

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
Université M^{ed} Khider, Biskra
Faculté des sciences exactes, des sciences de la nature et de la vie
Département d'informatique



THESE POUR OBTENIR LE TITRE DE DOCTORAT
EN INFORMATIQUE

Option: Intelligence artificielle et image

Sur le thème

Apprentissage et adaptation dans les
Systèmes bio-inspirés

Présentée par

M^{me} Leila DJEROU

Devant le jury composé de:

Président
Examineurs

N. Djedi
M. C. Babahenini
A. Boukerram
O. Kazar
M. K. Kholladi
M. Batouche

Professeur, Université de Biskra
Maître de Conf, Université de Biskra
Maître de Conf, Université de Sétif
Maître de Conf, Université de Biskra
Maître de Conf, Université de Constantine
Professeur, Université de Constantine

Directeur

Année 2009/2010

Résumé

Ce travail de thèse, s'inscrit dans le cadre des travaux qui s'inspirent des mécanismes collectifs présents dans les systèmes complexes biologiques. Il s'intéresse particulièrement au problème de traitement et d'analyse d'images, en exploitant les nouvelles possibilités offertes par ce nouveau paradigme pour la résolution de problème.

Dans ce contexte, notre contribution prend deux formes : la première consiste en la proposition d'une méthode de segmentation adaptative basée sur le concept d'agent et utilise une combinaison de deux techniques de l'intelligence en essaim, à savoir, l'Optimisation par Essaim de Particules (*PSO*) et la stigmergie, afin de résoudre les problèmes liés aux techniques de croissance de régions. La deuxième consiste en la proposition d'une méthode de segmentation non supervisée, où le problème de segmentation par seuillage est traité comme un problème d'optimisation. Ce faisant, nous avons adapté l'Optimisation par Essaim de Particules Binaires (*BPSO*) pour déterminer, automatiquement, le "nombre optimal" des seuils et chercher, simultanément, les seuils optimaux.

Mots clés : Informatique bio-inspirée, Intelligence en essaim, Optimisation par essaim de particules, Segmentation d'images, Stigmergie.

Table des matières

I	Introduction générale	1
1	Introduction générale	2
1.1	Introduction	2
1.2	Informatique bio-inspirée	2
1.2.1	L'ambition de création des machines intelligentes	3
1.2.2	Avancées technologiques	5
1.2.3	Complexité des systèmes informatiques	5
1.3	Caractéristiques des systèmes biologiques	5
1.4	Segmentation d'images	7
1.5	Contributions	8
1.6	Plan de la thèse	9
II	Etat de l'art	11
2	Mécanismes d'adaptation dans les systèmes biologiques	12
2.1	Introduction	12
2.2	Système nerveux	13
2.3	Systèmes évolutionnaires vivants	15
2.4	L'auto-organisation et les phénomènes collectifs et sociaux	17
2.4.1	L'auto-organisation	18
2.4.2	Déplacement collectif	19
2.4.3	Division du travail et l'organisation des rôles sociaux	20
2.4.4	Construction collective des nids	21
2.4.5	Tri du couvain	25
2.4.6	Recrutement et l'exploitation collective de sources de nourriture	26
2.5	Conclusion	29
3	Un survol des modèles bio-inspirés pour la résolution des problèmes complexes	30
3.1	Introduction	30
3.2	Modèles connexionnistes	30
3.3	Algorithmes Evolutionnaires	36
3.3.1	Principe d'un algorithme évolutionnaire	37
3.3.2	Algorithmes génétiques	39

3.4	Intelligence en essaim	41
3.4.1	Méthodes inspirées de l'organisation du travail chez les fourmis	42
3.4.2	Méthodes inspirées de tri collectif du couvain chez les fourmis	42
3.4.3	Méthodes d'optimisation inspirées des stratégies de fourrage- ment chez les fourmis	44
3.4.4	Méthodes inspirées de comportements de déplacements	48
3.5	L'optimisation par essaim de particules	50
3.5.1	Principe de la méthode	51
3.5.2	Voisinage d'une particule	51
3.5.3	Algorithme <i>PSO</i>	52
3.5.4	Modèle <i>Lbest</i>	55
3.5.5	Optimisation par essaim de particules binaires	56
3.6	Système multi-agents	56
3.7	Conclusion	57
4	Etat de l'art des méthodes de segmentation d'images	59
4.1	Introduction	59
4.2	Segmentation d'images basées contours	60
4.3	Segmentation d'images basées régions	60
4.3.1	Agrégation de pixels	61
4.3.2	Approche division-fusion	61
4.4	Approches coopératives	65
4.4.1	Coopération séquentielle	65
4.4.2	Coopération des résultats	66
4.4.3	Coopération mutuelle	67
4.5	Segmentation d'images par classification	68
4.6	Segmentation d'images par seuillage	71
4.6.1	Critères de seuillage	73
4.6.2	Métaheuristiques d'optimisation pour le seuillage à multi-niveaux.	77
4.6.3	Seuillage automatique	79
4.7	Systèmes multi-agents et segmentation d'images	79
4.8	Conclusion	83
III	Contributions à la Segmentation d'images	84
5	Segmentation d'images par la croissance auto-organisée de régions	85
5.1	Introduction	85
5.2	Spécification de notre approche	86
5.3	Environnement	88
5.4	Comportements des agents	88
5.4.1	Comportement Exploration	89
5.4.2	Comportement Agrégation	90
5.4.3	Comportement Fusion	91
5.4.4	Comportement Migration	91
5.5	Coordination d'agents par un champ de phéromone	92

5.6	Résultats expérimentaux	93
5.6.1	Mesures d'évaluation	94
5.6.2	Sélection de paramètres	95
5.6.3	Résultats	95
5.7	Conclusion	101
6	Segmentation d'images non supervisée par un seuillage dynamique à multi-niveaux utilisant le BPSO	103
6.1	Introduction	103
6.2	Algorithme <i>DMTBPSO</i>	103
6.2.1	Fonction Objectif	104
6.2.2	Algorithme <i>DMTBPSO</i>	105
6.3	Complexité de l'algorithme	108
6.4	Etude expérimentale	108
6.4.1	Caractéristiques des images de test	108
6.4.2	Mesures d'évaluation	111
6.4.3	Résultats	112
6.5	Conclusion	121
7	Conclusion générale	122
A	Contributions	141
A.1	Publications	141
A.1.1	Conférences avec comité de lecture	141
A.1.2	Articles avec comité de lecture	142
A.1.3	Articles IEEE	142
A.1.4	Monographie et livres	143

Table des figures

1.1	Schéma de Arkin concernant la méthodologie de conception en informatique bio-inspirée adapté de [159]	3
2.1	L'anatomie du neurone biologique	14
2.2	Structure de l'ADN	16
2.3	Synthèse moderne.	17
2.4	Nids de guêpes	21
2.5	Formation de chaînes chez les fourmis tisserandes <i>Oecophylla longinoda</i>	23
2.6	Nid de termites	23
2.7	Le cycle de construction de la termitière (d'après [136]).	25
2.8	Exemple (fictif) de carte des trajets et aires de récolte des fourrageuses <i>Pachycondyla apicalis</i> (inspiré de [89]).	27
3.1	Le neurone formel de Mac Culloch et Pitts	31
3.2	Forme générale d'un neurone formel	32
3.3	Formes de fonctions d'activation	32
3.4	Modèles de réseaux de neurones	34
3.5	Exploration locale de la fourmi autour de son nid de rattachement. Trois sites de chasses s_i sont identifiés autour du nid N . La fourmi effectue une exploration locale depuis le site s_2	48
3.6	Automate représentant le comportement individuel de fourrageur d'une fourmi. n_s représente le nombre de sites que la fourmi mémorise à un instant donné. e_j donne le nombre d'échecs successifs du site s_j mémorisé.	49
3.7	Les Boids de Reynolds, adapté de [186]	50
3.8	Principes utilisés pour la classification par nuages d'agents.	50
3.9	Déplacement d'une particule	52
3.10	Différentes topologies de voisinage	52
4.1	Représentation sous forme de Quadtree, (a) : Décomposition de l'image en blocs ; (b) : Diagramme hiérarchique de la décomposition.	62
4.2	Illustration de la méthode division-fusion, (a) : image originale, (b) : image de blocs déterminés par division, (c) : image segmentée.	63
4.3	Exemple de segmentation d'une image microscopique par K-means, inspirée de [166] : (a) : image originale, (b) : image segmentée en 2 classes, (c) : image segmentée en 3 classes ; (d) : image segmentée en 4 classes.	71

4.4	Exemples d'histogrammes, (a) : Histogramme bimodal, (b) : Histogramme multi-modal.	72
4.5	Exemple de seuillage à deux niveaux "manuel", avec un seuil = 103, (a) : Image macroscopique originale, (b) : Histogramme de l'image, (c) : Binarisation, (d) : Seuillage partiel.	74
4.6	Histogramme d'une image complexe	74
4.7	Exemple de seuillage à deux niveaux (a) : Image originale de grains de riz, (b) : image segmentée par Otsu, (c) : image segmentée par Kapur, (d) : Histogramme de l'image avec les valeurs de seuils trouvées par Otsu et Kapur.	76
5.1	Région de perception de l'agent	89
5.2	Le comportement d'un agent en fonction de son état et de sa perception.	90
5.3	Représentation de la région marquée et de la région de propagation de phéromone correspondante	92
5.4	Images de test	94
5.5	L'évolution du processus de construction des régions homogènes dans le temps.	96
5.6	Segmentation d'une image synthétique et influence de la croissance de taille de chaque région, en cours de construction, sur les comportements des agents.	97
5.7	Exemple d'images segmentées par notre approche.	99
5.8	Evolution de la durée de vie moyenne d'un agent pendant le processus de segmentation de l'image de synthèse "Image1".	100
6.1	Caractéristiques des images synthétiques de test.	109
6.2	Caractéristiques des images réelles de test (1)	110
6.3	Caractéristiques des images réelles de test (2)	111
6.4	Echantillons des images synthétiques segmentées résultant de <i>DMTBPSO</i> 113	
6.5	Echantillons des images réelles segmentées résultant de <i>DMTBPSO</i> (1)	114
6.6	Echantillons des images réelles segmentées résultant de <i>DMTBPSO</i> (2)	115

Liste des tableaux

3.1	Caractéristiques des principaux algorithmes évolutionnaires	39
5.1	Qualité de la segmentation des images de test par les méthodes SORG et RG.	101
6.1	Paramètres de DMTBPSO	112
6.2	Résultats de segmentation des images de test par DMTBPSO	115
6.3	Résultats computationnels pour les images de test (1)	117
6.4	Résultats computationnels pour les images de test (2)	118
6.5	Principe de fonctionnement des méthodes Otsu et Kapur(1)	119
6.6	Principe de fonctionnement des méthodes Otsu et Kapur (2)	120

Liste d'Abréviations

ACO	Ant Colony Optimisation Optimisation par colonie de fourmis
AE	Algorithmes Evolutionnaires
AG	Algorithme Génétique
API	Optimisation par stratégie de fourragement observé chez les fourmis "Pachycondyla APicalis"
AS	Ant System
ATC	Automatic Thresholding Criterion
BPSO	Binary Particle Swarm Optimization (Optimisation par essaim de particules binaires)
DMTBPSO	Dynamic Multilevel Thresholding using BPSO
HBMO	Honey Bee Mating Optimization d'accouplement des abeilles à miel
IA	Intelligence Artificielle
QPSO	Quantum Particle Swarm Optimization optimisation par essaim de particules quantiques
PSO	Particle Swarm Optimization optimisation par essaim de particules
PVC	Problème du Voyageur de Commerce
RN	Réseaux de Neurones
RG	Region Growing Croissance de région
SMA	Systèmes Multi-Agents
SORG	Self-Organizing Region Growing Croissance auto-organisée de régions
VA	Vie Artificielle

Première partie

Introduction générale

Chapitre 1

Introduction générale

1.1 Introduction

Plusieurs mécanismes dans les systèmes biologiques ont inspiré des ingénieurs et scientifiques pour développer des systèmes adaptatifs. Les notions de robustesse, d'émergence, d'auto-organisation et d'adaptabilité sont donc sous-jacentes dans ces systèmes et font même partie de leurs fondements.

Ce travail de thèse vise à transposer un certain nombre de ces mécanismes pour la résolution de problèmes de segmentation d'images.

1.2 Informatique bio-inspirée

L'informatique bio-inspirée ou naturo-inspirée s'intéresse à l'usage des systèmes naturels comme source d'inspiration pour le développement des systèmes informatiques. Une des sources de cette inspiration est apparue via l'observation des systèmes biologiques tels que : le cerveau, la génétique des populations, le système immunitaire, les écosystèmes, les colonies d'insectes sociaux...

Les activités de recherche sur cet axe, comprennent, tant l'intégration de méthodes bio-inspirées dans des systèmes existants, que la conception de solutions originales.

Le principe de base de développement d'un modèle résolvant un problème ou une catégorie de problèmes en s'inspirant des systèmes biologiques suit presque le même schéma de Arkin [7] présenté dans [159].

La première étape se base sur les études menées en biologie (par exemple en éthologie) et consiste à extraire de ces études un modèle réalisable du point de vue informatique. Puis, les résultats obtenus par des simulations permettent ensuite d'évaluer la qualité du modèle des études biologiques. Les conséquences peuvent être utiles pour les informaticiens afin d'améliorer le modèle lui-même, et aux éthologistes qui peuvent

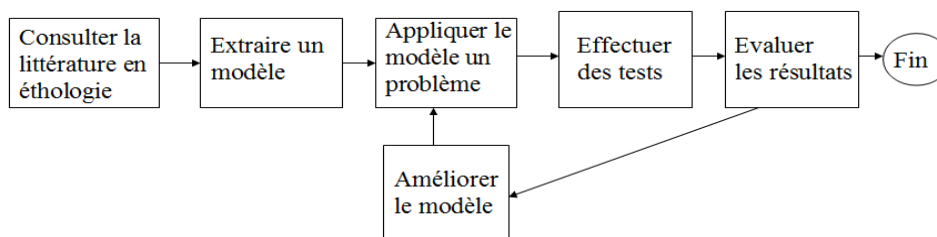


FIG. 1.1 – Schéma de Arkin concernant la méthodologie de conception en informatique bio-inspirée adapté de [159]

développer, et tester leurs théories du comportement animal.

L'émergence de cette discipline revient à plusieurs facteurs que nous allons les voir ci-dessous.

1.2.1 L'ambition de création des machines intelligentes

Depuis ses origines, l'informatique tente de s'inspirer des systèmes naturels, et en particulier de certains éléments des systèmes vivants. Dans les années 40, une première école de pensée concerne les systèmes vivants au travers des modèles informatiques qui est due aux travaux de Neumann, Wiener, Turing et McCulloch. John Von Neumann démontra que l'une des propriétés du vivant parmi les plus importantes, l'autoreproduction, peut être expliquée sans recourir à aucune "force de vie" mystique. Sa démonstration était basée sur une configuration d'automate cellulaire capable de se reproduire. Norbert Wiener [219] proposa le terme cybernétique, qu'il l'utilisa dans le sens du contrôle, de la régulation et de la communication dans l'être vivant et de la machine. Turing [215] proposa un test consistant à décider qu'une machine est intelligente si on peut la distinguer d'un être humain lors d'une conversation. McCulloch et Pitts proposèrent de modéliser la cellule nerveuse ou neurone par un automate à seuil dénommé le neurone formel. Quelques années plus tard, Rosenblatt eut l'idée d'organiser ces automates en réseau et conçut le Perceptron qui pouvait reconnaître des formes simples. Widrow conçut la machine Adaline qui fut utilisée pour la reconnaissance de parole.

Dans les années 50, les biologistes James Watson et Francis Crick découvrirent la structure en hélice de L'ADN (Acide Désoxiribo Nucléique) et mirent en évidence les mécanismes utilisés pour la reproduction, lesquels venaient confirmer le modèle de Von Neumann.

Les chercheurs en intelligence artificielle rejoignirent alors une nouvelle école de pensée, animée par Simon, Chomsky et McCarthy. Cette école de pensée faisant por-

ter ses efforts sur le calcul symbolique, basée sur l'hypothèse que le raisonnement modélisant la pensée est une combinatoire de symboles soumise à des règles logiques. Le terme intelligence artificielle fut utilisé pour la première fois dans une publication par John McCarthy en 1955.

A partir des années 60, les travaux visant à reproduire artificiellement certaines propriétés des êtres vivants se multiplièrent, nous pouvons citer les recherches sur les algorithmes évolutionnaires dues aux chercheurs : Holland [112], Rechenberg [184], Fogel [83]. Sans oublier de signaler les recherches sur les réseaux de neurones dues aux chercheurs Kohonen et Hopfield. Il convient également de noter l'explosion des recherches en intelligence artificielle après le développement des systèmes experts et la naissance des Systèmes Multi-Agents "*SMA*". Les systèmes experts sont des programmes informatiques capables de remplacer l'être humain dans des tâches réputées complexes et qui réclament de l'expérience, du savoir faire et une certaine forme de raisonnement. Le domaine *SMA* est né à la fin des années 70 et début des années 80, de l'idée de distribuer les connaissances et le contrôle dans les systèmes d'Intelligence Artificielle.

Toutes les tentatives de reproduire artificiellement certaines propriétés des êtres vivants ont conduit en 1987, à l'apparition officielle de la discipline nommée "Vie Artificielle" (*VA*). Sa "création" est alors motivée par l'incompréhension des processus complexes qui modélise la vie et le monde qui nous entoure. La vie artificielle propose une démarche en deux étapes [140]. Dans un premier temps, elle tente d'abstraire les principes fondamentaux du vivant, puis, dans un second temps, elle "réalise" ces principes sur un support arbitraire. Son principal objectif est la compréhension de la vie en essayant de généraliser les principes sous-jacents aux phénomènes biologiques et en récréant ces principes sur d'autres supports les rendant ainsi accessibles à nouveaux champs d'expérimentations. Dans ce contexte l'ordinateur représente un outil particulièrement intéressant.

A la fin des années 80, une autre école de pensée est née à partir de travaux dans le domaine de la robotique mobile et prise son inspiration des modèles cybernétiques et biologiques [8]; cette école a regroupé les travaux sur l'intelligence sans représentation : les systèmes Multi-agents situés [108], l'approche comportementale [33] et l'intelligence en essaim (swarm intelligence) [23], dont le principe s'appuie sur une démarche constructiviste de la cognition, selon une vision autopoïétique [151]. Dans cette vision, la cognition est la conséquence de la circularité et de la complexité de la forme de tout système dont le comportement inclut la maintenance de l'autoproduction de cette forme. Cette vision traduit un couplage structurel fort entre le système et son environnement [108].

1.2.2 Avancées technologiques

L'informatique évolue, tant du point de vue de la puissance de calcul que des langages de développement. Différents paradigmes apparaissent comme ceux de la vie artificielle, des automates cellulaires, de la programmation par objet ou des systèmes multi-agents, permettant de modéliser, de simuler des phénomènes biologiques et de compléter la simple preuve a priori fournie par une modélisation mathématique en autorisant des validations expérimentales a posteriori.

1.2.3 Complexité des systèmes informatiques

La progression réalisée dans les sciences et technologies modernes a conduit à une rapide augmentation de la complexité des systèmes créés par l'homme. Ces systèmes complexes artificiels se distinguent des systèmes "classiques" ou simples car ils sont composés de nombreux sous-systèmes hétérogènes et sont caractérisés par des comportements observables complexes. Ces comportements émergent du résultat des interactions entre les sous-systèmes qui les composent, rendant ainsi leur contrôle ardu : en effet, les interactions entre ces sous-systèmes peuvent avoir lieu à différents niveaux d'organisation ou d'abstraction du système, s'inscrivent dans le temps et dans l'espace et sont non linéaires [18].

Le développement de tels systèmes est devenu de nos jours un des objectifs majeurs de la science moderne, qui pousse de nombreuses disciplines à chercher l'inspiration dans les mécanismes des systèmes complexes biologiques déjà existants, tels que le cerveau, l'évolution des espèces, le système immunitaire, les écosystèmes, les colonies d'insectes sociaux.... Ces systèmes biologiques, composés pour certains de milliards de composants en interaction, semblent pouvoir fournir certaines clés d'une meilleure compréhension de la création de systèmes complexes artificiels.

1.3 Caractéristiques des systèmes biologiques

Différentes caractéristiques ont été identifiées dans les systèmes biologiques [132] : la robustesse, l'émergence, l'auto-organisation et l'adaptabilité

La robustesse est une des caractéristiques essentielles des systèmes biologiques. Ainsi, de nombreux auteurs ont montré que les systèmes biologiques sont capables de conserver les propriétés fonctionnelles du système malgré différentes perturbations et ceci peut être réalisé par différentes stratégies [189] : Premièrement, la mise en place de schémas de contrôle (par exemple la rétroaction) limite l'effet de la modification de paramètres cinétiques de voies de transmission. Deuxièmement, la redondance des gènes codant pour des fonctions biologiques importantes permet de conserver ces fonc-

tions en cas de mutation d'un exemplaire de ce gène. Enfin, une architecture modulaire telle que la cellule évite l'effondrement du système si l'un de ses modules est touché.

L'émergence est également un point important des systèmes biologiques. Elle est caractérisée par l'apparition d'une propriété dans un système alors qu'aucun composant de ce système ne contient ni n'exprime cette propriété. Selon Nicolis et Prigogine [168], on peut caractériser l'émergence par l'apparition des structures à un niveau d'organisation supérieur en ne définissant que des lois locales au niveau inférieur. L'expression "le tout est plus que la somme des parties" semble donc tout à fait adaptée pour décrire les systèmes biologiques.

La notion d'auto-organisation est une conséquence directe de la notion d'émergence [189]; ainsi l'auto-organisation ne serait que la résultante de l'ensemble des interactions existantes entre les entités d'un système. Camazine et ses collaborateurs [36] définissent l'auto-organisation dans les systèmes biologiques comme un processus dans lequel le pattern à un niveau global du système émerge d'une multitude d'interactions entre les composants des niveaux inférieurs. De plus, les règles déterminant les interactions qui ont lieu entre les composants du système sont réalisées uniquement à partir d'informations locales, sans référence au pattern global.

L'auto-organisation est donc, définie comme des processus anonymes, il s'agit de processus qui ne sont pas liés à l'identité des individus mais qui en sont au contraire indépendants. En effet, dans ce type de processus, les individus présentent des poids identiques et auront tous un même effet sur le système biologique considéré quel que soit leur statut sociodémographique au sein du groupe. Toutefois, ceci ne signifie en rien que la diversité des individus (comportementale, physiologique, ...) ne peut être intégrée à des processus auto-organisés mais elle apparaîtra, sur le plan de la dynamique d'un phénomène collectif, comme une "correction", cette correction pouvant avoir des conséquences très importantes sur la valeur adaptative des individus[156].

Trois niveaux d'adaptation peuvent être définis dans les systèmes biologiques. Premièrement, l'adaptation au niveau du système peut être associée à l'auto-organisation qui permet aux entités composant le système de s'adapter à leur environnement soit par spécialisation des fonctions (apprentissage), soit par modification de la topologie du groupe ce qui donne essence à un nouveau modèle d'organisation. En effet, par le biais de l'auto-organisation, un système doit conserver son adéquation fonctionnelle malgré les variations de son environnement. Deuxièmement, au niveau d'une espèce, l'adaptation se fait au fil des générations, au prix de nombreux essais. Par sélection naturelle, les plus aptes à se reproduire ont la chance de transmettre leurs gènes à leur descendance, et ainsi l'espèce reste adaptée à son environnement. Enfin, au niveau de l'individu est caractérisée par une capacité d'apprentissage. Une modification du comportement est souvent nécessaire afin de faire face aux aléas de la vie, une non

adaptation étant alors synonyme de mort certaine.

1.4 Segmentation d'images

Cette thèse aborde l'un des principaux problèmes de la vision par ordinateur : la segmentation d'images. Celle-ci est une opération de traitement d'images de bas niveau qui consiste à localiser dans une image les régions possédant une certaine uniformité pour une ou plusieurs caractéristiques (intensité, couleur, texture, ...) et qui sont différentes, pour au moins une de ses caractéristiques, pour des régions voisines. Cette opération est à la base de nombreuses applications tant en vision industrielle, qu'en imagerie médicale. Toutes les tâches ultérieures à ces applications, comme l'extraction de primitives, la détection d'une position, ou la reconnaissance d'un objet dépendent fortement de la qualité de la segmentation.

La segmentation d'images constitue, depuis quelques années, un axe important de recherche, c'est la conséquence de plusieurs éléments :

- la diversité des images et l'augmentation de leur utilisation ;
- l'augmentation de la puissance des ordinateurs qui autorise l'exploration de nouvelles approches et la mise en œuvre de techniques qui nécessitent de temps de calcul trop coûteux comme les techniques d'optimisation, et aussi facilite l'évaluation des résultats empiriques.
- la complexité du problème de segmentation ; la segmentation est un problème mal posé au sens de Hadamard [210] à cause de la non unicité de l'incertitude et de l'instabilité des solutions par rapport aux données d'entrées. C'est aussi un problème complexe, tant du point de vue de la formulation du problème [157] que de point de vue de choix de la technique de résolution qui est lié aux plusieurs facteurs tels que : les spécificités de l'image à traiter (présence de texture, présence de dégradations, non homogénéité de régions, ...), conditions d'acquisition, du type d'indices visuels à extraire ainsi que des contraintes d'exploitation [47].

Les recherches, dans ce domaine, ont donné un nombre considérable de méthodes [175]-[106], pour lesquelles plusieurs classifications ont été proposées, dans les différents ouvrages dédiés au traitement des images.

Ces méthodes peuvent à la fois faire appel à des principes heuristiques ou encore mathématiques. Parmi celles-ci, il existe une branche qui s'inspire plus spécialement de principes issus de la biologie. En effet, l'application des métaheuristiques d'optimisation, inspirées de l'évolution naturelle (algorithme génétique, ...) ou encore l'éthologie (colonies de fourmis, essaims de particules,...) [200]-[166], en segmentation d'images, a

connu un engouement particulier ces dernières années. Aussi, plusieurs travaux associent les systèmes multi-agents et le traitement d'images existant [152]. Le paradigme multi-agents a été utilisé dans deux formes différentes [30]. Dans le premier cas, l'approche multi-agents est utilisée comme un framework pour intégrer et coordonner les composants de traitement d'images [20]. Dans le deuxième cas, l'image est envisagée comme un environnement dans lequel les agents évoluent [149]-[29]-[61]-[59].

La segmentation d'images peut être réalisée en mode supervisé ou non supervisé. Dans le premier cas, le nombre de régions et leurs caractéristiques sont fournis par l'utilisateur, la segmentation est effectuée en déterminant, pour chaque site, la classe dont les propriétés se rapprochent le mieux de celles observées en ce site. Dans le second cas, le problème est plus complexe, car les informations nécessaires au processus de classification de pixels sont fournies de manière complètement automatique ; d'où la difficulté de ce type d'approche.

La segmentation d'images peut être, aussi, réalisée en appliquant un critère unique ou plusieurs critères ; l'utilisation d'un critère unique pour segmenter toute l'image peut engendrer de fausse segmentation, du fait que l'homogénéité, dans différentes parties de l'image, peut être à des degrés variables. Cependant l'utilisation des différents critères adaptés selon le contenu de l'image est un problème complexe et nécessite une connaissance a priori sur les objets constituant l'image qui n'est pas toujours disponible.

1.5 Contributions

Cette thèse part du constat simple que les systèmes biologiques possèdent des propriétés intéressantes permettant l'adaptation aux environnements fortement perturbés.

Les travaux menés dans le cadre de cette thèse portent sur différents aspects :

- L'étude de la capacité d'adaptation des systèmes biologiques et la compréhension de leurs mécanismes sous-jacents ;
- L'étude des systèmes bio-inspirés et la compréhension de leurs principes, afin de dresser un état de l'art dans ce domaine ;
- Fournir une preuve de concept en faveur de l'adéquation de ce type d'outils à la nature des applications de traitement d'images ;
- Mettre en place des approches (algorithmes), pour la résolution de la problématique de segmentation d'images, caractérisées par :
 - (a) La simplicité : les algorithmes développés ne codent pas les solutions aux problèmes posés, mais ils manipulent des données qui contiennent les solutions des problèmes ;

- (b) L'auto-organisation : du fait que les algorithmes proposés sont à base de population d'entités, définissant des interactions simples au niveau local, permettant l'émergence d'un comportement complexe au niveau global ;
- (c) La distribution : Elle offre intrinsèquement des algorithmes distribués qui peuvent employer le calcul parallèle tout à fait facilement.
- (d) La robustesse : les algorithmes développés montrent un niveau élevé de la robustesse au changement, en permettant aux entités de s'adapter dynamiquement aux changements globaux. D'un autre côté, la redondance de ces entités, leur interchangeabilité et la stochasticité de leurs items comportementaux permettent de s'affranchir de plusieurs situations inattendues et variées.

Nous décrirons en annexe, nos publications dans ce domaine.

1.6 Plan de la thèse

Comme, le sujet que nous venons d'évoquer, est extrêmement vaste, il est donc impensable d'en espérer un traitement exhaustif. Notre travail se focalisera sur trois parties, chacune d'entre elles étant associée à un ou plusieurs chapitres. Voici une description synthétique de leurs principales caractéristiques.

Partie 1 : Introduction générale

Introduction générale, qui est le présent chapitre, a pour but de définir le contexte scientifique sous lequel nos travaux ont été menés, d'explicitier la problématique au cœur de cette thèse et de faire la synthèse de nos contributions.

Partie 2 : dédiée aux éléments d'état de l'art, a pour but de présenter les différents domaines auxquels nous avons abordés dans le cadre de cette thèse. Elle comprend trois chapitres :

Le chapitre 2 qui est le plus biologique de la thèse, présente les mécanismes d'adaptation dans les systèmes biologiques à travers l'analyse de quelques exemples de ces systèmes. Nous insistons sur les principales caractéristiques de ces systèmes trouvant des développements artificiels.

Le chapitre 3 comporte un survol des modèles bio-inspirés pour la résolution des problèmes complexes, en mettant l'accent sur les mécanismes d'adaptation dans ces modèles. Nous tentons de regrouper les travaux portant sur les modèles bio-inspirés en trois catégories selon leurs sources d'inspiration : éthologique, évolutionnaire ou neuronal. Nous choisissons délibérément de restreindre la liste des modèles bio-inspirés présentés à des classes d'algorithmes parmi les plus connues, afin d'éviter une énumération

peu pertinente et forcément incomplète. Nous détaillons beaucoup plus, l'optimisation par essaim de particules qui représente un axe important dans nos travaux.

Le chapitre 4 présente des méthodes de segmentation d'images. L'accent sera mis sur le seuillage d'images, et de croissance de régions, notamment les méthodes avec lesquelles nous comparerons les performances de nos algorithmes. Nous présentons un état de l'art, pour ces deux méthodes, comportant les méthodes traditionnelles et de nouvelles méthodes qui sont basées sur des concepts bio-inspirés.

Partie 3 : Contributions à la segmentation d'images

Nos contributions consistent en l'utilisation des mécanismes bio-inspirés pour la segmentation d'images, elles seront présentées en deux chapitres :

Le chapitre 5 décrit notre première contribution qui consiste en la proposition d'une méthode de segmentation adaptative basée sur le concept d'agent et utilise une combinaison de deux techniques de l'intelligence en essaim, à savoir, l'Optimisation de l'Essaim de Particules (*PSO*) et la stigmergie, afin de résoudre les problèmes liés aux techniques de croissance de régions.

Le chapitre 6 présente notre deuxième contribution qui consiste en la proposition d'une méthode de segmentation non supervisée où le problème de segmentation par seuillage est traité comme un problème d'optimisation. Nous adaptons l'Optimisation par Essaim de Particules Binaires (*BPSO*) pour déterminer, automatiquement, le "nombre optimal" des seuils et chercher, simultanément, les seuils optimaux.

Le chapitre 7 est réservé à une conclusion générale du manuscrit, dans laquelle nous synthétisons nos contributions et proposons des perspectives sur la base des travaux effectués.

Deuxième partie

Etat de l'art

Chapitre 2

Mécanismes d'adaptation dans les systèmes biologiques

2.1 Introduction

L'une des motivations pour la modélisation des systèmes vivants est de reproduire artificiellement des propriétés biologiques du monde vivant liées à l'adaptation. Cette adaptation prend au moins trois formes en informatique :

La première considère un point de vu plutôt centré sur l'individu et sur la manière dont celui-ci peut apprendre au cours de sa vie, en fonction de ses conditions environnementales. Cette adaptation est caractérisée par une capacité d'apprentissage qui nécessite un système nerveux capable d'adaptation, ce que quasiment tous les animaux pluricellulaires possèdent.

La deuxième est plus axée du point de vue de l'espèce et des caractéristiques génétiques qui la composent. Elle se concentre sur la modélisation des théories de l'évolution et sur la manière dont elles peuvent améliorer les individus en les faisant évoluer.

Finalement, la dernière met l'accent sur les phénomènes collectifs et sociaux pouvant apparaître dans les groupes d'individus, qui peuvent être associées aux mécanismes d'auto-organisation.

L'objectif de ce chapitre est de présenter différents aspects du vivant ayant poussé la communauté scientifique à les imiter pour la résolution de problèmes d'ingénierie. Nous verrons qu'il s'agit toujours de systèmes adaptatifs, autant dans le sens d'apprentissage et d'évolution que dans le sens d'auto-organisation. Cet aperçu comporte la présentation du système nerveux, la description des systèmes évolutionnaires vivants, et enfin, l'étude des phénomènes collectifs et sociaux observés généralement chez les insectes sociaux. Bien que cette étude soit trop succincte en regard des travaux menés

dans l'analyse des comportements collectifs observés dans ces systèmes.

2.2 Système nerveux

Le cerveau, en particulier, le cerveau humain est la machine de traitement de l'information la plus sophistiquée que nous connaissions. En effet, même si nous sommes incapables de réaliser de calculs avec la rapidité et la précision d'un ordinateur, nous sommes autrement mieux adaptés que ce dernier aux problèmes de traitement de l'information dans son ensemble : perception, traitement de signal, extraction de caractéristiques pertinentes dans un monde complexe, et décision, sont des tâches que nous accomplissons couramment. Ces tâches sont accomplies avec une grande rapidité, et le cerveau adapte son comportement aux situations nouvelles, principalement selon deux processus [4] :

- l'adaptation rapide et automatique (réflexe) ; c'est le cas de notre pupille, par exemple, qui adapte son diamètre en fonction de la quantité de lumière.
- L'apprentissage ; cet apprentissage autorise une adaptation lente d'un individu à l'exécution d'une tâche nouvelle, par exemple, la conduite automobile, ou l'utilisation d'une nouvelle théorie des mathématiques.

Cet apprentissage peut avoir des effets à long terme sur les processus d'adaptation rapides et automatiques : le sportif de haut niveau répète inlassablement les mêmes gestes des milliers de fois, jusqu'à ce que ces derniers ne procèdent plus d'une action consciente. En d'autres termes, il se fabrique de nouveaux réflexes par apprentissage.

De même, le mathématicien exécute souvent des exercices mettant en œuvre les concepts de telle ou telle théorie, de manière à mieux intégrer ces concepts et être ainsi plus efficace sur la résolution de futurs problèmes. Il se fabrique de l'intuition, ou peut être des réflexes par apprentissage.

Toutes ces constatations ont constitué un ensemble de motivations importants poussant à analyser le fonctionnement de cette machine "cerveau".

Le cerveau est une machine très complexe. Le cortex cérébral dans les êtres humains contient approximativement 10^{11} neurones qui sont sur l'ordre du nombre d'étoiles dans notre galaxie et chaque neurone est connecté à 10^3 à 10^4 autres neurones. Dans le total, le cerveau humain contient 10^{14} à 10^{15} interconnexions approximativement [118].

Les neurones sont des cellules particulières ayant quatre fonctions, à savoir de transmettre de l'information extérieure ou intérieure, analyser cette information, et potentiellement la mémoriser. Ils servent à récupérer les données sensorielles telles que la lumière pénétrant les yeux ou le toucher sur la peau, à traiter et stocker cette

information, et à agir sur les muscles de manière à faire se mouvoir le corps. Bien que les neurones ne soient pas tous identiques, leur forme et certaines caractéristiques permettent de les répartir en quelques grandes classes (FIG. 2.1) :

- Le corps cellulaire ; Il contient le noyau du neurone ainsi que la machine biochimique nécessaire à la synthèse des enzymes. Ce corps cellulaire de forme sphérique ou pyramidale contient aussi les autres molécules essentielles à la vie de la cellule. Sa taille est de quelques microns de diamètre.
- Les dendrites : Ce sont de fines extensions tubulaires qui se ramifient autour du neurone et forment une sorte de vaste arborescence. Les signaux envoyés au neurone sont captés par les dendrites. Leur taille est de quelques dizaines de microns de longueur.
- L'axone : C'est le long de l'axone que les signaux partent du neurone. Contrairement aux dendrites qui se ramifient autour du neurone, l'axone est plus long et se ramifie à son extrémité là où il se connecte aux dendrites des autres neurones. Sa taille peut varier entre quelques millimètres à plusieurs mètres.

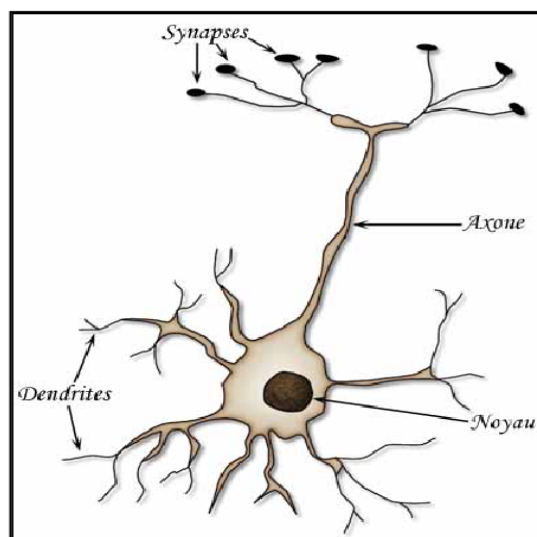


FIG. 2.1 – L'anatomie du neurone biologique

L'influx nerveux est assimilable à un signal électrique, se propageant dans les neurones de la manière suivante :

- Les dendrites reçoivent l'influx nerveux d'autres neurones.
- Le neurone évalue alors l'ensemble de la stimulation qu'il reçoit (c'est à dire sa dépolarisation par rapport à l'extérieur).

- En fonction de cette stimulation, le neurone transmet ou non un signal de type « tout ou rien » le long de son axone, selon une fréquence fonction du niveau de dépolarisation. On dira alors que le neurone est ou non excité.
- L'excitation du neurone est propagée le long de l'axone jusqu'aux autres neurones ou fibres musculaires qui y sont connectés via les synapses.

Le traitement très simple réalisé par chaque neurone indique que l'information n'est pas stockée dans les neurones, mais est plutôt le résultat du comportement de toute la structure interconnectée. L'information est, principalement, dans l'architecture des connexions et dans la force de ces connexions.

Cette description très brève de la physiologie du neurone a inspiré des ingénieurs et scientifique pour développer des systèmes adaptatifs avec des capacités d'apprentissage. Dans le chapitre suivant, nous décrirons les principaux computationnels modèles qui ont été développés ainsi, comme de résultat de telle inspiration biologique.

2.3 Systèmes évolutionnaires vivants

L'évolution a donné lieu à une extraordinaire diversité de formes vivantes aux capacités étonnantes. Différents organismes, adaptés aux conditions particulières de leur environnement, forment des populations coopérantes ou en compétition qui évoluent au gré de variations et de la sélection.

Le plan de la croissance des organismes, est stocké dans les chromosomes, qui sont composés d'*ADN* (Acide Désoxyribo Nucléique), varie de génération en génération. Les individus qui finissent par prévaloir sont ceux dont les capacités spécifiques s'accordent le mieux aux conditions environnementales.

L'*ADN* d'une cellule contient l'information décrivant la structure de l'organisme ; c'est une molécule formée d'un ensemble de mononucléotides, composés d'un atome de phosphore, d'un hydrate de carbone et d'une base. Les deux premiers éléments sont essentiellement structurels, alors que la base permet le codage de l'information. Il y a quatre types de bases : adénine (*A*), guanine (*G*), cytosine (*C*) et thymine (*T*). Les mononucléotides s'assemblent pour former une chaîne, appelée polynucléotide. Une molécule d'*ADN* consiste en deux polynucléotides enroulées pour former une double hélice. L'adénine se lie toujours avec la thymine, et la guanine avec la cytosine ; les deux polynucléotides sont donc deux chaînes complémentaires et il suffit de connaître une seule des deux branches pour reconstruire toute l'information (FIG. 2.2).

L'information codée dans le génome contrôle la synthèse des protéines. Une protéine est composée d'acides aminés. Un gène est une partie d'*ADN* (entre 100 et 5000 bases) codant l'ensemble d'acides aminés d'une protéine donnée. Il y a 20 types d'acides

aminés. Une séquence de trois bases est donc suffisante pour les coder : cette séquence de trois nucléotides est appelée un codon. Etant donné qu'il y a 64 codons possibles, pour seulement 20 acides aminés, il existe une certaine redondance dans le codage. De plus, certains codons servent de signaux de contrôle. Une protéine n'est pas construite directement à partir du gène, mais en passant par une formation intermédiaire d'*ARN* (Acide Ribo Nucléique).

ARN Macromolécule de structure semblable à l'*ADN* mais ne possédant qu'un brin. Elle est utilisée soit directement dans les mécanismes cellulaires (souvent des mécanismes apparus avant ceux assurés par des protéines) soit comme intermédiaire dans la production des protéines à partir de l'*ADN* (*ARN* messager) [170].

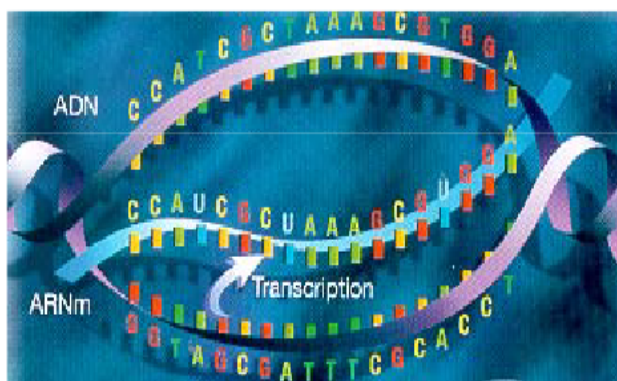


FIG. 2.2 – Structure de l'*ADN*

Les premiers principes de la théorie de l'évolution a été évoqués par Darwin qui a montré, dans son livre « De l'origine des espèces par voies de sélection Naturelle » [51], que l'apparition d'espèces distinctes se fait par le biais de la sélection naturelle de variations individuelles. Cette sélection naturelle est fondée sur la lutte pour la vie, due à une population tendant naturellement à s'étendre mais disposant d'un espace et de ressources finies. Il en résulte que les individus les plus adaptés tendent à survivre plus longtemps et à se reproduire plus aisément. Le terme " adapté " se réfère à l'environnement, que l'on peut définir comme étant l'ensemble des conditions externes à un individu, ce qui inclut les autres individus. Les lois de variation (croisements et mutations) furent expliquées plus tard par la théorie de l'hérédité de Mendel [155], puis par la synthèse moderne grâce à la découverte de l'*ADN*.

L'idée centrale de la synthèse moderne est l'adaptation au milieu. Celle-ci fait intervenir la notion de valeur d'adaptation (*fitness* en anglais). Lorsqu'un gène est modifié et qu'il améliore la survie de l'espèce, alors il augmente sa valeur d'adaptation. Cette valeur est souvent liée à "l'abondance" d'une population, qui peut être quantifiée

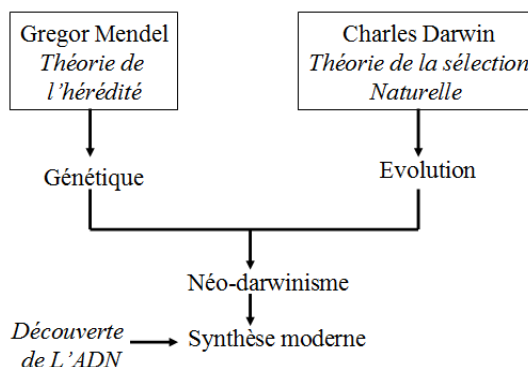


FIG. 2.3 – Synthèse moderne.

à l'évaluation de la descendance supplémentaire d'un individu ou, plus globalement, comme l'effet du gène sur le taux de croissance de population. Dans certaines théories "adaptationnistes", la valeur d'adaptation est également assimilée à la "persistance", c'est-à-dire la probabilité d'éviter l'extinction de l'espèce.

La théorie de l'évolution par sélection naturelle explique l'une des caractéristiques parmi les plus surprenantes du vivant : les organismes n'ont pas été "programmés" ou "construits" pour répondre à un problème spécifique. Ils évoluent et s'adaptent selon les nécessités que l'environnement leur impose.

Sur ce principe, quelques concepts et des stratégies informatiques ont été développés pour résoudre des tâches d'apprentissage et pour traiter des problèmes d'optimisation dans les systèmes artificiels que nous allons les voir dans le chapitre suivant.

2.4 L'auto-organisation et les phénomènes collectifs et sociaux

Ces dernières années, des travaux ont mis en évidence l'existence dans la nature de nombreux systèmes biologiques capables d'accomplir des tâches difficiles, dans des environnements variés, et ceci sans pilotage ni contrôle externe et sans coordination centrale. C'est le cas par exemple de la formation de pistes chez les fourmis, de la construction des termitières, de la coordination des bancs de poissons, ou encore des oiseaux ou autres vertébrés en déplacement collectif.

L'adaptation de ces systèmes à leurs environnements peut être associée à l'auto-organisation ; l'environnement agit sur l'organisation d'un système, et tout changement de l'environnement peut modifier l'organisation de ce système.

L'auto-organisation dans les systèmes biologiques peut être mise en œuvre selon

les processus de : stigmergie, renforcement ou de coopération. Ces processus sont à l'origine de ce que l'on dénomme, l'intelligence collective. On parle de l'intelligence collective quand un groupe social peut résoudre un problème dans un cas où un agent isolé en serait incapable.

En informatique les modélisations directement issues de ces comportements sont caractérisées par une "intelligence en essaim" ("swarm intelligence") [23] que nous allons la présenter dans le chapitre suivant.

Dans cette section nous parlerons de l'auto-organisation et de ses formes à travers d'analyse de certains phénomènes collectifs observés dans les systèmes sociaux.

2.4.1 L'auto-organisation

Le concept d'auto-organisation concerne une large gamme de processus liés à la formation de patterns, dans des systèmes biologiques allant des organismes unicellulaires à l'homme (voir [36] pour une revue), mais également dans des systèmes physiques ou chimiques avec, par exemple, les ondulations formées par le vent à la surface d'une dune ou encore certains patterns de convection [36]. La caractéristique commune de ces systèmes est la manière par laquelle ils acquièrent leur ordre et leur structure.

Une définition claire a été proposée par Camazine [36] : "l'auto-organisation caractérise un processus au cours duquel une structure émerge au niveau global uniquement d'un grand nombre d'interactions entre les composants de niveau local du système. De plus, les règles spécifiant les interactions entre composants du système sont suivies en utilisant uniquement des informations locales, sans référence à la structure globale".

Deux termes sont à préciser pour une bonne compréhension, "structure" et "émerger". Le mot structure est une traduction approximative du mot anglais "pattern", qui peut signifier : configuration générale, forme, schéma, type [154]. D'une manière générale, il s'applique à un "arrangement organisé d'objets dans l'espace ou le temps". Une propriété émergente d'un système est quant à elle une caractéristique qui apparaît "à l'improviste" (sans avoir été explicitement déterminée), de par les interactions entre les composants de ce système. On parlera donc d'émergence pour souligner le caractère non-déterminé d'une propriété, sans faire référence au fait que cette propriété soit organisée en structure ni qu'elle soit située à un niveau différent des interactions [73].

Les modes d'interactions qui ont lieu entre les unités d'un système auto-organisé sont de deux types : les feedbacks (rétroactions) positifs et les feedbacks négatifs.

Les rétroactions positives sont des processus dont le résultat renforce l'action, par exemple, par amplification, facilitation, auto-catalyse, etc. Les rétroactions positives sont capables d'amplifier les fluctuations du système, permettant la mise à jour d'informations peu apparentes. De tels processus peuvent facilement entraîner une divergence

du système, s'ils ne sont pas maintenus sous contrôle par des rétroactions négatives, qui jouent ainsi le rôle de stabilisateurs du système. Lorsqu'ils sont couplés, de tels processus de rétroaction sont de puissants générateurs de modèles [36].

Dans un système biologique, les objets seront par exemple, les individus d'un groupe social, le phénomène collectif sera considéré comme le pattern émergent alors que les rétroactions sont très souvent mises en oeuvre par le biais de la communication indirecte (stigmergie), de coopération et des mécanismes de renforcement.

2.4.2 Déplacement collectif

Le déplacement est un des phénomènes collectifs les plus étudiés. Il est observé dans de nombreuses espèces, plus particulièrement dans les espèces vivants en groupes organisés, présente dans plusieurs embranchements du règne animal. On trouve en effet, des déplacements collectifs aussi bien chez des invertébrés comme les insectes que chez des vertébrés comme les poissons, les oiseaux ou encore les mammifères, incluant l'espèce humaine.

L'effet de tels groupements et déplacements de masse a été largement discuté dans la littérature [156]-[3]-[2]-[211]-[218] ; On observe par exemple, que des nuages d'oiseaux peuvent se former et évoluer de manière très spectaculaire et utile pour les individus ayant ce comportement : l'effet de masse peut repousser des prédateurs éventuels ou alors certaines configurations précises permettent d'économiser de l'énergie (exemple du vol des canards).

Des comportements similaires se retrouvent dans la nage collective de certaines espèces de poissons, pour la chasse en groupe (par exemple banc de thons) où l'essaim, prend la forme d'un entonnoir ou pour échapper à un ennemi avec des mouvements d'explosion, ou encore dans le vol en groupe chez des insectes (criquets).

Enfin, le déplacement des poussins en train de picorer présente une forme simple de mouvement auto-organisé qui leur permet de rester à une certaine distance les uns des autres grâce aux signaux que constituent leurs pépiements.

Ces comportements collectifs apparaissent comme des patterns spatio-temporels complexes et coordonnés. Les mécanismes sous-jacents aux ces comportements collectifs ont été bien discutés dans [156].

Les points communs de ces diverses espèces sont exactement ceux qui caractérisent l'intelligence collective [18] :

- les individus aiment la proximité de leurs semblables et tendent à être grégaires car ils obtiennent un avantage substantiel à chasser, se déplacer ou vivre en groupe ;

- ils interagissent de manière locale par le moyen de signaux (grognement, phéromones, attitudes) ;
- l'individu seul répond instinctivement à certains stimuli. La coordination du groupe est implicite et se fait au travers de règles comportementales très simples au niveau individuel.

2.4.3 Division du travail et l'organisation des rôles sociaux

Les fourmis (ainsi que les abeilles ou les guêpes) ont la faculté de répartir dynamiquement les tâches en fonction des besoins de la colonie, et ce de manière totalement distribuée et sans aucun contrôle central. Les tâches que doivent accomplir les ouvrières sont en effet multiples [159] :

- La défense du nid ;
- La recherche de nourriture ;
- L'entretien et la construction du nid ;
- L'entretien des larves et leur approvisionnement en nourriture.

La division des tâches chez ces insectes sociaux est totalement exilée : le nombre d'individus appartenant aux différentes castes et la nature des tâches à accomplir sont sujets à des changements constants durant la vie de la colonie [97]. La proportion d'individus se consacrant à une tâche donnée varie en fonction des perturbations internes du nid et de celles, globales, de l'environnement.

Dans la mesure du possible, ce sont les individus les plus adaptés qui adoptent une tâche donnée, mais dans des cas extrêmes des individus de castes morphologiques inadaptées peuvent assumer certaines tâches. Par exemple, chez la fourmi *dimorphique* *Pheidole*, les ouvrières minor sont plus petites et morphologiquement différentes des ouvrières major. Tandis que les major découpent les grosses proies avec leurs larges mandibules ou défendent le nid, les minor le nettoient ou nourrissent le couvain. La suppression des ouvrières minor stimule les majors en les poussant à assumer, en l'espace de deux heures, l'accomplissement des tâches habituellement effectuées par les minor [220]. De manière plus générale, cette flexibilité a été observée chez beaucoup d'espèces d'insectes ; la suppression d'une caste ouvrière est rapidement compensée par les autres castes.

Dans cet exemple les processus auto-organisés peuvent jouer un rôle essentiel dans l'émergence de spécialisation individuelle ou de différenciation sociale.

2.4.4 Construction collective des nids

Les insectes organisés en sociétés, abeilles, fourmis, guêpes et termites, sont capables de construire collectivement des nids d'une remarquable complexité, au regard de la relative simplicité des individus. La construction du nid est un symbole de l'organisation distribuée de ces insectes. Le nid ne pourrait exister sans une forme de coordination entre les individus. Une telle coordination pourrait reposer sur l'architecture elle-même des constructions plutôt que sur des interactions directes entre les membres de la colonie. Pour montrer ce type de construction et l'organisation distribuée qui est sous-jacente, nous prenons les nids de : guêpes, de fourmis et de termites, comme exemples

Nid de guêpes

Les guêpes tropicales (par exemple, *Parachartergus*, *Epipona*) construisent des nids complexes, comprenant une enveloppe externe et des rayons de cellules régulièrement organisés liées l'une à l'autre par un périphérique ou par un trou d'entrée central (FIG. 2.4). L'une des caractéristiques de leurs nids est leur modularité. La répétition d'une même structure de base constitue en effet la manière la plus simple d'accroître la taille du nid. L'organisation des activités individuelles de construction s'effectue fondamentalement au travers des configurations de matière qui sont rencontrées sur le nid [23].



FIG. 2.4 – Nids de guêpes

Le processus de construction débute par la formation d'un pédicelle (attache) reliant le nid à son support, puis d'une première cellule dans son prolongement. Au

fur et à mesure que la construction progresse, le nombre de sites potentiels, susceptibles de recevoir une nouvelle cellule, augmente. Plusieurs activités peuvent alors être accomplies parallèlement, la construction ne suivant pas une séquence a priori bien définie. Cette capacité de construire simultanément, en plusieurs sites différents d'un même nid, constitue une étape importante dans l'évolution et la création d'architectures complexes. Mais elle a introduit en retour de nouvelles contraintes, puisque les informations locales qui régulent le comportement bâtisseur doivent être organisées dans l'espace et dans le temps de manière à assurer une construction collective cohérente. Ce type de contraintes a été étudié, par Theraulaz et Bonabeau [209], au moyen d'un modèle formel, dit "essais sur réseau", où les guêpes sont représentées par des agents qui se déplacent aléatoirement sur un réseau cubique tridimensionnel et ne possèdent aucune représentation globale de l'architecture qu'ils construisent. Ils ne peuvent percevoir que les configurations locales de matière qui les entourent, c'est-à-dire l'état des 26 cellules voisines. Seules certaines configurations géométriques ont été choisies comme stimulantes dans ce modèle et déclenchent le dépôt d'une brique. Les nids sont construits au moyen de briques élémentaires, choisies parmi deux types possibles, bleue ou rouge. A tout instant, les agents décident ou non de déposer une brique en se référant à une table de comportement constituée de l'ensemble des configurations locales stimulantes et qui indique le type de brique à déposer pour chacune de ces configurations.

Nid de fourmis

Chez la fourmi *Oecophylla*, les ouvrières forment des chaînes avec leurs propre corps, ce qui leur permet de franchir de larges espaces et même de plier littéralement une feuille rigide afin de construire un nid (FIG. 2.5) ; plusieurs chaînes de fourmis se rejoignent pour en former une plus importante pente sur laquelle les ouvrières peuvent aller et venir, de telles chaînes créent suffisamment de tension pour que les bords de la feuille se rapprochent, et lorsque ceux-ci sont correctement disposés les fourmis les soudent à l'aide d'un fil de soie sécrété par une larve «manipulée» par une ouvrière [114].

Nid de termites

Les nids de termites sont grands et complexes. Un nid peut être autant de 10^4 ou 10^5 fois aussi grand qu'une termite individuelle [22] ; une proportion incomparable dans le royaume animal. Leurs édifices prennent des formes variées et parfois surprenantes : colonnes, cathédrales, châteaux forts ou pyramides (FIG. 2.6).

Les nids de la termite africaine obtiennent une avance famille que *Macrotermitinae*

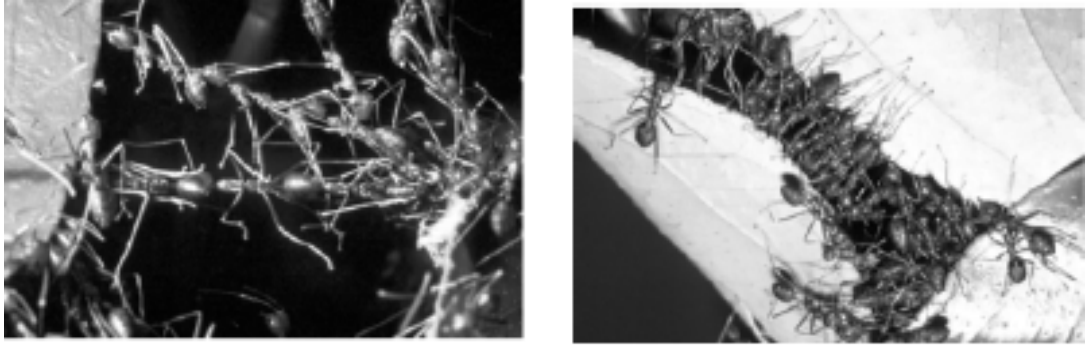


FIG. 2.5 – Formation de chaînes chez les fourmis tisserandes *Oecophylla longinoda*



FIG. 2.6 – Nid de termites

sont composés de beaucoup d'infrastructures, tel que remparts protecteurs, que les pillards couvent chambres, prises d'air du refroidissement en spirale, galeries de jardins de la moisissure, et des chambres royales. Pour toute la sophistication architecturale de nids de la termite, les termites elles-mêmes sont aveugles, faibles et apparemment pas sensible à un coordonnant autorité [22], ce qui fait apparaître un mécanisme curieux pour plusieurs chercheurs.

Dès la fin des années 1950, Pierre-Paul Grassé [102] avait montré que la clef réside dans l'étude des règles simples que semblent suivre les termites :

1. les termites se déplacent aléatoirement, déposant des granules composées d'un mélange de terre et de salive sur toute partie du sol légèrement sur élevée. Ainsi, de petit tas de terre moite commencent à se former ;
2. ces tas encouragent les termites à concentrer leurs dépôts et rapidement des tas plus importants se développent, formant des colonnes qui seront érigées jusqu'à une certaine hauteur (hauteur spécifique à l'espèce) ;
3. finalement, si une colonne est suffisamment proche d'autres colonnes, les termites la gravissent et dirigent leur construction vers ces colonnes proches.

Evidemment, ceci n'explique pas tout, mais ces observations mènent déjà au concept clef que l'on retrouve chez les insectes sociaux : l'action des termites n'est pas coordonnée, aucun des individus ne possède de "plan" à réaliser mais réagit plutôt aux modifications de son environnement, modifications auxquelles la population contribue. Grassé désigna par "stigmergie" [102] ce phénomène, qui se retrouvera finalement dans beaucoup d'activités collectives, à différentes échelles et dans divers domaines.

Le premier modèle dans le cadre de théorie de l'ordre de fluctuation porte sur les étapes initiales de la construction du nid chez les termites [67]. Il a été étendu par Kugler et Turvey [136] ; Ce modèle montre le cycle de construction de la termitière : ce cycle commence par des dépôts aléatoires, puis progresse par l'émergence de piliers, d'arches puis de coupes (FIG. 2.7). Le cycle recommence alors par des dépôts aléatoires. L'apparition des piliers dans une termitière pouvant être expliquée par l'amplification de multiples fluctuations chaotiques : la structure, modèle d'équilibre des forces par sa stabilité, naît de l'amplification de multiples déséquilibres.

A partir de ce modèle, quelques observations ont été notées [37] :

- lorsque le nombre de termites participant à l'élaboration du nid est faible, la concentration en phéromone est telle qu'il ne peut influencer le comportement d'autres insectes avant de disparaître. La phase de dépôt aléatoire persiste alors aussi longtemps que le nombre de termites demeure peu élevé ;
- au fur et à mesure que le nombre d'insectes participant à la construction augmente, l'accroissement du taux de dépôts sur des sites préférentiels accroît la

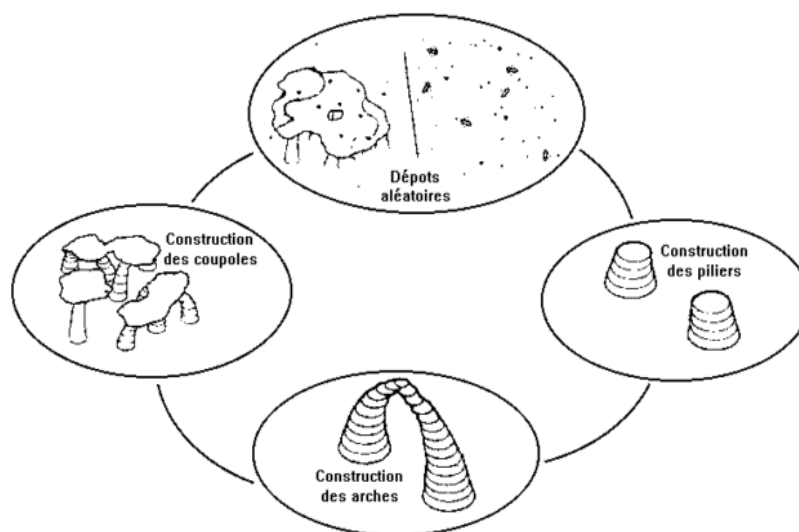


FIG. 2.7 – Le cycle de construction de la termitière (d'après [136]).

concentration en phéromone ce qui attire alors de plus en plus d'insectes, d'où le taux de dépôts... ainsi de suite jusqu'à atteindre un seuil où l'apparition de "patterns" traduit un mouvement coordonné des termites bâtisseuses. Le résultat est l'émergence de piliers due à une amplification rapide des dépôts de boulettes aux points de plus haute concentration en phéromone.

- L'auto-organisation repose sur la coopération entre termites, qui est mise en oeuvre par le biais de la communication indirecte "stigmergie", et aussi sur le mécanisme de rétroaction positive (auto-catalyse) permettant d'amplifier la construction de piliers puis d'arches. Le chaos intervient au tout début de la construction lors du dépôt des premières boulettes ; malgré ces fluctuations aléatoires, la régularité de la structure produite apparaît spontanément dans le système sans qu'aucune "mesure de distance" entre deux piliers ne soit nécessaire aux insectes pour la produire.

2.4.5 Tri du couvain

Le rassemblement des corps sous forme de tas "cimetières" afin de nettoyer le nid, et l'organisation spatiale de divers éléments du couvain : les oeufs, les larves et les nymphes, sont deux phénomènes observés chez plusieurs espèces de fourmis. De nombreuses expériences ont été menées afin de mettre en évidence ces deux phénomènes : la formation de cimetières [69]-[42]-[23] et le tri des couvains : [87]-[69]-[23]-[195]. Les mécanismes sous-jacents dans ces deux phénomènes ont été bien discutés dans [97].

Deneubourg et al. [68] ont proposé deux modèles proches, censés représenter les deux mécanismes ces deux phénomènes : agrégation des larves et ségrégation par type. L'idée générale est que les fourmis utilisent un modèle de règles relativement simple comme :

- Lorsqu'une fourmi rencontre un élément du couvain, la probabilité qu'elle s'en empare est d'autant plus grande que cet élément est isolé ;
- Lorsqu'une fourmi transporte un élément du couvain, elle le dépose avec une probabilité d'autant plus grande que la densité d'éléments du même type dans le voisinage est grande.

2.4.6 Recrutement et l'exploitation collective de sources de nourriture

La recherche de nourriture (le fourragement) est une activité essentielle à la vie de la colonie des insectes, elle est souvent dispersée spatialement. La manière dont celle-ci est organisée, chez les fourmis ou les abeilles par exemple, est très variable. Le fourragement met à jour des stratégies qui permettent aux insectes une grande adaptation à leur milieu. Par exemple, chez les fourmis, plusieurs stratégies d'exploitation des sources de nourritures ont été identifiées [21]. Nous examinerons, dans ce qui suit, deux stratégies de fourragement, la première se base sur la capacité individuelle sur et l'autre sur les pistes chimiques.

Fourragement solitaire

Le fourragement solitaire se trouve particulièrement chez les espèces peu peuplées qui développent des mécanismes d'apprentissage plus évolués [115], cas d'une espèce "*Pachycondyla apicalis*" à laquelle, une étude très complète est consacrée dans [89]. Il s'agit d'une espèce diurne chassant de petits insectes et de petites larves. Les récolteuses se répartissent en deux catégories, les unes demeurent au voisinage de nid, et récoltent ensemble dans la même zone où la densité de population est forte ; les autres au contraire visitent un ensemble de territoires éloignés du nid, chaque récolteuse ayant son territoire de chasse propre, sans qu'il y ait recouvrement entre ces territoires (FIG. 2.8).

Une telle organisation se produit sans qu'il y ait de communication directe entre les individus. Elle est liée à la nature de proies recherchées [159] ; la présence des proies étant relativement aléatoire, la capture d'une proie ne donne que peu d'information sur la stabilité spatiale de la source de nourriture. Autrement dit, la probabilité de retrouver une proie dans la même zone qu'une précédente capture n'est pas suffisante pour y canaliser les forces de fourragement de la colonie. Cependant, d'un point de

vue individuel, les ouvrières mémorisent leurs sites de capture et lors de leur prochaine sortie du nid, elles retournent systématiquement sur le dernier site de chasse fructueux. Cette spécialisation sectorielle est une réponse pour l'adaptation nécessaire à la découverte et l'exploitation de sources de nourriture.

Le comportement de fourrageur de *Pachycondyla apicalis* peut être résumé en trois règles [89] :

1. la découverte d'une proie entraîne toujours le retour sur ce site lors de la sortie suivante, c'est là que la fourrageuse reprend ses nouvelles prospections ;
2. la découverte d'une proie pèse sur la décision de sortie de fourrageuses en réduisant l'intervalle de temps qu'elles passent au nid ;
3. les fourrageuses semblent progressivement apprendre une association entre une direction de recherche opposée au nid et l'augmentation de la probabilité de succès.

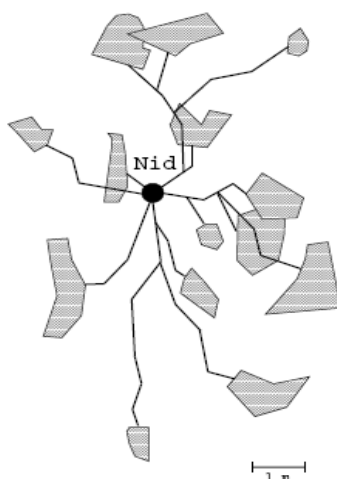


FIG. 2.8 – Exemple (fictif) de carte des trajets et aires de récolte des fourrageuses *Pachycondyla apicalis* (inspiré de [89]).

L'intérêt de la stratégie de fourrageur des fourmis *Pachycondyla apicalis* réside dans sa simplicité et la bonne couverture de l'espace de recherche qui en résulte. On a ici, un phénomène d'émergence : de règles de recherche simples et individuelles, qui ne tiennent pas compte du travail des autres fourmis, on obtient une exploration radiale de l'espace centrée sur le nid.

Fourragement collectif

Le fourragement collectif caractérise les espèces de fourmis, par exemple : *Lasius niger*, *Itridomyrmex humilis*, *Pheidole pallidula* et *Argentine*, dont le recrutement s'effectue essentiellement par piste chimique [21]. Ces espèces ont été soumises à des situations expérimentales [12]-[101]-[69] : dans une première phase, la recherche de sources de nourriture, se déroule de façon plus ou moins aléatoire mais, lorsqu'une fourmi découvre une source de nourriture, elle retourne au nid en déposant des phéromones le long du chemin parcouru, grâce à une glande située dans son abdomen, et former ainsi des pistes odorantes, cette piste de phéromones entraîne le recrutement d'autres individus et peut de les guider, qui eux-mêmes, après s'être alimentés, renforcent le marquage de la piste lors du retour au nid et recrutent à leur tour. Ce phénomène tend à s'inverser au fur et à mesure de l'épuisement de la ressource car les fourmis récoltent et recrutent de moins en moins.

Malgré le marquage, certaines fourmis arrivent à se perdre. Le système de recrutement peut alors paraître inefficace, mais il n'en est rien car les fourmis perdues, en explorant aléatoirement l'espace, vont alors peut être trouver de nouvelles sources de nourriture et ainsi recruter à leur tour.

Des études ont été faites concernant les états d'équilibre vers lesquels tend un système qui repose sur un tel recrutement [101]-[69]. Si on suppose un système comportant deux ressources de quantité égale, trois types de comportements peuvent être observés en faisant varier le nombre de fourmis :

- aucun recrutement n'est possible lorsque le nombre de récolteuses est trop faible ;
- un recrutement apparaît à partir d'une valeur critique correspondant à un certain nombre de fourmis. Ce recrutement conduit à une exploitation symétrique des deux sources et se maintient jusqu'à ce qu'une nouvelle valeur critique du nombre d'individus apparaisse, auquel cas un nouveau comportement collectif est observé ;
- lorsque le nombre de fourmis est encore plus important, une bifurcation se produit et la seule situation stable est alors une exploitation asymétrique des deux ressources, l'une étant fortement exploitée, l'autre faiblement. Cette distinction est d'autant plus flagrante que le nombre de fourmis croît.

Cet exemple montre que le marquage phéromonal introduit un certain ordre au sein de la colonie, appelé "ordre par fluctuation" [37]. Il permet de décrire l'évolution du système vers un nouveau régime dynamique où existent un ordre temporel et un ordre spatial entre les diverses fourmis : le système a un comportement "holiste". On peut également noter le caractère adaptatif du système, lorsqu'une nouvelle source

de nourriture est découverte, lors de l'égarment d'une fourmi... L'aléatoire ou plus exactement le chaos est, dans ce cas là, à l'origine de l'auto-organisation de la colonie. En effet, les fourmis qui s'écartent du chemin marqué par les phéromones mettent le système hors de son état d'équilibre. Le sous-ensemble du système ayant quitté l'état d'équilibre peut, à son tour, constituer une structure organisée en particulier lorsque les fourmis trouvent une nouvelle source de nourriture. Une forme organisée peut donc naître de mouvements chaotiques aléatoires pour ensuite se stabiliser.

L'auto-organisation repose aussi implicitement sur la coopération entre fourmis, cette coopération étant mise en oeuvre par le biais de la communication indirecte. Elle est régie par un mécanisme auto-catalytique élaboré à partir d'un processus de rétroaction. La rétroaction positive permet d'amplifier le fourragement vers la ressource découverte ; elle devient négative lorsque la source s'amoindrit.

2.5 Conclusion

Dans ce chapitre nous avons fait une étude sur les mécanismes d'adaptation dans les systèmes biologiques, qui se sont distingués aux niveaux : d'individu, d'espace et de système.

Nous avons présenté les principaux modèles biologiques trouvant des développements artificiels : les systèmes nerveux, systèmes évolutionnaires vivants et les modèles sociaux collectifs qui sont rattachés aux notions d'auto-organisation et d'intelligence collective.

L'objet du chapitre suivant est de présenter un certain nombre de travaux que l'on peut rassembler sous la notion, assez large, de modèles bio-inspirés.

Chapitre 3

Un survol des modèles bio-inspirés pour la résolution des problèmes complexes

3.1 Introduction

La biologie est la source d'inspiration de nombreux modèles pour la résolution des problèmes complexes. Ainsi, les théories de l'évolution ont inspiré les algorithmes évolutionnaires, la physiologie du neurone a inspiré les modèles connexionniste, les études éthologistes ont donné naissance au domaine de l'intelligence en essaim (swarm intelligence), l'étude de l'organisation des insectes sociaux a donné naissance les systèmes multi-agents situés.

Dans ce chapitre, nous présentons une analyse de ces modèles et en mettant en perspective les mécanismes d'adaptation.

3.2 Modèles connexionnistes

Les modèles connexionnistes sont fondés sur la métaphore du cerveau, c'est la tentative de donner à l'ordinateur les qualités de perception du cerveau humain qui a conduit à une modélisation électrique de celui-ci. C'est cette modélisation que tentent de réaliser les réseaux de neurones artificiels, dont l'élaboration repose sur base de la définition suivante, proposée par Haykin[109] : "*Un réseau de neurones est un processus distribué de manière massivement parallèle, qui a une propension naturelle à mémoriser des connaissances de façon expérimentale et de les rendre disponibles pour utilisation. Il ressemble au cerveau en deux points :*

1. la connaissance est acquise au travers d'un processus d'apprentissage ;
2. les poids des connexions entre les neurones sont utilisés pour mémoriser la connaissance. "

La première étude systématique du neurone artificiel est due au neuropsychiatre McCulloch et au logicien Pitts [148] qui, s'inspirant de leurs travaux sur les neurones biologiques, proposèrent, en 1943, le modèle illustré à la figure FIG. 3.1.

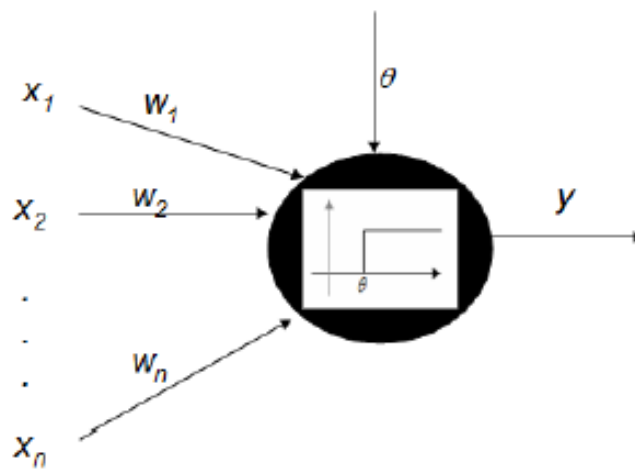


FIG. 3.1 – Le neurone formel de Mac Culloch et Pitts

Ce neurone formel est un dispositif qui s'active (réponse *oui* ou 1) ou non (réponse *non* ou 0) suivant que le potentiel d'activation est supérieur à un seuil θ ou non. Le potentiel d'activation est en fait ici, la somme pondérée des entrées (x_1, x_2, \dots, x_n) (signaux qui lui parviennent). Si cette somme pondérée dépasse un certain seuil θ alors le neurone fournit la valeur 1 ($y = 1$) sinon il fournit 0 ($y = 0$).

Ce neurone formel est donc, défini par le seuil θ et les poids des connexions d'entrées les w_i et par une fonction $f()$ qui est une fonction binaire définie comme suit :

$$y = f\left(\sum_{i=1}^n w_i x_i\right) \text{ et } y = \begin{cases} 0 & \text{si } \sum_{i=1}^n w_i x_i < \theta \\ 1 & \text{si } \sum_{i=1}^n w_i x_i \geq \theta \end{cases}$$

A partir de ce modèle, ont été définis divers modèles de neurones plus généraux pour de plus amples informations sur ces modèles, vous pouvez consulter [74].

Dans la figure FIG. 3.2, les possibilités les plus courantes sont les suivantes :

- Les entrées x_i et la sortie S sont de nature :

binaires (0, 1) ou (-1, +1)
réelles

- $E = h(x_1, x_2, \dots, x_n)$: fonction entrée totale, elle peut être :
 - booléenne ;
 - linéaire $\sum_{i=1}^n w_i x_i$:
 - affine $\sum_{i=1}^n w_i x_i - \theta$:
- $A = f(E)$: fonction d'activation, elle peut prendre plusieurs formes, comme elle présentée dans la figure FIG. 3.3.
- $g()$: fonction de sortie, en générale , elle est considérée comme la fonction d'identité.

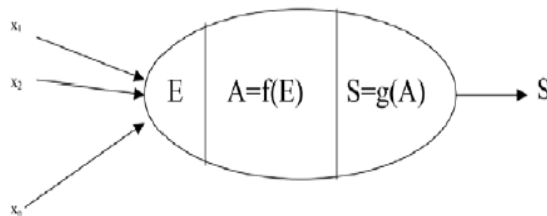


FIG. 3.2 – Forme générale d'un neurone formel

Heaviside		$f(x) = \begin{cases} 0 & \text{if } 0 > x \\ 1 & \text{if } x \geq 0 \end{cases}$
Sigmoïde		$f(x) = \frac{1}{1+e^{-\beta x}}$
Linéaire Seuillée		$f(x) = \begin{cases} 0 & \text{if } x \leq x_{min} \\ mx+b & \text{if } x_{max} > x > x_{min} \\ 1 & \text{if } x \geq x_{max} \end{cases}$
Gaussienne		$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
Identité		$f(x) = x$

FIG. 3.3 – Formes de fonctions d'activation

Un réseau de neurones est un assemblage de neurones connectés entre eux par des connexions pondérées (poids synaptique). Si le poids d'une connexion est positif le lien

est activateur, dans le cas contraire il est inhibiteur. Un neurone possède deux états : il est soit actif soit inactif. Chaque neurone met périodiquement à jour son état. Il effectue pour cela une sommation des signaux transmis par les neurones voisins. Si elle dépasse un certain seuil, le neurone devient actif et propage à son tour son potentiel d'action. Si la sommation est inférieure à ce seuil, le neurone reste inactif.

Il existe un grand nombre de modèles de réseau de neurones qui peuvent être distingués par [6] :

- La règle d'activation locale à chaque neurone qui est fonction des valeurs d'entrée reçues d'autres neurones ;
- La règle d'apprentissage qui permet de modifier les poids des connexions entre neurones ;
- La topologie décrivant la manière dont les neurones sont connectés entre eux.

Les réseaux de neurones possèdent une aptitude à l'apprentissage qui repose sur le renforcement des connexions entre les neurones concernés. On distingue trois types d'apprentissage [37] :

- L'apprentissage dit supervisé dans lequel on présente simultanément au réseau une entrée et la sortie désirée de façon à pouvoir calculer la différence entre la sortie produite et la sortie désirée et minimiser l'erreur commise ;
- L'apprentissage non supervisé qui consiste à détecter de manière automatique les similitudes dans les exemples traités et à modifier les poids de connexions entre neurones pour que des exemples ayant les mêmes caractéristiques fournissent les mêmes sorties ;
- L'apprentissage dit guidé ou renforcé dans lequel on indique au réseau si la sortie produite est correcte ou pas, ce qui entraîne une variation de poids de connexions.

Le fonctionnement d'un réseau neuronal est composé de deux phases distinctes :

- une phase d'apprentissage pendant laquelle les poids des connexions sont mis à jour de manière dynamique ;
- une phase d'exécution pendant laquelle le réseau est effectivement opérationnel.

A l'opposé des méthodes traditionnelles de résolution informatique, on ne construit pas avec les modèles connexionnistes, un programme pas à pas en fonction de la compréhension de celui-ci. Les paramètres les plus importants de ce modèle sont les coefficients synaptiques. Ce sont eux qui construisent le modèle de résolution en fonction des informations données au réseau [18].

Les modèles connexionnistes se caractérisent par leur capacité d'adaptation qui se manifeste tout d'abord dans les réseaux de neurones par la capacité d'apprentissage

qui permet de tenir compte de nouvelles contraintes ou de nouvelles données du monde extérieur. De plus, elle se caractérise dans certains réseaux par leur capacité d'auto-organisation qui assure leur stabilité en tant que système dynamique. Nous distinguons, dans ce contexte, les cartes auto-organisatrices de Kohonen et les Perceptrons multi-couches.

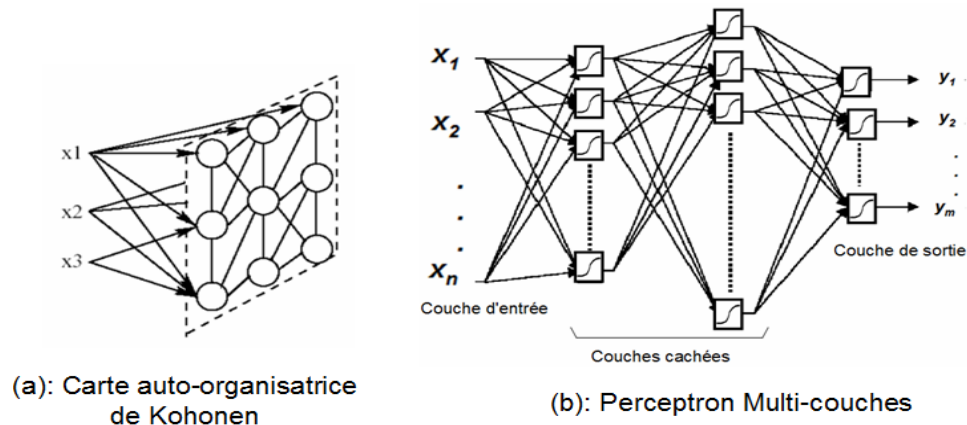


FIG. 3.4 – Modèles de réseaux de neurones

La carte auto-organisatrice de Kohonen est un modèle d'apprentissage non supervisé, l'algorithme de Kohonen [133] utilise la notion de neurone gagnant et surtout de voisinage. La topologie du réseau est en générale, formée de deux couches, une d'entrée classique, et une de sortie où tous les neurones sont connectés dans un plan FIG. 3.4(a).

Le principe de l'algorithme d'apprentissage de Kohonen est le suivant : après avoir initialisé les connexions aléatoirement, on sélectionne le neurone K tel que son vecteur de poids W_k soit le plus proche du vecteur d'entrée X et représente ainsi au mieux ce signal. L'augmentation de l'activation du neurone déterminé et du groupe de neurones qui l'entoure, lorsque le signal est présenté, nécessite la définition d'un voisinage V_K . La détermination V_K peut se faire de différentes façons ; les différentes simulations effectuées montrent qu'il vaut mieux choisir ce voisinage assez large au début de l'apprentissage pour le rétrécir par la suite. De même, la fonction d'oubli α , proportionnelle à l'inactivité des neurones, doit selon les simulations effectuées, être une fonction décroissante linéaire du temps d'apprentissage afin d'assurer l'aboutissement de celui-ci au bout d'un temps fini.

Algorithme 3.1 : Apprentissage d'une carte de Kohonen

1. Déterminer la taille de la carte de neurones. Initialiser les poids de connexion W entre les neurones et les vecteurs d'entrée. Définir le voisinage V_K du neurone ;
2. Présenter un vecteur d'entrée entrée X au réseau ;
3. Trouver le neurone le "plus proche" K (le gagnant) ;
 - Calculer la distance d_j entre cette entrée et le neurone j ;
 - Calculer $d_K = \min(d_j)$ pour $1 \leq j \leq M$, M est le nombre de neurones ;
4. Ajuster les poids du neurone gagnant K ; $\Delta W = \alpha(t)(X - W)$, ainsi que les poids des neurones du voisinage V_K du neurone gagnant, mais avec un taux d'apprentissage α réduit de manière à les spécialiser un peu moins que le neurone gagnant ;
5. Diminuer $\alpha(t)$ jusqu'à 0 et diminuer la taille du voisinage ;
6. Répéter 2 à 5 pour tous les vecteurs d'entrée et pour un certain nombre d'itérations.

Le réseau Perceptron multicouches est un modèle d'apprentissage supervisé de type rétro-propagation du gradient, constitué de neurones organisés en couches successives : une seule couche d'entrée, une seule couche de sortie et de une à plusieurs couches cachées. Chaque neurone est uniquement relié à tous les neurones de la couche suivante. Lorsque le vecteur de caractéristiques d'un objet est présenté à l'entrée (x_1, x_2, \dots, x_n) du réseau, il est communiqué à tous les neurones de la première couche (couche d'entrée). Les sorties des neurones de cette couche sont alors communiquées aux neurones de la couche suivante, et ainsi de suite. Les sorties de neurones (y_1, y_2, \dots, y_m) de la dernière couche sont les sorties du réseau (voir FIG. 3.4(b)). La fonction d'activation de chaque neurone est de type sigmoïde.

L'algorithme de rétro-propagation du gradient consiste, dans un premier temps, à propager vers l'avant les entrées jusqu'à obtenir une entrée calculée par le réseau. La seconde étape compare la sortie calculée à la sortie réelle connue. On modifie alors les poids de telle sorte qu'à la prochaine itération, l'erreur commise entre la sortie calculée et connue soit minimisée. On répète ce processus sur tous les exemples jusqu'à temps que l'on obtienne une erreur de sortie considérée comme négligeable.

Algorithme 3.2 : Apprentissage de Rétro-propagation du gradient

1. Initialisation des poids w_{ij} des neurones à des valeurs (petites) choisies au hasard ;

2. Présentation d'une entrée $X = (x_1, x_2, \dots, x_N)$ de la base d'apprentissage ;
3. Calculer la sortie du réseau en propageant des signaux en avant ;
4. Pour chaque sortie du neurone
 - Calculer $\delta_j = (y_j - d_j)$; d_j est la sortie désirée du neurone j et y_j est sa sortie courante : $y_j = g\left(\sum_i w_{ij}x_i\right) = \left(1 + e^{-\sum_i w_{ij}x_i}\right)^{-1}$; tel que g est une fonction d'activation sigmoïde ;
5. Pour tous les autres neurones (de la dernière couche cachée à la première)
 - Calculer $\delta_j = \sum_k w_{jk}g'(x)\delta_k$; où δ_k est le δ_j de la couche suivante et $g'(x) = y_k(1 - y_k)$;
6. Mettre à jour les poids selon : $w_{ij}(t+1) = w_{ij}(t) - \eta y_i y_j (1 - y_j) \delta_j$, où η est un paramètre appelé le taux d'apprentissage ;

Après toutes les capacités séduisantes exposées précédemment, on peut citer les principales limites à l'utilisation des réseaux de neurones :

- la structure (nombre de couches, connexions) du réseau de neurones nécessite d'avoir une vue complète de la tâche à exécuter ;
- l'incapacité de réseaux de neurones à exploiter les résultats qu'ils fournissent ; les réseaux se présentent comme des boîtes noires dont les règles de fonctionnement sont inconnues. La qualité de leurs performances ne peut être mesurée que par des méthodes statistiques.

3.3 Algorithmes Evolutionnaires

Les algorithmes évolutionnaires "AE" sont des techniques de recherche inspirées par l'évolution biologique des espèces, apparues à la fin des années 1950 [88]. Il s'agit de générer des populations de solutions potentielles à des problèmes donnés, de sélectionner les meilleures solutions et de les répliquer en faisant quelques modifications pour chercher une meilleure solution aux problèmes. Ces algorithmes sont fondés sur une analogie avec le monde biologique. En effet, pour un problème donné, une solution est un individu et un ensemble de solutions correspond à une population d'individus. Chaque individu peut être appelée chromosome, et chaque chromosome est constitué d'un ensemble de caractéristiques appelées les gènes. L'ensemble des gènes d'un individu est son génotype et l'ensemble du patrimoine génétique d'une espèce est le génome. Les différentes versions d'un même gène sont appelées allèles.

On peut distinguer trois classes d'algorithmes évolutionnaires ont été développés indépendamment, entre la moitié des années 1960 et 1970 :

- les algorithmes génétiques pour l'optimisation combinatoire [112]-[113]-[98]-[99] ;
- la programmation Evolutionnaire [83]-[84] qui a été conçue comme une méthode d'intelligence artificielle pour la conception d'automates à états finis ;
- les stratégies évolutionnistes [184]-[185] pour résoudre des problèmes d'optimisations continus.

Dans les années 90, ces trois champs ont beaucoup évolués et se sont rapprochés, pour finir par être regroupés sous le terme *Evolutionary Computation* [85]. D'autres variantes à ces algorithmes ont été développées, telles que la programmation génétique [134] qui est une extension de l'algorithmique génétique et a pour but de chercher des solutions dans l'espace des programmes.

3.3.1 Principe d'un algorithme évolutionnaire

Le principe d'un algorithme évolutionnaire se décrit simplement [73] : un ensemble de N individus (points) dans un espace de recherche, choisis à priori au hasard, constituent la population initiale ; chaque individu x de la population possède une certaine performance, qui mesure son degré d'adaptation à l'objectif visé ; dans le cas de la minimisation d'une fonction objectif f , x est d'autant plus performant que $f(x)$ est plus petit. Un *AE* consiste à faire évoluer progressivement, par générations successives, la composition de la population, en maintenant sa taille constante. Au cours des générations, l'objectif est d'améliorer globalement la performance des individus ; on s'efforce d'obtenir un tel résultat en mimant les deux principaux mécanismes qui régissent l'évolution des êtres vivants, selon la théorie de C. Darwin :

- la sélection, qui favorise la reproduction et la survie des individus les plus performants,
- et la reproduction, qui permet le brassage, la recombinaison et les variations des caractères héréditaires des parents, pour former des descendants aux potentialités nouvelles.

Le passage d'une génération à la suivante se déroule en quatre phases : une phase de sélection, une phase de reproduction (ou de variation), une phase d'évaluation des performances et une phase de remplacement. La phase de sélection désigne les individus qui participent à la reproduction. Ils sont choisis, éventuellement à plusieurs reprises, a priori d'autant plus souvent qu'ils sont performants. Les individus sélectionnés sont

ensuite disponibles pour la phase de reproduction. Celle-ci consiste à appliquer des opérateurs de variation sur des copies des individus sélectionnés pour en engendrer de nouveaux; les opérateurs les plus utilisés sont le croisement (ou recombinaison), qui produit un ou deux descendants à partir de deux parents, et la mutation, qui produit un nouvel individu à partir d'un seul individu. La structure des opérateurs de variation dépend étroitement de la représentation choisie pour les individus. Les performances des nouveaux individus sont ensuite mesurées, durant la phase d'évaluation, à partir des objectifs fixés. Enfin, la phase de remplacement consiste à choisir les membres de la nouvelle génération : on peut, par exemple, remplacer les individus les moins performants de la population par les meilleurs individus produits, en nombre égal. L'algorithme est interrompu après un certain nombre de générations, selon un critère d'arrêt à préciser. L'algorithme suivant illustre le principe de fonctionnement d'un algorithme évolutionnaire.

Algorithme 3.3 : Principe de fonctionnement d'un algorithme évolutionnaire

```
t := 0;
Initialiser_population P(t);
Evaluer P(t);
Tant que non critère d'arrêt Faire
    t := t + 1;
    P'(t) :=sélectionner-parents (P(t));
    Croiser (P'(t));
    Muter (P'(t));
    Evaluer (P'(t));
    P(t + 1) := sélectionner-survivants (P(t), P'(t));
Fin tant que
```

L'ensemble des algorithmes évolutionnaires utilise un algorithme similaire à celui présenté ci-dessus. Les principales différences entre eux proviennent de la nature du codage des solutions ainsi que des opérateurs génétiques utilisés. Le tableau TAB. 3.1 présente les caractéristiques de chacun des principaux algorithmes évolutionnaires, dans leur version originale [139].

Vu que les Algorithmes Génétiques "AG" sont les algorithmes les plus populaires des algorithmes évolutionnaires, nous allons détailler un peu le fonctionnement de ce type d'algorithmes.

	Algorithme génétique	Stratégie évolutionnaire	Programmation évolutionnaire	Programmation génétique
individu	chaîne de bit	tableau	quelconque	arbre
sélection	probabiliste	élitiste	élitiste	probabiliste
croisement	oui	non	non	oui
mutation	oui	oui	oui	non

TAB. 3.1 – Caractéristiques des principaux algorithmes évolutionnaires

3.3.2 Algorithmes génétiques

L'application d'un algorithme génétique, à tout problème pratique, met en jeu six composantes[166] :

1. Une représentation sous la forme d'un chromosome des solutions ;
2. Une fonction objectif pour évaluer la qualité, en termes de fitness, de chaque individu ;
3. Une méthode d'initialisation de la population des solutions candidates ;
4. Les valeurs des paramètres de l'algorithme génétique utilisé (par exemple la taille de la population) ;
5. Les opérateurs génétiques qui produisent l'ensemble des nouveaux individus ;
6. Le critère d'arrêt de l'algorithme génétique.

Parmi ces composantes, certaines dépendent entièrement de l'application (par exemple, évaluation et représentation), alors que les autres sont définies indépendamment du domaine d'application.

Représentation des individus

Dans un algorithme génétique, la représentation des individus est basée sur le codage binaire de l'information. Le problème est qu'il existe plusieurs façons de coder l'information, et le choix optimal n'est pas évident à trouver. Il doit en effet permettre une recherche efficace par les opérateurs qui seront appliqués aux chromosomes. Afin de résoudre ce problème, une adaptation de la représentation et des opérateurs à l'espace de recherche, associée au problème traité, est essentielle.

Génération

Une fois le codage des individus est déterminé, l'algorithme génétique démarre par la génération d'une population initiale. Cette génération est fortement dépendante du codage des individus mais ne présente pas de difficultés particulières. Notons toutefois

que de nombreuses approches utilisent des individus générés selon des méthodes d'optimisations connues, ce qui permet dans certains cas d'accélérer la vitesse de convergence de l'algorithme ou d'éviter certains minima/maxima locaux.

Evaluation, Fitness

L'évaluation doit fournir une valeur permettant de déterminer une relation d'ordre entre les différents individus, relation d'ordre liée à la qualité de l'individu en tant que solution du problème. Cette évaluation est réalisée par une fonction, la fonction de fitness, qui doit produire objectivement une estimation de la qualité du résultat. Il est généralement nécessaire de fournir des valeurs de fitness strictement positives pour permettre aux opérateurs de sélection de fonctionner correctement.

Sélection

Le principe de cet opérateur consiste à favoriser la propagation des meilleures solutions parmi la population, tout en préservant la diversité génétique, afin d'explorer de nouvelles régions de l'espace de recherche. Plusieurs opérateurs de sélection ont été proposés dans la littérature [71]-[18], comme, par exemple, celui proposé par Goldberg [98] qui sélectionne les individus proportionnellement à leur performance.

Croisement

Cet opérateur, appelé "crossover", est mis en œuvre dans la phase de recombinaison. Le but de cette opération est de maintenir la diversité en manipulant les composants des individus (chromosomes). Une fois la sélection effectuée, de nouveaux individus, appelés enfants, sont créés par croisement, c'est-à-dire en échangeant des parties des individus de la population (parents).

Cet échange se fait en sélectionnant un point de découpage, aléatoirement (avec une densité de probabilité uniforme), et en échangeant ensuite les deux sous-chaînes de chacun des deux parents. Considérons, par exemple, deux chaînes de parents A et B :

$$A = 110 : 10$$

$$B = 011 : 01$$

et supposons que le point de découpage est celui indiqué par " : ". Le résultat du croisement permet d'avoir deux nouveaux individus faisant partie de la nouvelle génération :

$$A' = 110 : 01$$

$$B' = 011 : 10$$

Cette procédure peut s'étendre à un découpage comportant plus de deux parties. Le croisement à découpage est très rapide à mettre en œuvre dans le cas où le problème utilise des entiers.

Mutation

Le principe de la mutation consiste à tirer aléatoirement un gène (par exemple, un bit, dans le cas du codage binaire) dans le chromosome et à le remplacer par une valeur aléatoire. La mutation est l'opérateur qui apporte aux algorithmes génétiques l'aléa nécessaire à une exploration efficace de l'espace de recherche. Elle garantit la possibilité d'atteindre tout l'espace d'état. Ainsi l'algorithme génétique peut converger sans croisement. L'utilisation de la mutation, en tant qu'opérateur de recherche locale, suggère de le combiner avec d'autres techniques de recherche locale. Dans ces cas, les algorithmes génétiques sont dits hybrides.

Elitisme

Une stratégie élitiste consiste à conserver au moins un individu de la population de la génération précédente, dans la génération suivante.

3.4 Intelligence en essaim

Les études éthologistes ont pu identifier chez certains animaux et insectes des comportements auto-organisés menant à l'émergence de tâches qualifiées de complexe sans contrôle global [22]. En effet, ces insectes sociaux sont capables de créer des communautés géantes et réalisent collectivement des tâches extraordinaires telles que le fourragement chez les fourmis, la construction de véritables cathédrales de terre chez les termites, les déplacements collectifs chez les oiseaux et les bancs de poissons. Ce comportement global complexe émerge à partir des comportements locaux et simples de chaque élément du groupe.

L'éthologie dans ce domaine a inspiré beaucoup de travaux, et a donné naissance au domaine de l'intelligence en essaim (swarm intelligence) [23].

L'intelligence en essaim est une forme particulière d'intelligence collective, qui est définie comme : "*N'importe quelle tentative de concevoir des algorithmes ou des dispositifs de résolution de problèmes distribués inspirés du comportement collectif de colonies d'insectes sociales et d'autres sociétés animales*" [23].

L'intelligence en essaim fournit une base avec laquelle il est possible d'explorer la résolution du problème, d'une manière distribuée, sans contrôle centralisé, et ni présentation globale de la tâche à accomplir.

Cette approche couvre des travaux de différentes natures comme le développement de métaheuristiques pour les problèmes d'optimisation, et pour le problème de classification, la robotique collective. . . Dans cette section, nous exposons quelques travaux, en les présentant selon leurs source d'inspiration.

3.4.1 Méthodes inspirées de l'organisation du travail chez les fourmis

Les modèles d'organisation du travail chez les fourmis ont inspiré des algorithmes pour l'allocation dynamique de tâches, où chaque machine se voit associée à un individu disposant d'un jeu de seuils de réponse θ_a , où représente $\theta_{a,j}$ le seuil de l'agent a pour la tâche j . La tâche j envoie aux agents un stimulus S_j représentant le temps d'attente de la tâche. L'agent a aura une probabilité d'effectuer la tâche j de :

$$P(\theta_{a,j}, S_j) = \frac{S_j^2}{S_j^2 + \theta_{a,j}^2} \quad (3.1)$$

L'algorithme dispose ensuite de règles de mise à jour des seuils et de règles de décision, au cas où deux agents tenteraient d'effectuer la même tâche (voir [45] pour plus de détails). Des améliorations à cet algorithme de base [169] ont permis d'augmenter sa rapidité et son efficacité sur des problèmes d'allocation dynamique de tâches.

3.4.2 Méthodes inspirées de tri collectif du couvain chez les fourmis

Dans la nature, les fourmis offrent un modèle stimulant pour le problème de classification. Le tri collectif du couvain ou de la constitution de cimetières observés chez certaines espèces de fourmis sont les plus marquants.

Une des premières études relatives à ce domaine a été menée par Deneubourg et al. [68] où une population d'agents fourmis se déplacent aléatoirement sur une grille à deux dimension et sont capables de déplacer des objets dans le but de les rassembler. Pour rassembler en tas un ensemble d'éléments (d'objets) de même type, les probabilités de ramasser un objet p_p et de le déposer p_d ont été explicitées :

Lorsqu'une fourmi ne transporte aucun objet, sa probabilité d'en ramasser un, observé sur son voisinage, est donné par :

$$P_p = \left[\frac{k_1}{k_1 + f} \right]^2 \quad (3.2)$$

Où k_1 est une constante, f représente la fréquence de rencontre d'objets dans un passé récent, Autrement dit, plus une fourmi rencontre d'objets, moins elle a de chance

d'en prendre un (elle se trouve dans une zone avec beaucoup d'objets). Ensuite, une fois un objet ramassé, la fourmi se déplace au hasard dans le plan, et elle dépose l'objet avec une probabilité :

$$P_d = \left[\frac{f}{k_2 + f} \right]^2 \quad (3.3)$$

Où k_2 est une constante positive, f correspond à la fréquence de rencontre d'objets dans un passé récent. Cette probabilité est d'autant plus grande que la fourmi a rencontré récemment des 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'objets) ; cet algorithme permet alors de trier des objets.

Ces principes ont trouvé leurs premières applications en robotique collective et de nombreux travaux en découlent, pour plus d'informations sur ces travaux, voir [13]-[159]-[23].

L'algorithme de tri d'objets [68] a été adapté par Lumer et Faieta [150] pour résoudre le problème de classification de données ; les données sont initialement réparties aléatoirement sur une grille $2D$. Chaque fourmi est située dans une case de cette grille et ne perçoit que les données situées dans son voisinage (8 voisins par exemple). Ensuite la fréquence f utilisée dans l'algorithme de tri vu précédemment peut être remplacée par une moyenne des similarités entre une donnée d_i portée par une fourmi et les données d_j situées dans son voisinage. Une donnée d_i sur la grille est ramassée avec une probabilité d'autant plus grande qu'elle est peu similaire aux données voisines. De la même manière, une donnée d_i portée par une fourmi est plus facilement déposée dans une région comportant des données qui lui sont similaires.

Une extension de l'algorithme de [150] a été présentée dans [158]. D'une part, les fourmis peuvent empiler les objets les uns sur les autres dans une même case de la grille. Lorsqu'elles rencontrent un tas d'objets, elles peuvent ainsi se saisir de l'objet le plus dissimilaire. D'autre part, une hybridation a été effectuée avec l'algorithme des K-Means (centres mobiles) [86] et a donné naissance à l'algorithme AntClass qui consiste en une succession d'itérations des fourmis et de centres mobiles. Les fourmis ont pour tâche de réduire le nombre de classes et les centres mobiles d'améliorer globalement la partition découverte par les fourmis.

3.4.3 Méthodes d'optimisation inspirées des stratégies de fourrage-ment chez les fourmis

Les stratégies de recherche de nourriture (fourrage-ment) sont en effet extrêmement diversifiées à cause de plusieurs facteurs, comme nous avons vu dans le chapitre précédent. Elles peuvent être effectuées de façon collective, chez certaine espèce de fourmis [12]-[101] ou bien de façon solitaire, comme chez l'espèce de fourmis *Pachycondylas apicalis* [89]

Ces stratégies de fourrage-ment sont à l'origine d'un certain nombre d'algorithmes d'optimisation : Optimisation par colonie de fourmis (Ant Colony Optimisation "ACO") inspirés de la stratégie du fourrage-ment collectif, qui sont basés sur l'utilisation de marquages chimiques (phéromones), et API "pour APICALIS" inspiré de la stratégie du fourrage-ment solitaire, qui se base sur la capacité individuel de la fourmi "son apprentissage".

Optimisation par colonie de fourmis

Les fourmis utilisent les pistes de phéromones pour marquer leur trajet, par exemple entre le nid et une source de nourriture. Une colonie est ainsi capable de choisir (sous certaines conditions) le plus court chemin vers une source à exploiter [12]-[101], sans que les individus aient une vision globale du trajet.

L'utilisation de ce type de marquages chimiques (phéromones) par les fourmis, pour la recherche de nourriture, est à l'origine d'un certain nombre d'algorithmes d'optimisation combinatoire. Le Problème du Voyageur de Commerce "PVC" est l'un des cas le plus abordé, Le premier algorithme de colonies de fourmis conçu, dans ce contexte, est Ant System "AS" [48]-[75].

La modélisation des fourmis artificielles pour la résolution du problème du voyageur de commerce est la suivante : les fourmis sont placées sur les sommets du graphe (c'est-à-dire, sur chaque ville). Elles se déplacent d'un sommet à l'autre en empruntant les arêtes du graphe. On note par $b_i(t)$ le nombre de fourmis dans la ville i à l'instant t et soit $m = \sum_{i=1}^n b_i(t)$ le nombre total de fourmis. Chaque agent fourmis possède les caractéristiques suivantes :

- La fourmi dépose une trace de phéromones sur l'arête (i, j) quand elle se déplace de la ville i à la ville j ;
- Elle choisit la ville de destination suivant une probabilité qui dépend de la distance entre cette ville et sa position et de la quantité de phéromones présente sur l'arête (règle de transition) ;

- Afin de ne passer qu'une seule fois par chaque ville, la fourmi ne peut rendre sur une ville qu'elle a déjà traversée, c'est pour cela que la fourmi doit être dotée d'une mémoire.

Pour éviter qu'une fourmi ne revienne sur ses pas, elle conserve la liste des villes qu'elle a déjà traversées. Cette liste, nommée liste-tabou, constitue la mémoire de la fourmi.

Les traces de phéromones sont modélisées par les variables $\tau_{i,j}(t)$ qui donnent l'intensité de la trace sur le chemin (i, j) à l'instant t la probabilité de transition du sommet i vers le sommet j par la fourmi k est donnée par :

$$p^k_{i,j}(t) = \begin{cases} \frac{(\tau_{i,j}(t))^\alpha (u_{i,j}(t))^\beta}{\sum_{l \in L_k(i)} ((\tau_{i,l}(t))^\alpha (u_{i,l}(t))^\beta)} & \text{si } j \notin L_k(i) \\ 0 & \text{sinon} \end{cases} \quad (3.4)$$

Où $L_k(i)$ représente la liste-Tabou de la fourmi k située sur le sommet i et $\tau_{i,j}(t)$ représente une mesure de visibilité qui correspond à l'inverse de la distance entre les villes i et j , α et β sont deux paramètres permettant de moduler l'importance relative des phéromones et de la visibilité.

La mise à jour de phéromones est effectuée une fois que toutes les fourmis sont passées par toutes les villes :

$$\tau_{i,j} = \tau_{i,j} (1 - \rho) + \sum_{k=1}^m \Delta\tau_{i,j}^k \quad (3.5)$$

où ρ est un coefficient représentant l'évaporation des traces de phéromones. $\Delta\tau_{i,j}^k$ représente le renforcement de l'arc (i, j) pour la fourmi k :

$$\Delta\tau_{i,j}^k = \begin{cases} Q/L_k & \text{si la fourmi } k \text{ est passée par l'arc } (i, j) \\ 0 & \text{sinon} \end{cases} \quad (3.6)$$

Avec Q une constante et L^k la longueur du chemin parcouru par la fourmi k
L'algorithme suivant donne la structure générale de *AS* pour *PVC* (noté *AS-TSP*)

Algorithme 3.4 : Algorithme AS-TSP

Initialisation

$$\tau_{i,j} \leftarrow \tau_0 \quad \forall (i,j) \in \{1, \dots, n\}^2$$

Placer aléatoire chaque fourmi sur une ville

Pour $t = 1$ à $t = t_{max}$ **Faire**

Pour chaque fourmi k **Faire**

 Construire un chemin $T^k(t)$ avec la règle de transition (3.4)

 Calculer la longueur $L^k(t)$ de ce chemin

FinPour

 Soient T^* le meilleur chemin trouvé et L^* la longueur correspondante

 Mettre à jour les traces de phéromones suivant la règle (3.5)

FinPour

Retourner T^* et L^*

Concernant le nombre de fourmis, il est raisonnablement proposé d'utiliser autant de fourmis que de villes ($m = n$).

Plusieurs extensions ont été proposées autour de *AS* [159] : *ANT-Q* [95], MAX-MIN Ant System [205]-[207], *AS_{rank}* [34], Ant Colony System [76]. Toutes ces variantes ont été regroupées sous une description plus large : L'heuristique *ACO* (Ant Colony Optimisation), afin de faciliter le rapprochement des méthodes entre elles et se soustraire aux spécificités du *PVC* [77]-[206].

L'applicabilité de l'*ACO* ne se résume pas uniquement au *PVC*. Bien que la métaheuristique soit relativement récente, son applicabilité a été élargie pour résoudre divers problèmes combinatoires : coloriage des graphes, assignment quadratique, ordonnancement des tâches, routage sur réseaux, routage de véhicule, satisfaction des contraintes, . . . Les références correspondant à ces travaux ont été regroupés dans un tableau dans la thèse de Monmarché [159].

Devant le succès rencontré par ces algorithmes dans la résolution des problèmes combinatoires, d'autres pistes commencent à être explorées : par exemple, l'utilisation de ces algorithmes dans des problèmes continus [73], problèmes dynamiques [70], ou encore l'exploitation de ce type d'algorithmes dans un cadre d'intelligence en essaim [23] et avec d'autres métaheuristiques [160]-[71]. Plus d'informations sur les algorithmes de colonies de fourmis et leurs aspects biologiques se trouvent dans [23]-[78]. L'applicabilité de l'*ACO* dans les problèmes de segmentation d'image est un peu faible ; il est concrétisé par les travaux de Han et al. [105] pour la classification d'image, et Liang et al. [144] pour le seuillage d'image.

L'optimisation par API

L'algorithme *API* (pour *APIcalis*) est un algorithme d'optimisation développé à la fin des années 90 s'inspirant de la stratégie de fourragement observée chez les fourmis de la famille *Pachycondyla Apicalis* [159]. L'intérêt de ces fourmis pour l'optimisation vient du fait qu'elles utilisent des principes relativement simple à la fois d'un point de vue global et local pour rechercher leurs proies. A partir de leurs nids, elles couvrent globalement une surface donnée en la partitionnant uniformément en site de chasses individuelles. Pour une fourmi donnée, on observe une stratégie d'exploration aléatoire des sites sensibles au succès rencontré. Ces principes peuvent être repris dans un problème analogue, par exemple : la recherche d'un minimum global d'une fonction f de R^n dans R .

L'algorithme *API*, est générique en ce qui concerne l'espace de recherche S , c'est là un de ses principaux atouts. La définition des deux opérateurs suivants est suffisante pour déterminer le déplacement des fourmis :

1. l'opérateur O_{rand} qui génère un point s de S de manière uniformément aléatoire ;
2. l'opérateur O_{explo} qui génère un point s' dans le voisinage d'un point s .

Concernant le deuxième opérateur, la taille du voisinage de s est paramétrée par une amplitude, notée A telle que $A \in [0, 1]$. Cette amplitude fixe la portée de l'exploitation autour de s relativement à la taille de l'espace. O_{explo} peut être une exploration aléatoire tout comme une heuristique inspiré par le domaine de recherche.

Au départ, et à chaque déplacement de nid, chaque fourmi a_i quitte le nid pour se constituer une liste de p sites de chasse qu'elle mémorise. Un site de chasse est un point de S construit par l'opérateur O_{explo} avec une amplitude $A_{site}(a_i)$ dans le voisinage de N . La fourmi a_i va ensuite procéder à une exploration locale autour d'un de ses sites de chasse (FIG. 3.5). Initialement, quand l'intérêt des sites est inconnu, la fourmi choisit un site s au hasard parmi les p sites dont elle dispose. L'exploration local consiste à construire un point s' de S dans le voisinage s grâce à l'opérateur O_{explo} avec une amplitude $A_{locale}(a_i)$. La fourmi a_i capture une proie si cette exploration locale a permis de trouver une meilleur valeur de f , ce qui revient à avoir $f(s') < f(s)$. Une amélioration de f modélise la capture d'une proie. A chaque fois qu'une fourmi parvient à améliorer $f(s)$, elle mémorise s' à la place de s et sa prochaine exploration locale aura lieu dans le voisinage de s' . Si l'exploration locale est infructueuse, pour la prochaine exploration, la fourmi choisira un site au hasard parmi les p sites qu'elle a en mémoire. Quand un site a été exploré successivement plus de P_{locale} fois sans avoir rapporté de proie, il est définitivement oublié et sera remplacé par un nouveau site à la prochaine itération (c'est-à-dire la prochaine sortie du nid). Le paramètre P_{locale} représente une patience locale.

Enfin, à chaque déplacement du nid, la mémoire des fourmis est vidée et elles doivent reconstruire leur p sites de chasse. Du point de vue de l'optimisation, cela permet d'éviter des minima locaux dans lequel les fourmis resteraient enfermées. Cela permet aussi de rassembler les fourmis autour du meilleur point trouvé et ainsi de concentrer les recherches. L'automate de la figure FIG. 3.6 résume le comportement individuel d'une fourrageuse.

Quelques extensions à cet algorithme ont été proposées dans [161]. Par exemple, une fourmi pourrait recruter un autre individu de la colonie si celui-ci explore une zone très fructueuse, augmentant ainsi les chances d'obtenir rapidement l'optimum recherché. Ou encore les paramètres pilotant le comportement des fourmis pourraient être fixés indépendamment pour chaque individu.

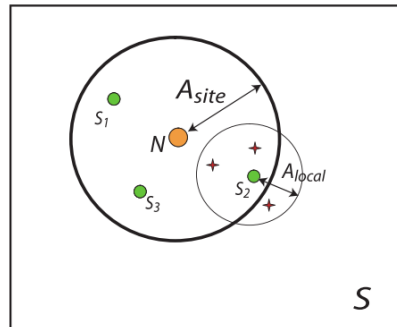


FIG. 3.5 – Exploration locale de la fourmi autour de son nid de rattachement. Trois sites de chasses s_i sont identifiés autour du nid N . La fourmi effectue une exploration locale depuis le site s_2 .

Une adaptation de l'algorithme *API*, au contexte de la recherche d'information sur Internet, est l'algorithme *Antsearch*, a été faite par Picarougne et al. [179].

3.4.4 Méthodes inspirées de comportements de déplacements

Dans la biologie, de nombreux chercheurs se sont intéressés à la manière dont les animaux se déplacent en groupe : les nuées d'oiseaux en vol, les bancs de poissons, les troupes mobiles. Aucun individu ne contrôle les autres mais pourtant des formes et des comportements complexes peuvent apparaître lors de ces déplacements.

Reynolds [186] et Heppner [111] ont proposé une utilisation informatique de tels modèles, simulations qui sont utilisées notamment dans l'industrie du cinéma pour donner des mouvements réalistes à des groupes d'individus "Boids". Dans ces travaux, chaque individu "Boid" évolue dans un espace 3D. Il est donc caractérisé par sa position et sa vitesse. Un individu perçoit les autres dans un voisinage donné. Des règles

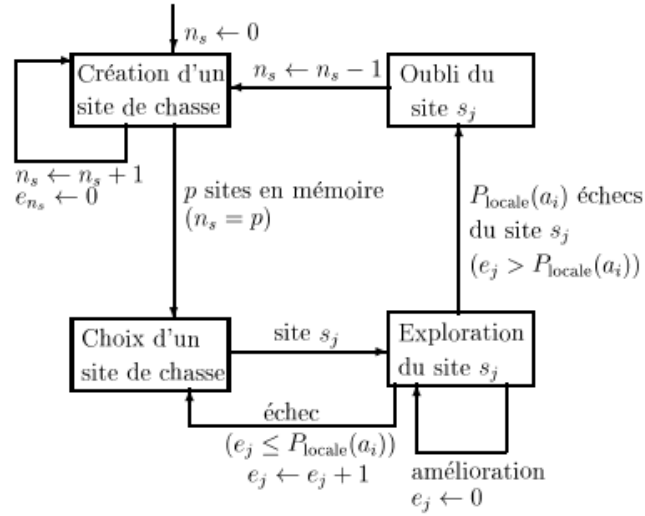


FIG. 3.6 – Automate représentant le comportement individuel de fourrage d’une fourmi. n_s représente le nombre de sites que la fourmi mémorise à un instant donné. e_j donne le nombre d’échecs successifs du site s_j mémorisé.

comportementales généralement simples comme : rester relativement proche des autres individus, aller dans la même direction, à la même vitesse, suffisent à maintenir la cohésion du groupe tout entier, et à permettre aux individus de se déplacer en groupe, d’éviter des obstacles, etc.

De ces comportements, on s’est inspiré des techniques pour la classification de données [181]-[162] et pour l’optimisation [126].

Dans ce qui suit, nous présentons, de façon générale, la méthode de classification de données par nuage d’insectes volants et nous détaillons la méthode d’optimisation par essaim de particules, sur laquelle nos travaux se sont basés.

Classification de données par nuages d’agents volants

Les principes de déplacements collectifs des agents [186] ont été appliqués pour la première fois à un problème de classification par Proctor et Winter [181] ; des nuages de poissons "agents" nageant en 3D peuvent servir à former des groupes de données. Initialement, les agents sont placés avec des coordonnées et des vecteurs vitesse aléatoires (voir figure FIG. 3.8 (a)). Les mouvements d’un agent dépendent des autres agents perçus dans son voisinage et des similarités entre les données qu’ils représentent. Le comportement local de chaque agent tend à former globalement des groupes d’agents similaires se déplaçant de manière cohérente (voir figure FIG. 3.8 (b)).

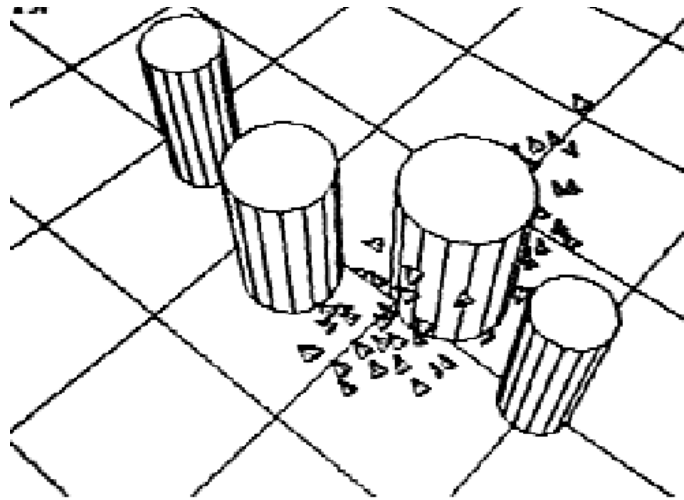


FIG. 3.7 – Les Boids de Reynolds, adapté de [186]

Ce travail a été amélioré et évalué d'une manière plus systématique par Monmarché et al [162]. Une distance idéale entre individus est définie, distance qui dépend de la similarité entre les données. Un critère d'arrêt est utilisé également en mesurant l'entropie spatiale du nuage d'agents. Ce travail a été intégré dans un système de fouille visuelle de données utilisant la réalité virtuelle.

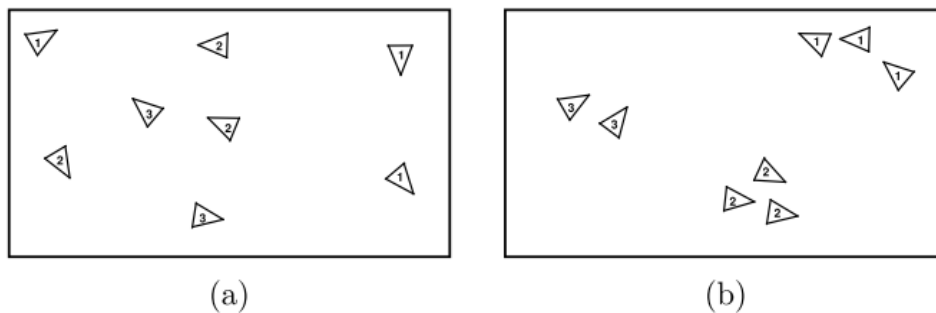


FIG. 3.8 – Principes utilisés pour la classification par nuages d'agents.

3.5 L'optimisation par essaim de particules

L'optimisation par essaim de particules (Particle Swarm Optimization, "*PSO*") est une technique d'optimisation parallèle développée par Kennedy et Eberhart [126],

en s'inspirant du comportement social des individus qui ont tendance à imiter les comportements réussis qu'ils observent dans leur entourage, tout en y apportant leurs variations personnelles.

L'origine de cette méthode vient des observations faites lors des simulations informatiques de vols groupés d'oiseaux et de bancs de poissons de Reynolds [186] et Heppner [111].

Les auteurs de la méthode d'optimisation par essaim de particules se sont inspirés des comportements collectifs d'un groupe d'individus en mouvement, en mettant en perspective la théorie de la socio-psychologie sur le traitement de l'information et les prises de décisions dans des groupes sociaux [129].

Cette méthode a connu beaucoup de succès auprès de la communauté d'optimisation. Ses bonnes performances dans différentes applications, et sa possibilité de résoudre une variété de problèmes d'optimisation tels que les problèmes d'optimisation discrète, continue et non linéaire [126]-[129]-[198]-[62], et aussi la possibilité d'hybridation avec d'autres métaheuristiques [166] ont contribué à cet engouement. Malgré son "jeune âge" en comparaison avec les autres métaheuristiques, un nombre important de travaux ont été publiés. Les travaux de Banks et al. [10]-[11] présentent un état de l'art complet sur cette méthode : ses variantes, ses différentes hybridations avec d'autres métaheuristiques,...

3.5.1 Principe de la méthode

La méthode en elle-même met en jeu des groupes de particules sous forme de vecteurs se déplaçant dans l'espace de recherche, en étant influencés par leur comportement passé et par celui de leurs voisins, ceux-ci définissent la notion socio-psychologie. Chaque particule est caractérisée par sa position et un vecteur de changement de position (appelé vitesse ou vecteur vitesse). La mise à jour de la position de chaque particule dépend de : la direction de son mouvement, sa vitesse, sa meilleure position par laquelle elle est déjà passée (sa meilleure performance) et la meilleure position connue au sein de son voisinage (la meilleure performance de ses informatrices) (voir figure FIG. 3.9).

3.5.2 Voisinage d'une particule

Le voisinage d'une particule constitue la structure du réseau social, à l'intérieur duquel les particules communiquent entre-elles. Différents topologies de voisinages ont été proposées [128]-[202] :

- Topologie en étoile (FIG. 3.10(a)) : le réseau social est complet, chaque particule est attirée vers la meilleure particule notée *gbest* et communique avec les autres.

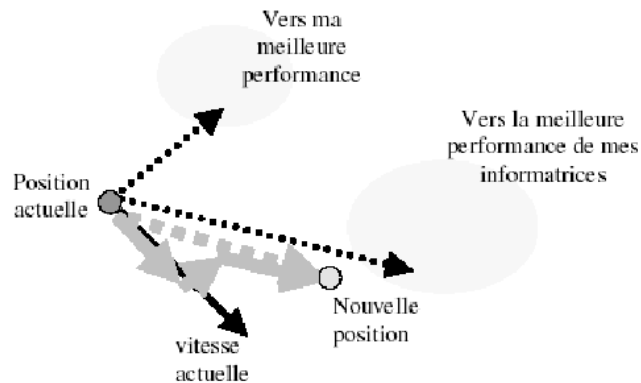


FIG. 3.9 – Déplacement d'une particule

- Topologie en anneau (FIG. 3.10(b)) : chaque particule communique avec n voisines immédiates ($n = 3$). Chaque particule tend à se déplacer vers la meilleure particule dans son voisinage local notée *lbest*.
- Topologie en rayon (FIG. 3.10(c)) : une particule "centrale" est connectée à tous les autres. Seule cette particule centrale ajuste sa position vers la meilleure, si cela provoque une amélioration l'information est propagée aux autres.
- Topologie Von Neumann (FIG. 3.10(d)), chaque particule est connectée à ses quatre particules voisines. Chaque particule tend à se déplacer vers la meilleure dans son voisinage local notée *lbest*.

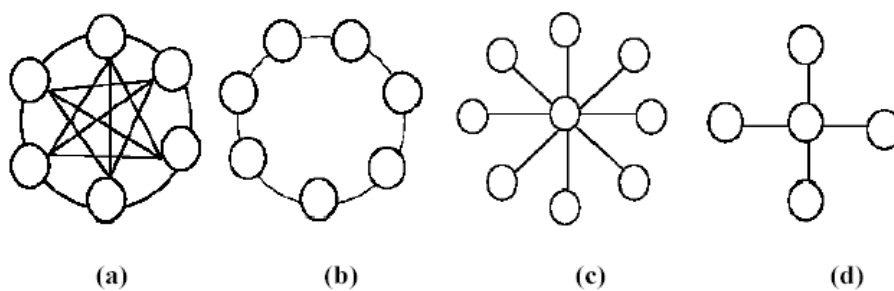


FIG. 3.10 – Différentes topologies de voisinage

3.5.3 Algorithme *PSO*

L'algorithme *PSO* est initialisé par une population de N solutions potentielles aléatoires, interprétées comme des particules se déplaçant dans l'espace de recherche.

Chaque particule i est modélisée par sa position \vec{x}_i et un vecteur de changement de position \vec{v}_i (appelé vitesse ou bien vitesse). La position d'une particule est influencée par sa meilleure position visitée par elle-même \vec{y}_i (la meilleure position personnelle désignant sa propre expérience) et la meilleure position parmi ses particules voisines $\vec{\hat{y}}$ (la meilleure position globale, désignant l'expérience de particules voisines).

Selon la taille du voisinage, on peut identifier deux modèles de *PSO* : le modèle *Gbest*, si le voisinage d'une particule est l'ensemble de l'essaim, et le modèle *Lbest* dans lequel le voisinage d'une particule est un sous-ensemble de l'essaim.

La performance de chaque particule (comment les particules se rapprochent de l'optimum global) est mesurée par une fonction fitness qui dépend du problème d'optimisation à traiter.

La meilleure position personnelle de la particule i est la meilleure position (c'est-à-dire, celle qui résulte de la meilleure valeur de la fitness) visitée par particule i . Soit f la fonction objectif (nous supposons qu'il s'agit en fait de minimisation), alors la meilleure position personnelle d'une particule est mise à jour comme suit :

$$y_i(t+1) = \begin{cases} y_i(t) & \text{si } f(x_i(t+1)) \succeq f(y_i(t)) \\ x_i(t+1) & \text{si } f(x_i(t+1)) < f(y_i(t)) \end{cases} \quad (3.7)$$

La meilleure position globale est déterminée selon le modèle *Gbest* comme suit :

$$\hat{y}(t) \in \{y_0(t), y_1(t), \dots, y_s(t)\} \text{ et } f(\hat{y}(t)) = \min \{f(y_0(t)), f(y_1(t)), \dots, f(y_s(t))\} \quad (3.8)$$

où s dénote la dimension de l'essaim.

Pour chaque itération de l'algorithme *PSO*, l'étape de la mise à jour de la vitesse v_i est spécifiée pour chaque dimension $j \in 1..N_d$, où N_d est la dimension du problème (espace de recherche). D'où, $v_{i,j}$ représente le $j^{\text{ième}}$ élément du vecteur de la vitesse de la particule i . La mise à jour de la vitesse de particule est :

$$v_{ij}(t+1) = \omega v_{ij}(t) + \rho_{1j}(y_{ij}(t) - x_{ij}(t)) + \rho_{2j}(\hat{y}_i(t) - x_{ij}(t)) \quad (3.9)$$

Tels que :

- ρ_1 et ρ_2 sont les coefficients de confiance, ils pondèrent les tendances de la particule à vouloir suivre son instinct de conservation ou son panurgisme. Les variables aléatoires ρ_1 et ρ_2 peuvent être définies de la façon suivante :

$$\begin{cases} \rho_1 = r_1 c_1 \\ \rho_2 = r_2 c_2 \end{cases} \quad (3.10)$$

où r_1 et r_2 suivent une loi uniforme sur $[0, 1]$ et c_1 et c_2 sont des constantes positives déterminées de façon empirique [217], et suivant la relation $c_1 + c_2 \leq 4$.

- ω est le facteur d'inertie qui a été introduit, pour la première fois, par Shi et Eberhart [198], il sert comme une mémoire de vitesses antérieures, pour contrôler l'influence de la vitesse obtenue au pas précédent. Un grand facteur d'inertie provoque une grande exploration de l'espace de recherche alors qu'un petit facteur d'inertie concentre la recherche sur un petit espace.
- La composante cognitive, $y_i(t) - x_i(t)$, représente l'expérience individuelle de la particule.
- La composante sociale $\hat{y}_i(t) - x_i(t)$, représente la communication sociale (la croyance de l'essaim).

Selon Van den Bergh [217], la relation entre le coefficient de l'inertie et les constantes d'accélération devraient satisfaire l'équation suivante pour garantir la convergence.

$$\frac{c_1 + c_2}{2} - 1 < \omega \leq 1 \quad (3.11)$$

Autrement, les particules peuvent exposer une divergence de comportement cyclique. Les détails sur la relation entre le coefficient de l'inertie et les constantes d'accélération se trouvent dans [46]-[217]-[199].

La modification de vitesse peut aussi être limitée par une vitesse maximale V_{max} définie par l'utilisateur, pour éviter que les particules se déplacent trop rapidement d'une région à une autre dans l'espace de recherche, et provoquer une convergence prématurée [217].

La position x_i de la particule i , est modifiée en utilisant l'équation suivante :

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (3.12)$$

Les équations de mise à jour de position et de vitesse sont exécutées plusieurs fois, jusqu'à atteindre un nombre maximal d'itérations, donné par l'utilisateur, ou un manque de progrès dans l'amélioration de la meilleure solution trouvée, pour un nombre d'itérations consécutif spécifié par l'utilisateur.

La qualité de particules est mesurée en utilisant une fonction fitness f , qui reflète l'optimalité d'une solution particulière. L'algorithme de base de *PSO* se décrit comme suit :

Algorithme 3.5 : Algorithme *PSO*

Pour chaque particule $i \in 1, \dots, s$ **Faire**
 Initialiser aléatoirement x_i ;
 Initialiser aléatoirement v_i ;
 Mettre $y_i = x_i$;
Fin pour
Répéter
Pour chaque particule $i \in 1, \dots, s$ **Faire**
 Calculer la fitness $f(x_i)$ de la particule i ;
 Mettre à jour y_i en utilisant l'équation (3.7) ;
 Mettre à jour \hat{y} en utilisant l'équation (3.8) ;
Pour chaque dimension $j \in 1 \dots, N_d$ **Faire**
 Mettre à jour la vitesse $v_{i,j}$ en utilisant l'équation (3.9) ;
FinPour
 Mettre à jour la position x_i (3.12) ;
FinPour
Jusqu'au critère d'arrêt soit satisfait

3.5.4 Modèle *Lbest*

Dans le modèle *Lbest*, un essaim est divisé en des voisinages chevauchants de particules. Pour chaque voisinage N_i , la meilleure particule est déterminée, avec la position \hat{y}_i . Cette position représente la meilleure position locale (dans le voisinage N_i). Soient i l'indice d'une particule i , et l est la dimension du voisinage, alors les équations de mise à jour sont :

$$N_i = \{y_{i-l}(t), y_{i-l+1}(t), \dots, y_{i-1}(t), y_i(t), y_{i+1}, \dots, y_{i+l}\} \quad (3.13)$$

$$\hat{y}_i(t+1) \in \{N_i / f(\hat{y}_i(t+1)) = \min \{f(y_i(t))\}, \forall y_i \in N_i\} \quad (3.14)$$

$$v_{ij}(t+1) = \omega v_{ij}(t) + \rho_{1j}(y_{ij}(t) - x_{ij}(t)) + \rho_{2j}(\hat{y}_i(t) - x_{ij}(t)) \quad (3.15)$$

L'équation de mise à jour de la position est la même ; comme elle est donnée par l'équation (3.12). Les voisins représentent le facteur social dans le *PSO*. Les voisinages

sont généralement déterminés, en utilisant les indices de particules ; cependant, les voisinages topologiques aussi peuvent être utilisés. Il est clair que le modèle *Gbest* est un cas spécial du modèle *Lbest* avec $l = s$.

3.5.5 Optimisation par essaim de particules binaires

Kennedy et Eberhart ont adapté le *PSO* pour la recherche dans l'espace binaire [127] et ont développé l'optimisation par essaim de particules binaires (Binary Particle Swarm Optimization "*BPSO*"). Dans le *BPSO*, les valeurs de composantes de x_i et y_i sont restreintes à l'ensemble $\{0, 1\}$. La vitesse v_i est interprétée comme étant une probabilité de changer un bit de 0 à 1, ou de 1 à 0, en modifiant la position de la particule i . Cela peut être réalisée à l'aide d'une fonction sigmoïde, définie par :

$$sig(x) = \frac{1}{1 + \exp(-x)} \quad (3.16)$$

Donc, l'équation de mise à jour de position (3.12), est remplacée par l'équation de mise à jour probabiliste suivante [217] :

$$x_{ij}(t+1) = \begin{cases} 0 & si \quad r_j(t) \succeq sig(v_{ij}(t+1)) \\ 1 & si \quad r_j(t) \prec sig(v_{ij}(t+1)) \end{cases} \quad (3.17)$$

avec $r_j(t) \sim U(0, 1)$

A partir de cette équation, on peut observer que si $sig(v_{i,j}) = 0$ alors $x_{i,j} = 0$. Cette situation se produit quand $v_{i,j} < -10$. En outre, $sig(v_{i,j}) = 1$ satisfera quand $v_{i,j} > 10$.

Pour éviter ce problème, il est suggéré de mettre $v_{i,j} = [-4, 4]$ et d'utiliser la vitesse limitant avec $Vmax = 4$ [129].

PSO a été aussi étendu pour traiter la représentation discrète arbitraire. Ces extensions sont, généralement, accomplies en arrondissant x à sa valeur discrète la plus proche après avoir appliqué l'équation de mise à jour de position (3.12).

3.6 Système multi-agents

Le domaine des systèmes multi-agents (*SMA*) est actuellement un domaine de recherche qui suscite beaucoup d'intérêt. Ce domaine est né à la fin des années 70 et début des années 80, de l'idée de distribuer les connaissances et le contrôle dans les systèmes d'Intelligence Artificielle (*IA*). Cette idée a émergé d'une part du besoin de faire face à la complexité croissante de ces systèmes et a été favorisée d'autre part par l'émergence des modèles et machines parallèles, rendant possible la mise en œuvre

opérationnelle de la distribution. Une bonne description de l'évolution historique de ce domaine est donnée par Ferber dans [81].

Un système multi-agents est défini comme un ensemble d'entités (physiques ou virtuelles) appelées agents, partageant un environnement commun (physique ou virtuel), qu'elles sont capables de percevoir et sur lequel elles peuvent agir. Les perceptions permettent aux agents d'acquérir des informations sur l'évolution de leur environnement, et leurs actions leur permettent entre autres de modifier l'environnement. Les agents interagissent entre eux directement ou indirectement, et exhibent des comportements corrélés créant ainsi une synergie permettant à l'ensemble des agents de former un collectif organisé [108].

L'analyse, le design, l'implémentation et le déploiement d'un système multi-agents peuvent être étudiés en fonction de quatre aspects fondamentaux : *AEIO* définie par Demazeau [66] :

- Agents : l'architecture interne des agents.
- Environnement : le milieu dans lequel évoluent les agents.
- Interaction : les moyens par lesquels les agents interagissent.
- Organisation : les moyens utilisés pour structurer l'ensemble des entités.

Actuellement, il existe deux tendances de recherche dans le domaine des multi-agents : la tendance cognitive, influencée par l'*IA* symbolique, et la tendance réactive influencée par la Vie Artificielle et l'intelligence "sans représentation" selon Brooks [33], appelée aussi embodied intelligence [108].

Le domaine des agents situés est né à la fin des années 80, à partir de travaux dans le domaine de la robotique mobile. Dans ces travaux, la prise en compte d'un environnement complexe et dynamique, rend le problème de sélection d'action difficile à résoudre avec des approches délibératives classiques [32]. La mise en situation de l'agent a donc été proposée, pour permettre des comportements plus réactifs, face à la dynamique de l'environnement. Dans ce contexte, l'approche comportementale a été proposée, elle est inspirée de la biologie pour la robotique mobile, comme moyen de mise en œuvre d'une intelligence "sans représentation" [33].

Cette approche s'est ensuite étendue à d'autres problématiques, comme la résolution distribuée de problèmes [82]-[79] ou la simulation multi-agents, qui a émergé depuis 1998 comme domaine de recherche à part entière [108].

3.7 Conclusion

Nous avons présenté, dans ce chapitre, un exposé non-exhaustif sur les méthodes bio-inspirées. Nous avons insisté, beaucoup plus, sur l'optimisation par essaim de par-

ticules qui présente un axe important dans nos travaux. Nous avons tenté de mettre en évidence les mécanismes d'adaptation, dans ces méthodes, à travers la notion d'auto-organisation.

Ce travail, nous a permis de mieux comprendre le fonctionnement des méthodes existantes, d'orienter la conception de nouvelles méthodes bio-inspirées, et de retenir les points suivants :

- Les techniques bio-inspirées nous permettent d'accéder à un espace plus large de solutions.
- Les solutions bio-inspirées nous offrent un éventail de caractéristiques telles que : l'adaptation, la tolérance aux bruit et aux pannes, le calcul distribué, la généralisation.

Chapitre 4

Etat de l'art des méthodes de segmentation d'images

4.1 Introduction

La segmentation d'images est une opération de traitement d'images de bas niveau qui consiste à localiser, dans une image, les régions (ensembles de pixels) appartenant à une même structure (objets ou scène imagés). Cette opération est à la base de nombreuses applications tant en vision industrielle, qu'en imagerie médicale. Le problème de segmentation a été, et encore est, un champ de la recherche important et beaucoup de méthodes de la segmentation ont été proposées dans la littérature (Pour plus de détails, voir l'étude de [106]-[175]. Par conséquent, plusieurs classifications des méthodes de segmentation ont été proposées dans les différents ouvrages dédiés au traitement des images. Nous avons opté pour la classification suivante :

- Méthodes de segmentation basées contours ;
- Méthodes de segmentation basées régions ;
- Méthodes de segmentation par seuillage ;
- Méthodes de segmentation par classification ;
- Méthodes de segmentation coopératives.

Dans ce chapitre, nous décrivons brièvement ces différentes classes de méthodes. Toutefois, l'accent est mis sur les méthodes de segmentation par seuillage et par régions. Nous présentons un état de l'art, pour ces deux méthodes, comportant les traditionnelles méthodes et de nouvelles méthodes qui sont basées sur la distribution et les techniques d'optimisation.

4.2 Segmentation d'images basées contours

La recherche des contours, dans une image numérique, est un des problèmes les plus étudiés depuis l'origine des travaux sur l'imagerie numérique. Ceci est en grande partie dû à la nature très intuitive du contour qui apparaît très naturellement comme l'indice visuel idéal dans la plus grande partie des situations.

Selon Pavlidis [178], un contour est une zone de transition séparant deux régions homogènes due à des discontinuités de profondeur et de réflectance de la scène observée. Les méthodes de segmentation basées sur l'approche contour ont donc pour objectif de trouver les lieux de fortes variations du niveau de gris. Un nombre important de méthodes a été développé. Ces méthodes s'appuient sur la détection des discontinuités dans l'image et peuvent être divisées en trois classes [166] : les méthodes dérivatives, les méthodes analytiques et les méthodes basées sur les contours actifs.

4.3 Segmentation d'images basées régions

Les méthodes de segmentation basées régions ont pour but de décomposer l'image en un ensemble de régions connexes suivant un critère d'homogénéité [119]-[92]. Une région est composée de l'ensemble des pixels connexes possédant les mêmes propriétés au sens d'un critère d'homogénéité donné. La formulation mathématique du problème de la segmentation d'une image I en différentes régions homogènes X_1, \dots, X_n selon un critère d'homogénéité H est présentée ci-dessous.

- $\cup_{i=1}^n X_i = I$ et $\forall i \neq j, X_i \cap X_j = \emptyset$.
- $\forall i X_i$ est connexe.
- $\forall i H[X_i]$ est vrai, la région X_i est dite homogène.
- $\forall (X_i, X_j)$ région voisines, $H[X_i, X_j]$ est faux.

Cette définition conduit à deux remarques très importantes. Tout d'abord, une segmentation dépend du critère employé d'homogénéité (H) [166]. Le choix du critère est donc primordial, il dépend du problème traité et du type d'images, d'où la large palette de critères d'homogénéité qui ont été utilisés [93]-[193]-[15]. A titre d'exemple, d'un point de vue statistique, la variance Var des niveaux de gris dans une région X peut être utilisée comme critère d'homogénéité (H) :

$$Var = \frac{1}{N} \sum_{(x,y) \in X} \left[I(x,y) - \frac{1}{N} \sum_{(x,y) \in X} [I(x,y)] \right]^2 \quad (4.1)$$

Ensuite, le nombre de régions obtenues est variable en fonction des critères d'homogénéité ainsi qu'aux seuils de tolérance contrôlant leur formation.

Dans ce type de méthodes de segmentation, en plus de l'information niveau de gris, l'information spatiale est automatiquement intégrée dans le processus de segmentation pour la formation de régions connexes et homogènes au sens d'un critère prédéfini [177]. Le contrôle géométrique peut être aussi utilisé pour contrôler le processus de segmentation [173]. Le problème fondamental associé à cette catégorie d'approches est justement lié à la définition du critère d'homogénéité utilisé dans le processus de segmentation de l'image.

La croissance de régions "*RG*" (*Region Growing*) est la méthode courante de cette catégorie [180] qui consiste à obtenir des régions homogènes d'une façon progressive [24]. Elle regroupe toutes les techniques d'agrégation de pixels, de fusion-division des régions.

4.3.1 Agrégation de pixels

Dans ce type de méthodes, chaque région croît pixel par pixel avec un processus itératif; le point de départ est le choix d'un ensemble de pixels appelés « *germes* ». À chaque *germe* on associe un vecteur de caractéristiques. Chaque *germe* est agrégé avec le premier pixel connexe qui possède un vecteur de caractéristiques suffisamment proche du sien. Divers vecteurs de caractéristiques et diverses mesures de similarité ont été proposés dans la littérature [47]. Généralement le vecteur de caractéristique est réduit à la valeur de niveau de gris et la distance euclidienne est utilisée comme mesure de similarité. Pour améliorer la robustesse, il est préférable de prendre en compte une partie du voisinage de chaque pixel [192] et d'utiliser des propriétés statistiques telles que la moyenne ou la variance. Les régions ainsi construites dépendent fortement du choix des *germes* de départ ainsi que de l'ordre de parcours des pixels agrégés et le choix de seuil de similarité.

Le choix des points de départ (les pixels *germes*) est une partie critique de l'algorithme. En effet, l'étape de croissance va utiliser une mesure de similarité pour choisir les pixels à agréger. Si le point de départ est situé dans une zone non homogène, la mesure de similarité va produire de fortes variations et la croissance va s'arrêter très tôt. Par conséquent, il convient de choisir les points de départs dans des zones les plus homogènes possibles.

4.3.2 Approche division-fusion

Le principe de ce type de méthode, est de partitionner progressivement l'image en régions, selon des critères de division plus ou moins complexes faisant intervenir par exemple la différence de moyenne des deux régions, leurs variances, leur nombre de pixels et le nombre de pixels adjacents, la différence entre deux niveaux de gris

d'une même région soit inférieure à un seuil. Puis des fusions des régions connexes et similaires sont faites au sens des critères de regroupement, par exemple : la différence de moyenne de niveau de gris soit inférieure à un seuil. Tous les couples qui ont une différence de moyenne inférieure à un seuil sont fusionnés.

La méthode de l'arbre quaternaire (*Quadtree*) est souvent utilisée pour réaliser l'étape de division [47]. Il s'agit d'une décomposition récursive en quatre sous régions carrées, des régions ne vérifiant pas le critère de division. En commençant par la racine de l'arbre qui est l'image entière comme bloc initial, supposée être de taille $N \times N$ avec $N = 2^n$, puis on examine le contenu de ce bloc. Si le bloc est homogène (i.e. contient exclusivement des pixels similaires) alors on arrête la décomposition. Sinon, on découpe la région en 4 sous-blocs (4 nœuds fils), et on examine le contenu de chaque sous-blocs. Et ainsi de suite jusqu'à ce qu'il n'y ait plus besoin de décomposer les blocs. La décomposition finale est définie par les blocs associés aux feuilles de l'arbre. On obtient ainsi une liste de blocs de différentes tailles et positions. Le résultat obtenu est, donc, un ensemble jointif de blocs de différentes tailles qui recouvrent entièrement l'image (FIG. 4.1).

Un exemple illustrant cette méthode, pris de [166] est présenté sur la figure FIG. 4.2. L'image originale est segmentée, en utilisant un seuil de tolérance de l'heuristique (critère) d'homogénéité de 0.2 (choisi arbitrairement). L'heuristique utilisée dans cet exemple veut que la différence, entre les valeurs maximales et minimales des pixels.

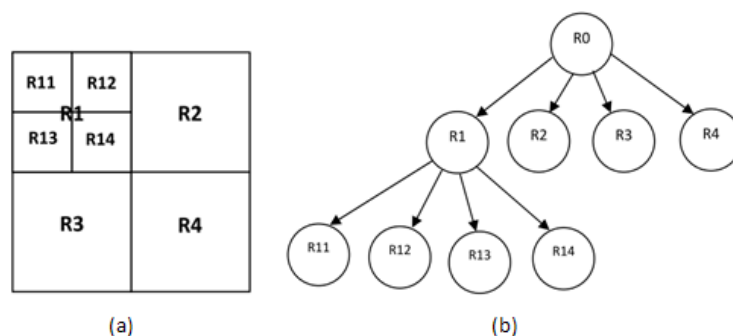


FIG. 4.1 – Représentation sous forme de Quadtree, (a) : Décomposition de l'image en blocs ; (b) : Diagramme hiérarchique de la décomposition.

Des problèmes combinatoires apparaissent avec cette méthode, en particulier, la nécessité de maintenir à jour un graphe d'adjacence des régions, pour pouvoir sélectionner rapidement toutes les régions voisines d'une région courante. Il devient aussi indispensable d'ordonner les couples de régions fusionnable en fonction de la valeur de leur prédicat d'homogénéité, dans le but de fusionner les régions les plus similaires en

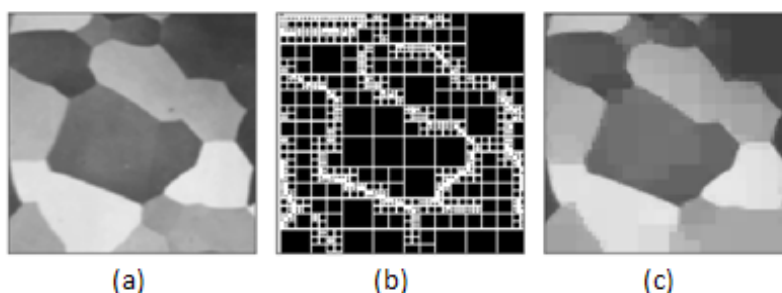


FIG. 4.2 – Illustration de la méthode division-fusion, (a) : image originale, (b) : image de blocs déterminés par division, (c) : image segmentée.

premier.

Horowitz et Pavlidis sont les premiers à avoir proposé une telle approche de segmentation "*split and merge*" [116]. Mais selon les règles de fusion, plusieurs techniques sont développées, telles que la croissance hiérarchique de Gagalowicz et Monga [94] ainsi que la pyramide adaptative inspirée des travaux de Jolion et Montanvert [121]-[122], Meer [153], Kropatsch [135] et de Chiarello et al. [41].

La croissance hiérarchique de Gagalowicz et Monga [94] consiste à une utilisation des critères hiérarchiques afin de guider un processus de fusion. Chaque critère est choisi pour être un peu moins restrictif que le précédent, et donc pour permettre de fusionner quelques régions supplémentaires par rapport au précédent critère. Les étapes principales de cette méthode sont :

1. Une première carte des régions est obtenue par *split and merge* (division et fusion) en utilisant un critère très strict qui impose que la différence entre deux niveaux de gris d'une même région soit inférieure à un seuil. Ce seuil est choisi très faible.
2. Tous les couples de régions adjacentes sont classés par ordre de différence de moyenne de niveau de gris. Tous les couples qui ont une différence de moyenne inférieure à un seuil sont fusionnés. Chaque fusion entraîne une mise à jour dans le graphe d'adjacence et le classement des couples.
3. Le même principe de fusion est appliqué, en classant les couples de régions adjacentes en fonction de la valeur de la variance résultant de l'union des deux régions. Un seuil de variance maximale acceptable est utilisé.
4. Tous les couples de régions qui ont un gradient moyen à la frontière inférieur à un seuil sont fusionnés.
5. Les petites régions de taille inférieure à un seuil sont rattachées aux plus grandes.

L'utilisation des critères hiérarchiques permet, d'une part, de limiter le problème du choix du critère de fusion et d'autre part, de corriger les erreurs de segmentation produite à un niveau supérieur. Cependant le temps de calcul par rapport aux méthodes de fusion classiques est plus important. Une autre difficulté de cette méthode réside dans l'ordre séquentiel de l'application de ces critères de fusion. La fusion idéale devrait exploiter par l'intégration de plusieurs de ces critères, de manière pondérée, au sein d'une même fonction de décision. Ceci permettrait de définir clairement l'importance que l'on apporte à chaque critère de fusion autrement que simplement par son ordre d'apparition dans le système [15].

Pour éviter la structure très séquentielle de cette méthode, l'approche pyramide adaptative consiste à effectuer les meilleures fusions locales de façon simultanée, puis à recommencer avec les régions formées au niveau suivant et ainsi de suite jusqu'à stabilité de la carte des régions. L'idée de base de cette approche est le suivant [121]-[163] : Etant donné un partitionnement initial de l'image en régions (un pixel est considéré comme une région), il est possible de classer toutes les fusions possibles selon un critère d'homogénéité et d'effectuer à chaque fois la meilleure fusion. Après chaque fusion, il y a mise à jour du graphe d'adjacence et du classement des fusions potentielles.

Chiarello et al. [41] proposent une croissance de région fondée sur la gestion d'une pyramide stochastique. En partant de choix aléatoires pour le mécanisme de décimation et d'expansion, ils orientent la carte région résultat en ajustant les fonctions de décision (au départ purement aléatoires) selon les besoins de leur propre application (une application de génération de cartographie cohérente). Le choix des régions devant disparaître (décimation), ainsi que le choix des régions devant être étendues dans les zones décimées (expansion) sont les deux points de contrôle de cette approche. L'originalité de ce travail est de restreindre le nombre de régions candidates à la fusion aux seules zones issues d'une décimation, ce qui repose sur un critère de choix purement géographique.

L'intégration de l'information sur les contours comme des critères additionnels pour fusionner et diviser les régions, ou pour agréger les pixels, a été utilisée par plusieurs travaux, comme nous allons voir dans le paragraphe suivant, cette intégration a été faite pour résoudre les problèmes liés à la difficulté d'adapter la décision au problème traité dans les approches de croissance de région (par agrégation de pixels ou par la division-région) [14].

Le premier type de ces problèmes apparaît lors de l'analyse des pixels situés près de la frontière d'une région. Il est en effet très difficile de déterminer à quel endroit la croissance de la région doit se terminer, les propriétés locales près des frontières étant nettement différentes des propriétés moyennes de la région, et pouvant biaiser la

suite des traitements . Un détecteur de contours peut déterminer la position exacte de la frontière. [177]. Dans ce cas, il n'est plus nécessaire de faire croître la région pour qu'elle épouse sa frontière.

Le second type problème est rencontré lorsque deux régions voisines de même nature sont séparées par une frontière faiblement marquée. La plupart des méthodes de croissance de région qui vont effectuer la fusion de ces deux régions en une seule (généralement car les seuils utilisés sont globaux à toute l'image) fourniront au niveaux supérieur un objet pratiquement inexploitable par la suite, puisque ses propriétés géométriques et photométriques ne sont plus représentatives d'aucune des deux parties qui le composent. Une gestion rigoureuse des informations et des prises de décision [192], et une croissance s'interrompant régulièrement pour rechercher d'éventuels faibles contours est souhaitable pour éviter ce type d'erreur.

L'exploitation des informations sur les contours pour contrôler l'expansion ou la fusion des régions a conduit à une naissance des approches coopératives.

4.4 Approches coopératives

Les approches coopératives de la segmentation peuvent être classées en trois catégories [194] : la coopération séquentielle, la coopération des résultats, la coopération mutuelle. Cette classification se manifeste d'une part, dans l'étape où l'intégration entre l'information sur les régions et celle sur les contours est réalisée. D'autre part, elle apparaît dans la façon que cette intégration est produite.

4.4.1 Coopération séquentielle

Dans la coopération séquentielle, l'une des techniques de segmentation (région ou contour) est réalisée en premier lieu ; son résultat va être exploité par l'autre technique pour renforcer la définition des critères ou des paramètres de la segmentation ; L'image contours peut être intégrée pour :

- forcer ou interdire des fusions entre les régions voisines [17] ; l'information sur le contour peut renseigner sur la possibilité d'une fusion entre deux régions adjacentes similaires, par exemple, Wrobel et Monga [222] proposent d'appliquer d'abord un détecteur de contours, puis de contraindre une méthode de fusion des régions en prenant en compte la présence des contours à leurs frontières. Benois et Barba [16] utilisent une carte précalculée des contours robustes de l'image pour contraindre une méthode de segmentation en régions de type *Split and Merge*. Le problème de l'ordre de fusion des régions est résolu à l'aide de la

carte des contours : la fusion privilégie les zones du quad-tree les plus éloignées de tout pixel contour (présümées les plus homogènes).

- la croissance des régions ; la formation de la région s'effectue sous la forme d'ajouts seulement les pixels ayant un faible gradient [223] ; la croissance de région est arrêtée quand elle rencontre un pixel à fort gradient, c'est-à-dire un contour. La coopération peut également s'effectuer sous la forme d'ajouts de pixels à la région jusqu'à ce qu'un maximum de gradient moyen calculé sur la région soit atteint [96].
- guider l'emplacement des germes de croissance de régions. En premier lieu, l'image des contours est obtenue par une ou plusieurs segmentations par contours. Ensuite, les centroïdes des régions entre ces contours sont pris comme germes initiaux pour une croissance de régions [131].
- accélérer le découpage de l'image lors du processus de division de l'image en quadrants homogènes [25], En effet, la présence de contours est une information sur la non homogénéité du quadrant. Par conséquent, l'intégration de l'information sur les contours permet d'accélérer le temps de traitement.
- ajuster les paramètres de segmentation par région, par exemple, l'analyse d'échantillons de pixels (germes) de part et d'autre de chaque contour permettra de rendre le critère d'homogénéité plus adéquat par rapport aux caractéristiques des régions [165].

Les résultats d'une segmentation par régions peuvent constituer les entrées d'un détecteur de contours. Par exemple, les limites obtenues par une segmentation par régions peuvent être considérées comme point de départ d'un détecteur de contours de type "snakes" [177]. Le contour initial est progressivement déformé, guidé par la minimisation d'une fonction d'énergie. Le détecteur de contours pourra ainsi localiser plus précisément les limites entre les régions.

4.4.2 Coopération des résultats

Dans la coopération des résultats, les deux types de segmentations sont réalisés d'une façon parallèle et indépendante, et la coopération sera faite au niveau de leurs résultats respectifs. Ils seront intégrés dans le but :

1. d'obtenir une segmentation plus complète [117], par exemple, dans [230], les informations sur les pixels-contours, provenant d'une segmentation à l'aide d'un opérateur de gradient, et les pixels des limites de régions, obtenus à l'aide d'une croissance de région, se complètent afin d'obtenir plus de contours détectés et d'assurer le raccordement entre les contours.

2. d'ajouter un critère additionnel aux résultats de segmentation obtenus. Le résultat de la segmentation par régions peut être amélioré, par exemple, en intégrant le résultat de la segmentation par contours [190]. Les régions sont subdivisées suivant la carte de contours de sorte qu'aucune région ne contient de contours. Les régions subdivisées de mêmes caractéristiques sont par la suite fusionnées en exploitant les informations sur les régions et sur les contours.
3. d'établir un consensus entre les résultats de diverses segmentations ; par exemple : dans [43], les résultats de diverses segmentations permettent la construction d'un graphe d'adjacence entre les régions indiquant, pour chaque paire de pixels, la probabilité d'appartenance à la même région (probabilité de cooccurrence). Les couples de pixels ayant une grande probabilité de cooccurrence sont ensuite groupés ensemble. Dans [44], des pondérations sont attribuées à chaque segmentation en fonction de leur degré de fiabilité. Toutes les segmentations sont transformées en cartes contours. Une procédure itérative, utilisant un estimateur de maximum de vraisemblance, est appliquée afin de converger vers un consensus.
4. d'ajuster les paramètres de segmentation par comparaison des résultats de différentes techniques de segmentation par exemple dans [130], l'ajustement a été intégré dans un processus itératif ; les itérations sont faites avec des critères de plus en plus tolérants jusqu'à la convergence vers des résultats cohérents et stables. La vérification de la cohérence est basée sur la minimisation de la dissimilarité entre les contours et les régions, jusqu'à la compatibilité entre les résultats contours et régions.
5. d'évaluer les résultats de segmentation dans l'absence d'informations de réalité de terrain ; plusieurs segmentations par régions avec différents paramètres et seuils sont réalisées, pour juger de la meilleure segmentation, les résultats sont comparés à une segmentation par contours. La segmentation dont les limites de régions sont les plus proches des contours sera retenue.

4.4.3 Coopération mutuelle

Dans l'approche de coopération mutuelle, les différentes techniques de segmentation sont exécutées en parallèle, tout en échangeant mutuellement des informations. L'information échangée sert à aider la technique de segmentation dans la prise de décisions plus sûres et plus fiables dans le cas de lacunes ou d'informations insuffisantes [193]-[15], comme elle peut être appliquée afin d'améliorer les résultats obtenus par des segmentations par régions et par contours [175].

1. la prise de décision dans le cas de lacunes ou d'informations insuffisantes, par exemple : dans [193] ; la coopération est faite entre un détecteur de contours et un

processus de croissance de régions. Les contours sont détectés par un détecteur de contours basé sur les grandes valeurs de gradients. Dans les cas où il est difficile de trouver le fort gradient, l'analyse des régions de part et d'autre du contour donne une information additionnelle : un nouveau seuil adapté à cette partie de l'image à faibles valeurs du gradient est calculé. De même, dans la segmentation par croissance de régions, les pixels pour lesquels la décision d'agrégation est difficile à prendre sont étudiés par un détecteur de contours pour savoir si ce sont des contours ou non. Dans [14] une approche est faite entre plusieurs techniques de segmentation. Elle est intégrée dans un processus de décision dans lequel les décisions difficiles sont mises en instance en attendant plus d'information. Ainsi, dans le cas où une technique de segmentation n'aboutit pas localement à une décision, le processus va recourir à une autre technique. Une fois la nouvelle information obtenue, elle est prise en compte par le processus pour prendre une décision et poursuivre la segmentation.

- L'amélioration des résultats obtenus par des segmentations par régions et par contours. Par exemple dans [167], un ensemble de règles est appliqué d'abord sur les régions et les contours détectés afin de les analyser et de définir des règles pour obtenir une segmentation plus fidèle. Ainsi, des règles liées au raccordement, à la suppression et à la prolongation des contours détectés sont définies en prenant en compte, entre autres, l'information sur les régions adjacentes. Aussi, les règles liées à la fusion ou à la division des régions détectées prennent en considération aussi la présence des contours.

4.5 Segmentation d'images par classification

La segmentation d'une image par une méthode de classification consiste à partitionner l'image en un ensemble de classes, telle que chaque classe rassemble des pixels ayant des vecteurs de caractéristiques (souvent réduit au niveau de gris) aussi similaires que possible et que les classes soient aussi distinctes que possible les unes des autres. Généralement, ces méthodes de segmentation ne prennent pas en compte la disposition spatiale des pixels et ne considèrent que le vecteur de caractéristiques utilisé. Elles identifient les classes de pixels présentes dans l'image et affectent à chaque pixel une étiquette indiquant la classe à laquelle il appartient. Ainsi les pixels appartenant à une classe peuvent former plusieurs régions non adjacentes dans l'image mais partageant les mêmes propriétés statistiques. Une segmentation en régions n'est obtenue qu'après analyse de la connexité des pixels dans l'image étiquetée. La plupart des algorithmes de classification sont basés sur une technique de partitionnement [90]-[143]-[171].

Le problème de partitionnement peut s'exprimer comme suit : étant donné une

image de N pixels, on cherche à regrouper ces pixels en K classes de telle sorte que les pixels d'une même classe soient plus proche entre eux en terme d'un (ou plusieurs) critère, qu'avec les pixels des autres classes. C'est un problème d'optimisation $NP - difficile$ [143]-[119]; pour trouver la partition qui minimise au mieux le critère de partitionnement, il faut énumérer toutes les partitions possibles de N pixels en K classes et choisir parmi elles celle pour qui ce critère est minimal. Le nombre des partitions possibles noté $S(N, K)$ est donné par la formule suivante :

$$S(N, K) = \frac{1}{K!} \sum_{i=1}^k (-1)^{k-1} \begin{bmatrix} k \\ i \end{bmatrix} (i)^N \quad (4.2)$$

On peut rendre compte que pour un nombre de classes et de pixels même très réduit, le nombre de combinaison explose considérablement. Il n'est pas possible alors d'énumérer complètement les partitions de l'image pour déterminer exactement la valeur optimale du critère de partitionnement. Une solution possible est d'utiliser des heuristiques qui permettent de calculer une valeur approchée de la valeur optimale [173]. La plupart des heuristiques de partitionnement minimisent l'inertie intraclasse de la partition I_{intra} , ou bien maximisent l'inertie interclasse de la partition I_{inter} , ou bien l'inertie totale I définie comme suit :

$$I = I_{intra} + I_{inter} \quad (4.3)$$

$$I_{intra} = \sum_{j=1}^k \sum_{i=1}^{|c_j|} d^2(X_i^{(j)}, \mu_j) \quad (4.4)$$

$$I_{inter} = \sum_{j=1}^k |c_j| d^2(\mu_j, \mu) \quad (4.5)$$

avec :

$X_i^{(j)}$: est le vecteur caractéristique du $i^{\text{ème}}$ pixel de la classe c_j ;

μ_j : est le vecteur de caractéristiques moyen (le centre de gravité) de la classe c_j .

$|c_j|$: est la taille de la classe c_j ;

μ : est le vecteur de caractéristiques moyen de l'image à segmenter ;

$d(\cdot)$: est une mesure de distance, la plus connue est la distance euclidienne.

Le vecteur de caractéristiques moyen (le centre de gravité) de la classe μ_j est calculé comme suit :

$$\mu_j = \frac{1}{|c_j|} \sum_{i=1}^{|c_j|} X_i^{(j)} \quad (4.6)$$

L'algorithme des centres mobiles K-means [86] est un algorithme de partitionnement des plus simples et des plus connus. Partant de centres de classes choisis arbitrairement parmi les vecteurs de caractéristiques de pixels (le vecteur de caractéristiques peut être réduit à une seule caractéristique qui est le niveau de gris du pixel) de l'image, le regroupement de pixels est itératif et est obtenu par minimisation de l'inertie intra-classe. La classification optimale est celle qui permet de minimiser l'inertie intra-classe I_{intra} (Eq.4.4). L'algorithme 4.1 résume les étapes de classification par les centres mobiles [159]-[166]. Une segmentation d'une image microscopique par cet algorithme est présentée dans la figure FIG. 4.3.

Algorithme 4.1 : Algorithme des centres mobiles

1. Définir un nombre de classes K ;
2. Définir une classification C aléatoire : choisir les K centres (μ_i) de classes de façon aléatoire, dans l'espace D des niveaux de gris $[0, L - 1]$;
3. **Tant que** l'inertie intra-classe I_{intra} n'est pas stable **Faire**
 - 3.1 Affecter chaque niveau de gris à la classe dont le centre est le plus proche ;
 - 3.2 Calculer les centres de gravité ($\mu_i, i = 1, K$) des classes de la nouvelle classification C' des classes ;
 - 3.3 $C \leftarrow C'$;
- Fin tant que**
4. Affiche la classification obtenue ;

Le principal inconvénient des centres mobiles est que la partition finale dépend du choix de la partition initiale. Le minimum global n'est pas obligatoirement atteint, on est seulement certain d'obtenir la meilleure partition à partir de la partition de départ choisie. Le nombre de classes K fixé au début peut diminuer au cours des itérations, en effet ; si une classe n'attire aucune pixel, elle sera vide et donc éliminée. De nombreuses variantes peuvent être rencontrées, par exemple au lieu de calculer le centre des classes après avoir affecté tous les pixels, le centre peut être recalculé après chaque affectation. La méthode de centres mobiles a été généralisée sous l'appellation de la méthode des *Nuées dynamiques*. Au lieu de définir une classe par un seul point, son centre de gravité, on la définit par un groupe de points formant un « noyau ».

Afin de déterminer le nombre de classes de façon automatique et au cours de la procédure de classification, différents algorithmes de classification dynamique ont été

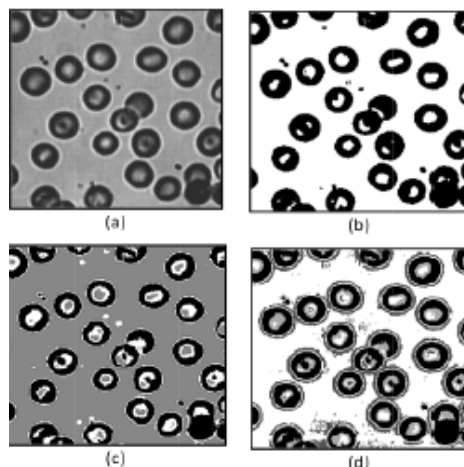


FIG. 4.3 – Exemple de segmentation d'une image microscopique par K-means, inspirée de [166] : (a) : image originale, (b) : image segmentée en 2 classes, (c) : image segmentée en 3 classes ; (d) : image segmentée en 4 classes.

proposés dans la littérature. Parmi ces algorithmes on peut citer l'algorithme *ISO-DATA* (Iterative Self-Organizing Data Analysis Techniques A) [9], et l'algorithme *Dynoc* (Dynamic optimal cluster seek) [213] qui sont basés sur le principe de division-fusion ; deux classes sont fusionnées, si la distance entre leur centres est inférieurs à un seuil prédéfini, et une classe peut être divisée en deux, si la dispersion de ses points est supérieure à un seuil prédéfini.

Plusieurs courants de penser utilisés dans le traitement du problème de partitionnement : les réseaux neuromimétiques, les champs de Markov et les méta-heureustique d'optimisation tels que les algorithmes génétiques et l'optimisation par essaim de particules ; un état de l'art sur les travaux effectués sur domaines, est présenté dans [173]-[171].

4.6 Segmentation d'images par seuillage

Le seuillage est la technique de segmentation la plus répandue pour extraire les objets du fond de l'image. Les avantages de cette technique sont sa facilité de mise en œuvre et son efficacité dans les systèmes en temps réel [47]-[166].

Cette technique se base sur l'étude la distribution des niveaux de gris des pixels (étude de la dynamique). Dans le cas d'une image composée d'une ou plusieurs régions de même type, et d'un fond visuellement différent, son histogramme est bimodal ; il possède deux modes (ou pics) et il est possible de séparer les régions du fond par un simple seuil qui se situe sur la vallée entre les deux modes (FIG. 4.4(a)). Si l'image est

constituée de régions ayant des caractéristiques différentes, son histogramme est multimodal, il possède plusieurs modes les seuils optimaux sont situés entre ces derniers (FIG. 4.4(b)), et chacun des modes correspond à une région. Une segmentation optimale par seuillage nécessite donc une localisation optimale des seuils de segmentation dans l'histogramme.

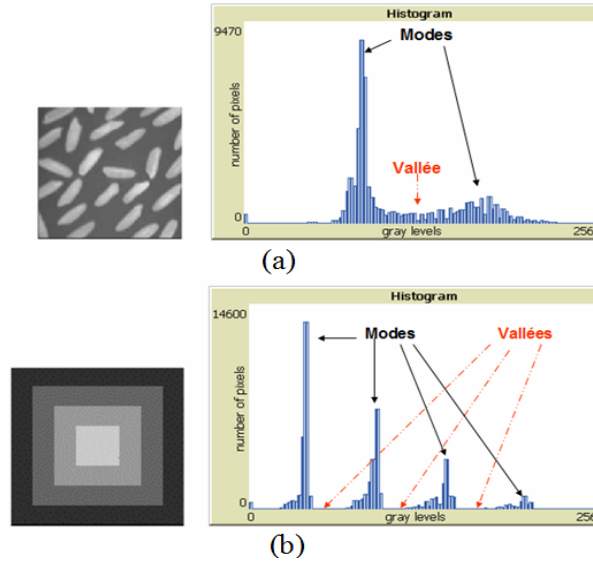


FIG. 4.4 – Exemples d'histogrammes, (a) : Histogramme bimodal, (b) : Histogramme multi-modal.

Le seuillage consiste à classer, suivant le nombre de classes, les différents pixels d'une image, en se basant sur l'histogramme de niveaux de gris de l'image. En d'autres termes, la segmentation d'une image en N classes consiste à chercher $N - 1$ seuils. Selon le nombre de seuils, le seuillage peut être, donc, classifié comme seuillage à deux niveaux ou à multi- niveaux.

Pour le seuillage à multi-niveaux, une image à niveau de gris f , dont les valeurs de niveaux de gris sont entre 0 et $L - 1$, est transformée à une image à multi-niveaux g , par un ensemble de seuils $T = \{t_1, t_2, \dots, t_k\}$ composé de k seuils, où la classification de chaque pixel, dont les coordonnées sont x et y , est définie comme suit :

$$g(x, y) = \begin{cases} 0 & \text{si} & 0 \leq f(x, y) < t_1 \\ t_i & \text{si} & t_i \leq f(x, y) < t_{i+1} \text{ et } 1 \leq i + 1 \leq k \\ t_k & \text{si} & t_k \leq f(x, y) \leq L - 1 \end{cases} \quad (4.7)$$

Dans le cas du seuillage à deux niveaux, les pixels de l'image f sont partitionnés entre deux classes selon la valeur d'un seul seuil t . les valeurs attribuées aux pixels de

l'image g peuvent être 0 ou 1, selon (Eq. 4.8), on parle, dans ce cas, de binarisation, comme elles peuvent être 0 ou $f(x, y)$ selon (Eq.4.9), on dit, est un seuillage partiel. La figure FIG. 4.5

$$g(x, y) = \begin{cases} 0 & \text{si} & 0 \leq f(x, y) < t \\ 1 & \text{si} & t \leq f(x, y) \leq L - 1 \end{cases} \quad (4.8)$$

$$g(x, y) = \begin{cases} 0 & \text{si} & 0 \leq f(x, y) < t \\ f(x, y) & \text{si} & t \leq f(x, y) \leq L - 1 \end{cases} \quad (4.9)$$

Le seuillage manuel d'une image comporte quatre étapes :

1. observation de l'histogramme de l'image,
2. choix des seuils dans les vallées,
3. définition des classes de régions par intervalles de couleurs,
4. classement des pixels.

La figure FIG. 4.5 illustre la méthode de segmentation par seuillage manuel appliquée à une image microscopique de bactéries. Cependant, dans la pratique, ce cas d'image ne se présente pas fréquemment (voir par exemple la figure FIG. 4.6), ce qui rend l'opération de segmentation plus complexe. La détermination d'un vecteur de seuils optimaux (une configuration) dans l'espace des niveaux de gris, rend la segmentation des images assimilable à un problème d'optimisation. Ceci est accompli en maximisant ou minimisant une fonction qui utilise les seuils comme des paramètres. Cette fonction est souvent connue sous le nom de fonction objectif ou de mesure du critère. Un aperçu des différents critères de segmentation d'images a été présenté dans [142]-[191]-[197]-[166].

4.6.1 Critères de seuillage

Le critère de mesure de l'entropie, proposé par Kapur et al. [123] et le critère discriminant de variance interclasse de l'Otsu [174] ont été largement utilisés dans la détermination de seuils optimaux. Les algorithmes originaux, de Kapur et Otsu, pour le seuillage optimal, ont été développés pour le seuillage à deux niveaux, mais elles s'avèrent ne pas être pratiques lorsqu'on les étend au seuillage à multi-niveaux. puisque leur complexité de calcul croît exponentiellement avec le nombre de niveaux de seuillage.

Le seuillage entropique, ou la méthode de Kapur et al. [123], est basé sur le principe de la maximisation de l'entropie totale de Shannon. Dans ce cas de la segmentation

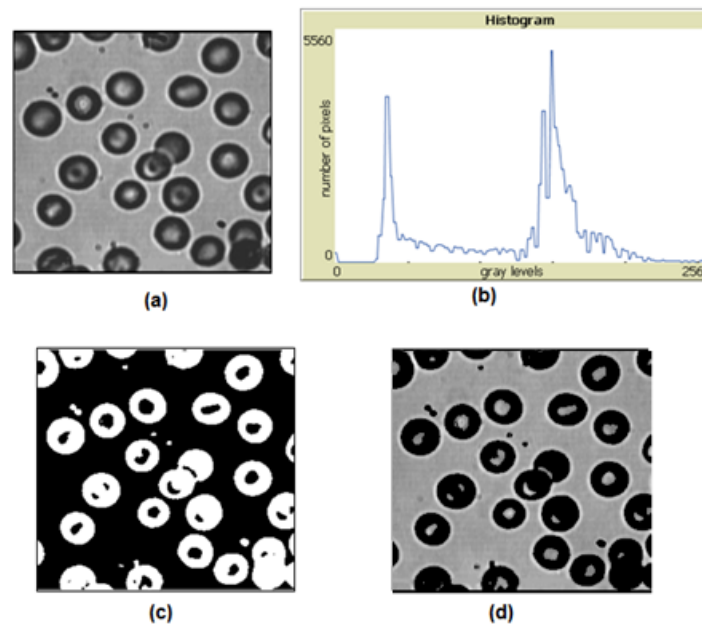


FIG. 4.5 – Exemple de seuillage à deux niveaux "manuel", avec un seuil = 103, (a) : Image macroscopique originale, (b) : Histogramme de l'image, (c) : Binarisation, (d) : Seuillage partiel.

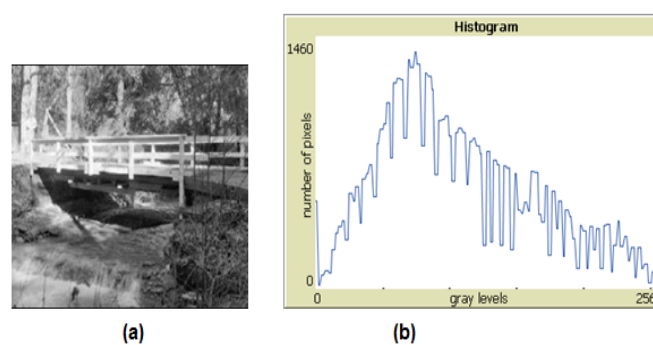


FIG. 4.6 – Histogramme d'une image complexe

de l'image en deux classes (C_0 et C_1), la méthode suppose que chaque classe possède une densité de probabilité indépendante. Le seuil optimal de segmentation est celui qui permet de maximiser l'entropie totale H_T de l'image partitionner.

$$H_T = H_0(t) + H_1(t) \quad (4.10)$$

Tels que :

$H_0(t)$ et $H_1(t)$ sont les entropies de classes C_0 et C_1 , qui sont définies par :

$$\begin{aligned} H_0(t) &= - \sum_{i=0}^{t-1} \frac{p_i}{w_0} \ln \frac{p_i}{w_0}, w_0 = \sum_{i=0}^{t-1} p_i \\ H_1(t) &= - \sum_{i=t}^{L-1} \frac{p_i}{w_1} \ln \frac{p_i}{w_1}, w_1 = \sum_{i=t}^{L-1} p_i \end{aligned}$$

w_0 et w_1 sont les densités de probabilités de classes C_0 et C_1 , et p_i est la probabilité du niveau de gris i .

$$p_i = \frac{h(i)}{\sum_{j=0}^{L-1} h(j)}$$

$$t^* = \text{Arg max}_{1 \leq t \leq L} \{H_T(t)\}$$

Le seuillage par la méthode d'Otsu est basé sur l'analyse discriminante. Dans son papier original [174], Otsu décrit trois critères discriminants possibles : la variance intra classe, la variance interclasse et la variance totale. Les trois critères sont équivalents et, suivant la situation, l'un d'eux peut être choisi. Du fait de la simplicité de la mise en œuvre de la variance interclasse, souvent les auteurs [166] la maximisent pour segmenter de l'image. Où la variance interclasse est définie par :

$$V_{inter} = w_0(t) w_1(t) (\mu_1(t) - \mu_0(t))^2 \quad (4.11)$$

et

$$\begin{aligned} w_0 &= \sum_{i=0}^{t-1} p_i \quad \text{et} \quad \mu_0 = \sum_{i=0}^{t-1} \frac{i \times p_i}{w_0} \\ w_1 &= \sum_{i=t}^{L-1} p_i \quad \text{et} \quad \mu_1 = \sum_{i=t}^{L-1} \frac{i \times p_i}{w_1} \\ p_i &= \frac{n_i}{\sum_{j=0}^{L-1} n_j} \quad \text{avec} \quad p_i \geq 0 \quad \text{et} \quad \sum_{i=0}^{L-1} p_i = 1 \end{aligned}$$

Tels que :

w_c est la densités de probabilités de classes C_c , $c = 0$ ou 1

μ_c est la moyenne de la classe C_c , $c = 0$ ou 1

p_i est la densité de probabilité du pixel i

n_i est le nombre d'occurrences du pixel de niveau de gris $i \in [0, L - 1]$ et L le nombre total de niveau de gris.

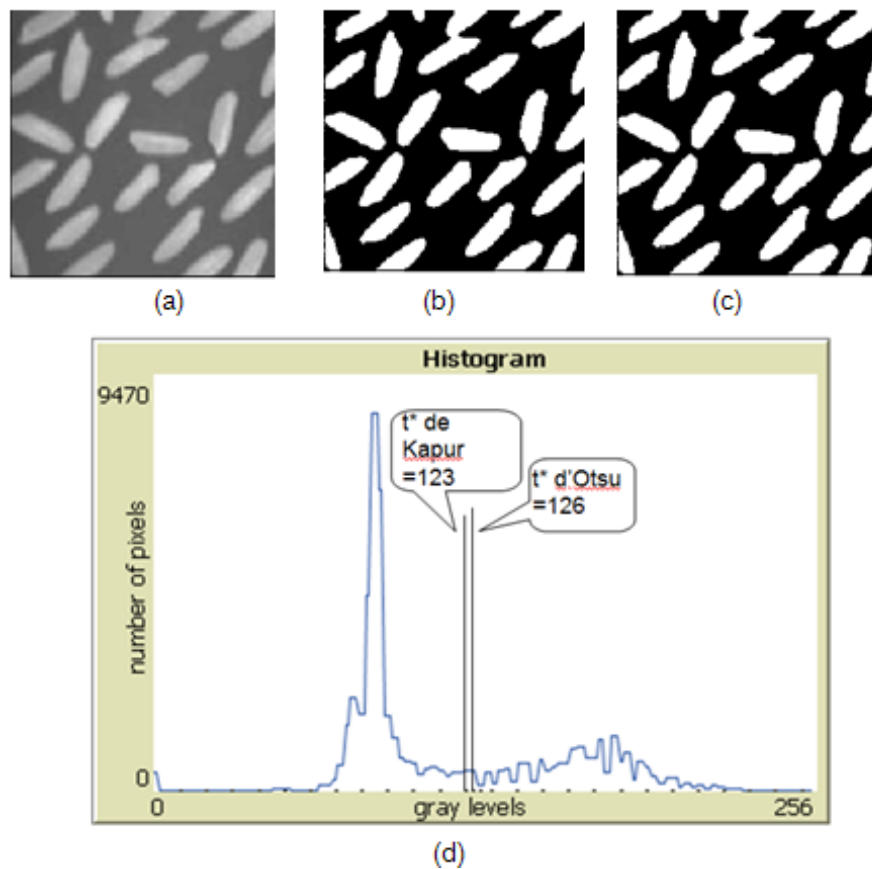


FIG. 4.7 – Exemple de seuillage à deux niveaux (a) : Image originale de grains de riz, (b) : image segmentée par Otsu, (c) : image segmentée par Kapur, (d) : Histogramme de l'image avec les valeurs de seuils trouvées par Otsu et Kapur.

Dans le cas d'un seuillage à deux niveaux le seuil optimal t^* est celui qui maximise la variance interclasse

$$t^* = \arg \max_{1 \leq t \leq L} \{V_{inter}(t)\}$$

La figure FIG. 4.7 montre un exemple d'application de la méthode d'Otsu et de la méthode Kapur d'une image de grains de riz..

Pour rendre ces techniques de seuillage plus pratique, Kim et al. [131] ont utilisé la transformée en ondelette pour réduire la résolution de l'histogramme. Avec l'utilisation d'un histogramme réduit, les seuils optimaux sont déterminés en optimisant la fonction de l'Otsu basé sur une recherche exhaustive. Les valeurs de seuils sélectionnées sont, donc, prolongées à l'échelle originale. D'autres auteurs utilisent une recherche itérative dont le principe est le suivant : on part avec un seuillage à deux niveaux, puis on utilise les résultats initiaux pour obtenir des seuils d'ordre plus élevé. Pour détailler cette procédure itérative, nous présentons l'algorithme de Yin [226].

Algorithme 4.2 : Recherche d'un ensemble optimal de seuils par une procédure itérative.

- **Entrée** : Un histogramme de niveaux de gris rangé dans l'intervalle $[0, L - 1]$, un paramètre C indique le nombre de seuils, une méthode de seuillage optimale à deux niveaux G (comme la méthode d'Otsu ou la méthode de Kapur) et sa fonction objectif associée f .
 - **Sortie** : les seuils sélectionnés.
1. Commencer par un ensemble de k seuils $[t_1, t_2, \dots, t_k]$ et ajouter deux seuils supplémentaires : $t_0 = 0$ and $t_{k+1} = L - 1$;
 2. Calculer la valeur de la fonction objectif f , en utilisant l'ensemble courant $[t_1, t_2, \dots, t_k]$;
 3. Mettre $i = 1$;
 4. Considérons la partie d'histogramme de niveau gris dans le sous-intervalle $[t_{i-1}, t_{i+1}]$, trouver le seuil optimal t^* entre t_{i-1} et t_{i+1} par G . Remplacer t_i avec t^* ;
 5. Mettre $i = i + 1$. Aller à l'étape 4 jusqu'à $i > k$;
 6. Calculer la valeur de la fonction objectif f en utilisant l'ensemble courant $[t_1, t_2, \dots, t_k]$;
 7. Si la fonction f est améliorée, allez à l'étape 3; sinon, arrêter.

Cette idée de recherche itérative a été utilisée dans plusieurs travaux, mais avec des particularités différentes, par exemple : Cheriet et al [39] ont appliqué la méthode d'Otsu de façon récursive, pour segmenter un seul type d'objet à la fois, dans l'image, à la fin, la segmentation complète se fait par regroupement de tous les objets segmentés. Lin [146] a proposé un calcul rapide de seuillage en utilisant la fonction d'Otsu. Son approche est basée sur la recherche des seuils où les dérivés partielles d'ordre un de la fonction d'Otsu prennent la valeur zéro avec une technique de substitution consécutive. Dans [145], la fonction d'Otsu est modifiée pour être optimisée par un algorithme récursif rapide en utilisant la technique "look-up table".

4.6.2 Métaheuristiques d'optimisation pour le seuillage à multi-niveaux.

Ces dernières années, plusieurs chercheurs tentent à utiliser des heuristiques comme des moyens alternatifs pour résoudre la segmentation par seuillage multiples. Plusieurs techniques utilisent les algorithmes génétiques (GA) ont été proposées pour résoudre le problème de seuillage à multi-niveaux : [227]-[120]-[224]-[38]-[138]-[208].

Yin [227] a proposé une méthode de seuillage basée sur un algorithme génétique où la fonction objectif est semblable à la fonction d'Otsu ou de Kapur. La solution est

codée comme une chaîne binaire T , tel que $T = t_1, t_2, \dots, t_{k-1}$ où le t_i est la valeur de l' $i^{\text{ème}}$ seuil qui possède $\log_2(L)$ bits et une valeur dans l'intervalle $[0, L - 1]$. Cette même technique a été utilisée dans [120]-[224].

Dans [120], la fonction objectif est semblable à la fonction de *Kapur* et dans [224] il est assimilé à l'entropie relative [182].

Chang et Yan [38] ont proposé une technique de seuillage en utilisant la Probabilité Conditionnelle de l'entropie (Conditional Probability Entropy (*CPE*)) basée sur la théorie Bayésiennes. Ils ont employé un *GA* pour maximiser le *CPE* afin de déterminer les seuils. Le *CPE* considère que les pixels avec le même niveau gris dans une image peuvent appartenir aux classes différentes avec des probabilités différentes. Une méthode de classification optimale pour ces pixels est de les mettre dans la classe avec plus grande probabilité. La structure du chromosome dans son algorithme génétique est basée sur la fonction de probabilité conditionnelle employée, avec deux paramètres pour chaque classe. Le chromosome est donc, constitué de $2k$ paramètres et chaque paramètre a $\log_2(L)$ bits. L désigne le nombre total de niveaux de gris présents dans l'image originale.

À côté des algorithmes génétiques, l'optimisation par essaim de particules (*PSO*) est une autre technique de l'optimisation qui a été utilisée pour le seuillage à plusieurs niveaux.

Zahara et al [229] ont proposé une approche appelé *NM-PSO*, basée sur l'algorithme de *PSO* [126] hybridé avec un algorithme de recherche locale Simplex [71] pour le seuillage à multi-niveaux. L'algorithme d'optimisation est utilisé pour déterminer les seuils qui minimisent la variance intraclasse d'*Otsu*, et aussi il est utilisé de pour déterminer les paramètres de gaussiennes permettant d'approcher au mieux l'histogramme de l'image pour la segmentation.

Yourui et Shuang [228] ont utilisé l'optimisation par essaim de particules quantiques "quantum particle swarm optimization (*QPSO*) pour maximiser la variance interclasse d'*Otsu*. Le *QPSO* codifie les positions courantes de particules en utilisant des bits quantiques "quantum bits", et la recherche s'effectue en utilisant la barrière de la rotation quantique "quantum rotation gate", et les mutations se réalisent par la non - barrière quantique "quantum non-gate" pour éviter la convergence prématurée.

Liou et al. [147] présentent un nouveau schéma de seuillage à multi-niveaux basé sur l'optimisation d'accouplement (de mariage) des abeilles à miel "the Honey Bee Mating Optimization (*HBMO*)" pour maximiser l'entropie de *Kapur*.

Liang et al. [144] ont utilisé l'optimisation par colonie de fourmis pour le seuillage à plusieurs niveaux.

On notera que ce genre d'algorithmes exige que le nombre de seuils, avec lesquels l'image devrait être segmentée, soit spécifié en avance. Connaître automatiquement le

nombre « optimum » de seuils dans l'image est un problème complexe et nécessite une connaissance à priori, sur les objets constituant l'image, qui n'est pas toujours disponible [141]-[103]. Afin de s'affranchir de cette contrainte, différents algorithmes de seuillage automatique ont été proposés dans la littérature.

4.6.3 Seuillage automatique

Yen et al. ont proposé un nouveau critère pour le seuillage automatique à multi-niveaux, appelé *ATC* (Automatic Thresholding Criterion) [225]. Ce critère est utilisé avec une technique dichotomique séquentielle [225]-[196]. Dans cette stratégie, l'histogramme est divisé en deux distributions, par l'utilisation d'un seuillage à deux niveaux, et la distribution avec la plus grande variance est divisée en suite en deux distributions en appliquant le même seuillage à deux niveaux. Ce processus dichotomique est répété jusqu'à une fonction coût (*ATC*) arrive à son minimum.

Dans [221], le même processus dichotomique est répété jusqu'à une fonction coût, dérivée de la fonction d'*Otsu* devient supérieure à une valeur spécifiée.

Les techniques dichotomiques sont des algorithmes plus rapides ; malheureusement, elles sont des techniques sous-optimales, ils ne permettent pas de fournir les valeurs optimales de seuils.

Hammouche et al. [105] propose une technique de seuillage, à plusieurs niveaux, rapide basée sur un *GA* pour déterminer de façon automatique le nombre approprié de seuils, aussi bien que les valeurs de seuils, par optimisation le critère *ATC* qui est proposé par Yen et al. La technique de transformée en ondelette est utilisée pour réduire le temps de calcul.

4.7 Systèmes multi-agents et segmentation d'images

Plusieurs systèmes multi-agents ont été proposés en général dans le domaine du traitement d'images et dans la segmentation en particulier. Ces systèmes introduisent des solutions intéressantes pour remédier à différents problèmes tels que la manipulation de connaissances hétérogènes, le contrôle de haut niveau sur les traitements de bas niveau, et la parallélisation et la distribution des traitements.

Selon la dichotomie établie entre l'approche cognitive et l'approche réactive au sein de la communauté de l'agent, les systèmes de traitement d'image basée sur l'agent se sont scindés en deux grandes catégories distinctes. La première est celle des systèmes multi-agents cognitifs, où les agents utilisés, dit dans ce cas cognitifs, sont en nombre restreint et ayant une forte granularité. Ces systèmes se basent sur la coordination des différentes tâches du processus de traitement d'image, où chacune de ces tâches

est prise en charge par un agent qui lui est dédié. L'affectation d'un agent à chacun des objets d'une image était également utilisée pour proposer des architectures de systèmes multi-agents cognitifs. La seconde catégorie, dite réactive, dont les travaux en traitement d'image sont rares et dont les systèmes proposés sont en stade de test et d'expérimentation, est basée sur l'utilisation d'une population dense d'agents réactifs avec une faible granularité. Ces systèmes utilisent l'image comme environnement dans lequel se déplacent les agents. Ces derniers utilisent et agissent sur l'image afin d'accomplir une tâche, souvent de bas niveau, du processus de traitement d'image.

Différentes propriétés et mécanismes de SMA ont été utilisés dans les travaux qui associent les SMA et le traitement d'image approches (coopération, négociation, adaptation, émergence, etc...).

La coopération permet la cohabitation de plusieurs algorithmes de traitement d'images. Elle augmente la qualité de la segmentation en affrontant les informations fournies par les algorithmes différents.

Spinu [204] s'intéresse particulièrement à la segmentation d'images par adaptation d'opérateurs. Ces opérateurs, regroupés au sein d'une bibliothèque, sont des opérateurs de détection de contours et de filtrage d'images. Les caractéristiques de l'image sont d'abord estimées (en termes de bruit, texture, contraste, etc.) et un chaînage d'opérateurs approprié est sélectionné. Les paramètres des opérateurs sont ajustés en fonction d'une évaluation de la qualité du résultat par des heuristique ou par optimisation). Cette méthode est implantée par un système multi-agents. Cette implantation permet une meilleure flexibilité et une parallélisation des traitements sur les régions de l'image. Les agents traitant des régions adjacentes peuvent coopérer entre eux.

Rodin et al. [189] ont proposé un système multi-agents formé d'agents réactifs pour la détection de contours dans des images biologiques en se basant sur le langage multi-agents oRis [107]. En fonction de certaines informations a priori sur le contenu des images, le système vise à produire une détection de contours meilleure que celle obtenue par les détecteurs traditionnels. Deux groupes d'agents dits respectivement agents de noircissement "darkening agents" et agents de luminance "lightening agents" suivent les régions sombres et les régions claires. Leurs actions visent à renforcer les régions par l'accroissement de leur contraste, permettant ainsi une détection fiable de ces régions. Le nombre et la topologie des régions attendues doivent être préalablement connus et codés dans les agents. Selon cette approche, les agents parcourant les différents contours sont totalement indépendants sans aucune d'interaction entre eux. Le système paraît comme un algorithme parallèle de segmentation qui était bien optimisé pour la détection de quelques types de contours. L'arrêt de ce système est réalisé à travers l'usage de mécanismes immunisés issus d'immunologie.

Richard et al. [188] ont proposé une architecture hiérarchique d'agents situés et

coopératifs pour la segmentation des IRM cérébrales. Trois types d'agents ont été utilisés : agent de contrôle global, agent de contrôle local, et agent dédié au tissu. Le rôle de l'agent de contrôle global est de partitionner le volume de données en territoires adjacents et d'affecter à chaque territoire un agent de contrôle local. Le rôle de ce dernier est de créer les agents dédiés au tissu (matière blanche, matière grise et liquide céphalorachidien) dont le rôle est d'effectuer un accroissement de région à l'intérieur du volume local. Les paramètres de distribution des données sont mis à jour par coopération entre les agents voisins. Par l'utilisation de différents types d'agents, le système proposé a pris en compte les traitements d'images de bas niveau, ainsi que le contrôle sur les tâches de segmentation. Cependant, il résulte des patterns complexes d'interaction dont la gestion est épineuse et coûteuse.

Ramos et Almeida [183] ont proposé un système multi-agents pour la détection des contours dans une image à niveau de gris. Les comportements des agents sont inspirés des fourmis Chialvo et Millonous [40]. L'image est l'environnement dans lequel les agents fourmis se déplacent. Les auteurs ont montré que certaines capacités perceptives (collectives) émergent des interactions locales ; entre les différents agents d'une part, et entre les agents et l'environnement d'une autre part.

Mazouzi et al. [152] ont proposé une approche non supervisée de segmentation d'images de profondeur, basée sur le paradigme agent. Elle consiste en l'utilisation d'une population d'agents réactifs qui se déplacent sur l'image et agissent sur les pixels situés sur les pourtours des régions. Après sa création, un agent se déplace aléatoirement dans l'image et cherche une région plane autour de sa position courante.

L'agent s'adapte à la région plane trouvée, puis lisse les pixels appartenant à cette région. La mémoire d'un agent représente une perception plus large qui, lorsqu'elle est combinée avec l'information locale de l'image, permet une décision plus fiable, en prenant en compte des données globales. Dans le but d'optimiser les déplacements des agents, un champ de potentiel inspiré du champ électrostatique est utilisé. Il permet aux agents de s'auto-organiser et de rationaliser leurs mouvements en se regroupant autour des régions d'intérêt (contours et bruit) et à concentrer leurs actions autour de ces régions. Les compétitions entre les agents qui s'auto-organisent autour des bordures des régions ont permis de faire émerger les lignes de contours dans l'image. Ces contours, dont aucune détection explicite n'est codée dans les agents, émergent des interactions et des actions collectives au sein de la population d'agents. Les agents utilisés sont faiblement couplés et communiquent indirectement via leur environnement (l'image).

Liu et al. [149] ont présenté un système basé sur des agents réactifs pour la segmentation d'IRM du cerveau. Quatre types d'agents sont utilisés pour étiqueter les pixels de l'image en fonction de leur degré d'appartenance aux différentes régions.

Les agents ont une perception très réduite de leur environnement et sont dotés de deux types de comportement : la diffusion et la reproduction de manière à s'adapter au mieux aux variations locales de l'image. Si un agent perçoit un stimulus dans son environnement, c'est à dire s'il est placé sur un pixel remplissant certains critères caractérisant l'appartenance à la classe homogène cherchée (pour laquelle il a été créé), il l'étiquette et déclenche un comportement de reproduction, de façon à poursuivre l'exploration de l'image par ses agents fils. S'il ne perçoit pas de stimulus, il déclenche un comportement de diffusion, au cours duquel il se déplace dans son environnement proche à la recherche d'un pixel satisfaisant ses critères.

Bourjot et ses collègues se sont inspirés du modèle de construction de toiles observé chez les araignées pour proposer un nouvel algorithme pour la segmentation d'images en régions homogènes [28]-[29]. Les agents correspondant aux araignées sont décrits par deux items comportementaux : le déplacement et la pose d'un fil et, en plus un comportement de retour sur la toile, qui a été introduit afin que l'araignée ne parcourt pas l'intégralité de l'image et ne tisse toutes les régions ayant le même niveau de gris. L'araignée est dotée d'un état interne pour permettre une pose contextuelle.

Les propriétés de ce modèle sont un nombre réduit d'items comportementaux, une perception limitée, un mécanisme de sélection d'action probabiliste, l'absence d'état interne et l'absence de représentation des autres agents. L'interaction s'effectue de manière indirecte via l'environnement et ce dernier permet de structurer l'activité des agents pour construire une toile respectant certaines propriétés.

Bellet [15] propose un système multi-agents spécialisés pour la segmentation d'images tel que certains agents exécutent une croissance de régions alors que d'autres exécutent un processus de détection de contours. La coopération entre les deux types d'agents est dynamique et permet de propager une information quand elle devient nécessaire pour une prise de décision.

Dans le même ordre d'idée Boucher [27] propose d'utiliser des agents spécialisés pour la segmentation et l'interprétation de séquences d'images cytologiques en mouvement. C'est une approche distribuée où chaque agent est spécialisé pour la reconnaissance d'un concept de l'image. Le modèle générique d'agents est composé de 4 comportements de base : perception croissance de région ou de suivi de contour), interaction fusion des primitives, différenciation interprétation des primitives, reproduction stratégie de focalisation des agents. Les agents du système sont asynchrones et concurrents Un agent ne travaille que sur une composante d'une image et peut accéder à toutes les informations du système soit de sa propre image ou celle de l'image précédente. L'utilisateur interagit avec le système par le biais de séquenceur qui gère l'exécution des différents agents du système.

4.8 Conclusion

Dans ce chapitre, nous avons fait un tour d'horizon sur les méthodes de segmentation d'images, un intérêt particulier est porté sur les méthodes de segmentation par seuillage et par croissance de régions car c'est le domaine d'application de cette thèse.

Ce tour d'horizon couvre les méthodes classiques de segmentation d'images ainsi que les heuristiques bio-inspirées. Cette présentation n'est pas exhaustive mais permet de présenter les principes de chacune des techniques de segmentation.

Troisième partie

**Contributions à la Segmentation
d'images**

Chapitre 5

Segmentation d'images par la croissance auto-organisée de régions

5.1 Introduction

Dans ce chapitre, nous présentons notre première contribution dans le domaine de la segmentation d'images, cette contribution consiste en la combinaison d'un sous-système d'intelligence en essaim avec un sous-système de segmentation d'images par la méthode de croissance de région. Cette dernière, comme nous l'avons vu dans la section 4.3, souffre de quelques problèmes tels que :

- elle nécessite le réglage de plusieurs seuils qui sont difficiles à fixer en l'absence de connaissance a priori ; par exemple, le choix du seuil de similarité : un seuil trop faible crée une sur-segmentation et s'il est trop fort conduit à une sous-segmentation ;
- il est encore difficile de cerner l'homogénéité au sein de l'image, par exemple en définissant globalement un simple seuil d'homogénéité, du fait que cette homogénéité peut être à des degrés variables pour différentes parties de l'image ;
- son résultat est fortement dépendant du choix des germes de départ et de l'ordre de parcours des pixels agrégés.

L'intelligence en essaim offre un cadre naturel permettant de résoudre ces problèmes d'une manière distribuée, flexible et adaptative, en se basant d'une part, sur les mécanismes d'auto-organisation et de stigmergie observés dans les processus de

construction de nid chez les termites, et d'autre part sur l'optimisation par essaim de particules "*PSO*".

Notre contribution, dans ce domaine, a été couronnée par les publications [55]-[56]-[57]-[58]-[59]-[60]-[61].

L'approche SORG "Self-Organizing Region Growing" qui sera décrite, dans ce chapitre, est une synthèse de ces publications, elle consiste à favoriser une croissance auto-organisée de régions homogènes en adoptant le paradigme multi-agents. Une population d'agents autonomes "*Constructeur*" explorent l'image, pour déterminer les pixels germes et estimer l'homogénéité au sein de l'image, en utilisant une variante de l'optimisation par essaim de particules (*PSO*). Les agents s'adaptent aux régions sur lesquelles se déplacent, puis effectuent des actions coopératives produisant une segmentation collective de l'image. Un champ de potentiel artificiel est introduit afin de coordonner les déplacements et les actions des agents, et aussi de régler la croissance des régions homogènes. Quelques expériences sont faites sur des images synthétiques et réelles pour montrer les caractéristiques de cette approche et présenter les résultats obtenus.

5.2 Spécification de notre approche

Notre approche s'inscrit dans le cadre de la mouvance de développement des systèmes de segmentation d'images, permettant d'avoir une meilleure qualité de la segmentation sur une grande variété d'images. Nous adoptons le paradigme d'agent réactif qui utilisent l'image comme un environnement dans lequel ils se situent et se déplacent.

Notre approche se matérialise par la définition de deux populations d'agents, nous distinguons la population d'agents *Constructeurs*, liés au marquage de régions homogènes, à la population d'agents *Environnement* permettant la gestion de l'environnement.

Les agents *Constructeurs* sont des entités autonomes, qui possèdent des mémoires simples, des capacités de calcul limitées et des perceptions locales de leur environnement. Ils mettent en œuvre des comportements simples en fonction des contraintes de l'environnement, ils peuvent se communiquer directement ou indirectement via l'environnement partagé.

Les agents *Constructeurs* se déplacent sur l'image et marquent leurs emplacements. Au début, les agents *Constructeurs* commencent à explorer l'image, sans déposer de marque (d'étiquette) sur leurs emplacements, pour détecter les pixels germes et aussi d'estimer, de façon locale, les valeurs de départ de seuils d'homogénéité aux différentes parties de l'image, à l'aide d'une variante du *PSO* [129].

Le pixel *germe* est le premier pixel, détecté et marqué, d'une région homogène

"région marquée", il est considéré comme une ébauche de construction de la région marquée, qui est en cours de construction.

Chaque région marquée produit un champ de potentiel artificiel phéromonal, qui se propage dans un environnement local "région de propagation" correspondant aux pixels qui entourent la région marquée.

Après l'apparition de quelques régions marquées (les ébauches de construction) dans l'image, les déplacements des agents *Constructeurs* deviennent coordonnés; les agents seront guidés par leur travail qui est les régions marquées (interaction stigmergique);

La dynamique de phéromone est caractérisée par les opérations : de propagation, d'évaporation, de concentration et de construction des champs de gradient de phéromone, et elle est gérée par les agents *Environnement*.

Un ensemble de paramètres a été utilisé pour décrire cette approche. Cet ensemble est divisé en deux sous-ensembles de paramètres :

1. Paramètres décrivant les agents *Constructeurs*

- N : nombre d'agents ;
- Ca : capacité d'agir d'un agent ;
- Cm : couleur de marquage d'un agent ;
- Dv : durée de vie d'un agent ;
- Ld : longueur d'adaptation d'un agent (le degré de confiance d'un agent) ;
- $vA(p)$: l'environnement local d'un agent situé sur le pixel p ;
- Np : nombre de pixels non marqués dans la région locale $vA(p)$ de l'agent. Paramètres décrivant la dynamique de phéromone

2. Paramètres décrivant la dynamique de phéromone

- Rm : région marquée (région homogène) dans l'image ;
- Rp : région de propagation du champ de phéromone produit par la région Rm ;
- δl : seuil de construction « seuil d'attraction de phéromone » au-dessous duquel l'attraction de la phéromone n'a pas un effet sur les comportement des agents ;
- s_i : concentration de phéromone dans le pixel p_i ;
- f_i : attraction relative au pixel p_i .

3. Paramètres décrivant l'environnement

- A : image à segmenter ;
- $U \times V$: dimension de l'image A .

5.3 Environnement

L'environnement est un composant essentiel de notre système, il sert non seulement au codage des données du problème "image à segmenter", mais aussi aux résultats intermédiaires permettant d'agir directement ou indirectement sur les comportements des agents. Il est représenté par un tableau de deux dimensions A de taille $U \times V$, dont les éléments sont les pixels de l'image. Chaque pixel p est indexé par ses coordonnées (i et j) dans A , il est caractérisé par son niveau de gris $A(i, j)$, son marquage, et par une liste de concentrations de phéromone. Initialement, l'environnement ne contient aucune étiquette (marque) et aucune concentration de phéromone. Les étiquettes seront ajoutées par les agents durant leur exécution, et les concentrations de phéromone seront formées après l'émergence de régions germes dans l'environnement.

5.4 Comportements des agents

Dans notre modèle, les agents *Constructeurs* ont une granularité faible, et leur nombre doit être suffisamment grand. A chaque instant t , un agent est caractérisé par :

- sa position (i, j) dans l'image ;
- sa capacité d'agir Ca sur les pixels de sa région locale ;
- sa couleur de marquage Cm ;
- sa durée de vie Dv ;
- sa longueur d'adaptation Ld ;

Au début, tous agents sont initialisés dans des emplacements aléatoires sur l'image à segmenter avec : une capacité d'agir Ca égale à faux, une longueur d'adaptation Ld égale à zéro, la même durée de vie, et des couleurs de marquage générées de façon aléatoire. Tous les agents sont incapables de marquer leurs emplacements de l'image. Un agent devient capable de marquer son emplacement ($Ca = vrai$), lorsqu'il détecte un pixel germe.

Les agents ont une perception limitée de leur environnement ; l'agent ne peut percevoir que les pixels voisins (8—connexes) de son emplacement courant, correspondant à son environnement local (voir figure FIG. 5.1), et il agit sur la base de son état et sa perception.

Selon l'état de son environnement local qui peut être : *PasDePheromone*, *TrouverPheromone*, *PhéromonesSuperposées* et *SituationDeBlocage*, l'agent peut suivre, respectivement un des comportements suivants : *Exploration*, *Agrégation*, *Fusion* et *Migration* (voir Figure FIG. 5.2).

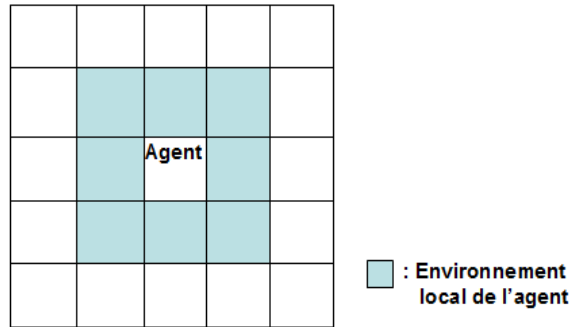


FIG. 5.1 – Région de perception de l'agent

5.4.1 Comportement Exploration

En général, l'agent suit le comportement *Exploration*, après son initialisation ($Ca = faux$) et aussi, quand l'état de son environnement local est "*PasDePheromone*", c'est-à-dire, si tous les pixels, de sa région locale, ne sont pas couverts par aucun champ de phéromone. Dans ce cas, l'agent commence à explorer l'image en cherchant un germe (un pixel de départ). Un germe cherché est un pixel $p(i, j)$ qui n'appartient à aucune région marquée (en cours de construction), à aucune région de propagation de phéromone et ayant une valeur de gradient la plus minimale $bestG$. Un germe est l'ébauche de construction d'une région homogène marquée Rm , dont le seuil d'homogénéité local (seuil de construction) est δ_l , cette valeur peut être changée dynamiquement pendant la croissance de la région marquée. La valeur initiale de δ_l est donnée comme suit :

$$\delta_l = \frac{bestG}{255} \text{ et } \delta_l \in [0, 1] \quad (5.1)$$

Pour optimiser la recherche des pixels germes, et ajuster les seuils d'homogénéité locaux, par la population des agents *Constructeurs*, nous adoptons une variante du

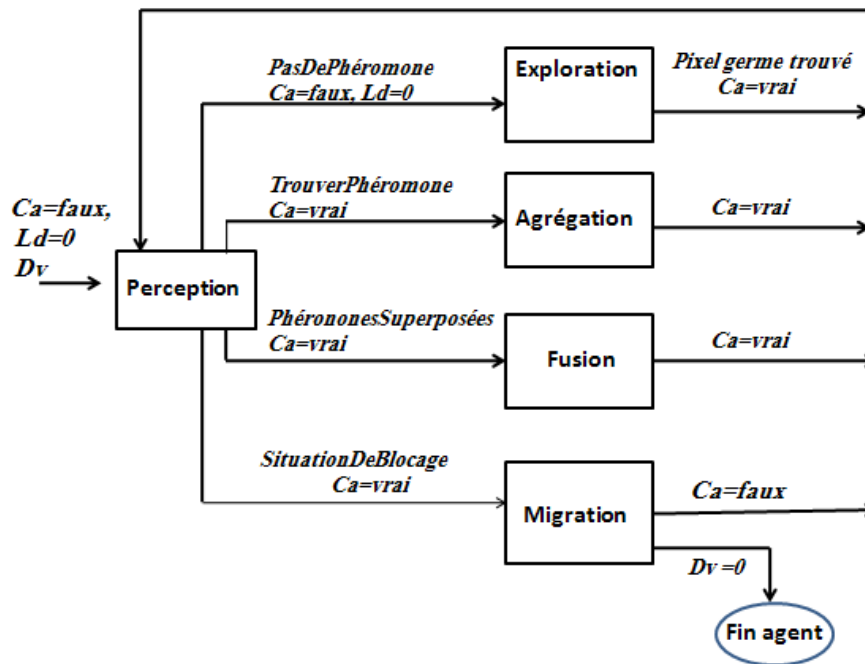


FIG. 5.2 – Le comportement d'un agent en fonction de son état et de sa perception.

PSO. Dans ce contexte, pour chaque nouvel emplacement, l'agent prend une estimation sur son emplacement actuel (i, j) en calculant la valeur de son gradient $G(i, j)$. Il se souvient de la meilleure estimation $bestG$ et $(besti, bestj)$ (la valeur du gradient la plus minimale $bestG$ et son emplacement correspondant $(besti, bestj)$) par lequel il est passé au cours de ses déplacements. Lorsqu'il rencontre un agent voisin (c'est-à-dire ces deux agents partagent au moins un pixel voisin (8-connexes) de leurs emplacements), ils s'engagent dans des interactions de confrontation. Dans ces interactions, les agents échangent et modifient ensuite leurs estimations, pour garder la meilleure. Après la dernière modification de sa valeur $bestG$, en faisant Ld déplacements, l'agent revient à l'emplacement $(besti, bestj)$ correspondant à la meilleure valeur d'estimation local $bestG$, et devient capable de marquer son emplacement ($Ca = \text{vrai}$). Ld est appelée la longueur d'adaptation, et représente le degré de confiance que l'agent soit à l'intérieur d'une région homogène.

5.4.2 Comportement Agrégation

L'agent peut suivre le comportement *Agrégation*, si sa capacité d'agir $Ca = \text{vrai}$ et si un des pixels de son environnement local est couvert par un des champs de phéromone (c'est-à-dire l'état de son environnement local est "*TrouverPhéromone*").

Dans ce cas, l'agent sélectionne son emplacement suivant, d'une manière probabiliste, en fonction de l'attraction relative f_i (Eq. 5.2), puis il le marque par sa propre couleur. Dans ce contexte, l'agent exploite le travail effectué par lui-même et par les autres agents.

$$f_i = \frac{S_i}{\sum_{p_j \in v_A(P)} S_j} \quad (5.2)$$

Avec

f_i : est l'attraction relative du pixel p_i ,

s_i : la concentration de phéromone dans le pixel p_i ,

$v_A(P)$: l'environnement local de l'agent situé sur le pixel P .

5.4.3 Comportement Fusion

L'agent peut suivre le comportement *Fusion*, si sa capacité d'agir $Ca = vrai$ et si un des pixels de son environnement local est couvert par, au moins, deux champs de phéromone (c'est-à-dire l'état de son environnement local est *PhéromonesSuperposées*). Dans ce cas, l'agent adapte sa couleur de marquage à la couleur de la région marquée produisant le champ de phéromone qui l'attire, puis il fait un tour en arrière en suivant la trace de son ancien marquage, et il marque son emplacement par la nouvelle couleur de marquage. Dans ce contexte, le marquage d'une région homogène se fait par une coopération d'agents.

5.4.4 Comportement Migration

L'agent suit le comportement *Migration* si sa capacité d'agir $Ca = vrai$ et si tous les pixels, de son environnement local, sont marqués et ne sont pas couverts par aucun champ de phéromone (c'est-à-dire l'état de l'environnement local est *SituationDeBlocage*). Dans ce cas, l'agent perd sa capacité de marquage ($Ca = false$) et sa confiance qu'il soit à l'intérieur d'une région homogène ($Ld = 0$), puis il recommence à chercher un nouveau pixel germe ou une nouvelle région de propagation de phéromone. Il choisit une direction de façon aléatoire, et se déplace vers un nouvel emplacement, avec un pas P . Selon le nombre de pixels voisins (8-connexes) non marqués " Np ", de son nouvel emplacement, la durée de sa vie Dv sera modifiée (Eq. 5.3). Si la durée de vie de l'agent Dv égale à 0, l'agent mourra.

$$Dv = Dv - 1 + Np \quad (5.3)$$

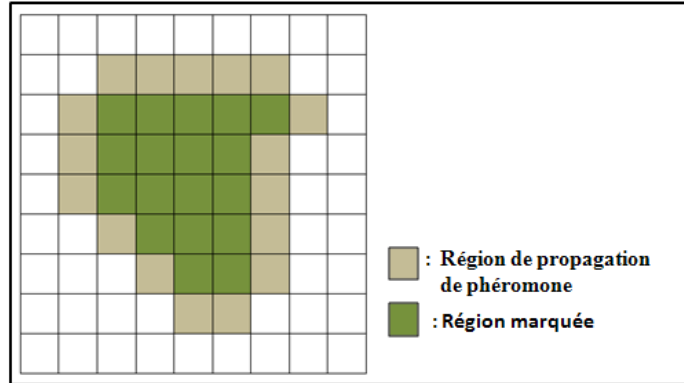


FIG. 5.3 – Représentation de la région marquée et de la région de propagation de phéromone correspondante

5.5 Coordination d'agents par un champ de phéromone

Afin de doter les agents d'un mécanisme d'auto-organisation spatiale, un champ potentiel phéromonal est créé et mis à jour autour de chaque région marquée. Il permet aux agents de se regrouper autour des régions d'intérêt dans l'image et de concentrer leurs actions sur les pixels qui entourent ces régions. Contrairement, à d'autres travaux où le champ de potentiel est créé à des positions prédéfinies correspondant aux objets de l'environnement (buts, obstacles...) [216]-[201]-[81], dans notre cas, le champ de potentiel résulte de l'interaction des agents avec les objets de l'environnement (pixels de l'image).

Les pixels marqués par une même couleur représente une région marquée, et l'ensemble de pixels situés aux voisinages (4-connexes) de la région marquée représente la région de propagation de phéromone produite par cette région marquée (FIG. 5.3).

La valeur de la concentration de phéromone $s(p)$ au pixel p , de la région de propagation de phéromone Rp_i , est le rapport entre sa valeur d'adéquation $Vad(p, Rm_i)$, par rapport à la région marquée Rm_i , et la valeur de son gradient $G(p)$ (Eq. 5.4).

$$S(p) = \frac{Vad(p, Rm_i)}{G_p} \quad (5.4)$$

L'évaluation de l'adéquation d'un pixel, par rapport à une région, est une combinaison linéaire pondérée de plusieurs critères d'évaluation (Eq. 5.5). Tous les critères d'évaluation fournissent des valeurs comprises entre 0 et 1. Chaque critère possède

un poids dans l'évaluation de l'adéquation du pixel, qui est la somme pondérée des différentes évaluations.

$$Adequation = \sum_{i=0}^k poids_i \times critère_i \quad (5.5)$$

Où $critère_i$ représente l'évaluation du pixel selon le critère i et le $poids_i$ est son poids dans l'évaluation de l'adéquation.

Différents critères d'évaluations peuvent être employés [15]-[27]-[193] : les critères qui caractérisent les propriétés photométriques, et les critères concernant les propriétés géométriques de la région.

Dans ce contexte, nous utilisons deux critères : un photométrique caractérisant l'homogénéité de l'image marquée, et l'autre géométrique caractérisant la compacité de la région marquée. Le critère d'homogénéité a pour effet de favoriser une croissance homogène de la région marquée, en mesurant la différence entre la variance du pixel p et la moyenne de variance des pixels de la région marquée Rm_i . Le critère de compacité a pour but de favoriser une croissance compacte de la région marquée, en mesurant le nombre de voisins (8-connexes), du pixel p , appartenant déjà à la région marquée Rm_i .

L'attractivité de la phéromone est prise en compte, seulement si, la concentration de la phéromone $s(p)$ dépasse le seuil de construction δ_l ($s(p) > \delta_l$). Ce seuil est adapté, selon les paramètres intrinsèques de la région (la moyenne de variance des pixels de la région marqué Rm_i). Il est recalculé à chaque fois que les informations nécessaires évoluent (Eq. 5.6), ce qui permet d'affiner le processus de construction au fur et à mesure que les agents progressent dans leur travail.

$$\delta_l = fact \times moyenne(variance) \quad (5.6)$$

Tels que :

δ_l : seuil de construction ;

$fact$: est un facteur $\in [0, 1]$;

$moyenne(variance)$: la moyenne de variance des pixels de la région marqué Rm .

5.6 Résultats expérimentaux

A fin de valider l'approche proposée, et de présenter ses caractéristiques, nous l'avons testée sur des images réelles et de synthèse. Les images réelles utilisées sont : une *IRM*, une image microscopique, image satellitaire de Lac Tahoe (FIG. 5.4).

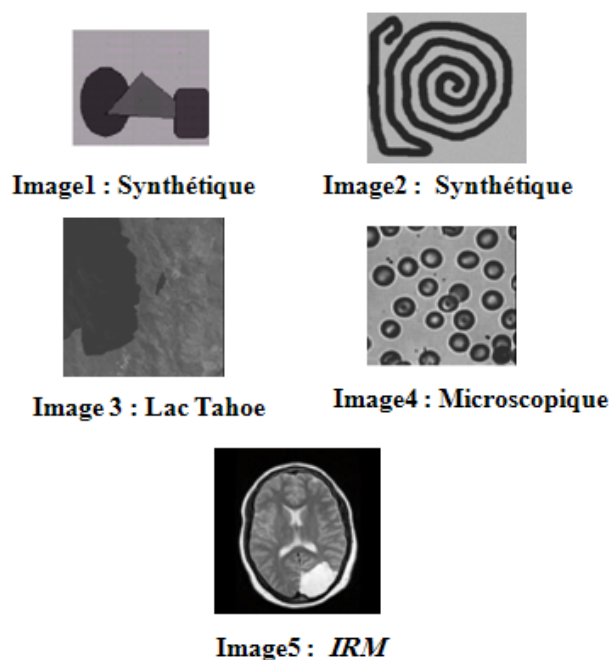


FIG. 5.4 – Images de test

5.6.1 Mesures d'évaluation

Les résultats obtenus, à partir d'un algorithme de segmentation, peuvent être évalués quantitativement selon deux approches [125]-[173] : la première approche se réfère à l'image originale, soit par une quantification a priori de l'erreur de mesure (dans le cas d'une classification), soit par des mesures d'homogénéité des régions ou de contraste entre régions (dans le cas d'une segmentation). La deuxième approche consiste à comparer le résultat fourni par l'algorithme avec une segmentation de référence. La segmentation de référence est produite par un opérateur humain. La difficulté inhérente à cette approche est, d'une part de trouver un échantillon d'images test représentatif de la classe d'images sur laquelle l'algorithme est sensé opérer, et d'autre part que la notion de bonne segmentation présente un sens identique pour des observateurs différents.

Pour évaluer la justesse de notre méthode, nous choisissons la première approche ; nous utilisons la mesure de Borsotti et al [26] dont l'objectif à pénaliser les résultats de segmentation présentant beaucoup de petites régions ainsi que des régions non homogènes et pour la rendre plus sensible aux petites variations de segmentation.

Cette mesure est définie comme suit :

$$Borsotti(Iseg) = \frac{1}{10000(card(Iseg))} \sqrt{NR} \times \sum_{i=1}^{NR} \left[\frac{e_i^2}{1+\log(Card(R_i))} + \left(\frac{f(Card(R_i))}{Card(R_i)} \right)^2 \right] \quad (5.7)$$

Où

- $f(Card(R_i))$ correspond au nombre de régions ayant une aire égale à $Card(R_i)$
- $Iseg$ l'image segmentée en un ensemble R de NR régions $\{R_1, R_2 \dots R_{NR}\}$,
- $Card(Iseg)$ correspond au nombre de pixels de l'image $Iseg$
- e_i correspond à l'erreur portant sur le niveau de gris moyen de la région R_i , définit par la somme des distances euclidiennes entre les niveau de gris des pixels de la région R_i et le niveau de gris moyen de la région R_i dans l'image segmentée.

Le corps de la somme est composé de deux termes : le premier pénalise les régions non homogènes (typiquement les grandes), alors que le second terme pénalise les régions dont l'aire $Card(R_i)$ est égale à l'aire de beaucoup d'autres régions dans l'image segmentée (typiquement les petites). Plus la valeur du critère est petite, meilleure est la segmentation évaluée.

5.6.2 Sélection de paramètres

Pour la méthode SORG, deux paramètres doivent être fixés : le nombre d'agents N , la valeur de la durée de vie d'agent Dv . Ces paramètres influencent la dynamique des agents. Le nombre d'agents doit être grand et dépend de la taille de l'image à segmenter $U \times V$.

Pour fixer ces deux paramètres, 100 combinaisons de ces paramètres, soit $(N, Dv) \in \{\sqrt{U \times V}, \sqrt{2 \times U \times V}, \sqrt{4 \times U \times V}, \sqrt{6 \times U \times V}, \sqrt{8 \times U \times V}\} \times \{5, 7, 9, 12, 14\}$ ont été exécutées sur le jeu d'images de test utilisé. Dans ce cas, le critère de performance choisi est la valeur moyenne de la mesure de Borsotti et al . Le calcul a aboutit aux valeurs de N égale à la valeur entière de $\sqrt{4 \times U \times V}$ et $Dv = 12$.

5.6.3 Résultats

Durant l'expérimentation, nous focalisons sur certains points, qui semblent constituer la force de l'approche proposée par rapport aux méthodes habituelles, tels que : la construction progressive et visuelle des régions homogènes, dans l'image, l'auto-organisation et le contrôle du processus de construction, la robustesse et la comparaison des résultats.

Construction progressive et visuelle des régions homogènes

Dans cette expérimentation, nous avons pris des poses (FIG.5.5) de l'évolution du processus de construction des régions homogènes, d'une *IRM* (FIG. 5.5(a)) , dans le temps.

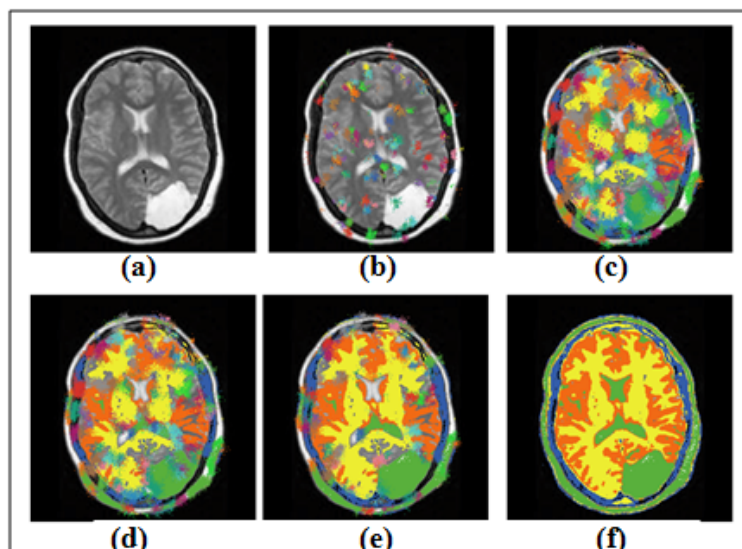


FIG. 5.5 – L'évolution du processus de construction des régions homogènes dans le temps.

Ce processus de segmentation est une succession d'étapes de la construction des régions homogènes dans l'image : l'émergence des ébauches de construction (régions marquées) dans l'image (FIG. 5.5(b)), la croissance de tailles et de nombres de régions marquées (FIG. 5.5(c)), la fusion des régions homogènes et connexes en une seule région marquée par une des couleurs de régions concernées (5.5(d) et FIG. 5.5(e)). L'image résultat est présentée dans la figure FIG. 5.5(f).

L'auto-organisation et le contrôle du processus de construction

L'auto-organisation caractérise le processus au cours duquel des structures spatiales (régions marquées) émergent, au niveau collectif, à partir de la multitude d'interactions entre agents.

Les interactions entre agents sont mises en oeuvre par le biais de la communication indirecte "stigmergique". Elles sont régies par un mécanisme auto-catalytique élaboré à partir d'un processus de rétroaction. La rétroaction positive permet d'amplifier le

marquage autour de la région marquée, elle devient négative lorsque la région marquée s'approche de sa frontière.

Afin d'exposer l'effet des phénomènes de rétroaction positive et rétroaction négative sur l'autorégulation du processus de croissance de régions homogènes dans l'image, un test a été mené sur une image de synthèse (FIG. 5.6(a)).

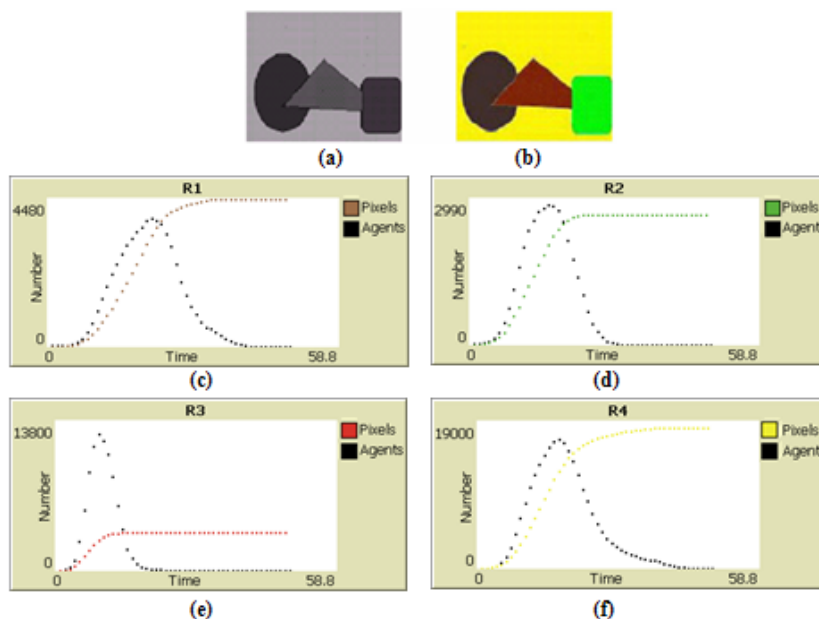


FIG. 5.6 – Segmentation d'une image synthétique et influence de la croissance de taille de chaque région, en cours de construction, sur les comportements des agents.

Dans cette expérimentation, une image synthétique FIG. 5.6(a) est segmentée par un nombre important d'agents qui sont initialisés dans des emplacements aléatoires sur l'image.

Un agent, qui a découvert un pixel germe, dépose sur ce pixel une étiquette (une marque). Le pixel germe marqué constitue une ébauche de concentration d'une région marquée. Cette dernière produit un champ de phéromone qui oriente les déplacements et les comportements des agents pour marquer les pixels qui l'entourent.

Les nombres d'agents réagissant aux champs de phéromone produits par les régions de couleur "marron", "verte", "rouge" et "jaune" (FIG.5.6(b)) sont respectivement présentés par des courbes de couleur noire dans les figures : FIG. 5.6(c), FIG. 5.6(d), FIG. 5.6(e) et FIG. 5.6(f). Le nombre d'agents est dynamique; sa modification, dans le temps, dépend de la taille de la région marquée qui produit le champ du gradient qui a son tour, une influence sur les comportements des agents. La taille de chaque région marquée (le nombre de pixels qui la constituent) est aussi présentée par une

courbe ayant la même couleur du marquage de cette région (FIG. 5.6(c), FIG. 5.6(d), FIG. 5.6(e) et FIG. 5.6(f)).

D'après ces graphes (FIG. 5.6(c), FIG. 5.6(d), FIG. 5.6(e) et FIG. 5.6(f)), on constate deux phénomènes :

1. Une croissance du nombre de pixels est accompagnée par une croissance du nombre d'agents ;
2. Une stabilisation du nombre de pixels est accompagnée par une décroissance du nombre d'agents.

Ces deux phénomènes sont interprétés comme suit :

1. Au fur et à mesure que le nombre d'agents, réagissant au champ de gradient produit par une région marquée (qui est en cours de construction), augmente, la taille de cette région augmente aussi ; l'accroissement du taux de marquage de pixels, dont la concentration de phéromone est élevée, augmente la taille du champ de phéromone (région de propagation de la phéromone) qui attire les agents, ce qui en retour, accroît le taux de marquage et ainsi de suite. Cette réaction autocatalytique (rétroaction positive) a pour résultat une amplification rapide de marquage des pixels de plus haute concentration en phéromone et l'émergence de la croissance de région.
2. Lorsque la région marquée (qui est en cours de construction) s'approche de sa frontière, sa taille se stabilise, dans ce cas, le champ de phéromone produit par cette région n'aura pas d'influence sur les comportements d'agents, car l'attractivité de ce champ devient faible. Dans ce cas, les agents orientent leurs mouvements vers d'autres champs de phéromone (produits par d'autres régions marquées) dont l'attractivité est plus forte, jusqu'à la fin de leurs durées de vie. On assiste, donc, à la mise en place, d'un phénomène de rétroaction négative.

Robustesse par rapport aux données à segmenter

Dans cette section, on peut montrer que notre système est capable de segmenter différents types d'images : synthétiques, satellitaires, médicales (voir figure FIG.5.7), cela revient à :

- La redondance et la non spécialisation des agents assurent une grande tolérance aux dysfonctionnements individuels ; tous les agents possèdent les mêmes items comportementaux, mais ils réagissent de façon différente, selon leurs états et leurs perceptions, ce qui permet que le travail entamé par un agent peut être achevé par un autre agent, par exemple, dans le cas de sa mort ou lorsqu'il se trouve dans une situation de blocage.

- L'agent se déplace sur l'image soit : avec un pas = 1, pendant le comportement *Agrégation* soit avec un pas = P , pendant le comportement *Migration*, ce qui permet d'une part, de répartir la recherche sur l'ensemble des régions avoisinantes, et d'autre part, de minimiser les mouvements de recherches infructueux, aidant ainsi à converger rapidement vers une segmentation complète de l'image.
- La durée de vie d'un agent ; tous les agents sont initialisés avec une même valeur de durée de vie Dv , mais cette valeur peut être modifiée au cours de l'exécution, selon l'état de son environnement local, ce qui donne à l'agent une grande adaptation à l'évolution de processus de construction. La figure FIG.5.8 présente l'évolution de durée de vie moyenne d'un agent.

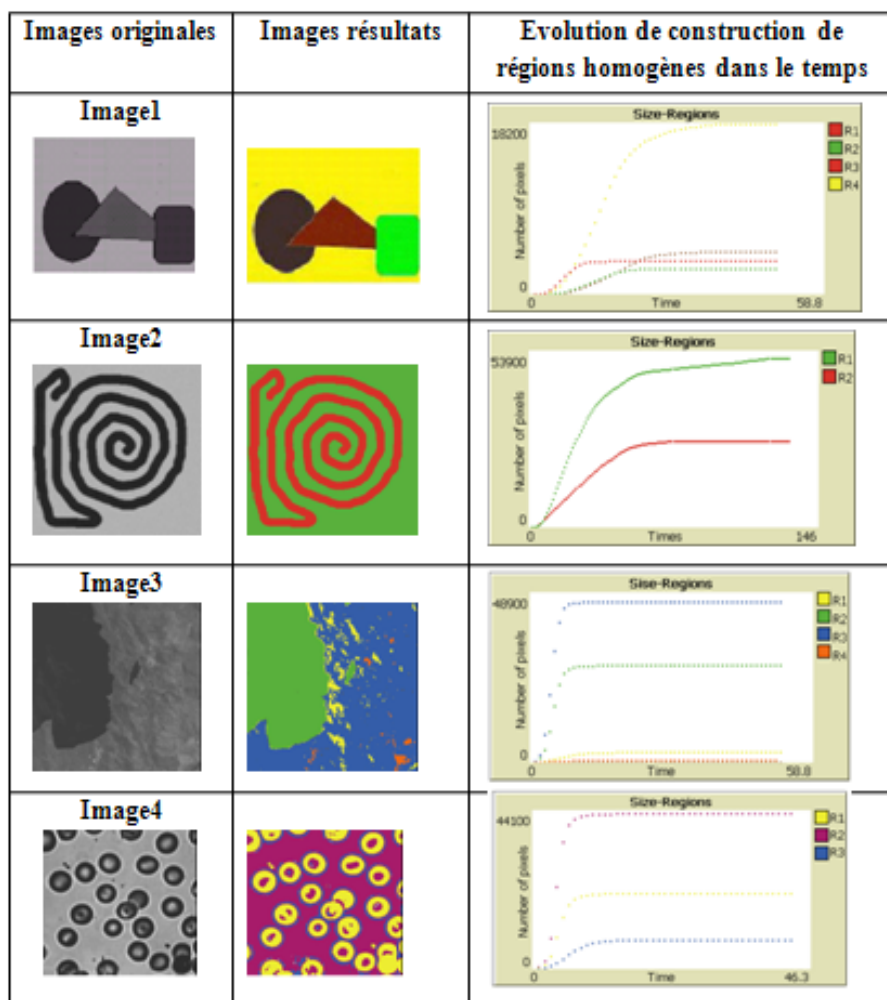


FIG. 5.7 – Exemple d'images segmentées par notre approche.

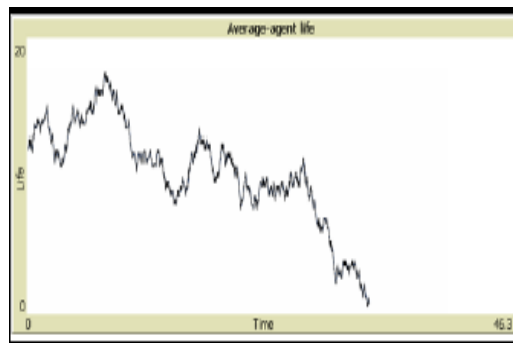


FIG. 5.8 – Evolution de la durée de vie moyenne d'un agent pendant le processus de segmentation de l'image de synthèse "Image1".

Comparaison des résultats

La qualité de segmentation obtenue par méthode *SORG* est comparée avec celle de la méthode de croissance de régions "*RG*", en utilisant la mesure de *Borsotti et al* [26] (TAB. 5.1).

Pour la méthode de croissance de régions, la sélection de pixels *germes* a été effectuée manuellement, et l'évaluation d'un pixel pour l'agrégation est choisi comme étant la différence entre la variance du pixel et la moyenne de variance des pixels de *Primitive* (la région qui est en cours de construction); cette évaluation doit être inférieure à un seuil globale d'agrégation qui égale à 0.7.

Algorithme 5.1 : Algorithme de croissance de régions

1. Choisir un ensemble de pixels *germes* ;
2. Mettre *Primitive* à l'ensemble vide ;
3. Mettre *ListePixelsCandidats* à l'ensemble vide ;
4. Mettre *ListeAgrégation* à l'ensemble vide ;
5. Pour chaque pixel *germe* Faire
 - Ajouter le pixel *germe* à *Primitive* ;
 - Ajouter les pixels voisins (8-connexes) du pixel *germe* à *ListeAgrégation* ;
 - Elire le pixel le plus apte pour l'agrégation (l'évaluation est supérieure au seuil d'agrégation) de *ListeAgrégation* ;
 - Ajouter le pixel à *Primitive* ;
 - Mettre à jour les informations sur *Primitive* ;

Images	Borsotti(Iseg) <i>SORG</i>	Borsotti(Iseg) <i>RG</i>
Image1	0.0086	2.8639
Image2	0.0043	1.5146
Image3	0.1409	6.7438
Image4	0.0426	4.5251
Image5	0.0751	7.0634

TAB. 5.1 – Qualité de la segmentation des images de test par les méthodes *SORG* et *RG*.

- Mettre à jour *ListePixelsCandidats* ;
- Evaluer les nouveaux pixels candidats ;
- Ajouter ces pixels à *ListePixelsCandidats* ;
- Itérer jusqu'à ce qu'il n'y ait plus de pixels aptes à l'agrégation.

D'après ces résultats (TAB. 5.1), nous constatons que la qualité des résultats obtenus par *SORG* est meilleur que celle obtenus de *RG* pour toutes les images de teste.

5.7 Conclusion

Dans ce chapitre, nous avons décrit une nouvelle approche à base d'agent pour la résolution des problèmes de segmentation d'images par la croissance de régions. Nous avons combiné deux techniques de l'intelligence en essaim, à savoir, le *PSO* et la stigmergie. Cette approche est spécifiée par quelques caractéristiques qui la distinguent :

- La construction des régions connexes et homogènes, dans l'image à segmenter, est progressive et visuelle.
- L'optimisation de la recherche des régions homogènes et l'estimation de leurs seuils d'homogénéité (les valeurs initiales de seuils de construction) s'effectuent par l'introduction de la technique de *PSO*.
- La construction des régions homogènes et connexes, dans l'image, n'est pas calculée explicitement, par les agents, mais elle émerge de l'interaction stigmergique.
- L'utilisation d'une structure phéromonale permettant l'agrégation des différentes informations de contrôle (critères de segmentation) qui aident dans la prise de décision (de la croissance de construction, la fusion des régions).
- Facilement parallélisable, vu sa nature distribuée.

La méthode a été testée sur des images variées, les résultats obtenus sont présentés, discutés et comparés à la méthode de croissance de régions. Ces résultats mettent en

5. Segmentation d'images par la croissance auto-organisée de régions 102

évidence les caractéristiques de cette méthode et sa robustesse à segmenter différents types d'images, grâce à sa capacité d'adaptation de ses paramètres en fonction du contenu de l'image et l'évolution du processus de construction.

Chapitre 6

Segmentation d'images non supervisée par un seuillage dynamique à multi-niveaux utilisant le BPSO

6.1 Introduction

Dans ce chapitre, nous proposons une nouvelle méthode non-supervisée pour la segmentation d'images par le seuillage à multi-niveaux. Dans cette approche, le problème de segmentation d'images par seuillage est traité comme un problème d'optimisation. Le seuillage optimal peut être accompli à travers une recherche des seuils qui optimisent une fonction objectif, en utilisant l'optimisation par essaim de particules binaires "*BPSO*". L'algorithme proposé "*DMTBPSO*" (Dynamic Multilevel Thresholding using Binary Particle Swarm Optimization) peut déterminer automatiquement le nombre approprié de seuils ainsi que leurs valeurs adéquates dans un temps raisonnable, qui est indépendant du nombre de seuils et de la qualité de l'image. Les expérimentations et les résultats comparatifs avec d'autres méthodes de seuillage à multi-niveaux sur les images synthétiques et réelles montrent l'efficacité de la méthode proposée. Ce travail a été couronné par des publications [63]-[64]-[65].

6.2 Algorithme *DMTBPSO*

Le *DMTBPSO* est un algorithme pour le seuillage dynamique à multi-niveaux, il permet de déterminer le nombre de seuils ainsi que les valeurs appropriées de seuils,

pour segmenter automatiquement une image, en se basant sur l'algorithme *BPSO*. Il commence par un grand nombre de seuils initiaux, qui seront raffinés dynamiquement pour améliorer la valeur d'une fonction objectif.

Un des avantages du *DMTBPSO* est qu'il peut utiliser toute fonction objectif (mesure du critère) choisie par l'utilisateur selon l'ensemble de ses données. Dans ce qui suit deux principaux critères de seuillage à multi-niveaux sont présentés, pour être optimisés, en utilisant l'algorithme proposé. Le premier est le critère de l'entropie (méthode de *Kapur*) [123], et le second est la variance interclasse (méthode d'*Otsu*) [174]. Ces critères ont été largement utilisés dans la littérature pour déterminer la ombinaison optimale de seuils [197]-[191]-[226]-[145]-[147]-[228]-[229]. Tous ces travaux (voir section 4.6.2) exigent que le nombre de seuils soit spécifié en avance. Trouver le "nombre optimal" de seuils, dans un espace de niveau gris, est généralement, un challenge car il nécessite de connaissances a priori.

Le problème de trouver le nombre optimal de seuils, dans un espace de niveaux de gris, a été le sujet de plusieurs efforts de recherche [105]-[221]-[197], cependant, le résultat est encore peu satisfaisant en dépit du montant de recherche dans ce domaine.

6.2.1 Fonction Objectif

Soit A une image contenant N pixels avec les niveaux de gris appartiennent à $[0, L-1]$, $h(i)$ est le nombre d'occurrences du pixel dont le niveau de gris i , $i \in [0, L-1]$, et p_i est la densité de probabilité du niveau de gris i , tels que :

$$N = \sum_{i=0}^{L-1} h(i) \text{ et } P_i = h(i) / N \text{ pour } 0 \leq i \leq L-1 \quad (6.1)$$

Dans le seuillage à multi-niveaux, il existe k seuils (t_1, t_2, \dots, t_k) , permettant de partitionner les pixels de l'image en $K+1$ classes (C_0, C_1, \dots, C_k) qui sont représentées comme suit :

$$C_0 = \{0, 1, \dots, t_1\}, \dots, C_n = \{t_n + 1, t_n + 2, \dots, t_{n+1}\}, \dots, C_k = \{t_k + 1, t_k + 2, \dots, L-1\}$$

Variance interclasse d'*Otsu*

La variance entre $K+1$ classes de l'image est donnée comme suit [228] :

$$f(t_1, t_2, \dots, t_k) = \begin{cases} w_0 w_1 (\mu_0 - \mu_1)^2 + w_0 w_2 (\mu_0 - \mu_2)^2 + w_0 w_3 (\mu_0 - \mu_3)^2 + \dots \\ + w_0 w_k (\mu_0 - \mu_k)^2 + w_1 w_2 (\mu_1 - \mu_2)^2 + w_1 w_3 (\mu_1 - \mu_3)^2 + \dots \\ + w_1 w_k (\mu_1 - \mu_k)^2 + \dots + w_{k-1} w_k (\mu_{k-1} - \mu_k)^2 \end{cases} \quad (6.2)$$

Avec :

$$w_{n-1}(t) = \sum_{i=t_{n-1}+1}^{t_n} p_i$$

$$\mu_{n-1} = \sum_{i=t_{n-1}+1}^{t_n} \frac{i \times p_i}{w_{n-1}}, \quad 1 \leq n \leq (k+1)$$

Les seuils optimaux $[t_1^*, t_2^*, \dots, t_k^*]$ font la variance d'interclasse maximale $t_1^*, t_2^*, \dots, t_k^* = \arg \max_{0 < t_1 < t_2 < \dots < t_k < L} f(t_1, t_2, \dots, t_k)$

Entropie de *Kapur*

L'entropie totale de l'image comportant $k+1$ classes est donnée comme suit :

$$f(t_1, t_2, \dots, t_k) = H_0 + H_1 + \dots + H_k \quad (6.3)$$

tels que :

$$H_0 = - \sum_{i=0}^{t_1-1} \frac{p_i}{w_0} \ln \frac{p_i}{w_0}, w_0 = \sum_{i=0}^{t_1-1} p_i$$

$$H_1 = - \sum_{i=t_1}^{t_2-1} \frac{p_i}{w_1} \ln \frac{p_i}{w_1}, w_1 = \sum_{i=t_1}^{t_2-1} p_i$$

$$H_2 = - \sum_{i=t_2}^{t_3-1} \frac{p_i}{w_2} \ln \frac{p_i}{w_2}, w_2 = \sum_{i=t_2}^{t_3-1} p_i$$

$$H_k = - \sum_{i=t_k}^{L-1} \frac{p_i}{w_k} \ln \frac{p_i}{w_k}, w_k = \sum_{i=t_k}^{L-1} p_i$$

Les seuils optimaux $t_1^*, t_2^*, \dots, t_k^*$, font la variance d'interclasse maximale :

$$t_1^*, t_2^*, \dots, t_k^* = \arg \max_{0 < t_1 < t_2 < \dots < t_k < L} f(t_1, t_2, \dots, t_k)$$

6.2.2 Algorithme *DMTBPSO*

Avant de présenter notre algorithme *DMTBPSO*, nous décrivons les symboles utilisés :

- A est une image contenant N pixels, les valeurs de niveaux de gris appartiennent à $[0, L-1]$; $L-1$ est le niveau de gris maximum dans l'image.
- $T = \{t_k, k = 1 \dots N_t\}$ est l'espace de recherche de particules, contenant N_t seuils; $N_t \leq L-2$.
- $S = \{x_1, \dots, x_i, \dots, x_s\}$ est un essaim de s particules, x_i indique la particule i , avec $x_{i,j} \in \{0, 1\}$, pour $j = 1, \dots, N_t$, tels que, si $x_{i,k} = 1$ alors le seuil t_k dans T , correspondant a été choisis comme une partie de la solution proposée par x_i . Autrement, si $x_{i,k} = 0$ alors le seuil t_k , dans T , correspondant ne présente pas une partie de la solution proposée par x_i .
- n_i est le nombre de seuils représentés par la particule x_i
- $n_i = \sum_{k=1}^{N_t} x_{i,k}$, avec $n_i \leq N_t$

- T_i est la solution de la segmentation représentée par la particule x_i tel que :
 $T_i = (t_k) \forall k : x_{i,k} = 1$, avec $T_i \subseteq T$
- n_{gbest} : le nombre de seuils représentés par la meilleure particule globale \hat{y} "the global best particle"

$$n_{gbest} = \sum_{k=1}^{N_t} \hat{y}_k$$
, avec $n_{gbest} \leq N_t$
- T_{gbest} la solution représentée par \hat{y} tel que : $T_{gbest} = (t_k) \forall k : \hat{y}_k = 1$ avec $T_{gbest} \subseteq T$
- p_{ini} est une probabilité spécifiée par l'utilisateur définie dans [137]-[172], elle est utilisée pour initialiser la position de particule x_i , comme suit :

$$x_{i,k} = \begin{cases} 0 & \text{si } r_k(t) \geq p_{ini} \\ 1 & \text{si } r_k(t) < p_{ini} \end{cases} \quad (6.4)$$

tel que $r_k(t) \sim \cup(0, 1)$ et $k = 1, \dots, N_t$.

Une grande valeur pour p_{ini} résulte de la sélection de la plupart des seuils dans T .

Le principe de l'algorithme *DMTBPSO* est le suivant : T est un ensemble de niveaux de gris ordonnés, initialisé par les niveaux de gris : $1..L-2$, $L-1$ est le niveau gris maximal dans image A . L'essai de particules, S , est initialisé de façon aléatoire. Le *BPSO* est appliqué pour trouver "le meilleur" ensemble de seuils, T_{gbest} , de T qui maximise la fonction objectif f (d'*Otsu* ou de *Kapur*).

L'algorithme est répété avec un nouvel ensemble T , qui est T_{gbest} , composé de n_{gbest} éléments, et avec des nouvelles positions de particules ; les n_{gbest} premiers éléments, de chaque position d'une particule seront initialisés d'une façon aléatoire et le reste sera initialisé à zéro. Quand les critères de terminaison sont satisfaits, T_{gbest} sera l'ensemble de seuils optimums et n_{gbest} sera le nombre optimal de seuils.

L'algorithme *DMTBPSO* est résumé ci-dessous :

Algorithme 6.1 : Algorithme *DMTBPSO*

1. Initialiser $T = \{t_k, t_k = 1 \dots L - 2\}$ est l'ensemble ordonné de N_t seuils, $N_t = L - 2$, pour une image dont les niveaux de gris $\in [0, L - 1]$;
2. Initialiser l'essai S , avec $x_{i,k} \sim U\{0, 1\}$ $i = 1, \dots, s$ et $k = 1, \dots, N_t$ en utilisant Eq.(6.4) ;
3. Initialiser de façon aléatoire la vitesse v_i de chaque particule i dans S telle que $v_{i,k} \in [-3, 3]$, $i = 1, \dots, s$ et $k = 1, \dots, N_t$;
4. Pour chaque particule x_i de S

- Calculer la valeur de la fonction objectif f_i (Eq.(6.2) ou Eq.(6.2) selon la fonction objectif utilisée) ;
- 5. Appliquer les équations de modification de position et de vitesse de *BPSO* Eq. (3.16) et Eq.(3.17)
- 6. Répéter les étapes 4) et 5) jusqu'à un critère de terminaison soit satisfait.
- 7. Réinitialiser T par T_{gbest} .
- 8. Initialiser l'essaim S , avec $x_{i,k} \sim U\{0,1\}$; $i = 1, \dots, s$ et $k = 1, \dots, N_{gbest}$ en utilisant l'Eq. (6.4), et $x_{i,k} = 0$, pour $i = 1, \dots, s$ et $k = N_{gbest} + 1, \dots, N_t$.
- 9. Répéter les étapes de 3) à 8) jusqu'à les critères de terminaison soient satisfaits.

Les Critères de Terminaison (TC) peuvent être un nombre maximal d'itérations donné par l'utilisateur, ou un manque de progrès dans l'amélioration de la meilleure solution trouvée, pour un nombre d'itérations consécutif spécifié, par l'utilisateur. Dans ce travail, nous avons pris le nombre d'itération NI_1 comme un critère de terminaison pour l'étape 6, et le nombre d'itération NI_2 comme un critère de terminaison pour l'étape 9.

Exemple illustratif

On considère l'exemple suivant comme une illustration de l'algorithme *DMLBPSO*, Soient $Nt = 9$ et T a la forme suivante :

29	40	70	85	120	150	165	179	200
----	----	----	----	-----	-----	-----	-----	-----

Un exemple d'une position de la particule x_i .

1	0	1	1	0	1	1	0	1
---	---	---	---	---	---	---	---	---

Soit la position de la meilleure particule globale \hat{y} est la suivant :

0	1	1	0	0	1	0	1	0
---	---	---	---	---	---	---	---	---

Le nombre de seuils choisis $n_{gbest} = 4$ et leurs valeurs T_{gbest} sont :

40	70	150	179
----	----	-----	-----

La réinitialisation de x_i :

0	1	0	1	0	0	0	0	0
---	---	---	---	---	---	---	---	---

Soit la position de la meilleure particule globale est la suivant :

1	0	1	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---

Le nombre de seuils choisis $n_{gbest} = 2$ et leurs valeurs T_{gbest} sont :

40	150
----	-----

Le nombre de seuils sélectionnés est $N_{gbest} = 2$, l'image dont les niveaux de gris $\in [0, 255]$ est, donc, partitionnée en $N_{gbest} + 1$ classes (3 classes) : $C_0 = \{0, 1, \dots, 40\}$, $C_1 = \{41, 42, \dots, 150\}$, $C_2 = \{151, 152, \dots, 255\}$

6.3 Complexité de l'algorithme

La complexité computationnelle des méthodes originales d'*Otsu* et de *Kapur* est $O(L^k)$ qui augmente exponentiellement avec le nombre de seuils k , par l'utilisation de l'algorithme proposé, la valeur de la fonction objectif sera calculée $s \times NI_1 \times NI_2$ fois où : s est le nombre de particules, NI_1 est le nombre d'itérations pris par le *PSO* pour converger (étape 6), NI_2 est le nombre d'itérations pris par le *PSO* pour converger (étape 9). Ces paramètres peuvent être fixés en avance. Typiquement, $s \times NI_1 \times NI_2 \ll N$ (nombre de pixels de l'image). Si ζ est le produit $s \times NI_1 \times NI_2$. Donc la complexité, dans le pire des cas, est $O(N)$ si $\zeta \simeq N$.

6.4 Etude expérimentale

Afin de tester les performances de l'algorithme *DMTBPSO*, nous avons utilisé des images synthétiques et réelles. L'intérêt des images synthétiques est qu'on connaît de manière précise le nombre exact de classes ainsi que la qualité de la segmentation de référence. Ces informations ne sont pas évidemment transmises à *DMTBPSO*, et ne sont utilisées que pour tester la capacité de *DMTBPSO* à retrouver le nombre de seuils et par conséquent le nombre de classes.

6.4.1 Caractéristiques des images de test

Les images réelles utilisées sont Lenna, Poivron, F16 Jet, Mandrill et une image de satellite de Lac Tahoe et une IRM. Les figures FIG. 6.1, FIG. 6.2 et FIG. 6.3 présentent les images de test, leurs histogrammes et le nombre optimal de classes C pour les images synthétiques et l'intervalle optimal C de classes, pour les images réelles, pris de [214].

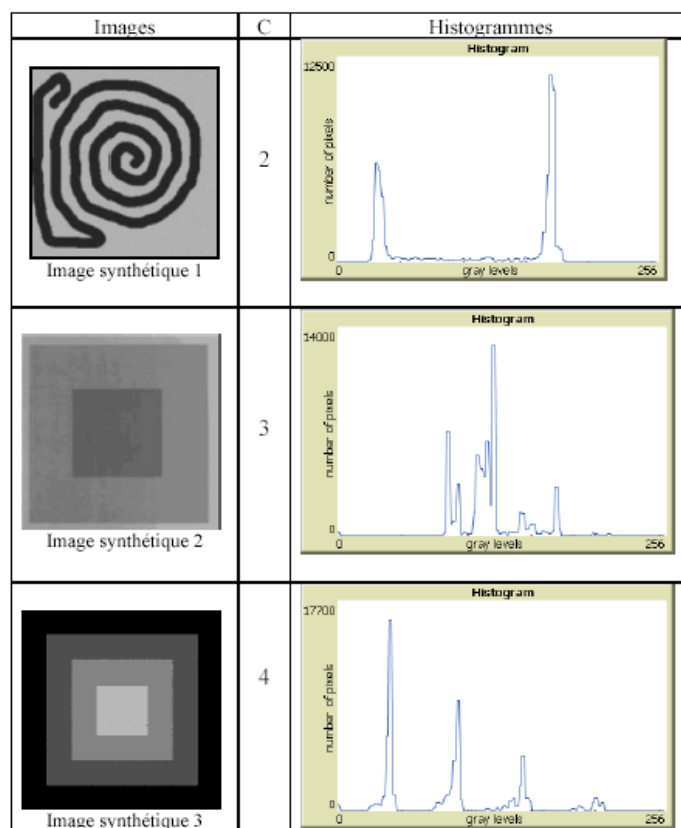


FIG. 6.1 – Caractéristiques des images synthétiques de test.

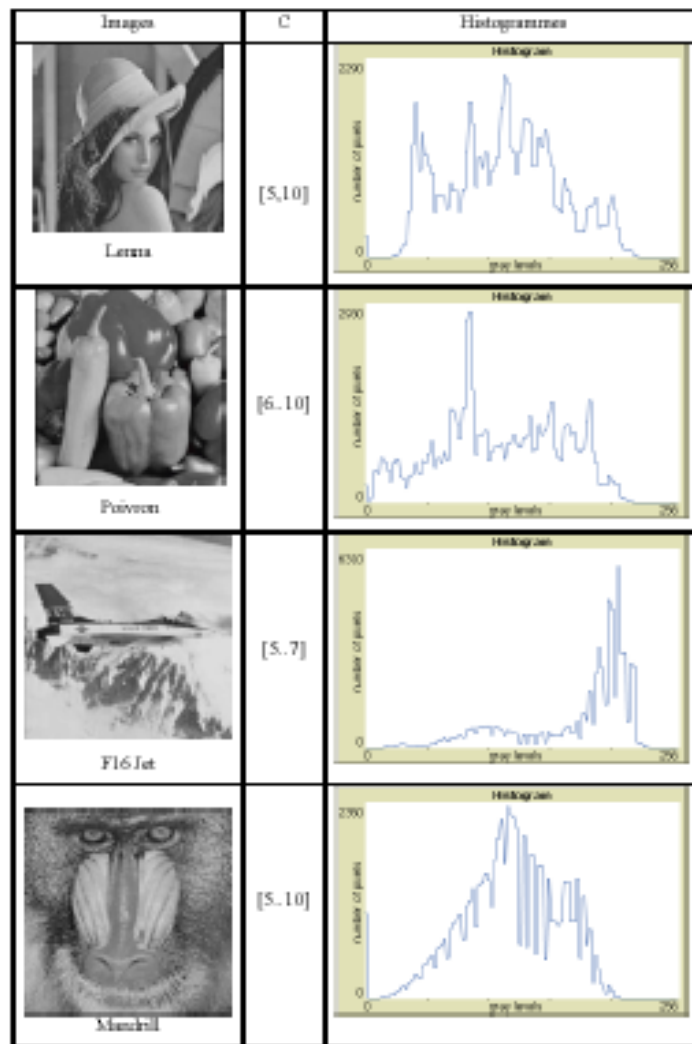


FIG. 6.2 – Caractéristiques des images réelles de test (1)

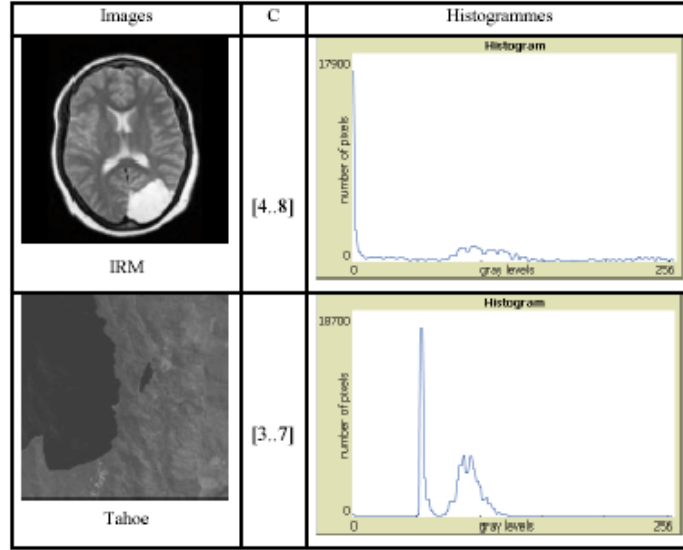


FIG. 6.3 – Caractéristiques des images réelles de test (2)

6.4.2 Mesures d'évaluation

La méthode proposée utilise la variance interclasse d'*Otsu* ou l'entropie de *Kapur* comme fonction objectif, pour décider si la meilleure combinaison de seuils $T_{g_{best}}$, dans l'espace des niveaux de gris, maximise la fonction objectif $f(T_{g_{best}})$.

La qualité d'une image segmentée peut aussi être évaluée selon la mesure de l'uniformité U , largement mentionnée dans la littérature [191]-[227] :

$$U = 1 - 2 \times k \times \frac{\sum_{j=0}^k \sum_{i \in R_j} (f_i - m_j)^2}{N (f_{\max} - f_{\min})^2} \quad (6.5)$$

Où :

k est le nombre de seuils ;

R_j est la région segmentée j ;

f_i est le niveau de gris du pixel i ;

m_j est le niveau de gris moyen des pixels dans la région segmentée j ;

N est le nombre total de pixels dans l'image donnée ;

f_{\max} est le niveau de gris maximal des pixels dans l'image donnée ;

f_{\min} est le niveau de gris minimal des pixels dans l'image donnée ;

La valeur de mesure de l'uniformité est entre 0 et 1. Une plus haute valeur de l'uniformité signifie que la qualité de l'image seuillée est meilleure.

Paramètre	Désignation	Valeur
p_{ini}	Probabilité utilisée pour initialiser les positions des particules	0.75
s	Nombre de particules dans l'essaim S	100
NI_1	Nombre d'Itérations, pour l'étape 6 de l'algorithme	50
NI_2	Nombre d'Itérations, pour l'étape 9 de l'algorithme	2
w	Coefficient d'inertie, sert comme une mémoire de vitesses antérieures	0.72
c_1	coefficient d'accélération ; $c_1+c_2 \leq 4$	2
c_2	coefficient d'accélération ; $c_1+c_2 \leq 4$	2
v_{max}	la vitesse maximale pour que les particules restent dans l'espace de recherche	255

TAB. 6.1 – Paramètres de DMTBPSO

6.4.3 Résultats

Nous présentons dans ce qui suit, quelques résultats expérimentaux que nous avons obtenus, lors de l'exécution de nos algorithmes, sur un processeur Intel Pentium 4 à 2.8 GHz. Durant l'expérimentation, nous avons pris en considération :

- l'estimation comportementale de l'algorithme *DMTBPSO* durant la simulation ;
- la comparaison des résultats obtenus par le *DMTBPSO* avec ceux de la recherche exhaustive d'*OTSU* et de *Kapur* ;
- l'influence des paramètres d'exécution sur le comportement de *DMTBPSO*.

L'application de notre algorithme nécessite la définition des paramètres d'exécution suivants : la valeur de p_{ini} , le nombre de particules s , les critères de la terminaison qui sont le nombre d'itérations NI_1 (pour l'étape 6) et le nombre d'itérations NI_2 (pour l'étape 9), de plus, les paramètres de *PSO* qui sont w , c_1 , c_2 et V_{max} . Le tableau TAB. 6.1 résume les valeurs des paramètres de *DMTBPSO* avec lesquels nous avons obtenu de bons résultats. Ces valeurs sont appliquées pour la segmentation de toutes les images de test.

Estimation comportementale de l'algorithme

Pour avoir une estimation moyenne du comportement de l'algorithme *DMTBPSO*, 50 tests ont été menés sur le jeu d'images de test utilisé. Le tableau TAB. 6.2 illustre la valeur moyenne ainsi que l'écart type : du nombre de classes C , de la valeur de la fonction objectif optimale f et de l'uniformité obtenus U , par ces tests. Des échantillons d'images segmentées résultant de *DMTBPSO* sont présentés dans les figures FIG. 6.4, FIG. 6.5 et FIG. 6.6.

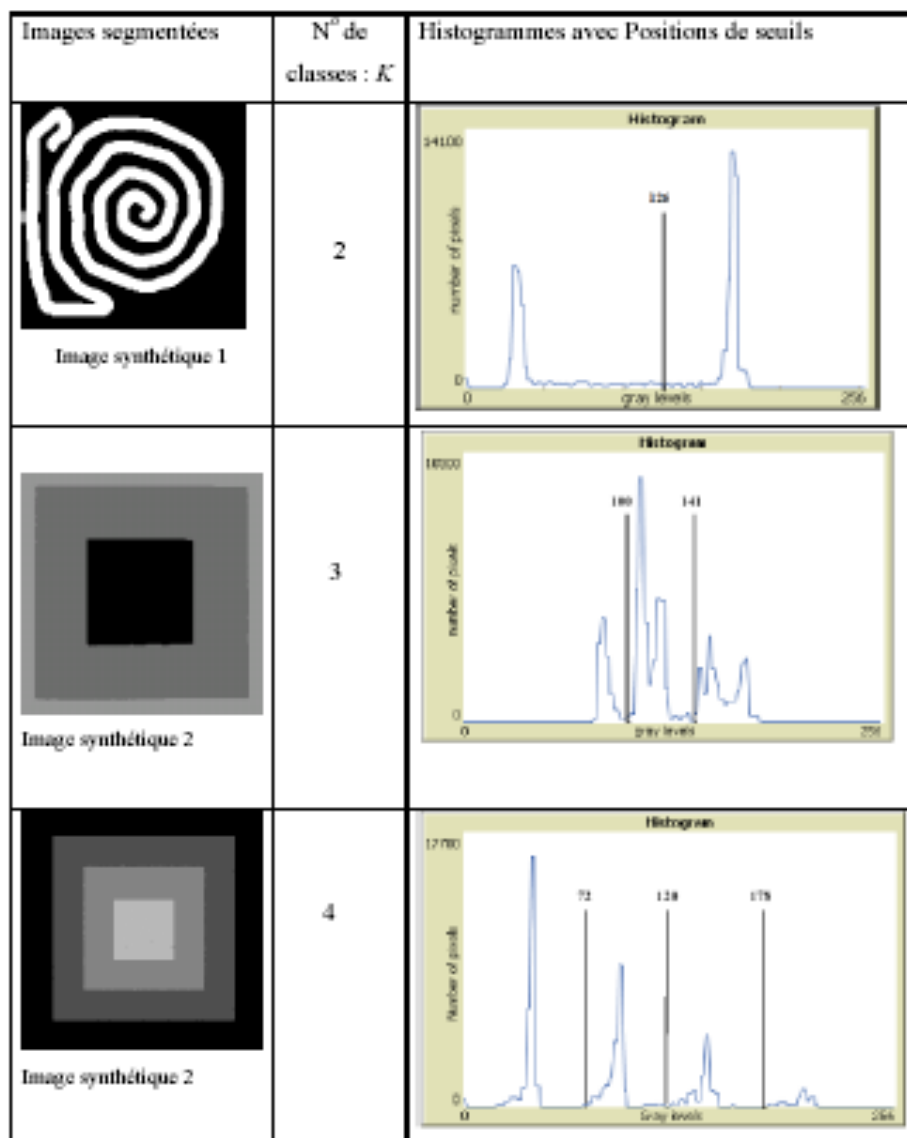


FIG. 6.4 – Echantillons des images synthétiques segmentées résultant de *DMTBPSO*

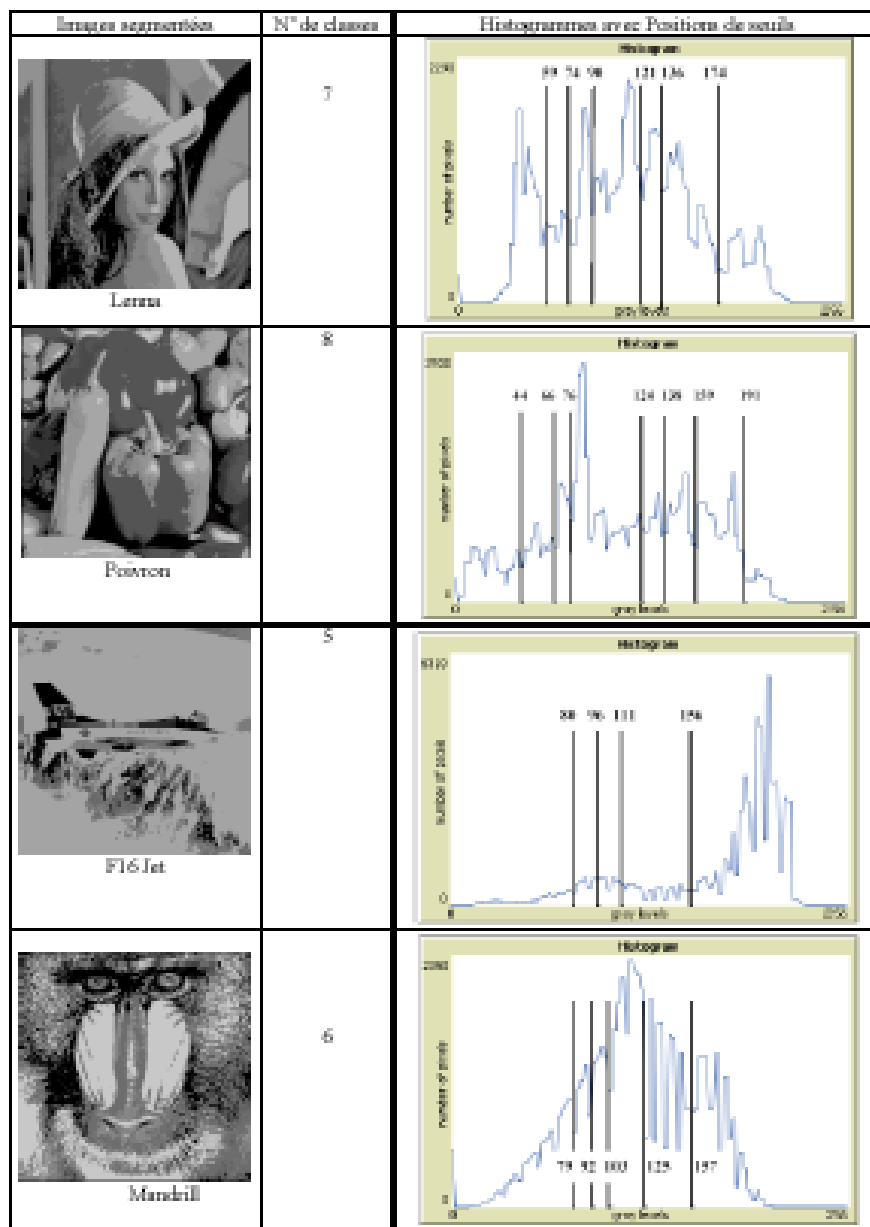


FIG. 6.5 – Echantillons des images réelles segmentées résultant de *DMTBPSO* (1)

Image	C_{opt}	DTBPSO-otsu		DMTBPSO-Kapur	
		C_{Otsu}	U_{Otsu}	C_{Kapur}	U_{Kapur}
syntétique1	2	2.13 ± 0.102	0.9841 ± 0.025	2.04 ± 0.074	0.9827 ± 0.018
syntétique2	3	3 ± 0	0.8508 ± 0.041	3 ± 0.0	0.8471 ± 0.248
syntétique3	4	4 ± 0.0	0.8598 ± 0.104	4 ± 0.10	0.8471 ± 0.248
Lenna	[5,10]	6.64 ± 0.443	0.9535 ± 0.140	6.45 ± 0.421	0.9485 ± 0.235
Poivron	[6,10]	6.32 ± 0.251	0.9214 ± 0.324	7.11 ± 0.160	0.9032 ± 0.248
F16 Jet	[5, 7]	5.20 ± 0.126	0.9642 ± 0.224	4.85 ± 0.546	0.9424 ± 0.455
Mandrill	[5,10]	6.23 ± 0.381	0.9156 ± 0.114	7.24 ± 0.187	0.9008 ± 0.301
Tahoe	[3,7]	4.89 ± 0.626	0.9458 ± 0.362	4.88 ± 0.413	0.9423 ± 0.445
IRM	[4,8]	5.18 ± 0.287	0.8968 ± 0.251	5.46 ± 0.661	0.8862 ± 0.332

TAB. 6.2 – Résultats de segmentation des images de test par DMTBPSO

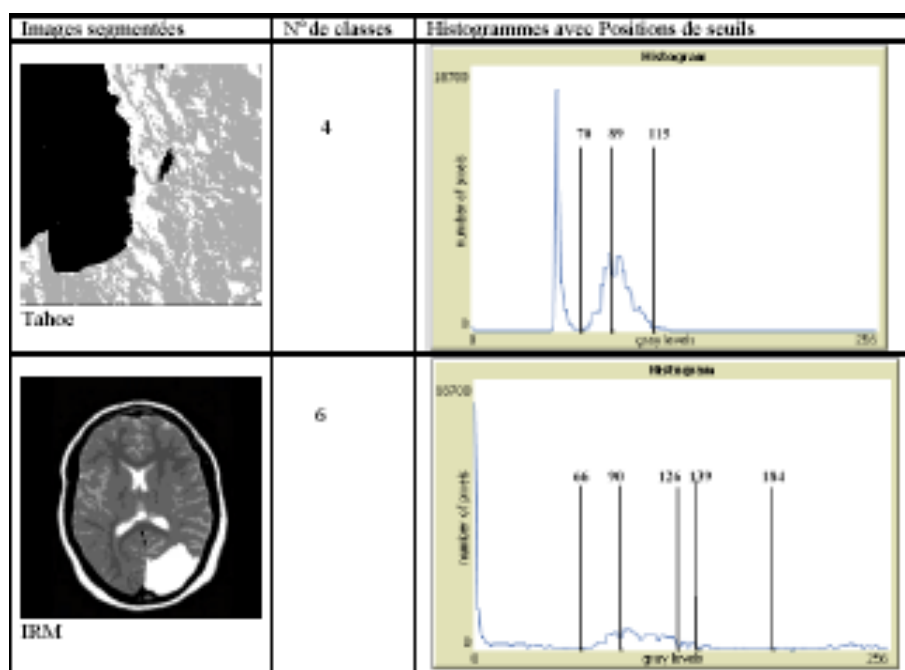


FIG. 6.6 – Echantillons des images réelles segmentées résultant de DMTBPSO (2)

Les résultats exposés dans les figures ci-dessus présentent clairement l'efficacité de DMTBPSO pour trouver des solutions dans l'intervalle optimal.

Comparaison des résultats

La qualité du seuillage obtenue par méthode *DMTBPSO*, avec la fonction objectif d'Otsu et de Kapur, est comparée avec les méthodes, de recherche exhaustive, Otsu et Kapur. Les tableaux TAB. 6.3 et TAB. 6.4 montrent le nombre de seuils K , la valeur de la fonction objectif f et la valeur de l'uniformité obtenus U pour toutes les images de test. La méthode *DMTBPSO* détermine automatiquement le nombre de seuils, cependant les méthodes *Otsu* et *Kapur* sont appliquées, en variant le nombre de seuils. Le nombre de seuils optimal qui maximise la valeur de l'uniformité, est donc déterminé. L'algorithme 6.2 et les tableaux TAB. 6.5 et TAB. 6.6 décrivent le principe de fonctionnement de ces méthodes (OTSU et Kapur).

Algorithme 6.2 : Recherche du nombre de seuils et les valeurs du seuil par la méthode Kapur et Otsu

1. $k = 2$;
2. Trouver l'ensemble de seuils T composé de K seuils, en appliquant la méthode de la recherche (*Otsu* ou *Kapur*);
3. Calculer la valeur de la fonction objectif f et l'uniformité U en utilisant l'ensemble de seuils T ;
4. Si $U(T)$ est améliorée, $k = k + 1$ et aller à l'étape 2; sinon arrêter;
5. Sortir le nombre de seuils et les valeurs de seuils.

Méthodes	k	Seuils Optimaux	U	Temps CPU(sec)
Image synthétique1				
Otsu	1	125	0.9852	0.00
DMTBPSO-otsu	1	126	0.9852	0.54
Kapur	1	126	0.9852	0.00
DMTBPSO-kapur	1	121	0.9852	0.56
Image synthétique2				
Otsu	2	100-141	0.8512	0.24
DMTBPSO-otsu	2	100-141	0.8512	0.54
Kapur	2	100-141	0.8512	0.24
DMTBPSO-kapur	2	99-140	0.8512	0.55
Image synthétique3				
Otsu	3	71-120-179	0.8612	4.70
DMTBPSO-otsu	3	71-120-178	0.8610	0.54
Kapur	3	71-120-179	0.8612	5.10
DMTBPSO-kapur	3	71-124-179	0.8608	0.56
Lenna				
Otsu	5	43-79- 104-144-177	0.9783	16206.25
DMTBPSO-otsu	5	44-80-104-144-177	0.9781	0.54
Kapur	5	43-80-104-141-177	0.9736	16289.14
DMTBPSO-kapur	5	45-83-104-141-176	0.9724	0.56
Poivron				
Otsu	5	55-89-119-141-191	0.9383	16240.30
DMTBPSO-otsu	6	43-76-100-136-159-190	0.9205	0.54
Kapur	5	53-90-119-145-193	0.9583	16297.28
DMTBPSO-kapur	7	79-90-103-125-169-185	0.9079	0.56

TAB. 6.3 – Résultats computationnels pour les images de test (1)

Méthodes	k	Seuils Optimaux	U	Temps CPU(sec)
Jet				
Otsu	4	86-96-113-156	0.9671	292.54
DMTBPSO-otsu	4	80-96-111-156	0.9642	0.54
Kapur	4	86-96 -125 -156	0.9568	301.05
DMTBPSO-kapur	4	74-96-113-156	0.9537	0.56
Mandrill				
Otsu	5	79-92-103-125-146	0.9254	16255.10
DMTBPSO-otsu	5	79-92-103-125-157	0.9233	0.54
Kapur	5	75-92-102-123-148	0.9246	16285.75
DMTBPSO-kapur	6	79-90-103-125-169-185	0.9079	0.56
Tahoe				
Otsu	3	70-89-113	0.9592	4.72
DMTBPSO-otsu	3	70- 89-115	0.9592	0.54
Kapur	3	70-89-113	0.9592	5.10
DMTBPSO-kapur	3	70-90-113	0.9592	0.56
IRM				
Otsu	4	23 70 90 173	0.9121	289.15
DMTBPSO-otsu	5	23 70 101 176	0.9147	0.54
Kapur	4	23 71 103 181	0.9054	298.62
DMTBPSO-kapur	4	66-90-126-139-214	0.8794	0.56

TAB. 6.4 – Résultats computationnels pour les images de test (2)

Images	Otsu		Kapur	
	Ordre hiérarchique de seuils	U	Ordre hiérarchique de seuils	U
Lenna	77-145	0.8026	77-138	0.8324
	61-106-156	0.8284	56-106-159	0.8035
	70-104-144-177	0.8783	64-110-144-177	0.8535
	43-79-104 -144-177	0.9783	43-80-104 -141-177	0.9736
	27-53-99-120- 142-170-211	0.8817	21-56-101-137- 149-177-207	0.8653
Poivron	84-149	0.8042	77-140	0.8204
	97-113-203	0.8147	97-113-198	0.8423
	66-123-136-211	0.8956	58-94-143-171	0.8890
	55-89-119 -141-191	0.9383	53-90-119 -145-193	0.9583
	47-76-99-141 -173-196	0.8545	34-66-90-133 -155-211	0.8177
Jet	111-172	0.9003	117-175	0.8984
	96-149-191	0.8323	72-128-183	0.8343
	86-96 -113-156	0.9671	86-96 -125-156	0.9568
	70-100-127 -156-183	0.8973	80-102-127 -156-191	0.8848

TAB. 6.5 – Principe de fonctionnement des méthodes Otsu et Kapur(1)

Images	Otsu		Kapur	
	Ordre hiérarchique de seuils	U	Ordre hiérarchique de seuils	U
Mandrill	95-121	0.8707	79-121	0.8423
	66-103-151	0.8221	66-102-156	0.8128
	102-123-151-175	0.8191	43-80-102-156	0.8994
	79-92-103 -125-146	0.9254	75-92-102 -123-148	0.9246
	54-82-101 -133-149-182	0.8727	49-88-106 -149-171-194	0.8841
Tahoe	82-128	0.8811	70-128	0.8358
	70-89-113	0.9592	70-89-113	0.9592
	69-88-102-136	0.9343	59-87-102-151	0.9008
IRM	71-139	0.8518	66-139	0.8479
	65-101-159	0.8781	54-103-193	0.8539
	23-70 - 90-173	0.9121	23-71 -103-181	0.9054
	39- 89-124 1 -46-192	0.8614	56-103-129 - 148-197	0.8151

TAB. 6.6 – Principe de fonctionnement des méthodes Otsu et Kapur (2)

De ces résultats, nous pouvons tirer plusieurs conclusions générales à la fin de notre expérience préliminaire :

- La méthode *DMTBPSO* est méthode non supervisée de sélection automatique des seuils. Le temps CPU ne dépend pas de niveaux de seuillage (nombre de seuils) ou de la complexité des images.
- La méthode *DMTBPSO* est une méthode potentielle d'accélérer les méthodes, de recherche exhaustive, *Otsu* et *Kapur* pour le seuillage multi-niveaux.
- La performance de la méthode proposée en utilisant la fonction d'*Otsu* est meilleure qu'en utilisant la fonction de *Kapur*.

6.5 Conclusion

Dans ce chapitre, nous avons proposé une méthode non supervisée, basée sur l'algorithme *BPSO*, permettant de déterminer automatiquement le nombre approprié de seuils et leurs valeurs adéquates, cela est accompli en optimisant une fonction qui utilise les seuils de niveaux de gris comme paramètres.

La méthode proposée présente plusieurs caractéristiques qui la distinguent :

- Facile à représenter et à implémenter, parce qu'elle est basée sur l'algorithme de *PSO*, qui est facile à implémenter ;
- L'utilisation d'un seuillage à multi-niveaux générique ; puisque l'algorithme *DMTBPSO* peut utiliser n'importe quelle fonction objectif (mesure du critère) choisie par l'utilisateur pour son ensemble de données. De plus, tout nouveau critère de seuillage peut être intégré facilement dans le *DMTBPSO*.
- La complexité algorithmique est indépendante du nombre de seuils ou de la complexité des images.

Chapitre 7

Conclusion générale

Ce travail de thèse, consiste à présenter des solutions bio-inspirées pour la problématique de la segmentation d'images. Cela, nous a poussés à faire une étude approfondie sur trois axes : le premier sur les systèmes biologiques pour analyser et comprendre leurs principes sous-jacents et afin de rechercher d'une nouvelle source d'inspiration. Le deuxième axe concerne les méthodes bio-inspirées, pour comprendre leur principe de fonctionnement, leur utilisation et leur adaptabilité au problème de segmentation. Le troisième axe porte sur les méthodes de segmentation existantes pour les analyser et comprendre leurs principes de fonctionnement et leurs difficultés et finalement pour les utiliser dans la comparaison et l'évaluation de nos résultats.

D'après cette étude, nous avons retenus les points suivants :

- Les systèmes biologiques tels que : le cerveau, la génétique des populations, les colonies d'insectes sociaux... ont été des sources d'inspiration pour le développement de travaux de recherches. Les sources d'inspiration que nous fournissent l'éthologie ne sont pas taries, vu la richesse comportementale des insectes sociaux.
- Les différents systèmes biologiques ne sont pas dissociés ; on trouve des combinaisons de mécanismes : génétique/neuronal, génétique/systèmes immunitaires, intelligence comportementale/génétique, intelligence comportementale/neuronal et génétique/ neuronal/intelligence comportementale/swarm ...
- Les méthodes bio-inspirées existantes sont issues de sources d'inspirations dissociées, cependant les phénomènes de mélange de mécanismes peuvent être réalisés par la technique d'hybridation.
- Le développement de solutions bio-inspirés est basé sur l'utilisation des systèmes biologiques comme une métaphore soit : pour les reproduire dans un environnement, ceci nous a conduit à une démarche mimétique, soit pour développer

des artefacts permettant la mise en œuvre de mécanismes sous-jacents comme la stigmergie, l'auto-organisation, etc...

- L'hybridation des méthodes a aussi largement été explorée pour différents types de problèmes.
- L'utilisation des techniques existantes et adaptées à un problème est certainement une voie prometteuse.

Dans cette thèse, notre contribution au développement de méthodes bio-inspirées pour la segmentation d'images, prend deux formes :

Dans un premier temps, nous avons adopté la métaphore bio-inspirée par la mise en œuvre des mécanismes d'auto-organisation et de stigmergie observés dans les processus de construction de nid chez les termites. Nous avons proposé un système multi-agents pour la construction des différentes régions homogènes dans une image en niveau de gris. Les interactions entre agents sont mises en œuvre par le biais de la communication indirecte "stigmergique", permettant l'auto-organisation qui a un effet sur l'auto-régulation du processus de construction de régions homogènes dans l'image et sur l'émergence de la segmentation de l'image. Nous avons aussi introduit la technique de l'optimisation par essaim de particules dans le processus de construction des régions homogènes dans l'image pour obtenir une segmentation adaptative.

Dans un deuxième temps, nous avons adapté une technique existante, qui est l'optimisation par essaim de particules binaires, à un problème de segmentation d'images par seuillage multi-niveaux, pour réaliser une segmentation non supervisée. Nous avons proposé un algorithme *DMTBPSO* pour le seuillage dynamique à multi-niveaux. Cet algorithme cherche les seuils optimaux, en optimisant une fonction objectif "critère de seuillage" qui utilise les valeurs de seuils (valeurs de niveaux de gris) comme des paramètres, en commençant avec un grand nombre de seuils initiaux qui sont dynamiquement raffiné pour améliorer la valeur de la fonction objectif. Il permet de détecter le nombre de seuils et leurs valeurs simultanément avec une complexité algorithmique qui est indépendante du nombre de seuils ou de la complexité des images. Cet algorithme est générique car il peut utiliser n'importe quel critère de seuillage spécifié par l'utilisateur.

Pour valider ces deux méthodes, nous avons fait des expérimentations sur des images synthétiques et réelles. Nous avons tenu en compte, pour la première méthode les caractéristiques de la méthode proposée du point de vu :

- de la robustesse par rapport aux données à segmenter ;
- du rôle du processus de l'auto-organisation dans l'émergence du résultat de la segmentation ;

- de la justesse des résultats obtenus par leur évaluation à travers la mesure de Borsotti ;
- de la comparaison des résultats obtenus avec ceux de la méthode croissance de régions.

Pour la deuxième méthode proposée, durant les expérimentations, nous avons tenu en considération les points suivants :

- l'utilisation de l'algorithme proposé *DMTBPSO* avec le critère Kapur et le critère Otsu pour présenter son caractère générique ;
- l'estimation comportementale de l'algorithme durant la simulation ;
- l'influence des paramètres d'exécution sur le comportement de l'algorithme ;
- l'évaluation de la performance de l'algorithme, en se basant sur l'aspect d'uniformité des régions obtenues, et en prenant comme des intervalles de référence optimaux présentés dans [214], concernant le nombre de classes pour les images réelles ;
- La comparaison des résultats obtenus avec ceux de la recherche exhaustive de Otsu et Kapur.

Les perspectives de ce travail sont relativement nombreuses, tant du point de vue théorique que pratique, nous pouvons les résumer dans les points suivants :

- approfondir l'étude des systèmes biologiques pour développer de nouveaux modèles (algorithmes) bio-inspirés car il existe plusieurs mécanismes qui ne sont pas encore exploités.
- une perspective importante peut être une formalisation plus poussée des relations entre les algorithmes bio-inspirés et les problèmes de segmentation d'images pour qu'on puisse intégrer d'autres algorithmes bio-inspirés dans la segmentation d'images qui n'ont pas encore été réellement exploités.
- une autre approche qui reste à approfondir est l'évaluation des algorithmes bio-inspirés pour la segmentation d'images. De nombreuses fonctions de test existent, mais restent des initiatives ponctuelles, vu que la variété de sources d'inspiration de ces algorithmes et aussi le problème de segmentation qui est un problème mal posé.
- il est possible d'améliorer les résultats de la segmentation par l'utilisation : soit par une coopération mutuelle de plusieurs techniques, en employant des critères différents de segmentation d'images, et qui sont exécutées en parallèle, tout en échangeant mutuellement des informations, ou bien par l'utilisation de l'optimisation multiobjectif (multicritère), pour optimiser, simultanément, plusieurs critères de segmentation.

- aussi, il est possible d'accélérer la convergence de notre algorithme *DMTBPSO* en réduisant l'espace de recherche (en compressant l'intervalle de niveau de gris) par l'utilisation de la transformée en ondelette.
- Enfin, une perspective intéressante reste l'application des métaheuristiques à la segmentation de séquences d'images. Ceci revient à résoudre un problème d'optimisation dynamique.

Bibliographie

- [1] R.Adams, L.Bischof, Seeded region growing. *IEEE Trans. Pattern Analysis Machine Intelligence*, 16(6), pages 641–647, Jun 1994.
- [2] E.Addressi, E.Visalberghi, Social facilitation of eating novel food in tufted capuchin monkeys (*Cebus apella*) : Input provided by group members and responses affected in the observer. *Animal Cognition*, 4 : pages 297-303, 2001.
- [3] R.D.Alexander, The evolution of social behaviour, *Annual Review of Ecology and Systematics*, 5 : pages 325-383, 1974.
- [4] J.L.Adams, G.Yahiaoui, Techniques avancées pour le traitement de l'information ; Réseaux de neurones, logique floue, algorithmes génétiques, Cépadues-Editions, 2ème édition, 2002.
- [5] H.L.Anderson, R.Bajcsy, M.Mintz, A modular feedback system for image segmentation. Technical report, University of Pennsylvania, GRAS P Lab, 1987.
- [6] P.U.Andrés, Structure-Adaptable Digital Neural Networks, Thèse de Doctorat No 2052 , EPFL, Lausanne 1999.
- [7] R.Arkin, Behavior-based robotics, MIT Press, Cambridge, MA, 1998.
- [8] P.Arnaud, des moutons et des robots ; Architecture de contrôle réactive et déplacement collectifs de robots. Presses Polytechniques et Universitaires Romandes, Collection META, 2000.
- [9] G.H.Ball, D.J.Hall, A clustering technique for summarizing multi-variate data, *Behavioral Science*, Vol.12, pages 153-155, 1967
- [10] A.Banks, J.Vincent, C.Anyakoha, A review of particle swarm optimization. Part I : background and development. *Nat. comput.*, pages 1-18, 2007.
- [11] A.Banks, J.Vincent, C.Anyakoha, A review of particle swarm optimization. Part II : hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications. *Nat. comput.*, pages 1-16, 2007.
- [12] R.Beckers, J.L.Deneubourg, S.Goss, Trails and U-Turns in the Selection of a Path by the Ant *Lasius Niger* », *J.Theor.Biol.*,vol. 159, pages 397–415, 1992.
- [13] R.Beckers, O.Holland, J.Deneubourg, From local action to global tasks : stigmergy and collective robotics, *Proceeding Of Artificial life 4*, page 181-189.Mit Press,1994.
- [14] F.Bellet, M.Salotti, C.Garbay : Une approche opportuniste et coopérative pour la vision de bas niveau, *Traitement du Signal*, Volume 12 - n° 5,1995.

- [15] F.Bellet, Une approche incrémentale, coopérative et adaptative pour la segmentation des images en niveau de gris. Thèse de Doctorat, Institut National Polytechnique de Grenoble, France, juin 1998.
- [16] J.Benois ; D.Barba, Image segmentation by region-contour cooperation for image coding, In Proceedings of the 11th International Conference on Pattern Recognition, volume 3, pages 331-334, 1992.
- [17] P.Bertolino, A.Montanvert, Coopération régions contours multirésolution en segmentation d'image, pages 299-307, in Actes du 10e Congrès AFCET/Reconnaissance des formes et intelligence artificielle, Rennes, Association française pour la cybernétique économique et technique, Paris, 16-18 janvier 1996.
- [18] G.Beurier, Codage indirect de la forme dans les systèmes Multi-agents ; Emergence multi-niveaux, Morphogénèse et Evolution, Thèse de Doctorat en Informatique, Université des science et techniques du Languedoc Montpellier II, 2007.
- [19] C.Blum, Ant colony optimization : Introduction and recent trends. Physics of Life Reviews. Vol. 2, pages 353-373, 2005.
- [20] O.Boissier, Y.Demazeau, ASIC : an architecture for social and individual control and its application to computer vision. In proc. of MAAMAW'94, Lectures Notes in AI 1069, pages 135-149, 1994.
- [21] E.Bonabeau, G.Théraulaz, Intelligence collective , Hermès, Paris, 1994.
- [22] E.Boneabeau, G.Theraulaz, J.L.Deneubourg, Franks, Rafelsberger, Joly, Blaco, A model for the emergence of pillars, walls and royal chambers in termite nests. Phil. Trans. R. Soc. London 353 : pages 1561-1576, 1997.
- [23] E.Bonabeau, M.Dorigo, G.Theraulaz, Swarm Intelligence, From Natural to Artificial Systems, Oxford University Press, 1999.
- [24] P.Bonnin, Méthode Systématique de conception et de réalisation d'applications en vision par ordinateur, Thèse de Doctorat, Université de Paris VII, France, 1991.
- [25] P.Bonnin, B.Hoeltzener-Douarin, E.Pissaloux, A new way of image data fusion : the multi-spectral cooperative segmentation, pages 572-575, in Proceedings of the 1995 International Conference on Image Processing. Washington, IEEE Computer Society, Washington, D.C., 3572 pages 23-26 octobre, 1995.
- [26] M.Borsotti, P.Campadelli, R.Schettini, Quantative evaluation of color image segmentation results , Patt. Rec Lett., vol.19, pages 741-747, 1998.
- [27] A.Boucher, Une approche décentralisée et adaptative de la gestion d'informations en vision. Thèse de doctorat de l'université Joseph Fourier Grenoble I,1999.
- [28] C.Bourjot, V.Chevrier, A.Bernard, B.Krafft, Coordination par le biais de l'environnement : une approche biologique, In, actes des 7èmes JFIADSMA, St Gilles les Bains, pp 237-250, Hermes, 1999.
- [29] C.Bourjot, V.Chevrier, De la simulation de construction collective à la détection de régions dans les images à niveaux de gris : l'inspiration des araignées sociales, JFIADSMA, 2001.

- [30] C.Bourjot, V.Chevrier, V.Thomas, A new swarm mechanism based on social spiders colonies : from web weaving to region detection. *Web Intelligence and Agent Systems*, 1(1) : 47-64, IOS Press, Amsterdam, The Netherlands, 2003.
- [31] P.Bourret, J.Reggia, Une approche connexionniste de l'intelligence artificielle, ISBN :2-87717-016-0, Edition Teknea-1991.
- [32] R.A.Brooks, A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 286(1), pages 14–23, 1986.
- [33] R.A.Brooks, Intelligence without representation, *Artificial Intelligence Journal*, 47, pages 139–159, 1991.
- [34] B.Bullnheimer, R.F.Hartl, C.Strauss, An Improved Ant system Algorithm for the Vehicle Routing Problem. Sixth Viennese workshop on Optimal Control, Dynamic Games, Nonlinear Dynamics and Adaptive Systems, Vienna (Austria). 1997.
- [35] B.Bullnheimer, R-F.Hartl, C.Strauss, A New Rank Based Version of the Ant System : A Computational Study, *Central European Journal for Operations Research and Economics*, 7(1), pages 25-38,1999.mentation results , *Patt. Rec Lett.*, vol.19, pages 741-747, 1998.
- [36] S.Camazine, J.L.Deneubourg, N.R.Franks, J.Sneyd, E.Bonabeau, G.Theraulaz, *Self-Organization in Biological Systems*, Princeton : Princeton University Press, 2001,
- [37] V.Camps, Vers une théorie de l'auto-organisation dans les systèmes multi-agents basée sur la coopération : Application à la recherche d'information dans un système d'information répartie, Thèse de Doctorat en Informatique, Université Paul Sabatier de Toulouse, 1998.
- [38] Y.Chang, H.Yan, An effective multilevel thresholding approach using conditional probability entropy and genetic algorithm, in : J.S. Jin, P. Eaqdes, D.D. Feng, H. Yan (Eds.), *Conferences in Research and Practice in Information Technology*, vol. 22, ACS, 2003 (21).
- [39] M.Cheriet, J.N.Said, C.Y.Suen, A recursive thresholding technique for image segmentation, *IEEE Trans. on Image Processing*, Vol. 7, 6, pages 918-921, 1998.
- [40] D.Chialvo, M.Millonas, How Swarms Build Cognitive Maps, in : L. Steels (editor), *The Biology and Technology of Intelligent Autonomous Agents*, pages 439-450, NATO ASI Series, 1995.
- [41] E.Chiarello, L.M.Jolion, C.Amoros. Regions growing with the stochastic pyramid : Application in landscape ecology. *Pattern Recognition*, 29(1), pages 61-75, 1996.
- [42] L.Chrétien, Organisation spatiale du matériel provenant de l'excavation du nid chez *Messor barbarus* et des cadavres d'ouvrières chez *Lasius niger* (Hymenopterae : Formicidae), Thèse de Doctorat, Département de Biologie Animale, Université Libre de Bruxelles, 1996.
- [43] K.Cho, P.Meer, Image Segmentation from Consensus Information, *Computer Vision and Image Understanding*, vol. 68, n° 1, pages 72-89, 1997.

-
- [44] C.C.Chu, J.K.Aggarwal, The integration of image segmentation maps using region and edge information . IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(12), pages 1241-1252, 1993.
- [45] V.Cicirello, S.Smith, Wasp-like Agents for distributed Factory Coordination, rapportno CMU-RI-TR-01-39, Décembre 2001, Robotics Institute, Carnegie Mellon University, Pittsburgh, 2001.
- [46] M.Clerc, J.Kennedy, The Particle Swarm : Explosion, Stability, and Convergence in a Multi-Dimensional Complex Space, In Proceedings of the IEEE Transactions on Evolutionary Computation, volume VI, pages 58–73. 2002.
- [47] J.P.Cocquerez, S.Philipp,. Analyse d'images, filtrage et segmentation. Masson, Paris, 1995.
- [48] A.Coloni, M.Dorigo, V.Maniezzo, Distributed Optimization by Ant Colonies, In Proceedings of First European Conference on Artificial Life, pages134-142, 1991.
- [49] A.Coloni, M.Dorigo, V.Maniezzo, Distributed Optimization by Ant Colonies, Proceedings of the First European Conference on Artificial Life, Paris, France, F.Varela and P.Bourgine (Eds.), Elsevier Publishing, pages 134-142, 1992.
- [50] A.Coloni, M.Dorigo, F.Maffioli, V.Maniezzo, G.Righini, M.Trubian, Heuristics from Nature for Hard Combinatorial Problems. International Transactions in Operational Research, 3(1), pages1-21, 1996.
- [51] C.Darwin, On the origin of species (De l'origine des espèces), John Murrey, London, 1859.
- [52] C.Darwin, On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life. 6 éd. : Domaine public, février 1872.
- [53] D.Dasgupta, N.Attoh-Okine, Immunity-based systems : a survey. Systems, Man, and Cybernetics, 1997.Computational Cybernetics and Simulation, 1997 IEEE International Conference on, 1, 1997.
- [54] D.Dasgupta, Artificial Immune Systems and their applications, Springer Verlag,1999.
- [55] L.Djerou, N.Khelil, M.Batouche, La vie artificielle pour la segmentation d'images, ISBN : 9973-51-546-3, Sousse, Tunisie, 2006.
- [56] L.Djerou, N.Khelil, M.Batouche, Une approche adaptative pour la segmentation d'image, pages 211- 216 TAIMA2007, Hammamet, Tunisie,. ISBN : 978-9973-61-802- 3, 2007.
- [57] L.Djerou, N.Khelil, M.Batouche, Une approche incrémentale et adaptative pour la segmentation d'image, la première Journée Nationale sur les Applications des Métaheuristiques (JNAM'07), USTHB, Bab Ezzouar (Alger), 29 mai 2007.
- [58] L.Djerou, N.Khelil, M.Batouche, A biologically inspired approach for image segmentation , In the proceedings of MCSEAI, pages 296-301, Oran April 28-30, 2008.

- [59] L.Djerou, N.Khelil, M.Batouche, Image Segmentation by Self-organizing Region Growing, In the proceedings of CISIM-08, pages 171-176, IEEE press, Ostrava, The Czech Republic, June 26 - June 28, 2008.
- [60] L.Djerou, N.Khelil, M.Batouche, From termite construction to region growing, In proceedings of the International Conference on Metaheuristics and Nature Inspired Computing - META'08, October 29-31 2008, Hammamet, Tunisia.
- [61] L.Djerou, M.Batouche, Une approche biomimétique pour la segmentation d'image, courrier du savoir- N°09, pages 33-38, ISSN-1112-3338, MARS 2009.
- [62] L. Djerou, N.Khelil, A.Zerarka, M.Batouche, Searching for the best Points of interpolation using swarm intelligence techniques, chapter book in the book 'Particle swarm optimization' ISBN 978-953-7619-48-0. Itech Education and Publishing, Vienna, Austria, 2009.
- [63] L.Djerou, H.Dehipi, N.Khelil and M.Batouche, Using the BPSO Algorithm in Image Segmentation for Dynamic Thresholding, in the proceeding of the International Conference on Bio-Inspired Computing : Theories and Applications (BIC-TA 2009). pp.402-407, IEEE press, Beijing, China, 2009.
- [64] L.Djerou, H.Dehipi, N.Khelil and M.Batouche Automatic Multilevel Thresholding using Binary Particle Swarm Optimization for image segmentation, in the proceeding of the 2009 International Conference of Soft Computing and Pattern Recognition, pages.67-71, IEEE Computer society, Malaysia, 2009.
- [65] L.Djerou, H.Dehipi, N.Khelil, M.Batouche, Self- adaptative thresholding for image segmentation, 5ème SYMPOSIUM INTERNATIONAL Images Multimédias Applications Graphiques et Environnements, Biskra, Algérie 03, 04 - 05 Novembre 2009.
- [66] Y.Demazeau, From interactions to collective behaviors in agent-based systems, In Proceeding of First European Conference on Cognitive Science, Saint-Malo, France, April 1995.
- [67] J.L.Deneubourg, Application de l'ordre par fluctuations la description de certaines étapes de la construction du nid chez les termites. Insectes Sociaux 24, pages 117-130, 1977.
- [68] J.L.Deneubourg, S.Goss, N.Franks, A.B.Sendova-Franks, C.Detrain, L.Chretien, The dynamic of collective sorting robot-like ants and ant-like robots, In J.Meyer, et S.Wilson, éditeurs : SAB'90 - 1st Conf. On Simulation of Adaptive Behavior : From Animals to Animats, pages 356-365, 1990.
- [69] J.L.Deneubourg, S.Aron, S.Goss, J.M.Pasteels, The self-organizing exploratory pattern of the argentine ant, Journal of Insect Behavior, 3, pages 159-168, 1990.
- [70] G.Di Caro, M.Dorigo, AntNet : Distributed stigmergic control for communications networks, Journal of Artificial Intelligence Research, vol. 9, pages 317-365, 1998.
- [71] J.Dréo, A.Pétrowski, P.Siarry, E.Taillard, Métaheuristiques pour l'optimisation difficile. Eyrolles, 2003.
- [72] J.Dréo, Adaptation de la méthode des colonies de fourmis pour l'optimisation en variables continues. Application en génie biomédical, Thèse de Doctorat, Université de Paris 12, 2004.

- [73] J.Dréo, P.Siarry, Métaheuristiques pour l'optimisation et auto-organisation dans les systèmes biologiques ARTICLE VOL 25/2, pages 221-240, doi :10.3166/tsi.25.221-240, 2006.
- [74] G.Dreyfus, M.Samuelides, J.Martinez, M.Gordon, F.Badran, S.Thiria, L.Hérault, Réseaux de neurones : Méthodologies et applications, 2^{ème} edn. Algorithmes, 2004.
- [75] M.Dorigo, V.Maniezzo, A.Colomi, Positive Feedback as a Search Strategy, Rapport Technique 91-016, Politecnico Milano, Italy,1991.
- [76] M.Dorigo, L.M.Gambardella, Ant Colony System : A Cooperative Learning Approach to the Traveling Salesman Problem. IEEE Transactions on Evolutionary Computation, 1(1) :53-66, 1997.
- [77] M.Dorigo, G.D.Caro, The ant colony optimization metaheuristic, New Ideas in Optimization, D.Corne, M.Dorigo, and F. Glover, Eds. New York : McGraw-Hill, 1999.
- [78] M.Dorigo and C.Blum, Ant colony optimization theory : A survey. Theoretical Computer Sc. 2005, Vol. 344, pp. 243-278, 2005
- [79] A.Drogoul, J.Ferber, From tom-thumb to the dockers : Some experiments with foraging robots. In From Animals to Animats II, pages 451-459, Cambridge MIT Press, 1993.
- [80] J.Fan, D.K.Y.Yau, A.K.Elmagarmid, W.G.Aref, Automatic Image Segmentation by Integrating Color-Edge Extraction and Seeded Region Growing, IEEE Transactions on Image Processing, vol. 10, n° 10, pages 1454-1466, 2001.
- [81] J.Ferber, Les systèmes multi-agents : vers une intelligence collective. Informatique, Intelligence Artificielle. Interéditions, 1995.
- [82] J.Ferber, J.P.Muller, Influences and reaction : A model of situated multiagent systems. In Proceedings of International Conference on MultiAgents Systems (ICMAS'95), 1995.
- [83] L.J.Fogel, A.J.Owens, M. J.Walsh. Artificial Intelligence Through Simulated Evolution. Wiley, 1966.
- [84] L.J.Fogel, P.J.Angeline, D.B.Fogel,. An Evolutionary Programming Approach to Self-Adaptation on Finite State Machines, pages 355-365 of : Evolutionary Programming, 1995.
- [85] D.B.Fogel, Evolutionary Computation : A New Transactions ; Evolutionary computation, IEEE Transactions on Volume 1, pages 1-2, April 1997.
- [86] E.W.Forgy, Cluster analysis of multivariate data : efficiency versus interpretability of classification. Biometrics,21, 1965.
- [87] N.R.Franks, A.Sendova-Franks, Brood sorting by ants : Distributing the workload over the work-surface. Behavioral Ecology and Sociobiology, 30, p.109-123, 1992.
- [88] A.S.Fraser, Simulation of genetic systems by automatic digital computers, Australian Journal of Biological Sciences, vol. 10, pages 484-491, 1957.

- [89] D.Fresneau, Biologie et comportement social d'une fourmi ponérine néotropical (Pachycondyla Apicalis). Thèse d'Etat, Université de Pris XIII, Laboratoire D'Ethologie Expérimentale et Comparée,1994.
- [90] H.Frigui, R.Krishnapuram, A Robust Competitive Clustering Algorithm with Applications in Computer Vision, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21(5), pages 450-465, 1999.
- [91] K.Fu, J.Mui, A survey on image segmentation, Pattern Recognition13, pages 3–16, 1981.
- [92] C.Fuh, S.Cho, K.Essig, Hierarchical Color Image Region Segmentation for Content-Based Image Retrieval Systems. IEEE Transactions on Image Processing, vol. 9, no. 1, pages 156-162, 2000.
- [93] K.Fukunaga, Introduction to statistical pattern recognition, Academic Press, 1972.
- [94] A.Gagalowicz, O.Monga, Un algorithme de segmentation hiérarchique, Actes du 5ème congrès RFIA de l' AFCET, pages 163-177, 1985.
- [95] L.M.Gambardella, M.Dorigo, Ant-Q : A Reinforcement Learning Approach to the Traveling Salesman Problem. In Proceedings of ML-95, Twelfth International Conference on Machine Learning, Tahoe City, CA, A. Prieditis and S. Russell (Eds.), Morgan Kaufmann, pages 252-260. 1995.
- [96] J.P.Gambotto, A new approach to combining region growing and edge detection. Pattern Recognition Letters, vol. 14, n° 11, p. 869-875, 1993.
- [97] L.Gaubert, Auto-organisation et émergence dans les systèmes couplées, individuation de données issues de systèmes biologiques couplés, Thèse de Doctorat de l'Université de Bretagne Occidentale,2007.
- [98] D.E.Goldberg, Genetic Algorithms in Search, Optimization and Machine learning, Addison-Wesley, 1989.
- [99] D.E.Goldberg, Algorithmes génétiques. Exploration, optimisation et apprentissage automatique, Addison-WesleyFrance, 1994.
- [100] R.C.Gonzalez, R.E.Woods, Digital image processing. Pearson Prentice Hall, 2002.
- [101] S.Goss, S.Aron, J.L.Deneubourg, J.M.Pasteels, Self-organized shortcuts in the argentine ant. Naturwissenschaften, Vol 76, ages 579–581, 1989.
- [102] P.P.Grassé. La reconstruction du nid et les coordinations inter-individuelles chez bellicositermes natalensis et cubitermes sp. la théorie de la stigmergie : essai d'interprétation du comportement des termites constructeurs. Insectes Sociaux, 6, pages 41–81, 1959.
- [103] G.Hamerly, Learning Structure and Concepts in Data using Data Clustering, PhD Thesis. University of California, San Diego, 2003.
- [104] KHamrouche, M.Diaf, P.Siarry, A multilevel automatic thresholding method based on a genetic algorithm for a fast image segmentation, Comput. Vis. Image Understand. (2007), doi :10.1016/j.cviu.2007.09.001, 2007.

-
- [105] Y.Han, P.Shi, An improved ant colony algorithm for fuzzy clustering in image segmentation, *Neurocomputing*, Vol. 70, 4, pages 665-671, 2007.
- [106] R.Haralick, L.Shapiro, Image segmentation techniques, *Computer Vision, Graphics and Image Processing* 29, pages 100–132, 1985.
- [107] F.Harrouet, J.Tisseau, P.Reignier, P.Chevallier. oRis : un environnement de simulation interactive multi-agents. *Technique et Science Informatiques*, 21(4) :499–524, 2002.
- [108] S.Hassas, Systèmes Complexes à base de Multi-Agents Situés, Mémoire d’Habilitation à Diriger les Recherches en Informatique, Université Claude Bernard, Lyon 1, 2003.
- [109] S.Haykin, *Neural Networks - A comprehensive Foundation* Macmillan College Publishing Company, New York, 1994.
- [110] F.Heppner, U.Grenander, A stochastic nonlinear model for coordinated bird flocks. AAAS Publication, Washington, DC, 1990.
- [111] F.Heppner, U.Grenander, A stochastic nonlinear model for coordinated bird flocks. In S. Krasner, Ed., *The Ubiquity of Chaos*. AAAS Publications, Washington, DC, 233-238, 1990.
- [112] J.H.Holland, Outline for logical theory of adaptive systems, *J.Assoc.Comput. Mach.*, vol. 3, pages 297–314, 1962.
- [113] J.H.Holland, *Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence*, Ann Arbor : University of Michigan Press, 1975.
- [114] B.Hölldobler, E.O.Wilson, The multiple recruitment systems of the African weaver ant *Oecophylla longinoda* (latreille). *Behav. Ecol. Sociobiol*, 3 :19–60, 1978.
- [115] B.Hölldobler, E.O.Wilson, *The Ants*. Harvard University Press, Cambridge, 1990.
- [116] S.L.Horowitz, T.Pavlidis, Picture segmentation by tree transversal algorithm. *J. ACM*, Vol. 32, 2, pages 368-388, 1976.
- [117] N.Ito, R.Kamekura, Y.Shimazu, T.Yokoyama, Y.Matsushita, The Combination of Edge Detection and Region Extraction in Nonparametric Color Image Segmentation. *Information Sciences*, vol. 92, pages 277-294, 1996.
- [118] A.Jain, J.Mao, K.Mohiuddin, Artificial neural networks : A tutorial. *IEEE Computer*, pages 31-44, March 1996.
- [119] A.Jain, R.Duin, J.Mao, *Statistical Pattern Recognition : A review*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No.1, pages 4-37, 2000.
- [120] C.Jinsong, W.Hongqi, Z.Xiaokuan, Entropic thresholding method using genetic algorithm, in : *Proceedings of Geoscience and Remote Sensing Symposium*, Hamburg, Germany, vol. 2, pages 1247–1249, 1999.
- [121] J.M.Jolion, A.Montanvert, La pyramide adaptative : construction et tome 1, p. 197-206. utilisation pour l’analyse de scènes 2D, Actes du 7ème congrès RFIA de l’AFCET, Paris, novembre 1989.

-
- [122] J.M.Jolion, A.Montanvert, the adaptive pyramid : framework for 2D image analysis. CVGIP : Image Underst., Vol.55, 3, pp. 339-348, 1992.
- [123] J.N.Kapur, P.K.Sahoo, A.K.C.Wong, A new method for gray-level picture thresholding using the entropy of the histogram , Computer Vision Graphics Image Processing, 29, pages 273-285, 1985.
- [124] R.Kara-Falah, P.Bolon, J.P.Cocquerez, A region-region and region-edge cooperative approach of image segmentation. In Proceedings of the IEEE ICIP Conference, volume 3, pages 470-474, Nov 1994.
- [125] R.Kara-Falah. Segmentation d'images : Coopération, fusion, évaluation. Thèse de doctorat, Université de Savoie, Juin 1995.
- [126] J.Kennedy, R.C.Eberhart, Particle swarm optimization. In Proc. IEEE Int. Conf. on Neural Networks, volume IV, pages 1942-1948, Piscataway, NJ : IEEE Service Center.1995.
- [127] J.Kennedy, R.C.Eberhart, A Discrete Binary Version of the Particle Swarm Algorithm, Proceedings of the Conference on Systems, Man, and Cybernetics, pages 4104-4109, 1997.
- [128] J.Kennedy, Small Worlds and Mega-Minds : Effects of Neighborhood Topology on Particle Swarm Performance. In IEEE Congress on Evolutionary Computation, volume III, pages 1932-1938, 1999.
- [129] J.Kennedy, R.C.Eberhart, Y.Shi, Swarm intelligence, Morgan Kaufmann Publishers, San Francisco, 2001.
- [130] C.D.Kermad, K.Chehdi, Automatic image segmentation system through iterative edge-region co-operation, Image and Vision Computing, vol. 20, pages 541-555, 2002.
- [131] B.G.Kim, J.I.Shim, D.J.Park, Fast image segmentation based on multi-resolution analysis and wavelets, Pattern Recognit. Lett. 24, pages 2995-3006, 2003.
- [132] H.Kitano, System biology : a brief overview. Science, 295 : pages1662-1664, March 2002.
- [133] T.Kohonen, Self-organizing Maps, volume 30, Springer Series in Information Sciences, April 1995.
- [134] J.R.Koza, Genetic Programming : On the Programming of Computers by Means of Natural Selection, MIT Press, Cambridge, MA, 1992.
- [135] W.Kropatsch, Building irregulars pyramids by dual graph contraction, In IEEE Proc. Vision, Image and signal Processing, Vol.142, pages 366-374, 1995.
- [136] P.N.Kugler, M.T.Turvey, Information, natural law, and the self-assembly of rhythmic movement, Hillsdale, NJ : Erlbaum, 1987.
- [137] L.Kuncheva, J.Bezdek, Nearest Prototype Classification : Clustering, Genetic Algorithms, or Random Search?, IEEE Transactions on Systems, Man, and Cybernetics-Part C : Applications and Reviews, vol.28(1), pages 160-164, 1998.
- [138] C.C.Lai, D.C.Tseng, A hybrid approach using Gaussian smoothing and genetic algorithm for multilevel thresholding, Int. J. Hybrid Intell. Syst. 1 (3), pages143-152, 2004.

-
- [139] S.Landau, S.Doncieux, A.Drogoul, J.A.Meyer, SFERES : Un frame work pour la conception de systèmes multi-agents adaptatifs, RSTI-TSI-21. Systèmes multi-agents, pages 427-446, 2002.
- [140] C.Langton, Artificial life, Proceeding of the first workshop on synthesis and simulation of living systems (1987), Addison-Wesley,1989.
- [141] C.Lee, E.Antonsson, Dynamic Partitional Clustering Using Evolution Strategies. In The Third Asia-Pacific Conference on Simulated Evolution and Learning, 2000.
- [142] S.U.Lee, S.Y.Chung, R.H.Park, A comparative performance study of several global thresholding techniques for segmentation, Comput. Vision Graphics Image Process. 52, pages 171-190, 1990.
- [143] Y.Leung, J.Zhang, Z.Xu, Clustering by Space-Space Filtering, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22(12), pages 1396-1410, 2000.
- [144] Y.C.Liang, A.L.Chen, C.C.Chyu, Application of a Hybrid Ant Colony Optimization for the Multilevel Thresholding in Image Processing, Lecture notes in computer science ISSN 0302-9743, 2007.
- [145] P.S.Liao, T.S.Chen, P.C.Chung, A fast algorithm for multilevel thresholding, J. Inf. Sci. Eng. 17, pages 713-727, 2001.
- [146] K.C.Lin, Fast image thresholding by finding the zero(s) of the first derivative of between-class variance, Mach. Vis. Appl. 13, pages 254-262, 2003.
- [147] R.J.Liou, M.H. Horng, T.W.Jiang, Multi-level Thresholding Selection by using the Honey Bee Mating Optimization, 2009 Ninth International Conference on Hybrid Intelligent Systems, 10.1109/HIS.2009.37,pages 147-151, 2009.
- [148] R.Lippmann, An Introduction To Computing with Neural Nets, IEEE ASSP Magazine, volume 4, n° 2, pages 4-22, Avril 1987.
- [149] J.Liu and Y.Y.Tang, Adaptive image segmentation with distributed behavior-based agents. IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(6), pages 544-551, 1999.
- [150] E.D.Lumer, B.Faieta, Diversity and adaptation in populations of clustering ants, In Proceedings of the Third International Conference on Simulation of Adaptive Behaviour, pages 501-508, 1994.
- [151] H.Maturana, F.Varela. Autopoiesis and cognition : The realization of the living. In Robert S.Cohenand Marx W.Wartofsky, editors, Boston Studies in the Philosophy of Science, volume 42. Reidel Publishing Co,1980.
- [152] S.Mazouzi, Z.Guessoum, F.Michel, Une approche multi-agent pour la segmentation d'images de profondeur, RFIA 2008.
- [153] P.Meer, Stochastic pyramids, comput. Vision Graph. Image Process, Vol.45, 5, pages 269-294, 1989.
- [154] H.Meinhardt, Models of Biological Pattern Formation, Academic Press, London, 1982.

- [155] G.J.Mendel, Experiments in Plant-hybridization, Princeton University Press, 1865.
- [156] H.Meunier, Etude des mécanismes sous-jacents aux phénomènes collectifs chez un primate non humain (*Cebus capucinus*) : De l'expérimentation à la modélisation, Thèse de Doctorat, Université Louis Pasteur de Strasbourg, 2007.
- [157] O.Monga, B.Wrobel, Segmentation d'images : vers une méthodologie, Traitement du Signal, vol. 4, n°3, pp 169-193, 1987.
- [158] N.Monmarché, M.Slimane, G.Venturini. On improving Clustering in numerical data bases with artificial ants. In D.Floreato, J.D.Nicoud, F. Mondala, editors, 5th European Conference on Artificial Life (ECAL'99), Lecture Notes in Artificial Intelligence, volume 1674, pages 626–635, Swiss Federal Institute of Technology, Lausanne, Switzerland, Springer-Verlag, 13-17 September 1999.
- [159] N.Monmarché, Algorithmes de fourmis artificielles : applications à la classification et à l'optimisation, Thèse de Doctorat, Université François Rabelais de Tours, 2000.
- [160] N.Monmarché, N.Ramat, L.Desbarat, G.Venturini, Search with genetic algorithms and ant colonies, WU A., Ed., Proceedings of the 2000 Genetic and Evolutionary Computation Conference Workshop, pages 209–211, 2000.
- [161] N.Monmarché, G.Venturini, M.Slimane, On how *Pachycondyla apicalis* ants suggest a new search algorithm. Future Generation Computer Systems, 16(8), pages 937–946, 2000.
- [162] N.Monmarché, C.Guinot, G.Venturini, Fouille visuelle et classification de données par nuage d'insectes volants. RSTI-RIA-ECA : Méthodes d'optimisation pour l'extraction de connaissances et l'apprentissage, (6), pages 729–752, 2002.
- [163] A.Montanvert, P.Meer, A.Rosenfeld, Hierarchical image analysis using irregular tessellations, IEEE Trans. PAMI, vol. 13, pages 307-316, avril 1991.
- [164] M.Mueller, K.Segl, H.Kaufmann, Edge- and region-based segmentation technique for the extraction of large, manmade objects in high-resolution satellite imagery, Pattern Recognition, vol. 37, pages 1619-1628, 2004.
- [165] X.Muñoz, X.Cufi, J.Freixenet, J.Marti, A new approach to segmentation based on fusing circumscribed contours, region growing and clustering. P. 800-803, in Proceedings of the IEEE International Conference on Image Processing 2000 (ICIP'00). Vancouver, 1045 p, 10-13 septembre 2000.
- [166] A.Nakib, Conception de métaheuristiques d'optimisation pour la segmentation d'images. Application à des images biomédicales, Thèse de Doctorat, Université Paris 12-Val Marne UFR de Science et Technologie, 2007.
- [167] A.M.Nazif, M.D.Levine, An Optimal Set of Image Segmentation Rules. Pattern Recognition Letters, vol. 2, pages 243-248, 1984.
- [168] G.Nicolis, I.Prigogine, Self-organization in Non-equilibrium Systems, New York, 1977.
- [169] S.Nouyan, Agent-Based Approach to Dynamic task Allocation, M.Dorigo, G. Di Caro, M.Sampels, Eds., Proceedings of the Third International Workshop on

- Ant Algorithms (ANTS'2002), vol. 2463 de Lecture Notes in Computer Science, Brussels, Belgium, Springer Verlag, pages 28–39, Septembre 2002.
- [170] N.Omont, La convergence des modularités structurelles et fonctionnelles des systèmes complexes, Thèse de Doctorat en informatique, Université d'Evry Val d'Essonne, Ecole doctorale des Génomes Aux Organismes, Programme Epigénomique, 2009.
- [171] M.Omran, Particle Swarm Optimization Methods for Pattern Recognition and Image Processing. PhD Thesis, Faculty of Engineering, Built Environment and Information Technology, University of Pretoria, 2005.
- [172] M.Omran, A.Engelbrecht, A.Salman, Dynamic Clustering using Particle Swarm Optimization with Application in Unsupervised Image Classification, ENFORMATIKA V9 ISSN 1305-5313, 2005.
- [173] S.Ouadfel, Contributions à la Segmentation d'images basées sur la résolution collective par colonies de fourmis artificielles, Thèse de Doctorat, Université Hadj Lakhdar de Batna, Faculté des Sciences de l'Ingénieur, 2006.
- [174] N.Otsu, A Threshold Selection Method from Gray-Level Histograms, IEEE Trans. on Syst., Man and Cyb.. 1979, Vol. 9, 1, pages 62-66, 1979.
- [175] N.Pal, S.Pal, A review on image segmentation techniques, Pattern Recognition 26, pages 1277–1294, 1993.
- [176] W.E.Paul. The immune system : an introduction. Fundamental Immunology, 3, 1993.
- [177] T.Pavlidis, Y.T.Liow, Integrating Region Growing and Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, n° 3, pages 225-233, 1990.
- [178] T.Pavlidis, Why progress in machine vision is so slow, Pattern Recognition Letters, vol. 13, pages 221-225, 1992.
- [179] F.Picarougne, N.Monmarché, M.Slimane, G.Venturini, C.Guinot, Two bio-inspired metaheuristics for information search on the web. In Proceedings of the 6th International Conference on Artificial Evolution, pages 422–434, Marseille, France, 27-30, octobre 2003.
- [180] K.N.Plataniotis, A.N.Venetsanopoulos, Color image processing and applications, CRC Press, Springer-Verlag, Berlin, 355 p. 2000.
- [181] G.Proctor, C.Winter, Information flocking : Data visualization in virtual worlds using emergent behaviours. In J.-C. Heudin, editor, Proc. 1st Int. Conf. Virtual Worlds, VW, volume 1434, pages 168–176, Springer-Verlag, 1998.
- [182] L.C.Ramac, P.K.Varsheney, Image thresholding based on AliSilvey distance measures, Pattern Recognit. 30, pages 1162–1174, 1997.
- [183] V.Ramos,F.Almeida, Artificial Ant Colonies in Digital Image Habitats, in : Dorigo, M. et al (editors), Proceedings of ANTS'2000 - 2nd International Workshop on Ant Algorithms, pages 113-116, Brussels, Belgium,2000.
- [184] I.Rechenberg, Cybernetic Solution Path of an Experimental Problem, Royal Aircraft Establishment Library Translation, 1965.

- [185] I.Rechenberg, *Evolutions strategie : Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog, Stuttgart, 1973.
- [186] C.W.Reynolds, *Flocks, herds and schools : a distributed behavioral model*. *Computer Graphics*, 21(4), pages 25-34, 1987.
- [187] P.Riccardo, J.Kennedy, and T.Blackwell, *Particle swarm optimization*. *Swarm Intelligence*, 1(1) :33–57, 2007.
- [188] N.Richard, M.Dojat, C.Garbay, *Automated segmentation of human brain MR images using a multiagent approach*. *Artificial Intelligence in Medicine*, 30(2), pages 153–176, 2004.
- [189] V.Rodin, A.Benzinou, A.Guillaud, P.Ballet, F.Harrouet, J.Tisseau, J.Le Bihan, *An immune oriented multi-agent system for biological image processing*, *Pattern Recognition*, 37(4) : pages 631–645, 2004.
- [190] E.Saber, A.M.Tekalp, *Integration of color, edge, shape, and texture features for automatic region-based image annotation and retrieval*, *Journal of Electronic Imaging*, vol. 7, n° 3, pages 684-700, 1998.
- [191] P.K.Sahoo, S.Soltani, A.K.C.Wong, *A survey of thresholding techniques*, *Computer Vision, Graphics, and Image Processing.*, Vol. 41, pages 233-260, 1988.
- [192] M.Salotti, C.Garbay, *Cooperation between edge detection and region growing : the problem of control*, In Vernazza, Venetsanopoulos, and Braccini, editors, *Image Processing, Theory and Applications*, pages 95-98 . Elsevier, 1993.
- [193] J.M.Salotti, *Gestion des informations dans les premières étapes de la vision par ordinateur*, Thèse de Doctorat, Institut national polytechnique de Grenoble, 1994.
- [194] I.Sebari, D.C.He, *Les approches de segmentation d'image par coopération régions- contours*, *Revue Télédétection*, vol. 7, n° 1-2-3-4, pages 499-506, 2007.
- [195] A.B.Sendova-Franks, N. R.Franks, *Social resilience in individual worker ants and its role in division of labour*. *Philosophical Transactions of the Royal Society of London B*, 256, pages 305–309, 2004.
- [196] M.Sezgin, R.Tasaltin, *A new dichotomization technique to multilevel thresholding devoted to inspection applications*, *Pattern Recognit. Lett.* 21, pages 151–161, 2000.
- [197] M.Sezgin, B.Sankur, *Survey over image thresholding techniques and quantitative performance evaluation*. *J.of Electronic Imaging*, Vol.13, 1, pages 146-165, 2004.
- [198] Y.Shi, R.C.Eberhart, *A modified particle swarm optimizer*, *IEEE International Conference on Evolutionary Computation*, Piscataway, NJ : IEEE Press, pages 69-73, 1998.
- [199] Y.Shi, R.C.Eberhart, *Empirical study of particle swarm optimization*, In : *Proc. Congress on Evolutionary Computation*, pages 1945-1950, Piscataway, IEEE Service Center, 1999.
- [200] P.Siarry (coordinateur), *Optimisation en traitement du signal et de l'image*. Hermes, 2007.

- [201] O.Simonin, Construction of numerical potential fields with reactive agents, In AAMAS'05 : Proceedings of the fourth international joint conference on Autonomous agents and multi-agent systems, pages 1351–1352, NewYork, NY, USA, 2005.
- [202] P.Suganthan, Particle Swarm Optimizer with Neighborhood Operator. In IEEE Congress on Evolutionary Computation, volume III, pages 1958–1961, 1999.
- [203] A.Somayaji, S.Hofmeyr, S.Forrest, Principles of a computer immune system, New Security Paradigms Workshop, 1997.
- [204] C.Spinu, Une approche multi-agents pour la segmentation d'images associant estimation et évaluation. Thèse de Doctorat. Grenoble : Université Joseph Fourier, 1997.
- [205] T.Stützle, H.H.Hoos, The MAX-MIN Ant System and local Search for Combinatorial Optimization Problems : Towards Adaptive Tools for Global Optimization. 2nd Metaheuristics International Conference (MIC-97), Sophia-Antipolis, France, pages 21-24, 1997.
- [206] T.Stutzle, M.Dorigo, ACO Algorithms for the Traveling Salesman Problem. In K.Miettinen, M.Makela, P.Neittaanmaki, J.Periaux, editors, Evolutionary Algorithms in Engineering and Computer Science, Wiley, 1999.
- [207] T.Stutzle, H.H.Hoos, MAX-MIN ant system, Future Generation Computer Systems Journal, 16(8), pages 889-914, 2000.
- [208] W.B.Tao, J.W.Tian, J.Liu, Image segmentation by three-level thresholding based on maximum fuzzy entropy and genetic algorithm, Pattern Recognit. Lett. 24, pages 3069–3078, 2003.
- [209] G.Théraulaz, E.Bonabeau, Coordination in distributed building, Science, 269, pages 686-688, 1995.
- [210] A.N.Tikhonov, V.Y.Arsenin, Méthodes de résolution de problèmes mal posés, MIR, Moscou, 1974.
- [211] J.E.Treherne, W.A.Foster, Group transmission of predator avoidance behaviour in a marine insect : the Trafalgar effect, Animal Behaviour, 28 : pages 1119-1122, 1981.
- [212] M.D.Toksari, Ant colony optimization for finding the global minimum, Applied Mathematics and Computation, 2005.
- [213] J.Tou, DYNOC- A Dynamic Optimal Cluster-seeking Technique. International Journal of Computer and Information sciences, vol.8, no.6, pp. 541-547, 1979.
- [214] R.H.Turi Clustering-Based Color Image Segmentation, Phd Thesis. Monash University, Australia 2001.
- [215] A.M.Turing, Computing machinery and intelligence, Report to the National Physical Lab, 1947.
- [216] T.Tsuji, Y.Tanaka, P.Morasso, V.Sanguineti, M.Kaneko, Bio-mimetic trajectory generation of robots via artificial potential field with time base generator, IEEE Transactions on Systems, Man, and Cybernetics, PartC, 32(4) :426–439, 2002.

- [217] F.Van den Bergh, An Analysis of Particle Swarm Optimizers. PhD thesis, Department of Computer Science, University of Pretoria, 2002.
- [218] H.Weimerskirch, J.Martin, Y.Clerquin, P.Alexandre, S.Jiraskova, Energy saving in flight formation - pelicans flying in a "V" can glide for extended periods using the other birds' air streams, *Nature*. 413 : pages 697-698, 2001.
- [219] N.Wiener, *Cybernetics : Or the Control and Communication in the Animal and the Machine*. Cambridge, MA : MIT Press, 1965.
- [220] E.Wilson, The relation between Caste Ratios and Division of Labour in the Ant Genus *Pheidole* (Hymenoptera : Formicidae), *Behav. Ecol. Sociobiol.*, vol. 16, pages 89–98, 1984.
- [221] B.F.Wu, Y.L.Chen, C.C.Chiu, Recursive algorithms for image Segmentation based on a discriminant criterion, *Int. J. Signal Process.* 1, pages 55–60, 2004.
- [222] B.Wrobel, O.Monga, Segmentation d'images naturelles : coopération entre un détecteur-contours et un détecteur-région . In 11^e Colloque « Traitement du Signal et des images » (GRETSI), pages 539-542, Nice (France), 1987.
- [223] Y.Xiaohan, J.Yla-Jaaski, O.Huttunen, T.Vehkomaki, O.Sipild, T.Katila, Image segmentation combining region growing and edge detection. In *International Conference on Pattern Recognition*. Volume C., The Hague, Netherlands, p.p 481– 484, 1992.
- [224] Z.H.Yang, Z.B.Pu, Z.Q.Qi, Relative entropy multilevel thresholding method based on genetic algorithm, in : *IEEE International Conference on Neural Networks and Signal Processing*, Nanjing, China, pages 583–586, 2003.
- [225] J.C.Yen, F.J.Chang, S.Chang, A new criterion for automatic multi-level thresholding, *IEEE Trans. Image Process.*IP-4, pages 370–378, 1995.
- [226] P.Y.Yin, L.H.Chen, A fast iterative scheme for multilevel thresholding methods, *Signal Process.* 60, pages 305–313, 1997.
- [227] P.Y.Yin, A fast scheme for optimal thresholding using genetic algorithms, *Signal Process.* 72, pages 85–95, 1999.
- [228] H.Yourui, W.Shuang, Multilevel thresholding methods for image segmentation with Otsu based on QPSO, *Congress on Image and Signal Processing*, pages 701-705, 2008.
- [229] E.Zahara, S.K.S.Fan, D.M.Tsai, Optimal multithresholding using a hybrid optimization approach, *Pattern Recognition Letters* - 26, pages :1082–1095, 2005.
- [230] D.Zugaj, V.Lattuati, A new approach of color images segmentation based on fusing region and edge segmentations outputs. *Pattern Recognition*, vol. 31, n° 2, pages 105-113, 1998.

Annexe A

Contributions

Notre recherche dans ce domaine nous a permis de publier un certain nombre de communications. Donnons les publications réalisées à l'issue de ce travail.

A.1 Publications

Cette annexe liste toutes les publications résultant du travail menant à cette thèse.

A.1.1 Conférences avec comité de lecture

1. L.Djerou, M.Batouche, Segmentation des images par émergence, 49-56, VIVATI2003', Jijel, Algérie, 2003
2. L.Djerou, M.Batouche, La construction collective des régions homogènes dans des images à niveau de gris, PP-265-270, TAIMA2003, Hammamet, Tunisie, ISBN :9973-51-138-7, 2003.
3. L.Djerou, M.Batouche, A.Fareh, M.Ouamane, L'optimisation par essaim de particules Application au problème du voyageur de commerce COSI'05, Bejaia, Algérie, 2005.
4. L.Djerou, M.Batouche, M.Batouche, La vie artificielle pour la segmentation d'images, ISBN : 9973-51-546-3, Sousse, Tunisie,2006.
5. N.Khelil, L.Djerou, A.Zerarka, Toward the best points of interpolation using the particles swarm optimization approach, ROADEF'06, Lille, France, 2006.
6. N.Khelil, L.Djerou, A.Zerarka, M.Batouche, Vers les meilleurs points d'interpolation en utilisant l'approche PSO. Meta'06, Hammamat ; Tunisie, 2006.
7. L.Djerou , N.Khelil , M.Batouche, Auto-reconfiguration d'un robot modulaire Meta'06, Hammamet, Tunisie, 2006.
8. L.Djerou , N.Khelil , M.Batouche, Une approche adaptative pour la segmentation d'image, Traitement et Analyse de l'information : Méthodes et Applications. Hammamet du 21 au 26 mai 2007. ISBN : 978-9973-61-802-3 Pages : 211- 216, 2007

9. L.Djerou , N.Khelil , M.Batouche, Une approche incrémentale et adaptative pour la segmentation d'image, la première Journée Nationale sur les Applications des Métaheuristiques (JNAM'07), 29 mai 2007, USTHB, Bab Ezzouar (Alger), 2007.
10. L.Djerou, N.Khelil, M.Batouche, A biologically inspired approach for image segmentation , In the proceedings of MCSEAI, pp.296-301, Oran April 28-30, 2008.
11. N.Khelil, L.Djerou, A.Zerarka, M.Batouche, On searching the best points of interpolation using swarm optimisation techniques , In the proceedings of MCSEAI, pp.472-476, Oran April 28-30, 2008
12. L.Djerou. N.Khelil, M.Batouche, From termite construction to region growing, in the proceedings of the International Conference on Metaheuristics and Nature Inspired Computing - META'08, Hammamet, Tunisia, October 29-31 2008.
13. L.Djerou. N.Khelil, A.Zerarka, M.Batouche, Numerical Integration using Particle Swarm Optimization, in the proceedings of the International Conference on Metaheuristics and Nature Inspired Computing - META'08, October 29-31 2008, Hammamet, Tunisia, October 29-31 2008.
14. N.Khelil, L.Djerou, A. Zerarka, M.Batouche, Improvement of Gregory's formula using Particle Swarm Optimization, in the proceeding of International Conference on Computer and Applied Mathematics, pp.940-942, volume 58, Venice, Italy, October 2009
15. L.Djerou, H.Dehipi, N.Khelil, M.Batouche, Self- adaptative thresholding for image segmentation, 5ème SYMPOSIUM INTERNATIONAL Images Multimédias Applications Graphiques et Environnements, Biskra, Algérie 03, 04 - 05 Novembre 2009.

A.1.2 Articles avec comité de lecture

16. L.Djerou, M.Batouche, Résolution collective du problème du segmentation, courrier du savoir- N°04, PP-89-94, ISSN-1112-3338, JUIN 2003.
17. L.Djerou, M.Batouche, Une approche biomimétique pour la segmentation d'image, courrier du savoir- N°09, ISSN-1112-3338, SEPT 2008.

A.1.3 Articles IEEE

18. L.Djerou, M.Batouche, N.Khelil, A.Zerarka, Towards the Best Points of Interpolation Using Particles Swarm Optimisation Approach, IEEE Congress on Evolutionary Computation. Singapore September 25- 28, 2007.
19. L.Djerou, N.Khelil, M.Batouche, Image Segmentation by Self-organizing Region Growing, In the proceedings of CISIM-08, pp.171-176, IEEE press, Ostrava, The Czech Republic, June 26 - June 28, 2008.
20. L.Djerou, H.Dehipi, N.Khelil, M. Batouche, Using the BPSO Algorithm in Image Segmentation for Dynamic Thresholding, in the proceeding of the International Conference on Bio-Inspired Computing : Theories and Applications pp.402-407, IEEE press, Beijing, China (BIC-TA 2009).

21. L.Djerou, H.Dehipi, N. Khelil, M.Batouche, Automatic Multilevel Thresholding using Binary Particle Swarm Optimization for image segmentation, in the proceeding of the 2009 International Conference of Soft Computing and Pattern Recognition, pp.67-71, IEEE Computer society, Malaysia, 2009.

A.1.4 Monographie et livres

22. Publication d'un polycopié intitulé " Introduction à la programmation en Fortran".
23. L.Djerou. N.Khelil, A. Zerarka, M.Batouche, Searching for the best Points of interpolation using swarm intelligence techniques, chapter book in the book 'Particle swarm optimization' ISBN 978-953-7619-48-0. Itech Education and Publishing, Vienna, Austria, 2009.