

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la Technologie d'Oran
-Mohamed Boudiaf-
USTO-MB
Faculté des sciences
Département d'informatique

Mémoire de MAGISTER
Spécialité : Informatique
Option : Reconnaissance des Formes - Intelligence Artificielle

Présenté et soutenu publiquement par :

Adlania SENOUCI

Titre :
**Les Algorithmes à Base de Fourmis Artificielles
Application à la Classification**

Soutenu le*Octobre 2009* devant le jury composé de :

M ^r Kaddour SADOUNI	Maître de conférence	USTO-MB	Président
M ^r Abdelkader BENYETTOU	Professeur	USTO-MB	Rapporteur
M ^r Bachir DJEBBAR	Maître de conférence	USTO-MB	Examineur
M ^r Khaled BELKADI	Maître de conférence	USTO-MB	Examineur
M ^{elle} Nacéra BENAMRANE	Maître de conférence	USTO-MB	Examinatrice

Laboratoire Signal Image Parole SIMPA



Liste des Figures

1.1	Anatomie d'une fourmi	7
1.2	Deux fourmis communiquent à l'aide des antennes	9
1.3	Attouchement d'antennes	9
1.4	Danse de recrutement	10
1.5	Pont binaire de Deneubourg	16
1.6	Les fourmis suivent indifféremment les deux branches de pont	17
1.7	Les fourmis ont tendance à emprunter le même chemin (celui d'en bas) ..	17
1.8	Expérience du pont avec des branches de tailles différentes	17
1.9	Effet de dépôt d'un obstacle entre la source de nourriture et le nid	18
1.10	Grille utilisée dans [Lumer et Faieta, 1994]	23
1.11	Principes d'acceptation et du rejet entre deux fourmis i et j	29
2.1	AntClass permet la construction de tas d'objets sur la grille	33
2.2	Les différents cas possibles pour le ramassage	35
2.3	Les différents cas possibles pour déposer un objet o_i sur un tas $T_j = \{o_1, o_2, o_3, o_4\}$	36
3.1	Evolution du nombre de classes au cours des itérations de AntClass pour la base Breast cancers	48
4.1	Part de marché par technologie biométrique (Evaluation 2003 par International Biometric Group)	50
4.2	Architecture typique d'un système biométrique mono-modal	53
4.3	Distributions des taux de vraisemblance des utilisateurs légitimes et des imposteurs d'un système biométrique	54
4.4	Composition d'un oeil humain : partie parfaitement annulaire de l'iris dans l'œil	55

4.5	Schéma général d'un système de vérification par reconnaissance d'iris	56
4.6	Image d'un iris capturé (a) dans le visible, (b) dans le proche infrarouge ..	57
4.7	Exemple d'erreur de localisation du centre de la pupille par les opérateurs intégrés différentiels	59
4.8	Repère pseudo polaire sans dimension	62
4.9	Exemple de transformation Cartésienne Polaire	63
4.10	Le principe de codage de phase sur quatre quadrants et en deux bits	63
5.1	Courbes des variations des scores intra-classe et inter-classe pour un Système biométrique	67
5.2	Courbes intra-classe et inter-classe pour les images du tableau 5.1	71
6.1	Principe de l'algorithme de localisation de l'iris	82
6.2	Image 001_1_1 de la base de données CASIA	84
6.3	Détection des contours de l'image 001_1_1 de la base de données CASIA par l'algorithme de Canny	84
6.4	Analyse d'histogramme de l'image 001_1_1 de la base de données CASIA	85
6.5	Détection approximative du centre de la pupille de l'image 001_1_1 de la base de données CASIA par une analyse d'histogramme	85
6.6	Résultat de l'algorithme des fourmis sur l'image 001_1_1 de la base de données	86
6.7	Segmentation de l'iris par l'algorithme des fourmis sur l'image 001_1_1 de la base de données CASIA	87
6.8	Transformation polaire	88
6.9	a : Illustration des résultats de l'application de l'algorithme sur l'image 001_1_3	89
6.9	b : Illustration des résultats de l'application de l'algorithme sur l'image 0093_1_1	90
6.9	c : Illustration des résultats de l'application de l'algorithme sur l'image 018-1-2	90
6.10	a : Taux de détection d'iris pour chaque personne	91
6.10	b : Taux de détection d'iris pour chaque personne	91
6.10	c : Taux de détection d'iris pour chaque personne	91
6.10	d : Taux de détection d'iris pour chaque personne	92

Liste des Tableaux

3.1	Les bases de données réelles utilisées issues de [Blake et Merz, 1998]	45
3.2	Les paramètres de AntClass	46
3.3	Résultats de la classification	47
5.1	Représentation des distances de Hamming entre quelques images de la base CASIAv1	70
5.2	Les taux de TFA et TFR calculés pour chaque seuil	71
5.3	Résultats de l'algorithme AntClust	73
6.1	Taux global de détection de l'iris sur 548 images d'oeil de la base CASIA ..	92

Table des Matières

Remerciement	i
Table des Matière	ii
Liste des figures	vi
Liste des tableaux	vii
Introduction générale	1
1 Les fourmis artificielles	4
1.1 Introduction	4
1.2 Les fourmis réelles	6
1.2.1 Le succès écologique	6
1.2.2 Généralités sur les fourmis	6
1.2.2.1 Description	6
1.2.2.2 L'intelligence collective	9
1.3 Concepts de base	11
1.3.1 L'émergence	12
1.3.2 L'auto-organisation comme technique d'émergence	12
1.3.3 Les mécanismes d'auto-organisation	13
1.3.3.1 La rétroaction	13
1.3.3.2 La gestion des flux	14
1.4 Les algorithmes de fourmis artificielles pour l'optimisation combinatoire	15
1.4.1 Inspiration biologique	15
1.4.2 Les expériences	16
1.4.2.1 Le pont binaire	16
1.4.2.2 Le pont binaire à longueurs inégales	17

1.4.2.3	Le pont avec un obstacle	18
1.4.3	Modèles de fourragement pour la résolution des problèmes d'optimisation.....	18
1.4.4	Autres domaines d'application	20
1.5	Les algorithmes de fourmis artificielles pour la classification automatique	21
1.5.1	Inspiration biologique	21
1.5.2	Modèles de tri de couvain et application à la classification de données	22
1.5.2.1	Les pionniers	22
1.5.2.2	Modèles de Lumer et Faieta	23
1.5.2.3	L'algorithme AntClass	24
1.5.3	Autres domaines d'application	25
1.6	Modélisation de la fermeture coloniale chez les fourmis pour la classification.....	26
1.6.1	Inspiration biologique	26
1.6.2	L'algorithme AntClust	27
1.6.2.1	Principe	27
1.6.2.2	Modèle des fourmis artificielles.....	28
1.7	Autres sources d'inspiration	29
1.8	Conclusion	30
2	L'algorithme AntClass	31
2.1	Introduction	31
2.2	Notations et heuristiques de AntClass	33
2.2.1	Les objets et leurs métriques.....	33
2.2.2	L'algorithme AntClass	34
2.2.2.1	Déplacement des fourmis	34
2.2.2.2	Ramasser un objet	35
2.2.2.3	Déposer un objet	36
2.2.2.4	Mémoire locale des fourmis	37
2.2.2.5	Patience des fourmis	37
2.2.3	Hybridation avec les centres mobiles.....	38

2.3	Motivations	40
2.4	L'algorithme AntClass amélioré	40
2.4.1	Données dont dispose une fourmi dans son environnement	41
2.4.2	Modélisation du comportement des fourmis	41
2.4.2.1	Environnement et déplacement des fourmis	41
2.4.2.2	Ramassage et dépôt des objets	42
2.4.3	Hybridation	44
3	AntClass : Mise en œuvre et résultats	45
3.1	Etude expérimentale	45
3.1.1	Description des données	45
3.1.2	Les paramètres de AntClass	46
3.1.3	Les mesures d'évaluation	46
3.1.4	Résultats et discussion	47
3.2	Conclusion	48
4	La reconnaissance biométrique basée sur l'iris	49
4.1	Introduction à la biométrie.....	49
4.2	Principe de fonctionnement d'un système biométrique	53
4.3	Performance d'un système biométrique	53
4.4	La reconnaissance par l'iris	54
4.4.1	Historique et généralités	54
4.4.2	Schéma général	56
4.4.2.1	Système d'acquisition	57
4.4.2.2	Traitement des données	57
4.4.3	Processus de traitement	58
4.4.3.1	Localisation de l'iris au sein de l'image	58
4.4.3.2	Normalisation	61
4.4.3.3	Extraction de l'information	63
4.4.3.4	Assortiment	64
4.5	Conclusion	64
5	Utilisation des fourmis artificielles dans un système de vérification biométrique par reconnaissance de l'iris	65
5.1	Problématique	65
5.2	Motivations	67

5.3	Etude expérimentale	68
5.3.1	Ensemble des données	68
5.3.2	Etude des distributions des distances intra-classe et inter-classe pour quelques images de la base CASIAv1	69
5.3.3	Mise en œuvre de l'algorithme AntClust	72
5.4	Résultats	72
5.5	Conclusion	73
6	Segmentation de l'iris par les fourmis artificielles	74
6.1	Introduction	74
6.2	Motivations	75
6.3	La métaheuristique « optimisation par colonies de fourmis »	77
6.3.1	Représentation de problème	77
6.3.2	Construction des solutions	78
6.4	Transposition de ACO à la segmentation d'iris	79
6.4.1	Modélisation du problème (environnement et comportement) ..	80
6.5	Données utilisées	81
6.6	La segmentation de l'iris	82
6.6.1	Détection de contours	82
6.6.2	Initialisation	84
6.6.3	Construction de la solution	86
6.6.4	L'optimisation finale	87
6.7	Déroulement de l'iris	88
6.8	Résultats et discussion	89
6.9	Conclusion	92
7	Conclusion et perspectives	93
	Références bibliographiques	97

Introduction Générale

Les méthodes classiques de résolution de problèmes consistent à décomposer le problème en sous problèmes et à définir dès le départ les étapes de résolution. Cette méthode a permis aux informaticiens de résoudre un grand nombre de problèmes. Néanmoins, plusieurs problèmes demeurent : la complexité croissante des problèmes à résoudre, l'apparition de nouveaux besoins reflétant la nécessité de disposer de systèmes robustes et fiables dans des domaines dynamiques et incertains. Face à ces nouveaux problèmes, la tendance des chercheurs informaticiens s'est orientée vers le monde naturel afin de trouver des sources d'inspiration. Il s'agit d'adopter et de modéliser les concepts mis en œuvre par le monde du vivant pour la résolution des problèmes difficiles.

La source d'inspiration que constitue la biologie a de plus en plus de succès dans une branche de l'intelligence artificielle que l'on peut nommer l'informatique bio-inspirée.

L'approche des systèmes bio-inspirés procède de la démarche suivante : certains phénomènes naturels sont capables de mettre en œuvre des heuristiques originales susceptibles de résoudre des problèmes pour lesquels il est difficile de trouver des solutions de manière déterministe par des algorithmes classiques. De plus ces heuristiques sont robustes.

Parmi les approches s'inspirant des systèmes biologiques, on peut citer : les réseaux de neurones [McCulloch, 1943], les algorithmes génétiques [Holland, 1975], le système immunitaire [Jerne, 1974], l'intelligence en essaim [Bonabeau et al., 1999], et les sociétés d'insectes [Camazine, 2002] telles que les termites, les fourmis [Bonabeau, 1997], ou araignées [Bourjot, 1999] qui font effectivement preuve de remarquables

capacités pour effectuer des tâches telles que : la construction de nids complexes, la construction de pont, la recherche efficace de ressources, la capture de proies...

Les systèmes de fourmis artificielles sont largement utilisés ces dernières années pour résoudre des problèmes de classification ou d'optimisation [Dorigo, 1991]. Il s'agit d'une nouvelle approche qui s'intéresse aux comportements individuels des fourmis réelles, aux interactions entre ces entités autonomes et à l'émergence de comportements complexes pouvant être qualifié d'intelligent.

Ce mémoire porte sur l'étude des fourmis artificielles dans le cadre de la classification. En effet, les fourmis résolvent de nombreux problèmes de manière très efficace dans leur environnement, donc il doit être possible de transporter un certain nombre de ces mécanismes pour la résolution de problèmes informatiques notamment la classification. Le travail réalisé dans ce mémoire vise à appliquer trois approches bio-inspirées à base de fourmis artificielles pour résoudre des problèmes de classification.

La première approche modélise la faculté des fourmis à trier collectivement leur couvain ou à construire des cimetières pour résoudre un problème de partitionnement.

Dans la deuxième approche, nous utilisons un algorithme à base de fourmis artificielles inspiré de système de reconnaissance chimique des fourmis dans un système de vérification biométrique par reconnaissance de l'iris lors de processus de matching.

La troisième approche consiste à s'inspirer du comportement de fourrage des fourmis et de leur capacité à s'auto-organiser pour trouver le plus court chemin entre le nid et une source de nourriture pour résoudre un problème biométrique. Plus précisément, il s'agit d'une résolution collective du problème de segmentation d'iris.

Pour atteindre ces trois objectifs, ce mémoire se compose de six chapitres : Le chapitre 1 est constitué de deux parties dont la première décrit les caractéristiques des fourmis réelles qui peuvent être exploitées pour la résolution des problèmes d'ingénierie. La seconde partie est dédiée aux éléments d'état de l'art. Nous présentons une tentative d'algorithmes de fourmis artificielles que nous avons appliquée dans ce mémoire. Les chapitre 2 et 3 présentent respectivement les travaux que nous avons menés sur la classification non supervisée en s'inspirant des fourmis et des travaux existant ainsi que l'expérimentation de nos travaux dans un problème de type classification non supervisée. Le chapitre 4 présente les principes de la reconnaissance biométrique basée sur l'iris. Après avoir introduit les concepts principaux et les notions relatifs à ce domaine, nous pouvons passer aux chapitres 5 et 6 dans lesquels nous présentons un autre type d'utilisation des modèles de fourmis artificielles dans ce domaine. Dans le chapitre 5, nous appliquons un algorithme de classification inspiré de système de reconnaissance chimique des fourmis dans un système de vérification biométrique par reconnaissance de l'iris lors de processus de matching, afin de fournir une authentification précise des individus. Quand au chapitre 6, nous montrons comment certains comportements collectifs des fourmis réelles peuvent être exploités pour la résolution collective du problème de segmentation d'iris. A la fin de ce document, nous donnons une conclusion sur l'ensemble de ces travaux ainsi que les perspectives d'avenir.

Chapitre 1

Les fourmis artificielles

L'étude des comportements des fourmis donne un champ d'inspiration très important pour les chercheurs en informatique dans plusieurs domaines et a permis d'introduire de nouvelles heuristiques pour résoudre différents problèmes. En effet, ces insectes offrent un modèle naturel d'un système collectif auto-organisé, décentralisé, fondé sur des individus autonomes au comportement relativement simple et probabiliste qui sont distribués dans l'environnement et ne disposent que d'informations locales. Dans ce chapitre, nous allons décrire les modèles de fourmis artificielles se rapportant à l'optimisation et à la classification ainsi que les principaux algorithmes qui leur sont associés et leurs domaines d'application. Mais tout d'abord nous allons commencer par un aperçu des principales caractéristiques des fourmis réelles et de quelques uns de leurs comportements qui peuvent être exploités pour la résolution des problèmes.

1.1 Introduction

L'étude des fourmis et des autres insectes sociaux (les abeilles, les termites, les guêpes, les araignées...) occupe une place centrale en éthologie et donne un champ d'inspiration très important pour les chercheurs en informatique dans plusieurs domaines. Cela se justifie particulièrement quand on connaît la richesse comportementale de ces animaux. L'analogie que nous croyons relever entre leurs activités et les nôtres nous fait interpréter de manière anthropomorphique le fonctionnement de leurs sociétés. En effet, ces insectes chassent "comme nous", se

défontent "comme nous", élèvent d'autres espèces animales "comme nous". C'est ainsi que nous parlons de la société des fourmis comme s'il s'agissait d'une société humaine hiérarchisée, où un individu, qui aurait cherché un maximum d'informations, distribuerait ses ordres à ses congénères. Cette analogie exprime l'idée répandue, mais fautive, que la complexité des réalisations d'une société ne peut trouver son origine que dans la complexité des individus qui la composent. La réalité est tout autre. Les sociétés d'insectes nous proposent un modèle de fonctionnement bien différent du modèle humain : un modèle décentralisé, fondé sur la coopération d'unités autonomes au comportement relativement simple et probabiliste, qui sont distribuées dans l'environnement et ne disposent que d'informations locales (elles ne disposent d'aucune représentation ou connaissance explicite de la structure globale qu'elles ont à produire ou dans laquelle elles évoluent, bref, elles n'ont pas de plan) [Bonabeau et Theraulaz, 1994].

Plus particulièrement, les fourmis ont fournis un des premiers modèles ayant obtenu un succès notable. Selon Hölldobler et Wilson dans [Hölldobler et Wilson, 1990], les fourmis apparurent il y a une centaine de millions d'années au milieu des dinosaures, pour se répandre ensuite rapidement sur la planète. Elles se sont ensuite multipliées en une dizaine de milliers d'espèces différentes. Leurs capacités à travers les époques montrent l'efficacité induite par leurs comportements sociaux. Par les comportements simples de chacune des fourmis, des interactions limitées à travers une coopération inconsciente, émergent des comportements collectifs intelligents et des modèles d'auto-organisation [Bonabeau, 2000]. Ces modèles de comportements collectifs ont été repris par les informaticiens pour construire de nouvelles méthodes de résolution de problèmes complexes d'où l'essor d'une nouvelle classe d'algorithmes apparue sous le nom « *Algorithmes de fourmis artificielles* ».

A la place du domaine des fourmis artificielles, il est plus opportun de parler d'intelligence artificielle collective ou d'intelligence en essaim (*Swarm Intelligence*) [Bonabeau et Theraulaz, 1994 ; Bonabeau et al, 1999].

1.2 Les fourmis réelles

1.2.1 Le succès écologique

La place des fourmis dans l'étude des sociétés animales est centrale car elles ont développé des formes très avancées de socialité allant jusqu'à partager leur activité de reproduction en confiant la transmission de leurs gènes à quelques individus de la colonie (les reines et les mâles).

Le nombre d'espèces sociales (environ 13 500 connues [Hölldobler et Wilson, 1996]) est assez réduit par rapport au nombre d'espèces d'insectes répertoriées, soit environ 750 000, alors que les insectes sociaux représentent la moitié de la biomasse des insectes. La grande diversité des fourmis (environ 10 000 espèces connues [Hölldobler et Wilson, 1996]) propose une large variété de morphologies et de comportements. L'étude des fourmis, la *myrmécologie*, est donc un vaste et passionnant champ d'investigation.

1.2.2 Généralités sur les fourmis

1.2.2.1 Description

Les fourmis appartiennent à la famille des Formicidae, elles sont classées dans l'ordre des hyménoptères (il s'agit des insectes dont les deux paires d'ailes sont membraneuses et fines), la fourmi peut mesurer de quelques millimètres à plusieurs centimètres de long, son corps se compose de trois parties principales : la tête, le thorax et l'abdomen (figure 1.1). La tête est ornée d'une paire d'antennes qui abritent plusieurs organes de sens : le toucher, l'odora et le goût. On trouve également à l'extrémité de la tête, la bouche qui sert à la fois à l'alimentation et très souvent d'outil de saisie. Les yeux au nombre de deux, sont composés et formés d'une multitude de facettes. Quelques fourmis sont complètement aveugles sans aucun organe visuel. Le thorax, ou corselet, porte trois paires de pattes très longues et terminées par deux griffes. Les organes tactiles sont bien développés et sont constitués de poils répartis sur tout le corps. Leur densité est plus forte sur les antennes et aux extrémités des pattes. L'abdomen est relié au thorax par un

pédoncule, il renferme le jabot social, l'estomac, l'intestin et les conduits respiratoires et chez les reines et les mâles l'appareil reproducteur. La fourmi possède un estomac un peu spécial, le "jabot social", elle y garde en réserve de la nourriture qui peut être régurgitée et transmise aux autres fourmis. La carapace (ou exosquelette) est composée d'une substance, résistante et imperméable que l'on nomme chitine. Elle renferme différents pigments bruns, rouges, et noirs qui sont responsable de la couleur de l'animal.

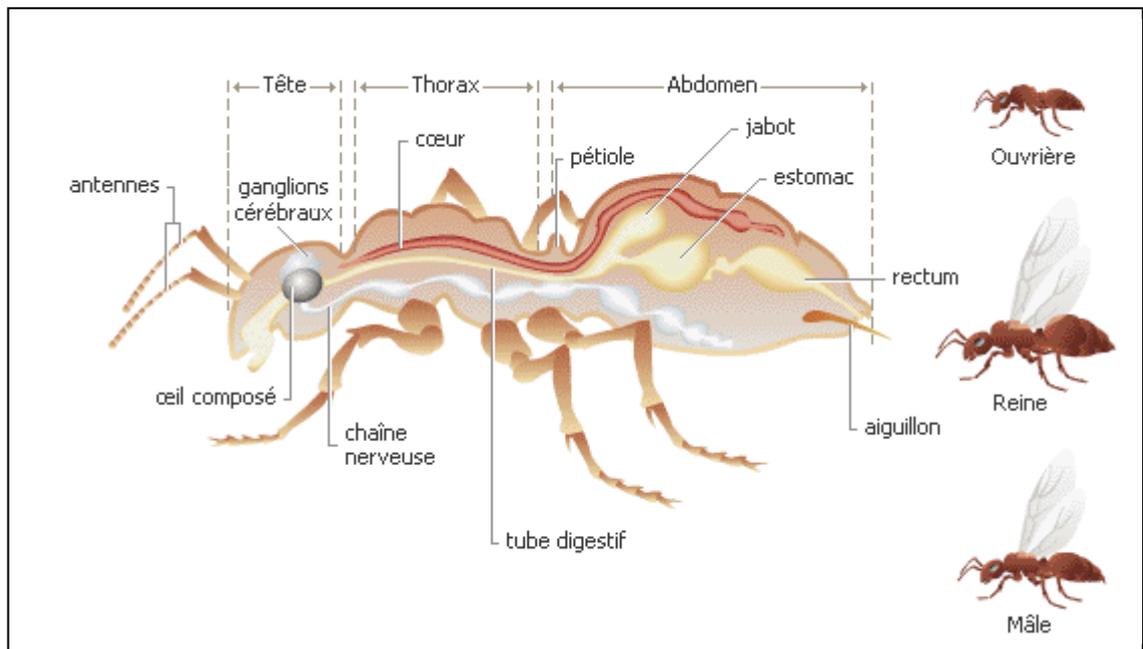


FIG 1.1. Anatomie d'une fourmi

Selon les espèces, les fourmis vivent dans le sol, sous un caillou, dans le bois mort ou dans les arbres. Certaines fourmis arboricoles construisent leurs nids dans les arbres, en cousant les feuilles sur elles, ou en élaborant une architecture faite de grains de sables, de salive et de déjections. D'autres, comme les célèbres fourmis " légionnaires " d'Afrique, n'ont pas de nid et se déplacent par centaines de milliers d'individus sur des vastes territoires, à la manière des nomades.

La fourmi est un insecte exclusivement social vivant en société dans des habitations collectives : la fourmilière, elle est constituée par un ensemble de cellules réunies entre elles par un réseau complexe de galerie qui peut être très important. Dans ces cellules, les fourmis déposent les œufs, les larves et les nymphes. La fourmilière et ses environs constituent le centre de la vie communautaire. Les

activités des communautés des fourmis sont caractérisées par un certain degré de division de travail souligné par une différenciation fonctionnelle et anatomique des individus. Une fourmilière peut abriter de 50 000 à plus de 1 000 000 individus bien différenciés tant au niveau physique qu'au niveau des comportements et des tâches à accomplir. On les sélectionne en caste :

- Les reines : dans une fourmilière, on trouve une ou plusieurs reines. Les reines sont nettement plus grosses que les autres fourmis et peuvent vivre jusqu'à dix ou quinze ans. Leur rôle se résume essentiellement à pondre des œufs et sont donc les fondatrices de nouvelles colonies...
- Les ouvrières : elles forment la majorité des habitants de la cité et se chargent de la défense et de l'entretien de la colonie, qui comprend la construction des galeries, les soins apportés aux jeunes, la quête de la nourriture, etc.
- Les soldats : ils sont plus massifs que les ouvrières, et possèdent souvent de grosses mandibules. Leur rôle est de défendre la fourmilière, et de transporter des charges lourdes. Mais certains comme, chez les fourmis "*Grand Galop*" *camponotus maculatus*, participent aux soins des larves, et donnent à manger aux fourmis qui le demandent. Chez cette espèce qui est la plus grosse fourmi vivant à la Réunion, on peut distinguer des formes intermédiaires entre la petite ouvrière grêle et le puissant soldat.
- Les jeunes sexués : ce sont des fourmis femelles et mâles. Ils sont facilement reconnaissables par leur plus grande taille, par la présence de deux paires d'ailes membraneuses sur le thorax et par trois ocelles disposés en triangle sur le dessus de la tête. Les femelles sont plus grosses que les mâles. Ce sont les futurs rois et reines qui iront fonder des nouvelles colonies.
- Le couvain : il est constitué par les œufs, les larves et les nymphes. Au bout de quelques jours les œufs donnent naissance à des larves qui bien nourries par les ouvrières pendant quinze jours à trois semaines, se transforment en nymphes. Pendant la nymphose, la nymphe ne se nourrit plus, son corps tout entier subit de profondes mutations internes et externes qui vont faire d'elle petit à petit, une fourmi.

1.2.2.2 L'intelligence collective

- Communication : les insectes sociaux en général, et les fourmis en particulier, ont développé des mécanismes de communications très élaborés [Vander Meer et al., 1998]. La communication entre fourmis peut être de différentes natures et varie d'une espèce à une autre. Les principales formes de communication sont :

La communication sonore : les fourmis émettent des sons au moyen des organes stridulatoires pour remplir différentes fonctions selon l'espèce et les circonstances pour attirer les autres fourmis, et parfois elles émettent des sons sans organes stridulatoires, par exemple en tapant la tête contre le mur ou les parois de leur fourmilière avec leur abdomen pour prévenir les autres.

La communication tactile : chez la plupart des insectes, certains messages, simples et directs, sont transmis par rapport physique, comme par attouchement, tapotement ou bien, effleurement. Ce type de communication est réalisé par l'intermédiaire des antennes et parfois aussi par l'intermédiaire des pattes.



FIG 1.2. Deux fourmis communiquent à l'aide des antennes.



FIG 1.3. Attouchement d'antennes.

La communication visuelle : ce type de communication qui était une des formes couramment utilisées au stade primaire de l'évolution, est de moins en moins utilisé. Cependant, certaines espèces s'en servent encore pour des situations bien particulières. Les fourmis entament une danse pour recruter souvent combinée avec de la communication tactile. Certaines fourmis retrouvent leur chemin d'une façon visuelle.



FIG 1.4. Danse de recrutement.

La communication chimique : c'est la forme de communication la plus sophistiquée. Elle consiste à émettre par diverses glandes des substances chimiques volatiles et odorantes, appelées les phéromones, ces phéromones sont captées par les antennes et servent à l'attraction, le recrutement, et l'alerte des autres fourmis, la reconnaissance des différents castes et stades de développement auxquels appartiennent les fourmis rencontrées, la discrimination entre fourmi congénère et étrangère.

Ainsi la fourmi utilise de nombreux types de communication, comme la communication sonore, tactile et visuelle ; cependant elles restent de point de vue de l'efficacité très négligeables comparées à la communication chimique.

- La division des tâches: les fourmis ont la faculté de répartir dynamiquement les tâches en fonction des besoins de la colonie, de manière distribuée, sans aucun contrôle central. Le signe le plus ostensible d'une répartition effective du travail au sein de la colonie, est l'existence des castes qui peuvent être de deux types : morphologiques et comportementales. Ces derniers sont multiples : nourrir la reine, nourrir les larves, fourrager, chasser des proies, attaquer des colonies ennemies.

Un individu peut changer de castes au cours de sa vie, mais il devient aussi de plus en plus spécialisé au fil du temps.

- La construction du nid : l'architecture des nids construits par les fourmis est un exemple frappant de structure complexe. L'intérêt pour des modèles pouvant expliquer l'apparition de telles structures provient encore une fois de l'organisation distribuée qui est sous-jacente. Il n'y a pas, a priori, de contrôle centralisé, de coordination de niveau supérieur à l'individu. La structure émerge des interactions inter-individuelles et avec l'environnement.
- La quête de la nourriture : une colonie doit s'alimenter pour survivre. Pour cela, les fourmis recherchent des sources de nourriture dans le voisinage du nid. Dans une première phase, elles errent de manière plus ou moins aléatoire. Lorsqu'une fourmi découvre une source de nourriture, elle retourne au nid en déposant des phéromones le long du chemin parcouru. En chemin, la fourmi qui a découvert la source de nourriture recrute d'autres individus grâce aux phéromones déposées, mais aussi en réalisant des trophallaxies qui leur permettent d'évaluer la qualité de la source afin de choisir de suivre la piste ou non.

1.3 Concepts de base

Après cet aperçu biologique des fourmis, nous allons présenter par la suite de ce chapitre une tentative de synthèse des différents domaines auxquels nous allons toucher dans les chapitres suivants. C'est en quelque sorte l'état de l'art du domaine

des fourmis artificielles. Mais il convient tout d'abord de présenter quelques concepts de bases qui seront utilisés tout au long de ce mémoire :

1.3.1 L'émergence

Bien que l'émergence soit encore aujourd'hui l'une des notions les plus floues et les plus discutées, elle est actuellement la plus utilisée pour la conception des systèmes artificiels. Son origine viendrait d'après Ali et Zimmer [Ali, 1997], du postulat datant de l'antiquité Grèce : "*le tout est plus que la somme de ses parties*". La notion d'émergence peut être définie comme une propriété macroscopique d'un système qui ne peut pas être déduite à partir de son fonctionnement microscopique [Holland, 1999]. On parle d'émergence quand il y a apparition de structures, et de comportements plus complexes que ceux des entités qui forment le système. Ces comportements sont non programmés explicitement et donc non prévisibles. Cette propriété initialement dans les domaines de la biologie, de la thermodynamique est reprise depuis quelques années dans le domaine de l'informatique et a été exploitée pour la conception des systèmes artificiels.

1.3.2 L'auto-organisation comme technique d'émergence

Depuis son apparition dans les domaines de biologie, de chimie et de physique, l'utilisation du terme « auto-organisation » s'est largement répandue ces dernières années pour la conception des systèmes informatiques artificiels. Plusieurs définitions du concept d'auto-organisation existent dans la littérature. Nous pouvons en citer les suivantes :

Définition 1: « Un système auto-organisateur est un système qui change sa structure de base en fonction de son expérience et de son environnement. » [Ünsal, 1993].

Définition 2: « L'auto-organisation est définie comme un processus dans lequel un modèle de niveau global émerge uniquement d'un grand nombre d'interactions entre les comportements de bas niveau du système. De plus les règles spécifiant les

interactions entre composantes du système sont suivies en utilisant uniquement des informations locales sans références au modèle global » [Camazine, 2000].

Définition 3: « Tout processus au cours duquel des structures émergent au niveau collectif (ou plus généralement apparition d'une structure à l'échelle $N + 1$ à partir d'une dynamique définie à l'échelle N), à partir de la multitude des interactions entre individus, sans être codées explicitement au niveau individuel.» [Théraulaz, 1997].

Toutes ses définitions font référence les mêmes concepts : structuration, organisation, interaction, autonomie et enfin émergence d'un comportement global à partir de plusieurs comportements locaux [Georgé, 2004]. L'auto-organisation peut être alors définie comme un moyen permettant à un système de se structurer et de se maintenir sans aucune intervention de l'extérieur. Chaque composant du système réagit aux stimulus par des règles locales simples et modifie ainsi son environnement et donc le comportement des autres composants (par exemple le dépôt de traces de phéromones chez les fourmis). De ce processus émerge une intelligence collective qui permet au système de réaliser des tâches difficiles voire complexes non explicites. On voit là que le concept d'émergence est fortement lié à celui de l'auto-organisation.

1.3.3 Les mécanismes d'auto-organisation

Devant la complexité des structures obtenues par le phénomène d'auto-organisation, on se demande comment ces structures émergent t-elles sans un plan prédéfini. En fait leur apparition est due à de nombreuses interactions entre les éléments du système ainsi qu'avec l'environnement externe. Nous pouvons distinguer en particulier les principaux facteurs suivants :

1.6.2.1 La rétroaction

Pendant son exécution, les composants du système interagissent ensemble ou avec leur environnement. On considère une forme particulière d'interaction est la rétroaction (ou feed-back) qui est obtenue lorsque les résultats (obtenus par

transformation des données d'entrée) sont retransmis au système sous la forme de nouvelles données d'entrée. Deux types de rétroaction peuvent alors être observés :

- *La rétroaction positive :*

Appelé aussi « auto-catalyse », c'est un mécanisme qui permet de renforcer une action impliquant une convergence très rapide. Il est observé quand les résultats agissent de la même manière que les données d'entrée c'est à dire renforcent le processus de transformation. Ce phénomène est observé chez les fourmis lors de la recherche de nourriture. Le cumul de traces de phéromones déposées par chaque fourmi sur le chemin qu'elle emprunte, permet de renforcer ultérieurement le choix de ce chemin par les autres fourmis de la même colonie. [Deneubourg, 1987].

- *La rétroaction négative :*

C'est un mécanisme de maintien d'équilibre qui va agir dans le sens inverse de l'amplification des fluctuations du système. Une rétroaction positive peut conduire à une stagnation prématurée et à une explosion du système alors qu'une rétroaction négative le stabilise. Ce phénomène est aussi observé dans le comportement de fourrageage des fourmis avec le mécanisme d'évaporation des traces de phéromones. Les chemins les moins choisis par les fourmis seront de moins en moins imprégnés de phéromones [Deneubourg, 1987].

1.6.2.2 La gestion des flux

Ce sont des moyens de communication entre les composants du système et avec leur environnement. La communication peut être directe par messages ou signaux ou bien indirecte par le biais de modifications de l'environnement. Cette deuxième possibilité de communication a été appelée par le chercheur Pierre-Paul Grassé « stigmergie » [Grassé, 1959] à partir des racines stigma qui signifie piqûre, et ergon. qui veut dire travail ou oeuvre.

Dans un système auto-organisé, la stigmergie utilise l'environnement comme une sorte de mémoire collective à travers lequel les composants du système interagissent.

L'environnement est aussi utilisé comme support d'inscription des effets de leurs actions qui ont pour conséquence la modification de l'environnement. Chaque composant effectue une action individuelle en fonction de sa perception locale de l'environnement. Cette action va engendrer des modifications qui vont modifier cet environnement d'une manière pertinente et donc influencer son futur comportement ainsi que celui de tous les autres composants du système. L'auto-organisation ainsi obtenue sans aucune supervision des différentes actions des composants est alors observable de l'extérieur du système sans être au préalable programmée ni explicitée.

1.4 Les algorithmes de fourmis artificielles pour l'optimisation combinatoire

Les fourmis résolvent de nombreux problèmes liés à leur survie. De là à estimer que dans certains cas elles résolvent un problème d'optimisation, Dorigo et ses collègues ont transposé cette capacité naturelle pour résoudre des problèmes d'optimisation combinatoire à travers de la métaheuristique *Ant Colony Optimization* (ACO) [Colomi et al., 1991; Dorigo et al., 1991; Dorigo, 1992]. Cette métaheuristique qui s'inspire du comportement de fourrageage observé chez les fourmis réelles a été appliquée à la résolution du problème du voyageur de commerce (PVC) ou Traveling Salesman Problem. Ce dernier, consiste à trouver un chemin Hamiltonien dans un graphe complètement connecté. Il s'agit pour un voyageur de commerce de trouver le chemin le plus court pour visiter une et une seule fois chacune des n villes dans lesquelles il doit se rendre. L'espace de recherche est l'ensemble des combinaisons possibles des n villes. Il s'agit sans doute du problème d'optimisation combinatoire NP-complet le plus utilisé comme test pour les nouvelles méthodes d'optimisation.

1.4.1 Inspiration biologique

Les études éthologistes ont montré que les fourmis utilisent la stigmergie pour mener à bien la tâche de recherche de nourriture appelée aussi fourrageage. Ainsi lorsqu'une fourmi se déplace de son nid à la recherche de nourriture, elle dépose une substance chimique sur le sol en proportion avec le potentiel de la source de nourriture appelée « phéromone ». Cette substance attire les autres fourmis de la

colonie. Ainsi, lorsque plusieurs possibilités de chemin s'offrent à un élément de la colonie, ce dernier choisira avec une plus grande probabilité les directions marquées par de plus fortes concentrations de phéromone. Ce procédé basé sur le mécanisme de rétroaction positive, assure que pendant le fourragement pour la nourriture, les fourmis utilisent la voie d'accès la plus courte car elle sera la plus imprégnée par la phéromone.

1.4.2 Les expériences

L'idée originale provient des travaux de Deneubourg et ses collègues [Goss , 1989; Deuneubourg, 1990], dans lesquels ils ont mené des expériences différentes et ont permis l'élaboration d'un modèle mathématique comportemental.

1.4.2.1 Le pont binaire

Dans cette expérience, Deneubourg [Deneubourg, 1990] a séparé le nid et la source de nourriture par un pont binaire constitué de deux branches identiques (figure 1.5.), Comme au départ, il n'y a aucune trace de phéromone sur les deux branches, chacune peut être choisie avec la même probabilité (figure 1.6.). Après une phase de fluctuations, la quantité de phéromone a tendance à s'accumuler sur un chemin et il a été observé alors que les fourmis prenaient collectivement le même itinéraire après quelques minutes (figure 1.7.).

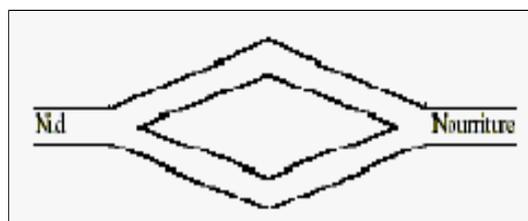


FIG 1.5. Pont binaire de Deneubourg



FIG 1.6. Les fourmis suivent indifféremment les deux branches de pont

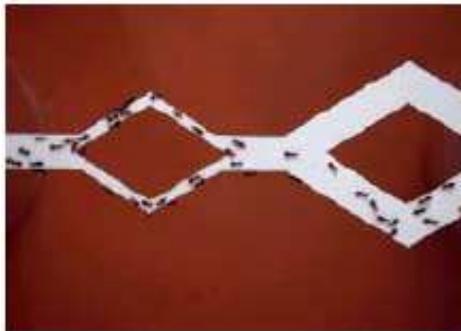


FIG 1.7. Les fourmis ont tendance à emprunter le même chemin (celui d'en bas)

1.4.2.2 Le pont binaire à longueurs inégales

Dans une deuxième expérience, les chercheurs ont étudié l'effet de l'utilisation d'un pont avec des longueurs inégales (figure 1.8.). En raison du même mécanisme de dépôt de phéromone, les fourmis qui ont pris la branche la plus courte arrivent à la source de nourriture en premier, et retournent (le long du même chemin) en premier au nid augmentant de ce fait les quantités de phéromone sur ce plus court chemin, renforçant ainsi le choix des autres fourmis.

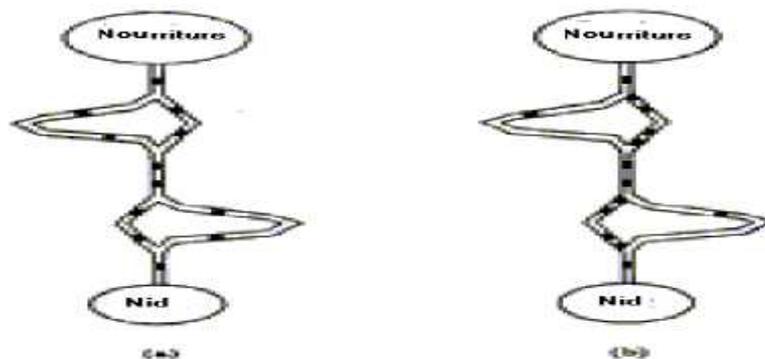


FIG 1.8. Expérience du pont avec des branches de tailles différentes

1.4.2.3 Le pont avec un obstacle

Dans une troisième expérience, alors que les fourmis suivent leur chemin (figure 1.9.a), un obstacle est placé afin de leur barrer la route entre le nid et la source de nourriture (figure 1.9.b). Les fourmis doivent alors contourner cet obstacle par la droite ou par la gauche afin de poursuivre leur chemin. Comme initialement, il n'y a aucune trace de phéromone le long de l'obstacle, la moitié des fourmis passent par le côté gauche et l'autre moitié par le côté droit. Après un certain temps, les chercheurs ont observé que les fourmis qui ont choisi le chemin le plus court, parviennent à reconstituer plus rapidement la concentration en phéromones sur ce nouveau chemin que celles qui ont choisi le chemin le plus long. Ainsi la concentration de phéromone sur le plus court chemin va augmenter, incitant ainsi d'autres fourmis à choisir le chemin riche en phéromones, alors, les fourmis vont finalement choisir le chemin le plus court (figure 1.9.c).

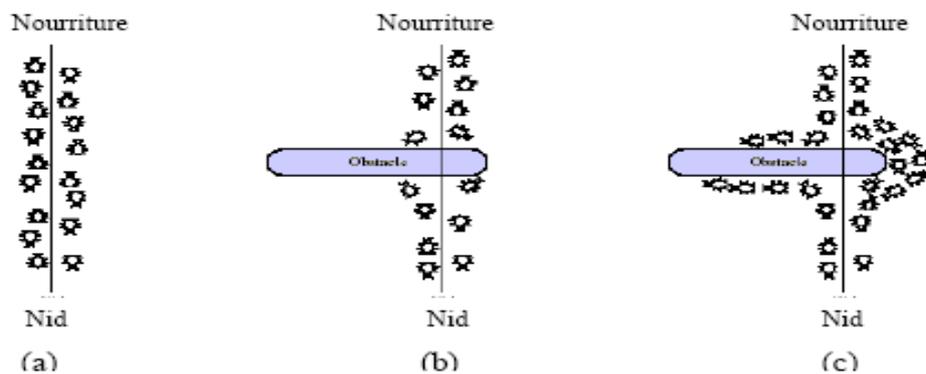


FIG 1.9. Effet de dépôt d'un obstacle entre la source de nourriture et le nid

1.4.3 Modèles de fourrage pour la résolution des problèmes d'optimisation

Le premier système ACO introduit par Dorigo est appelé Ant System (AS) [Dorigo et al, 1996]. Cet algorithme qui est à la base des travaux de Deneubourg, a été appliqué initialement à la résolution du problème (PVC). Voici la modélisation du problème qui est proposée : Le PVC est modélisé par un graphe $G(V,E)$ où E est l'ensemble des noeuds représentant les villes à visiter, et V est l'ensemble des arêtes.

Initialement, les m fourmis sont placées aléatoirement sur les noeuds du graphe. Ensuite chacune des fourmis se déplace d'un noeud à un autre en parcourant les arêtes du graphe. Ce déplacement dépend de la liste des villes déjà visitées représentant la mémoire de la fourmi et d'une probabilité établissant ce choix. L'algorithme AS est constitué d'un nombre d'itération appelée « cycle ». A chaque cycle, chaque fourmi k ($k = 1, \dots, m$) placée sur la ville i à l'instant t choisira la ville j en fonction de la visibilité (heuristique) η_{ij} et des phéromones $\tau_{ij}(t)$. Le choix du passage d'un noeud i à un noeud j se fait aléatoirement en fonction d'une probabilité donnée par :

$$p_{ij}^k(t) = \begin{cases} \frac{(\tau_{ij}^k)^\alpha \cdot (\eta_{ij}^k)^\beta}{\sum_{l \in J_i^k} (\tau_{il}^k)^\alpha \cdot (\eta_{il}^k)^\beta} & \text{si } j \notin J_i^k \\ 0 & \text{sinon} \end{cases} \quad (1.1)$$

J_i^k est l'ensemble des villes non visitées par la fourmi k placée sur le noeud i . α et β sont des constantes paramétrables pour régler l'importance relative de l'intensité et de la visibilité d'une arête. A la fin de chaque cycle les variables "phéromones" sont mises à jour selon :

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \sum_{k=1}^m \Delta \tau_{ij}^k(t) \quad (1.2)$$

Où $\rho \in [0, 1[$ est un coefficient d'évaporation des traces de phéromones, $\Delta \tau_{ij}(t)$ est la somme des phéromones que chaque fourmi dépose sur l'arc (i, j) entre l'itération t et $t + 1$. Cette valeur est donnée par :

$$\Delta \tau_{ij}^k(t) = \begin{cases} Q/L_k(t) & \text{si } (i, j) \in T_k(t) \\ 0 & \text{sinon} \end{cases} \quad (1.3)$$

Où $T_k(t)$ est le trajet effectué par la fourmi k à l'itération t , Q est un paramètre constant et $L_k(t)$ est la longueur du trajet.

Le procédé est alors recommencé jusqu'à ce l'on obtienne une solution optimale ou jugée acceptable. L'encadré 3.1. décrit le comportement de l'algorithme AS pour le PVC.

Algorithme 1.1 : L'Algorithme AS-TSP

- (1) **Pour** $t = 1, \dots, t_{\max}$
 - (2) **Pour** chaque fourmi $k = 1, \dots, m$
 - (3) Choisir une ville au hasard
 - (4) **Pour** chaque ville non visitée i
 - (5) Choisir une ville j , dans la liste des villes restantes J_i^k , selon la formule 1.1
 - (6) **Fin Pour**
 - (7) Déposer une piste $\Delta\tau_{ij}(t)$ sur le trajet $T_k(t)$ selon l'équation 1.2
 - (8) **Fin Pour**
 - (9) Evaporer les pistes selon la formule 1.3
-

Plusieurs extensions à l'algorithme AS ont vu le jour afin d'améliorer la répartition des phéromones sur le graphe ainsi que d'ajuster plus finement la probabilité de choix d'une ville par les fourmis. Nous ne détaillerons pas ces mises à jour dans ce document mais nous pouvons cependant citer les travaux suivants :

[Gambardella et Dorigo, 1995] introduit un système d'apprentissage par renforcement, [Stützle et Hoos, 1997] propose le système *MAX-MIN AS* introduisant des seuils minimum et maximum à la trace de phéromone et [Dorigo et Gambardella, 1997] décrit l'algorithme Ant Colony System (ACS). Ce dernier introduit principalement plus de choix probabiliste dans la décision d'exploration d'une ville et correspond à l'heuristique à base de fourmis artificielles apportant les meilleurs résultats pour le PVC.

1.4.4 Autres domaines d'application

Les bonnes performances des algorithmes basés sur les fourmis obtenues lors de leur application au problème du voyageur du commerce ont incité beaucoup de chercheurs à les utiliser dans d'autres domaines d'application. Sans vouloir dresser une liste exhaustive de toutes les applications et variantes qui ont été produites, on peut citer le problème d'affectation quadratique [Taillard, 1997; Maniezzo, 1994],

Dans [Abbattista et al., 1995; Abbattista et Dalbis, 1996], AS est utilisé pour accélérer le processus de recherche d'un l'Algorithme Génétique (AG). Les tests sont effectués sur le PVC à titre de comparaison. Il s'agit de combiner l'aspect coopératif de AS avec l'aspect évolutif de l'AG, le problème de coloration de graphe[Costa, 1997; Dudot, 2005], le problème de routage [Bullnheimer et al, 1997], les réseaux de communication [Di Caro, 1998], le problème d'ordonnancement [Gambardella, 1997; Merkloe, 2002], les problèmes de satisfaction de contraintes [Solnon, 2000], la fouille de données[Parpinelli, 2002], l'optimisation de site d'enseignement en ligne (notion d'«E-learning»)[Semet, 2003]. Une bonne synthèse de ces algorithmes est reportée dans [Dorigo, 2004].

1.5 Les algorithmes de fourmis artificielles pour la classification automatique

Les algorithmes de classification automatique sont un autre type d'algorithmes inspirés de comportements collectifs observés chez les fourmis : le tri collectif de couvains ou la constitution de cimetières. Ces algorithmes dérivent des travaux de biologiste Deneubourg et son équipe [Deneubourg, 1990] lors des expériences de simulation.

1.5.1 Inspiration biologique

Les travaux de Deneubourg ont montré que certaines espèces de fourmis trient spontanément les œufs de leur couvain par stade d'évolution : elles font de petits tas d'œufs, de larves et de nymphes, elles sont même capables de distinguer plusieurs stades d'évolution pour les œufs. Le principe de base de ce comportement utilise les règles suivantes :

- lorsqu'une fourmi rencontre un élément du couvain, elle le ramasse avec une probabilité d'autant plus grande que cet élément est isolé.
- Lorsqu'une fourmi transporte un élément, elle le dépose avec une probabilité d'autant plus grande que la densité des éléments de même type dans le voisinage est grande.

Là aussi le tri du couvain est le résultat du processus de rétroaction positive et l'utilisation d'informations locales à chaque fourmi.

1.5.2 Modèles du tri du couvain et application à la classification de données

1.5.2.1 Les pionniers

Deneubourg apparaît comme un pionnier dans le domaine du tri d'objets par des fourmis artificielles. Dans [Deneubourg, 1990], Deneubourg et son équipe modélisent ce comportement : Les fourmis artificielles se déplacent sur un plan, où sont disposés les objets aléatoirement. Chaque fourmi n'a qu'une perception locale de ces objets et ne communique pas avec les autres fourmis. Elle effectue des opérations de ramassage et de dépôt des objets afin de construire des tas d'objets similaires. Pour cela, les probabilités de ramasser un objet (P_p) et de le déposer (P_d) ont été explicitées [Deneubourg, 1990] : quand une fourmi ne transporte aucun élément, sa probabilité d'en ramasser un trouvé sur son chemin dépend de la probabilité (P_p), une fois un objet est ramassé, la fourmi se déplace et le dépose suivant la probabilité (P_d).

$$P_p = \left(\frac{k_1}{k_1 + f} \right)^2 \quad (1.4)$$

$$P_d = \left(\frac{f}{k_2 + f} \right)^2 \quad (1.5)$$

f est une estimation du nombre d'objets rencontrés dans un passé récent. k_1 et k_2 sont des constantes positives. Quand $f \ll k_1$ cela signifie qu'il y a peu d'objets dans le voisinage de l'objet et donc la probabilité de le prendre est élevée (P_p est proche de 1). Inversement quand $f \gg k_1$ la probabilité de prendre l'objet est faible s'il est entouré de plusieurs objets. Ces principes relativement simples font qu'il apparaît des regroupements d'objets. L'approche peut être généralisée à plusieurs types d'objets (les fréquences f sont spécifiques à chaque type d'objet) : cet algorithme permet alors de trier des objets. En plus de ce mécanisme d'auto-organisation collective, les fourmis peuvent aussi se spécialiser pour un certain type d'objets par un mécanisme

d'apprentissage [Sendova et Franks, 1992]. Ces principes ont trouvé leurs premières applications en robotique collective.

1.5.2.2 Modèle de Lumer et Faieta

L'algorithme proposé par Deneubourg a été repris et étendu par Lumer et Faieta [Lumer et Faieta, 1994] pour la classification des données numériques. Les extensions portées concernent en particulier les points suivants :

- Les données sont représentées par des vecteurs de caractéristiques (numériques) et sont initialement réparties sur une grille 2D.
- La similarité entre deux données est mesurée comme une distance euclidienne entre leur vecteur de caractéristiques.
- La fourmi est capable de percevoir une région de R_S de $s \times s$ cases autour de sa position courante (figure 1.7).

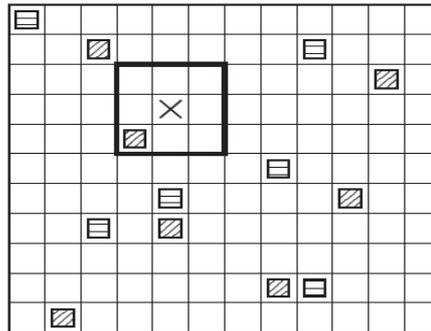


FIG 1.10. Grille utilisée dans [Lumer et Faieta, 1994] : la fourmi est représentée par \times et son périmètre de détection par un trait épais. Les objets sont représentés par des carrés dont l'intérieur représente la classe d'origine.

Les probabilités de déplacement et de dépôts des objets deviennent alors :

$$P_p(o_i) = \left(\frac{k_1}{k_1 + f(o_i)} \right)^2 \quad (1.6)$$

$$P_d(o_i) = \begin{cases} 2f(o_i) & \text{si } f(o_i) < k_2 \\ 1 & \text{sinon} \end{cases} \quad (1.7)$$

Avec

$$f(o_i) = \begin{cases} \frac{1}{s^2} \sum_{o_j \in R_s} 1 - \frac{d(o_i, o_j)}{\alpha} & \text{si } f > 0 \\ 0 & \text{sinon} \end{cases} \quad (1.8)$$

$f(o_i)$ est la mesure de similarité de l'objet o_i avec les objets o_j présents dans son voisinage, le facteur α contrôle la consistance de la fonction de dissimilarité entre les objets. L'algorithme 1.1 résume les étapes de la méthode en utilisant A fourmis $\{a_1, \dots, a_A\}$ pour la classification de N objets.

Algorithme 1.1 : Algorithme de LF [Lumer et Faieta, 1994]

- (1) Placer aléatoirement les N objets o_1, \dots, o_N sur la grille G .
 - (2) **pour** $T = 1$ à T_{\max} **faire**
 - (3) **pour tout** $a_j \in \{a_1, \dots, a_A\}$ **faire**
 - (4) **si** la fourmi a_j ne transporte pas d'objets et $r(o_i) = r(a_j)$ **alors**
 - (5) Calculer $f(o_i)$ et $P_p(o_i)$
 - (6) La fourmi a_j ramasse l'objet o_i suivant la probabilité $P_p(o_i)$
 - (7) **sinon**
 - (8) **si** la fourmi a_j transporte l'objet o_i et la case $r(a_j)$ est vide **alors**
 - (9) Calculer $f(o_i)$ et $P_d(o_i)$
 - (10) La fourmi a_j dépose l'objet o_i sur la case $r(a_j)$ avec une probabilité $P_d(o_i)$
 - (11) **Finsi**
 - (12) **Finsi**
 - (13) Déplacer la fourmi a_j sur une case voisine non occupée par une autre fourmi
 - (14) **Finpour**
 - (15) **Finpour**
 - (16) Retourner l'emplacement des objets sur la grille
-

Les résultats obtenus ont montré que l'algorithme a tendance à toujours générer plus de groupes que le nombre attendu idéalement. De plus la configuration de la grille a tendance à créer des effets de bord. Enfin, des clusters différents placés sur des cases adjacentes peuvent être fusionnés.

1.5.2.3 L'algorithme AntClass

Dans [Monmarché, 1999], Monmarché introduit l'algorithme AntClass. Celui-ci découvre automatiquement les classes dans des données numériques sans connaître

le nombre de classes à priori, sans partition initiale. Il s'inspire de la méthode de Lumer et Faieta et y rapporte des modifications pour en éviter les travers décrits précédemment. Ainsi, les fourmis se déplacent sur une grille toroïdale et chaque fourmi a la possibilité de transporter plusieurs objets à la fois et de déposer un tas d'objets sur une même case de la grille. Un cluster dans la partition finale devient donc un élément de la grille, plutôt qu'un ensemble connexe de ces derniers. Enfin l'apport majeur de la méthode AntClass réside dans son hybridation avec un algorithme classique de type K-means. En fait l'heuristique comporte deux exécutions successives d'un motif formé d'une application des fourmis suivie de K-means. Cela permet de réduire le nombre de classes trouvées et autorise l'affinage des résultats d'une application d'un motif à l'autre en changeant les paramètres des fourmis ou encore la taille de la grille.

1.5.3 Autres domaines d'application

Les travaux de Lumer et Faieta ont inspiré d'autres auteurs pour la résolution de problèmes de classification par les fourmis. Kuntz et al [Kuntz et al, 1997] se sont inspirés pour le partitionnement de graphes. A la place des objets manipulés par l'algorithme LF, les sommets d'un graphe sont répartis sur une grille à deux dimensions et les agents-fourmis rassemblent les sommets en fonction de leur connexité dans le graphe. Les applications de ce type d'approche vont de la conception de circuits VLSI à la représentation géométrique de graphe.

On peut trouver une version supervisée de ces algorithmes dans [Bonabeau et al., 1999] se basant sur la capacité des insectes sociaux à suivre des patrons, par exemple mis en oeuvre lors de la construction d'une termitière.

Une autre extension des travaux de [Deneubourg, 1991] et de [Lumer et Faieta, 1994] dans [Ramoss, 2002] pour la classification de mots dans des documents. L'algorithme proposé utilise les traces de phéromones comme moyen de communication entre les fourmis et comme mémoire commune à toutes les fourmis du système de classification. Si une classe disparaît de la grille, la phéromone tend à s'évaporer dans sa direction, ce qui permet d'éviter que les fourmis se dirigent vers des directions non intéressantes. Ce travail a été repris par la suite par Abraham et

Ramos [Abraham, 2003] et appliqué dans le domaine de l'Internet. Une autre variante de l'algorithme de Lumer et Faieta a été proposée dans [Handl, 2003] pour la recherche visuelle des documents pour les moteurs de recherche sur Internet. Malheureusement, le papier est très court pour décrire tous les travaux mis en œuvre dans ce domaine.

1.6 Modélisation de la fermeture coloniale chez les fourmis pour la classification

Les fourmis vivent au sein de sociétés dont la survie est garantie par un mécanisme de fermeture coloniale comparable au système immunitaire humain. Ce dernier repose en fait sur un système de reconnaissance mettant en jeu des messages chimiques portés par chaque individu et le désignant, à la manière d'une carte d'identité, comme appartenant à un tel ou tel nid.

Dans [Labroche, 2002] a été introduit un nouvel algorithme de classification appelé AntClust inspiré du système de reconnaissance chimique des fourmis. Celui-ci connu sous le nom fermeture coloniale, repose sur l'apprentissage et le partage d'une odeur coloniale commune à toutes les fourmis d'un même nid.

1.6.1 Inspiration biologique

Comme tous les insectes sociaux, les fourmis ont développé un mécanisme de fermeture coloniale qui leur permet de privilégier les relations avec les membres de leur nid et de rejeter les intrus qui peuvent être de la même espèce.

La discrimination repose sur la comparaison de l'odeur émise par chaque fourmi, le "label", et d'un modèle de référence nommé "template" : on parle alors de "*phenotype matching*" [Hölldobler et Wilson, 1990]. Chaque fourmi apprend les labels propres à sa colonie à sa naissance, en s'imprégnant physiquement des odeurs des ouvrières de son nid, lorsqu'elles la nourrissent. Par la suite, la fourmi remet à jour continuellement son template en intégrant les labels des autres fourmis et en

diffusant le sien pour être reconnue à son tour. Ces échanges répétés conduisent à la mise en place d'une odeur coloniale commune à tous les membres du nid.

Les labels sont principalement constitués d'hydrocarbures cuticulaires et de substances chimiques extraites de la nourriture ou bien issues de matériaux constitutifs du nid. Selon les espèces certains facteurs peuvent influencer la reconnaissance entre fourmis. On note le rôle particulier que peut alors jouer la reine [Carlin et Hölldobler, 1987], outre celui de donner naissance aux nouvelles fourmis. Soit la reine ne participe qu'à la diffusion de l'odeur coloniale au plus grand nombre, soit elle intervient directement dans sa composition chimique. D'un point de vue individuel, les hydrocarbures sont générés par les cellules oenocytes selon le génome de chaque fourmi et sont ensuite distribués par des circuits internes vers la glande post-pharyngienne (GPP) ou vers la cuticule des fourmis. Lors d'auto-toilettages les fourmis sont capables de renforcer leur propre label en déversant une partie du contenu de leur GPP sur leur cuticule et assurent ainsi leur reconnaissance au sein de la colonie. L'odeur coloniale est un mélange de tous les labels des fourmis du nid échangés par le biais de toilettages sociaux (une fourmi déverse un peu du contenu de sa GPP sur la cuticule d'une autre), par trophallaxie (une fourmi transfère le contenu de sa GPP dans la GPP d'une autre), ou plus simplement par contacts cuticulaires dans le nid. Ce modèle de répartition des labels à tous les membres d'une même colonie de façon homogène est appelé modèle Gestalt et conduit à l'établissement de l'odeur coloniale.

1.6.2 L'algorithme AntClust

1.6.2.1 Principe

AntClust [Labroche et al. 2002] est un algorithme de classification inspiré du système de reconnaissance chimique des fourmis. Dans celui-ci, chaque fourmi possède une odeur propre appelée label, partiellement définie par son génome, qui lui permet d'être acceptée au sein de son nid, ainsi qu'un modèle de reconnaissance de ce que doit être l'odeur d'un membre du nid, appelé template. Ces « odeurs » individuelles évoluent lors des rencontres entre fourmis et définissent à l'échelle du nid une odeur coloniale (« Gestalt theory »).

Similairement, dans cet algorithme, le génome de chaque fourmi artificielle est associé à un objet du jeu de données. Chaque fourmi apprend ensuite un seuil d'acceptation (son template), défini par les similarités observées entre son génome et celui d'autres fourmis choisies aléatoirement. Chaque fourmi va ensuite réaliser des rencontres aléatoires de façon à déterminer le label (son appartenance à un nid ou cluster) qui correspond le mieux à son génome grâce à un ensemble de règles comportementales individuelles. A la fin, les fourmis ayant des génomes similaires sont réparties dans les mêmes nids, ce qui forme la partition désirée des objets.

1.6.2.2 Modèle des fourmis artificielles

Dans cette méthode, chaque donnée de l'espace de départ va être représentée par une fourmi artificielle et plus précisément par son génome. Tout au long des rencontres qu'elle va effectuer, la fourmi va tenter d'accorder son label et son template à son génome pour trouver la colonie qui lui ressemble le plus. Nous définissons donc les paramètres suivants pour une fourmi :

- Le label $Label_i$ indique le nid d'appartenance de la fourmi i et est modélisé par une variable représentant l'indice du nid. Au départ, les fourmis n'appartiennent à aucun nid et donc $Label_i = 0$. Cette valeur évolue jusqu'à ce que la fourmi trouve le nid qui lui convient plus.
- Le génome $Génome_i$ correspond à un objet de jeu de données. Lorsqu'elles se rencontrent, deux fourmis comparent leur génome pour évaluer leur similarité.
- Le template est à la fois défini par le $Génome_i$ de la fourmi et par un seuil d'acceptation noté $Template_i$. Celui-ci fait l'objet d'un apprentissage à l'initialisation des fourmis artificielles et d'une mise à jour continue pendant la classification. Le calcul de $Template_i$ s'appuie sur l'estimation par la fourmi i des similarités maximales et moyennes observées lors des rencontres avec d'autres fourmis et notées respectivement $\max(Sim(i, \cdot))$ et $\overline{Sim}(i, \cdot)$. une fourmi i définit son seuil initial d'acceptation $Template_i$ de la manière suivante :

$$Template_i \leftarrow \frac{\overline{Sim}(i, \cdot) + \max(Sim(i, \cdot))}{2} \quad (1.9)$$

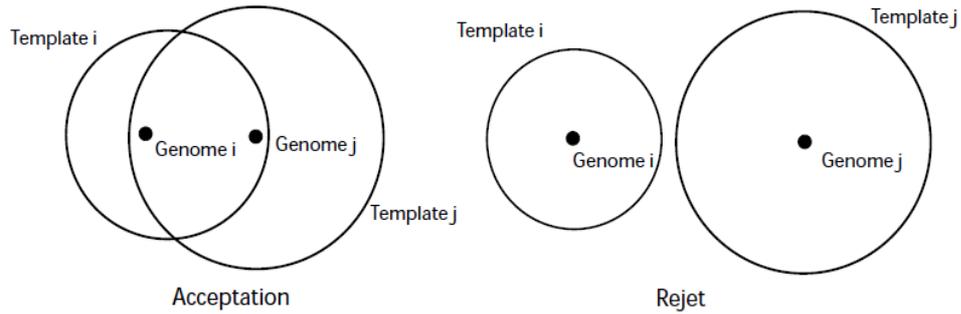


FIG 1.11. Principes d'acceptation et du rejet entre deux fourmis i et j

- L'estimateur M_i reflète la réussite des rencontres de la fourmi i . Au départ, M_i vaut 0 puisque la fourmi i n'a pas encore réalisé de rencontres. M_i estime la taille du nid de la fourmi i , c'est à dire le nombre de fourmis ayant le même label que la fourmi i . M_i est augmenté quand la fourmi i rencontre des individus de son nid et est diminué dans le cas contraire. M_i permet à chaque fourmi d'estimer la taille de son propre nid.
- L'estimateur M_i^+ mesure l'intégration de la fourmi i dans son nid. Il est augmenté si la fourmi i et une autre de son nid se rencontrent et s'acceptent et est diminué sinon.
- L'âge A_i qui au départ vaut 0 est utilisé dans les calculs de mise à jour du seuil d'acceptation $Template_i$.

Algorithme 1.3 : Algorithme principal d'AntClust

- (1) Initialiser les fourmis artificielles.
 - (2) $Génome_i \leftarrow i^{ème}$ objet des données à classer.
 - (3) $Label_i \leftarrow 0$.
 - (4) $Template_i$ est initialisé selon l'équation **(1.9)**
 - (5) $M_i \leftarrow 0$, $M_i^+ \leftarrow 0$, $A_i \leftarrow 0$.
 - (6) Simuler Nb_{iter} durant lesquelles chaque fourmi en rencontre une autre choisie aléatoirement.
 - (7) Détruire les nids inintéressants.
 - (8) Réaffecter chaque fourmi son nid, au nid de la fourmi la plus similaire.
-

Labroche a introduit également un ensemble de règles comportementales des fourmis artificielles s'appliquant lors de chaque rencontre entre fourmis. Ces règles

entraînent l'évolution des paramètres (*label*, *template*, M_i , M_i^+) jusqu'à ce que un ensemble de nids stable soit formé.

1.7 Autres sources d'inspiration

Il existe dans la littérature d'autres sources d'inspiration pour résoudre les problèmes de la classification. Dans [AZZ 03], a été introduit un nouveau modèle pour la construction automatique d'un site portail (données textuelles). Ce modèle permet d'effectuer rapidement une classification hiérarchique. Il s'agit de copier la manière dont les fourmis construisent des structures vivantes en s'accrochant les unes aux autres en fonction de critères locaux (la forme de la structure influençant le comportement d'accrochage ou de décrochage). Dans ce modèle, chaque fourmi artificielle représente une donnée. Les fourmis sont placées initialement à la racine de l'arbre et vont pouvoir se déplacer dans cet arbre et s'accrocher afin de construire une structure hiérarchique dont chaque nœud représente une donnée.

1.8 Conclusion

Dans ce chapitre, nous avons donné un aperçu des principales caractéristiques et de comportements des fourmis réelles pouvant être mis à profit pour la conception d'algorithmes de résolution des problèmes. En effet, ces insectes offrent un modèle naturel d'un système collectif auto-organisé, décentralisé, fondé sur des individus autonomes au comportement relativement simple et probabiliste qui sont distribués dans l'environnement et ne disposent que d'informations locales. Le projet global n'est donc pas programmé explicitement chez les individus, mais émerge d'un grand nombre d'interactions entre ces individus, ou entre individus et environnement. La coordination des activités des individus est basée sur un processus stigmergique.

L'étude des comportements des fourmis a inspiré les chercheurs dans plusieurs domaines liés à l'informatique et à la robotique et a permis d'une part aux informaticiens d'introduire de nouvelles heuristiques pour résoudre différents problèmes (optimisation, classification) et d'autre part aux éthologistes de développer et de tester leur théories de comportement animal.

Chapitre 2

L'algorithme AntClass

Nous présentons dans ce chapitre, un nouvel algorithme de classification non-supervisée appelé *AntClass Amélioré* inspiré du comportement de tri collectif observé chez les fourmis. Cet algorithme est une nouvelle extension de l'algorithme *AntClass*. Celui-ci découvre automatiquement les classes dans des données numériques sans connaître le nombre de classes a priori, sans partition initiale. Il utilise les principes exploratoires stochastiques d'une colonie de fourmis avec les principes heuristiques et déterministes de l'algorithme des centres mobiles. Les fourmis se déplacent sur une grille à deux dimensions et peuvent transporter des objets. La saisie ou la dépose d'un objet sur un tas dépend de la similarité entre cet objet et les objets du tas. L'algorithme des centres mobiles est ensuite utilisé pour améliorer la partition obtenue par les fourmis. L'algorithme *AntClass Amélioré* est basé sur les mêmes principes. Cependant, dans l'objectif d'accélérer la convergence de l'algorithme tout en améliorant la qualité de la partition, il introduit des modifications de base concernant le comportement des fourmis en face d'un objet

2.1 Introduction

La classification aide à fournir une vue synthétisée des données qu'elle traite. De nos jours, la classification est à la base de la plupart des applications en reconnaissance des formes et constitue une phase primordiale lors de développement d'un processus d'extraction des connaissances à partir de données. Ainsi les problèmes de la classification de données sont identifiés comme une des problématiques majeures de ces domaines.

Comme nous l'avons constaté dans le chapitre précédent, la source d'inspiration que constituent les modèles biologiques de fourmis a de plus en plus de succès notamment dans le domaine de la classification. Une des premières études relatives à ce domaine, a été menée par [Deneubourg et al, 1990] où un ensemble de fourmis se déplacent aléatoirement sur une grille à deux dimensions, et sont capables de déplacer des objets dans le but de les rassembler. Cette méthode a été reprise et étendue par [Lumer et Faieta, 1994] pour la classification des données numériques. Plusieurs variantes de cette méthode ont vu le jour ultérieurement pour tenter d'améliorer l'approche initiale. Nous étudions dans ce chapitre l'algorithme AntClass [Monmarché, 1999] qui s'inspire de la méthode de Lumer et Faieta [Lumer et Faieta, 1994]. Ces algorithmes sont à la base des développements que nous présentons par la suite, l'algorithme AntClass amélioré.

En se basant sur les travaux existants, nous contribuons à l'étude des fourmis artificielles dans l'objectif de résoudre des problèmes de partitionnement.

Pour cela, nous présentons une nouvelle extension de l'algorithme AntClass [Monmarché, 1999]. Celui-ci découvre automatiquement les classes dans des données numériques sans connaître le nombre de classes a priori, sans partition initiale. Il utilise les principes exploratoires et stochastiques d'une colonie de fourmis avec les principes heuristiques et déterministes de l'algorithme des centres mobiles. Les fourmis se déplacent sur une grille à deux dimensions et peuvent transporter des objets. La saisie ou la dépose d'un objet sur un tas dépend de la similarité entre cet objet et les objets du tas. L'algorithme des centres mobiles est ensuite utilisé pour améliorer la partition obtenue par les fourmis. L'algorithme AntClass Amélioré [Senouci, 2008] est basé sur les mêmes principes. Cependant, dans l'objectif d'accélérer la convergence de l'algorithme tout en améliorant la qualité de la partition, il introduit des modifications de base concernant le support et le déplacement des fourmis ainsi que leur comportement en face d'un objet. L'intérêt de ce travail est d'introduire les principes de classification par les fourmis dans des problèmes d'analyse des données.

La suite de ce chapitre est organisée de la façon suivante : la section 2.2 décrit l'algorithme AntClass développé par Monmarché [Monmarché, 1999]. La section 2.3

présente les motivations qui nous sont apparues pour proposer une amélioration aux travaux existants. La section 2.4 décrit l'algorithme AntClass Amélioré.

2.2 Notations et heuristiques de AntClass

2.2.1 Les objets et leurs métriques

- Etant donné un ensemble O de n données $O = \{o_1, \dots, o_n\}$ collectées par un expert du domaine, chaque donnée est un vecteur de k valeurs numériques v_1, \dots, v_k correspondant aux k attributs numériques A_1, \dots, A_k .
- La similarité entre deux objets est mesurée par la distance euclidienne notée d par la suite, sachant que d'autres distances peuvent être utilisées.
- Une grille G à deux dimensions est utilisée et considérée comme une matrice de $m \times m$ cellules, cette matrice est toroïdale afin que les fourmis passent d'un bord à l'autre. La taille de la grille est déterminée automatiquement en fonction du nombre d'objets à traiter. Si N représente le nombre d'objets :

$$m = \sqrt{2N} \quad (2.1)$$

- Un tas T_i est un ensemble d'au moins deux objets localisés dans une case donnée. Par rapport à la représentation de Lumer et Faieta, cette représentation adoptée par Monmarché a l'avantage d'identifier facilement une classe (figure 2.1).

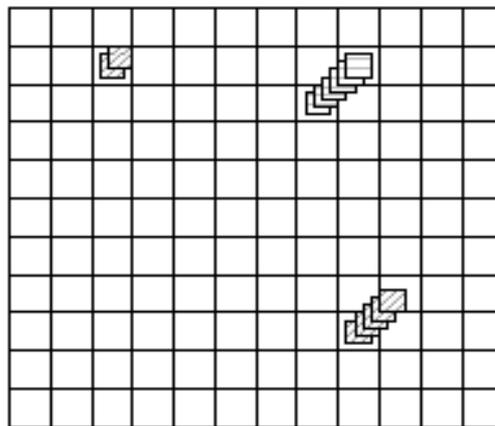


FIG 2.1. AntClass permet la construction de tas d'objets sur la grille

Les notations suivantes vont servir par la suite :

- La distance maximale entre deux objets de l'ensemble :

$$d^*(O) = \max \{d(x_i, x_j) \mid (i, j) \in \{1, \dots, m\}^2\} \quad (2.2)$$

- La distance moyenne entre deux objets de l'ensemble :

$$\bar{d}(O) = \frac{2}{n(n-1)} \sum_{(i,j) \in \{1, \dots, m\}^2, i < j} d(x_i, x_j) \quad (2.3)$$

- La distance maximale entre les objets d'un tas T_j et son centre de gravité g_j :

$$d_g^*(T_j) = \max \{d(x_i, g_j) \mid x_i \in T_j\} \quad (2.4)$$

- La distance moyenne entre les objets d'un tas T_j et son centre de gravité g_j :

$$\bar{d}_g(O) = \frac{1}{|T_j|} \sum_{x_i \in T_j} d(x_i, g_j) \quad (2.5)$$

2.2.2 L'algorithme AntClass

2.2.2.1 Déplacement des fourmis

La colonie se compose de m fourmis a_1, \dots, a_m . Chaque fourmi est positionnée aléatoirement et uniformément sur une case de la grille. Pas de contrôle centralisé dans la colonie. Chaque fourmi a_i se déplace et peut alors ramasser ou déposer un objet. Le déplacement n'est pas totalement aléatoire. Initialement, a_i choisit une direction aléatoire parmi les 8 possibles. a_i a désormais une probabilité $P_{direction}$ de continuer dans cette direction à son prochain déplacement, sinon elle génère aléatoirement une nouvelle direction. Chaque fourmi possède un paramètre *vitesse* qui indique le nombre de pas qu'elle fera dans la direction sélectionnée. Le critère d'arrêt de cet algorithme est le nombre d'itérations.

2.2.2.2 Ramasser un objet

Si la fourmi a_i ne transporte pas d'objets et qu'elle se trouve sur une case contenant un objet ou un tas d'objets T_j , elle a une probabilité $P_p(T_j)$ de ramasser un objet jusqu'à ce que sa capacité soit atteinte.

$$P_p(T_j) = \begin{cases} 1 & \text{si } |T_j| = 1 \\ \min \left\{ \left(\frac{\bar{d}_g(T_j)}{\bar{d}(o)} \right)^{k_1}, 1 \right\} & \text{si } |T_j| = 2 \\ 1 - 0.9 \left(\frac{\bar{d}_g(T_j) + \varepsilon}{d_g^*(T_j) + \varepsilon} \right)^{k_1} & \text{sinon} \end{cases} \quad (2.6)$$

Où ε est une petite valeur positive $(10)^{-5}$ et $|T_j|$ le nombre d'objets dans le tas, k_1 est un paramètre réel positif.

La figure 3.2 présente les différents cas pouvant être rencontrés par une fourmi pour le ramassage d'un objet.

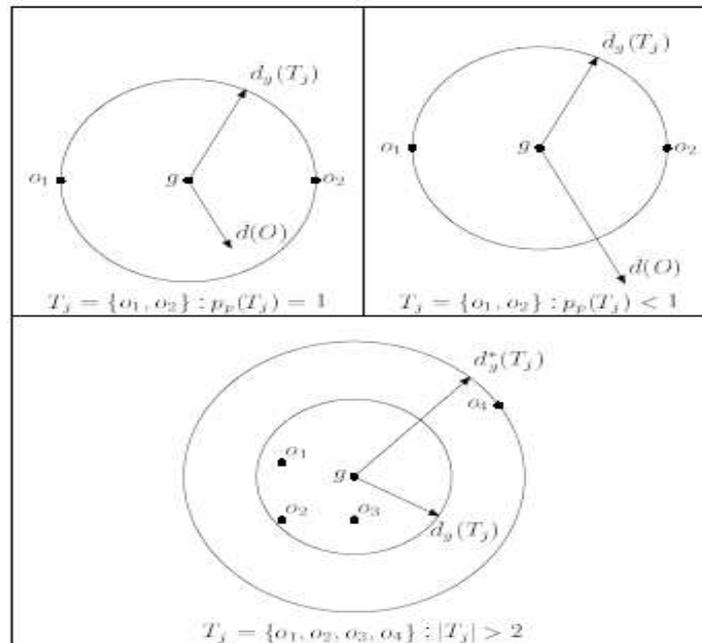


FIG 2.2. Les différents cas possibles pour le ramassage

2.2.2.3 Déposer un objet

Si la fourmi transporte un objet, et qu'elle se trouve sur une case contenant un ou plusieurs objets, sa probabilité de déposer l'objet o_i sur le tas T_j est donnée par :

$$P_d(o_i, T_j) = \begin{cases} 1 & \text{si } d(x_i, g_j) \leq d_g^*(T_j) \\ 1 - 0.9 \min \left\{ \left(\frac{d(x_i, g_j)}{\bar{d}(O)} \right)^{k_2}, 1 \right\} & \text{sinon} \end{cases} \quad (2.7)$$

Où k_2 est un paramètre réel positif. La figure 3.3 présente les différents cas pouvant être rencontrés par une fourmi pour déposer un objet.

Si la capacité de la fourmi est supérieure à 1 et qu'elle transporte plusieurs objets, la probabilité de déposer le tas T_i qu'elle transporte sur le tas T_j est calculée de la même façon que pour un objet unique en remplaçant x_i par le centre de gravité g_i du T_i .

Afin d'éviter qu'une fourmi ne transporte un objet pendant trop longtemps, dans le cas d'objets très dissimilaires aux autres par exemple, la fourmi dépose cet objet automatiquement, après Max_{carry} itérations.

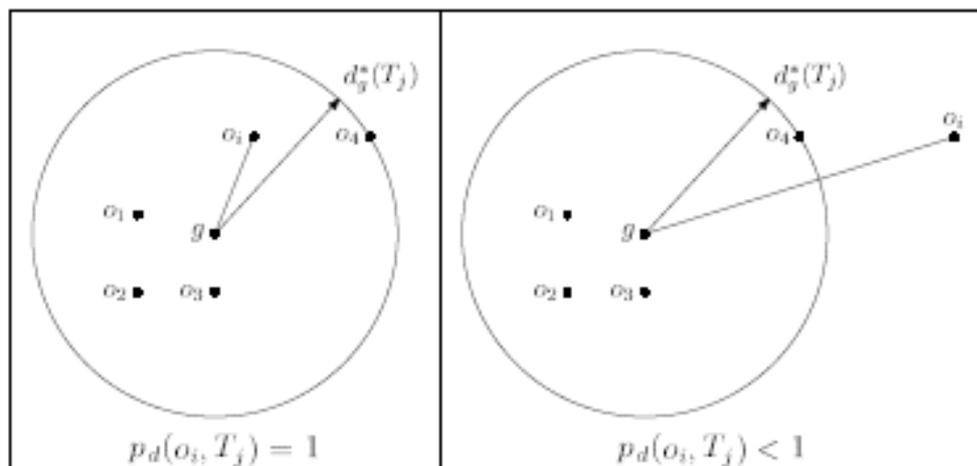


FIG 2.3. Les différents cas possibles pour déposer un objet o_i sur un tas

$$T_j = \{o_1, o_2, o_3, o_4\}$$

2.2.2.4 Mémoire locale des fourmis

Afin d'accélérer le processus de regroupement, la fourmi n'a pas un mouvement complètement désordonné. Pour cela une version modifiée du mécanisme de mémoire à court terme introduit dans [Lumer et Faieta, 1994] et [Monmarché, 1999] est proposée. Dans l'approche de Lumer et Faieta [Lumer et Faieta, 1994], chaque fourmi mémorise les m derniers objets qu'elle a ramassés ainsi que leurs emplacements sur la grille. À chaque fois qu'elle ramasse un nouvel objet, il est comparé aux objets contenus dans sa mémoire. Elle se dirige après vers l'emplacement de l'objet qui lui est le plus similaire en terme de distance euclidienne. Monmarché [Monmarché, 1999] reprend ces idées et utilise la distance entre le centre de gravité du tas transporté par la fourmi et les tas qu'elle a mémorisés.

2.2.2.5 Patience des fourmis

S'il y a trop de fourmis par rapport au nombre de tas ou d'objets, le problème suivant peut survenir : tous les tas ou tous les objets sont transportés ce qui n'offre plus de possibilité aux fourmis de déposer ce qu'elles transportent. La solution proposée est de doter les fourmis d'une certaine pertinence, qui peut être individuelle, et notée $p(a_i)$. Quand la fourmi a effectué plus de $p(a_i)$ déplacements sans avoir réussi à déposer les objets qu'elle transporte, elle les dépose sur la case où elle se trouve si elle est vide ou l'une de son voisinage dans le cas contraire.

L'algorithme 2.1 donne la structure générale de la méthode la classification par les fourmis (appelé Ants par la suite).

Algorithme 2.1 : Algorithme Ants

- (1) **pour** $t = 1$ à T **faire**
 - (2) **pour** $k = 1$ à A **faire**
 - (3) Déplacer la fourmi a_k sur une case non occupée par une autre fourmi.
 - (4) **si** il y a un tas d'objets T_j sur la même case que a_k **alors**
 - (5) **si** la fourmi a_k transporte l'objet o_i [un tas d'objets T_i] **alors**
 - (6) Déposer l'objet o_i [le tas T_i] transporté par la fourmi sur le tas T_j suivant la probabilité $P_d(o_i, T_j) [P_d(T_i, T_j)]$.
 - (7) **sinon**
 - (8) Ramasser l'objet o_i le plus dissimilaire du tas T_j [jusqu'à ce que la capacité $c(a_k)$ soit atteinte ou que le tas soit vide] selon la probabilité $P_p(T_j)$.
 - (9) **finsi**
 - (10) **finsi**
 - (11) **finpour**
 - (12) **finpour retourner** la grille G .
-

2.2.3 Hybridation avec les centres mobiles

Considérons qu'une partition O a été obtenue sous la forme de K classes c_1, c_2, \dots, c_K composées respectivement de $|c_1|, \dots, |c_K|$ objets. Comme chaque objet appartient à une et une seule classe, on a $\sum_{i=1}^K |c_i| = N$. Le centre de gravité g_i de la classe c_i est donné par :

$$g_i = \frac{1}{|c_i|} \sum_{j=1}^{|c_i|} x_j^{(i)} \quad (2.8)$$

Où $x_j^{(i)}$ est le j -ième point de la classe c_i . L'erreur quadratique sur la classe est donnée par :

$$\mathcal{E}_i^2 = \sum_{j=1}^{|c_i|} d^2(x_j^{(i)}, g_i) \quad (2.9)$$

Notons par I_w , l'inertie intra-classe :

$$I_w = \sum_{j=1}^K \mathcal{E}_j^2 \quad (2.10)$$

Le partitionnement fourni par les fourmis n'est pas obligatoirement optimal au sens d'inertie intra-classe I_w , l'algorithme des centres mobiles [Jain et Dubes, 1988] (algorithme 3.2) (ou plus couramment K-means) offre une réponse à la non-uniformité de la partition construite par les fourmis et même aux objets qui ne sont affectés à aucun tas (ces objets peuvent correspondre à des objets qui sont encore transportés par les fourmis ou à des objets qui sont seuls sur la grille). K-means est alors utilisé pour corriger ce type de problèmes.

Algorithme 2.2 : Algorithme K-means (Partition P de K classes).

- (1) **tanque** l'inertie intra classe s'est pas stabilisée **faire**
 - (2) Générer une nouvelle partition P' en affectant chaque objet à la classe dont le centre est le plus proche.
 - (3) Calculer les centres de gravité des classes de la nouvelle partition P' .
 - (4) $P \leftarrow P'$
 - (5) **fin tanque**
 - (6) **retourner** P
-

En résumé l'heuristique comporte deux exécutions successives d'un motif formé d'une application des fourmis suivie de K-means. Cela permet de réduire le nombre de classes trouvées et autorise l'affinage des résultats d'une application d'un motif à l'autre en changeant les paramètres des fourmis ou encore la taille de la grille.

L'algorithme 3.3 donne le schéma général de AntClass, l'algorithme K-means est initialisé avec la partition obtenu par Ants.

Algorithme 2.3 : Algorithme AntClass

- (1) Soit P_0 la partition initiale formée de N classes.
 - (2) **pour** $t = 1$ à $T_{AntClass}$ **faire**
 - (3) Initialiser la grille G à partir de la partition P_{t-1}
 - (4) $G' \leftarrow Ants(G)$.
 - (5) Construire la partition P' associée à la grille G'
 - (6) $P_t \leftarrow K - Means(P')$
 - (7) **fin pour**
 - (8) **retourner** la partition $P_{T_{AntClass}}$.
-

2.3 Motivations

L'algorithme AntClass été expérimenté sur des données réelles et les résultats étaient prometteurs. Les principales critiques concernent le temps de calcul relativement important, aussi plusieurs paramètres se rattachent aux fourmis et donc il n'est pas facile de trouver un paramétrage adéquat, ce qui complique relativement l'algorithme. Nous citons en particulier :

- Lorsqu'une fourmi se déplace vers une case pour y déposer un objet suivant $P_p(T_j)$, alors que $P_p(T_j)$ ne lui permet pas de le déposer dans cette case, la fourmi peut passer un certain temps à trouver une case libre sur la grille.
- Lors de déplacement d'une fourmi, celle-ci peut être dérangée en se coïncidant durant son déplacement avec les autres fourmis qui se trouvent dans la grille occupant ainsi les cases sans aucune activité.
- Les fourmis sont dotées de plusieurs paramètres : *vitesse* des fourmis, probabilité d'une direction, patience de fourmis, et même Max_{carry} . En outre, ces paramètres sont hétérogènes ce qui rend le paramétrage complexe alors que l'utilisation de l'algorithme par des experts du domaine doit être simplifiée.
- Lorsque la mémoire de la fourmi intervient, l'objet est comparé avec les éléments de cette mémoire et ajouté à l'élément le plus proche en terme de distance or dans le cas où cet objet est comparé avec les éléments rencontrés dans la grille, il est ajouté selon la probabilité $P_p(T_j)$.

2.4 L'algorithme AntClass amélioré

Nous reprenons les travaux de Lumer et Faieta et ceux de Monmarché pour en améliorer certains points cités en section 2.3.

2.4.1 Données dont dispose une fourmi dans son environnement

Les fourmis sont capables d'émettre et de capter des molécules appelées phéromones. Un œuf de fourmi émet des phéromones différentes suivant son stade d'évolution. On peut donc supposer qu'une fourmi est capable d'estimer la concentration en œufs du même stade au point où elle se trouve en reniflant les phéromones (qu'elles captent avec leurs antennes). Par contre, la myopie des fourmis leur interdit d'avoir une vue d'ensemble du couvain [Deneubourg, 1990]. Comme les fourmis réelles ont la possibilité de mémoriser plusieurs sites dans leur environnement [Fresneau, 1985], nous allons aussi supposer qu'une fourmi dispose d'une mémoire des œufs qu'elle a rencontrés durant son déplacement.

2.4.2 Modélisation du comportement des fourmis

Nous allons supposer que la fourmi dispose d'une perception spatiale et d'une perception temporelle. Donc nous pouvons réaliser deux modèles de comportement, le premier suppose que la fourmi a une perception spatiale des phéromones émises par les œufs de son voisinage au point où elle se trouve et mémorise les œufs et leur emplacement dans une mémoire à court terme. Le deuxième fait intervenir la mémoire à long terme des œufs et des tas d'œufs rencontrés ou construits durant son déplacement.

2.4.2.1 Environnement et déplacement des fourmis

L'environnement ressemble à celui utilisé dans AntClass, une grille de $m \times m$ cases. Initialement les N objets à partitionner sont éparpillés aléatoirement sur la grille. Les fourmis seront capables de ramasser et de déposer des objets afin de créer des tas d'objets qui correspondent aux classes dans ce processus de classification.

2.4.2.2 Ramassage et dépôt des objets

Lorsqu'une fourmi a_i arrive à une case vide ou comportant un tas d'objets, elle ignore cette case et se déplace vers une autre case jusqu'à ce qu'elle se retrouve sur une case contenant un seul objet o_i . Dans ce cas, elle explore les cases voisines v_j et prend une décision concernant le dépôt des objets des cases voisines (non vides) sur l'objet o_i avant de les ramasser contrairement à l'algorithme AntClass dans lequel une fourmi peut passer un certain temps à trouver une case libre sur la grille pour un objet ramassé (voir deuxième motivation de la section 3.3). Pour bien comprendre, voici les détails :

- Etape 1 : la fourmi a_i explore son voisinage à l'aide de sa perception spatiale à partir de la case courante.
- Etape 2 : elle considère chaque case voisine v_j et vérifie si elle peut y déposer l'objet o_i suivant la probabilité $P(o_i, T_j)$. Si cette dernière est vérifiée, l'emplacement de cette case voisine et son contenu seront mémorisés dans la mémoire $M(a_i)$ à court terme, dont la taille maximale est 8 (le nombre de voisins maximale est 8).
- Etape 3 : après avoir mémorisé toutes les cases voisines dont il est possible d'y déposer l'objet o_i suivant $P(o_i, T_j)$, la fourmi a_i compare o_i avec les $M(a_i)$ éléments de la mémoire pour l'affecter à l'élément le plus similaire.
- Etape 4 : ce nouveau tas T_j sera également mémorisé dans cette mémoire à court terme après avoir supprimé les cases modifiées (les cases dont les contenus sont agglomérés).
- Etape 5 : la fourmi considère le nouveau tas T_i pour l'ajouter à l'élément le plus similaire de la mémoire suivant la probabilité $P(T_i, T_j)$ et revient à l'étape 4. S'il

n'est pas possible de l'ajouter, la fourmi considère l'élément suivant de la mémoire et refait les étapes 5 et 4 et ainsi de suite, jusqu'à ce qu'elle ne peut plus faire des agrégations car soit les éléments restants dans la mémoire ne peuvent pas être agglomérés suivant $P(T_i, T_j)$, soit il ne reste qu'un seul tas en mémoire. Comme on peut le remarquer, la fourmi effectue un balayage complet de son voisinage afin de vérifier toutes les possibilités d'agglomérer les objets contenant dans cette région. Ce mécanisme introduit ici représente l'apport majeur de cette méthode et permet surtout d'accélérer la convergence de l'algorithme tout en tenant compte de la qualité de la classification.

- Etape 6 : le travail n'est pas terminé, en considérant les nouveaux tas T_i construits dans ce voisinage, il faut intervenir la mémoire à long terme pour vérifier s'il est possible d'ajouter chacun d'eux à l'élément le plus similaire (dans le cas ou ce dernier n'a pas été modifié par les autres fourmis) dans cette mémoire suivant $P(T_i, T_j)$ ou pour le mémoriser dans le cas contraire.
- Etape 7 : la fourmi met à jours la mémoire à long terme, quand à l'autre mémoire, elle est libérée pour le prochain déplacement et le même processus se répète.

Remarque : au niveau de l'étape 2, si $P(o_i, T_j)$ ne permet pas de déposer l'objet o_i avec aucun élément de voisinage, la fourmi passe directement à l'étape 6 mais en considérant cette fois ci l'objet o_i au lieu de T_i .

Le schéma général de l'algorithme est le suivant :

Algorithme 2.4 : Le nouvel algorithme Ant

- (1) **pour** $t = 1$ à T **faire**
 - (2) **pour** $k = 1$ à A **faire**
 - (3) Faire les 7 étapes .
 - (4) **finpour**
 - (5) **finpour**
-

2.4.3 Hybridation

L'algorithme précédent fournit un regroupement pertinent de l'espace de départ sans aucune information initiale. Cependant, deux problèmes demeurent. Le premier est dû au fait que certains objets ne sont affectés à aucun tas quand l'algorithme s'arrête. Le second problème est que si un objet a été affecté à un mauvais tas, cela peut prendre un certains temps pour que cet objet soit transporté sur le bon tas. La solution proposée par Monmarché dans AntClass est d'appliquer l'algorithme k-means. Puisque la partition initiale est déjà générée par les fourmis, il suffit de calculer le centre de chaque, puis tous les objets seront affectés à la classe dont le centre est le plus proche.

Les expériences que nous avons menées montrent que les deux étapes précédentes (AntClass Amélioré et kmeans) génèrent des petits tas très homogènes et donc le nombre de classes est surestimé. La solution proposée par Monmarché dans une autre version de AntClass[Monmarché, 1999] et que nous avons reprise est de considérer ces petits tas comme des objets eux-mêmes et d'y appliquer les fourmis à nouveau.

AntClass : Mise en œuvre et résultats

*D*ans ce chapitre, nous allons présenter les travaux que nous avons menés sur la classification non supervisée en appliquant l'algorithme AntClass sur des bases de données issues de Machine Learning Repository .

3.1 Etude expérimentale

3.1.1 Description des données

Afin d'évaluer les résultats obtenus par AntClass Amélioré, nous avons utilisé des bases de données réelles issues de *Machine Learning Repository* [Blake et Merz, 1998] et semblables à celles utilisées dans [Monmarché, 1999], les informations générales sur ces bases sont représentées dans le tableau 3.1.

Base de données	Nombre d'instances	Nombre de classes
Iris	150	3
Breast Cancer	699	2
Soybean	47	4

TAB 3.1. Les bases de données réelles utilisées issues de [Blake et Merz, 1998].

3.1.2 Les paramètres de AntClass

La performance de AntClass dépend d'un certain nombre de paramètres. Le tableau 3.2 résume les valeurs des paramètres de AntClass avec lesquels nous avons obtenu de bons résultats. Ces valeurs sont appliquées pour la classification de toutes les bases de données décrites dans le tableau 3.1.

Paramètre	Description	Valeur ou intervalle
A	Nombre de fourmis	Proportionnel à la taille de la grille
P	Probabilité de ramasser ou déposer un objet	[0.7, 0.8]
T _{create1}	La dissimilarité maximale autorisée pour agréger deux objets	[0.2, 1]
T _{create2}	La dissimilarité maximale autorisée pour agréger deux tas d'objets	[0.2, 0.3]

TAB 3.2. Les paramètres de AntClass

3.1.3 Les mesures d'évaluation

La validité d'une partition dépend de :

- Nombre de classes obtenues.
- Taux de classification des données.

Afin de pouvoir évaluer la qualité du partitionnement obtenue, les bases de données sont supervisées (pour chaque objet on en connaît la classe) cette classe n'est pas donnée à AntClass. Ceci nous permettra de calculer le taux de classification des données de chaque classe.

3.1.4 Résultats et discussions

Le processus de classification a comporté deux exécutions successives d'un motif formé d'une application des fourmis suivie de K-means. Les résultats que nous avons obtenus sont donnés dans le tableau 3.3 où L , K , T_B , E_c , représentent respectivement le nombre de classes réel, le nombre de classes trouvé, le taux de classification de toute la base, et l'erreur de classification. Afin de pouvoir comparer ces résultats avec ceux obtenus dans [Monmarché, 1999], nous avons ajouté les informations K_{an} et E_{an} représentant respectivement le nombre de classes et l'erreur de classification correspondante aux travaux de [Monmarché, 1999].

La base	L	K	T_B	E_c	K_{an}	E_{an}
Iris	3	3	92 %	8 %	3	15.4 %
Breast cancers	2	2	95.27 %	4.72%	4	3.50 %
Soybean	4	5	100 %	0 %	5	0.13 %

TAB 3.3. Résultats de la classification

Les résultats montrent clairement que l'algorithme arrive à découvrir un nombre égal ou très proche du nombre correct de classes avec un taux de reconnaissance élevé pour les bases Iris et Breast cancers, pourtant cette dernière est la plus grande base car elle comporte 699 instances, pour la base Soybean l'algorithme tente d'atteindre un nombre de classes approximatif tout en évitant les erreurs de classification.

En comparant ces résultats avec ceux de [Monmarché, 1999], l'algorithme s'avère très compétitif de point de vue de performance et de vitesse de convergence notamment dans les grandes bases de données comme Breast cancers.

Le graphe 3.1 donne pour la base Breast cancers l'évolution de la construction des tas au cours des 27 itérations, ainsi que le nombre d'objets non classés, il s'agit des objets isolés sur les quels on a appliqué l'algorithme K-means pour les attribuer aux tas construit par les fourmis.

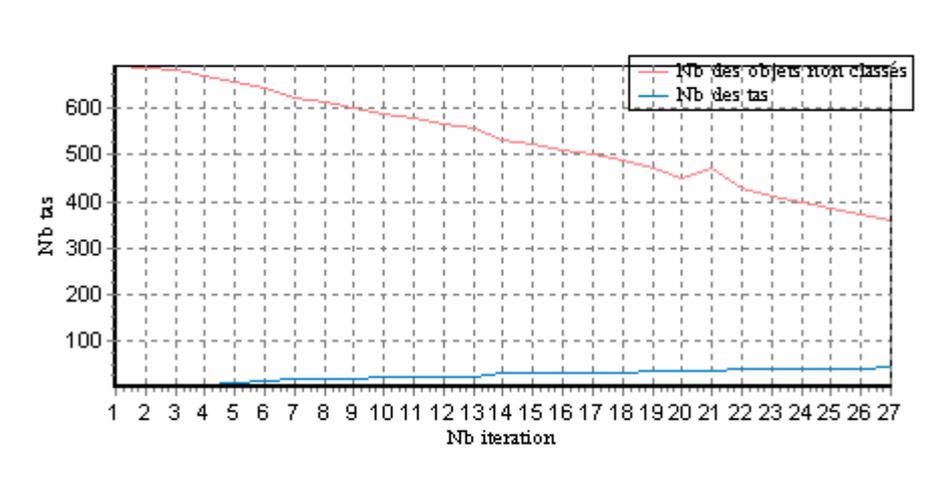


FIG 3.1. Evolution du nombre de classes au cours des itérations de AntClass pour la base Breast cancers

3.2 Conclusion

Au cours de ce chapitre, nous avons présenté un nouvel algorithme pour la classification non supervisée des données numériques. Cet algorithme bien qu'il ne nécessite pas une connaissance préalable du nombre de classe ni une partition initiale des données, arrive à identifier un nombre égal (ou très proche) du nombre réel de classes avec un taux de classification appréciable. AntClass Amélioré est à la base des travaux de Lumer et Faieta et ceux de Monmarché, cependant, il introduit des modifications qui visent essentiellement à améliorer certains points pour accélérer la convergence tout en améliorant les résultats : le comportement des fourmis lorsqu'elles ramassent un objet et passent beaucoup de temps pour lui trouver une case libre si cet objet n'a pas pu être déposé dans une case voisine. Les expériences effectuées montrent que AntCass Amélioré fournit de très bons résultats qui s'avèrent très compétitifs par rapports aux résultats des travaux antérieurs, de plus il permet de tirer profit des méthodes classiques de classification comme k-means.

La reconnaissance biométrique basée sur l'iris

Une définition de la biométrie la désigne comme la science des variations biologique et des phénomènes qui s'y rattachent. Un système est appelé « à reconnaissance biométrique », ou par abus de langage « biométrique » s'il est capable de déterminer automatiquement l'identité d'un être humain sur la base de la mesure des caractères biologiques variables entre les individus. La première partie de ce chapitre introduit quelques notions et concepts biométriques, la seconde traite les techniques mises en œuvre dans un système de reconnaissance biométrique basée sur l'iris. L'intérêt de ce chapitre est d'expliquer le processus de la reconnaissance par l'iris d'une part et de pouvoir aborder les chapitres suivant d'une autre part.

4.1 Introduction à la biométrie

De nos jours, la croissance des terminaux d'accès et des moyens de communication tels que les déplacements physiques, les transactions financières, l'accès aux services et bien d'autres encore impliquent le besoin de s'assurer de l'identité des individus. Les moyens classiques de vérification de l'identité tels que le mot de passe, une pièce d'identité, une clef ou un badge ont chacun des faiblesses,

par exemple le mot de passe peut être oublié par son utilisateur, le badge (la clef ou la pièce d'identité) peut être perdu ou volé. Les systèmes biométriques éliminent certains de ces inconvénients et permettent de reconnaître les personnes à partir de leurs caractéristiques physiques (comme le visage, les empreintes digitales, l'iris, etc.) et comportementales (comme la voix, l'écriture, etc.). Les champs et les domaines d'application de la biométrie sont larges, alors que son utilisation fut pratiquement réservée aux applications judiciaires jusqu'à la fin du XX^{ème} siècle, le marché de la biométrie est en très forte croissance depuis les cinq dernières années. En effet, avec la diversification des technologies, on s'attend à une véritable explosion de l'industrie biométrique dans de nouveaux secteurs tels que le service public, le contrôle d'accès aux systèmes d'information et au commerce électronique. En 2001, une évaluation de *International Biometric Group* sur la répartition de ce marché par technologie montrait que l'approche par empreintes digitales dominait avec près de 50% du chiffre d'affaires total (hors applications judiciaires). La reconnaissance du visage atteignait 15,4% du marché et dépassait depuis peu la reconnaissance de la main avec 10,4%. En 2003 ces chiffres ont quelque peu évolué (Figure 3.1), et la plus forte progression en part de marché concerne la reconnaissance de l'iris qui atteint 7,3% (revenu en 2007 estimé à 210 millions de dollars).

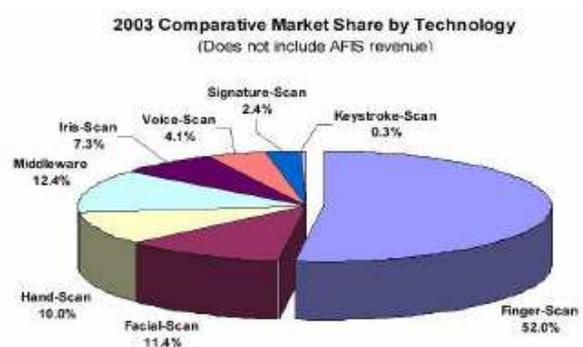


FIG 4.1. Part de marché par technologie biométrique (Evaluation 2003 par International Biometric Group).

On peut diviser les applications de la biométrie selon quatre groupes : service public (contrôle automatique de l'identité des personnes entrantes ou sortantes d'un territoire), application de la loi (l'authentification de criminels par reconnaissance automatique de leur empreintes digitales, l'identification d'enfants kidnappés ou disparus, dont la véritable identité a été masquée), transaction commerciale et bancaire (la vérification des individus par leur iris pour le retrait automatique de l'argent, l'achat d'un bien ou d'un service à distance), accès physique (sécuriser l'accès à un lieu) et logique(sécuriser l'accès à un terminal, serveur et réseau informatique ou de télécommunication).

Les solutions biométriques ont été standardisées par de nombreux comités, associations et organisations dont on peut citer :

- *The National Institute of Standard and Technology, anciennement The National Bureau of Standards* (US: <http://www.nist.gov>) qui a publié un guide d'évaluation pour la sélection de système de vérification
- *L'Union Européenne* (Europe : <http://www.cordis.lu>) pour l'établissement d'une réglementation européenne visant à protéger la vie privée des futurs utilisateurs, à assurer la qualité des techniques employées (cf. Norme EN 50133-1), mais aussi pour promouvoir les technologies biométriques au travers de projets européens (BIOTEST, CASCADE, BANCA, M2VTS).
- *The Association for Biometrics* (UK : <http://www.afb.org.uk>) à l'origine du projet BioVision pour aider à la coordination et au développement de la biométrie en Europe.
- *The Biometrics Consortium* (US : <http://www.biometrics.org>) créé à la demande du département de la défense américaine pour unifier les méthodologies de test des solutions biométriques par les différentes organisations gouvernementales aux Etats-Unis.

- *The Federal Bureau of Investigation* (US : <http://www.fbi.gov>) qui a proposé un standard pour la compression d'images d'empreintes digitales (*Wavelet Scalar Quantization*) et l'échange de données biométriques.
- *The Standards Committee X9* (US : <http://www.x9.org>) accrédité par l'ANSI dont le sous-comité sur la sécurité de l'information est chargé d'établir un standard sur le cryptage des données biométriques transmises lors de transactions financières.

Les systèmes biométriques permettent de reconnaître une personne (ou d'authentifier son identité) dont l'identité a préalablement été enregistrée dans une base de données (de N personnes « autorisées »). Les termes généraux d'authentification ou de reconnaissance couvrent aussi bien l'identification que la vérification. Mais ces deux derniers processus sont bien différents.

L'identification au sens strict du terme suppose un contexte de groupe fermé. Cela signifie que nous sommes sûrs que le client du système biométrique appartient aux N personnes autorisées. Il s'agit de déterminer quelle personne parmi les N correspond le mieux au client, et il n'est donc pas nécessaire de fixer de seuil pour accepter ou rejeter le client. On parle souvent d'identification « 1-parmi-N ».

La vérification au sens strict du terme opère dans un contexte de groupe ouvert. Autrement dit nous ne sommes absolument pas sûrs que l'identité du client est réellement connue par le système. En pratique, le client revendique l'identité d'un des N individus de la base de données. S'il n'est pas reconnu comme tel, c'est qu'il s'agit d'un imposteur. On parle souvent de vérification « 1-pour-1 ».

4.2 Principes de fonctionnement d'un système biométrique

Les systèmes biométriques automatiques mono-modal en mode vérification fonctionnent selon les principes suivants. Dans un système fonctionnel typique, un capteur, adapté à la biométrie observée, génère des données de mesure. Des techniques de traitement du signal ou d'image sont alors employées pour extraire de ces données des caractéristiques. En général, chaque biométrie a sa propre série de caractéristiques. Enfin des techniques de comparaison de motifs quantifient les similitudes entre les caractéristiques de l'utilisateur avec celles stockées dans une base de données de référence sous l'identité requise. L'architecture d'un système mono-modal en mode vérification est décrite dans la figure 4.2. Elle comprend deux principaux modules : (i) le comparateur qui délivre un nombre scalaire (ou score de ressemblance), suivi (ii) d'un module de décision comparable à un opérateur de seuillage.

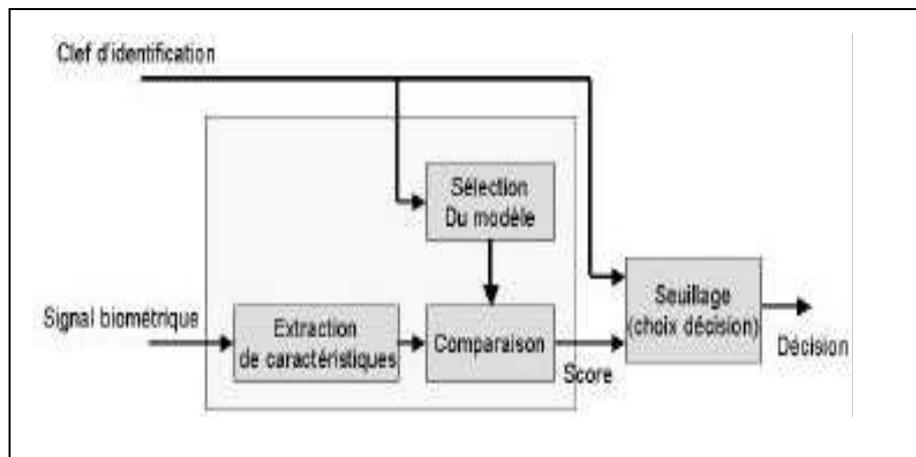


FIG 4.2. Architecture typique d'un système biométrique mono-modal

4.3 Performance d'un système biométrique

Pour mesurer la performance des systèmes biométriques, deux mesures principales furent créées : le taux de faux rejets (TFR) exprime le pourcentage de

personnes autorisées qui sont rejetées par le système qui n'arrive pas à les reconnaître, le taux de fausses acceptations (TFA) donne le pourcentage de personnes non autorisées (imposteurs) qui sont acceptées de façon erronée par le système. La figure 4.3 illustre les distributions des taux de vraisemblance des utilisateurs légitimes et des imposteurs d'un système biométrique en fonction d'une métrique de discrimination (seuil de décision), propre au système.

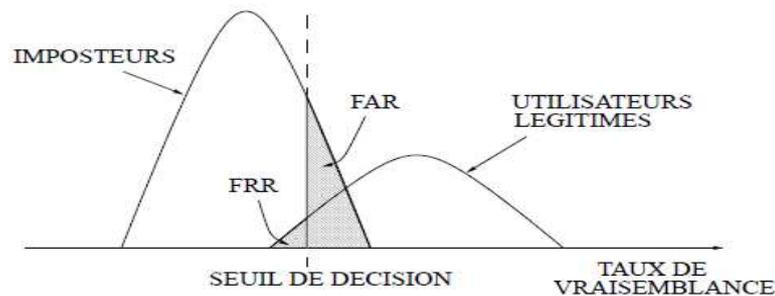


FIG 4.3. Distributions des taux de vraisemblance des utilisateurs légitimes et des imposteurs d'un système biométrique.

Idéalement, le système devrait avoir des FAR et FRR égaux à zéro. Comme ce n'est jamais le cas en pratique, il faut choisir un compromis entre FAR et FRR. Plus le seuil de décision θ est bas, plus le système acceptera d'utilisateurs légitimes mais plus il acceptera aussi d'imposteurs. Inversement, plus le seuil de décision θ est élevé, plus le système rejettera d'imposteurs mais plus il rejettera aussi d'utilisateurs légitimes. Il est donc impossible en faisant varier le seuil de décision de faire diminuer les deux types d'erreurs en même temps. C'est l'une des raisons qui a motivé l'introduction de la multi modalité puisqu'il est possible de diminuer les deux types d'erreur à la fois en combinant correctement plusieurs modalités.

4.4 La reconnaissance par l'iris

4.4.1 Historique et généralités

La reconnaissance de l'iris pour l'identification d'une personne été proposée à l'origine en 1936 par l'ophtalmologue Frank Burch. Dans les années 1980, l'idée réapparut avec les films sur James Bond, mais cela restait toujours de la science-

fiction. En 1987 les ophtalmologues Aran Safir et Léonard Flom ont fait breveter cette idée et en 1989 ils ont demandé à John Daugman (alors enseignant à l'université de Harvard), d'essayer de créer des algorithmes pour l'identification par l'iris. Ces algorithmes (méthode basée sur les ondes de Gabor), que Daugman a fait breveter en 1994, sont la base de tous les systèmes d'identification par l'iris.

- La formation de l'iris pour un œil humain commence pendant le troisième mois de gestation.
- Les structures qui créent éléments distinctifs sont complètes à partir de troisième mois même si la pigmentation parfois peut continuer dans les premières années après la naissance.
- La probabilité de trouver deux iris suffisamment identiques est 1 sur 10 puissance 72 (selon les estimations de Daugmann).
- Deux vrais jumeaux ont assez d'éléments distinctifs permettant de les distinguer l'un de l'autre.

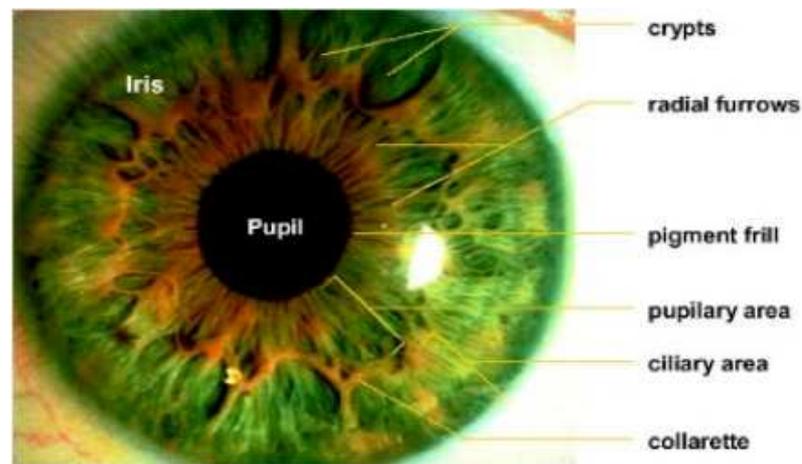


FIG 4.4. Composition d'un œil humain : partie parfaitement annulaire de l'iris dans l'œil.

- L'iris est la zone colorée visible entre le blanc de l'œil et la pupille.
- L'iris est un réseau de tubes fins vus du dessus et dont le diamètre est inférieur à celui d'un cheveu.
- La forme de l'iris c'est-à-dire l'enchevêtrement des tubes est fixe et ne varie que très peu durant la vie de l'individu.
- Par contre, la couleur des tubes varie un peu avec le temps (5 à 10 ans) et pour certaines maladies.

- L'iris n'est pas liée à l'ADN : les deux iris d'un même individu ont à peu près la même couleur mais leur forme (enchevêtrement de tubes) est aussi différente que celle de l'iris d'une autre personne.
- L'iris contient une quantité d'informations particulièrement importante que certains n'hésitent pas à comparer à la quantité d'informations contenues dans l'ADN.
- L'iris n'occupe qu'une surface très faible. L'observation pratique à travers un système optique ne permet de déceler que des contours macroscopiques et pas de descendre au niveau de tubes élémentaires. Toutefois, ceci évolue avec la précision des capteurs. Mais il n'est pas non plus indispensable de recueillir toute l'information contenue dans l'iris pour authentifier un individu. En effet, les iris sont suffisamment variés pour qu'une approximation de l'information totale suffise à certifier l'identité d'un individu.

4.4.2 Schéma général

Un système de vérification par reconnaissance d'iris peut se décomposer en deux unités principales (Figure 4.4) :

- (1) une unité optique de capture de l'image de l'iris (dispositif de vision),
- (2) une unité de traitement des données (extraction et comparaison des informations discriminantes avec celles stockées préalablement lors de l'enrôlement).

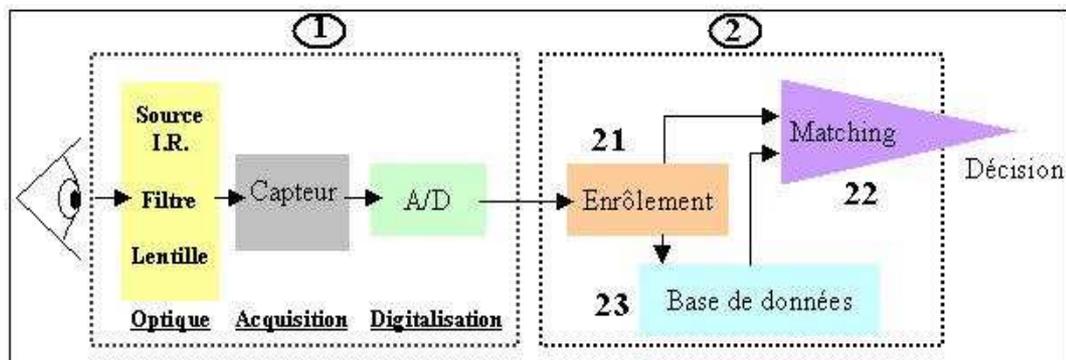


FIG 4.5. Schéma général d'un système de vérification par reconnaissance d'iris.

Par la suite, ce chapitre porte davantage sur la partie algorithmique de traitement des données (2), constituées des processus suivants :

- d'enrôlement (21) par lequel l'identité d'une personne et son image biométrique (image de l'iris) sont utilisées pour constituer une base de données (23).
- de *matching* (22) qui quantifie la ressemblance entre deux gabarits (signatures numériques dont l'une provient de la base de données et l'autre créée au cours de l'enrôlement présent).

4.4.2.1 Système d'acquisition

L'acquisition d'une image d'iris peut donc s'effectuer avec une caméra monochrome. Mais comme nous pouvons le constater sur la figure 3.4-a, du fait de la réflexion spéculaire des sources lumineuses environnantes sur la cornée, la capture d'image d'un iris s'effectue généralement dans le domaine du proche infrarouge (non visible, donc non éblouissant pour l'utilisateur).

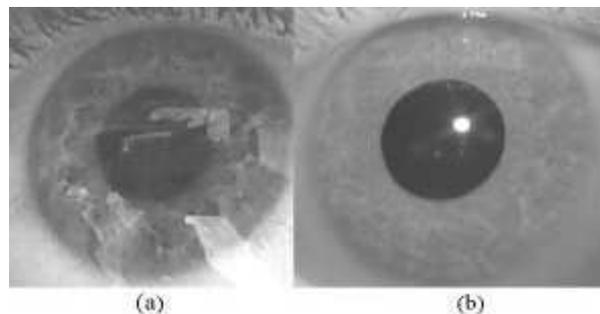


FIG 4.6. Image d'un iris capturé (a) dans le visible, (b) dans le proche infrarouge.

4.4.2.2 Traitement des données

Contrairement à l'empreinte digitale, l'utilisation de la technologie de l'iris en reconnaissance d'individus est relativement récente. Ceci explique sûrement le faible nombre de publications de recherche dans le domaine. Entre 1992 et 2003, on dénombre environ 9 groupes de recherche ayant contribué au développement d'algorithmes de vérification ou d'identification de personnes par reconnaissance de

l'iris. Toutes les méthodes proposées reposent sur le même principe qui se décompose en 4 étapes : (i) localisation de l'iris au sein de l'image, (ii) normalisation des dimensions des données, (iii) extraction de caractéristiques discriminantes, et enfin (iv) comparaison des informations. On constate que l'étape (i) est l'étape la plus importante dans la mesure où le résultat des autres étapes dépend de cette étape. Le reste de ce chapitre est consacré à la description de chacune de ces étapes ainsi que les principales techniques qui lui sont associées.

4.4.3 Processus de traitement

Comme nous l'avons vu, l'iris humain est une région annulaire entre la pupille et la sclérotique. C'est la texture de l'image d'iris, unique pour chaque individu, qui est utilisée comme base de la classification. Deux grandes étapes sont alors opérées.

La première étape consiste à chercher la position de l'iris dans l'image de l'oeil. Elle vise à retourner trois paramètres qui sont les coordonnées x_0, y_0 de la position du centre de l'iris et son rayon r .

La seconde étape consiste à extraire les paramètres caractéristiques de l'iris. L'iris détecté est alors « déroulé » sur une région rectangulaire puis on y effectue une extraction des caractéristiques les plus distinctives présentes dans le modèle d'iris pour qu'une comparaison entre des modèles (gabarits) puisse être faite

4.4.3.1 Localisation de l'iris au sein de l'image

L'algorithme d'extraction de l'anneau (constituant l'iris) consiste à déterminer la position des centres ainsi que les rayons des deux motifs circulaires de la pupille et de l'iris.

- Les opérateurs integro-différentiels :

Un procédé connu de détermination des centres et rayons de la pupille et de l'iris, [Daugman, 1995] et [Sanchez-Reillo et Sanchez-Avila et Martin-Pereda, 1999], considère successivement une grille de points de l'image comme le centre potentiel (x_0, y_0) et mesure les gradients de la somme des intensités des points situés sur des arcs de cercle centrés sur le centre potentiel considéré. Les rayons r de ces arcs de cercle varient dans une plage de rayons possibles d'une pupille et d'un iris, et les centres de la pupille et de l'iris sont alors les points pour lesquels la variation du gradient (d'un arc de cercle à l'autre) est la plus importante, et ce pour tous les points (x_0, y_0) de l'image considérés comme centre potentiel. Il s'agit d'opérateurs integro-différentiels (Eq.4.1)

$$\max_{(r, x_0, y_0)} \left| G_\sigma(r) * \frac{\partial}{\partial r} \oint_{r, x_0, y_0} \frac{I(x, y)}{2\pi} ds \right| \quad (4.1)$$

Où $I(x, y)$ représente l'image de l'œil. $G_\sigma(r)$ une fonction de lissage (type noyau gaussien par exemple) définie par le facteur d'échelle σ , et s et le contour obtenu par r, x_0, y_0 .

Les inconvénients majeurs de cette approche résident (i) dans la répétition successive des mêmes opérations pour affiner la recherche des centres (nécessitant un grand nombre de calculs), et (ii) l'algorithme peut échouer quand le reflet de la diode infrarouge prend place à proximité de la frontière entre l'iris et la pupille et/ou que cette pupille n'est pas parfaitement ronde (figure 4.7).

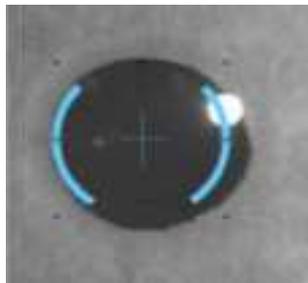


FIG 4.7. Exemple d'erreur de localisation du centre de la pupille par les pérateurs integro-différentiels.

- *La transformée de Hough :*

Un tout autre procédé [Wildes, 1997] [Kong et Zhang, 2001] [Tisse et Martin et Torres Robert, 2002] [Ma et Wang et Tan, 2002] consiste à rechercher des formes géométriques concentriques au moyen d'une transformation de Hough. Il s'agit de localiser approximativement les coordonnées (x_p, y_p) des cercles en considérant successivement chacun des pixels de l'image comme étant sur la périphérie d'un cercle de rayon r donné. D'abord, un détecteur de contours est généré par le calcul des dérivées premières des valeurs d'intensités dans l'image de l'œil, puis pour chaque diamètre possible du cercle recherché, on associe à l'image un accumulateur de mêmes dimensions que l'image. Chaque accumulateur mémorise, pour un rayon donné, le nombre de fois où un point de l'image est déterminé comme le centre potentiel du cercle recherché. Le point de poids maximal pour les différents rayons considérés et l'indice de l'accumulateur dans lequel il se trouve fixent respectivement le centre du cercle recherché et son rayon.

La transformée de Hough pour une frontière circulaire et un ensemble de points $(x_j, y_j) j = 1, \dots, n$ récupérés des contours est définie par :

$$H(x_c, y_c, r) = \sum_{j=1}^n h(x_j, y_j, x_c, y_c, r) \quad (4.2)$$

Avec :

$$h(x_j, y_j, x_c, y_c, r) = \begin{cases} 1 & \text{si } g(x_j, y_j, x_c, y_c, r) = 0 \\ 0 & \text{sinon} \end{cases} \quad (4.3)$$

Et :

$$g(x_j, y_j, x_c, y_c, r) = (x_j - x_c)^2 + (y_j - y_c)^2 - r^2 \quad (4.4)$$

L'inconvénient majeur de cette méthode est qu'elle impose le recours à une mémoire encombrante du fait qu'il faille stocker plusieurs matrices de la taille de l'image à traiter et peut ne pas convenir ainsi aux applications en temps réel.

- *Les contours actifs (snakes) :*

Les contours actifs ont été utilisés pour localiser l'iris [Ritter, 1999], [Ritter et Cooper, 2003]. Le contour est défini comme un ensemble de sommets connectés d'une courbe fermée, qu'on va tenter de déplacer pour leur faire épouser une forme. On utilise les notions d'énergies internes et externes pour caractériser respectivement la forme du contour et tous les éléments qui lui sont propres, et le positionnement du contour sur l'image. Le déplacement du contour est causé par les énergies internes et externes agissant sur les sommets jusqu'à ce que l'équilibre soit atteint. Chaque sommet est déplacé entre t et $t+1$ par :

$$v_i(t+1) = v_i(t) + F_i(t) + G_i(t) \quad (4.5)$$

Où F_i est l'énergie interne, G_i est l'énergie externe et v_i est la position du sommet i .

Les contours actifs s'avèrent très pertinents pour isoler des formes convexes régulières et idéales. Cependant, le positionnement des paramètres des différents éléments des énergies reste un point très délicat à prévoir, surtout si on commence à tenir compte de beaucoup d'aspects de chaque énergie. De plus, ces valeurs sont liées à la forme à extraire.

4.4.3.2 Normalisation

Le passage dans un système pseudo-polaire sans dimension de l'image de la texture de l'iris a été proposé par Daugman [Daugman, 1993] pour les objectifs (i) de s'affranchir des déformations induites par les constriction pupillaires et (ii) de convertir l'image de l'iris en un équivalent rectangulaire de taille fixe (tout en tenant compte du fait que la pupille et l'iris ne sont pas concentriques). Ceci est en quelque sorte effectué en convertissant les coordonnées cartésiennes de l'iris en coordonnées polaires, au moyen d'une conversion polaire à angle constant. Les pixels de ce nouveau repère sont décrits par le paramètre (ρ, θ) où ρ est dans $[0,1]$ et θ est l'angle $[0, 2\pi]$.

Soient les coordonnées (x_{p0}, y_{p0}) du centre de la pupille, (x_{i0}, y_{i0}) du centre de l'iris, (x_p, y_p) d'un point du contour de la pupille, (x_i, y_i) d'un point du contour de l'iris, (x, y) d'un point appartenant à la surface iridienne, et r_p la valeur du rayon de la pupille et r_i la valeur du rayon de l'iris. Le repère pseudo polaire sans dimension est représenté sur la figure 4.8.

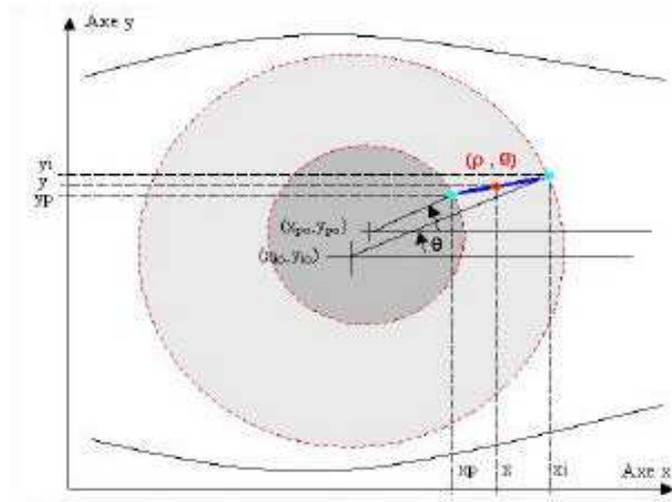


FIG 4.8. Repère pseudo polaire sans dimension

Ainsi, L'iris localisé est déroulé suivant une transformation polaire « $I(x(\rho, \theta), y(\rho, \theta)) \rightarrow I(\rho, \theta)$ » (exemple sur la figure 4.9) qui se traduit par les relations suivantes :

$$x(\rho, \theta) = (1 - \rho)x_p(\theta) + \rho x_i \quad (4.6)$$

$$y(\rho, \theta) = (1 - \rho)y_p(\theta) + \rho y_i \quad (4.7)$$

Avec :

$$x_p(\theta) = x_{p0}(\theta) + r_p \cos(\theta) \quad (4.8)$$

$$y_p(\theta) = y_{p0}(\theta) + r_p \sin(\theta) \quad (4.9)$$

$$x_i(\theta) = x_{i0}(\theta) + r_i \cos(\theta) \quad (4.10)$$

$$y_i(\theta) = y_{i0}(\theta) + r_i \sin(\theta) \quad (4.11)$$

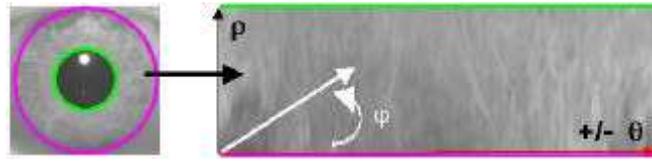


FIG 4.9. Exemple de transformation Cartésienne Polaire

4.4.3.3 Extraction de l'information

Afin de fournir l'identification précise des personnes, l'information la plus distinctive présentée dans un modèle d'iris doit être extraite. Les ondelettes peuvent être employées pour extraire les données dans la région d'iris.

Daugman a utilisé le filtre de Gabor 2 dimensions (2D) [Daugman, 1994] afin de coder le modèle d'iris, il démodule la sortie du filtre afin de compresser les données. Ceci est fait par quantification de l'information de phase dans quatre niveaux. Il a été montré par Oppenheim et Lim [Oppenheim et Lim, 1981] que l'information de phase, plutôt que celle de l'amplitude fournit les informations les plus significatives dans une image. La démodulation et le processus de quantification de phase peuvent être représentés comme :

$$h_{\{\text{Re}, \text{Im}\}} = \text{sgn}_{\{\text{Re}, \text{Im}\}} \int_{\rho} \int_{\phi} I(\rho, \phi) e^{-i\nu(\theta_0 - \phi)} e^{-(r_0 - \rho)^2 / \alpha^2} e^{-(\theta_0 - \phi)^2 / \beta^2} \rho d\rho d\phi \quad (4.12)$$

Avec (r_0, θ_0) indique la fréquence centrale du filtre, (α, β) indiquent la largeur et la longueur efficaces. $I(\rho, \phi)$ représente l'image en coordonnées polaires sans dimension et h peut être considéré comme un bit complexe dont la partie réelle et imaginaire dépend du signe de l'intégrale 2D. Ainsi chaque phase de Gabor sera codée en deux bits selon le principe de codage quatre quadrants illustré sur la figure 4.10.

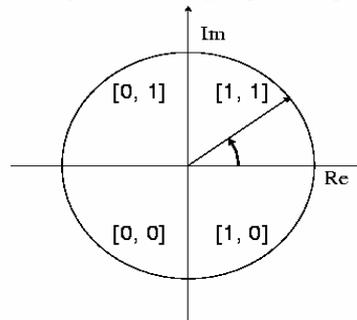


FIG 4.10. Le principe de codage de phase sur quatre quadrants et en deux bits

4.4.3.4 Assortiment

L'essai d'unicité des modèles d'iris est important. L'unicité a été déterminée en comparant des calibres produits à partir des yeux différents, et après on examine la distribution des valeurs des distances de Hamming. La distance de Hamming est définie comme la somme des bits en désaccord [Daugman, 1994] (somme de Ou - exclusif entre les deux configurations binaires).

4.5 Conclusion

L'iris humain est un attribut biométrique particulièrement discriminant entre individus. Le domaine de la reconnaissance des personnes par l'iris est dominé aujourd'hui par la méthode proposée et développée par le professeur Daugman.

Un procédé d'identification de personne en utilisant l'iris se décompose en quatre modules: le module d'isolation d'iris, le module de normalisation, le module d'extraction et génération d'iris code et enfin le module de comparaison des calibres biométriques.

Nous avons exposé dans ce chapitre plusieurs méthodes de segmentation (localisation) de l'iris, ainsi que les limites de chaque méthode. Dans le chapitre 6, nous allons appliquer une approche bio-inspirée à base de fourmis artificielles dans le problème de la segmentation d'iris.

Utilisation des fourmis artificielles dans un système de vérification biométrique par reconnaissance de l'iris

Comme il a été décrit dans le chapitre précédent, l'iris est une modalité biométrique performante. Cependant, la biométrie de l'iris présente plusieurs difficultés et défis que nous devons surmonter pour définir un système de reconnaissance fiable basé sur cette modalité. Parmi ces défis, le système doit réussir les tests d'imposture et de convivialité envers les utilisateurs. Dans cet objectif, ce chapitre présente l'utilisation d'un algorithme à base de fourmis artificielles inspirant de système de reconnaissance chimique des fourmis dans un système de vérification biométrique par reconnaissance de l'iris lors de processus de matching.

5.1 Problématique

Un système biométrique a pour résultat un score qui peut être considéré comme une mesure de similarité ou de dissimilarité entre les échantillons comparés. Dans le cas d'un score de similarité, plus la valeur est grande, plus la probabilité que l'échantillon test provient de la même personne que l'échantillon de référence est

grande. Un exemple d'un score de similarité est la mesure de corrélation entre deux échantillons. Le score de dissimilarité au contraire, indique le degré d'éloignement de l'échantillon test par rapport à l'échantillon de référence. Par conséquent, plus il tend vers zéro plus la probabilité que les deux échantillons comparés proviennent de la même personne est grande.

Le calcul de score s'effectue au moyen du calcul de la distance de Hamming qui est le rapport des bits qui sont différents et fiables d'un code d'iris à l'autre sur le nombre total des bits fiables, et le résultat sera donc compris entre 0 et 1.

Nous appellerons score 'intra-classe' ou distributions 'intra-classe', les scores résultant de comparaisons entre des gabarits biométriques et des échantillons de test provenant d'une même personne et des scores 'inter-classe' ou distributions 'inter-classe', des comparaisons entre deux échantillons provenant de deux personnes différentes.

L'objectif d'un système de reconnaissance de l'iris est de pouvoir réaliser une séparation distincte entre les distributions intra-classe et inter-classe. Ceci est effectué en fixant un seuil de décision par rapport au score calculé. Le choix d'un tel seuil est une problématique. En effet, le système doit avoir des TFR (taux de faux rejets) et TFA (taux de fausses acceptations) égaux à zéro. Comme ce n'est jamais le cas en pratique, il faut choisir un compromis entre TFR et TFA. Plus le seuil de décision est bas, plus le système acceptera d'utilisateurs légitimes, mais plus il acceptera aussi d'imposteurs. Inversement, plus le seuil de décision est élevé, plus le système rejettera d'imposteurs mais plus il rejettera aussi d'utilisateurs légitimes. Ces deux cas vont engendrer un chevauchement des deux distributions intra-classe et inter-classe. Ce chevauchement peut être traduit par le taux d'erreur du système. Donc le choix du seuil reste une problématique et dépendra du degré de sécurité souhaité pour l'application, car il est impossible en faisant varier le seuil de décision de faire diminuer les deux types d'erreurs en même temps.

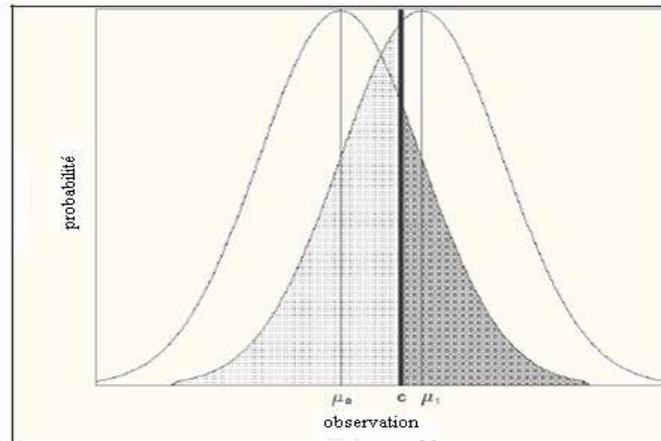


FIG 5.1. Courbes des variations des scores intra-classe et inter-classe pour un Système biométrique

5.2 Motivations

Dans le chapitre 1, nous avons décrit une tentative d'algorithmes de fourmis artificielles, dont nous nous intéressons ici à l'algorithme AntClust inspiré du système de reconnaissance chimique des fourmis. Rappelons que celui-ci connu sous le nom de fermeture coloniale, permet aux fourmis de privilégier les relations avec les membres de leur nid et de rejeter les intrus qui peuvent être de la même espèce. Donc, on peut bien constater que le système de reconnaissance chimique des fourmis et le système de vérification biométrique ont un objectif commun. En effet, le système de reconnaissance chimique des fourmis vise à accepter les fourmis appartenant au nid et à rejeter les intrus, la même chose pour le système de vérification biométrique qui vise aussi à accepter les utilisateurs légitimes et à rejeter les imposteurs. La raison qui nous a motivée à transposer le modèle du système de reconnaissance chimique des fourmis dans un système de vérification biométrique par reconnaissance de l'iris.

5.3 Etude expérimentale

Pour évaluer l'utilisation de l'algorithme AntClust inspiré du système de reconnaissance chimique des fourmis dans un système de vérification biométrique en terme de performance, nous avons utilisé un système de référence développé par Libor Masek de l'université de Western Australia.

Le système Masek est un système 'Open -Source' de reconnaissance des personnes par l'iris [Masek et al, 2003]. Le système inclue un module de segmentation basé sur la transformée de Hough qui permet de localiser l'iris. Le système est composé aussi d'un module de normalisation basé sur la méthode de normalisation pseudo polaire. Un troisième module du système est celui de la reconnaissance où un filtrage 1D de Log-Gabor est effectué sur 4 niveaux pour coder la phase de Gabor selon le procédé de codage 4 quadrants. La distance de Hamming est finalement employée dans le dernier module pour la prise de décision.

En utilisant ce système, nous allons intégrer les fourmis artificielles en tant qu'un module de classification lors du processus de matching. Autrement, nous allons employer l'algorithme AntClust inspiré du système de reconnaissance chimique des fourmis dans le dernier module du système Masek à la place de la distance de Hamming.

5.3.1 Ensemble des données

Nous avons utilisé la base d'images d'œils de l'Académie Chinoise des Sciences – Institut d'Automation CASIAv1 [CASIA, 2003]. Cette base inclue 756 images de 108 personnes. Pour chaque personne, 7 images ont été acquises en deux sessions séparées de quelques semaines. La résolution des images de CASIAv1 est de 320*280. Les images ont été capturées particulièrement pour la recherche en reconnaissance de l'iris.

Nous n'allons pas utiliser cette base dans sa totalité. Cependant, nous utiliserons 504 images de 72 personnes.

5.3.2 Etude des distributions des distances intra-classe et inter-classe pour quelques images de la base CASIAv1

Avant d'entamer la mise en œuvre de l'algorithme AntClust, nous allons analyser la distribution des distances intra-classe et inter-classe entre quelques images de la base CASIAv1. Ceci permet de bien voir la problématique définie au début de chapitre. Le tableau 5.1 illustre les distances de Hamming entre ces images.

Pour bien comprendre le tableau 5.1, certaines informations sont nécessaires : tout d'abord, le tableau représente la distance de Hamming entre 28 images choisies arbitrairement de la base CASIAv1. Ces images sont désignées suivant le format xxx_s_y où :

- xxx : indique l'identificateur unique de l'œil, varie de 1 à 4.
- s : indique le numéro de la session. 1 dénote la première session et 2 dénote la deuxième session.
- y : indique l'index de l'image dans la même session. Il varie de 1 à 3 dans la première session et de 1 à 4 dans la deuxième session.

Donc en suivant ce format, on a 4 yeux de personnes différentes sur les 28 images représentées dans le tableau 5.1. Nous allons considérer les images de la session 2 comme des images de référence et les images de la session 1 comme des images de test. Les distances mesurées entre les images d'oeils appartenant à la même personne représentent les distances intra-classe, et les distances mesurées entre les images appartenant aux personnes différentes représentent des distances inter-classe.

Images	001_1_1	001_2_2	001_2_3	001_2_4	002_2_1	002_2_2	002_2_3	002_2_4	003_2_1	003_2_2	003_2_3	003_2_4	004_2_1	004_2_2	004_2_3	004_2_4
001_1_1	0.32	0.23	0.25	0.27	0.47	0.46	0.45	0.46	0.43	0.43	0.43	0.44	0.50	0.53	0.53	0.49
001_1_2	0.30	0.28	0.28	0.26	0.48	0.44	0.44	0.44	0.49	0.48	0.47	0.46	0.51	0.52	0.52	0.50
001_1_3	0.34	0.33	0.27	0.28	0.47	0.47	0.44	0.46	0.47	0.48	0.48	0.48	0.50	0.50	0.54	0.48
002_1_1	0.49	0.47	0.47	0.47	0.37	0.45	0.39	0.46	0.48	0.49	0.50	0.48	0.52	0.51	0.51	0.50
002_1_2	0.48	0.47	0.44	0.45	0.51	0.26	0.43	0.27	0.49	0.48	0.47	0.45	0.51	0.52	0.53	0.49
002_1_3	0.48	0.47	0.47	0.49	0.48	0.30	0.42	0.31	0.48	0.47	0.47	0.48	0.55	0.52	0.51	0.49
003_1_1	0.47	0.44	0.47	0.45	0.47	0.47	0.44	0.45	0.26	0.24	0.23	0.26	0.48	0.52	0.52	0.50
003_1_2	0.44	0.42	0.46	0.46	0.49	0.51	0.47	0.49	0.16	0.21	0.28	0.27	0.47	0.49	0.50	0.49
003_1_3	0.44	0.43	0.47	0.47	0.50	0.52	0.48	0.50	0.19	0.20	0.27	0.28	0.48	0.49	0.51	0.48
004_1_1	0.49	0.51	0.49	0.49	0.49	0.52	0.51	0.52	0.47	0.48	0.50	0.50	0.34	0.43	0.40	0.35
004_1_2	0.51	0.51	0.51	0.51	0.50	0.50	0.51	0.50	0.51	0.50	0.52	0.50	0.37	0.45	0.43	0.33
004_1_3	0.50	0.49	0.49	0.48	0.50	0.52	0.49	0.49	0.46	0.47	0.48	0.48	0.35	0.47	0.44	0.36

Tab 5.1. Représentation des distances de Hamming entre quelques images de la base CASIAv1

Les résultats du tableau 5.1 peuvent être mieux interprétés par des courbes représentées dans la figure 5.2

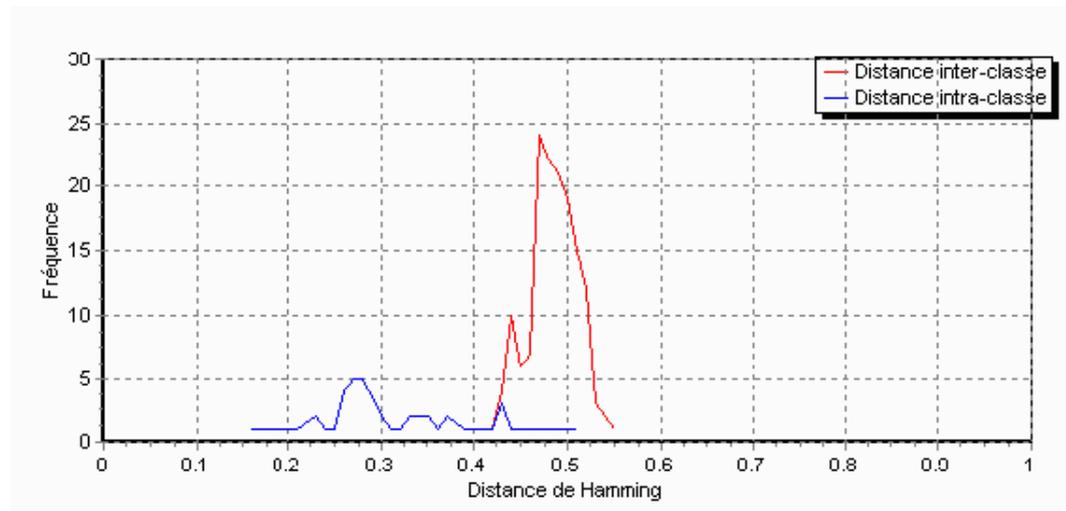


FIG 5.2. Courbes intra-classe et inter-classe pour les images du tableau 5.1.

Pour déterminer si deux images appartiennent à une même personne, il faut fixer un seuil. Pour cela, nous avons essayé plusieurs seuils.

En biométrie, évaluer des résultats revient à évaluer le taux de fausses acceptations TFA, et le taux de faux rejets TFR. Les résultats du TFA et TFR obtenus pour chaque seuil sont illustrés dans le tableau 5.2.

		TFA	TFR
Seuil1	0.4	0 %	30.55 %
Seuil2	0.5	46.90 %	2.77 %
Seuil3	0.43	0.56 %	30.55 %

TAB 5.2. Les taux de TFA et TFR calculés pour chaque seuil

A travers les résultats illustrés dans le tableau 5.2, nous remarquons que le fait de choisir plusieurs seuils pour un nombre très petit d'images, ne permet pas néanmoins de faire diminuer le TFA et le TFR en même temps. Ceci est dû au chevauchement des courbes intra-classe et inter-classe illustré dans la figure 5.2.

5.3.3 Mise en œuvre de l'algorithme AntClust

Nous avons utilisé 504 codes biométriques (gabarits) de 72 personnes de la base CASIA. Ces codes ont été extraits à travers le système Masek (décrit plus haut). Pour chaque personne sont associés 7 codes biométriques divisés en 2 classes : une classe de référence pour l'apprentissage contient 4 codes biométriques de référence, et l'autre pour le test contient 3 codes biométriques de test. Sachant que ces codes sont représentés par des matrices binaires de 20 lignes et 480 colonnes. Donc nous disposons de deux bases, une base de référence (constituée de classes de référence) et une base de test (constituée de classes de test).

En ce qui concerne le modèle des fourmis artificielle : Chaque classe de référence va être représentée par un nid (le Label). Chaque code biométrique va être représenté par une fourmi artificielle, et plus précisément par son génome. Chacune des fourmis des nids va tenter de faire des rencontres avec les fourmis appartenant à son nid afin d'apprendre le seuil d'acceptation (le Template) qui correspond à son nid. L'apprentissage de seuil s'appuie sur l'estimation de chaque fourmi des similarités maximales et moyennes lors des rencontres avec les fourmis de son nid (ses congénères). Ce qui permet à chaque fourmi d'avoir une image de son nid. Ensuite les fourmis appartenant aux nids de référence vont réaliser des rencontres avec les fourmis appartenant aux nids de test pour reconnaître qui fait partie du groupe et qui doit être rejeté. D'une autre façon : chaque classe de test doit être comparée aux classes de référence pour déterminer après chaque comparaison si elle sera acceptée correctement ou rejetée correctement par l'algorithme. Ainsi l'algorithme doit faire 5184 comparaisons inter-classe et 72 comparaisons intra-classe.

5.4 Résultats

Nous allons présenter les résultats de l'application de l'algorithme sur le même ensemble d'images représenté dans le tableau 5.1. Puis sur l'ensemble de toutes les images utilisées.

	TFA	TFR
Résultat sur les images du tableau 5.1	2.27 %	0 %
Résultat sur toutes les images utilisées	9.25 %	2.77 %

TAB 5.3. Résultats de l'algorithme AntClust.

5.5 Conclusion

Ce chapitre propose l'utilisation d'un nouvel algorithme de classification, inspiré de système de reconnaissance chimique des fourmis dans un système de vérification biométrique par reconnaissance de l'iris. L'intérêt d'introduire cet algorithme était de réaliser un processus de matching avec un assortiment précis. La capacité des fourmis à reconnaître les membres appartenant à leur colonie et à rejeter les intrus grâce à leur système de reconnaissance chimique nous a fait penser que ce dernier serait adapté à la vérification biométrique qui vise aussi accepter les membres appartenant à la base de données et à rejeter les imposteurs. L'algorithme AntClust a donné des résultats encourageants et qui s'avèrent très compétitifs par rapport aux résultats obtenus par l'emploi de la distance de Hamming.

Chapitre 6

Segmentation de l'iris par les fourmis artificielles

*D*ans ce chapitre, nous présentons un autre type d'utilisation des fourmis artificielles dans un système de reconnaissance biométrique basé sur l'iris. Plus précisément, nous proposons une nouvelle méthode inspirée du comportement de fourragement des fourmis et de leur capacité à s'auto-organiser pour trouver le plus court chemin entre le nid et une source de nourriture, il s'agit de transporter le concept ACO à la résolution collective du problème de segmentation d'iris selon une approche des systèmes multi agents réactifs.

6.1 Introduction

La segmentation ou la localisation de l'iris constitue une étape initiale et fondamentale dans la reconnaissance de l'iris, elle consiste à détecter les contours interne et externe de l'iris dans l'image de l'œil. La modélisation se résulte par un cercle pour les contours internes et une ellipse pour les contours externes. L'algorithme de localisation de l'iris doit déterminer la position des centres ainsi que les rayons des deux motifs circulaires de la pupille et de l'iris. Ceci est très important,

car une région d'iris précise et correcte est nécessaire pour produire des calibres permettant un assortiment (matching) précis. Dans ce chapitre, nous allons proposer une nouvelle méthode inspirée de comportement de fourragement des fourmis et de leur capacité à s'auto-organiser pour trouver le plus court chemin entre le nid et une source de nourriture. Cette méthode repose sur la transposition de ACO à la résolution collective du problème de segmentation d'iris selon une approche multi agents réactifs.

Ce chapitre a pour objet dans une première partie de présenter les motivations qui nous ont amenées à utiliser la métaphore des fourmis pour la résolution de problème de localisation de l'iris. Dans un second point, nous décrivons les principes de base de fonctionnement de notre système. Nous mettons en évidence comment les fourmis arrivent à détecter facilement les contours internes et externes de l'iris. Enfin, après avoir localisé la région d'iris, une dernière partie s'attachera à évaluer l'utilisation d'une telle approche dans le problème de localisation de l'iris.

6.2 Motivations

Dans notre cas, la localisation de l'iris s'effectue à partir d'une image de contours générée à partir de l'image de l'oeil, la solution revient alors à détecter les contours internes et externes de l'iris représentant respectivement la pupille et l'iris. Autrement dit, il faut détecter les pixels contours appartenant à l'un des motifs circulaires de la pupille ou de l'iris ce qui permet de considérer le problème de la segmentation de l'iris comme un problème de classification.

Comme nous l'avons décrit dans le chapitre 4, plusieurs méthodes ont été proposées pour localiser l'iris au sein de l'image d'oeil. Cependant, chacune de ces méthodes avait des inconvénients qui rendent son utilisation compliquée et par conséquent moins efficace (le recours à une mémoire encombrante, définition des paramètres, un grand nombre de calcul,..) notamment quand il s'agit d'une application à temps réelle comme dans les aéroports par exemple. Nous avons constaté que la source d'inspiration que constituent les modèles biologiques de fourmis a de plus en plus de succès dans plusieurs domaines, notamment la

métaheuristique ACO qui a été adaptée aussi pour les problèmes de la classification [Jayaraman et al, 2003].

En bref, le problème de la segmentation de l'iris en tant qu'un problème de classification peut être un bon candidat pour les ACO qui sont des algorithmes stochastiques construisant des solutions d'une manière incrémentale. L'idée de base est de représenter le problème de la segmentation de l'iris sous la forme d'un problème d'une recherche stochastique des pixels appartenant aux contours des cercles et de simuler le comportement de fourragement des fourmis lors de la recherche de plus court chemin. Les fourmis artificielles se déplacent sur l'image des contours à la recherche des pixels appartenant au contour d'un cercle afin de détecter ce dernier. Les fourmis communiquent indirectement en changeant les propriétés de leur environnement par le dépôt des traces de phéromones sur leur chemin, cette trace attire les autres fourmis dans une boucle de rétroaction positive pour faire émerger la solution recherchée correspondant à la détection des deux motifs circulaire de la pupille et de l'iris dans l'image de l'œil.

Dans ce problème, nous avons choisi le paradigme ACO basé sur l'intelligence artificielle distribuée car ce domaine utilise une approche locale de résolution. C'est à dire une approche où les décisions ne sont pas prises par un observateur global qui possède la synthèse du système et qui en manipule les composants, mais par chacun des composants. Ces composants, appelés agents sont très simples et peut intelligents pris individuellement, ils n'ont qu'une vision partielle de l'univers dans lequel ils évoluent et avec lequel ils interagissent. Chaque agent possède un cycle d'exécution durant le quel il commence par percevoir son environnement, ensuite, en fonction des informations provenant de l'environnement, il prend une ou plusieurs décisions. Chaque décision modifie l'état interne de l'agent, son comportement ainsi que l'environnement qui l'entoure orientant ainsi le comportement des autres agents. Donc, on remarque bien que ces définitions s'appliquent bien à la modélisation d'une fourmi par un agent réactif ou d'une colonie de fourmis par un système multi agents.

6.3 La métaheuristique « optimisation par colonie de fourmis »

Dans ACO, le voisinage du nid des fourmis est associé à l'espace des solutions, telle que chaque solution s'apparente à une source de nourriture dont la qualité est fournie par une fonction objectif. Chaque fourmi est assimilée à un processus stochastique de construction de solutions itératif. La construction est guidée par une phéromone artificielle et une information heuristique propre au problème à résoudre. Les traces de phéromones représentent la mémoire des fourmis sur les éléments de la solution, et sont régulièrement mise à jour par les processus de construction (en fonction de la qualité de la solution) et par un mécanisme simulant l'évaporation de la phéromone.

D'une manière générale, l'utilisation de ACO nécessite de choisir une représentation formelle du problème d'optimisation à traiter et de définir le processus de construction de solutions par les fourmis en utilisant cette représentation.

6.3.1 Représentation du problème

Un problème d'optimisation $O = \{S, f, \Omega\}$ est représenté par un ensemble de solutions S , une fonction objective f assignant une qualité à une solution et un ensemble de contraintes Ω . L'objectif est de trouver une solution optimale $f_{\min}(s)$, $\forall s \in S$ qui satisfait les contraintes Ω . Suivant cette représentation, les fourmis construisent les solutions possibles en se déplaçant sur un graphe $G = (C, L)$ associé au problème d'optimisation, tel que les noeuds sont les composantes C du problème qui forment les solutions et les arcs sont les transitions possibles L qui connectent les composantes de C , telle que à chaque transition on associe une fonction de coût. Les contraintes du problème sont implémentées directement dans les règles de déplacement des fourmis. Par conséquent, la solution optimale représente le chemin de coût minimum.

6.3.2 Construction des solutions

Les fourmis artificielles construisent les solutions en ajoutant selon un processus stochastique et itératif les séquences de composants à partir de C . Chaque fourmi construit une solution possible (même de mauvaise qualité) d'une manière stochastique en se déplaçant dans le graphe $G = (C, L)$. La solution optimale émergera de la coopération entre les fourmis de la colonie. Chaque connexion (arc) $(c_i, c_j) \in L$ peut être pondérée par une valeur de phéromone artificielle τ_{ij} représentant une mémoire collective de la colonie de fourmis et permettant les interactions entre les fourmis (stigmergie), et si elle existe une valeur heuristique η_{ij} qui représente une information a priori propre au problème à résoudre. Pour certains problèmes, les traces de phéromones et la valeur de l'heuristique peuvent aussi être associées aux composants du graphe. A chaque fourmi k on associe un état de départ (noeud de départ du graphe) et des critères de terminaison e^k , en plus d'une mémoire T_k qui stocke les informations sur les composants qu'elle a ajouté à sa solution. Cette mémoire peut être utilisée pour :

- Evaluer la qualité de la solution trouvée.
- Parcourir le chemin à l'envers pour faciliter la mise à jour des traces de phéromones une fois la solution générée.
- Empêcher la fourmi de boucles sur un même noeud.

La construction de la solution par une fourmi k se fait d'une manière incrémentale en ajoutant à chaque déplacement un nouveau composant. Le processus de construction se termine pour la fourmi k si au moins un des critères de terminaison e^k est vérifié. Quand la fourmi k est sur un noeud C_i , elle choisit un prochain noeud C_j parmi les noeuds possibles dans un voisinage accessible N_i^k . Ce déplacement est choisi en appliquant une règle de décision probabiliste. Cette règle est en fonction :

- Des informations locales qui sont fonctions des traces de phéromones et de la fonction heuristique.
- De la mémoire de la fourmi T_k , indiquant son historique.
- Des informations locales spécifiques au problème traité.

Au cours du processus, les fourmis mettent à jour la trace de phéromone des transitions (ou composant) qu'elles ont utilisé. Pour cela Plusieurs schémas de mise à jour de la phéromone ont été proposés dans la littérature [Dorigo et al, 1997, Stutzle et al, 1997, Dorigo, 1996]. Ils consistent essentiellement en deux parties : l'évaporation et l'amplification. Pendant la phase d'évaporation, toutes les traces de phéromone sont diminuées suivant une valeur donnée (pourcentage, probabilité...), alors que pendant la phase d'amplification, les meilleures traces de phéromone sont amplifiées, de la même façon que dans la nature, la phéromone s'évapore et les fourmis entretiennent la concentration des meilleures traces en empruntant le même chemin un grand nombre de fois.

6.4 Transposition de ACO à la segmentation d'iris

Nous expliquons maintenant, comment la métaheuristique ACO a été exploitée et adaptée pour la détection de la région d'iris au sein de l'image de l'œil.

Un algorithme basé sur ACO peut être résumé en trois étapes :

- i. Génération des solutions par les fourmis en fonction des propriétés locales et de la phéromone.
- ii. Utilisation d'une optimisation locale pour améliorer les résultats fournis par les fourmis.
- iii. Mise à jour de la phéromone.

En ce qui concerne le point (iii), plusieurs schémas de mise à jour de la phéromone ont été proposés dans la littérature [Dorigo, 1997, Stutzle, 1997, Dorigo, 1996].

Un modèle général à base d'agents réactifs est composé de :

- i. Un environnement invariant E_I , complexe constitué d'un nombre fini de composantes $E_I = E_1, E_2, E_p$.
- ii. Un environnement modifiable E_M , ensemble de fonctions définies sur les composantes de E_I dont les valeurs sont modifiables par les agents.
- iii. Un ensemble X d'agents dont les caractéristiques correspondent au schéma "stimulus-réponse" :
 - Fonctions de perception : renseignements pris dans un voisinage de l'agent (V partie de E_I et restrictions de E_M à V), sur l'environnement et la société d'agents, ce qui inclut l'état propre de l'agent.
 - Fonction d'action: permet à l'agent de modifier l'environnement qui inclut ses propres caractéristiques et celles d'autres agents.
- iv. Espace temps T discret : $\forall t \in T$, il \exists 1 état de E_M et de X déterminé pour des valeurs en t de leurs fonctions.

6.4.1 Modélisation du problème (environnement et comportement)

Le problème consiste à mettre en oeuvre un système collectif, décentralisé fondé sur une population d'agents fourmis qui sont autonomes et capables de s'auto-organiser pour la résolution collective de problème de détection des contours internes et externes de l'iris dans une image de contours. D'une autre façon, il s'agit d'appliquer un algorithme à colonie de fourmis à la détection des cercles de la pupille et de l'iris dans une image de l'œil après avoir détecté les contours de cette dernière.

Les fourmis naturelles vivent en colonies et sont capables de distinguer leur propre trace de phéromone et celles des autres membres de leur colonie. Dans le cas de notre algorithme, nous considérons que la population est divisée en deux colonies que nous nommons « équipes », chaque équipe partage le même type de phéromone et n'est pas attirée par les traces des individus d'autres équipes, l'intérêt d'utiliser plusieurs colonies sera expliqué plus loin dans ce chapitre, notons qu'un tel procédé a

été déjà utilisé dans une application d'un algorithme à colonie de fourmis à la détection des structures rétentives en eaux côtières [Segond, 2004]. L'environnement dans lequel évoluent les fourmis est un espace à deux dimensions formé par l'image de contours générée par l'algorithme de Canny, cette image est partitionnée en des cellules dont chacune correspond à un pixel.

Chaque fourmi dispose d'une perception locale (son voisinage dans l'image), et se déplace d'une cellule à l'autre selon une connectivité à huit voisins. Grâce à des comportements simples, les fourmis sont capables de percevoir des contours locaux dans l'image tout en retrouvant la continuité de ces contours lors de leurs déplacements. Chaque fourmi possède un cycle d'exécution pendant lequel elle se comporte au départ d'une manière aléatoire car aucune information n'est mise à sa disposition, ensuite, en fonction des informations provenant de l'environnement et stockées dans les cellules (présence d'un contour, quantité de phéromone), elle prend des décisions lui permettant de déterminer ses mouvements (continuer dans cette direction à son prochain déplacement, générer une nouvelle direction) ainsi que de construire un élément de la solution.

Les fourmis interagissent sur l'image et communiquent en changeant les propriétés locales de leur environnement, leurs actions ont pour objectif le dépôt et le renforcement des traces de phéromones permettant une détection fiable des cercles de la pupille et de l'iris. La communication entre les fourmis et la coordination de leurs actions sont basées sur un processus stigmergique.

6.5 Données utilisées

Pour tester l'efficacité de notre méthode, nous allons utiliser la base de données CASIA v1, celle-ci inclue 756 images de 108 personnes. Pour chaque personne, 7 images ont été acquises en deux sessions séparées de quelques semaines. La résolution des images de CASIAv1 est de 320*280.

6.6 La segmentation de l'iris

Le processus d'extraction de la région d'iris comporte deux applications successives de l'algorithme de colonies de fourmis sur l'image des contours, la première vise à détecter le contour de la pupille, la seconde concerne la détection de contour de l'iris. Chaque application comporte également deux phases, dont chacune vise à extraire une partie de la région d'iris (figure 6.1).

Rappelons que la population de fourmis est divisée en deux équipes (ceci été mentionné dans la section 6.4.1), ce qui permet à chaque équipe de se focaliser sur l'extraction d'une partie en particulier.

L'intérêt de diviser la tâche entre les fourmis est qu'il facilite la détection des cercles tout en diminuant les calculs et par conséquent le temps nécessaire à la réalisation de la tâche, car la taille des images de la base de données utilisé est relativement importante, en outre, ceci permet de palier aux perturbations éventuelles dues à la présence de paupière ou de cils.

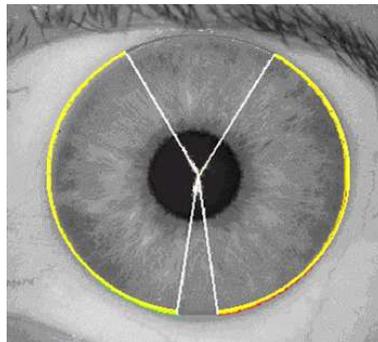


FIG 6.1. Principe de l'algorithme de localisation de l'iris.

6.6.1 Détection de contours

Les actions des fourmis ont pour objectif de détecter les contours des cercles de la pupille et de l'iris, pour cela, nous avons utilisé l'algorithme de Canny [Canny, 1986] pour générer l'image des contours. Le principe général de l'algorithme de Canny peut être résumé en cinq étapes :

Etape 1 : appliquer un filtre gaussien.

Etape 2 : calculer le gradient en chaque pixel de l'image.

$$\text{La norme de gradient : } |\Delta f| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \quad (5.1)$$

$$\text{L'angle du gradient : } \theta = \arctg \left(\frac{\frac{\partial f}{\partial y}}{\frac{\partial f}{\partial x}} \right) \text{ si } \frac{\partial f}{\partial x} \neq 0 \quad (5.2)$$

$$\theta = \frac{\pi}{2} \text{ sinon}$$

Etape 3 : approximation de l'angle de gradient à un multiple de 45° près.

Etape 4 : si la norme de gradient en pixel (x,y) est inférieure à la norme du gradient en un de ses deux voisins le long du gradient, mettre la norme du gradient en (x,y) à 0 (on supprime les gradients qui ne sont pas des maximums locaux).

Etape 5 : (hystérésis) soit $\tau_h > \tau_l$ deux seuils. Si la norme du gradient en un pixel (x,y) est supérieure à τ_h ajouter (x,y) au contour ainsi que tous les point reliés à (x,y) le long de la normale au gradient pour lesquels la norme du gradient est supérieure à τ_l .

La figure 6.2 illustre une image de la base de données CASIA, le résultat de l'application de l'algorithme de Canny est illustré dans la figure 6.3

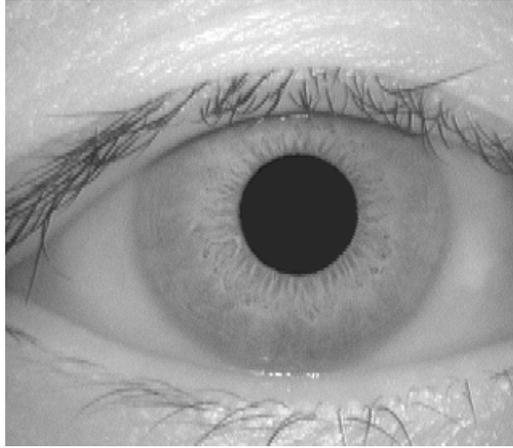


FIG 6.2. Image 001_1_1 de la base de données CASIA.

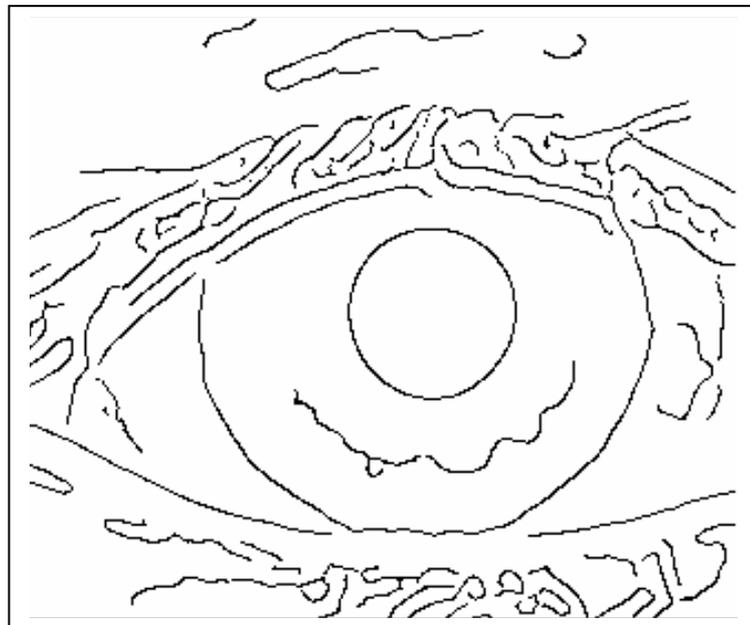


FIG 6.3. Détection des contours de l'image 001_1_1 de la base de données CASIA par l'algorithme de Canny.

6.4.2 Initialisation

Les algorithmes basés sur ACO peuvent être initialisés par des solutions générées par autres méthodes, Dorigo, par exemple a utilisé la méthode de plus proche voisin pour initialiser l'algorithme de PVC. Dans notre cas, l'algorithme commence par une analyse d'histogramme permettant de détecter la zone la plus sombre au centre de l'image, et ainsi de localiser approximativement le centre de la pupille (figure 6.5).

Ceci est effectué par une segmentation sur le premier pic de l'histogramme lissé et seuillé. Ce centre approximatif va nous servir comme un point de départ de chaque équipe de fourmis pour la détection de contour de la pupille dans chacune des deux parties de l'image. Le processus de la détection de cercle de la pupille est similaire à celui de la détection de cercle de l'iris, la seule différence réside dans le point de départ des fourmis, car une fois le cercle de la pupille est détecté, il sera considéré comme un point de départ des fourmis pour la détection de cercle de l'iris. Aussi le nombre de fourmis utilisé pour la détection de cercle de l'iris est plus grand que celui utilisé pour la détection de cercle de la pupille.

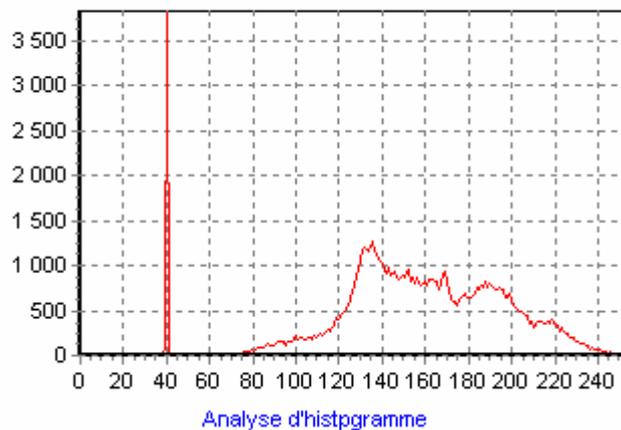


FIG 6.4. Analyse d'histogramme de l'image 001_1_1 de la base de données CASIA.

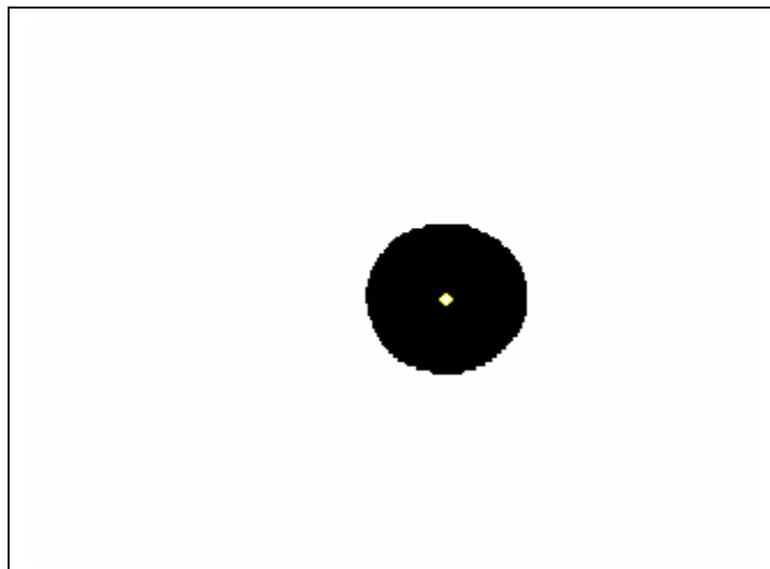


FIG 6.5. Détection approximative du centre de la pupille de l'image 001_1_1 de la base de données CASIA par une analyse d'histogramme.

6.4.3 Construction de la solution

Chaque équipe est constituée de K fourmis, chaque fourmi est assimilée à un processus stochastique de construction de solutions itératif, Le comportement de chaque fourmi est décrit par son cycle d'exécution durant lequel, elle fait des déplacements d'une cellule à l'autre (selon une connectivité à 8 voisins) pour construire une solution possible (même de mauvaise qualité) de manière stochastique et incrémentale en ajoutant de nouvelles composantes à cette solution. Les déplacements ou plus précisément le choix de la cellule suivante pour le prochain déplacement est guidé par :

- La concentration de phéromone laissée par les autres fourmis de la même équipe dans cette cellule.
- L'information heuristique qui dans ce problème, représente l'existence d'un contour dans cette cellule.

Les traces de phéromones représentent une mémoire collective de la colonie de fourmis et permettent les interactions entre les fourmis (stigmergie), elles sont régulièrement mises à jour par le processus de construction et par un mécanisme de l'évaporation de la phéromone. Ce dernier permet d'éviter une convergence prématurée (stagnation) due à la découverte d'un optimum local, en outre, il est même utile pour éliminer les solutions non satisfaisantes. Plusieurs schémas de mise à jour de la phéromone ont été proposés dans la littérature [Dorigo, 1997, Stutzle, 1997, Dorigo, 1996]. Dans notre cas, l'évaporation est effectuée après chaque itération en diminuant les traces de phéromones. La mise à jour des phéromones est effectuée par la fourmi en déposant une quantité à chaque déplacement durant son chemin, si ce chemin parvient à une solution, les traces de phéromone appartenant à ce chemin sont amplifiées, de la même façon que dans la nature, la phéromone s'évapore et les fourmis entretiennent la concentration des meilleures traces en empruntant le même chemin un grand nombre de fois. Le cycle d'exécution d'une fourmi peut être interrompu suivant deux cas :

- Quand elle atteint les bords de l'image, dans ce cas elle est détruite.
- Quand elle génère une solution, elle est détruite.

A chaque fourmi K , est associée une mémoire dans laquelle elle stocke les composantes qu'elle ajoute à sa solution. Cette mémoire permet à la fourmi de :

- parcourir son chemin à l'envers pour faciliter la mise à jour des traces de phéromones une fois la solution est générée.
- l'empêcher aussi de boucler sur un même chemin de parcours.

6.4.4 L'optimisation finale

Les algorithmes à colonie de fourmis sont souvent complétés par une optimisation locale afin d'améliorer les solutions trouvées par les fourmis et accélérer la convergence de l'algorithme vers les meilleurs solutions [Oudfel, 2005]. Chaque solution créée par une fourmi est ainsi améliorée avant son évaluation. Pour notre cas, une étape d'optimisation est effectuée après la détection d'un cercle. La figure 6.6 illustre le résultat de l'algorithme sur l'image 001_1_1 de la base de données CASIA.

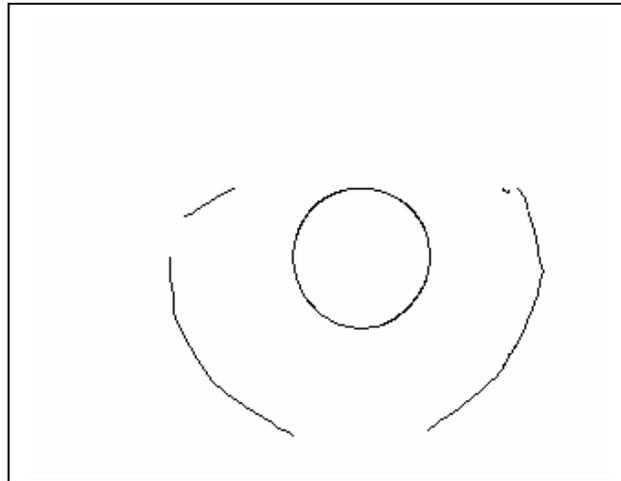


FIG 6.6. Résultat de l'algorithme des fourmis sur l'image 001_1_1 de la base de données.

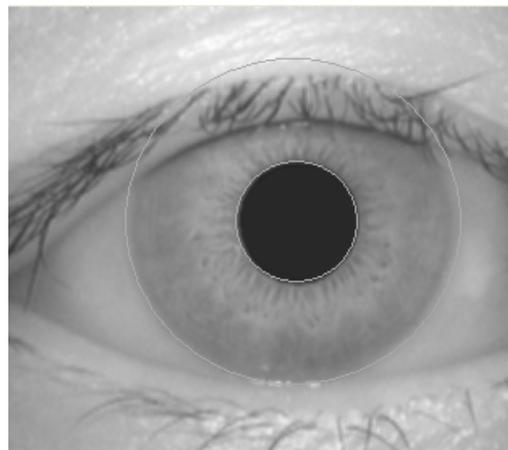


FIG 6.7. Segmentation de l'iris par l'algorithme des fourmis sur l'image 001_1_1 de la base de données CASIA.

Le processus de fonctionnement de notre algorithme est relativement simple et peut être résumé en quatre étapes :

1. Détection approximative de centre de la pupille par une analyse d'histogramme ;
2. Détection de contours par l'algorithme de Canny ;
3. Détection de la pupille par application des fourmis artificielles ;
4. Détection de l'iris par application des fourmis artificielles.

6.7 Déroulement de l'iris

L'iris localisé est déroulé suivant la transformation polaire proposée par Daugman. Les dimensions de l'image déroulées sont fixées à 128*8 pixels. L'intérêt de cette étape rajoutée n'est pas de concevoir un système biométrique de reconnaissance de l'iris, mais nous voulions juste voir la région d'iris extraite par notre algorithme car ceci permet aussi d'évaluer la performance de notre algorithme.

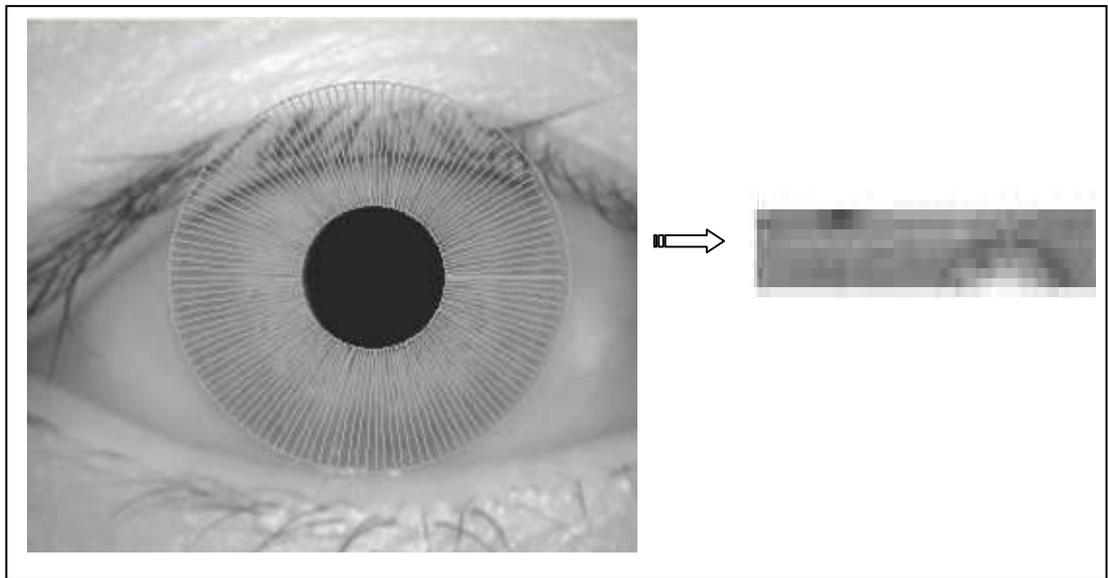


FIG 6.8. Transformation polaire.

6.8 Résultats et discussion

Nous avons appliqué l'algorithme sur 548 yeux de 88 personnes de la base CASIAv1. Pour bien visualiser les résultats, nous avons illustré le résultat de segmentation de quelques images. Cependant, pour bien analyser les résultats, nous avons établi des histogrammes qui donnent pour chaque personne parmi les 88 personnes le taux de détection de l'iris par rapport au nombre d'iris pour cette personne. Ceci permettra une évaluation plus précise des résultats obtenus. Le tableau 6.1 donne le taux de détection globale de système.

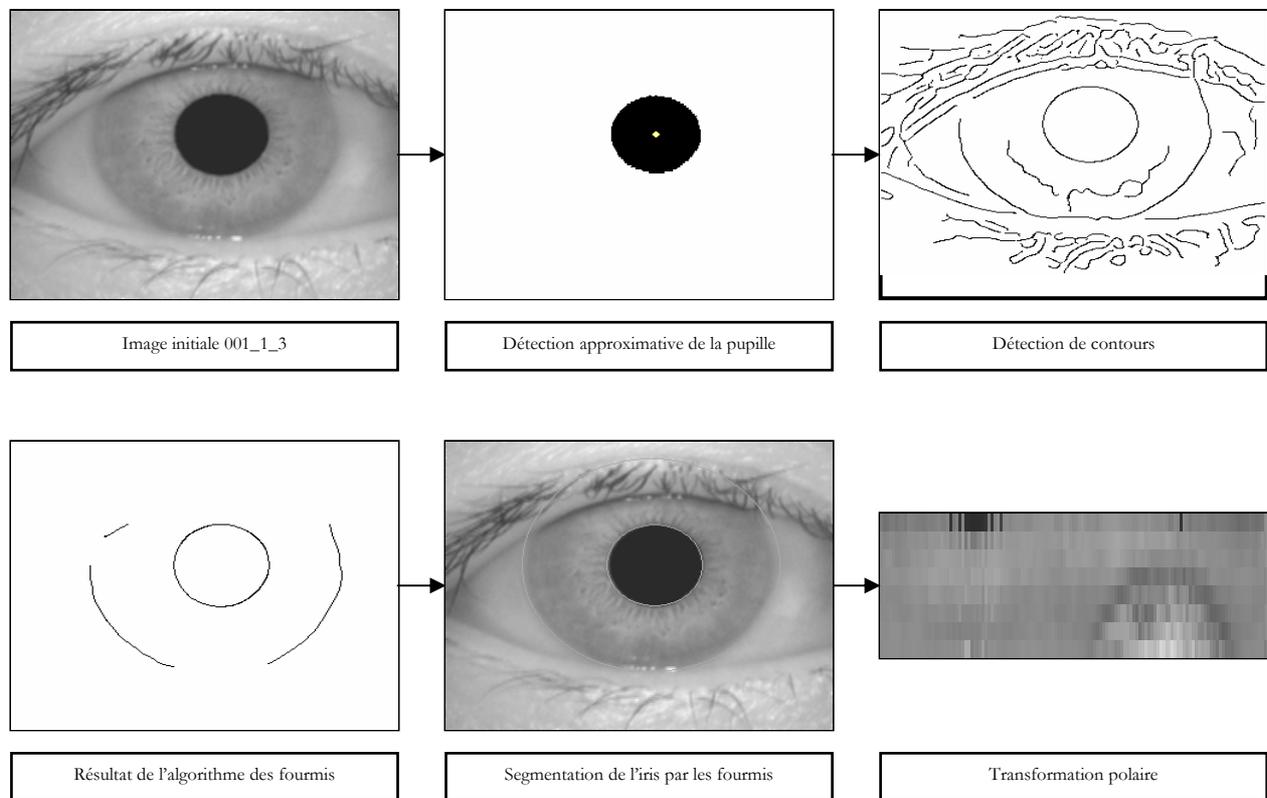


FIG 6.9.a Illustration des résultats de l'application de l'algorithme sur l'image 001_1_3.

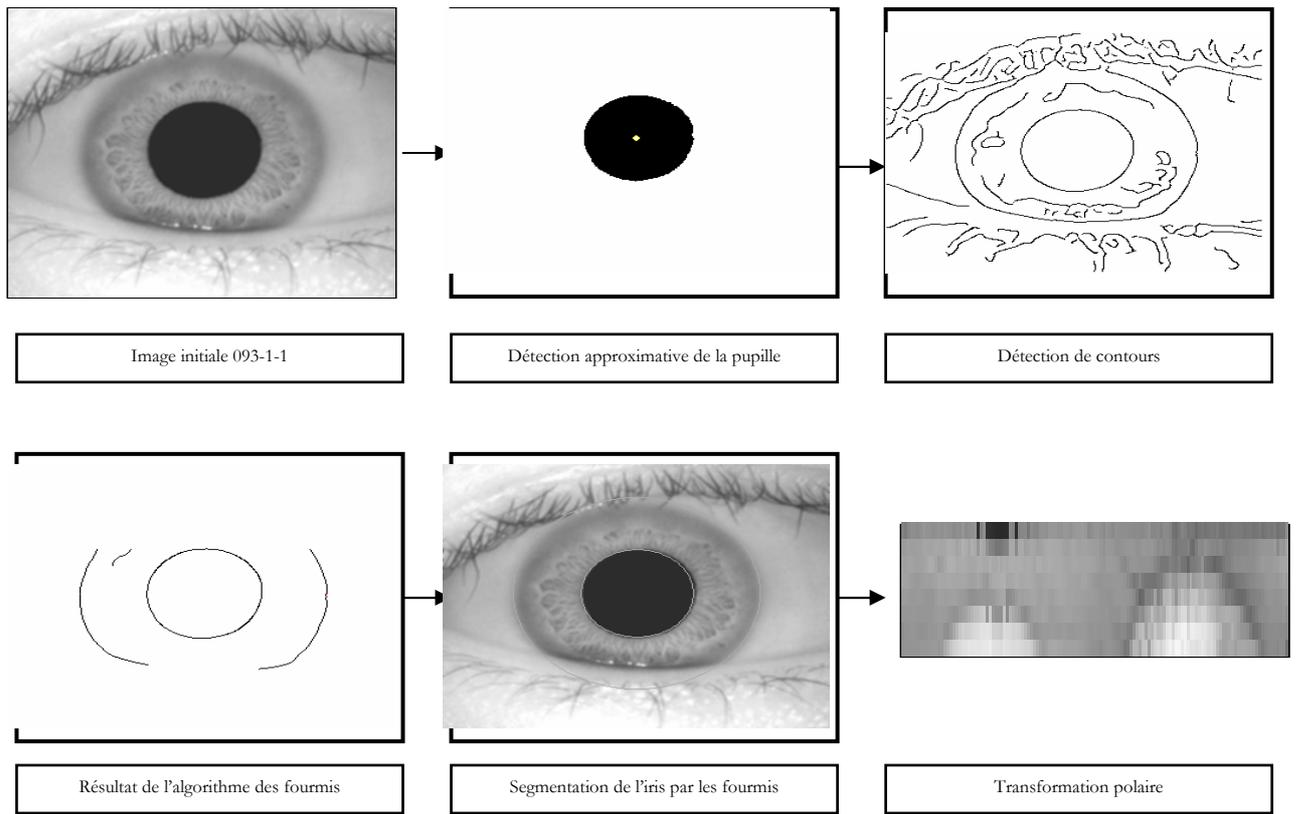


FIG 6.9.b Illustration des résultats de l'application de l'algorithme sur l'image 0093_1_1

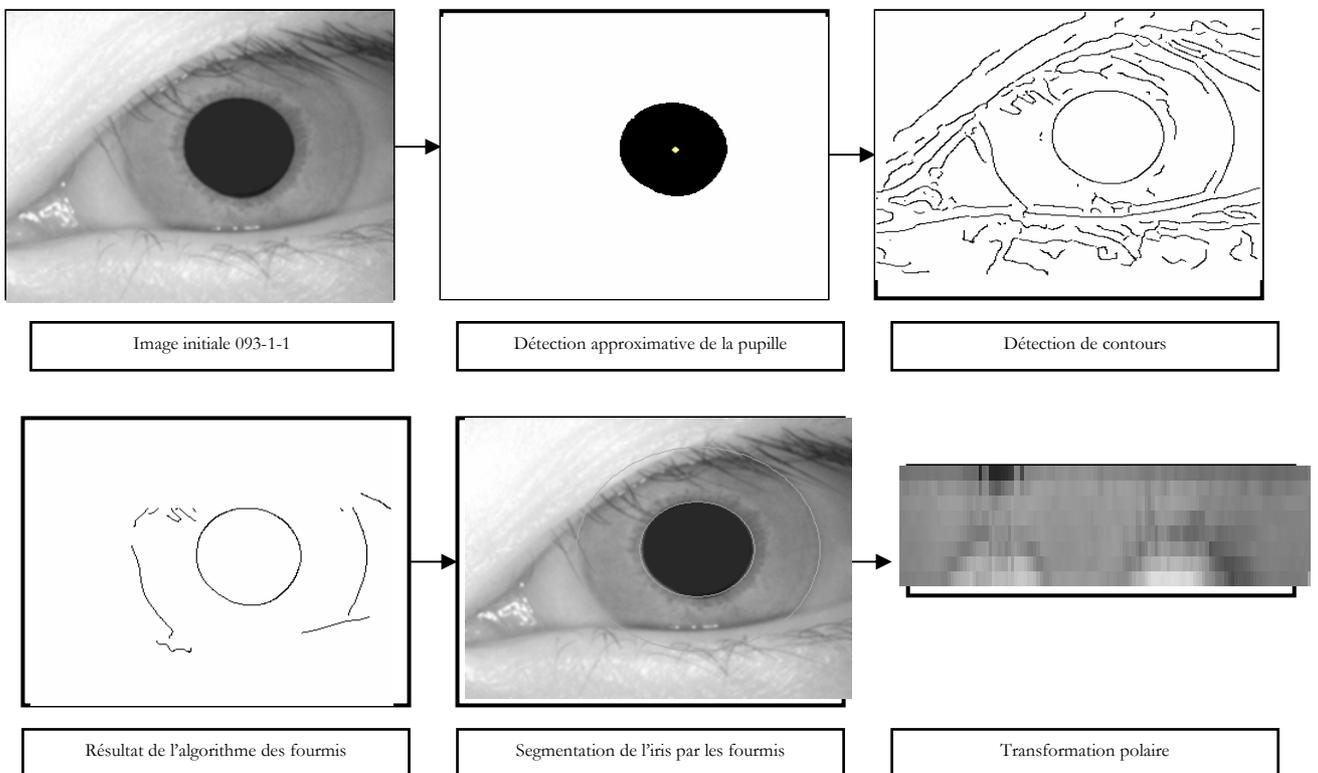


FIG 6.9.c Illustration des résultats de l'application de l'algorithme sur l'image 018-1-2

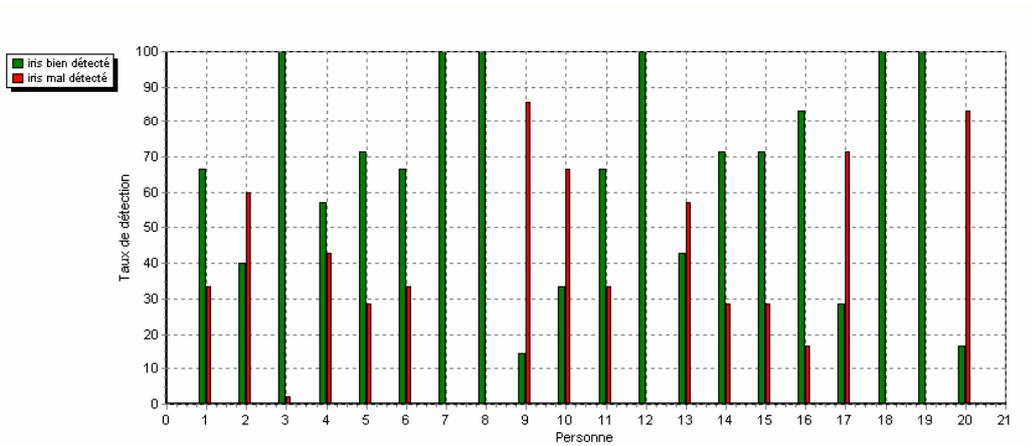


FIG 6.10.a Taux de détection d'iris pour chaque personne

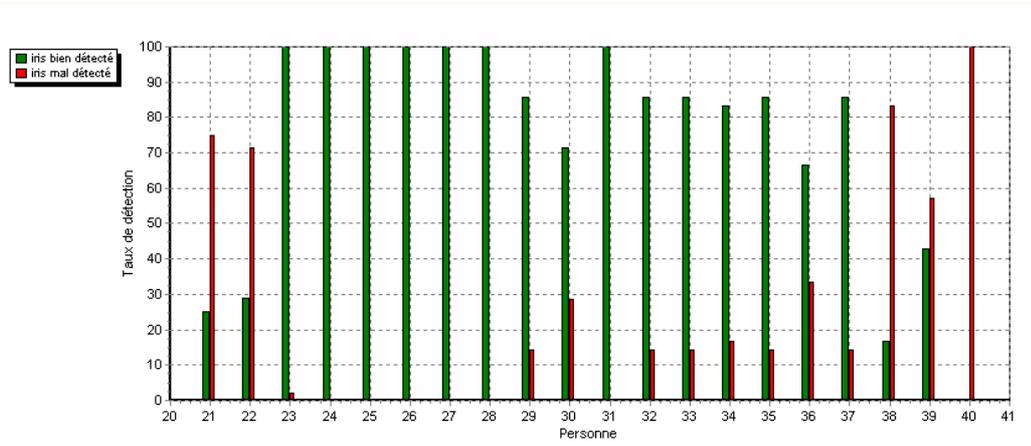


FIG 6.10.b Taux de détection d'iris pour chaque personne

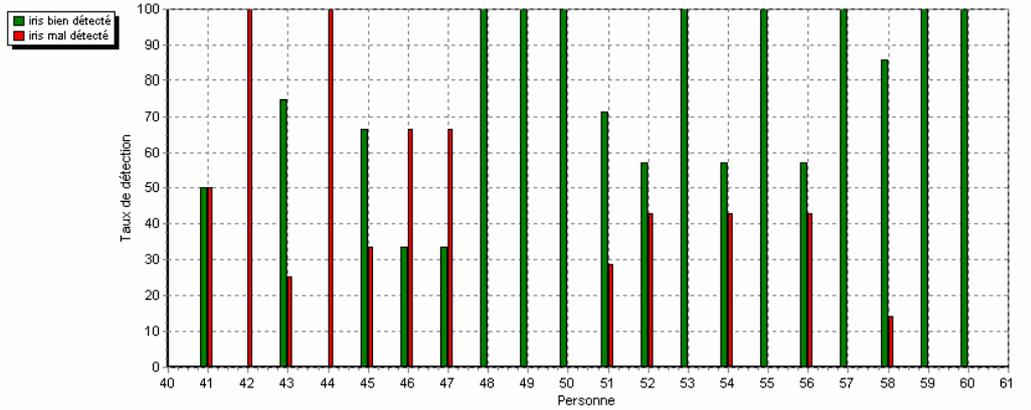


FIG 6.10.c Taux de détection d'iris pour chaque personne

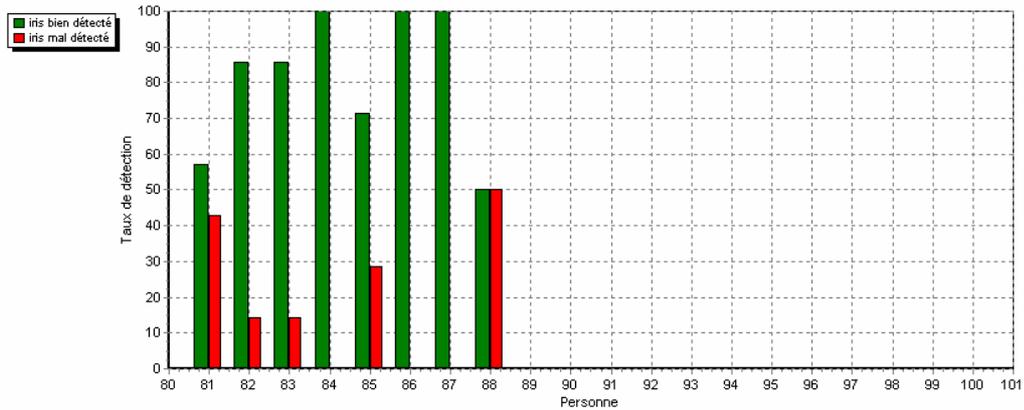


FIG 6.10.d Taux de détection d'iris pour chaque personne

Nombre d'iris	Nombre d'iris bien détectés	Taux de bonne détection	Nombre d'iris mal détectés	Taux de mauvaise détection
548	404	73.72	144	26.28

TAB 6.1. Taux global de détection de l'iris sur 548 images d'œil de la base CASIA

A travers ces résultats obtenus, nous voyons bien que la détection d'iris était fiable bien que dans certaines images, on remarque une mauvaise détection de l'iris. Ceci peut être expliqué par une mauvaise détection de contours car l'algorithme est basé sur la recherche d'un contour, sa direction, sa suite lorsqu'il y a des discontinuité et enfin fait il partie de contour d'iris. An niveau de la transformée polaire, on remarque que l'iris était bien déroulé.

6.9 Conclusion

Dans ce chapitre, nous avons transposé l'approche ACO à la segmentation d'iris, il s'agit d'un autre domaine d'utilisation car la plupart des algorithmes ACO étaient dédiés à l'optimisation combinatoire. On peut conclure que cette approche s'avère adéquate puisque les résultats étaient satisfaisants.

Conclusion et perspectives

Le comportement des insectes sociaux est caractérisé par l'auto-organisation. Les individus communiquent en changeant les propriétés locales de leur environnement, et, par le biais de ce moyen de communication limitée, une sorte d'intelligence collective émerge. Les fourmis naturelles ont inspiré les "algorithmes à colonies de fourmis" initialement proposés par Marco Dorigo. L'idée générale est d'imiter le comportement coopératif d'une colonie de fourmis pour trouver le plus court chemin entre une source de nourriture et leur nid pour résoudre des problèmes complexes d'optimisation combinatoire. L'idée originale s'est depuis diversifiée pour résoudre une classe plus large de problèmes et plusieurs algorithmes ont vu le jour, s'inspirant de divers aspects du comportement des fourmis.

Dans ce mémoire, Nous avons abordé différents modèles de fourmis artificielles ainsi que les différents aspects de leur utilisation.

Nous avons tout d'abord commencé par décrire les principales caractéristiques des fourmis réelles qui peuvent être exploitées pour la résolution des problèmes. Les fourmis possèdent des comportements auto-organisés, à travers lesquels des interactions simples au niveau local permettent l'émergence d'un comportement globale complexe.

Les travaux menés dans ce mémoire portent sur trois approches principales s'inspirant des comportements collectifs des fourmis.

La première approche concerne la faculté des fourmis à trier leur couvain. Les travaux fondateurs dans ce domaine sont ceux de Deneubourg et ses collègues. Ils ont élaboré un modèle mathématique qui définit les probabilités sur les quelles une fourmi a tendance à ramasser ou à déposer un objet en fonction de son environnement qui est modélisé par un plan. Ces travaux ont été ensuite repris et

étendus par Lumer et Faieta pour la classification des données numériques. Les extensions portés concernent en particulièrement les point suivants : Les donnés sont représentés par des vecteurs de caractéristiques et initialement éparpillés dans une grille 2D. La similarité entre deux données est mesurée comme une distance euclidienne entre leur vecteur de caractéristiques. En se basant sur les travaux de Lumer et Faieta, Monmarché introduit AntClass, un algorithme de classification utilisant une population de fourmis avec des modifications de base. AntClass utilise une grille toroïdale et chaque fourmi a la possibilité de transporter plusieurs objets à la fois et de déposer un tas d'objets sur une même case. De plus AntClass est une hybridation d'un algorithme de fourmi et d'un algorithme de classification classique de type K-means. L'algorithme AntClass été expérimenté sur des données réelles et les résultats étaient prometteurs. Les principales critiques concernent le temps de calcul relativement important, aussi plusieurs paramètres se rattachent aux fourmis. Dans le même contexte, et en se basant sur ces travaux, nous contribuons à l'étude des fourmis artificielles dans l'objectif de résoudre des problèmes de partitionnement. Pour cela nous avons repris ces travaux et nous les avons étendus : La fourmi dispose d'une perception spatiale lui permettant de percevoir tous les objets se trouvant dans sa région au point où elle se trouve et d'une perception temporelle pour mémoriser les objets et leurs emplacements durant ses déplacements. La fourmi perçoit sa région, et commence à balayer cette région dans sa totalité en considérant chaque élément et vérifie si elle peut le réunir avec d'autres éléments appartenant à la même région avant de le ramasser. Une fois elle construit tous les tas possibles dans cette région, elle fait intervenir sa mémoire pour comparer les tas obtenus avec ceux construits ou rencontrés durant ses déplacements. Ainsi ce nouveau comportement permet de réduire considérablement le temps. Par rapport aux travaux initiaux, nous obtenons de bons résultats avec beaucoup moins d'itérations.

Dans la deuxième approche, nous nous intéressons au système de reconnaissance chimique des fourmis. Celui-ci, connu sous le nom de fermeture coloniale, repose sur l'apprentissage d'une odeur propre à chaque nid et permet la détection des intrusions. Dans cette étude, nous avons mis l'accent sur la biométrie par l'iris. Après avoir introduit les concepts principaux et les notions relatifs à ce domaine, nous nous sommes focalisés sur la problématique des systèmes de vérification biométrique par reconnaissance de l'iris en terme de pouvoir réussir les

tests d'impostures et d'acceptations des utilisateurs légitimes. Pour cela nous avons utilisé l'algorithme AntClust inspiré du système de reconnaissance chimique des fourmis, et dont la capacité à reconnaître les membres appartenant au nid et à rejeter les intrus nous a fait penser que l'emploi de cet algorithme dans un système de vérification biométrique par reconnaissance de l'iris lors du processus de matching serait adapté à la résolution de ce problème. Cette approche s'est révélée adéquate puisque les résultats obtenus sont encourageants.

En restant toujours dans le domaine de la biométrie par l'iris, un autre aspect de notre contribution est la transposition de l'approche ACO à la résolution collective du problème de segmentation de l'iris selon une approche de système multi agents réactifs. Notre motivation est que le problème de la segmentation de l'iris en tant qu'un problème de classification peut être un bon candidat pour les ACO qui ont été adaptés aussi pour les problèmes de la classification. L'idée de base est de représenter le problème de la segmentation de l'iris sous la forme d'un problème d'une recherche stochastique des pixels appartenant à l'un des motifs circulaires de la pupille ou de l'iris et de simuler le comportement de fourragement des fourmis lors de la recherche de plus court chemin entre le nid et une source de nourriture. Des fourmis artificielles se déplacent sur l'image des contours à la recherche des pixels appartenant au contour d'un cercle afin de détecter ce dernier. Les fourmis communiquent indirectement en changeant les propriétés de leur environnement par le dépôt des traces de phéromones sur leur chemin, cette trace attire les autres fourmis dans une boucle de rétroaction positive pour faire émerger la solution recherchée correspondant à la détection des deux motifs circulaire de la pupille et de l'iris dans l'image de l'œil. Nous avons choisi le paradigme ACO basé sur l'intelligence artificielle distribuée car ce domaine utilise une approche locale de résolution dont les définitions s'appliquent bien à la modélisation d'une fourmi par un agent réactif et d'une colonie de fourmis par un système multi agents. L'algorithme développé dans cette optique a été testé sur des images issues de la base CASIAv1 et les résultats de la détection de l'iris étaient très fiables.

Le but de cette étude était d'exploiter les modèles de fourmis artificielles dans le domaine de la classification. Un travail de recherche n'est pas une fin en soi, mais plutôt une base pour les recherches futures. Nous allons donc donner quelques perspectives qui nous sont apparues pour certains de nos travaux :

Nous proposons pour l'algorithme AntClass d'abandonner la grille et de rechercher une autre modélisation de l'environnement, car une représentation pareille de l'environnement avec une base de données volumineuse rend la tâche difficile et l'algorithme sera incapable de classifier les données correctement. Nous proposons aussi d'utiliser le système de reconnaissance chimique des fourmis dans les autres techniques biométriques et même de l'utiliser dans des systèmes d'identification biométrique. En ce qui concerne l'approche ACO. Il paraît intéressant de l'utiliser comme un outil d'optimisation pour les réseaux de neurones afin de rechercher les poids optimaux.

Pour finir, notons que les modèles de fourmis ne sont pas certainement complets dans la mesure où il reste beaucoup de chose à apprendre des fourmis et que ce que l'on connaît dépend d'une espèce étudiée. Donc la découverte d'une nouvelle propriété biologique et l'élaboration d'un modèle mathématique adéquat ouvrent certainement la voie vers de nouvelles solutions à des problèmes informatiques.

Résumé

L'étude des comportements collectifs des fourmis constitue une source riche d'inspiration pour la résolution des problèmes informatiques. Ces insectes sociaux résolvent de nombreux problèmes de manière très efficace dans leur environnement, donc, il doit être possible et utile de transporter un certain nombre de ces mécanismes pour la résolution des problèmes, notamment les problèmes qui ne sont pas clairement et mathématiquement définis. Les travaux menés dans ce mémoire portent sur l'étude de trois approches principales s'inspirant des comportements collectifs des fourmis dans le cadre de la classification. La première approche modélise la faculté des fourmis à trier collectivement leur couvain pour résoudre un problème de partitionnement. La seconde approche concerne la modélisation de système de reconnaissance chimique des fourmis. Nous avons adapté ce modèle à un système de vérification biométrique par reconnaissance de l'iris afin de surmonter la performance de ce dernier en terme de réussir les tests d'impostures et d'acceptations des utilisateurs légitimes. Enfin, la dernière approche représente un autre aspect de l'utilisation des fourmis artificielles dans la biométrie par l'iris. Il s'agit de transporter le concept ACO à la résolution collective du problème de segmentation d'iris selon une approche des systèmes multi agents réactifs.

Mots-clés : ACO, Biométrie, Canny, Classification, Fermeture coloniale, Fourmis artificielles, Iris, Optimisation, Pheromone artificielle, Segmentation.

- [Bonabeau et al, 1999] E. Bonabeau, M. Dorigo, et G. Theraulaz, *Swarm Intelligence : From Natural to Artificial Systems*. Oxford University Press, New York, 1999.
- [Bonabeau, 2000] E. Bonabeau, G. Theraulaz, *L'intelligence en essaim*, Pour la science, N° 271 mai 2000, 282 (3) : pp. 66-73.
- [Bonabeau et Theraulaz, 1994] E. Bonabeau, G. Theraulaz, *Intelligence Collective*, Hermes, 1994.
- [Bullnheimer, 1997] B. Bullnheimer, R.Hartl, et C. Strauss, *Applying the Ant System to the Vehicle Routing Problems*, 2nd Metaheuristic International Conference (MIC-97), Sophia-Antipolis, France, 1997.
- [Canny, 1986] J. Canny, *A computational approach to edge detection*, IEEE Trans. On Pattern Analysis and Machine Intelligence, vol. 8, n°6, pp. 679-698, novembre 1986.
- [Carlin et Hölldobler, 1987] N.F. Carlin, B. Hölldobler, *The Kin Recognition system of Carpenter Ants*, Behave Ecole Sociobiologie, 1987, N° 20, pp.209-217.
- [Casia, 2003] Chinese Academy of Sciences – Institute of Automation, *Database of 756 Greyscale Eye Images*.
<<http://www.sinobiometrics.com> Version >.
- [Colorni et al, 1991; Dorigo et al., 1991; Dorigo, 1992] A. Colorni, M., et V. Maniezzo, *Distributed Optimization by Ant Colonies*, In (Varela and Bourguine) 1991, pp. 134–142.
- [Costa, 1997; Dudot, 2005] D. Costa, A. Hertz, *Ants Can Colour Graphs*, Journal of the Operational Research Society, 2005, 48: 295–305.

- [Daugman, 1993] J. Daugman, *High confidence visual recognition of persons by a test of statistical independence*, IEEE Trans 1993, pp. 1148-1161.
- [Daugman, 1994] J. Daugman, *Biometric Personal Identification System Based on Iris Analysis*, US Patent, 1994, vol. No. 5,291,560.
- [Daugman, 1995] J. Daugman, *High confidence recognition of persons by rapid video analysis of iris*, IEE European Convention on Security and Detection, n°408, Mai 1995.
- [Deneubourg, 1990] J.L Deneubourg, S Aron, S. Goss, et J.M Pasteels, *The self-organizing exploratory pattern of the argentine ant*, Dans Journalon insect Behavior,1990, (3): pp. 159-168.
- [Deneubourg, 1991] J.L Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, et L. Chretien,. *The dynamic of collective sorting robot-like ants and ant-like robots*, in J. A. Meyer and S. W. Wilson (eds), SAB 90 - 1st Conference On Simulation of Adaptive Behavior: From Animals to Animats, MIT Press. 1991.
- [Di Caro, 1998] G.Di Caro, M. Dorigo, *Two Ant Colony Algorithms for Best-Effort Routing in Datagram Networks*, In Proceedings of PDCS'98 - 10th International Conference on Parallel and Distributed Computing and Systems, Las Vegas, Nevada, October 28-31, 1998.
- [Dorigo, 2004] M .Dorigo, T. Stuzle, *Ant Colony Oprimization*, MIT Press, Cambridge, MA, 2004.
- [Dorigo, 1996] M. Dorigo, V. Maniezzo, A. Colorni,, *The Ant System: Optimization by a Colony of Cooperating Agents*, IEEE Transactions on Systems, Man and Cybernetics-Part B, 1996, 1(26): pp. 29-41.
- [Dorigo et Gambardella, 1997] M. Dorigo, L.M. Gambardella. *Ant colonies for the Traveling Salesman Problem*, BioSystems, 1997, 43: pp. 73–81.

- [Fresneau, 1985] D.Fresneau, *Individual foraging and path fidelity in a ponerine ant*, Insectes Sociaux, Paris,1985, 32(2) :109–116.
- [Gambardella et Dorigo, 1995] L.M. Gambardella, M. Dorigo. *Ant-Q: A reinforcement learning approach to the Travelling Salesman Problem*, In A. Prieditis, S. Russell, editors, Proceedings of the Twelfth International Conference on Machine Learning, Morgan Kaufmann, San Mateo, California, 1995, pp. 252–260.
- [Gambardella. 1997; Merklöe, 2002] L. Gambardella, M. Dorigo, HAS-SOP : *An Hybrid Ant System for the Sequential Ordering Problem*, Technical Report 11, IDSIA, Lugano, Switzerland, 1997.
- [Gambardella. 1997; Merklöe, 2002] L. Gambardella, M. Dorigo, E. Taillard, *Ant Colonies for the QAP*, Technical Report 4, IDSIA, Lugano, Switzerland, 1997.
- [Goss , 1989; Deuneubourg, 1990] S. Goss, S. Aron, J.-L.Deneubourg , J.M. Pasteels, *Self organized shortcuts in the argentine ant*, Naturwissenschaften, Vol. 76, 1989, pp. 579-581.
- [Handl, 2003] J. Handl, *Ant-based methods for tasks of clustering and topographic mapping: extensions, analysis and comparison with alternative techniques*, Masters Thesis, universität Erlangen-Nürnberg, Erlangen, Germany. 2003.
- [Holland, 1975] J.H. Holland, *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor, 1975.
- [Holland, 1999] O.E. Haolland, C. Melhuish, *Stigmergy, self-organization, and sorting in collective robotics*. Artificial Life, N° 5, pp :173–202, 1999.
- [Hölldobler et Wilson, 1990] B. Hölldobler, E. Wilson, *The Ants*, Springer Verlag, Berlin, Germany, 1990.

- [Hölldobler et Wilson, 1996] B. Hölldobler, E. Wilson, *Voyage chez les Fourmis*. Seuil, 1996.
- [Jain et Dubes, 1988] A. Jain, R. Dubes, *Algorithms for Clustering Data*. Prentice Hall Advanced, Reference Series, 1988.
- [Jayaraman et al, 2003] V.K. Jayaraman, P.S, Shelokar, B.D, Kulkarni, *An ant colony approach for clustering*, Technical report, Chemical Engineering and Process Division, National Chemical Laboratory, India, 2003.
- [Jerne, 1974] N.K. Jerne, *Towards a network theory of the immune system*, Ann. Immunol. (Inst. Pasteur), 125C :373–389, 1974.
- [Kong et Zhang, 2001] W. Kong et D. Zhang, *Accurate iris segmentation based on novel reflection and eyelash detection model*, Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing, Hong Kong, 2001.
- [Kuntz et al, 1997]P. Kuntz, P. Layzell, D. Snyers, *A Colony of Ant-like Agents for Partitioning in VLSI Technology*, In Husbands and Harvey, 1997, pp. 417–424.
- [Labroche, 2002] N. Labroche, N. Monmarché, G. Venturini, *A new clustering algorithm based on the chemical recognition system of ants*, Proc. of the 15th European Conference on Artificial Intelligence, France, IOS Press. 2002, pp. 345-349.
- [Lumer et Faieta, 1994] E. Lumer, B. Faieta, *Diversity and adaptation in populations of clustering ants*, Proceedings of the 3rd International Conference on the Simulation of Adaptive Behaviour, 1994, pp. 501–508.
- [Ma et Wang et Tan, 2002] L. Ma, Y. Wang et T. Tan, *Iris recognition using circular symmetric filters*, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, 2002.

- [McCulloch, 1943] W. McCulloch, W.Pitts, *A logical calculus of the ideas immanent in nervous activity*, Bulletin of Mathematical Biophysics 5: 115-133, 1943.
- [Masek et al, 2003] L. Masek, P. Kovesi, *MATLAB Source Code for a Biometric Identification System based on Iris Patterns*, The University of Western Australia, 2003.
<<http://www.csse.uwa.edu.au/pk/studentprojects/libor>>.
- [Monmarché, 1999] N. Monmarché, *On Data Clustering with Artificial Ants*, In Freitas, 1999, pp. 23–26.
- [Monmarché, 1999] N. Monmarché, G. Venturini, M. Slimane, *Classification non supervisée par une population de fourmis artificielles*, In Actes des Colloques Insectes Sociaux, Tours, France, (1999), pp. 43–52.
- [Oppenheim et Lim, 1981] A. Oppenheim et J. Lim, *L'importance de la phase dans les signaux*, Proc. IEEE, 1981, vol 69, pp. 529-541,
- [Ouadfel et Batouche, 2005] S. Ouadfel, M. Batouche, *Des Fourmis pour la Segmentation des images*, JIG : Journées en Informatiques Graphique, Biskra, Algérie, Novembre 2005.
- [Parpinelli, 2002] R.S. Parpinelli, H. S. Lopez et A. A. Freita, *data mining with ant colony optimisation algorithm*, IEEE Transaction on Evolutionary Computation, August 2002, 6(4): 321-332.
- [Ramos, 2002] V. Ramos, F. Muge, F. and P. Pina, *Self-organized data and image retrieval as a consequence of inter-dynamic synergistic relationships in artificial ant colonies*, Vol. 87, IOS Press.2002.
- [Ritter, 1999] N. Ritter, *Location of the Pupil-Iris Border in Slit Lamp Images of the Corne*, Proceedings of the International Conference on Image Analysis and Processing, 1999.

- [Ritter et Cooper, 2003] N. Ritter et J. Cooper, *Locating the iris: A first step to registration and identification*, Proceedings of the 9th IASTED International Conference on Signal and Image processing, 2003, pp. 507-512.
- [Sanchez-Reillo et Sanchez-Avila et Martin-Pereda, 1999] R. Sanchez-Reillo, C. Sanchez-Avila, et J-A. Martin-Pereda, *Minimal template size for iris recognition*, Proc. of the Conference BMES/EMBS, Atlanta, octobre 1999.
- [Segond, 2004] M. Segond, *Ant Algorithms for Detection in Coastal Waters*,. Magistec, Octobre 2004.
- [Semet, 2003] Y. Semet, E. Lutton et P.Collet, *Ant Colony Optimization for e-learning : Observing the emergence of pedagogic suggestions*, In IEEE Swarm Intelligence Symposium 2003, Indianapolis, Indiana, april 2003.
- [Sendova et Franks, 1992] A. Sendova, N. Franks, *Task allocation in ant colonies within variable environments (a study of temporal polyethism)*, Bulletin of Mathematical Biology, 1992, N°55, pp. 75–96.
- [Senouci, 2008] A. Senouci, L. Mekaliche, *AntClass amélioré : Algorithme à Base de Population de Fourmis pour la Classification Non Supervisée*, 10^{ème} conférence maghrébine sur les technologies de l’information, Oran, Algérie, Avril 2008.
- [Solnon, 2000] C. Solnon, *Solving permutation constraint satisfaction problems with artificial ants*, In Proceedings of ECAI’2000, IOS Press, Amsterdam, The Netherlands, 2000, pp 118–122.
- [Stützle et Hoos, 1997] T. Stützle, H. Hoos, *MAX –MIN Ant System and local search for the Traveling Salesman Problem*, In Proceedings of the fourth International Conference on Evolutionary Computation, IEEE Press, 1997, pp. 308–313.

[Taillard , 1997; Maniezzo. 1994] E.Taillard, L.M. Gambardella, *An Ant Approach for Structured Quadratic Assignment Problems*, In 2nd Metaheuristics International Conference, Sophia-Antipolis, France (1997).

[Tisse et Martin et Torres Robert, 2002] C. Tisse et L. Martin et L. Torres et M. Robert, *Person identification technique using human iris recognition*, International Conference on Vision Interface, Canada, 2002.

[Vander Meer et al., 1998] Vander Meer, R., M., K. E, et M. L., W., *Pheromone Communication in Social Insects*, Westview Press, editor 1998.

[Wildes, 1997] RP. Wildes, *Iris Recognition: An Emerging Biometric Technology*, Proc. of the IEEE, vol.85, pp.1348-1363, Sept. 1997.