles une valeur de sortie. Cette valeur de sortie dépend de la partie conclusion des règles qui peut prendre plusieurs formes selon le type de système d'inférence flou. Il y a plusieurs sortes d'interférences floues, on s'intéressera à l'étude de deux systèmes d'inférence flous, les plus utilisés actuellement, ceux du type **Takagi-sugeno** et ceux du type **Mamdani**.

#### ➤ Méthode de Takagi-sugeno (TKS)

Chaque fonction d'appartenance de la sortie est une combinaison linéaire de valeurs d'entrée. La sortie précise est la moyenne pondérée des poids d'activation et des sorties de fonctions d'appartenance. La méthode de TKS, où les fonctions d'appartenance sont des valeurs précises, constitue un cas particulier de cette méthode [44]. On parle parfois de méthode de TKS simplifiée ou d'ordre zéro quand la conclusion est une constante. Pour un vecteur d'entrée  $X = (x_1, x_2, \dots, x_n)^i$  la sortie inférée est obtenue par :

## Chapitre 1 : Apprentissage Automatique et Classification

- *Calcul du degré d'appartenance de chaque entrée aux différents sous ensemble flous,*

$$\mu_{Aj}^i(x_i)$$

- *Calcul de la valeur de vérité de chaque règle,*

$$\alpha_i(x) = ET \left( \mu_{A1}^1(x_1), \dots, \mu_{An}^i(x_n) \right)$$

- *Calcul de la sortie du SIF,*

$$y = \frac{\sum \alpha_i f(x_1, \dots, x_n)}{\sum \alpha_i}$$

La sortie obtenue n'est pas floue, ce qui supprime une étape dans l'inférence. Les SIF de type TKS permettent donc le passage aisé d'une expression symbolique (la base de règles) à sa traduction numérique.

### ➤ *Méthode de Mamdani*

L'identification du modèle flou Mamdani s'effectue soit d'une manière analytique comme le cas du modèle Takagi-Sugeno soit en intégrant une connaissance à priori. Par exemple, un expert peut définir la partition des entrées et un certain nombre de règles. Le formalisme explicite du modèle permet d'unifier les règles définies à partir des données et celles formulées à partir d'une connaissance.

C'est un avantage quand les mesures, ou l'expertise, ou les deux sont incomplètes. Cependant, pour approcher le comportement global d'un système par un modèle flou Mamdani, un grand nombre de règles est nécessaire et le nombre de paramètres à identifier augmente.

Malgré cet inconvénient, si un ensemble de règles linguistiques complet et compact est identifiable (toutes les situations sont traduites dans la base de règles), la précision du modèle flou Mamdani est correcte voire supérieure à celle des modèles Takagi-Sugeno.

## Chapitre 1 : Apprentissage Automatique et Classification

Pour un vecteur d'entrée  $X = (x_1, x_2, \dots, x_n)^i$  la sortie inférée est obtenue par :

- **Calcul du degré d'appartenance de chaque entrée aux différents sous ensemble flous,**

$$\mu_{Aj}^i(x_i)$$

- **Calcul de la sortie du SIF,**

$$\alpha_i(x) = ET \left( \mu_{A1}^1(x_1), \dots, \mu_{An}^i(x_n) \right)$$

- **Calcul de la valeur de vérité de chaque règle,**

$$\alpha_i(x) = OU \left( \mu_{A1}^1(x_1), \dots, \mu_{An}^i(x_n) \right)$$

En conclusion, les modèles flous Takagi-Sugeno et Mamdani sont deux représentations complémentaires et non concurrentes. Le choix d'une structure dépend de l'objectif recherché avec une précision fixée à l'avance qui peut être soit au niveau de la simulation du modèle, soit la connaissance explicite des relations liant les entrées et les sorties du système.

### 6.5.2.4 Défuzzification

En sortie, le système flou ne peut pas communiquer des valeurs floues qu'il peut seul exploiter. Il lui est donc nécessaire de fournir des valeurs précises, c'est le rôle de la défuzzification. Cette étape s'effectue toujours à l'aide des fonctions d'appartenance. A partir des degrés d'appartenance, on obtient autant de valeurs qu'il y a d'états. Pour déterminer la valeur précise à utiliser, il y a plusieurs méthodes proposées :

#### ➤ **Méthode du centre de gravité**

C'est une des méthodes les plus utilisées. Lorsque la fonction  $\mu(z)$  est discrétisé, le centre de gravité est donnée par :

$$Z = \frac{\sum_{i=1}^n \mu_i Z_i}{\sum_{i=1}^n \mu_i}$$

## Chapitre 1 : Apprentissage Automatique et Classification

Où  $n$  est le nombre de niveaux de quantisation,  $z_i$  la valeur de sortie pour le niveau  $i$  et  $\mu_i$  sa valeur d'appartenance [45].

### ➤ *Méthode de moyenne de maximum*

Cette méthode génère une commande précise en calculant la moyenne des valeurs pour lesquelles l'appartenance est maximale. Si la fonction est discrétisée la valeur de défuzzification est donnée par :

$$Z = \sum_{i=1}^l \frac{r_i}{l}$$

Où  $l$  est le nombre de valeurs quantifiées  $r$  pour lesquelles l'appartenance est maximale [46].

### ➤ *Méthode de Tsukamoto*

Si l'on utilise les fonctions d'appartenance monotones et non décroissantes pour la sortie, la valeur précise de la commande est calculée comme suit :

$$Z = \frac{\sum_{i=1}^n w_i z_i}{\sum_{i=1}^n w_i}$$

Où  $n$  est le nombre de règles activées avec le poids  $w_i > 0$  et  $z_i$  la valeur de la fonction de sortie pour la règle  $i$  [47]

### ➤ *Méthode de moyenne pondérée*

Cette méthode est utilisée lorsque les sorties sont définies comme fonction linéairement dépendantes d'entrées [46]. Si  $w_i$  est le poids d'activation de la règle  $i$ , la valeur précise de la commande est :

$$Z = \frac{\sum_{i=1}^n W_i f(x_i, y_i)}{\sum_{i=1}^n W_i}$$

Où  $n$  est le nombre de règles activées.

### 7 Principes d'optimisation des modèles flous

Ces derniers temps, de nombreuses techniques d'apprentissage automatique ont été considérées pour la conception du système dit intelligent. Récemment les modèles flous ont été appliqués avec succès à des problèmes de classification [47, 48, 49]. Un système de classification fondé sur les systèmes d'inférence flou est un cas particulier de la modélisation floue où la sortie du système est discrète.

Cette thèse se concentre spécifiquement sur l'utilisation de la méthode de modélisation floue pour détecter des maladies et des pathologies bien ciblées qui repose sur la découverte de la connaissance humaine. La logique floue est proposée à l'origine par Zadeh [39], qui visait à améliorer la classification et les systèmes d'aide à la décision en utilisant des ensembles flous pour définir les classes qui se chevauchent. Les systèmes de classification flous sont capables de modéliser les incertitudes et ont donc un potentiel pour une bonne performance dans les applications de classification complexes. Leur application améliore également l'interprétabilité et fournit une meilleure compréhension de la connaissance dans la structure de classifieur et le processus de prise de décision [50]. La performance du système de classification flou dépend des règles flous "si-alors" et leurs nombres. Ces règles sont générées à partir de données numériques ou des expériences humaines.

Fondamentalement, la conception de tels systèmes consiste à construire des fonctions d'appartenance et de trouver un ensemble de règles floues appropriées pour le système de classification. La création de modèle flou est une tâche basée sur l'identification des paramètres et de structure du système d'inférence floue de sorte qu'un comportement désiré est obtenue. En général, l'identification du modèle flou concerne les points suivants [51] [52]:

- Sélection du type de modèle flou
- Sélection des variables d'entrée et de sortie du modèle

## Chapitre 1 : Apprentissage Automatique et Classification

- Identification de la structure du modèle flou, qui comprend la détermination du nombre et le type de fonctions d'appartenance des variables d'entrée et de sortie et le nombre de règles floues
- Identification des paramètres pour chacune des fonctions d'appartenance au niveau des prémisses et de conséquence des règles floues.

Dans les premiers temps, les systèmes flous dans la plupart des applications sont générés par des experts du domaine, notamment pour des problèmes de contrôle avec seulement quelques entrées.

Avec l'augmentation du nombre de variables, le nombre de règles est augmenté de plus en plus de façon exponentielle ce qui rend plus difficile pour les experts de définir l'ensemble de règles pour une bonne performance du système. Ce problème peut être abordé grâce à un processus d'apprentissage automatique basé sur les échantillons d'apprentissage.

Plusieurs méthodes d'apprentissage ont été développées pour l'identification du modèle flou comme les algorithmes de clustering ou les méthodes de partitionnements d'espace en plusieurs sous-espaces.

Les modèles flous peuvent être formulés également comme un problème d'optimisation dans un espace de recherche où chaque point représente un modèle flou avec différentes structures et paramètres [53, 54]. Cela rend les méthodes évolutionnaires des bons candidats pour l'identification des modèles flous en raison de leur puissante capacité de recherche globale.

On rencontre souvent, dans la littérature, trois stratégies d'application des méthodes d'optimisation pour la conception des systèmes d'inférence floue.

- La base de règles floues est bien définie et leurs fonctions d'appartenances sont optimisées par une méthode d'optimisation.
- Les fonctions d'appartenance associées aux variables d'entrées et de sorties sont fixées et la méthode est utilisée pour l'optimisation des conclusions des règles floues.

- Les fonctions d'appartenance et les règles floues associées sont optimisées simultanément

### 8 Conception des classifieurs flous dans la littérature

De nombreuses méthodes intelligentes, telles que les approches heuristiques [55], les réseaux de neurones [56, 57, 58], les méthodes de clustering [59, 60, 61] ont été proposés pour une construction optimale de classifieurs flous, mais depuis quelques années les Algorithmes Evolutionnaires (AEs) ont été aussi largement utilisés pour optimiser les Classifieurs flous. Dans la littérature plusieurs AEs comme les algorithmes génétiques (GAs), l'optimisation par essaim particulaire (OEP), l'optimisation par colonie de fourmis (ACO) ont été proposés pour produire des systèmes de classifications flous.

Les algorithmes génétiques sont des algorithmes de recherche qui suivent les principes de l'évolution biologique et génétique pour évaluer un ensemble de solutions d'un problème donné. L'idée de base est de conserver une population de chromosomes (représentant les solutions candidates à des problèmes spécifiques) qui évolue au fil du temps à travers un processus de concurrence et de variation [62]. Les GAs ont été prouvés à fournir une robuste capacité de recherche dans des espaces complexes théoriquement et empiriquement, et ils offrent un mécanisme de recherche performant et efficace pour résoudre des problèmes d'optimisation complexes.

Depuis quelques années les algorithmes génétiques (GAs) ont été généralement utilisés pour optimiser les Classifieurs flous [63, 64, 65, 66, 67, 68, 69]. La plupart des études utilisent les GAs pour générer des règles floues descriptives [70, 71, 72, 73, 74]. Les GAs ont été appliqués pour obtenir des règles floues lorsque les partitions floues sont déterminées à l'avance. Ces approches ne peuvent pas garantir une bonne solution parce que les règles floues et les fonctions d'appartenance floue sont dépendantes. D'autres travaux ont adopté les AGs pour extraire des règles floues [75, 76, 77], où chaque règle a sa propre définition de

## Chapitre 1 : Apprentissage Automatique et Classification

fonctions d'appartenance. Ces approches optimisent les fonctions d'appartenance floue et les règles floues simultanément, ce qui conduit à une recherche difficile dans un espace plus grand.

Généralement, deux approches d'apprentissage génétique ont été proposées: l'approche Pittsburgh [78, 79] et l'approche Michigan [80, 81]. La première considère tout l'ensemble de règles comme un chromosome; ou une population d'ensembles de règles candidat est évolué à travers des opérateurs génétiques, et finalement l'ensemble de règles optimal est conservé. Les classifieurs flous suivant l'approche de Pittsburgh ont été étudiés dans [73, 68, 78, 69]. L'approche Michigan considère une seule règle comme un chromosome et l'ensemble de règles est représenté par la population. Ishibuchi [82] a présenté un classifieur flou basé sur l'approche Michigan.

Récemment, certains chercheurs ont élaboré plusieurs Classifieurs génétiques à base de règles floues (GFRBCs) suivant un nouveau modèle d'apprentissage, l'approche itérative [75, 71, 76]. Dans le modèle itératif, chaque chromosome représente une seule règle comme dans l'approche du Michigan, mais la population ne représente pas l'ensemble complet des règles; une nouvelle règle est obtenue et ajoutée à l'ensemble des règles dans chaque itération de l'algorithme génétique.

Ganji et al. [83] utilisent l'algorithme d'optimisation par colonie de fourmis (ACO) pour générer des règles de classification floues appelé FSC-ANTMINER. Ils ont utilisé des fourmis artificielles afin d'explorer l'espace de recherche progressivement pour établir des règles floues candidates. Dans [84] Sanie et al. utilisent un algorithme génétique hybride pour produire des règles floues cet algorithme a été renforcé par une hybridation avec colonie de fourmis (ACO) en se basant sur une recherche locale pour améliorer la qualité de leur système de classification.

D'autre part, des approches fondées sur l'optimisation par essaim particulière (OEP) ont prouvé leur efficacité lors de l'optimisation du système de classification flou. Dans [85] Sousa et al. Proposent une première application d'OEP comme un nouvel outil pour la découverte de règles de classification. Holden et Freitas [86]

ont utilisé une optimisation hybride entre l'optimisation par essaim particulaire et l'optimisation par colonies de fourmis afin de générer une base de règles floues. Rani et Deepa [87] ont proposé une approche d'optimisation par OEP pour concevoir un classifieur flou optimal appelé PSOFLC. Les expériences sont réalisées sur la base de données IRIS, où la technique proposée a été comparée à deux techniques de classification bien connues, y compris classifieur flou génétique et classifieur flou gaussien. Une conception d'un système flou à partir des données d'entrées-sorties disponibles basées sur l'algorithme OEP a été proposée dans [88] et la méthode détaillée de codage a été également fournie. Khosla et al. [89] ont comparé les efforts computationnels d'OEP et GA en utilisant un système similaire au système présenté dans [88]. [82] qui a démontré une approche efficace basée sur OEP pour construire une base de règles floues à partir d'un ensemble d'exemples de données. Cette méthode suppose que les systèmes flous peuvent également être formulés comme un problème d'espace, où chaque point correspond à un système flou représenté par des fonctions d'appartenance, base de règles et donc de comportement du système correspondant.

## 9 Conclusion

Dans ce chapitre, nous avons présenté le concept de l'apprentissage automatique supervisé, les différents types d'apprentissage automatique, les techniques utilisées pour évaluer les résultats de classifieurs et les algorithmes de classification couramment utilisés dans la littérature. Nous avons aussi montré les notions et les connaissances préliminaires sur la théorie des ensembles flous, la structure interne du système d'inférence flou ainsi que le problème d'identification de ce système. Dans le chapitre suivant, nous présentons et nous analysons les différentes méthodes d'optimisation basées sur les méta-heuristiques utilisées dans la génération des systèmes d'inférence flous.

# Chapitre 2

## Optimisation des classifieurs

### Contenu

---

1 Introduction .....	45
2 Concept d'optimisation.....	45
3 Les étapes du processus d'optimisation.....	46
4 Les méthodes d'optimisation .....	47
5 Les méta-heuristiques .....	47
6 Classification des méta-heuristiques.....	50
6.1 Méthodes de trajectoire.....	51
6.2 Méthodes basées sur la population.....	53
7 Conclusion.....	71

---

# 1 Introduction

L'optimisation est un processus spontané qui fait partie de notre vie. Dans un sens plus simple, elle peut être définie comme une technique qui sélectionne la meilleure solution parmi un ensemble de solutions donné. Dans ce chapitre, nous présentons le concept d'optimisation, les différentes méthodes d'optimisation et plus particulièrement les méthodes d'optimisation basées sur l'intelligence en essaim. Les méta-heuristiques Colonie d'Abeille Artificielle (ABC) et Optimisation par Essaim Particulaire (OEP) sont décrites.

# 2 Concept d'optimisation

Les problèmes d'optimisation apparaissent dans diverses disciplines telles que les systèmes de production, l'économie, les sciences, la reconnaissance des formes, etc. Ces problèmes peuvent être divisés en plusieurs catégories selon : continues ou discrètes, avec contrainte ou sans contrainte, mono ou multi-objectif, statique ou dynamique.

En fait, les techniques d'optimisation sont utilisées pour optimiser un ou plusieurs objectifs dans certaines situations complexes qui peuvent être représentées par un modèle mathématique lorsque les décisions doivent être prises. Donc elle peut être considérée comme l'un des principaux outils dans le système de prise de décision.

Le terme optimisation peut être défini comme:

- Rendre aussi parfait, efficace, ou fonctionnelle que possible.
- Rendre optimale; tirer la meilleure partie de; mieux utiliser.

En mathématiques, l'optimisation est un moyen de trouver la meilleure solution à un problème. Cette solution est considérée comme une solution acceptable (ou satisfaisante), qui doit être absolument la meilleure par rapport à toutes les solutions de candidats.

### 3 Les étapes du processus d'optimisation

La figure 2.1 présente les trois étapes du processus d'optimisation : analyse, synthèse et évaluation. Tout d'abord, il convient d'analyser le problème et d'opérer un certain nombre d'éléments de base. Engelbrecht définit dans [90] les éléments de base dans un problème d'optimisation:

- Une fonction d'objectif qui représente la quantité à optimiser, c'est à dire, la quantité qui doit être minimisée ou maximisée.
- Un ensemble de variables inconnues qui affectent la valeur de la fonction objective, le nombre de variables qui définit la dimension du problème.
- Un ensemble de contraintes qui limitent les valeurs qui peuvent être attribuées à des variables. La plupart des problèmes ont au moins un ensemble de contraintes qui définissent la plage de valeurs de chaque variable.

Une fois effectuées ces différentes opérations, la méthode choisie synthétise des solutions potentielles qui sont évaluées. Puis éliminées jusqu'à l'obtention d'une solution acceptable. Si nécessaire, le problème peut être redéfini à partir des solutions déjà obtenues.

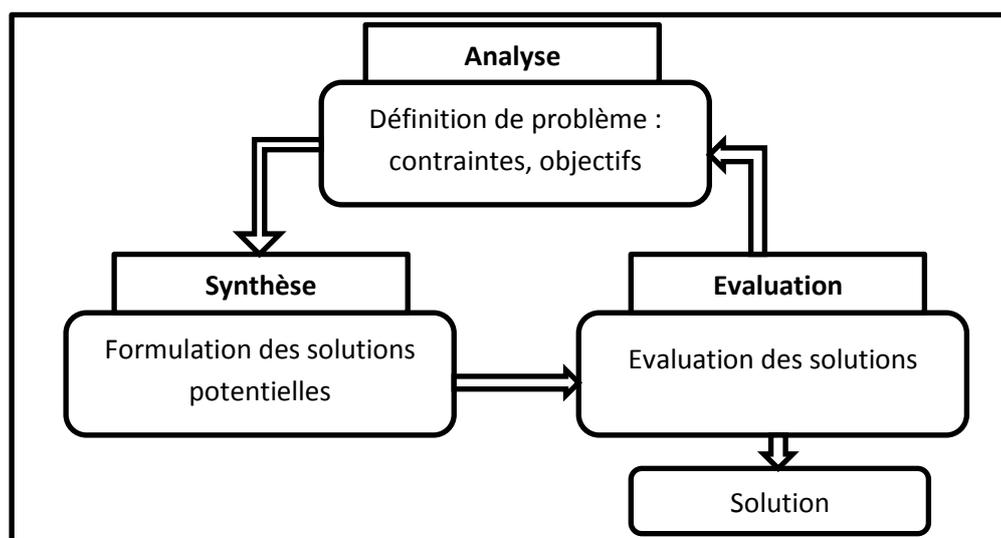


Figure 2.1: le processus d'optimisation

### 4 Les méthodes d'optimisation

Une fois définie la fonction à optimiser, il s'agit de choisir une méthode adaptée au problème posé. Les méthodes d'optimisation peuvent être classées de différentes manières : nous les classerons en méthodes déterministes et méthodes non déterministes. Les problèmes d'optimisation difficiles ne peuvent être résolus ou garantie l'optimalité par une méthode exacte (déterministe) dans un délai raisonnable. En vue de l'utilité pratique de ces problèmes, il est nécessaire de développer des algorithmes efficaces et robustes, qui peuvent résoudre sur des machines les problèmes d'optimisation (de petit ou de grande taille) qui se posent dans différents domaines.

Michalewicz et Fogel [91] décrivent les caractéristiques qui rendent difficile à résoudre un problème d'optimisation dans le monde réel:

- Le nombre de solutions possibles (espace de recherche) est trop grand.
- la complexité du problème.
- la variation de La fonction d'évaluation qui décrit la qualité de chaque solution dans l'espace de recherche.
- Les solutions possibles sont très limitées, ce qui rend difficile de générer au moins une solution réalisable (satisfaire les contraintes du problème).

Afin de trouver des solutions satisfaisantes à ces problèmes, les méta-heuristiques peuvent être utilisées comme des méthodes d'optimisation non déterministes.

### 5 Les méta-heuristiques

Les Méta-heuristiques sont une grande famille de méthodes d'optimisation non déterministes visant à trouver des solutions précises à des problèmes d'optimisation complexes lorsque les méthodes exactes (déterministes) ne sont pas applicables. Les Méta-heuristiques ne peuvent pas, en général, garantir de trouver des solutions optimales, mais ils ont tendance à produire des solutions proche à l'optimale avec un relativement faible effort de calcul.

## Chapitre 2 : Optimisation des classifieurs

En effet, le préfixe grec « méta », présent dans le nom, est utilisé pour indiquer que ces algorithmes heuristiques « sont de niveau plus élevé », à la différence des heuristiques spécifiques à un problème. Les Méta-heuristiques sont généralement appliquées à des problèmes pour lesquels il n'existe pas d'algorithme spécifique pour les résoudre. Ils sont largement utilisés pour résoudre des problèmes complexes dans divers domaines. La majorité des méta-heuristiques partagent les caractéristiques suivantes [92]:

- Ils sont inspirés de la nature (sur la base des principes de la physique, la biologie ou l'éthologie).
- Ce sont des stratégies générales ou des modèles qui guident le processus de recherche.
- Leur schéma de base a une structure prédéfinie
- Leur objectif est de fournir une exploration efficace de l'espace de recherche pour trouver des solutions (proche) optimales.
- Ils ne sont pas algorithmes exacts et leur comportement est généralement non déterministe.
- Ils utilisent des composants stochastiques (impliquant des variables aléatoires).
- Ils peuvent intégrer des mécanismes pour éviter la visite des régions non prometteuses de l'espace de recherche.
- Ils n'utilisent pas le gradient pour la fonction objectif.
- Ils ont plusieurs paramètres qui doivent être réglés dans le problème à résoudre.

Nous essayons d'annoncer certaines étapes qui ont marqué l'histoire des méta-heuristiques. L'ordre chronologique de l'élaboration de certaines méta-heuristiques populaires inspirées de la nature est donné dans le tableau 2.1. Ces

## Chapitre 2 : Optimisation des classifieurs

algorithmes ont fait leurs preuves dans la résolution de problèmes d'optimisation complexes qui se posent dans divers domaines.

2010	Artificial Bee Colony Algorithm
2001	Harmony Search Algorithm
	- Honey Bee Algorithm
	- Bacterial Foraging Algorithm
2000	Estimation of Distribution Algorithm
	- Particle Swarm Optimization
1995	- Differential Evolution
	Ant Colony Optimization Algorithm
1990	Tabu Search
1985	Simulated Annealing
1980	Genetic Algorithm
1975	Evolution Programming
1970	Evolution Strategies
1965	

**Tableau 2.1: L'ordre chronologique de l'élaboration de certaines méta-heuristiques populaires**

Au cours des trente dernières années, un grand intérêt a été consacré aux méta-heuristiques. Plusieurs algorithmes d'optimisation globale qui sont inspirés de la nature ont été développés. Ils sont également appelés algorithmes d'usage général en raison de leur applicabilité à une grande variété de problèmes. Certains algorithmes d'optimisation sont populaires comme : les algorithmes génétiques (AG) [93], l'optimisation par essaim particulaire (OEP) [94], évolution différentielle (ED) [95], la programmation évolutive (PE) [96], l'optimisation par colonie de fourmis (OCA) [97], etc

En se basant sur plusieurs caractéristiques intéressantes, il a été observé que ces algorithmes ne fonctionnent pas toujours selon les attentes. Le succès de la plupart des méta-heuristiques dépend en grande partie de l'équilibre entre deux objectifs contradictoires, l'exploration (diversification) et l'exploitation (intensification). Bien que l'exploration est importante de s'assurer que chaque partie du domaine de recherche de solution est suffisante pour fournir une estimation fiable de l'optimum global; l'exploitation, d'autre part, focalise l'effort de la recherche au voisinage des meilleures solutions trouvées pour atteindre d'autres meilleures solutions [98]. Les algorithmes de recherche pour atteindre ces deux objectifs en utilisant des méthodes de recherche locale, ou des approches de recherche globale, ou une hybridation des deux stratégies globales et locales.

## 6 Classification des méta-heuristiques

Blum et Roli [92] proposent un nombre d'approches pour classer les méta-heuristiques selon certains critères. Une des taxonomies les plus reconnues divise les méta-heuristiques dans les méthodes de trajectoire (*recherche basée sur un point unique*) et les méthodes basées sur la population (*recherche basée sur la population*). Ce choix est justifié par le fait que cette catégorisation permet une description plus claire des algorithmes. De plus, la tendance actuelle est l'hybridation dans le sens de l'intégration des algorithmes à point unique dans les algorithmes basés sur la population. Le tableau 2.2 contient une liste des méta-heuristiques selon cette classification.

Méthodes basées sur la population	Méthodes de trajectoire
<p>Méthodes évolutionnaires :</p> <ul style="list-style-type: none"> <li>- Algorithme génétique (AG)</li> <li>- stratégie d'évolution (ES)</li> <li>- Evolution différentielle (ED)</li> </ul> <p>L'Intelligence en essaim:</p> <ul style="list-style-type: none"> <li>- Optimisation par essaims particulières (OEP)</li> <li>- Algorithme Artificiel des colonies d'abeilles (ABC)</li> <li>- L'optimisation par colonies de fourmis (ACO)</li> </ul>	<ul style="list-style-type: none"> <li>- Recherché Tabou (TR)</li> <li>- Result Simulé (RS)</li> <li>- Recherche local basic</li> </ul>

*Tableau 2.2: Classification des méta-heuristiques*

## 6.1 Méthodes de trajectoire

Dans cette section, nous présentons les méthodes dites de trajectoire. Le processus de recherche effectué par les méthodes de trajectoire manipulent une seule solution à la fois et tente itérativement d'améliorer cette solution. Il construit une trajectoire dans l'espace de recherche en tentant de se diriger vers des solutions optimales.

### 6.1.1 Recuit Simulé

L'algorithme de recuit simulé (RS) [99, 100] est une heuristique probabiliste qui reproduit le procédé de recuit utilisé dans la science des matériaux. Durant ce processus un matériau est chauffé à des températures élevées, ce qui provoque des atomes qui se déplacent à partir de leurs positions initiales et de façon aléatoire à travers des états élevés de l'énergie. Lorsque la température du matériau est réduite lentement, les atomes s'établissent dans une nouvelle configuration qui possède une énergie plus faible. Traduire ce processus à un problème d'optimisation, l'état initial peut être considéré comme un minimum local. L'échauffement de la matière se traduit par le remplacement de la solution courante avec une nouvelle solution aléatoire. La nouvelle solution peut être

acceptée selon une probabilité fondée sur la diminution de la valeur de la fonction objective et sur une mesure «température», ce qui diminue lentement comme itérations continues. Le paramètre de la température permet aux solutions d'être acceptées, il peut avoir une valeur objective plus élevée afin d'éviter les minima locaux. La structure de base d'un algorithme SA est présentée dans le tableau suivant [101] :

**Tableau 2. 3: L'algorithme Recuit Simulé**

```

Begin
  Select an initial state  $i \in S$  and initial temperature  $T > 0$  ;
  Set temperature change counter  $t := 0$ ;

  While (bf not termination condition) do
    Begin
      Set repetition counter  $n := 0$ ;
      Repeat
        Generate state  $j$ ,  $a$  randomly chosen neighbor of  $i$ ;
        Calculate  $\delta = f(j) - f(i)$ ;
        If  $\delta < 0$  then  $i \leftarrow j$ 
        Else if  $\text{random}(0, 1) < \exp(-\delta/T)$  then  $i \leftarrow j$ 
         $n \leftarrow n + 1$ ;
      Until  $n = N(t)$ 
    End
  End

```

### 6.1.2 Recherche Tabou

La méthode Tabou (RT), formellement proposée par Glover en 1989 [102], est une heuristique de recherche locale qui travaille avec d'autres algorithmes pour dominer les limitations d'optimalité locale. Elle est appliquée à des problèmes d'optimisation combinatoire qui sont de nature discrète. Pour décrire le processus de recherche tabou, nous choisissons une solution initiale  $x \in X$  où  $X$  est un ensemble réalisable. Nous laissons  $S(x)$  l'ensemble des mouvements qui bougent  $x$  à un point extrême adjacent. Soit  $T \subset S$ , où  $T$  est l'ensemble des mouvements de tabou. L'ensemble  $T$  est déterminé par une fonction qui utilise l'information précédente du processus de recherche jusqu'à  $t$  itérations avant l'itération courante. Pour déterminer l'appartenance à  $T$ , il peut y avoir une liste détaillée ou un ensemble de conditions de tabou, c'est à dire,

$$T(x) = \{s \in S : s \text{ est contraire aux conditions de tabu}\}$$

La structure de base d'un algorithme RT est présentée dans le tableau 2.4 [102] :

<b>Tableau 2.4: L'Algorithme Recherche Tabou</b>
<pre> Begin   Select an initial <math>x \in X</math>;   <math>x^* := x, T := \emptyset, k \leftarrow 0</math>;   Begin     If <math>S(x) - T</math> is empty       Stop     Else       Set <math>k \leftarrow k + 1</math>;       Select <math>s_k \in S(x) - T</math> such that         <math>s_k(x) = \mathbf{Optimum}(s(x) : s \in S(x) - T)</math>       End       Let <math>x := s_k(x)</math>       If <math>f(x) &lt; f(x^*)</math>         <math>x^* = x</math>       End       Check stopping conditions       Update <math>T</math>;     End   End End </pre>

## 6.2 Méthodes basées sur la population

Les Méthodes basées sur la population traitent à chaque itération de l'algorithme un ensemble (c'est à dire, une population) de solutions plutôt qu'une solution unique. Comme elles portent sur une population de solutions, les algorithmes basés sur la population constituent un moyen naturel pour l'exploration de l'espace de recherche. Le rendement final dépend fortement de la manière dont la population est manipulée.

Les méthodes basées sur la population la plus étudiée dans l'optimisation sont : les méthodes évolutionnaires et les méthodes basées sur l'intelligence en essaim. Dans les algorithmes évolutionnaires, une population d'individus est modifiée par les opérateurs de recombinaison et de mutation, et dans l'intelligence en essaim une population est basée sur le comportement collectif des insectes sociaux.

### 6.2.1 Les algorithmes évolutionnaires

*“Evolutionary computation has as its objective to mimic processes from natural evolution, where the main concept is survival of the fittest: the weak must die.” A. Engelbrecht [103].*

Les algorithmes évolutionnaires sont une classe d'algorithmes basés sur la population. Ils sont utilisés dans nombreux domaines d'optimisation. Ces méthodes utilisent des techniques qui imitent l'évolution naturelle. Ils suivent les quatre étapes générales : la reproduction, la mutation, la recombinaison et la sélection et utilisent une fonction d'évaluation pour déterminer les conditions qui favorisent la survie. L'algorithme génétique (AG) est probablement l'algorithme évolutionnaire le plus couramment utilisé. Cette méthode a été proposée à l'origine par John Holland en 1975 [93]. Un AG sélectionne une population initiale de solutions possibles, dites  $P(t)$ , à chaque itération  $t$  pour un problème donné. En utilisant la transformation stochastique  $s$ , des solutions seront soumises à une étape de mutation ou de croisement. Ces nouvelles solutions sont considérées comme une nouvelle population, dites  $C(t)$ . à partir de  $P(t)$  et  $C(t)$ , les solutions avec les meilleures valeurs objectives sont choisies pour former une nouvelle population,  $P(t + 1)$ . Après l'évaluation de plusieurs générations, l'algorithme converge vers une solution (proche) optimale de la fonction objective. En général, la structure de l'algorithme AG est la suivante [104]:

<b>Tableau 2.5: L'Algorithme génétique</b>
<pre>Begin   Initialize and evaluate <math>P(t)</math>;   while (not termination condition) do     begin       Recombine <math>P(t)</math> to yield <math>C(t)</math>;       Evaluate <math>C(t)</math>;       Select <math>P(t + 1)</math> from <math>P(t)</math> and <math>C(t)</math>;     end   end end</pre>

### 6.2.2 L'Intelligence en essaim

L'Intelligence en essaim est une méta-heuristique dans le domaine de l'intelligence artificielle qui est utilisé pour résoudre des problèmes d'optimisation. Elle est

basée sur le comportement collectif des insectes sociaux, des troupes d'oiseaux, ou des bancs de poissons. La première utilisation et la première définition du terme «intelligence en essaim» remontent à l'article de Beni et Wang [105] en 1989 intitulé «Swarm intelligence in cellular robotics systems». Cet article propose une alternative à l'expression de robotique cellulaire pour décrire un groupe de robots coopérant afin d'accomplir une tâche ou résoudre un problème. Dans un article plus récent [106], Beni revient sur cette terminologie et son extension à d'autres contextes scientifiques.

D'un point de vue biologique, Bonabeau, Dorigo et Theraulaz dans [107] décrivent de plus cette forme d'intelligence comme une intelligence collective émergeant d'un groupe d'agents simples. Ils proposent la définition suivante :

*« L'intelligence en essaim inclut toute tentative de conception d'un algorithme ou d'un dispositif visant à résoudre des problèmes de façon distribuée, inspirée du comportement collectif des insectes sociaux ou d'autres sociétés animales »*

En général, l'intelligence en essaim concerne la modélisation des comportements collectifs de simples agents qui interagissent localement entre eux dans leur environnement, ce qui conduit à l'émergence d'un modèle global fonctionnel cohérent [108]. Ces modèles sont inspirés par le comportement social des insectes et autres animaux [109]. Du point de vue informatique, les algorithmes basés sur l'intelligence en essaim sont des algorithmes de calcul qui sont utiles pour résoudre des problèmes d'optimisation

Le principe fondamental de l'intelligence en essaim repose sur des algorithmes de recherche probabilistes. Tous les modèles de l'intelligence en essaim présentent un certain nombre de propriétés générales [110]. Chaque entité de l'essaim est constituée d'un simple agent. La communication entre les agents est généralement indirecte et courte. La coopération entre les agents est réalisée d'une manière distribuée sans aucun mécanisme de contrôle centralisé. Ces propriétés rendent les modèles de l'intelligence en essaim faciles à réaliser. Cependant, le comportement collectif au niveau de la colonie de l'essaim qui émerge des interactions est utile dans la réalisation des objectifs complexes [111].

### 6.2.2.1 Caractéristiques générales de l'intelligence en essaim

Il existe de nombreux types d'essaims dans le monde. Le niveau d'intelligence pourrait être varié d'un essaim à un autre. L'auto-organisation est un élément essentiel dans un essaim qui entraîne un comportement collectif au moyen d'interactions locales entre les agents. Bonabeau et al [112] interprètent l'auto-organisation en essaims à travers quatre caractéristiques:

- 1) La rétroaction positive: la promotion de la création de structures avantageuses.
- 2) La rétroaction négative: équilibrer la rétroaction positive et aider à stabiliser la structure collective.
- 3) Les fluctuations: marches aléatoires, erreurs, commutation aléatoire de tâche entre les individus d'essaims qui sont essentielles pour la créativité.
- 4) Les Interactions multiples: les agents de l'essaim utilisent les informations provenant des autres agents afin que l'information se propage à travers le réseau.

En plus de ces caractéristiques, l'exécution de tâches simultanément par des agents spécialisés, appelés la répartition des tâches, est également un élément important dans un essaim ainsi que l'auto-organisation pour l'apparition de l'intelligence [113].

Selon Millonas [114], pour appeler un essaim intelligent, l'essaim doit respecter les principes suivants :

- L'essaim devrait être en mesure de faire un simple espace et un calcul de temps (le principe de proximité).
- L'essaim devrait être en mesure de répondre aux critères de qualité de l'environnement (le principe de la qualité).
- L'essaim devrait être en mesure d'avoir des réponses variées.
- L'essaim ne devrait pas changer son mode de comportement à chaque fluctuation de l'environnement (principe de stabilité).

- L'essaim doit être en mesure de changer le mode de comportement lorsque cela est nécessaire (le principe de la capacité d'adaptation).

Plusieurs algorithmes d'optimisation inspirés du comportement de l'intelligence en essaim dans la nature sont proposés. Dans la partie suivante nous décrivons en détail deux algorithmes d'optimisation globale : l'optimisation par essaim particulaire et colonie d'abeilles artificielles.

### 6.2.3 La Colonie d'Abeilles Artificielles (ABC)

Les Algorithmes d'abeilles sont des nouveaux types d'algorithmes qui sont apparus dans le domaine de l'intelligence en essaim au cours des dernières années. Ces algorithmes tentent d'utiliser les principes du comportement collectif des abeilles. Ils ont déjà été appliqués à divers domaines tels que la robotique, les systèmes multi-agents, l'optimisation, etc. Les abeilles sont des insectes sociaux qui vivent en colonies composées d'une seule reine et des milliers de travailleurs. Le comportement collectif de la colonie dans son ensemble permet de résoudre plusieurs tâches complexes telles que le maintien d'une température constante dans la ruche, le suivi de l'évolution des conditions de recherche de nourriture et la sélection du meilleur site de nourritures. Il est remarquable que toutes ces tâches ne dépendent pas d'un contrôle central, mais sont réalisées en utilisant des principes d'auto-organisation.

En général, on peut distinguer deux grandes classes d'algorithmes d'optimisation d'abeilles: les algorithmes qui utilisent des mécanismes génétiques et le comportement d'accouplement d'abeilles, et les algorithmes qui prennent leur inspiration du comportement de butinage d'abeilles. La première classe d'algorithmes d'optimisation utilise le fait qu'une colonie d'abeilles comprend un grand nombre d'individus qui sont génétiquement hétérogène en raison de l'accouplement de la reine avec plusieurs mâles. Les algorithmes d'accouplement existants se basent sur le principe des algorithmes évolutionnaires en introduisant les opérateurs de mutation ou croisement.

## Chapitre 2 : Optimisation des classifieurs

La deuxième classe d'algorithmes se base sur le processus de recherche de nourriture. Ce type d'algorithmes utilise généralement des abeilles artificielles pour la recherche de solutions et donc les solutions sont associées à des sources de nourriture. Selon le nombre de sources de nourriture (solutions) trouvées et sur leur qualité, un sous-ensemble d'abeilles artificielles explore l'environnement (espace de recherche) pour trouver de nouvelles sources de nourriture (nouvelles solutions), tandis que les abeilles restantes exploitent l'environnement autour de sources de nourriture trouvées afin d'essayer de trouver de meilleures solutions (par exemple, ils effectuent des opérations de recherches locales afin d'améliorer une solution trouvée).

Dans le cas des algorithmes basés sur le comportement de butinage, les quatre caractéristiques sur lesquelles l'auto-organisation [107] repose peuvent être exprimées comme suit:

- 1) La rétroaction positive: Comme la quantité de nectar d'une source de nourriture augmente, le nombre de spectateurs visitant cette source augmente proportionnellement.
- 2) La rétroaction négative: Le processus de l'exploitation des mauvaises sources de nourriture est arrêté par les abeilles.
- 3) Fluctuations: Les scouts effectuent un processus de recherche aléatoire pour découvrir de nouvelles sources de nourriture.
- 4) les interactions multiples: Abeilles Employées partageant leurs informations sur les sources de nourriture avec leurs camarades (abeilles spectatrices) à travers une piste de danse.

Lorsque le comportement de butinage des abeilles expliqué ci-dessus est réexaminé, on voit que les principes définis par Millonas [113] sont entièrement satisfaits. Le premier algorithme basé sur les abeilles a été proposé par Sato et Hagiwara en 1997 [114].

Baykasoglu et al. [115], présentent une synthèse des différents travaux de recherche sur le développement des algorithmes d'abeilles dans la littérature. Ces techniques ont été classées en 3 groupes différents : le comportement de butinage,

le comportement du mariage et la reine des abeilles. Bien qu'il existe plusieurs modèles basés sur le comportement de butinage qui est lié au processus de nourriture des abeilles, notre travail de thèse est basé sur la colonie d'abeilles artificielles qui a été proposé par Karaboga [116].

### 6.2.3.1 Comportement de butinage

Le processus de recherche de nectar dans les fleurs par les abeilles a été considéré comme un processus d'optimisation. La manière de recherche sur des grandes quantités de sources de nourriture par les insectes sociaux a été modélisée comme une heuristique d'optimisation. Le modèle minimal biologique [116] de sélection de nourriture qui fait apparaître l'intelligence collective des essaims d'abeilles se compose de trois éléments essentiels comme nous pouvons le voir dans la figure 2.2 :

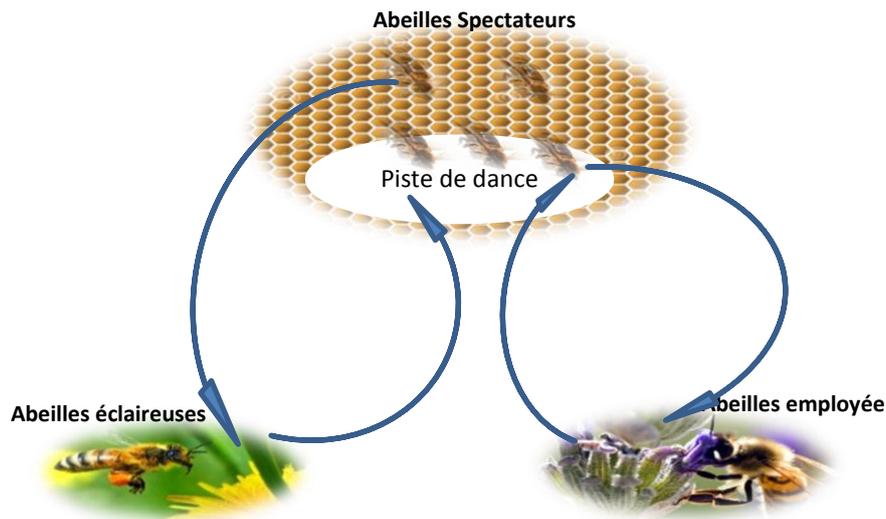


Figure 2 2: les éléments de base pour le comportement de butinage

- **Sources alimentaires:**

La valeur d'une source de nourriture dépend de nombreux facteurs tels que :

- la proximité d'une source de nourriture à la ruche
- la richesse de la source de nourriture
- la concentration de l'énergie de la source de nourriture
- la facilité d'extraire l'énergie à partir d'une source de nourriture.

Pour une raison de simplicité, la «rentabilité» d'une source de nourriture peut être représentée par une seule quantité.

- **Les abeilles Employées:**

Ces abeilles sont associées à une source de nourriture particulière qui est exploitée par elles. Les abeilles Employées partagent des informations sur la source de nourriture avec d'autres abeilles telles que l'emplacement et la rentabilité.

- **Les abeilles non Employées:**

Elles sont destinées à la recherche d'une source de nourriture pour l'exploiter. Il existe deux types d'abeilles non employées : l'éclaireuse qui cherche dans le voisinage de la ruche sur des nouvelles sources alimentaires et le spectateur qui attend dans la ruche afin de choisir une bonne source de nourriture basée sur l'information partagée par une abeille Employée.

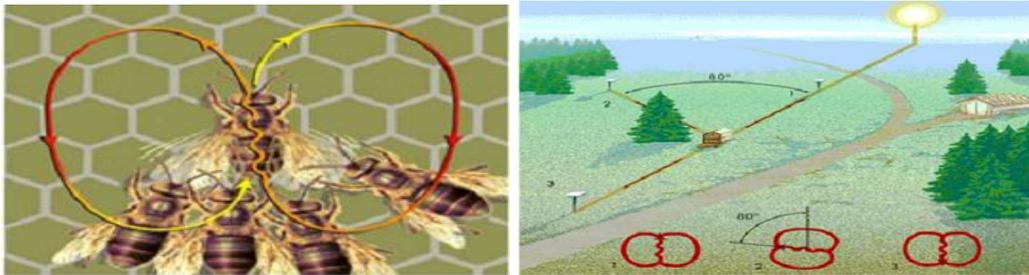
L'échange d'informations entre les abeilles est la partie la plus importante dans la formation de connaissance collective. Lorsqu'on examine la ruche, on trouve qu'il existe plusieurs zones. La zone la plus importante concernant cet échange est dite région de la danse. Cela signifie que L'abeille est capable de communiquer aux autres abeilles l'endroit où se trouve la nourriture par une danse [117]. Elle **danse en rond** (Figure 2.3) quand elle a trouvé du pollen à faible distance (moins de 25 mètres).



*Figure 2.3: La danse en rond qu'effectue l'abeille en fonction De la direction de la source de nourriture.*

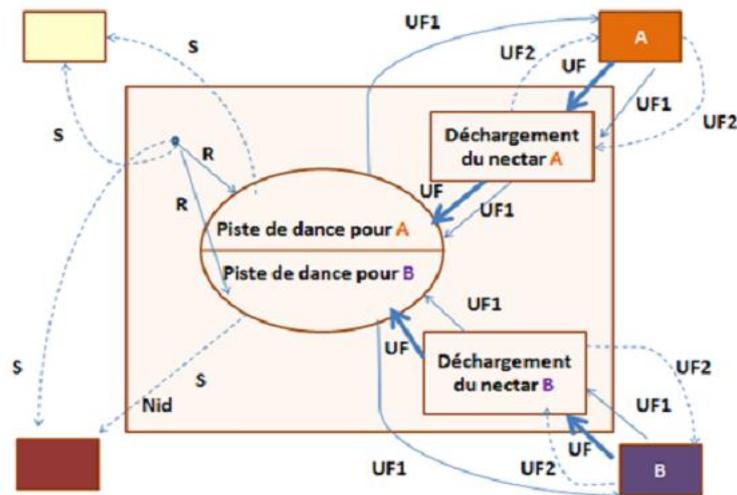
## Chapitre 2 : Optimisation des classifieurs

Elle utilise une danse très compliquée dite la **danse frétilante** [117] (Figure 2.4), ou danse en huit, si la nourriture se trouve à moins de 10 kilomètres. La direction de la nourriture est exprimée par rapport à la position du soleil (figure 2.3). La distance est exprimée par le nombre et la vitesse des tours effectués par l'abeille sur elle-même.



*Figure 2.4: la danse frétilante*

Afin de comprendre les caractéristiques de comportement de base des abeilles, nous allons examiner la Figure 2.5 suivante :



*Figure 2.5: comportement des abeilles*

Supposons qu'il existe deux sources de nourriture: A et B. Au début, une abeille n'a aucune information sur les sources de nourriture autour de la ruche elle est considérée comme une abeille non employée. Il y a deux possibilités pour une telle abeille:

## Chapitre 2 : Optimisation des classifieurs

- Elle peut être une éclaireuse qui cherche spontanément autour de la ruche sur une nourriture en raison d'une motivation interne ou externe (S sur la figure 2.5).
- Elle peut être un spectateur qui observe les danses frétilantes pour localiser une source de nourriture (R sur la figure 2.5).

Après avoir localisé la source de nourriture, l'abeille devient une «abeille employée» et elle utilise sa propre capacité pour mémoriser l'emplacement et puis commence à l'exploiter immédiatement. L'abeille prend une charge de nectar de la source et retourne à la ruche afin de le décharger dans une partie bien déterminée. Après le déchargement de la nourriture, l'abeille a trois possibilités:

- Elle abandonne la source de nourriture (UF)
- Elle danse dans la ruche et recrute d'autres abeilles avant de revenir à la même source de nourriture (EF1)
- Elle continue à se nourrir à partir de la source de nourriture sans recruter d'autres abeilles (EF2).

### 6.2.3.2 Principe fondamental de l'ABC

Comme dans le modèle minimal biologique, la colonie d'abeilles artificielles (ABC) [116] contient trois groupes d'abeilles: les abeilles Employées associées à des sources alimentaires spécifiques, les abeilles spectateurs observant la danse des abeilles Employées dans la ruche pour choisir une source de nourriture, et les éclaireuses (scouts) qui cherchent de nouvelles sources de nourriture de façon aléatoire. Les spectateurs et les éclaireuses sont aussi appelées abeilles non Employées. Initialement, toutes les positions des sources de nourriture sont découvertes par les éclaireuses. Par la suite, le nectar des sources de nourriture est exploité par les abeilles Employées et les abeilles spectateurs. Si la source de nourriture ne contient plus de nectar, l'abeille employée devient une abeille éclaireuse.

Dans l'algorithme ABC, la position d'une source de nourriture représente une solution possible au problème et la quantité de nectar d'une source de nourriture

correspond à la qualité (fitness) de la solution associée. Dans la forme de base, le nombre d'abeilles salariées est égal au nombre de sources de nourriture (solutions) puisque chaque abeille Employée est associée à une et une seule source de nourriture. La structure générale de la démarche d'optimisation par l'algorithme ABC est donnée comme suit:

<b>Tableau 2.6: L'Algorithme de La Colonie d'Abeilles Artificielles (ABC)</b>
<p>1: Parameter initialization:  <i>Number of Solutions: SN, number of Cycle: C The parameters (limit).</i></p> <p>2: Initialize the population of solutions <math>x_i</math>; <math>i = 1; \dots; SN</math> randomly.            3: Evaluate the fitness function <math>fit(x_i)</math> of the population.            4: <math>C = 1</math></p> <p>Repeat</p> <p style="text-align: center;"><b>{--Employed Bees' Phase--}</b></p> <p>5: Produce new solutions <math>v_i</math> for the employed bees and evaluate them (<math>fit(v_i)</math>).            6: Apply the greedy selection process for the employed bees            7: Calculate the probability values <math>P_i</math> for the solutions <math>x_i</math>            8: Produce the new solutions <math>v_i</math> for the onlookers from the solutions <math>x_i</math> selected depending on <math>P_i</math> and evaluates them (<math>fit(v_i)</math>)</p> <p style="text-align: center;"><b>{--Onlooker Bees' Phase--}</b></p> <p>9: Apply the greedy selection process for the onlookers</p> <p style="text-align: center;"><b>{--Scout Bee Phase--}</b></p> <p>10: Determine the abandoned solution for the scout, if exists, and replace it with a new randomly produced solution <math>x_i</math>            11: Memorize the best solution achieved so far            12: <math>C = C + 1</math>            Until (<math>C = Maximum\ Cycle\ Number</math>)</p>

Comme tous les autres algorithmes basés sur la population, ABC a un processus itératif. Les étapes de l'ABC de base peuvent être expliquées comme suit:

- **Initialisation de la population**

L'algorithme génère aléatoirement une population initiale de  $SN$  solutions. Soit  $x_i$  ( $i = 1, 2, \dots, SN$ ) représente la  $i^{\text{ème}}$  source de nourriture dans la population. Chaque source de nourriture est générée comme suit:

$$x_{i,j} = x_{min,j} + rand(0,1)(x_{max,j} - x_{min,j}),$$

Où  $i = 1, 2, \dots, SN, j = 1, 2, \dots, n$ .  $x_{min,j}$ , et  $x_{max,j}$  sont les limites inférieures et supérieures de la dimension  $j$ . Ces sources de nourritures sont réparties au hasard sur le nombre d'abeilles employées et leur fitness sont évalués.

- **Etape des abeilles employées :**

Dans cette étape, chaque abeille employée  $X_i$  génère une nouvelle source de nourriture  $V_i$  dans le voisinage de sa position actuelle à l'aide de l'équation suivante:

$$V_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj})$$

où  $k \in \{1, 2, \dots, SN\}$  and  $j \in \{1, 2, \dots, n\}$  sont des indexes choisis aléatoirement;  $k$  doit être différent de  $i$ ;  $\phi_{ij}$  est un nombre aléatoire dans l'intervalle  $[-1, 1]$ . Une fois que  $V_i$  est obtenue, elle sera évaluée et comparée à  $X_i$ . Si la valeur de fitness de  $V_i$  est égale ou supérieure à celle de  $X_i$ ,  $V_i$  remplacera pour devenir un nouveau membre de la population, sinon  $X_i$  est conservée. En d'autres termes, un mécanisme de sélection est utilisé entre l'ancienne solution et la solution candidate.

- **Calcul des valeurs de probabilité impliquées dans la sélection probabiliste :**

Après que toutes les abeilles employées complètent leurs recherches, ils partagent leurs informations relatives aux quantités de nectar et les positions de leurs sources avec les abeilles spectateurs sur la piste de danse. L'abeille spectateur

évalue les informations de nectar tiré de toutes les abeilles employées et choisit une source de nourriture avec une probabilité liée à sa quantité de nectar. Cette sélection probabiliste dépend des valeurs de la fonction d'évaluation des solutions dans la population

Dans l'algorithme de base, le schéma de sélection de la roue de roulette, dans lequel chaque tranche est proportionnelle à la taille de la valeur de fitness est utilisée comme suit:

$$p_i = \frac{f_i}{\sum_{j=1}^{SN} f_j}$$

Où  $f_i$  est la valeur de fitness de la solution  $i$ . Évidemment, plus le  $f_i$  est grand, plus la probabilité que la  $i$ ème source de nourriture est sélectionnée.

- **Etape d'abeille spectateur:**

L'abeille spectateur évalue les informations de nectar tiré de toutes les abeilles employées et sélectionne une source de nourriture  $X_i$  en fonction de sa valeur de probabilité  $p_i$ . Une fois que le spectateur a choisi sa source de nourriture  $X_i$ , elle produit une modification de  $X_i$  en utilisant l'Equation. Comme dans le cas des abeilles employées, si la source de nourriture modifiée a une quantité de nectar meilleure ou égale à  $X_i$ , la source de nourriture modifiée remplacera  $X_i$  pour devenir un nouveau membre dans la population.

- **Etape d'abeille éclairieuse :**

Si une source alimentaire  $X_i$  ne peut être encore améliorée par un nombre prédéterminé et limité d'essais, la source de nourriture est censée être abandonnée, et l'abeille employée correspondante devient une abeille éclairieuse. L'abeille éclairieuse produit une source de nourriture au hasard comme suit:

$$x_{i,j} = x_{min,j} + rand(0,1)(x_{max,j} - x_{min,j})$$

Où  $j = 1, 2, \dots, n$ .

### 6.2.4 L'Optimisation par Essaim Particulaire (OEP)

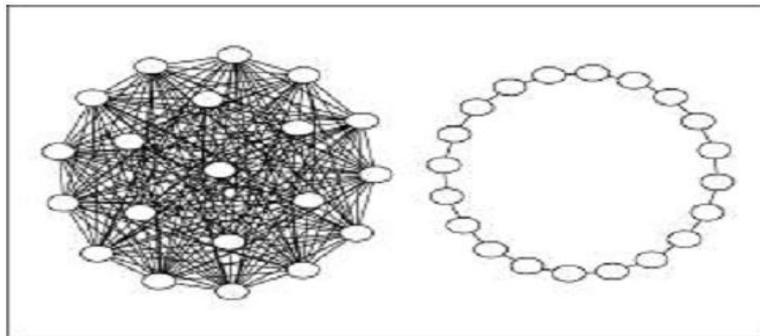
L'optimisation par essaim particulaire (OEP) est une méta-heuristique bio-inspirée qui a été proposée par James Kennedy et Russell Eberhart en 1995 [94]. OEP effectue une recherche basée sur la population, en utilisant des particules pour représenter les solutions possibles dans un espace de recherche.

L'origine de l'OEP est décrite comme inspiration sociologique, car il a été initialement conçu comme un outil [118, 109] pour la simulation des trajectoires de vol des oiseaux, qui a été principalement régit par deux préoccupations majeures: l'évitement des collisions, l'adaptation de la vitesse. D'autre part, les raisons présentées pour le comportement de flocage observé dans la nature sont: la protection contre les prédateurs et la recherche efficace en ce qui concerne la nourriture. La dernière raison assume une grande importance, quand la nourriture est inégalement répartie sur une vaste région.

Dans l'algorithme OEP nombreuses entités autonomes (particules) sont stochastique-ment générées dans l'espace de recherche. Chaque particule est une solution candidate au problème et elle est représentée par une vitesse, un emplacement dans l'espace de recherche et a une mémoire où elle aide à se rappeler sa précédente meilleure position. Un essaim se compose de  $N$  particules qui volent autour d'un espace de recherche de  $D - dimensions$ . En outre, chaque essaim de particules a une sorte de topologie décrivant les interconnexions entre les particules. L'ensemble de particules auquel une particule  $i$  est typologiquement connectée est appelé voisinage de  $i$ . Le voisinage peut être l'ensemble de la population ou un sous-ensemble.

L'algorithme OEP est assez simple, car il se compose uniquement de deux règles pour l'obtention d'une nouvelle solution à partir d'une précédente. Il a montré une convergence rapide à l'optimum (ou ses environs) dans une grande variété de problèmes d'optimisation, ce qui a considérablement augmenté sa popularité au cours des dernières années [119].

Diverses topologies ont été utilisées pour interconnecter les particules de l'essaim. Les deux les plus couramment utilisées sont connues comme *gbest* (pour meilleur global) et *lbest* (pour meilleur locale). La topologie *gbest* peut être conceptualisée comme un graphique entièrement connecté où le meilleur voisin dans l'ensemble de la population influence la particule cible. La topologie *lbest*, introduite dans [120], est un réseau en anneau simple où chaque individu est relié à  $K = 2$  éléments adjacents dans la population (naturellement, cela peut être généralisé à  $K > 2$ ). Kennedy et al. [109] a souligné que la topologie de *gbest* avait une tendance à converger très rapidement avec une plus grande chance de se retrouver dans des optima locaux. D'autre part, la topologie *lbest* a été plus lente mais plus complètement explorée, et typiquement finie avec un meilleur optimum. Les effets de différentes topologies de la population sur l'algorithme OEP ont été étudiés dans [121].



**Figure 2.6: Topologies GBEST (à gauche) et LBEST (à droite)**

### 6.2.4.1 Les éléments de l'OEP

L'algorithme de l'OEP travaille sur une population de particules qui représente un ensemble de solutions possibles. Ces particules sont placées aléatoirement dans un espace de recherche. Pour appliquer l'OEP il faut définir une fonction objective à optimiser. Le principe de l'algorithme est de bouger ces particules pour trouver l'optimum. Chacune de ces particules est dotée :

- D'une position, c'est-à-dire ses coordonnées dans l'espace de recherche.
- D'une vitesse qui permet à la particule de se déplacer. Au cours des itérations, chaque particule change sa position. Elle évolue en fonction de

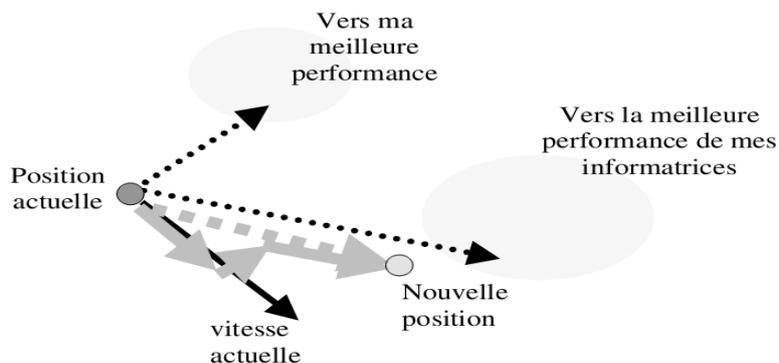
son meilleur voisin, de sa meilleure position, et de sa position précédente. cette évolution permet de trouver la meilleure particule.

- D'un voisinage, c'est-à-dire un ensemble de particules qui interagissent directement sur la particule.

A tout instant, chaque particule est caractérisée par :

- Sa meilleure position visitée. On retient essentiellement la meilleure valeur de la fonction objective ainsi que ses coordonnées.
- La position du meilleur voisin de l'essaim qui correspond à l'ordonnancement optimal.
- La valeur de la fonction objective dans la position actuelle car à chaque itération il faut une comparaison entre la valeur de la position actuelle par rapport à la valeur de la meilleure position.

Maurice Clerc et al. [122], déterminent que l'évolution d'une particule est une combinaison de trois types de tendances : *volontariste* (suivre sa voie selon sa vitesse actuelle), *conservatrice* (revenir en arrière en prenant en compte sa meilleure performance) et *suiviste* (suivre aveuglement la meilleure particule).



**Figure 2.7: principe du déplacement d'une particule.**

### 6.2.4.2 Principe fondamental de l'OEP

Les étapes de l'algorithme OEP pour trouver la meilleure position peuvent être résumées comme suit:

**Tableau 2.7: L'Algorithme de L'Optimisation par Essaim Particulaire (OEP)**

1: Initialization:

- Number of particles  $L$ , number of iterations  $G$ , and the constants for the OEP ( $\gamma$ ,  $C1$  and  $C2$ ),
- generate randomly initial velocity for all particles of the population

2: Calculate the fitness value of each particle:

$$fit_i = fit(P_i)$$

- Update  $P_{besti}$ ,  $F_{besti}$ ,  $P_{Gbest}$  and  $F_{Gbest}$

If  $fit_i > F_{besti}$  then  $F_{besti} = fit_i$  and  $P_{besti} = P_i$

If  $fit_i > F_{Gbest}$  then  $F_{Gbest} = fit_i$  and  $P_{Gbest} = P_i$   
Else

- Set  $G=1$

3:

- Update particles velocity by:  $V_i = \gamma V_i + C_1 r_1 (P_{besti} - P_i) + C_2 r_2 (P_{Gbest} - P_i)$
- Update particles position by:  $P_i = P_i + V_i$

4:  $G = G + 1$ .

- If  $G > \text{number of iterations}$  then go to Step 5 else go to Step 2.

5: -The optimal solution is desired by the best particle  $P_{Gbest}$  with the best fitness  $F_{Gbest}$ .

A chaque itération, les particules se déplacent en prenant en compte leur meilleure position (déplacement volontariste) et la meilleure position de son voisinage (déplacement suiviste). On calcule la nouvelle vitesse à partir de la formule suivante :

$$V_i^{k+1} = \underbrace{\gamma V_i^k}_1 + \underbrace{C_1 r_1 (P_{besti}^k - x_i^k)}_2 + \underbrace{C_2 r_2 (P_{Gbesti}^k - x_i^k)}_3$$

Ou

- $V_i^k$  est la vitesse de la particule aux itérations k et k+1.
- $P_{besti}$  est la meilleure position de la particule
- $P_{Gbesti}$  est la meilleure position de son voisinage à l'itération k

- $X_i^k$  est la position de la particule à l'itération  $k$
- $C_1$  et  $C_2$  sont des coefficients fixés et générés aléatoirement à chaque itération.
- $\gamma$  est le facteur d'inertie.

On peut ensuite déterminer la position suivante de la particule par :

$$P_i^{k+1} = P_i^k + V_i^{k+1}$$

Où :  $P_i^k$  est la position de la particule à l'itération  $k$

- **Initialisation de la population :**

La position des particules ainsi que leur vitesse initiale doivent être initialisées aléatoirement. Cependant, en ce qui concerne la position des particules, il est préférable d'utiliser un générateur pertinent dans la disposition homogène des particules dans un espace de dimension  $n$ .

- **Vitesse maximale :**

Afin d'éviter que les particules ne se déplacent pas trop rapidement hors l'espace de recherche, passant éventuellement à côté de l'optimum, Eberhart et al. [123] définit un système de blocage pour limiter la vitesse de chaque particule, de telle sorte que chaque composante de  $V_i$  est maintenue dans l'intervalle  $[-V_{max}, +V_{max}]$

- **Coefficients de confiance :**

Les variables de confiance  $c_1$  et  $c_2$  sont des termes multiplicateurs constants appelés coefficients d'accélération. Ils pondèrent les tendances de la particule à vouloir suivre son instinct de conservation ou son panurgisme. Dans presque toutes les applications, les coefficients de confiances sont choisis au hasard dans un intervalle donné. Les variables  $r_1$  et  $r_2$  peuvent être définies de façon aléatoire dans l'intervalle  $[0..1]$ .

- **Facteur d'inertie**

Le facteur d'inertie  $\gamma$ , introduit dans [124], joue un rôle d'équilibrage entre la recherche globale et la recherche locale. Une grande valeur d'inertie encourage

l'exploration globale (c'est à dire, de diversifier la recherche dans l'ensemble de l'espace de recherche), tandis qu'une valeur d'inertie plus petite encourage l'exploitation locale (ie, intensifie la recherche dans la région actuelle) [125].

- **Critères d'arrêt**

Comme indiqué précédemment, la convergence vers la solution optimale globale n'est pas garantie dans tous les cas même si les expériences dénotent la grande performance de la méthode. Donc l'algorithme doit alors s'exécuter tant que l'un des critères de convergence suivant n'a pas été atteint :

- nombre maximum d'itérations a été atteint ;
- la variation de la vitesse est proche de 0 ;
- le fitness de la solution est suffisant.

- **La Fonction d'évaluation :**

L'évaluation des particules s'effectue à l'aide d'une fonction, la fitness ( $fit_i$ ) ou valeur sélective, qui permet d'évaluer et d'associer une valeur à chaque particule. Ces valeurs aideront le processus de sélection des meilleures solutions.

## 7 Conclusion

Dans ce chapitre, nous avons présenté le concept d'optimisation. Tout d'abord, nous avons donné la définition d'un problème d'optimisation et les étapes d'un processus d'optimisation. Puis nous avons présenté une famille de méthodes de résolution de problèmes d'optimisation: les méta-heuristiques et plus particulièrement les méthodes basées sur la population. Enfin, les algorithmes (optimisation par essaim particulaire et colonie d'abeilles artificielles) sont décrits en détail.

# Chapitre 3

## Résultats et discussion

### Contenu

---

1 Introduction .....	72
2 La maladie du cancer du sein .....	72
2.1 Définition.....	72
2.2 La base de données du cancer du sein .....	73
2.3 Travaux de littérature sur la maladie du cancer du sein .....	75
3 La maladie du diabète .....	76
3.1 Définition.....	76
3.2 Base de données de diabètes .....	76
3.3 Traitement des valeurs manquantes.....	77
3.4 Travaux de littérature sur la maladie du diabète:.....	77
4 L'approche adaptée pour l'optimisation du modèle flou.....	78
4.1 Le modèle des règles floues adapté .....	80
4.2 Codification de règles floues.....	81
4.3 L'initialisation de la Population.....	83
4.4 Fonction d'évaluation.....	83
5 Modèle 1: conception d'un classifieur flou par l'algorithme de colonie d'abeilles artificielles (MABC).....	84
5.1 Opérateur de croisement mixte : BLX- $\alpha$ .....	84
5.2 Principe de l'algorithme ABC modifié (MABC) .....	85
5.3 Présentation de l'algorithme ABC modifié (MABC).....	86
5.4 Résultats et discussion.....	87
6 Modèle 2: conception d'un classifieur flou par l'algorithme d'optimisation par essaim particulaire (MPSO) .....	95
6.1 Principe de l'algorithme OEP modifié (MPSO) .....	96
6.2 Présentation de l'algorithme d'OEP modifié .....	96
6.3 Résultats et discussion.....	97
7 Comparaison de l'approche proposée avec les travaux de la littérature.....	100
8 Conclusion.....	102

---

# 1 Introduction

Les algorithmes d'apprentissage automatique sont utilisés dans les systèmes d'aide au diagnostic médical avec un grand succès. Même si la décision de l'expert est l'élément le plus important lors du diagnostic médical, les techniques dites intelligentes fournissent une aide importante et substantielle, car elles réduisent les erreurs causées par la fatigue d'experts et le temps nécessaire pour le diagnostic médical. Chaque Diagnostic nécessite une procédure de détection précise et fiable pour détecter une pathologie donnée. Afin de garantir un diagnostic rapide, précis et significatif, l'utilisation d'un système de classification robuste est la solution la plus appropriée. Dans le cadre de cette thèse nous nous intéressons à la conception d'un classifieur basé sur l'extraction et l'optimisation de la connaissance médicale par une approche floue.

Deux nouvelles variantes des méta-heuristiques (Optimisation par essaim particulaire et par colonies d'abeilles artificielles) sont proposées dans notre étude pour construire un classifieur flou optimal. Nous avons proposé deux algorithmes différents modifiés et ajustés pour créer et optimiser automatiquement une base de règles floues. La construction d'un classifieur flou passe par deux étapes :

- i. La base initiale de règles floues est vide.
- ii. Les fonctions d'appartenance et la base de règles sont créées et optimisées par des méta-heuristiques.

Afin de valider nos approches et évaluer les performances des méthodes proposées, nous avons utilisé deux bases de données médicales (diabète et cancer du sein) [127].

## 2 La maladie du cancer du sein

### 2.1 Définition

Le cancer est un type de maladies dans lesquelles les cellules du corps se développent, changent, et se multiplient d'une manière non contrôlée.

Habituellement, le nom de chaque cancer indique l'organisme qui présente son origine. Ainsi, le cancer du sein se réfère à la croissance de cellules irrégulières dans le tissu mammaire. Un groupe de cellules qui divise rapidement peut former une masse de tissu supplémentaire. Ces masses sont appelées tumeurs. Les tumeurs du sein peuvent être bénignes (non cancéreuses) ou malignes (cancéreuses). Le terme "cancer du sein" désigne une tumeur maligne développée à partir des cellules dans le sein. Habituellement le cancer du sein soit il commence dans les cellules des lobules, qui sont les glandes productrices de lait, ou les conduits, les passages qui drainent le lait des lobules au mamelon. Plus rarement, le cancer du sein peut commencer dans les tissus stromas, qui comprennent les acides gras et les tissus conjonctifs fibreux du sein. Au fil du temps, les cellules cancéreuses peuvent envahir les tissus sains environnants du sein et faire leur chemin dans les ganglions lymphatiques des aisselles, de petits organes qui filtrent les substances étrangères dans le corps. Si les cellules cancéreuses pénètrent dans les ganglions lymphatiques, ils ont alors une voie d'accès à d'autres parties du corps. L'étape du cancer du sein se réfère à quel point les cellules cancéreuses se sont propagées au-delà de la tumeur d'origine [126]. Le cancer du sein est devenu la principale cause de décès chez les femmes dans les pays développés. Le moyen le plus efficace pour réduire la mortalité du cancer du sein est le détecter le plus tôt possible. Le diagnostic précoce nécessite une procédure de détection précise et fiable qui permet aux médecins de distinguer les tumeurs bénignes du sein de celles malignes. Il est donc nécessaire de développer une méthode d'identification pour reconnaître le cancer du sein. L'objectif de ces méthodes de détection (de prévision) consiste à attribuer les patients soit à un groupe «bénigne» qui est non-cancéreux ou à un groupe «maligne» qui est cancéreux.

### 2.2 La base de données du cancer du sein

La base de données du cancer du sein que nous avons utilisé est fournie par l'UCI (Université de Californie à Irvine) référentiel de l'apprentissage automatique [127]. Depuis 1988, l'Université du Wisconsin alimente une base avec des données médicales (WBCD pour Wisconsin Breast Cancer Data) concernant des

### Chapitre 3 : Résultats et discussion

prélèvements cytologiques (Épaisseur, uniformité de la taille et de la forme des cellules, etc.) avec leurs diagnostics : bénin ou malin. Ces échantillons arrivent sur plusieurs groupes :

- {1er groupe : 367 cas (janvier 1989).
- {2ème groupe : 70 cas (octobre 1989).
- {3ème groupe : 31 cas (février 1990).
- {4ème groupe : 17 cas (avril 1990).
- {5ème groupe : 48 cas (avril 1990).
- {6eme groupe : 49 cas (janvier 1991).
- {7eme groupe : 31 cas (juin 1991).
- {8eme groupe : 86 cas (novembre 1991).

La base comporte environ 683 exemples qui sont caractérisés par neuf attributs décrits dans le tableau suivant :

Nombre d'attributs	9	
Les Noms d'attributs	Epaisseur de l'échantillon	
	Uniformité de la taille	
	Uniformité de la forme	
	Adhésion Marginale	
	Taille cellule épithéliale	
	Noyaux	
	Chromatine	
	Nucléole normal	
	Mitose	
Valeur Minimale	1.0	
Valeur Maximale	10.0	
Classes	Bénin, Malin	
Nombre d'échantillons	683	239 cas Malin
		444 cas Bénin

**Tableau 3.1: Attributs des données de la base de cancer de sein**

## 2.3 Travaux de littérature sur la maladie du cancer du sein

Plusieurs études ont été publiées pour tenter de résoudre le problème de diagnostic du cancer du sein. Dans le travail [128] de Pena-Reyes et al. ont développé la technique de classification qui utilise la méthode Fuzzy-GA ou ils ont obtenus une précision de 97,36%. Dans un autre travail de Goodman et al [129] trois méthodes différentes ont été proposées, l'apprentissage par quantification vectorielle optimisées (LVQ), Big LVQ, et le système de reconnaissance immunitaire artificiel. Les précisions obtenues étaient respectivement de 96,7%, 96,8% et 97,2%,.

Pola et al. [130] ont appliqué le système de reconnaissance immunitaire artificiel flou pour créer la base des règles floues. Leur approche a atteint 98,5% de précision. Gadaras et Mikhailov [131] ont proposé un nouveau cadre de classification floue utilisé pour l'extraction automatique des règles floues. Ils ont rapporté 96,08% de précision. Maglogiannis et al. [132] ont proposé trois méthodes différentes : SVM, les réseaux bayésiens et les réseaux de neurones artificiels, ils ont obtenus les taux de reconnaissances de 97,54%, 92,80% et 97,90% respectivement. Karabatak et Ince [133] ont utilisés les règles d'association et réseaux de neurones (AR-NN), ils ont obtenu une précision de 95,6% .

Chang et al. [134] ont proposé une approche hybride efficace d'exploration de données du cancer du sein basée sur Discrete Particle Swarm Optimization et la méthode statistique, ils ont obtenu un taux de classification 98,71%. Dans [135] les auteurs ont proposé une méthode d'extraction des caractéristiques importantes (sélection de variables) d'un ensemble de données et ils ont utilisé le modèle de Markov caché (HMM)-floue pour classifier le type de cancer du sein en utilisant la validation croisée, ils ont obtenu un taux de classification de 97.07% avec 9 caractéristiques et un taux de classification de 97.98% avec 3 caractéristiques.

## 3 La maladie du diabète

### 3.1 Définition

Le diabète est une maladie complexe et compliquée caractérisée soit par une absence d'insuline ou une résistance à l'insuline qui est une hormone essentielle pour le métabolisme du sucre dans le sang. Chez une personne en bonne santé, le pancréas produit l'insuline pour aider à métaboliser le sucre dans le sang et maintenir la glycémie (sucre) dans leur plage normale. Les diabétiques sont incapables de produire de l'insuline ou sont résistants à l'insuline, et par conséquent ne peuvent pas régler le glucose dans le sang. S'il existe une insuffisance d'insuline ou de la résistance à l'insuline, les taux de glucose dans le sang augmentent et provoquent des problèmes de santé grave.

La classification du diabète comprend deux grandes catégories cliniques: le diabète type 1 (anciennement appelé "diabète juvénile") est une maladie auto-immune dans laquelle les cellules bêta productrices d'insuline sont détruites par le système immunitaire de l'organisme. En conséquence, le corps est incapable de produire de l'insuline, le diabète Type 2 (précédemment appelé «diabète de l'adulte»); dans ce type l'insuline est produite en quantité insuffisante et / ou ne peut pas être utilisé par le corps pour contrôler les niveaux de sucre dans le sang [136].

### 3.2 Base de données de diabètes

Dans cette thèse nous utilisons la base de données médicale réelle (Indians Diabetes Pima). L'ensemble de données a été choisi du dépôt d'UCI [127] qui réalise une étude sur 786 femmes Indiennes Pima, Ces mêmes femmes, qui ont stoppé leurs migrations en Arizona, États-Unis, adoptant un mode de vie occidental, développent un diabète dans presque 50% des cas. Le diagnostic est une valeur binaire variable «classe» qui permet de savoir si le patient montre des signes de diabète selon les critères de l'Organisation Mondiale de la Santé. Les huit descripteurs cliniques sont présentés dans le tableau suivant:

Nombre d'attributs	8	
Les Noms d'attributs	Npreg : nombre de grossesses	
	Glu : concentration du glucose plasmatique	
	BP : tension artérielle diastolique	
	SKIN : épaisseur de pli de peau du triceps	
	Insuline : dose d'insuline	
	BMI : index de masse corporelle	
	PED : fonction de pedigree de diabète (l'hérédité).	
	Age : âge	
Classes	Diabétique, Non Diabétique	
Nombre d'échantillons	768	268 cas Diabétiques
		500 cas Non Diabétiques

*Tableau 3.2: Attributs des données de la base Pima*

### 3.3 Traitement des valeurs manquantes

La base de données Pima que nous avons utilisé dans notre thèse contient des valeurs manquantes. Pour remédier à ce problème, nous avons procédé à un prétraitement de la base de données en remplaçant les valeurs manquantes par la moyenne de toutes les valeurs du paramètre en question.

Cette astuce n'est pas l'unique technique utilisée dans l'état de l'art, il y a d'autres chercheurs qui préfèrent remplacer la valeur manquante par celle du patient qui le suit, ou le précède.

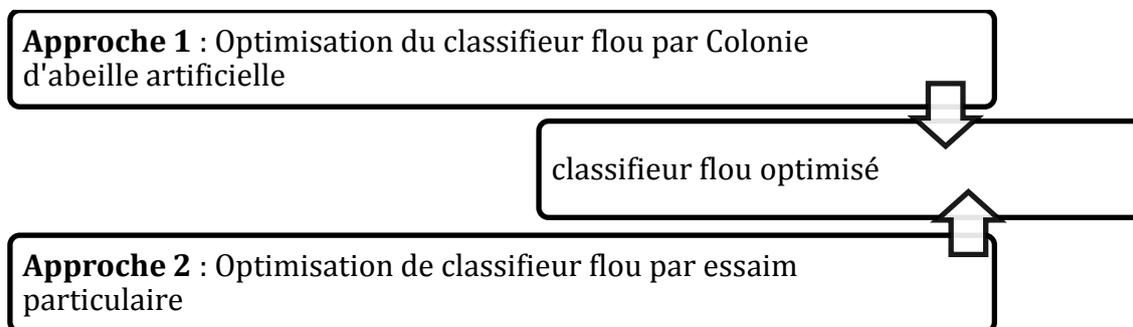
### 3.4 Travaux de littérature sur la maladie du diabète

Beaucoup de recherches ont été réalisées jusqu'aujourd'hui pour examiner et reconnaître le diabète dans la littérature, en général, les travaux sont basés sur des techniques d'intelligence artificielle. Dans [137], une analyse discriminante généralisée utilise les moindres carrées avec la technique SVM pour diagnostiquer le diabète, Ils ont rapporté 79,16% de précision. Dans [138], Polat et al. ont utilisé un système immunitaire artificiel pondéré avec une validation croisée de 10 sous-ensembles; ils ont obtenu une précision de 75,87%. Ping-hung et Ming-Hseng

[139] ont développé des méthodes basées sur des algorithmes génétiques (AGs) pour estimer un vecteur de poids des différentes caractéristiques appliquées dans l'estimation k-NN flou, la précision était de 81,6% en utilisant des algorithmes génétiques et le codage binaire (BGA) et 82% en utilisant des algorithmes génétiques et le codage réel (RGA). Polat et Günes [141] ont obtenu 89.47% de précision en utilisant une analyse en composantes principales (ACP) et une inférence neuro-flou adaptatif dans le diagnostic de la maladie de diabète. Dans les travaux de Temurtas et al. [142] un réseau de neurones multicouches et un réseau de neurones probabilistes (PNN) ont été utilisés pour le diagnostic du diabète. Ils ont rapporté respectivement 79,62% et 78,05% en terme de taux de classification.

### 4 L'approche adaptée pour l'optimisation du modèle flou

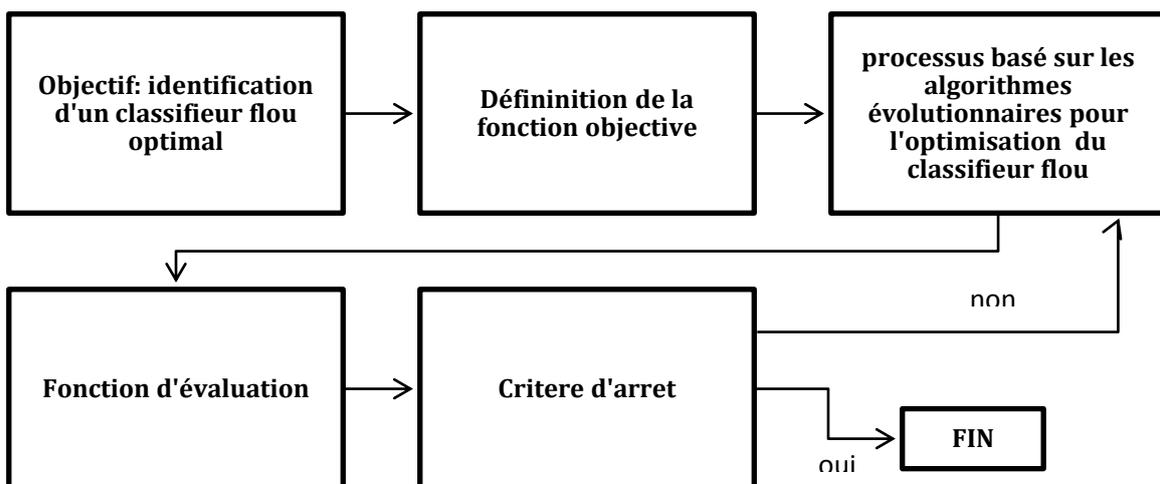
L'objectif principal de cette thèse est de développer une approche efficace qui peut améliorer les performances des classifieurs des données médicales, comme pour les cas du diabète et du cancer du sein. La contribution majeure dans notre approche est l'utilisation de l'algorithme de la colonie d'abeilles artificielles et l'algorithme d'optimisation par essaim particulière avec quelques modifications au niveau des algorithmes de base, afin de concevoir un classifieur flou optimal avec un petit ensemble de règles qui sont simples et compréhensibles par les médecins. Notre but principal est de garder une interprétabilité des résultats à notre classifieur. La conception des classifieurs flous par les deux méta-heuristiques est représentée selon deux approches distinctes (voir la figure 3.1).



*Figure 3.1: les deux approches d'optimisation du classifieur flou par les méta-heuristiques*

À travers les algorithmes évolutionnaires, la sélection des règles d'un classifieur flou peut être considérée comme un processus d'optimisation où une partie ou la totalité des paramètres d'un classifieur flou constituent l'espace de recherche. Chaque point dans l'espace de recherche correspond à un classifieur flou (i.e. représentation des fonctions d'appartenances, base de règles).

La performance du classifieur constitue une hyper-surface et la conception d'un classifieur flou optimal est équivalente à trouver l'emplacement optimal de cette hyper-surface. L'hyper-surface est généralement jugée infiniment grande, non différentiable, complexe. Dans ce cas l'utilisation des algorithmes classiques induit au problème du minimum local. Pour cela, il serait important d'utiliser une technique d'optimisation à base de population, ce qui rend les algorithmes évolutionnaires très appropriés pour la recherche sur l'hyper-surface que les méthodes à base de gradient traditionnels. Les algorithmes évolutionnaires comme OEP et ABC ont la capacité de trouver la solution (proche) optimale dans un espace de recherche complexe (minimum globale) et peuvent être utilisés pour modifier et fixer les paramètres d'un classifieur flou. La méthodologie pour l'identification des classifieurs flous en utilisant les algorithmes évolutionnaires comme un système d'optimisation est représentée dans la figure 3.2.



*Figure 3.2: Identification d'un classifieur flou par le processus d'optimisation des méta-heuristiques*

Il est nécessaire donc de définir au départ le modèle des règles floues utilisé et leurs codifications, l'espace de solution, la fonction d'objective et les contraintes.

### 4.1 Le modèle des règles floues adapté

Un classifieur flou comprend des règles linguistiques qui sont faciles à interpréter par l'utilisateur. Le classifieur n'est pas une boîte noire; il peut être interprété via une base de regles floues. Ceci est très important pour les systèmes d'aide au diagnostic médical, où les médecins n'acceptent pas le diagnostic d'une maladie, à moins qu'ils comprennent pourquoi et comment un diagnostic a été établi. Par conséquent, le classifieur flou peut être un outil pratique dans le processus de diagnostic. Le problème de classification concerne une population de  $m$  échantillons  $X_p (x_{p1}, \dots, x_{pn}) (p = 1, 2, \dots, m)$  et  $M$  classes où  $x_{pi}$  est le  $i$ -ème valeur d'attribut ( $i = 1, 2, \dots, n$ ) pour le  $p$  ième vecteur d'apprentissage. Dans le cadre de notre thèse, les règles floues peuvent être exprimées comme suit:

$$R_j : \text{if } x_{p1} \text{ is } A_{j1} \text{ and } x_{p2} \text{ is } A_{j2} \text{ and } \dots \text{ and } x_{pn} \text{ is } A_{jn} \text{ THEN } x_p (x_{p1}, \dots, x_{pn}) \\ \text{belongs to class } C \quad j = 1, 2, \dots, N$$

Ici,  $x_p (x_{p1}, \dots, x_{pn}) (p = 1, 2, \dots, m)$  est le vecteur d'entrée,  $R_j$  est l'étiquette de la  $j$  ème règle floue,  $A_{j1}, \dots, A_{jn}$  sont des ensembles flous définis dans la partie antécédent par des fonctions d'appartenance,  $\mu_{A_{ji}} (x_i)$  est le degré d'appartenance dans un ensemble flou qui est exprimé par un nombre compris entre 0 et 1.  $C_i \{C1, C2 \dots CM\}$  est la classe conséquente de sortie de la  $j$  ème règle floue. Nous avons utilisé la fonction d'appartenance de type trapèze sachant que ces types de fonction gardent la lisibilité des règles ; (i.e. la Figure 3.3). Cette fonction est définie par une limite inférieure  $a$ , une limite supérieure  $d$ , support inférieur limite  $b$  et un support supérieur limite  $c$ , où  $a < b < c < d$ :

$$\mu_A(x) = \begin{cases} 0, & (x < a) \text{ or } (x > d) \\ \frac{x - a}{b - a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d - x}{d - c}, & c \leq x \leq d \end{cases}$$

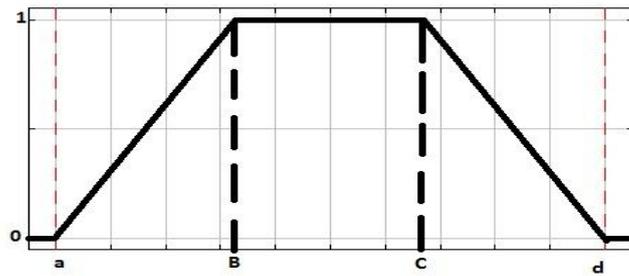


Figure 3.3: Fonction d'appartenance de type trapèze

La règle gagnante est utilisée pour prendre une décision [143]. Étant donné une base de règles constituée de  $N$  règles, un modèle d'entrée est classé selon la classe conséquente de la règle gagnante  $R_w$ . la règle gagnante est la règle qui a le degré de vérité le plus élevé

## 4.2 Codification de règles floues

Dans ce travail, nous avons utilisé deux ensembles flous (Low (L) et High (H)) pour chaque attribut avec des fonctions d'appartenance trapézoïdales. Ces ensembles flous sont définis par six valeurs réelles  $P_0, P_1, P_2, P_3, P_4, P_5$  comme représenté dans la Figure 3.4. Où  $P_0$  et  $P_5$  sont fixes et représentent la valeur minimale et la valeur maximale de la variable d'entrée. Les autres valeurs sont utilisées pour la création des fonctions d'appartenance et chacun d'eux a ses propres limites telles que  $P_1$  a  $\{P_0, P_5\}$ ,  $P_2$  a  $\{P_0, P_5\}$ ,  $P_3$  a  $\{P_1, P_5\}$ , et  $P_4$  a  $\{P_2, P_5\}$ .

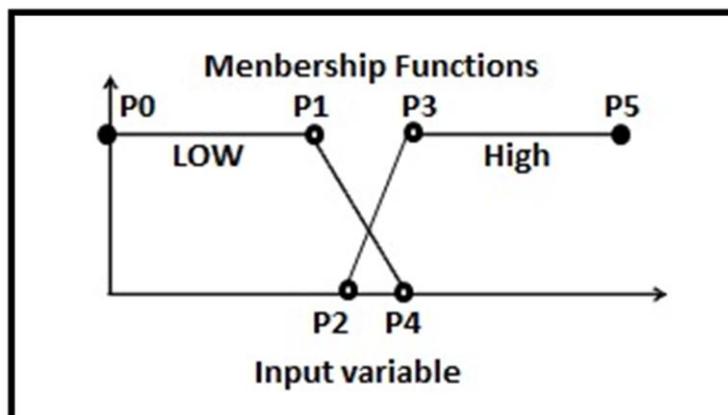


Figure 3.4: Fonction d'appartenance pour chaque variable d'entrée

Chaque individu est représenté en vue de déterminer un modèle de classification flou. La codification d'un modèle de classification flou est présentée dans la Figure 3.5 ou la solution  $S_i$  du problème est représentée. Cette solution comprend deux types de vecteurs de paramètres  $S_i = [R_i, f_i]$ . L'un est le paramètre  $f_i = [f_1^i, \dots, f_k^i, \dots, f_n^i]$  représenté par  $n$  bits binaires pour la sélection des variables. Si  $f_j^i = 0$ , la variable d'entrée  $k$ -ième est exclue des règles  $R_i = [R_1^i, \dots, R_j^i, \dots, R_M^i]$  sinon  $f_j^i$  est inclu ( $k = 1 \dots n$ ). L'autre vecteur  $R_i = [R_1^i, \dots, R_j^i, \dots, R_M^i]$  se compose d'un vecteur de contrôle  $m^{f_{jk}} \in \{1, 2\}$  pour spécifier le type de fonction d'appartenance (Low, high) défini pour  $A_{ji}$ , et un vecteur paramétrique. Le vecteur paramétrique représente quatre valeurs réelles des paramètres  $p_{jk,1}^i, p_{jk,2}^i, p_{jk,3}^i, p_{jk,4}^i$  qui déterminent les points des fonctions d'appartenance. L'étiquette  $l_j^i$  est représentée par un bit à 0/1 signifie que la règle correspondante est utilisée ou non utilisée dans la base de règles codées dans la solution correspondante.

Après que les parties « antécédent » ont été créées, nous avons sélectionné la partie « conclusion » appropriée à chaque partie « antécédent » par la procédure suivante [49]:

**Step1:** Calculate the compatibility of each training pattern  $x_p(x_{p1}, \dots, x_{pn})$  for the  $j^{th}$  fuzzy rule as:

$$\mu_j(x_p) = \mu_{j1}(x_{p1}) * \mu_{j2}(x_{p2}) * \dots * \mu_{jn}(x_{pn})$$

**Step2:** For each class  $C$  ( $C=1 \dots M$ ) calculate the relative sum of the compatibility grades for the  $j^{th}$  rule

$$\alpha_{\text{class } c} (R_j) = \sum_{x_p \in \text{class } c} \mu_j(x_p)$$

**Step3:** Find class  $C$  for the  $j^{th}$  rule with the maximum value of  $\alpha_{\text{class } c}$

$$\alpha_{\text{class } c} (R_j) = \text{argmax} \{ \alpha_{\text{class } i} (R_j) \mid i = 1, 2, \dots, M \}$$

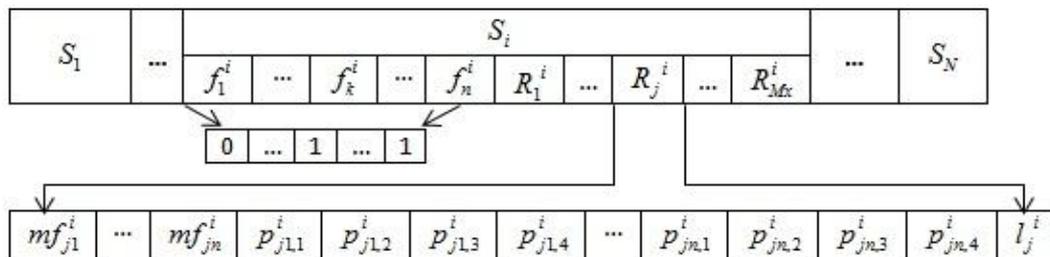


Figure 3.5: La structure de codage de la solution

### 4.3 L'initialisation de la Population

L'initialisation de la population est une tâche importante dans le processus de recherche de méta-heuristiques; cela signifie qu'une bonne initialisation peut conduire à une convergence rapide vers la solution optimale. Dans la méthode proposée, nous remplaçons l'initialisation aléatoire par la méthode d'initialisation basée sur l'opposition en vue d'améliorer l'exploration de l'algorithme standard et garantir une population initiale avec une certaine qualité et diversité. Le détail du mécanisme d'initialisation est décrit dans le tableau 3.3.

**Tableau 3.3: Algorithme d'initialisation basée sur l'opposition**

- (1). Initialize population randomly.**
- (2). Calculate opposite of each solution in the population by:**

$$OPu_{ij} = u_{\min,j} + u_{\max,j} - u_{ij}$$
- (3). Selecting  $SN$  fittest individuals from  $OPu(SN) \cup u(SN)$  as initial population**

### 4.4 Fonction d'évaluation

De la même manière que les autres méta-heuristiques, nous avons besoin d'une fonction d'évaluation des solutions appelées fonction objective. Dans cette thèse la fonction d'évaluation pour le modèle de classification flou est définie comme suit:

$$fit(S_i) = C_R(S_i)^2 * (L - Num_R(S_i) + 1)$$

Où  $s_i$  est la  $i^{\text{ème}}$  solution,  $C_R(s_i)$  est le taux de classification calculé par :

$$\text{Classification Rate} = 100 * \frac{TP + TN}{TN + TP + FN + FP}$$

Où VP, VN, FP et FN désignent respectivement : vrais positifs, vrais négatifs, faux positifs et de faux négatifs. L est le nombre de règles générées à l'étape précédente, et  $Num_R(s_i)$  est le nombre de règles actives. De cette manière, nous assurons un meilleur taux de classification avec un petit nombre de règles floues.

## 5 Modèle 1: conception d'un classifieur flou par l'algorithme de colonie d'abeilles artificielles (MABC)

Pour ce modèle, une nouvelle méta-heuristique a été appliquée pour la reconnaissance de la maladie du diabète, qui est l'une des maladies les plus importantes. Nous avons proposé un nouvel algorithme de la colonie d'abeilles artificielles dans lequel un opérateur de mutation est ajouté à l'algorithme standard pour améliorer sa performance. Lorsqu'une meilleure solution ne peut pas convergée vers une autre, l'opérateur de croisement mixte (BLX- $\alpha$ ) [144] de l'algorithme génétique est appliqué, pour augmenter la diversité dans l'algorithme ABC. Cette version modifiée d'algorithme ABC est utilisée pour créer et optimiser automatiquement les fonctions d'appartenance et la base de règles d'un classifieur flou.

### 5.1 Opérateur de croisement mixte : BLX- $\alpha$

Les Opérateurs de croisement sont capables de produire une exploration ou une exploitation en fonction de la manière dont ils gèrent la diversité dans la population. Ils peuvent générer une diversité supplémentaire à partir des solutions existantes, Où l'utilisation de cette diversité permet de créer d'autres solutions meilleures. L'exploration introduite par ces opérateurs est limitée au voisinage des parents, si l'opérateur établit un bon équilibre entre l'exploration et l'exploitation, il est possible d'éviter la perte de la diversité et garantie une bonne convergence.

Le choix de cet opérateur BLX- $\alpha$  [144] est justifié par son comportement positif dans les études expérimentales précédentes et il peut générer des individus dans les deux régions d'exploration et d'exploitation. Le pseudo-code pour générer une nouvelle solution par l'opérateur de croisement mixte (BLX- $\alpha$ ) est résumé dans le tableau 3.4.

<b>Tableau 3.4: Algorithme d'opérateur de croisement mixte (BLX-<math>\alpha</math>)</b>	
<b>1:</b>	<b>Choose two solutions <math>x_1, x_2</math> randomly from the population</b>
<b>2:</b>	<b>A value of new solution <math>Y</math> is generated by :</b>
	$Y = \begin{cases} a + r * (b - a) & : \text{if } x_{\min} \leq Y \leq x_{\max} \\ \text{repeat generating} & : \text{otherwise} \end{cases}$
<b>Where:</b>	$a = x_1 - \alpha(x_2 - x_1)$ $b = x_2 - \alpha(x_2 - x_1)$ $r : \text{random number in } [0,1]$ $x_{\min}, x_{\max} \text{ represent the variable's lower and upper bounds respectively and } \alpha \text{ is a positive parameter}$

## 5.2 Principe de l'algorithme ABC modifié (MABC)

Dans l'algorithme ABC, une source de nourriture représente une solution ou chaque abeille employée (abeille spectateur) génère une nouvelle source de nourriture dans un espace de recherche de sorte que la nouvelle solution candidate peut-être bonne ou mauvaise selon la fonction d'évaluation. Cela signifie que s'il y a une amélioration dans la fonction d'évaluation, la nouvelle solution candidate est bonne et elle remplace l'ancienne solution dans l'itération suivante. Sinon, la nouvelle solution candidate est mauvaise et elle peut poser un problème dans la convergence vers la solution optimale. Cela est dû au mécanisme de mise à jour de solution qui est limité dans la génération d'une variété de solutions. Dans une pareille situation, il est nécessaire d'introduire un nouveau mécanisme de recherche qui peut aider à produire une grande variété de solutions qui conduisent à atténuer la faiblesse du mécanisme de mise à jour. Cela peut être fait par l'intégration d'un opérateur de croisement mixte (BLX- $\alpha$ ) de l'algorithme

génétique dans l'algorithme ABC afin d'améliorer l'exploitation et la capacité d'exploration des abeilles employées et les abeilles spectateurs. La version proposée de l'algorithme ABC modifiée se distingue des autres versions dans le sens où s'il n'y a pas une amélioration dans la fonction d'évaluation dans la phase des abeilles employées ou spectateurs, les solutions sont mutées en utilisant l'opérateur de croisement mixte.

### 5.3 Présentation de l'algorithme ABC modifié (MABC)

Le pseudocode de l'algorithme ABC modifié proposé est donné ci-dessous dans le tableau 3.5:

<b>Tableau 3.5: L'Algorithme de La Colonie d'Abeilles Artificielles modifié (MABC)</b>	
1: Parameter initialization:	<i>Number of Solutions: SN, number of Cycle: C The parameters (limit).</i>
2: Initialize the population of solutions $x_i$ ; $i = 1; \dots; SN$ randomly.	
3: Evaluate the fitness function $fit(x_i)$ of the population.	
4: $C = 1$	
Repeat	
	<b>{--Employed Bees' Phase--}</b>
5: Produce new solutions $v_i$ for the employed bees and evaluate them ( $fit(v_i)$ ).	
	<b><i>If there is no improvement in fitness function, perform Algorithm of BLX-<math>\alpha</math> to produce new candidate solutions <math>v_i</math> and evaluate them (<math>fit(v_i)</math>).</i></b>
6: Apply the greedy selection process for the employed bees	
7: Calculate the probability values $P_i$ for the solutions $x_i$	
8: Produce the new solutions $v_i$ for the onlookers from the solutions $x_i$ selected depending on $P_i$ and evaluates them ( $fit(v_i)$ )	
	<b>{--Onlooker Bees' Phase--}</b>
	<b><i>If there is no improvement in fitness function, perform Algorithm of BLX-<math>\alpha</math> to produce new candidate solutions <math>v_i</math> and evaluate them (<math>fit(v_i)</math>)</i></b>
9: Apply the greedy selection process for the onlookers	

{--Scout Bee Phase--}
10: Determine the abandoned solution for the scout, if exists, and replace it with a new randomly produced solution $x_i$
11: Memorize the best solution achieved so far
12: $C = C + 1$
Until ( $C = \text{Maximum Cycle Number}$ )

## 5.4 Résultats et discussion

La performance de la méthode proposée a été testée sur la base de données Pima selon les critères suivants : le taux de classification, la sensibilité et la spécificité. Deux expérimentations ont été réalisées dans cette étude. Dans la première expérimentation, le vecteur d'entrée a été utilisé avec toutes les caractéristiques dans la seconde expérimentation, la sélection des caractéristiques d'entrée a été appliquée et les meilleures caractéristiques pour chaque base de règles ont été choisies par l'algorithme ABC modifié.

Les valeurs des paramètres optimaux d'algorithme ABC modifié dans le processus d'évolution sont présentées dans le Tableau 3.6.

<b>Tableau 3.6: Spécification des paramètres dans l'algorithme proposé</b>	
Spécification des paramètres	Valeur
<b>Nombre de règles dans chaque solution " <math>M_x</math> "</b>	20
<b>Nombre de source de nourriture pour les abeilles employées et spectateurs</b>	100
<b>Nombre maximum de cycle " <math>C</math> "</b>	200
<b>Valeur de paramètre « limite »</b>	10
<b>Valeur de paramètre " <math>\alpha</math> "</b>	0,5

Les valeurs de paramètres ont été déterminées expérimentalement par le suivi des changements de valeurs de la fonction objective, afin de donner une meilleure performance de classification.

5.4.1 Résultats des tests

Après la réalisation d'une validation croisée (10 fois) les résultats obtenus par l'algorithme ABC modifié en termes de taux de classification, nombre de règles et le nombre de variables pour tous les sous-ensembles de données sont présentés dans le Tableau 3.7.

**Tableau 3.7: Les résultats obtenus pour la base PIMA en utilisant 10 fois-validation croisée.**

Toutes les variables				La sélection des variables			
Sous ensemble	Ensemble d'apprentissage	Ensemble de test	Nombre de règles	Ensemble d'apprentissage	Ensemble de test	Nombre de règles	Num de variables
1	79,71	79.48	11	83.18	82.05	7	4
2	84.05	83.33	12	83.76	84.61	5	3
3	82.60	80.76	13	83.33	85.89	9	4
4	83.33	85.89	14	84.04	83.33	9	5
5	82.60	83.33	12	84.63	85.89	7	3
6	83.76	82.05	10	83.91	84.61	8	4
7	81.15	82.05	11	84.34	83.33	10	5
8	83.33	84.61	14	84.20	83.33	3	2
9	80.00	83.33	13	84.78	83.33	6	4
10	82.83	82.05	12	84.49	85.89	7	3
<b>Moy</b>	<b>82.32</b>	<b>82.68</b>	<b>12</b>	<b>84.20</b>	<b>84.21</b>	<b>7</b>	<b>4</b>

Selon le Tableau 3.7, nous remarquons un petit nombre de règles floues sélectionnées avec une moyenne de 12 règles (sans sélection de caractéristiques). Le nombre de caractéristiques et les règles utilisées par les différents sous-ensembles de données Lors de la sélection de caractéristiques varient de façon significative avec une moyenne de 4 caractéristiques et 7 règles. La différence dans le nombre de règles et le nombre de caractéristiques entre les sous-ensembles est due à la modification des données d'apprentissage à partir d'un sous-ensemble de données à l'autre.

5.4.2 La connaissance extraite par l'algorithme ABC modifié

Nous savons que le nombre de règles floues si-alors augmente de façon exponentielle selon le nombre d'attributs et/ou le nombre des fonctions

d'appartenances. Dans la conception d'un classificateur flou, l'interprétabilité de l'ensemble de règles est considérée comme un facteur important. Cette interprétabilité est mesurée en fonction de nombre de règles; cela signifie que les classifieurs flous contenant moins de règles floues restent toujours interprétables avec une simple lisibilité.

Dans cette partie, chaque attribut d'entrée a deux fonctions d'appartenance, donc 256 règles floues ont été extraites logiquement, la complexité globale de la base de connaissances des classificateurs flous augmente d'une manière exponentielle. Pour réduire cette complexité, nous avons appliqué l'algorithme ABC modifié.

Le Tableau 3.8 représente les règles floues en présence de l'ensemble des caractéristiques existantes d'une part et seulement les caractéristiques sélectionnées d'autre part pour les résultats obtenus par le deuxième sous-ensemble de données.

**Tableau 3.8: les règles floues obtenues par le deuxième sous-ensemble de données pour la base PIMA**

*Les règles avec l'ensemble des caractéristiques :*

**R1:** *If (Npreg is H) and (Glu is L) and (BP is H) and (Skin is L) and (Insulin is L) and (BMI is H) and (PED is H) and (Age is L) then (Class is Non- diabetic)*

**R2:** *If (Npreg is L) and (Glu is L) and (BP is L) and (Skin is L) and (Insulin is L) and (BMI is L) and (PED is H) and (Age is L) then (Class is Non- diabetic)*

**R3:** *If (Npreg is H) and (Glu is L) and (BP is H) and (Skin is L) and (Insulin is L) and (BMI is H) and (PED is L) and (Age is L) then (Class is Non- diabetic)*

**R7:** *If (Npreg is L) and (Glu is L) and (BP is L) and (Skin is L) and (Insulin is L) and (BMI is L) and (PED is L) and (Age is L) then (Class is Non- diabetic)*

**R8:** *If (Npreg is H) and (Glu is H) and (BP is H) and (Skin is L) and (Insulin is H) and (BMI is L) and (PED is H) and (Age is L) then (Class is diabetic)*

**R9:** *If (Npreg is L) and (Glu is H) and (BP is L) and (Skin is L) and (Insulin is H) and (BMI is H) and (PED is H) and (Age is H) then (Class is diabetic)*

<p><b>R4:</b> <i>If (Npreg is L) and (Glu is L) and (BP is L) and (Skin is H) and (Insulin is L) and (BMI is L) and (PED is L) and (Age is L) then (Class is Non-diabetic)</i></p>	<p><b>R10:</b> <i>If (Npreg is L) and (Glu is H) and (BP is L) and (Skin is H) and (Insulin is L) and (BMI is L) and (PED is L) and (Age is L) then (Class is diabetic)</i></p>
<p><b>R5:</b> <i>If (Npreg is H) and (Glu is L) and (BP is H) and (Skin is L) and (Insulin is H) and (BMI is L) and (PED is L) and (Age is L) then (Class is Non-diabetic)</i></p>	<p><b>R11:</b> <i>If (Npreg is H) and (Glu is H) and (BP is H) and (Skin is H) and (Insulin is H) and (BMI is H) and (PED is H) and (Age is H) then (Class is diabetic)</i></p>
<p><b>R6:</b> <i>If (Npreg is L) and (Glu is L) and (BP is H) and (Skin is L) and (Insulin is L) and (BMI is L) and (PED is L) and (Age is L) then (Class is Non-diabetic)</i></p>	<p><b>R12:</b> <i>If (Npreg is H) and (Glu is H) and (BP is H) and (Skin is H) and (Insulin is L) and (BMI is H) and (PED is L) and (Age is H) then (Class is diabetic)</i></p>
<p><b>Les règles avec quelques caractéristiques sélectionnées:</b></p>	
<p><b>R1:</b> <i>if (Glu is L) and (BMI is L) and (Age is L) then (Class is Non-diabetic)</i></p>	
<p><b>R2:</b> <i>if (Glu is L) and (BMI is H) and (Age is L) then (Class is Non-diabetic)</i></p>	
<p><b>R3:</b> <i>if (Glu is H) and (BMI is L) and (Age is L) then (Class is diabetic)</i></p>	
<p><b>R4:</b> <i>if (Glu is H) and (BMI is H) and (Age is L) then (Class is diabetic)</i></p>	
<p><b>R5:</b> <i>if (Glu is H) and (BMI is H) and (Age is H) then (Class is diabetic)</i></p>	
<p><b>Remarque:</b> L=LOW, H=HIGH, Npreg=Number of times pregnant, Glu=Plasma glucose concentration, BP=Diastolic blood pressure, Skin=Triceps skin fold thickness, BMI=Body mass index, PED=Diabetes pedigree function.</p>	

L'initialisation des points modaux dans la première itération et les fonctions d'appartenance optimales (points modaux optimaux) obtenus après 200 itérations pour ces caractéristiques sont présentées dans Le tableau 3.9 :

### Chapitre 3 : Résultats et discussion

	L'initialisation des points modaux dans la première itération		Les points modaux optimaux obtenus après 200 itérations	
	Low	High	Low	High
Npreg	0-2-15	1-18-17	0-5-10	1-6-17
Glu	56-120-171	160-190-199	56-108.2-160	92.5-139-199
BP	24-55-60	28-90-122	24-75-110	57-80-122
SKIN	7-9-30	15-48-99	7-35-62	9-42-99
Insuline	14-225-580	100-300-846	14-120-510	50-520-846
BMI	18.2-59-65.3	58.1-60-67.1	18.2-33.5-60	20.1-35-67.1
PED	0.078-1.20-2.30	1.80-2.39-2.42	0.078-1.24-1.8	0.45-1.71-2.42
Age	21-30-48	25-50-81	21-30-60	21-42-81

**Tableau 3.9: représentation des points modaux.**

Le Tableau 3.10 présente la matrice de confusion qui montre les résultats de classification du deuxième sous-ensemble (de la validation croisée) pour la base de données diabète. Dans ce cas, le taux de classification obtenu avec l'algorithme ABC modifiée était de 83,33% en utilisant toutes les variables. L'algorithme ABC modifié avec la sélection de variables obtient un taux de classement de 84,61%.

<b>Tableau 3.10: Matrices de confusion de l'algorithme ABC modifié</b>			
Réel	Prédit		Méthode
	Les diabétiques	Non-diabétiques	
Les diabétiques	22	4	ABC Modifié avec toutes les caractéristiques
Non-diabétiques	9	43	
Les diabétiques	21	5	ABC Modifié avec la sélection de caractéristiques
Non-diabétiques	8	44	

### ➤ Discussion sur les cas erronés

Dans ce que suit nous allons présenter des cas de FP et de FN de la base de données Pima avec notre méthode proposée MABC ainsi que les règles activées avec ces exemples. Notez bien que ce sont ces cas-là qui ont induit une baisse du taux de classification. Nous n'allons pas détailler ces cas-là avec la base de données Breast Cancer vu que le taux de classification de cette base frotte les 99% de bonne classification.

Nous allons prendre l'un des résultats et voir là où notre modèle a commis des erreurs. Nous avons pris le deuxième sous ensemble de la validation croisée qui contient 78 exemples de test. Les résultats sont résumés comme suit :

### ➤ Exemple de FN

Cet exemple présente les valeurs de différentes caractéristiques d'un cas diabétique dans la base PIMA

	Npreg	Glu	Bp	SKIN	Insuline	BMI	PED	Age	class
Patient	5	85	74	22	79.79	29	1,224	32	1

Pour cet exemple la deuxième règle floue est activée avec un degré de vérité de 93%. Il est reconnu comme Non diabétique par notre classifieur ce qui est logique vue les valeurs des différents paramètres.

### ➤ Exemple de FP

Cet exemple présente les valeurs de différentes caractéristiques d'un cas Non diabétique dans la base PIMA.

	Npreg	Glu	Bp	SKIN	Insuline	BMI	PED	Age	class
Patient	4	197	70	39	744	36,7	2,329	31	0

Pour cet exemple la neuvième règle floue est activée avec un degré de vérité de 75%. Il est reconnu comme diabétique par notre classifieur ce qui est logique vue les valeurs des différents paramètres

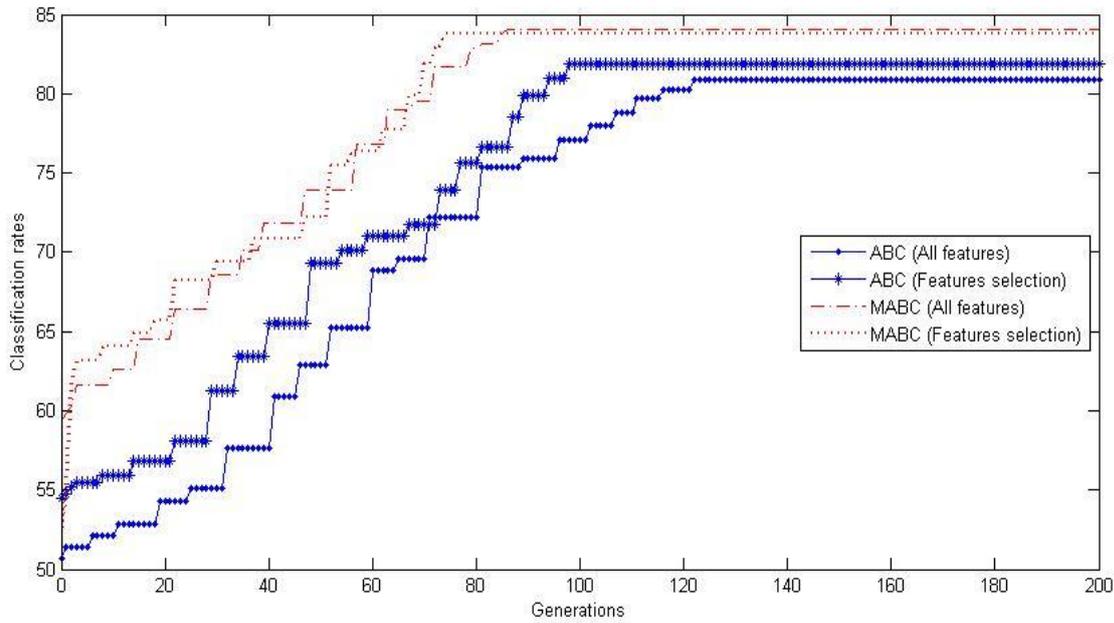
5.4.3 Comparaison entre L'algorithme ABC et l'algorithme ABC modifié

Pour évaluer la performance de l'algorithme ABC modifié, l'algorithme de base ABC est développé et la performance de chaque algorithme est présentée dans le tableau 3.11. Dans nos expérimentations, nous avons utilisé les mêmes paramètres pour chaque problème de test.

**Tableau 3.11: Les résultats de la comparaison entre ABC et MABC**

Toutes les variables						
	MABC			ABC		
	Meilleur	Pire	Moyenne	Meilleur	Pire	Moyenne
<b>Taux Classification (%)</b>	85.89	78.48	<b>82.68</b>	82.05	76.98	<b>79.61</b>
<b>Sensibilité (%)</b>	84.61	76,92	<b>81.91</b>	80.76	73.92	<b>78.54</b>
<b>Spécificité (%)</b>	86.53	78.84	<b>83.07</b>	82.69	76,92	<b>80.18</b>
La sélection des variables						
	MABC			ABC		
	Meilleur	Pire	Moyenne	Meilleur	Pire	Moyenne
<b>Taux Classification (%)</b>	85.89	82.05	<b>84.21</b>	81.88	80.43	<b>81.40</b>
<b>Sensibilité (%)</b>	84.61	80.76	<b>83.45</b>	84.61	76,92	<b>80.76</b>
<b>Spécificité (%)</b>	86.53	80.76	<b>84.60</b>	84.61	80.76	<b>82.88</b>

Selon le tableau 3.11, les meilleurs résultats ont été obtenus avec la version modifiée de l'algorithme ABC en termes de moyenne et de taux de classification. Le meilleur résultat a été de 85,89% et le résultat moyen était de 82,68% lors de l'utilisation de toutes les caractéristiques. Le meilleur résultat lors de l'utilisation de la sélection de caractéristiques a été 85,89% et le résultat moyen était de 84,21%. Par rapport à l'algorithme ABC de base, l'algorithme ABC modifié a amélioré le taux de classification par une moyenne de 3,07% (en utilisant toutes les caractéristiques) et 2,81% (en utilisant la sélection de caractéristiques), d'autre part, le meilleur taux de classification a été amélioré par 3,84% (en utilisant toutes les caractéristiques) et 4,0% (en utilisant la sélection de caractéristiques). La convergence pour le deuxième sous-ensemble de données de l'algorithme ABC modifié et l'algorithme ABC de base a été également étudiée. Ceci est illustré dans la Figure 3.6.



**Figure 3.6: La convergence de l'algorithme ABC et l'algorithme ABC modifié pour la base de données diabète (sous-ensemble 2)**

De cette figure, nous remarquons que l'algorithme ABC de base prend plus de générations par rapport à l'algorithme ABC modifié (plus de générations, signifie plus d'itérations et ainsi plus de temps d'exécution), l'algorithme ABC n'est pas en mesure d'atteindre la convergence avant la 122<sup>ème</sup> génération en utilisant toutes les caractéristiques et la 99<sup>ème</sup> génération en utilisant une sélection de caractéristiques. L'algorithme ABC modifié prend moins de générations (converge à la 88<sup>ème</sup> génération en utilisant toutes les caractéristiques et à la 75<sup>ème</sup> génération Lors de la sélection de caractéristiques). Par conséquent, les résultats montrent que l'algorithme ABC modifié est plus efficace que l'algorithme ABC, et la solution finale obtenue par l'algorithme ABC modifié est meilleure que celle de l'algorithme ABC dans la plupart des cas.

Pour justifier d'avantage l'efficacité de l'approche proposée, une autre base de données a été testée (breast cancer). Les résultats de la méthode proposée sur le cancer du sein en termes de taux de classification, nombre de règles, nombre de caractéristiques pour tous les sous-ensembles sont présentés dans le Tableau 3.12.

**Tableau 3.12: Les résultats obtenus pour la base breast cancer en utilisant 10 fois-validation croisée**

Sous ensemble	base de données sur le cancer du sein						
	Toutes les variables			La sélection des variables			
	Ensemble d'apprentissage	Ensemble de test	Nombre de règles	Ensemble d'apprentissage	Ensemble de test	Nombre de règles	Num de variables
<b>1</b>	99.18	98,55	8	98.69	98,55	7	5
<b>2</b>	98.69	97.10	9	98.04	98,55	6	3
<b>3</b>	98,37	98,55	11	98,37	97.10	5	4
<b>4</b>	98.04	98,55	13	99.18	98,55	8	7
<b>5</b>	97.71	97.10	12	98,37	98,55	5	3
<b>6</b>	98.20	98,55	11	99.18	97.10	6	4
<b>7</b>	98,37	97.10	9	98.85	98,55	5	3
<b>8</b>	98.85	97.10	14	99.02	98,55	6	4
<b>9</b>	99.02	97.10	10	97.71	97.10	9	6
<b>10</b>	97.71	98,55	11	98.04	98,55	4	3
	<b>98.41</b>	<b>97.82</b>	<b>11</b>	<b>98.54</b>	<b>98.11</b>	<b>6</b>	<b>4</b>

D'après ce tableau, nous pouvons voir que la précision de la classification obtenue par notre méthode est de 97,82% avec une moyenne de 11 règles (sans sélection de caractéristiques), et 98.11% avec une moyenne de 6 règles (et 4 variables lors de la sélection des caractéristiques). S'appuyant sur les résultats présentés dans le tableau 3.12, Il est montré que notre méthode proposée est conforme avec le raisonnement du médecin et fournit des informations explicatives à l'aide de quelques règles floues.

## **6 Modèle 2: conception d'un classifieur flou par l'algorithme d'optimisation par essaim particulière (MPSO)**

Dans le domaine médical, l'interprétabilité représente un élément clé lors de la mise en œuvre des classifieurs flous. L'expert devrait être en mesure de comprendre le processus de raisonnement du classifieur et d'évaluer ses résultats obtenus. L'utilisation d'OEP et le classifieur flou est une nouvelle méthode que nous proposons pour explorer efficacement un grand espace de recherche. Cette technique permet de réaliser simultanément la modification des règles et la

réduction de la taille de la base des règles. Pour ce modèle, nous avons modifié l'algorithme OEP afin de construire une base de règles floues optimale.

### 6.1 Principe de l'algorithme OEP modifié (MPSO)

L'algorithme OEP modifié diffère de la version originale sur deux points : Tout d'abord, nous avons changé l'initialisation aléatoire par la méthode d'initialisation basée sur l'opposition et d'autre part nous avons intégré dans l'algorithme OEP l'opérateur de croisement mixte qui a été utilisé dans le modèle précédent (MABC), pour améliorer la diversité sans compromettre la qualité de la solution. En outre, l'algorithme OEP modifié permet de générer l'ensemble de règles les plus représentatives.

Dans la version de base de l'algorithme OEP, la mise à jour de la vitesse et la position sont deux opérations majeures pour produire une nouvelle solution candidate, donc s'il n'y a aucune amélioration dans la valeur de la fonction objective d'un individu, la convergence vers la solution optimale n'est pas garantie. La limite du mécanisme de mise à jour de solution concerne la création d'une variété de solutions. Pour pallier à cette limite, nous avons inclus l'opérateur de croisement mixte (BLX- $\alpha$ ) de l'algorithme génétique pour induire la diversité dans la population.

### 6.2 Présentation de l'algorithme d'OEP modifié

Le pseudo-code de l'algorithme OEP modifié proposé est donné ci-dessous dans le tableau 3.13:

**Tableau 3.13: L'Algorithme de L'Optimisation par Essaim Particulaire modifié (MPSO)**

1: Initialization:

- Number of particles  $L$ , number of iterations  $G$ , and the constants for the OEP ( $\gamma$ ,  $C1$  and  $C2$ ),
- generate randomly initial velocity for all particles of the population

2: Calculate the fitness value of each particle:

```

    fiti=fit(Pi)
- Update Pbesti, Fbesti, PGbest and FGbest

    If fiti > Fbesti then Fbesti = fiti and Pbesti= Pi

    If fiti > FGbest then FGbest= fiti and PGbest = Pi

    Else

//BLX-α

    Find a new particle using BLX-α
    Let this particle as NewP
    If fit(NewP) > FGbest then FGbest= fit(NewP)
    and PGbest =NewP

- Set G=1
3:
- Update particles velocity by:   Vi = γVi + C1r1 (Pbesti - Pi) + C2r2 (PGbest - Pi)
- Update particles position by:   Pi = Pi + Vi

4: G = G + 1.
    -If G > number of iterations then go to Step 5 else go to Step 2.

5: -The optimal solution is desired by the best particle PGbest with the best fitness FGbest.

```

### 6.3 Résultats et discussion

Dans ce modèle, nous avons validé notre algorithme sur deux bases de données distinctes : cancer du sein et diabète. Les performances de la méthode proposée ont été testées par le taux de classification, la sensibilité et la spécificité du classifieur flou. Le taux de classification moyen obtenu par notre algorithme MPSO est de 82.55%, et de 97.82% pour le cancer du sein et le diabète respectivement en utilisant la validation croisée. Le processus divise la base de données en 10 sous-ensembles de même taille, ou chaque sous-ensemble sera utilisé une fois pour le test, tandis que les autres (9 sous-ensembles restants) sont appliquées pour la construction de la base de règles. Les valeurs des paramètres optimaux de l'algorithme OEP modifié impliqués dans le processus d'évolution sont présentées dans le Tableau 3.14. Ces valeurs ont été déterminées

expérimentalement en suivant les changements des valeurs de la fonction objective, afin d'obtenir une meilleure performance de classification.

**Tableau 3.14: spécification des paramètres dans l'algorithme MPSO**

Spécification des paramètres	Valeur
nombre de règles dans chaque particule	20
Nombre de particules	50
Nombre maximum d'itérations	80
constantes d'accélération: $C_1$ Et $C_2$	2
La valeur du paramètre " $\alpha$ "	0,45
Le poids de l'inertie: $\gamma$	0,7

Selon ces paramètres, les résultats obtenus par l'algorithme OEP modifié sur les deux bases de données en termes de taux de classification, nombre de règles pour tous les sous-ensembles sont présentés dans le tableau 3.15.

**Tableau 3.15: les résultats obtenus par l'algorithme MPSO**

base de données de diabète				base de données du cancer du sein			
Sous ensemble	Ensemble d'apprentissage	Ensemble de test	Nombre de règles	Sous ensemble	Ensemble d'apprentissage	Ensemble de test	Nombre de règles
1	84.98	87.17	14	1	99.18	98,55	10
2	81.86	79.48	8	2	98.69	97.10	9
3	83.56	82.05	9	3	98,37	98,55	9
4	83.00	84.61	11	4	98.04	98,55	11
5	82.15	84.61	13	5	97.71	97.10	9
6	83.00	84.61	10	6	98.20	98,55	10
7	83.28	79.48	12	7	98,37	97.10	9
8	81.86	76,92	8	8	98.85	97.10	11
9	82.43	82.05	11	9	99.02	98,55	12
10	84.70	84.61	12	10	97.71	97.10	9
<b>Moy</b>	<b>83.08</b>	<b>82.55</b>	<b>11</b>	<b>Moy</b>	<b>98.41</b>	<b>97.82</b>	<b>10</b>

D'après le tableau 3.15, les règles floues générées par l'algorithme OEP modifié ont un nombre relativement faible avec une moyenne de 10 règles pour le cancer du

sein et 11 règles pour le diabète. La différence dans le nombre de règles entre les sous-ensembles est due à la modification des données d'apprentissage à partir d'un sous ensemble à l'autre.

Le Tableau 3.16 résume les résultats et la comparaison entre l'algorithme standard OEP (PSO) et notre algorithme modifié (MPSO) en terme de la sensibilité, la spécificité et le taux de classification.

<b>Tableau 3.16: Les résultats de la comparaison entre PSO et MPSO</b>						
<b>base de données de diabète</b>						
	<b>MPSO</b>			<b>PSO</b>		
	<b>Meilleur</b>	<b>Pire</b>	<b>Moyenne</b>	<b>Meilleur</b>	<b>Pire</b>	<b>Moyenne</b>
<b>Taux</b>	87.17	76,92	82.55	82.05	74.35	78.71
<b>Classification</b>						
<b>Sensibilité</b>	84.61	69,23	79,99	84.61	69,23	76,92
<b>Spécificité</b>	88.46	80.76	83.84	84.61	80.76	81.53
<b>base de données sur le cancer du sein</b>						
	<b>MPSO</b>			<b>PSO</b>		
	<b>Meilleur</b>	<b>Pire</b>	<b>Moyenne</b>	<b>Meilleur</b>	<b>Pire</b>	<b>Moyenne</b>
<b>Taux</b>	98,55	97.10	97.82	97.10	92.75	95.07
<b>Classification</b>						
<b>Sensibilité</b>	100	92.00	97,60	96.00	92.00	94.00
<b>Spécificité</b>	100	95.45	97,95	97.72	93.18	95.90

La convergence du deuxième sous-ensemble de données de l'algorithme OEP modifié et l'algorithme standard OEP est également étudiée. Ceci est illustré dans la Figure 3.7 et la Figure 3.8, pour les deux bases de données. A partir de ces deux figures, nous pouvons voir que pour l'algorithme OEP modifié, le taux de classification converge rapidement vers la meilleure solution (à la 20<sup>ème</sup> génération pour le cancer du sein et à la 15<sup>ème</sup> génération pour le diabète). Par contre, l'algorithme standard OEP n'est pas en mesure de converger vers la meilleure solution avant la 30<sup>ème</sup> génération pour l'ensemble de données du cancer du sein et la 20<sup>ème</sup> génération pour l'ensemble de données du diabète. Par conséquent, il est

évident que l'OEP modifié est capable d'améliorer de manière significative la capacité de recherche de l'algorithme.

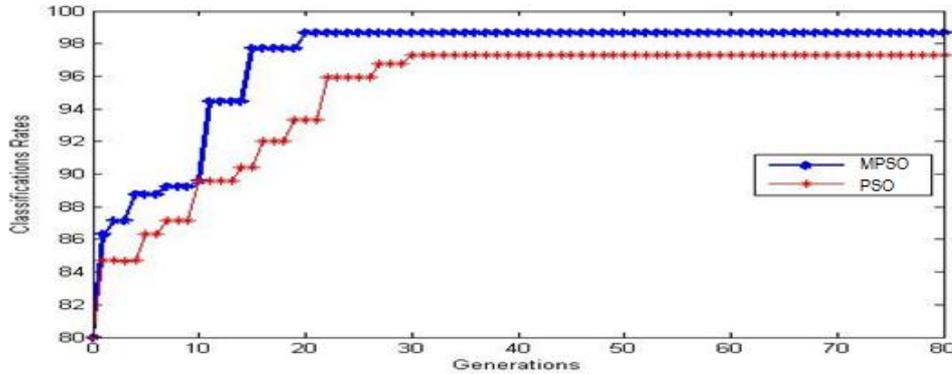


Figure 3.7: la Convergence du MPSO et OEP (cancer du sein (sous-ensemble 2))

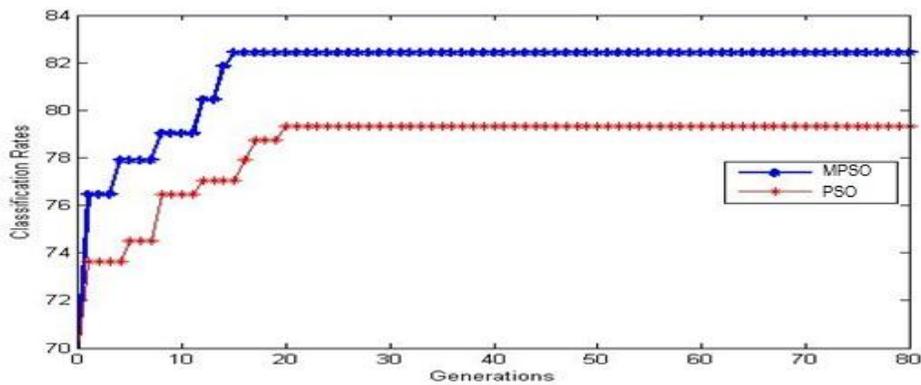


Figure 3.8: la Convergence du MPSO et OEP (Diabète (sous-ensemble 2))

## 7 Comparaison de l'approche proposée avec les travaux de la littérature

Les résultats obtenus pour la base de données Pima et la base de données cancer du sein par les autres chercheurs dans la littérature sont présentés dans le tableau 3.19 et le tableau 3.20.

**Tableau 3.17: taux de classification obtenu par notre classifieur et d'autres classifieurs de la littérature (base de données diabète)**

Méthodes	Précision (%)	Étude
Fuzzy-AIRS [138]	84.42	Polat <i>et al.</i> (2007)
Neuro-fuzzy inference system[141]	89.47	Polat <i>et al.</i> (2007)
GDA_LSSVM [138]	79.16	Polat <i>et al.</i> (2008)
Fuzzy Modeling [151]	77.65	Ghazavi <i>et al.</i> (2008)
BGA-Fuzzy-KNN [139]	81.6	Tand <i>et al.</i> (2009)
RGA-Fuzzy-KNN [139]	82	Tand <i>et al.</i> (2009)
ML-NN [142]	79.62	Temurtas <i>et al.</i> (2009)
PNN [142]	78.05	Temurtas <i>et al.</i> (2009)
GA-Fuzzy classifieur [146]	71.49	Guzaitis <i>et al.</i> (2009)
ACO-Fuzzy classifieur [83]	84.24	Ganji <i>et al.</i> (2011)
MABC (All features)	<b>82.68</b>	<b>Nos Travaux</b>
MABC (feature selection)	<b>84.21</b>	
ABC (All features)	<b>79.61</b>	
ABC (feature selection)	<b>81.40</b>	
MPSO (All features)	<b>82.55</b>	
PSO (All features)	<b>78.71</b>	

**Tableau 3.18: taux de classification obtenu par notre classifieur et d'autres classifieurs de la littérature (base de données cancer du sein)**

Methods	Accuracy (%)	Study
Fuzzy-GA method [128]	97.36	Pena-Reyes <i>et al.</i> (1999)
LVQ [129]	96.70	Goodman <i>et al.</i> (2002)
Big LVQ [129]	96.80	Goodman <i>et al.</i> (2002)
AIRS [129]	97.20	Goodman <i>et al.</i> (2002)
DPSO [85]	94.00	Sousa <i>et al.</i> (2004)
Constricted-PSO [85]	93.40	Sousa <i>et al.</i> (2004)
LDW-PSO [85]	93.00	Sousa <i>et al.</i> (2004)
GAs [147]	96.99	Chen <i>et al.</i> (2006)
FRBCs Framework [131]	96.08	Gadaras <i>et al.</i> (2009)
SVM [132]	97.54	Maglogiannis <i>et al.</i> (2009)
Bayesian classifiers [132]	92.80	Maglogiannis <i>et al.</i> (2009)
ANN [132]	97.90	Maglogiannis <i>et al.</i> (2009)
AS-NN [133]	95,60	Karabatak <i>et al.</i> (2009)
HMM-Fuzzy model [135]	97.07	Rafiul Hassan <i>et al.</i> (2010)
MABC	<b>97.82</b>	<b>Nos Travaux</b>
MPSO	98.55	

Nous pouvons remarquer à travers ces deux tableaux que les taux de classification obtenus par nos méthodes proposées sont améliorés par rapport aux autres classifieurs rapportés dans la littérature. Les règles floues obtenues par nos approches sont plus proches au raisonnement de l'expert médical. L'expert peut facilement confirmer les résultats obtenus par le classifieur flou, il peut aussi voir pourquoi quelques résultats de classification ont été affectés à un patient donné. Les résultats obtenus dans cette thèse sont très prometteurs et peuvent être utilisés pour aider le clinicien dans les problèmes de diagnostic. Nous pouvons conclure que les deux algorithmes proposés (MABC et MPSO) pourraient être très efficace dans la conception optimale d'un classifieur flou pour un système d'aide au diagnostic médical.

## 8 Conclusion

Dans ce travail nous avons présenté deux algorithmes efficaces et fiables pour la génération d'un classifieur flou, l'algorithme OEP modifié et l'algorithme ABC modifié. Ces deux algorithmes proposés ont été utilisés pour créer un classifieur flou optimal. Nous avons expliqué dans cette étude comment à la fois la base de règles et les fonctions d'appartenance sont optimisées simultanément. L'efficacité des deux algorithmes proposés a été confirmée par des tests sur deux bases de données médicales bien connues (cancer du sein et le diabète). Selon les résultats obtenus, nous dirons que les méthodes présentées permettent de proposer à la communauté des experts en médecine un outil qui permet de les aider à prendre des décisions avec le minimum d'erreur.

# Conclusion Générale

Les classifieurs sont largement appliqués dans le domaine médical, sachant que les données médicales représentent beaucoup de difficultés en termes de variabilité et de bruit. Nous nous sommes intéressés dans cette thèse à la conception automatique des classifieurs flous pour une classification interprétable des données médicales.

Le développement d'un classifieur efficace et transparent, qui offre un compromis acceptable entre l'interprétabilité et la précision, est une exigence majeure. Dans le monde réel, une grande partie de la connaissance est imprécise et ambiguë, mais la logique floue offre aux classifieurs une meilleure représentation de la connaissance, qui est proche du langage naturel en utilisant la notion des ensembles flous.

Les avantages des classifieurs flous se résument dans leur interprétabilité linguistique, traitement de l'incertitude dans les connaissances et la simulation du raisonnement humain, qui sont des facteurs importants pour l'acceptation et l'utilisation d'une solution. La présence des règles floues (Si-Alors) confirme la validité et l'acceptation des résultats obtenus par les experts du domaine.

La conception du classifieur flou nécessite la recherche d'une structure efficace et d'une base de connaissance cohérente. Deux approches différentes sont utilisées principalement pour la conception des modèles flous. Dans la première approche, la conception est construite à partir des connaissances acquises par l'expert, alors que dans la seconde, les données d'apprentissage sont utilisées pour la construction de modèle.

L'utilisation des algorithmes évolutionnaires a montré leur efficacité dans le réglage paramétrique et structurelle des classifieurs à partir des données d'apprentissage.

Dans le cadre de cette thèse, deux nouvelles approches basées sur les algorithmes évolutionnaires sont présentées pour ajuster les paramètres d'un classifieur flou

des données médicales afin d'augmenter ses performances du point de vue précision et interprétabilité. Notre souci majeur est d'assurer une classification des données médicale avec beaucoup de transparence.

Dans la première approche, Nous avons proposé un nouvel algorithme de la colonie d'abeilles artificielles dans lequel un opérateur de croisement mixte (BLX- $\alpha$ ) est ajouté à l'algorithme standard pour améliorer sa performance et augmenter sa diversité. Cette version d'algorithme ABC modifiée est utilisée pour créer et optimiser automatiquement les paramètres des fonctions d'appartenances et réduire le nombre de règles. Dans la deuxième approche, l'algorithme modifié d'optimisation par essaim particulière est proposé pour la conception d'un classifieur flou. Pour cet algorithme, nous avons proposé deux changements majeurs : tout d'abord, nous avons changé l'initialisation aléatoire par la méthode d'initialisation basée sur l'opposition puis nous avons intégré l'opérateur de croisement mixte (BLX- $\alpha$ ) dans un mécanisme de mise à jour de solution pour améliorer la diversité des solutions.

Nos deux approches nous ont permis d'apporter une valeur ajoutée dans les systèmes d'Aide au Diagnostic Médical avec comme objectif principal la transparence du modèle. Chaque donnée classifiée est justifiée par une ou plusieurs règles floues distinctes. Cette qualité est absente chez la majorité des classifieurs de données classiques sachant que l'interprétabilité des résultats peut convaincre les médecins d'utiliser largement les outils d'aide au diagnostic Médical dits Intelligents.

Les résultats obtenus dans ce travail de thèse nous ont permis de dresser les perspectives de recherche suivantes :

- Evaluation de nos approches proposées en utilisant des ensembles de données de grande dimension (passage à l'échelle).
- Amélioration des algorithmes proposés pour réduire la complexité de recherche et minimiser leur coût de calcul.
- Application de nos approches dans un processus d'apprentissage en ligne.

## Bibliographie

- [1] M. Fieschi, *intelligence Artificielle en Médecine : des systèmes experts*. Ed. MASSON, 1986.
- [2] J.W. Shavlik and T.G. Dietterich, *Readings in machine learning*, Morgan Kaufman Publishers, San Francisco, CA, 1990.
- [3] D. Michie, D.J. Spiegelhalter and C.C. Taylor, *Machine learning, neural and statistical classification*, Ellis Horwood, 1994.
- [4] T. Mitchell, *Machine Learning*, McGraw Hill, 1997
- [5] R.S. Michalski, I. Bratko, and M. Kubat, *Machine Learning, Data Mining and Knowledge Discovery: Methods and Applications*, John Wiley & Sons, 1998.
- [6] M.H. Dunham, *Data Mining: Introductory and Advanced Topics*, Prentice Hall, 2003.
- [7] B. Roy, *Méthodologie multicritère d'aide à la décision*, Paris : Economica, 1985.
- [8] D.B. Aronow, T.H. Payne and S.P. Pincetl, *Postdoctoral training in medical informatics: A survey of national library of medicine-supported fellows*, *Med Decis Making*, 11(1):29-32, February 1991.
- [9] E.S. Berner, R.S. Maisiak, C.G. Cobbs, and O.D. Taunton, *Effects of a decision support system on physicians' diagnostic performance*, *Journal of the American Medical Informatics Association : JAMIA*, 6(5):420-427, 1999.
- [10] G. Kong, D.L. Xu, and J.B. Yang, *Clinical decision support systems: a review on knowledge representation and inference under uncertainties*, *International Journal of Computational Intelligence Systems*, 1(2):159-167, 2008.
- [11] L.A. Stojkowska and S. Loskovska, *Clinical decision support systems: medical knowledge acquisition and representation methods*, in *Electro/Information Technology (EIT)*, 2010 IEEE International Conference on, p. 1-6.
- [12] K. Kawamoto, C.A. Houlihan, E.A. Balas and D.F. Lobach, *Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success*, *BMJ*, 330(7494):765, 2005.
- [13] P. Degoulet and M. Fieschi, *Traitement de l'information médicale : Méthodes et applications hospitalières*, Edition Masson, 1991.
- [14] F. Steimann, *On the use and usefulness of fuzzy sets in medical AI*, *Artificial Intelligence in Medicine*, 21:131-7, 2001.

- [15] L.A. Zadeh, *Biological application of the theory of fuzzy sets and systems*, In: Proceedings of international symposium on biocybernetics of the central nervous system. Boston, USA: Little Brown and Co, 199–212, 1969.
- [16] R.S. Ledley and L.B. Lusted, *symbolic logic, probability, and value theory aid our understanding of how physicians reason*. Reasoning foundations of medical diagnosis, 130(3366):9–21, 1959.
- [17] D.A. Zighed, R. Rakotomalala, *Graphes d'induction (Apprentissage et Data Mining)*. Edition hermès, 25-27, 2000.
- [18] G. Huang, S. Song, J. Gupta, and C. Wu, *Semi-supervised and unsupervised extreme learning machines*, Cybernetics, IEEE Transactions, 44(12): 2405-2417, Dec 2014.
- [19] X. Zhu and A.B. Goldberg, *Introduction to semi-supervised learning*. Morgan & Claypool, 2009.
- [20] D. Chen and D. Stow, *The effect of training strategies on supervised classification at different spatial resolutions*, Photogrammetric Engineering and Remote Sensing, 68(11);155-1161, 2002.
- [21] M. Stone, *Cross-validators Choice and Assessment of Statistical Predictions*, Royal Statistical Society, 36:111–147, 1974
- [22] T. Fawcett, *ROC Graphs – Notes and Practical Considerations for Data Mining Researchers*. Technical Report HPL-2003-4, Intelligent Enterprise Technologies Laboratories, HP
- [23] N.M. Adams and D.J. Hand, *Improving the Practice of Classifier Performance Assessment*, Neural Computation, 12(2):305–312, 2000.
- [24] A. Andersson, P. Davidsson, and J. Lindén, *Measure based Classifier Performance Evaluation*. Pattern Recognition Letters, 20(11-13):1165–1173, 1999.
- [25] T.L. Bailey and C. Elkan, *Estimating the Accuracy of Learned Concepts*, In 13th International Joint Conference on Artificial Intelligence, 895–901, Chambéry, France, 1993.
- [26] T.G. Dietterich, *Approximate Statistical Test for Comparing Supervised Classification Learning Algorithms*, Neural Computation, 10(7):1895–1923, 1998.
- [27] F. Provost, T. Fawcett, and R. Kohavi, *The Case Against Accuracy Estimation for Comparing Induction Algorithms*, In 15th International Conference on Machine Learning, 445–453, Madison, WI, USA, 1998. Morgan Kaufmann Laboratories, Palo Alto, 2003.
- [28] L. Devroye, and T. J. Wagner, *Distribution-Free performance Bounds for Potential Function Rules*. IEEE Transaction in Information Theory, 25(5):601–604, 1979.

- [29] D.M.W Powers, *Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation*, Journal of Machine Learning Technologies, 2(1): 37–63, 2011.
- [30] C. Domeniconi, J. Peng and D. Gunopulos, *Locally adaptive metric nearest-neighbor classification*, IEEE Trans Pattern Anal. Mach. Intell. 24(9), 1281–1285, 2002.
- [31] M. Latourrette, *Toward an explanatory similarity measure for nearest-neighbor classification*, In: ECML '00: Proceedings of the 11th European Conference on Machine Learning, London, UK, pp. 238–245. Springer-Verlag, Heidelberg, 2000.
- [32] L. Kozma, *k Nearest Neighbours Algorithm*. Helsinki University of Technology, Available: <http://www.lkozma.net/knn2.pdf>, 2008.
- [33] J.R. Quinlan, *Induction of decision trees*, Mach. Learn. ,1(1):81-106, 1986.
- [34] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone, *Classification and Regression Trees*, Ed. Belmont, CA: Wadsworth International Group, 1984.
- [35] J.R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc, 1993.
- [36] L. Rokach and O. Maimon, *Top-down induction of decision trees classifiers a survey* Systems, Applications and Reviews, IEEE Transactions on, 35(4): 476-487, Nov 2005.
- [37] C. Cortes and V. Vapnik, *Support-vector networks*, Mach. Learn, 20(3):273-297, 1995.
- [38] L. Bottou and C.J. Lin, *Support vector machine solvers*, in Large Scale Kernel Machines, L. Bottou, O. Chapelle, D. De Coste, and J. Weston, Eds. Cambridge, MA. : MIT Press, 301-320, 2007
- [39] L.A. Zadeh, *Fuzzy Sets*, Information and Control , 8:338-352, 1965
- [40] R.R. Yager and L.A. Zadeh, *Fuzzy Sets*, Neural Networks, and Soft Computing. Van Nostrand Reinhold, New York, 1994
- [41] C. Rosental, *Histoire de la logique floue*. Une approche sociologique des pratiques de démonstration, Revue de Synthèse, 4(4): 575-602, 1998.
- [42] L.A. Zadeh, *Similarity relations and fuzzy ordering*, if. Sci, 3:117, 1971.
- [43] L.A. Zadeh, *Outline of a new approach to the analysis of complex systems and decision processes*, IEEE Trans. On systems, Man and Cybernetics, 3(1): P 28-44, 1973.
- [44] T. Takagi and M. Sugeno, *Derivation of fuzzy control rules from human operator's control actions*, Proc. Of the IFAC Symp. On Fuzzy Information, Knowledge Representation and Decision analysis, 555-560, 1983.

- [45] H. Bruler, *Réglage par logique floue*, Presses polytechniques et universitaires romandes, collection électricité, 7-38, 1994.
- [46] M. Gupta, R.K. Ragade and R.R. Yager, *An approach to fuzzy reasoning method*, Editors, Advances in Fuzzy Set Theory and Applications, North Holland, Amsterdam, 1979.
- [47] B. Carse and T.C. Forgarty, *A fuzzy classifier system using the Pittsburgh approach*, Parallel Problem Solving from Nature-PPSN III, 260–269, 1994.
- [48] L. Hu, H.D. Cheng and M. Zhang, *A high performance edge detector based on fuzzy inference rules*, Information Sciences, 177(21):4768–4784, 2007.
- [49] H. Ishibuchi, K. Nozaki and H. Tanaka, *Distributed representation of fuzzy rules and its application to pattern classification*, Fuzzy Sets and Systems, 52(1): 21–32, 1992.
- [50] M. Valenzuela-Redon, *The fuzzy classifier system: motivations and first result*, Parallel Problem Solving from Nature II, 330–334, 1991.
- [51] J. Valente, *Semantic constraints for membership function optimization*, IEEE Transactions on Fuzzy Systems, 19:128–138, 1999.
- [52] H. Hellendoorn and D. Driankov, *Fuzzy Model Identification – Selected Approaches*. Springer-Verlag, 1997.
- [53] J. Yen and R. Langari, *Fuzzy Logic - Intelligence, Control and Information*. Pearson Education, Delhi, 2003.
- [54] M.S. Kim, C.H. Kim and J.J. Lee, *Evolving compact and interpretable Takagi–Sugeno fuzzy models with a new encoding scheme*, IEEE Trans. Syst. Man Cybern. B, Cybern, 36(5):1006–1023, 2006.
- [55] Y. Shi, R. Eberhart and Y. Chen, *Implementation of evolutionary fuzzy systems*, IEEE Trans. Fuzzy Syst, 7(2):109–119, 1999.
- [56] D. Chakraborty and N.R. Pal, *A neuro-fuzzy scheme for simultaneous feature selection and fuzzy rule-based classification*, IEEE Transactions on Neural Networks, 15(1):110–123, 2004.
- [57] C.T. Lin and C.S.G. Lee, *Neural network-based fuzzy logic control and decision system*, IEEE Transactions on Computers, 40(12):1320–1336, 1991.
- [58] D. Nauck and R. Kruse, *A neuro-fuzzy method to learn fuzzy classification rules from data*, Fuzzy Sets and Systems, 89(3):277–288, 1997.
- [59] S. Abe and R. Thawonmas, *A fuzzy classifier with ellipsoidal regions*, IEEE Transactions on Fuzzy Systems, 5(3):358–368, 1997.

- [60] C.Y. Lee, C.J. Lin and H.J. Chen, *A self-constructing fuzzy CMAC model and its applications*, Information Sciences, 177(1):264–280, 2007.
- [61] J.A. Roubos, M. Setnes and J. Abonyi, *Learning fuzzy classification rules from labeled data*, Information Sciences, 150 (1-2):77–93, 2003.
- [62] O. Cordón, *Ten years of genetic fuzzy systems: current framework and new trends*, Fuzzy Sets and Systems, 141(1):5–31, 2004.
- [63] R. Alcal et al., *Genetic learning of accurate and compact fuzzy rule based systems based on the 2-tuples linguistic representation*, International Journal of Approximate Reasoning, 44 (1):45–64, 2007.
- [64] A.F. Gómez-Skarmeta et al., *Approximative fuzzy rules approaches for classification with hybrid-GA techniques*, Information Sciences, 136 (1-4):193–214, 2001.
- [65] Y.C. Hu, *Finding useful fuzzy concepts for pattern classification using genetic algorithm*, Information Sciences, 175 (1-2):1–19, 2005.
- [66] H. Ishibuchi, T. Nakashima and T. Murata, *Three-objective genetics-based machine learning for linguistic rule extraction*, Information Sciences, 136 (1-4) (2001) 109–133.
- [67] L. Sanchez, I. Couso and J.A. Corrales, *Combining GP operators with SA search to evolve fuzzy rule based classifiers*, Information Sciences, 136 (1-4):175–191, 2001.
- [68] C. Setzkorn and R.C. Paton, *On the use of multi-objective evolutionary algorithms for the induction of fuzzy classification rule systems*, Biosystems, 81(2):101–112, 2005.
- [69] E. Zhou and A. Khotanzad, *Fuzzy classifier design using genetic algorithms*, Pattern Recognition, 40(12):3401–3414, 2007.
- [70] A. Fernández, M.J. del Jesus and F. Herrera, *Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets*, International Journal of Approximate Reasoning, 2008.
- [71] A. Gonzblez and R. Perez, *SLAVE: a genetic learning system based on an iterative approach*, IEEE Transactions on Fuzzy Systems, 7(2):176–191, 1999.
- [72] H. Ishibuchi, T. Yamamoto and T. Nakashima, *Hybridization of fuzzy GBML approaches for pattern classification problems*, IEEE Transactions on Systems, Man, and Cybernetics, 35(2): 359–365, 2005.
- [73] H. Ishibuchi and Y. Nojima, *Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning*, International Journal of Approximate Reasoning, 44(1):4–31, 2007.

- [74] E.G. Mansoori, M.J. Zolghadri and S.D. Katebi, *SGERD: a steady-state genetic algorithm for extracting fuzzy classification rules from data*, IEEE Transactions on Fuzzy Systems, 16(4):1061–1071, 2008.
- [75] M.J. del Jesus et al., *Induction of fuzzy-rule-based classifiers with evolutionary boosting algorithms*, IEEE Transactions on Fuzzy Systems, 12(3):296–308, 2004.
- [76] F. Hoffmann et al., *Inferring descriptive and approximate fuzzy rules for credit scoring using evolutionary algorithms*, European Journal of Operational Research, 177(1):540–555, 2007.
- [77] H. Shinn-Ying et al., *Design of accurate classifiers with a compact fuzzy-rule base using an evolutionary scatter partition of feature space*, IEEE Transactions on Systems, Man, and Cybernetics, 34(2):1031–1044, 2004.
- [78] K.A. Jong, W.M. Spears and D.F. Gordon, *Using genetic algorithms for concept learning*, *Machine Learning*, 13(2):161–188, 1993.
- [79] S.F. Smith, *A Learning System Based on Genetic Adaptive Algorithms*, Doctoral Dissertation, Department of Computer Science, University of Pittsburgh, 1980.
- [80] J.H. Holland and J.S. Reitman, *Cognitive systems based on adaptive algorithms*, *Pattern-Directed Inference Systems*, 313–329, 1978.
- [81] T. Kovacs, *Strength or Accuracy: Credit Assignment in Learning Classifier Systems*, Springer-Verlag, New York, 2004.
- [82] H. Ishibuchi, T. Nakashima and T. Murata, *Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems*, IEEE Transactions on Systems, Man, and Cybernetics, 29(5):601–618, 1999.
- [83] M.F. Ganji and M.S. Abadeh, *A fuzzy classification system based on Ant Colony Optimization for diabetes disease diagnosis*, *Expert System with Application*, 38:14650–14659, 2011.
- [84] M.S. Abadeh, J. Habibi and E. Soroush, *Induction of fuzzy classification systems using evolutionary ACO-based algorithms*, in *Proceedings of the First Asia International Conference on Modeling and Simulation (AMS '07)* (IEEE, 2007).
- [85] T. Sousa, A. Silva and A. Neves, *Particle swarm based data mining algorithms for classification tasks*, *Parallel Computing*, 30(5-6):767–783, 2004.
- [86] N. Holden and A. Freitas, *A hybrid particle swarm/ant colony algorithm for the classification of hierarchical biological data*, in *Proceedings of the conference companion on Genetic and evolutionary computation (GECCO '07)*, 2745–2750 (ACM, New York NY, USA, 2007).

- [87] C. Rani and S.N. Deepa, *Design of optimal fuzzy classifier system using Particle Swarm Optimization*, International Conference on Innovative Computing Technologies (ICICT'10), 1-6 (IEEE, 2010).
- [88] A. Khosla, S. Kumar, and K.K. Aggarwal, *A framework for identification of fuzzy models through particle swarm optimization*, in IEEE Indicon Conference, 388–391 (Chennai, India, 2005).
- [89] A. Khosla, S. Kumar and K.R. Ghosh, *A comparison of computational efforts between particle swarm optimization and genetic algorithm for identification of fuzzy models*, in Annual Conference of the North American Fuzzy Information Processing Society, 245–250 (IEEE, 2007).
- [90] A.P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*, chapter 1 Introduction, 1–4. Wiley and Sons , 2005.
- [91] Z. Michalewicz and D.B. Fogel, *How to Solve It: Modern Heuristics* . Second edn. Springer –Verlag, 2004.
- [92] C. Blum and A. Roli, *Metaheuristics in combinatorial optimization: overview and conceptual comparison*, ACM Comput Surv, 35(3):268–308, 2003.
- [93] J.H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*, Ann. Arbor., MI, University of Michigan Press, 1975.
- [94] J. Kennedy and R.C. Eberhart, *Particle Swarm Optimization*, in: Proceedings of the IEEE International Conference on Neural Networks, Piscataway, 5:1942–1948, 1995.
- [95] K. Price and R. Storn, *Differential evolution a simple and efficient adaptive scheme for global optimization over continuous spaces*, Technical Report, International Computer Science Institute, Berkley, 1995.
- [96] L.J. Fogel, A.J. Owens and M.J. Walsh, *Artificial intelligence through a simulation of evolution*, Proceedings of the 2nd Cybernetic Sciences Symposium, 131–155, (Spartan Books), 1965.
- [97] M. Dorigo, V. Maniezzo and A. Colorni, *Positive feedback as a search strategy*, Technical Report 91-016, Dipartimento di Elettronica, Politecnico di Milano, IT, 1991.
- [98] T.A. Zilinskas, *Global Optimization, Lecture Notes in Computer Science*, Springer-Verlag, 1989.
- [99] S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi, *Optimization by simulated annealing*, 220 (4598):671–80, 1983.

- [100] V. Cerny, *Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm*. J Optimiz Theory Appl, 45(1):41–51, 1985.
- [101] R.W. Eglese, *Simulated annealing: a tool for operational research*. Eur J Oper Res, 46(3):271–81, 1990.
- [102] F. Glover, *Tabu search – part 1*, ORSA J Comput, 1(2):190–206, 1989.
- [103] A.P. Engelbrecht, *Computational Intelligence*, 2nd edn. Wiley, New York (2007)
- [104] M. Gen and R. Cheng, *Genetic algorithms and engineering optimization*, engineering design and automation, Wiley-Interscience, 1999.
- [105] G. Beni and J. Wang, *Swarm intelligence in cellular robotics systems*, In Proceeding of NATO Advanced Workshop on Robots and Biological System, 26-30, Tuscany, Italy, 1989.
- [106] G. Beni, *From swarm intelligence to swarm robotics*, Swarm Robotics, 3342:1-9, Springer Berlin Heidelberg, 2005.
- [107] E. Bonabeau, M. Dorigo and G. Theraulaz, *Swarm Intelligence*, From Natural to Artificial Systems, chapitre 1, pages 7-8. Studies in the Sciences of Complexity, Oxford University Press, New York Oxford, 1999
- [108] G.K. Venayagamoorthy and R.G. Harley, *Swarm Intelligence for Transmission System Control*, In: IEEE Power Engineering Society General Meeting, 1–4, 2007.
- [109] J. Kennedy and R.C. Eberhart, *Swarm Intelligence*, Morgan Kaufmann Publisher, San Francisco, 2001.
- [110] H. Bai and B. Zhao, *A Survey on Application of Swarm Intelligence Computation to Electric Power System*, In: Proceedings of the 6th World Congress on Intelligent Control and Automation, 2:7587–7591, 2006.
- [111] E. Bonabeau, M. Dorigo and G. Theraulaz, *Inspiration for Optimization from Social Insect Behavior*, Nature 406, 39–42, 2002.
- [112] E. Bonabeau, A. Sobkowski, G. Theraulaz and J.L. Deneubourg, *Adaptive task allocation inspired by a model of division of labor in social insects*, In: Biocomputing and emergent computation: proceedings of BCEC97, World Scientific Press, pp 36–45, 1997
- [113] M.M. Millonas, *Swarms, phase transitions and collective intelligence*, In: Langton C (ed) Artificial life III, Addison-Wesley, Reading, MA, pp 417–445, 1994.
- [114] T. Sato and M. Hagiwara, *Bee System: Finding Solution by a Concentrated Search*, Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, 4(C), 3954-3959, 1997.

- [115] A. Baykasoglu, L. Ozbakir and P. Tapkan, *Artificial bee colony algorithm and its application to generalized assignment problem*, In Felix T.S. Chan and Manoj Kumar Tiwari, editors, *Swarm Intelligence, Focus on Ant and Particle Swarm Optimization*, 113–144, I Tech Education and Publishing, Vienna, Austria, December 2007.
- [116] D. Karaboga, *An idea based on honey bee swarm for numerical optimization*, Technical Report TR06, Computer Engineering Department, Erciyes University, Turkey, 2005.
- [117] H. YAHYA, *Book "The Miracle of the honeybee"*, G. M. D. Cd., Ed, Okmeydani-Istanbul-Turkey, March 2007.
- [118] C.W. Reynolds, *Flocks, herds and schools: A distributed behavioral model*, *Computer Graphics*. 25-34, 1987.
- [119] A.P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*, John Wiley & Sons Chichester, 2006.
- [120] R.C. Eberhart and J. Kennedy, *A new optimizer using particle swarm theory*, in: *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, EP '98, Piscataway: IEEE, Nagoya, pp. 39–43, Japan, 1995,
- [121] J. Kennedy and R. Mendes, *Population structure and particle swarm performance*, *Proceedings of the World on Congress on Computational Intelligence 2 (2002)* 1671–1676.
- [122] M. Clerc and J. Kennedy, *The particle swarm: explosion stability and convergence in a multi-dimensional complex space*, *IEEE Transactions on Evolutionary Computation* 6(1):58-73, 2002.
- [123] R. Eberhart, P. Simpson and R. Dobbins, *Computational Intelligence PC Tools*, Academic Press Professional, Inc., San Diego, CA, USA, 1996.
- [124] Y. Shi and R. Eberhart, *A modified particle swarm optimizer*, in: *Proceedings of IEEE International Conference on Evolutionary Computation*, IEEE Computer Society, Washington, DC, USA, 69–73, 1998.
- [125] Y. Shi and R.C. Eberhart, *Empirical study of particle swarm optimization*, in: *Proceedings of the 1999 Congress on Evolutionary Computation*, CEC 99, 3:1945–1949, 1999.
- [126] American cancer society. <http://www.cancer.org/Cancer/BreastCancer/index>
- [127] J. Merz and P.M. Murphy, *UCI repository of machine learning databases*, <http://www.ics.uci.edu/learn/MLRepository.html>, 1996.
- [128] C.A. Pena-Reyes and M. Sipper, *A fuzzy-genetic approach to breast cancer diagnosis*, *Artificial Intelligence in Medicine*, 17(2):131-155, 1999.

- [129] D.E. Goodman, L. Boggess and A. Watkins, *Artificial immune system classification of multiple-class problems*, In Proceedings of the artificial neural networks in engineering, pp. 179–183, 2002.
- [130] K. Polat, S. Sahan, H. Kodaz and S. Gunes, *Breast cancer and liver disorders classification using artificial immune recognition system (AIRS) with performance evaluation by fuzzy resource allocation mechanism*, Expert Systems with Applications, 32:172–183, 2007.
- [131] I. Gadaras and L. Mikhailov, *An interpretable fuzzy rule-based classification methodology for medical diagnosis*, Artificial Intelligence in Medicine, 47(1):25-41, 2009.
- [132] I. Maglogiannis, E. Zafiroopoulos and I. Anagnostopoulos, *An intelligent system for automated breast cancer diagnosis and prognosis using SVM based classifiers*, Applied Intelligence, 30:24–36, 2009.
- [133] M. Karabatak and M.C. Ince, *An expert system for detection of breast cancer based on association rules and neural network*, Expert Systems with Applications, 36:3465–3469, 2009.
- [134] W. Yeh, W. Chang and Y.Y. Chung, *A new hybrid approach for data mining breast cancer pattern using Discrete Particle Swarm Optimization and Statistical method*, Expert Systems and Applications, 36(4):8204-8211, 2009.
- [135] M.d. Rafiul Hassan, M. MaruffHossain, R.K. Begg and Y. Morsi, *Breast-Cancer identification using HMM-Fuzzy-approach*, Computers in Biology and Medicine, 40(3):240-251, 2010.
- [136] National Diabetes Information Clearinghouse (NDIC), <http://diabetes.niddk.nih.gov/dm/pubs/d-iagnosis>
- [137] A. Frank and A. Asuncion, *UCI Machine Learning Repository*, <http://archive.ics.uci.edu/ml>, University of California, Irvine, CA : School of Information and Computer Science, (access 06-02-2011), 2010.
- [138] K. Polat, S. Gunes and A. Arslan, *A cascade learning system for classification of diabetes disease: Generalized discriminant Analysis and Least Square Support Vector Machine*, Expert Systems with Applications, 34(1):482-487, 2008.
- [139] P.H. Tand and M.H. Tseng, *Medical Data Mining using BGA and RGA for weighting of features in fuzzy K-NN classification*, in Proceedings of the Eighth International Conference on Machine Learning and Cybernetics, pp. 3070-3075, Baoding, 2009.
- [140] K. Polat and S. Gunes, *An improved approach to medical data sets classification: artificial immune recognition system with fuzzy resource allocation mechanism*, Expert Systems with Applications, 24(4):252-270, 2007.

- [141] K. Polat and S. Günes, *An expert system approach based on principal component analysis and adaptive neuro-fuzzy inference system to diagnosis of diabetes disease*, Digit. Signal Process, 17(4):702–710, 2007.
- [142] H. Temurtas, N. Yumusak and F. Temurtas, *A comparative study on diabetes disease diagnosis using neural networks*, Expert systems with applications, 36(4):8610- 8615, 2009.
- [143] O. Cordón, M.J. Jesus and F. Herrera, *A proposal on reasoning methods in fuzzy rule-based classification systems*, International Journal of Approximate Reasoning, 20:21–45, 1999.
- [144] L.J. Eshelman and J.D. Schaffer, *Real-Coded Genetic Algorithms and Interval-Schemata*, Foundations of Genetic Algorithms, 187-202, 1993.
- [145] S.N. Ghazavi and T.W. Liao, *Medical data mining by fuzzy modeling with selected features*, Artificial Intelligence in Medicine, 43(3):195-206, 2008.
- [146] J. Guzaitis, A. Verikas, A. Gelzinis and M. Bacauskiene, *A framework for designing a fuzzy rule-based classifier*, in Proceedings of the 1st International Conference on Algorithmic Decision Theory (ADT '09) Venice, Italy, eds. F. Rossi and A. Tsoukias, pp. 434-445, Springer-Verlag, Berlin, Heidelberg, 2009.
- [147] T.C. Chen and T.C. Hsu, *A GAs based approach for mining breast cancer pattern*, Expert Systems with Applications, 30(4):674–681, 2006.
- [148] Design of fuzzy classifier for diabetes disease using Modified Artificial Bee Colony algorithm. Computer Methods and Programs in Biomedicine 112(1): 92-103, 2013