

**République Algérienne Démocratique et Populaire**

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

**Université Hadj Lakhdar de Batna**

**Faculté des Sciences de l'Ingénieur**

# **Thèse**

*En vue de l'obtention du diplôme de*

**Doctorat en Informatique**

**Contributions à la Segmentation  
d'images basées sur la résolution  
collective par colonies de fourmis  
artificielles**

*Présentée le 04/07/ 2006*

*par*

**OUADFEL SALIMA**

Devant le jury composé de :

<i>Président</i>	N. Bouguechal	Prof., Université de Batna
<i>Examineurs</i>	L. Hamami	MC., l'E .N.P.A
	A. Moussaoui	MC., Université de Setif
	A. Zidani	MC., Université de Batna
<i>Rapporteur</i>	M. Batouche	Prof., Université de Constantine



## Remerciements

Je remercie Mr BATOUCHE Mohamed, Professeur à l'Université de Constantine qui a encadré mes travaux durant ces années de thèse. Par ses conseils, son dévouement constant et ses nombreuses discussions, il a permis à ce travail d'être ce qu'il est aujourd'hui. Je le remercie également pour la liberté et la confiance qu'il m'a toujours accordé.

Je remercie MME MESHOUH Sihem, Maître de conférence à l'Université de Constantine pour l'attention et l'amitié qu'elle m'a apporté durant toutes ces années ainsi que pour sa lecture et ces commentaires qui m'ont permis d'améliorer ce manuscrit..

Je remercie Mr BOUGHECHAL Noureddine, Professeur à l'Université de Batna pour avoir accepté de présider le jury de cette thèse.

Je remercie Mme HAMAMI Latifa, Maître de conférence à l'E. N. P .A, Mr MOUSSAOUI Abdelouaheb, Maître de conférence à l'université de Sétif et Mr ZIDANI Madjid, Maître de conférence à l'université de Batna pour avoir accepté de participer à ce jury et pour l'intérêt qu'ils ont manifesté envers ce travail.

Merci également à ma famille et en particulier mes parents, pour l'encouragement constant et leur aide moral sans retenue tout au long de mes études.

Je remercie mon mari pour le soutien et ses encouragements pour la finalisation de ce travail.

## Résumé

Le travail présenté dans cette thèse décrit une nouvelle approche pour la segmentation d'images. Cette approche s'inspire des comportements collectifs et auto-organisés des fourmis dans la nature. Elle se base sur une population de fourmis artificielles simples capables de s'auto-organiser pour faire émerger une segmentation optimale.

Dans un premier temps, on s'est inspiré du comportement collectif de tri de couvain observé chez les fourmis pour développer une méthode de classification non supervisée. Les pixels de l'image sont initialement placés sur un tableau de cases représentant l'environnement des fourmis. Selon une fonction de similarité locale, les fourmis vont déplacer les pixels d'une case à une autre dans le but d'obtenir des classes homogènes et bien séparées. La méthode testée sur des images synthétiques et réelles a montré son efficacité et sa capacité à extraire un nombre correct de classes avec une partition de bonne qualité en comparaison à l'algorithme classique Kmeans.

Dans un second temps, le comportement collectif de recherche de nourriture a été utilisé pour la segmentation d'images basée sur les champs de Markov. L'image est segmentée en maximisant la probabilité a posteriori (MAP) utilisant des algorithmes inspirés de la métaheuristique « Optimisation par les Colonies de Fourmis ». Une colonie de fourmis artificielles construit de nouvelles partitions selon un processus itératif et stochastique en utilisant une information heuristique et des traces de phéromones artificielles. Les nouvelles partitions sont ensuite optimisées en utilisant un algorithme de recherche local. Une étape de diversification est aussi introduite afin de diversifier la recherche. Les résultats obtenus donnent de bons résultats comparés à ceux obtenus par d'autres méthodes d'optimisation.

**Mots clés** : Segmentation, Classification non supervisée, champ de Markov, Optimisation, fourmis artificielles.

## Abstract

The work described a new approach for image segmentation. This approach is inspired from some swarm and auto-organized compartments of real ants in the nature. It is based on a population of artificial and simple ants capable of auto-organizing in order to emerge an optimal segmentation.

In the first part of the work, we have used the self-organizing and autonomous brood sorting behavior observed in real ants for unsupervised image classification. Ants move on a discrete array of cells representing the ants' environment. Ants may move pixels that are scattered within the cells of the array according to a local similarity function. In this way, ants form clusters without the initial knowledge of the number of clusters and initial partition. Experimental results on synthetic and real images demonstrate the ability of the method to extract the correct number of clusters with good clustering quality compared to the results obtained from the classical clustering algorithm Kmeans.

In the second part, the food hunting compartment has been used for image segmentation based on the Markov Random Field. The image is segmented by maximizing the a posteriori probability (MAP) using algorithms inspired from the ant colony optimization metaheuristic. A colony of artificial ants build new partitions within a stochastic iterative process, by adding solution components to partial solutions using a combination of heuristic information and an artificial pheromone trail. The new partitions found by ants are then optimized using a local search algorithm. Pheromone trails are reinforced according to the quality of the partitions found by the best ant. A diversification phase is also used to diversify the search. The experimental results presented outperform those obtained with other optimization methods.

**Keywords:** Segmentation, Unsupervised Classification, Markov Random Field, Optimization, Artificial ants.

## Liste des figures

Figure 1.1 .	Niveaux de la vision par ordinateur .....	14
Figure 2.1.	Le proto-émergentisme (en haut) et le néo-émergentisme (en bas) .....	24
Figure 2.2.	Codage de l'information.....	33
Figure 2.3.	Modèle de la roulette.....	33
Figure 2.4.	Opérateur de croisement monopoint appliqué à deux chromosomes codés sur 8 bits.	34
Figure 2.5.	Opérateur de mutation appliqué à deux chromosomes codés sur 8 bits.....	34
Figure 2.6.	Structure générale d'un algorithme génétique. ....	35
Figure 2.7.	Structure d'un neurone biologique .....	37
Figure 2.8.	le neurone formel de Mac Culloch et Pitts .....	38
Figure 2.9.	Quelques topologie de réseau de neurones .....	38
Figure 2.10.	Quelques types de fonction d'activation .....	38
Figure 2.11.	limitation du Perceptron. ....	40
Figure 2.12.	Les boids de Reynolds [Reynolds 2001].....	41
Figure 2.13.	Le voisinage local d'un boid. ....	41
Figure 2.14.	Les trois règles trouvées par Reynolds pour programmer des mouvements collectifs complexes. ....	42
Figure 2.15.	Règles simples d'application locale utilisées dans un essaim de particules. ...	42
Figure 3.1.	Anatomie d'une fourmi .....	47
Figure 3.2.	Pont Binaire de Deneubourg .....	52
Figure 3.3.	Les fourmis suivent indifféremment les deux branches du pont.....	52
Figure 3.4.	Les fourmis ont tendance à emprunter le même chemin (celui d'en bas).....	53
Figure 3.5.	Expérience du pont avec des branches de taille différentes .....	53
Figure 3.6.	Effet de dépôt d'un obstacle entre la source de nourriture et le nid.....	54
Figure 3.7.	Grille de classification de Lumer et Faieta (d'après Labroche, 2003] .....	64
Figure 4.1.	Schéma d'un système de vision .....	73
Figure 4.2.	Exemple de segmentation d'images .....	75
Figure 4.3.	Différents types de contours.....	75
Figure 4.4.	Dérivées première et seconde d'un contour de type "saut d'amplitude" .....	76
Figure 4.5.	Quad Tree d'une image 8*8. ....	83
Figure 4.6.	Diagramme de Voronoi.....	83
Figure 4.7.	La segmentation par division-Fusion. ....	84
Figure 4.8.	Taxonomie de la classification inspirée de celle de Jain et al.....	88
Figure 4.9.	Exemples d'histogrammes. (a) Histogramme bimodal. (b) Histogramme multi-modal	92
Figure 4.10.	La distance entre un pixel $x$ et le plus proche pixel de $R_1$ .....	109
Figure 4.11.	types de pixels classés .....	110
Figure 5.1.	L'environnement des fourmis artificielles .....	121
Figure 5.2.	Les images synthétiques de test .....	126
Figure 5.3.	Les images réelles de test .....	126
Figure 5.4.	Evolution du nombre de classes au cours des itérations de AntClust pour l'image synthétique 1. ....	130
Figure 5.5.	Evolution de la distance moyenne des pixels au cours des itérations de AntClust pour l'image synthétique 1. ....	130

Figure 5.6.	Evolution de l'index de Rand sur l'image synthétique1.....	131
Figure 5.7.	Exemples d'images segmentées par l'algorithme AntClust.....	132
Figure 6.1.	.....	121
Figure 6.2.	Représentation graphique du problème de segmentation.....	142
Figure 6.3.	Structure de la phéromone artificielle .....	143
Figure 6.4.	.....	148
Figure 6.5.	.....	148
Figure 6.6.	Structure de la phéromone artificielle .....	149
Figure 6.7.	Image synthétique. A) image originale, B) image bruitée SNB=10DB.....	154
Figure 6.8.	Images IRM Fantômes .....	154
Figure 6.9.	Images réelles: a) image de muscle , b) image de maison .....	154
Figure 6.10.	Graphe de convergence des algorithmes d'optimisation ACS et ICM : (a) utilisation de ACS sans recherche locale, (b) utilisation de ACS avec recherche locale. 157	
Figure 6.11.	Evolution de la qualité de la fonction objectif en utilisation les trois algorithmes de fourmis ACS ,MMAS et HASEG.....	158

## Liste des tableaux



## Liste des algorithmes

## TABLE DES MATIERES

Chapitre 1. introduction générale.....	<b>13</b>
1.1 Contexte de l'étude : la vision par ordinateur .....	13
1.2 La segmentation d'images.....	15
1.3 Motivations.....	16
1.4 Résolution de problèmes par émergence.....	17
1.5 Contributions.....	19
1.6 Structure et contenu de la thèse.....	21
<b>Chapitre 2. Le phénomène d'émergence et l'informatique.....</b>	<b>24</b>
2.1 <i>Introduction</i> .....	24
2.2 Caractérisation d'un phénomène émergent.....	25
2.3 L'auto-organisation comme technique d'émergence .....	26
2.3.1 Définitions.....	27
2.3.2 Les mécanismes de l'auto-organisation .....	28
2.4 L'intelligence en essaim.....	30
2.5 Emergence et systèmes artificiels collectifs.....	30
2.5.1 Algorithmes évolutionnaires et algorithmes génétiques .....	31
2.5.2 Les réseaux de neurones.....	36
2.5.3 Algorithmes à essaims de particules .....	40
2.5.4 Les algorithmes de fourmis artificielles .....	42
2.6 Conclusion.....	44
<b>Chapitre 3. Les Fourmis artificielles .....</b>	<b>46</b>
3.1 Introduction .....	46
3.2 Généralités sur les fourmis.....	46
3.3 Quelques concepts de base.....	50
3.4 Les algorithmes de fourmis artificielles pour l'optimisation combinatoire .....	51
3.4.1 Inspiration biologique : le fourragement collectif par stigmergie.....	51
3.4.2 Les expériences .....	52
3.4.3 Modèles de fourragement pour la résolution des problèmes d'optimisation ...	54
3.4.4 Autres domaines d'application.....	58
3.4.5 La métaheuristique « optimisation par les colonies de fourmis ».....	59
3.5 Les algorithmes de fourmis artificielles pour la classification automatique.....	62
3.5.1 Inspiration biologique .....	62
3.5.2 Modèles du tri du couvain et application à la classification de données.....	62
3.5.3 Autres domaines d'application.....	67
3.5.4 Autres sources d'inspiration.....	67
3.6 Conclusion.....	68
<b>Chapitre 4. La segmentation d'images.....</b>	<b>71</b>
4.1 Introduction .....	71
4.2 La vision artificielle .....	71
4.3 Architecture typique d'un système de vision .....	72
4.4 Techniques de segmentation .....	74
4.4.1 Détection de contours.....	75
4.4.2 Détection de régions homogènes.....	78

4.4.3	Les méthodes de classification .....	87
4.4.4	Les méthodes de segmentation biomimétiques .....	99
4.5	Méthodes d'évaluation des résultats de segmentation en régions .....	106
4.5.1	Evaluation par comparaison avec une segmentation de référence .....	107
4.5.2	L'évaluation se référant à l'image originale .....	111
4.6	Conclusion .....	116
<b>Chapitre 5. Des fourmis pour la classification automatique des images.....</b>		<b>119</b>
5.1	Introduction .....	119
5.2	L'algorithme AntClust .....	120
5.2.2	L'environnement des fourmis .....	120
5.2.3	Les fourmis et leurs comportements .....	121
5.3	Etude expérimentale .....	126
5.3.1	Les images de test .....	126
5.3.3	Les mesures d'évaluation .....	127
5.3.4	Comparaison des résultats .....	128
5.3.5	Les résultats .....	128
<u>5.3.6.</u>	Conclusion .....	132
<b>Chapitre 6. Résolution collective de la segmentation par relaxation Markovienne.....</b>		<b>135</b>
6.1	Introduction .....	135
6.2	Motivations.....	135
6.3	Segmentation d'images par classification markovienne .....	136
6.3.1	Modélisation des observations .....	137
6.3.2	Modélisation des connaissances a priori .....	138
6.3.3	L'énergie totale .....	139
6.4	Transposition de OCF à la segmentation d'images par relaxation Markovienne ..	140
6.4.1	Le modèle de représentation .....	140
6.4.2	Comportement individuel de la fourmi .....	141
6.4.3	Le modèle de coopération entre les fourmis .....	141
6.4.4	La construction de partitions .....	142
6.5	Résultats expérimentaux .....	153
6.5.1	Etude de la convergence.....	157
6.5.2	Etude de la robustesse .....	158
6.6	Conclusion.....	159
<b>Chapitre 7. Conclusion générale.....</b>		<b>161</b>
	Perspectives.....	162
<b>Annexe A .....</b>		<b>164</b>
<b>Modélisation Markovienne pour la segmentation d'images .....</b>		<b>164</b>
<b>Annexe B .....</b>		<b>164</b>
	Nos publication.....	168
<b>Bibliographie.....</b>		<b>175</b>

# Chapitre 1

## Introduction générale

<b>Chapitre 1. Introduction Générale .....</b>	<b>13</b>
1.1 Contexte de l'étude : la vision par ordinateur .....	13
1.2 La segmentation d'images.....	15
1.3 Motivations.....	16
1.4 Résolution de problèmes par émergence.....	17
1.5 Contributions .....	19
1.6 Structure et contenu de la thèse.....	21

---

## Chapitre 1. Introduction Générale

---

### ***1.1 Contexte de l'étude : la vision par ordinateur***

Depuis très longtemps, les chercheurs ont été fascinés par la capacité du système de vision humain à percevoir l'espace qui l'entoure sans aucune difficulté. Avec la naissance de machines de plus en plus puissantes et les progrès techniques au niveau du traitement automatique des images, une nouvelle discipline est apparue sous le nom : « *vision par ordinateur* ». Ce nouveau paradigme avait pour ambition de réaliser un système de vision artificiel capable de reproduire certaines fonctionnalités de la vision humaine au travers l'analyse des images de la scène captée par un dispositif d'acquisition. Pour réaliser un tel système, le traitement visuel a été formalisé d'une manière scientifique ce qui a donné naissance la *théorie de vision*.

Cette théorie a été résumée par Radu Horaud et Olivier Monga [Horaud, 1995] de la manière suivante : « *La vision est un processus de traitement de l'information. Elle utilise des stratégies bien définies afin d'atteindre ses buts. L'entrée d'un système de vision est constituée par une séquence d'images. Le système lui-même apporte un certain nombre de connaissances qui interviennent à tous les niveaux.* »

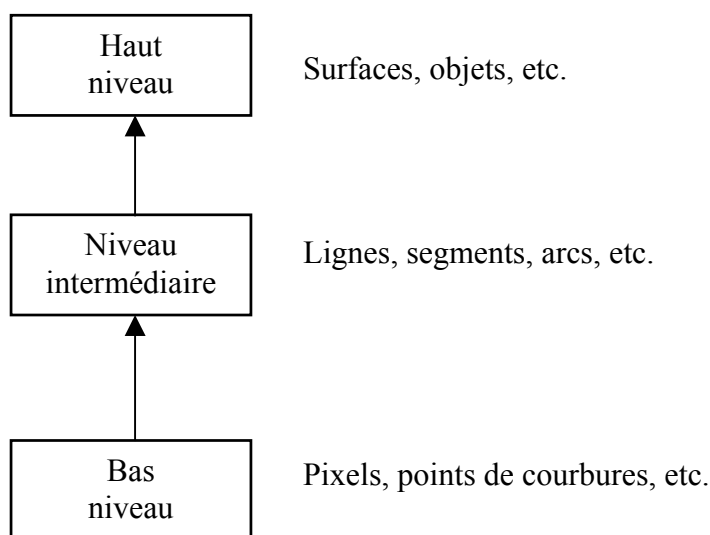
David Marr fut le premier à rechercher une formalisation scientifique du problème de vision. Il proposa à la fin des années 70, un paradigme de la vision définit de la façon suivante[Marr, 1980] :

- La reconnaissance des contours des objets permet de les délimiter et de les structurer.
- L'accès à la troisième dimension se fait à partir d'images en deux dimensions (appelées ébauche 2,5D) et des connaissances sur le monde 3D.
- L'extraction du relief à partir de l'ombre portée.
- La perception du mouvement dans des scènes animées.

Ses travaux ont permis d'engendrer un grand nombre d'application en donnant une formulation algorithmique aux différents points de son paradigme.

Une partie des traductions algorithmiques du paradigme de David Marr, qui a débouché sur des algorithmes efficaces, est organisée en trois grands niveaux tel qu'illustré à la figure 1.1 :

- Le bas-niveau : Il concerne le problème de description scientifique de la grande quantité d'information contenue dans l'image. Les traitements du bas niveau travaillent directement sur les caractéristiques attachées aux pixels de l'image et nécessitent très peu d'informations sur son contenu. Il s'agit ici des processus d'acquisition, de filtrage, d'amélioration et de segmentation d'images.
- Le niveau intermédiaire : l'ensemble des indices visuels issus du bas-niveau sont regroupés au niveau intermédiaire pour former des structures représentant des entités présentes dans l'image (ligne, objet, etc.).
- Le haut-niveau : Il s'agit des processus de description des objets qui permettent l'identification de ses derniers et l'interprétation de la scène observée.



**Figure 1.1 . Niveaux de la vision par ordinateur**

Ces niveaux sont parcourus selon une approche ascendante, où les structures obtenues sont de plus en plus complexes selon le niveau de représentation utilisé.

Le cadre général dans lequel s'inscrit cette thèse est celui de la segmentation d'images. Cette opération a pour objectif d'extraire à partir d'une image numérique des indices visuels ou des primitives pertinentes permettant de la représenter sous une forme plus condensée et facilement exploitable dont dépend la performance des systèmes de vision artificielle.

## 1.2 La segmentation d'images

La segmentation consiste à diviser l'ensemble de pixels en régions connexes, homogènes et bien séparées. Ces régions possèdent une certaine uniformité pour une ou plusieurs caractéristiques (intensité, couleur, texture, ...) et sont différentes pour au moins une de ses caractéristiques des régions voisines.

Formellement [Zucker, 1976], la segmentation d'une image numérique  $I$  consiste à chercher une partition de  $I$  en un sous-ensemble  $R = \{R_1, R_2, \dots, R_n\}$  telle que:

$$\forall i, R_i \neq \emptyset$$

$$\forall i \neq j, R_i \cap R_j = \emptyset$$

$$I = \cup_i R_i$$

Le problème de la segmentation d'images est alors caractérisé par :

1. Un ensemble de critères d'homogénéités des régions que l'on cherche ;
2. Un algorithme permettant l'obtention de régions optimisant au mieux ces critères.

Beaucoup de méthodes de segmentation existent dans la littérature et peuvent être séparées en trois grandes familles.

1. Les méthodes de segmentation par contours basées sur la recherche des discontinuités locales présentes dans l'image.
2. Les méthodes de segmentation en régions homogènes consistent à trouver des ensembles de pixels qui partagent des propriétés similaires. Les régions homogènes sont construites à partir des parties connexes de ses ensembles. On distingue deux grandes techniques de segmentation en régions :
  3. Les méthodes de classification qui fournissent une partition de l'image en regroupant des pixels ayant des niveaux de gris similaires dans une même classe de pixels. Les régions sont définies par les ensembles maximaux de pixels connexes appartenant à la même classe.

Bien que le problème de la segmentation d'images ait fait l'objet d'une vaste littérature [Pal 1993], ce problème est encore loin d'être résolu et la segmentation, telle qu'elle est définie n'est évidemment pas unique [Cocquerez 1995a]. Le choix d'une méthode est lié à 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. Ballard a exprimé cette difficulté comme suit: « *the most severe chicken-and-egg problem of static computer vision is the segmentation problem. It is difficult to separate figure from ground without recognizing objects, and it is difficult to recognize objects if they are not separated from the background and other objects* » [Ballard 1992]. De plus, elle n'est pas une fin en soi et de sa qualité dépend des résultats des traitements ultérieurs obtenus à partir des primitives extraites. Les régions extraites peuvent être présentées comme entrées pour un système de reconnaissance d'objets ou bien pour un processus de prise de décision.

### 1.3 Motivations

La segmentation est un problème mal posé au sens de Hadamard [Tikhonov, 1974], comme est le cas pour de nombreuses tâches de vision artificielle, 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 que de point de vue de choix de la technique de résolution et est confrontée au problème d'ambiguïté et de bruit qui affecte certains pixels. Ces difficultés expliquent le fait qu'il soit donc difficile de définir *une bonne segmentation* d'une image et que le problème de segmentation reste ouvert jusqu'à ce jour.

Ce travail de thèse est consacré aux méthodes de classification qui considèrent une classe comme un ensemble de pixels connexes ou non, possédant des intensités similaires. Le problème de partitionnement revient à chercher une partition qui regroupe d'une manière optimale les  $N$  pixels de l'image en  $K$  classes (clusters) 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. On peut essayer de résoudre ce problème par une méthode brute en engendrant toutes les partitions possibles et à retenir celle qui minimise au mieux le critère de partitionnement. Malheureusement, la taille de l'espace des partitions possibles est de l'ordre de  $O\left(\frac{N^k}{K!}\right)$ . La classification se ramène alors à un problème d'optimisation complexe NP-difficile pour lequel les méthodes locales s'avèrent très vite impraticables même pour une image de petite taille.

Les méthodes de partitionnement sont dites supervisées lorsque des informations a priori sont introduites dans le processus de construction de classes, et non supervisés quand aucune connaissance n'est disponible. L'inconvénient majeur des méthodes de partitionnement supervisées classique est qu'elles exigent de connaître au préalable le nombre possible de



classes de pixels ainsi qu'une bonne partition de départ dont dépend le résultat final de classification. Ces contraintes rendent l'utilisation de ces algorithmes peu intéressante quand on veut segmenter automatiquement une image.

Durant le processus de segmentation, certaines méthodes de classification privilégient les informations de luminance des pixels par rapport aux informations spatiales. Ces dernières ne sont prises en considération qu'à la fin du processus afin de construire des régions connexes. Ces méthodes sont faciles à implémenter mais lors de la classification, le contexte spatial est perdu. D'autres méthodes de classification tiennent compte de la topologie de l'image dès le début du processus d'étiquetage, et fournissent ainsi de bons résultats. Parmi ces techniques, on trouve les méthodes basées sur la théorie des champs de Markov. Dans le cadre de champs de Markov la probabilité qu'un pixel appartienne à une classe dépend non seulement de sa couleur, mais aussi de celles de ses voisins. La segmentation d'images associée se ramène alors à un problème d'optimisation. L'objectif est de chercher le minimum global d'une fonction d'énergie représentant les interactions locales entre pixels en terme de niveau de gris et en terme d'étiquettes. La résolution d'un tel problème s'avère difficile car d'une part la fonction d'énergie globale est loin d'être convexe, d'une autre part la minimisation doit se faire sur l'ensemble des configurations de toutes les étiquettes possibles dont le cardinal est très grand.

La conviction de l'auteur est que la résolution de tels problèmes nécessite l'utilisation d'approches stochastiques qui permettent d'explorer d'une manière judicieuse l'espace des solutions. Ce type de méthodes permet de rester piégé dans un optimum local et converge vers une solution optimale ou proche de l'optimale.

#### ***1.4 Résolution de problèmes par émergence***

Les méthodes classiques de résolution de problèmes consistent à décomposer le problème en sous problèmes et à définir dès le départ les étapes de résolution. Si auparavant cette méthode de résolution a donné de bons résultats, son efficacité a été remise en cause ces dernières années avec la complexité croissante des problèmes à résoudre et l'apparition de nouveaux besoins reflétant la nécessité de disposer de systèmes robustes et fiables dans des domaines dynamiques et incertains. Face à ces nouvelles difficultés, il est devenu nécessaire voire inévitable de chercher de nouvelles solutions en explorant de nouveaux paradigmes que ceux habituellement utilisés.

Depuis quelques années, les chercheurs informaticiens ont trouvé en le monde naturel, une source d'inspiration inépuisable pour la conception de nouveaux systèmes informatiques. Il s'agit de puiser dans les comportements des êtres naturels de nouvelles approches pour la

résolution de problèmes difficiles. Le rôle de l'informaticien est d'observer et comprendre les mécanismes et processus qui régissent les comportements dits « intelligents » de ces individus pour la résolution des problèmes courants, puis extraire à partir de ces études des modèles implantables sur des machines dont les résultats pourront être validés par rapport à ceux observés dans la nature.

En Biologie par exemple, de nombreux systèmes naturels composés d'individus autonomes exhibent des aptitudes à effectuer des tâches qualifiées de complexes sans contrôle global. De plus, ils peuvent s'adapter à leur milieu soit pour y survivre, soit pour améliorer le fonctionnement du collectif. C'est le cas des colonies d'insectes sociaux [Camazine, 2002] tels que les termites, fourmis [Bonabeau, 1997], ou araignées [Bourjot, 1999] qui font effectivement preuve de remarquables capacités pour effectuer des tâches telles que : la construction de nids complexes, la construction de pont, la recherche efficace de ressources, la capture de proies... L'étude des déplacements collectifs de vols d'oiseaux migrateurs ou de bancs de poissons montre également le fait que la tâche collective est le résultat des interactions des individus autonomes [Theraulaz, 1997]. Le fonctionnement du système immunitaire est lui aussi représentatif du fonctionnement d'un système complexe composé d'un ensemble d'agents autonomes.

Tous ces systèmes naturels présentent un point commun : l'émergence d'un comportement global collectif et complexe à partir de simples interactions entre des insectes simples dotés d'une intelligence très réduite et ne possédant qu'une vision très partielle de leur environnement [Bonabeau, 1999]. Ce comportement émergent leur permet de résoudre collectivement des problèmes très complexes

Il paraît donc légitime et inévitable d'étudier ce phénomène d'émergence afin de pouvoir en comprendre le fonctionnement et être capable de l'utiliser comme nouvelle approche pour la conception de systèmes artificiels. On parle alors d'intelligence artificielle en essaim pour désigner une telle approche.

Parmi ces systèmes, on trouve les méthodes d'optimisation par essaim particulières inspirés de l'étude de l'organisation de groupes d'animaux, [Eberhart, 2001], les algorithmes inspirés des essaims d'insectes volants [Aupetit, 2003], et les systèmes de fourmis artificielles largement utilisés ces dernières années pour résoudre des problèmes de classification ou d'optimisation [Dorigo, 1991]. Il s'agit d'une nouvelle approche qui s'intéresse aux comportements individuels des fourmis réelles, aux interactions entre ces entités autonomes et à l'émergence au niveau supérieur de l'ensemble du système de comportements complexes pouvant être qualifié d'intelligent.

Le problème de segmentation étant par sa nature un problème d'optimisation complexe difficile à résoudre, nous nous proposons dans ce travail de thèse de puiser dans l'étude des comportements des fourmis de nouvelles méthodes de segmentation flexibles et fortement distribuées qui permettent la formation des primitives d'une manière coopération, collective et guidée.

### **1.5 Contributions**

Cette thèse part du constat simple que les colonies de fourmis résolvent des problèmes complexes, bien que l'intelligence d'une fourmi soit limitée. Les travaux menés dans le cadre de cette thèse portent sur différents aspects :

- L'étude de la capacité des entités-fourmis simples non intelligentes individuellement de s'auto-organiser pour créer un tout dont le comportement peut être qualifié d'intelligent.
- L'étude du problème de la segmentation et l'examen des principales méthodes afin de dresser un état de l'art dans ce domaine ;
- Concevoir de nouvelles méthodes de segmentation basées sur une population de fourmis autonomes non intelligentes capables de s'auto-organiser pour créer un comportement global complexe et faire émerger collectivement une segmentation optimale de l'image par l'intermédiaire des interactions qu'ils entretiennent avec leur environnement.

Deux approches sont possibles pour développer de nouvelles méthodes basées fourmis comme on le verra dans cette thèse. La première approche modélise la faculté des fourmis à trier collectivement leur couvain ou à construire des cimetières. Ce modèle est relativement simple et est très stimulant pour résoudre le problème du partitionnement. Les algorithmes à base de ce modèle ont été proposés pour la première fois par Deneubourg et son équipe [Deneubourg, 1990] et ont été adaptés par la suite au problème de la classification non supervisée par d'autres chercheurs dans différents domaines.

La deuxième approche consiste à s'inspirer du comportement de fourrage des fourmis et leur capacité à s'auto-organiser pour trouver le plus court chemin entre le nid et la source de nourriture. En effet via les concepts de mémoire collective représentée par la phéromone, de diversification et d'intensification, les fourmis arrivent à trouver une solution optimale sans aucune supervision. Les algorithmes de fourmis à base de phéromones ont été développés à l'origine par Dorigo et ses collègues [Dorigo, 1991] suite aux travaux de Deneubourg et

sont utilisés depuis avec succès pour la résolution des problèmes d'optimisation combinatoire. Ces algorithmes ont été regroupés sous le nom générique de « Optimisation par Colonies de Fourmis » (OCF)..

Nous proposons en suivant la première approche dans le chapitre 5 AntClust, un nouvel algorithme de classification non supervisée. Dans AntClust, on cherche à déterminer automatiquement les classes présentes dans l'image sans connaissance a priori sur leur nombre et sans qu'il soit nécessaire d'effectuer des paramétrages complexes. Les pixels sont placés initialement sur un tableau de cases représentant l'environnement des fourmis artificielles. Chaque fourmi peut déplacer d'une manière probabiliste les pixels d'une case à une autre en utilisant une fonction de similarité locale, de manière à regrouper les pixels d'intensités similaires dans une même case représentant une seule classe.

La seconde approche a permis de concevoir différents algorithmes de classification qui prennent en compte la disposition spatiale des pixels en plus de leurs intensités durant le processus de partitionnement. Ces algorithmes utilisent le principe de l'exploration stochastique par une colonie de fourmis combiné avec la théorie des champs de Markov pour la modélisation du champ des étiquettes et du champ des observations. La segmentation est réalisée selon le critère du Maximum A Posteriori (MAP), qui est optimisé par un algorithme inspiré de la métaphore des colonies de fourmis, l'algorithme « Ant Colony Optimisation (ACO) ». L'évolution du champ des étiquettes se fait par minimisation d'une fonction d'énergie globale calculée en chaque pixel de l'image en tenant compte de son voisinage immédiat. L'idée est de représenter le problème de segmentation sous la forme de la recherche d'une «meilleure» partition dans un graphe. Des fourmis artificielles circulent dans ce graphe de façon aléatoire et incomplète, à la recherche de «bons» chemins qui représentent les bonnes partitions. Elles communiquent entre elles, à travers l'environnement, en déposant sur les arcs du graphe une trace d'hormone volatile appelée «phéromone»: cette hormone tend à attirer les fourmis artificielles dans une boucle de rétroaction positive, guidant de manière émergente la colonie vers une solution satisfaisante, si ce n'est la meilleure. Dans ce contexte, des algorithmes de fourmis utilisant différentes stratégies de construction des solutions et de manipulation des traces de phéromones sont développés et décrits dans le chapitre 6.

Globalement, les algorithmes présentés dans cette thèse sont basés sur des processus stigmergiques [Grassé, 1959]. Chaque fourmi dispose d'un comportement individuel influencé par le comportement des autres fourmis qui évoluent dans le même environnement. Le comportement collectif est assuré par la stigmergie et par l'environnement qui est considéré comme la mémoire collective de la colonie. Les principales caractéristiques de notre approche sont les suivantes : :

*La distribution* : Elle offre intrinsèquement des algorithmes distribués qui peuvent employer le calcul parallèle tout à fait facilement.

*La robustesse* : les algorithmes développés montrent un niveau élevé de la robustesse au changement en permettant aux fourmis de s'adapter dynamiquement aux changements globaux. D'un autre côté, la stochasticité des items comportementaux permet d'explorer différentes pistes qui peuvent être meilleurs voir optimaux.

*L'Adaptabilité* : étant donné qu'il n'y a aucun contrôle central dans nos algorithmes de segmentation les interactions locales des entités individuelles permet une grande adaptabilité du système global. Cette propriété permet à l'ensemble sans explicitation de but ni mécanismes de communication, de s'adapter globalement à des configurations différentes grâce à la combinaison des adaptations locales effectuées en parallèle.

*La Simplicité* : comme on le verra dans la suite de cette thèse, on peut concevoir un système collectif au fonctionnement complexe à partir d'individus très simples et donnant de très bons résultats.

## **1.6 Structure et contenu de la thèse**

D'un point de vue d'ensemble, cette thèse est divisée en trois grandes parties:

- Introduction générale : le présent chapitre dont le but est d'expliciter la problématique au cœur de cette thèse, de définir le contexte scientifique sous lequel nos travaux ont été menés et de faire la synthèse de nos contributions.
- Chapitres 2,3, et 4 dédiés aux éléments d'état de l'art : ils ont pour but de présenter une tentative de synthèse des différents domaines auxquels nous avons touché dans le cadre de cette thèse à savoir l'émergence et son application en informatique, les fourmis informatiques et la segmentation d'images.
- Chapitres 5 et 6 : La deuxième partie est dédiée aux développements effectués dans ce travail de thèse. Elle a pour objectif de montrer comment certains comportements collectifs des fourmis réelles ont pu être exploités pour la résolution collective du problème de la segmentation d'images.

D'une manière plus détaillée l'organisation de la thèse est comme suit :

**Le deuxième chapitre est consacré à l'étude du phénomène de l'émergence et du concept d'auto-organisation** qui sont la base de nos travaux. Pour comprendre ces concepts, une présentation générale des différentes techniques utilisant ces notions (algorithmes génétiques, réseaux de neurones, algorithmes de fourmis artificielles, optimisation par essaims de particules, ...) est alors effectuée.

**Le troisième chapitre est consacré aux différents travaux inspirés par les comportements collectifs des fourmis.** Après une introduction générale sur les fourmis réelles, le chapitre se focalise d'une part sur la métaheuristique OCF et ses domaines d'application et d'autre part sur les algorithmes inspirés du tri de couvain chez les fourmis.

**Dans le quatrième chapitre, nous passons en revue les principales méthodes de segmentation d'images.** Ce chapitre présente un tour d'horizon des méthodes de segmentation d'images de la littérature. Un intérêt particulier est porté pour les méthodes de segmentation en classes homogènes et sur les méthodes inspirées de la biologie. Un état de l'art des critères d'évaluation des résultats de segmentation est aussi présenté.

**Les chapitres 5 et 6 portent sur l'approche de segmentation d'images inspirée des comportements collectifs des fourmis et donnent quelques résultats expérimentaux ainsi qu'une étude comparative avec les algorithmes classiques de segmentation.** Le chapitre 5 présente une adaptation du comportement collectif des fourmis pour le tri de leurs couvains pour réaliser la classification automatique des images. Le chapitre 6 décrit trois algorithmes de segmentation par relaxation markovienne inspirés du comportement de fourrage des fourmis.

**Le chapitre 7 est réservé à une conclusion générale** sur l'ensemble de nos travaux ainsi qu'aux améliorations qui pourraient être apportées et aux perspectives qu'elle offre.

# Chapitre 2

## Le phénomène d' émergence et l' informatique

<b>Chapitre 2. Le phénomène d'émergence et l'informatique.....</b>	<b>24</b>
2.1 Introduction .....	24
2.2 Caractérisation d'un phénomène émergent .....	25
2.3 L'auto-organisation comme technique d'émergence .....	26
2.3.1 Définitions .....	27
2.3.2 Les mécanismes de l'auto-organisation .....	28
2.4 L'intelligence en essaim.....	30
2.5 Emergence et systèmes artificiels collectifs.....	30
2.5.1 Algorithmes évolutionnaires et algorithmes génétiques .....	31
2.5.2 Les réseaux de neurones.....	36
2.5.3 Algorithmes à essaims de particules .....	40
2.5.4 Les algorithmes de fourmis artificielles .....	42
2.6 Conclusion.....	44

---

## Chapitre 2. Le phénomène d'émergence et l'informatique.

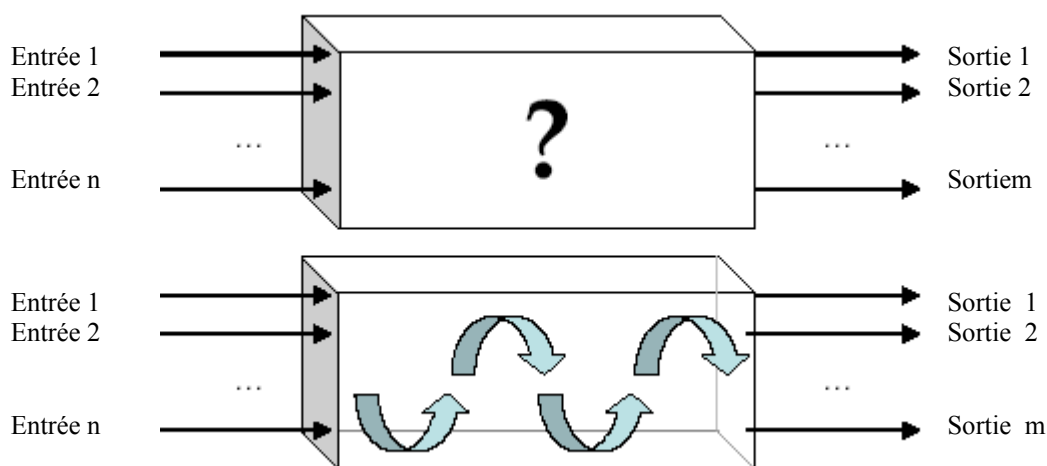
---

« L'Émergence : apparition plus ou moins soudaine d'une idée, d'un fait social, politique, économique ». D'après Petit Larousse Illustré.

### 2.1 Introduction

Bien que l'émergence soit encore aujourd'hui l'une des notions les plus floues et les plus discutées, elle est actuellement la plus utilisée pour la conception des systèmes artificiels. Son origine viendrait d'après Ali et Zimmer [Ali, 1997], du postulat datant de l'antiquité Grèce : *"le tout est plus que la somme de ses parties"*. Ce concept se retrouve aussi dans des écrits de Thalès et Anaximandre et bien plus tard dans le *"tout avant les parties"* d'Aristote et dans les écrits de J. W. von Goethe fondateur de la théorie de la *"gestalt"*.

Vers les années 1920, on assistera à l'apparition du mouvement dit proto-émergentisme, auquel participeront de nombreux chercheurs de différentes disciplines ayant pour objectif principal l'étude du phénomène d'émergence afin de reconnaître un phénomène émergent et le différencier de phénomènes explicables grâce à d'autres théories ou modèles [Georgé 2004]. Ainsi dans le cadre du proto-émergentisme, le phénomène d'émergence est vu comme une boîte noire possédant des entrées de plus bas niveaux et des sorties de plus haut niveau (voir figure 2.1). Ce modèle permet l'identification de tout phénomène émergent mais sans pour autant pouvoir l'expliquer.



**Figure 2.1.** Le proto-émergentisme (en haut) et le néo-émergentisme (en bas)



Ce dernier point a donné naissance à un nouveau courant le « *néo-émergentisme* » qui a pour ambition d'explorer cette boîte noire afin de comprendre les processus contrôlant les phénomènes émergents (voir figure 2.1) [Holland, 1999]. Ce mouvement est fortement lié à la théorie de la complexité et trouve ses bases dans diverses approches comme la dynamique des systèmes en physique, en mathématiques et en informatique. Il permet d'élaborer des méthodes visant à expliquer le phénomène d'émergence.

D'une manière intuitive, la notion d'émergence peut être définie comme une propriété macroscopique d'un système qui ne peut pas être déduite à partir de son fonctionnement microscopique. On parle d'émergence quand il y a apparition de structures, et de comportements plus complexes que ceux des entités qui forment le système. Ces comportements sont non programmés explicitement et donc non prévisibles. Cette propriété initialement dans les domaines de la biologie, de la thermodynamique est reprise depuis quelques années dans le domaine de l'informatique et a été exploitée pour la conception des systèmes artificiels.

## 2.2 Caractérisation d'un phénomène émergent

Un phénomène émergent peut être traité selon deux points de vue : un point de vue de l'observateur qui cherche à identifier le phénomène d'émergence et le point de vue du concepteur qui cherche à développer un système qui va produire le phénomène émergent [Gleize, 2004].

Selon le premier point de vue, l'émergence doit être définie afin de reconnaître un phénomène émergent. Plusieurs définitions de l'émergence ont été proposées dans la littérature. Nous tirons à partir d'elles des propriétés inter-reliées et communes qui permettent d'identifier un phénomène comme émergent:

***L'observation de nouveauté*** : l'émergence signifie qu'il y a apparition au niveau macro d'un phénomène nouveau non observé au niveau micro. Il est alors impossible de décrire, de prédire ou d'expliquer à base des conditions de base définies au niveau micro [Van de Vijver, 1997 ; Gleize, 2004].

***L'observation d'un phénomène ostensible*** : il s'impose à l'observateur au macro-niveau sans que rien au micro-niveau ne permet de le prédire [Atlan, 2000]. Autrement dit en terme d'irréductibilité il est impossible de déduire à partir des propriétés de micro-niveau celles de macro-niveau [Ali, 1997 ; Gleize, 2004].

***L'observation d'une cohérence et d'une corrélation du phénomène*** : il a une identité propre mais liée aux parties de micro-niveau qui le produise. Selon Goldstein, l'émergence est définie comme l'apparition de processus d'auto-organisation dans un système complexe [Goldstein, 1999]. Le système doit être capable de modifier de son comportement en fonction de son environnement sans l'aide d'un superviseur[Gleize, 2004].

***L'observation d'une dynamique*** le phénomène n'est pas connu au départ, s'auto-crée et s'auto-maintient. Langton définit l'émergence en termes de relation de feedback entre les niveaux (macro et micro) dans un système émergent [Langton, 1991]. Autrement dit, les interactions entre les éléments du bas\_niveau causent celles du haut\_niveau et les macro-interactions contraignent les micro-interactions[Odell, 2001].

Du point de vue concepteur d'un système produisant un phénomène émergent, le système doit présenter les caractéristiques suivantes :

***La non-linéarité*** : La linéarité permet l'interprétation du comportement du niveau global à partir des activités du niveau local. Pour qu'il ait émergence, il faut donc que les interactions entre les éléments du système soient non linéaires. Ceci signifie que tout composant du système peut être influencée de manière indirecte par d'autres composants du système (qui ne lui sont pas directement liés).

***Un état proche de l'équilibre*** : Au lieu de s'intéresser aux points qui conduisent à l'équilibre du système, on s'intéresse au voisinage de ces points, comme en théorie de la complexité. En de tels points, l'apparition de phénomènes non prévisibles explique le caractère inattendu de l'émergence [Prigogine, 1977].

***L'auto-organisation*** : Un système émergent est un système capable de s'auto-organiser et de changer de comportement en fonction de son environnement.

***Les attracteurs*** : Contrairement aux premiers systèmes pour lesquels il n'existait qu'un seul attracteur valide qui menait à l'équilibre, les systèmes émergents possèdent différents types : le point fixe, le cycle limite et l'attracteur étrange. Ces attracteurs ne sont pas pré-donnés et ne dictent pas au système l'état à atteindre, mais plutôt les moyens de passer d'état en état [Goldstein, 1999].

### **2.3 L'auto-organisation comme technique d'émergence**

Depuis quelques années l'utilisation du concept d'émergence pour la conception de systèmes complexes s'est très vite répandue. En effet, de tels systèmes ne permettent pas d'appliquer

l'approche cartésienne classique pour laquelle la tâche globale est décomposée en sous-tâches car les étapes de résolution ne peuvent pas être programmées explicitement au départ. L'approche émergentiste se révèle donc comme un moyen de passage entre l'activité du "micro-niveau" (les interactions locales entre les composants du système et celui du "macro-niveau" (le comportement global). Tenant compte des définitions présentées dans le paragraphe précédent, il apparaît que l'auto-organisation est un élément essentiel pour l'obtention d'un phénomène émergent. Dans ce qui suit, nous allons définir le terme « auto-organisation » ainsi que son principe.

### 2.3.1 Définitions

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

Définition1: « L'auto-organisation est une description d'un comportement, elle a une valeur heuristique et elle permet d'indiquer un phénomène. Elle est condamnée à rester une simple description, tant qu'on ne se préoccupe pas de rechercher le mécanisme qui est à son origine » [Varela, 1988]

Définition 2 : « Un système auto-organisateur est un système qui change sa structure de base en fonction de son expérience et de son environnement. » [Ünsal ,1993].

Définition 3 : « L'auto-organisation est un processus où l'organisation (contrainte, redondante) d'un système croît de manière spontanée, i.e. sans que cet accroissement soit contrôlé par l'environnement ou ce qui l'entoure ou encore un système externe. » [Krippendorff, 1997].

Définition 4 : « L'auto-organisation est définie par un ensemble de mécanismes dynamiques, permettant à des structures d'apparaître au niveau global d'un système (niveau macro) à partir des interactions de ses composants de plus bas-niveau (niveau micro). Les règles spécifiant les interactions entre les composants du système, sont exécutées sur la base d'informations purement locales (niveau micro), et ne font en aucun cas référence à la configuration globale (du niveau macro). Cette configuration globale est une propriété émergente du système plutôt qu'une propriété imposée au système, sous l'influence d'une commande extérieure » [Bonabeau, 1999; Bonabeau, 1997],.

Définition 5 : « L'auto-organisation est définie comme un processus dans lequel un modèle de niveau global émerge uniquement d'un grand nombre interaction entre les comportements de

bas niveau du système. De plus les règles spécifiant les interactions entre composantes du système sont suivies en utilisant uniquement des informations locales sans références au modèle global » [Camazine, 2000].

Définition 6 : « Tout processus au cours duquel des structures émergent au niveau collectif (ou plus généralement apparition d'une structure à l'échelle  $N + 1$  à partir d'une dynamique définie à l'échelle  $N$ ), à partir de la multitude des interactions entre individus, sans être codées explicitement au niveau individuel. » [Théraulaz, 1997].

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

### 2.3.2 Les mécanismes de l'auto-organisation

Devant la complexité des structures obtenues par le phénomène d'auto-organisation, on se demande comment ces structures émergent-elles sans un plan prédéfini. En fait leur apparition est due à de nombreuses interactions entre les éléments du système ainsi qu'avec l'environnement externe. Nous pouvons distinguer en particulier les principaux facteurs suivants :

*La rétroaction* : pendant son exécution, les composants du système interagissent ensemble ou avec leur environnement. On considère une forme particulière d'interaction est la **rétroaction** (ou feed-back) qui est obtenue lorsque les résultats (obtenus par transformation des données d'entrée) sont retransmis au système sous la forme de nouvelles données d'entrée. Deux types de rétroaction peuvent alors être observés :

- *La rétroaction positive* : appelé aussi « auto-catalyse », c'est un mécanisme qui permet de renforcer une action impliquant une convergence très rapide. Il est observé quand les résultats agissent de la même manière que les données d'entrée c'est à dire renforcent le processus de transformation. Ce phénomène est observé chez les fourmis lors de la recherche de nourriture. Le cumul de

traces de phéromones déposées par chaque fourmi sur le chemin qu'elle emprunte, permet de renforcer ultérieurement le choix de ce chemin par les autres fourmis de la même colonie. [Deneubourg, 1987].

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

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

En effet Grassé a montré vers la fin des années 1950 que chez les termites, la construction d'une bâtisse est guidée par la construction elle-même et ne dépendait pas seulement des termites bâtisseuses : l'insecte ne dirige pas son travail mais il est guidé par lui [Grassé 1959]. Ainsi, toute nouvelle forme construite, devient un nouveau point de départ matériel pour les autres termites produisant ainsi une nouvelle forme stimulante, qui peut orienter et déclencher en retour une nouvelle activité bâtisseuse chez les autres membres de la colonie. Grâce à la stigmergie, les termites arrivent à s'auto-organiser pour réaliser un travail collectif sans aucune coordination directe.

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

## 2.4 L'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 [Bonabeau 1997]. 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 [Van Dyke Parunak, 1997].

Cette capacité à passer de comportements individuels simples à des comportements collectifs complexes est appelée "intelligence en essaim" ("swarm intelligence") ou «intelligence collective» sous domaine de l'intelligence artificielle distribuée. D'après Eric Bonabeau et Guy Théraulaz dans [Bonabeau, 1994], « ces capacités cognitives collectives qui caractérisent les phénomènes d'intelligence collective, sont mises en œuvre sans régulateur central capable de détecter et de compenser des déviations globales, et sans non plus qu'une connaissance suffisante existe au niveau de chaque élément pour que l'on puisse lui attribuer une perception globale de la situation ».

Dans une approche basée sur l'intelligence en essaim, les entités formant l'essaim ont un comportement relativement simple, que l'on ne peut pas qualifier d'intelligent. Cependant ces comportements individuels peuvent faire émerger grâce à des interactions locales inter-individus et avec l'environnement des règles locales un comportement complexe, adaptatif sans aucun contrôle central. « Dans les sociétés d'insectes, le «projet» global n'est pas programmé explicitement chez les individus, mais émerge de l'enchaînement d'un grand nombre d'interactions élémentaires entre individus, ou entre individus et environnement. Il y a en fait intelligence collective construite à partir de nombreuses simplicités individuelles.»[Deneubourg,1991]. L'intelligence en essaim tente alors de simuler les mécanismes produisant ce type de comportements, afin de proposer de nouvelles techniques pour la résolution de problèmes.

## 2.5 Emergence et systèmes artificiels collectifs

Comme nous l'avons souligné auparavant, la méthode classique de résolution de problèmes nécessite de formuler a priori le domaine du problème ainsi que les étapes de la résolution. Chose difficile lorsqu'il s'agit d'un domaine dynamique et incertain. Ces difficultés ont amené les informaticiens à chercher de nouveaux modèles, de nouvelles théories, de nouveaux systèmes pour résoudre de tels problèmes. D'où l'apparition de nouveaux systèmes à fonctionnalité émergente. La particularité de tels systèmes réside dans deux points. Le

premier est que le calcul n'est pas explicite, ni programmé à l'avance mais est de nature émergente. On parle alors de « calcul émergent » ou « emergent computation » [Forrest 1990]. Le deuxième point est qu'ils sont constitués d'entités dont le comportement est en général très simple et réactif, cependant ils sont simples capables, au niveau du groupe, de comportements d'apprentissage, d'adaptation aux changements, de robustesse vis-à-vis des cas non prévus. Pour cette raison, ces systèmes sont généralement qualifiés de « systèmes collectifs intelligents ».

La plupart des applications utilisant ce type de systèmes l'ont été dans le domaine de l'optimisation. Il s'agit de maximiser ou de minimiser une fonction objectif afin de trouver les meilleures solutions à un problème donné. C'est généralement un problème NP-complet, pour lequel les méthodes exactes classiques ne donnent pas de bons résultats en un temps acceptable d'où la nécessité d'utiliser des heuristiques. Dans la suite, nous allons présenter certains de ses systèmes.

### 2.5.1 Algorithmes évolutionnaires et algorithmes génétiques

*« L'évolution est une création sans cesse renouvelée », Henri Bergson.*

En 1859, Charles Darwin évoqua, dans son livre « De l'origine des espèces par voies de sélection Naturelle » [Darwin, 1859], les premiers principes de la théorie de l'évolution. Selon lui l'évolution des systèmes vivants au cours des générations s'opère en deux étapes : la sélection et la reproduction :

- La sélection naturelle est le mécanisme central qui opère au niveau des populations, en entraînant la mort sélective des plus faibles et la survie des individus les mieux adaptés à leur environnement.
- La reproduction fait subir des modifications successives aux individus de la population par mutations et recombinaisons, aboutissant ainsi à une grande diversité entre deux populations d'une même espèce.

De son côté, John Mendel, le fondateur de la génétique, avait effectué des travaux sur l'hérédité, entre 1859 et 1866 [Mendel, 1865] qui expliquent les lois de transmission des caractères à travers les générations dans le cadre d'une reproduction sexuée. Les caractéristiques héréditaires sont ainsi localisées dans le génome, qui constitue le patrimoine génétique de chaque individu. Le patrimoine génétique est composé de gènes, codant les protéines responsables de l'architecture et du fonctionnement au niveau cellulaire. Les gènes constituent, à ce titre, l'unité sémantique du langage génétique. La reproduction sexuée met

en jeu deux mécanismes fondamentaux et complémentaires : le croisement et la mutation génétiques. Le croisement implique la combinaison du patrimoine génétique des deux parents pour former le patrimoine de l'enfant. Il permet ainsi de former des individus distincts, bien que possédant des traits propres à chaque parent. La mutation intervient durant la phase de croisement et se traduit par la modification spontanée de quelques gènes..

La synthèse des ses deux théories augmentée de la découverte de l'ADN a donné naissance au néo-darwinisme qui a inspiré plus tard les chercheurs informaticiens pour le développement des modèles artificiels de l'évolution intégrant des propriétés telles que la mutation et le croisement et leur application au domaine de l'optimisation. Parmi ces modèles on trouve une classe d'algorithmes regroupés sous le nom générique les algorithmes évolutionnaires (AE) apparus dans les années 70 avec les travaux de Ingo Rechenberg et ceux de Hans-Paul Schwefel [Schwefel, 1981].

Comme exemple des AE, les Algorithmes génétiques (AG) sont certainement la branche des AE la plus connue et la plus utilisée de ces techniques qui se différencient dans leurs façons de coder les individus (donc modéliser le problème à résoudre) et par leur façon de faire évoluer la population, mais ces versions sont toutes basées sur les mêmes principes de base. Une étude comparative de ses méthodes est présentée dans [Hoffmeister, 1991].

#### 2.5.1.1 Les algorithmes génétiques

Initialement créés par John Holland, [Holland, 1975], puis développés par d'autres chercheurs comme De Jong [De Jong, 1975], Goldberg [Goldberg, 1989], Davis [Davis, 1991] et Michalewicz [Michalewicz, 1996], les AG doivent leur nom à l'analogie des mécanismes d'évolution naturelle des espèces vivantes et utilisent les deux principes du néodarwinisme : la survie des individus les mieux adaptés de Darwin et la recombinaison génétique de Mendel.

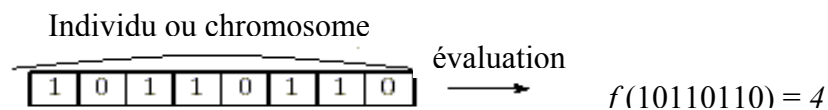
Le principe de base est de mimer ces deux mécanismes pour faire évoluer une population de solutions (représentant des individus) afin d'obtenir des solutions de qualité de plus en plus meilleure. Un algorithme génétique est simple à mettre en œuvre et se décrit par les points suivants :

- 1) Choix d'un codage approprié pour les individus de la population. Ce codage doit être complet et capable de coder toutes les solutions possibles. Dans un premier temps, Holland a utilisé un codage sous forme de chaîne de bits de longueur fixe afin de maintenir le plus possible l'analogie avec la structure protéinique de l'ADN. Le codage binaire possède l'avantage supplémentaire de fournir un langage quasi-universel, indépendant du problème traité. Le codage binaire d'un individu est appelé



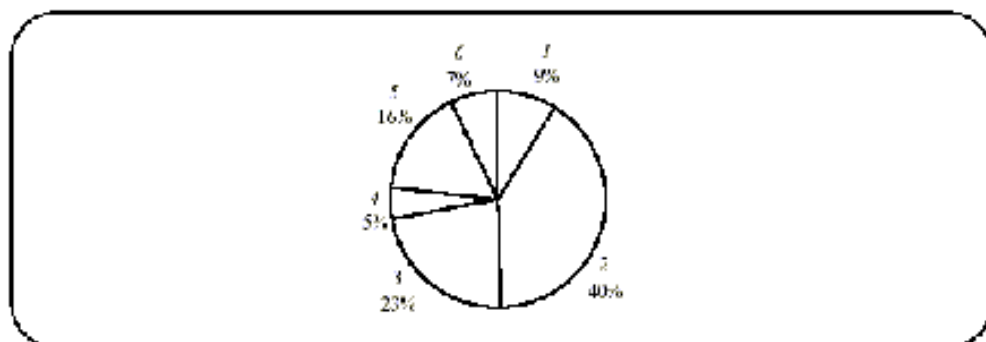
« chromosome » et ses éléments sont appelés « gènes ». Plus récemment, d'autres types de codage sont apparus pour pallier les limitations du codage binaire, trop limitatif ou trop complexe à mettre en œuvre, tels que le codage entier ou réel [Goldberg, 1989],

- 2) Construction de la population initiale d'individus;
- 3) Association à chaque individu de la population une fonction d'évaluation la fitness qui mesure son degré d'adaptation à l'objectif visé. La figure 2.2 donne un exemple de codage d'un chromosome sur 8 bits. L'évaluation de cet individu consiste à transformer la chaîne 0/1 en une valeur réelle.



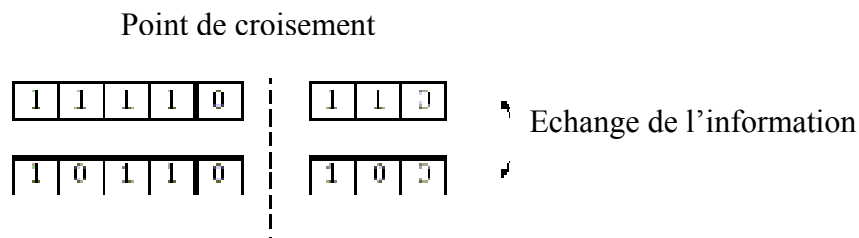
**Figure 2.2. Codage de l'information**

- 4) 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 par l'application des opérateurs génétiques : sélection, croisement et mutation.
  - a. La sélection favorise la survie des individus les plus performants et la transmission de leurs caractéristiques à la génération suivante. La sélection peut se faire selon plusieurs procédures, les plus couramment utilisées sont : la sélection par la roue de la fortune, par rang de classement, et par tournoi. La Figure 2.3 représente la sélection par la roulette.



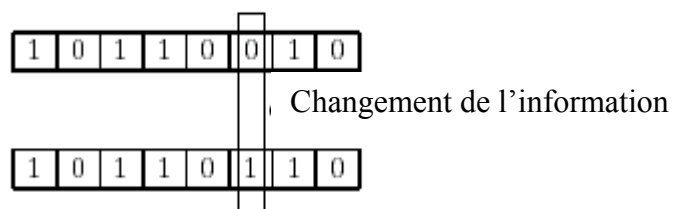
**Figure 2.3. Modèle de la roulette.**

- b. Le croisement et la mutation sont des opérateurs de reproduction qui agissent sur un ou plusieurs chromosomes issus de la phase de sélection en fonction d'une certaine probabilité. Le croisement permet le brassage du patrimoine génétique présent dans la population en produisant de nouveaux chromosomes (enfants) à partir de chromosomes parents. Ainsi chaque chromosome a une probabilité  $P_c$  de se croiser avec un autre chromosome pour produire un chromosome nouveau qui sera insérer dans la nouvelle population. Il existe une variété de méthodes pour faire croiser deux chromosomes, les plus généralement utilisés sont le croisement à un ou plusieurs points (figure 2.4). D'autres variantes de croisement existent avec un nombre de points qui peut être de deux (trois, voire plus, ou encore le croisement uniforme [Syswerda, 1991]).



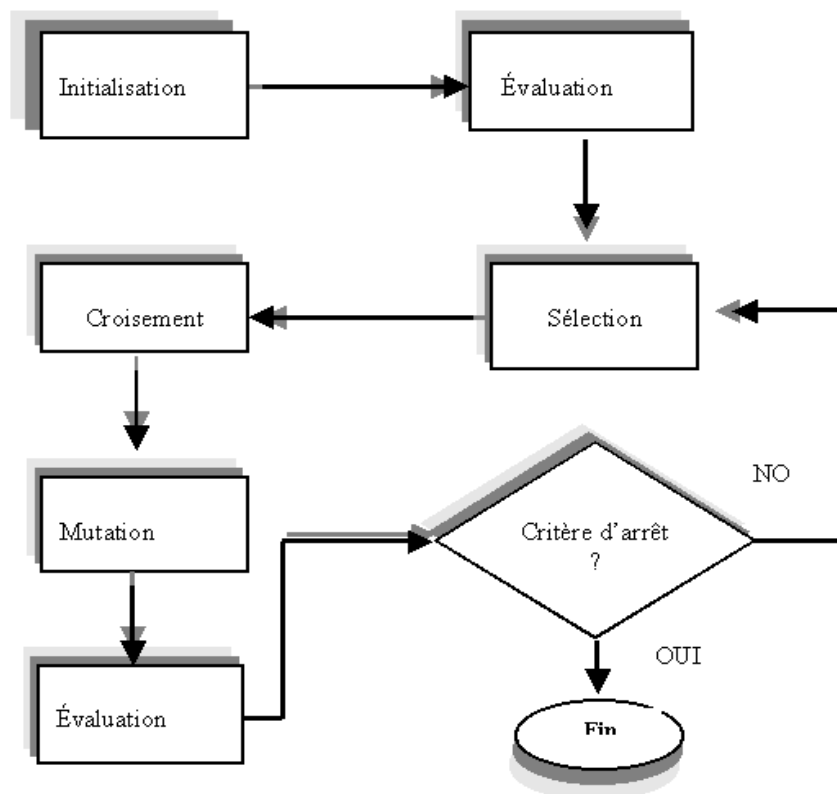
**Figure 2.4. Opérateur de croisement monopoint appliqué à deux chromosomes codés sur 8 bits.**

- c. L'opérateur de mutation agit sur un seul chromosome choisi avec une probabilité  $P_m$ . La mutation consiste à changer aléatoirement les valeurs de certains gènes du chromosome. Cet opérateur permet de créer une diversité dans la population afin d'éviter une convergence prématurée. La figure 2.5 montre un exemple de mutation sur un chromosome codé sur 8 bits.



**Figure 2.5. Opérateur de mutation appliqué à deux chromosomes codés sur 8 bits.**

La structure générique d'un AG est donnée par la figure 2.6.



**Figure 2.6. Structure générale d'un algorithme génétique.**

Les algorithmes génétiques utilisent le même principe que l'évolution naturelle, la compétition entre individus, pour créer des systèmes complexes pouvant s'adapter à de nombreuses situations. Cela permet de résoudre efficacement des problèmes d'optimisation combinatoires en explorant complètement l'espace de recherche des solutions sans pour autant avoir un coût démesuré. Les algorithmes génétiques recherchent une « bonne » solution à un problème parmi un ensemble de solutions en tirant parti de l'historique de cette recherche afin de favoriser l'émergence des solutions les mieux adaptées. Les algorithmes génétiques ont fait leur preuve dans des domaines très variés. Afin de les appliquer à un problème particulier, il est nécessaire de définir convenablement la fonction d'évaluation et le codage d'une solution. Ces algorithmes nécessitent également la définition d'opérateurs, aussi bien de mutation que de croisement, en adéquation avec le problème.

Toutefois, la recherche d'une solution par les AG ne garantit pas l'obtention d'une solution optimale. Une population initiale mal choisie, une convergence trop rapide vers un optimum local, etc., peuvent bloquer le processus de résolution. Il n'existe pas de méthodes qui permettent de dire quelle est la taille optimale de la population, quel est le meilleur codage, quel taux de mutation et de croisement utiliser. Ces paramètres de fonctionnement dépendent de l'application et, bien souvent, ils sont fixés de façon empirique.

### 2.5.1.2 Références

Plusieurs types d'évolution ont été développés, donnant naissance à différents types d'AE qu'on peut regrouper en trois grandes classes : les algorithmes génétiques [Holland, 1975]; [Goldberg, 1989], les stratégies d'évolution, [Rechenberg, 1973 ; Schwefel, 1981] et [Herdy, 1991], la programmation évolutive, [Fogel, 1966], et la programmation génétique, [Koza, 1992 ; Koza 2003]. On peut aussi citer les travaux d'Evelyne Luthon [Lutton 2003] qui utilisent les fractales pour la compréhension des fonctionnement du calcul évolutionnaire ainsi que celui de Michel Sebag [Ratle, 2003] qui s'intéresse de plus près à la programmation génétique.

### **2.5.2 Les réseaux de neurones**

De la même manière que le Calcul Evolutionnaire a été inspiré de l'évolution naturelle des espèces, le domaine des réseaux de neurones (RN) représente une tentative de reproduire artificiellement le fonctionnement du cerveau humain.

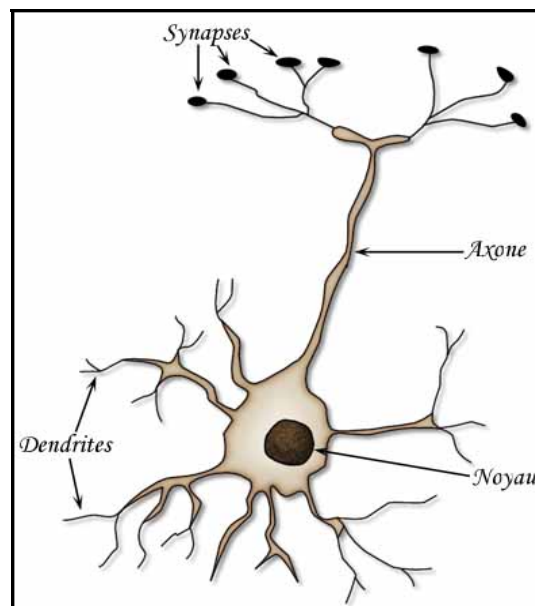
Les réseaux de neurones formels sont des modèles théoriques de traitement de l'information inspirés des observations relatives au fonctionnement des neurones biologiques et du cortex cérébral. Par analogie aux neurones biologiques, les neurones artificiels ont pour but de reproduire des raisonnements « intelligents » d'une manière artificielle. Ces neurones peuvent adopter de certaines qualités habituellement propres au biologique, c'est-à-dire, la généralisation, l'évolutivité, et une certaine forme de déduction.

#### 2.5.2.1 Le modèle neurophysiologique

Les cellules nerveuses, appelées neurones sont les éléments de base du système nerveux central qui se compose d'environ  $10^{12}$  neurones (mille milliards), Le neurone est une cellule composée de trois parties (figure 2.7.) :

- Un corps cellulaire qui contient le noyau et se charge d'effectuer les transformations biochimiques nécessaires à la synthèse des éléments assurant la vie du neurone ;
- Les dendrites qui sont des ramifications du corps cellulaire. Elles permettent au neurone de capter les signaux lui parvenant de l'extérieur ;
- L'axone généralement plus long que les dendrites, il se ramifie à son extrémité où il communique avec les autres neurones. Il sert de moyen de transport pour les signaux émis par le neurone.

- Les connexions entre neurones sont réalisées au niveau des synapses, lieu de proximité d'axone émetteur et de dendrites réceptrices.

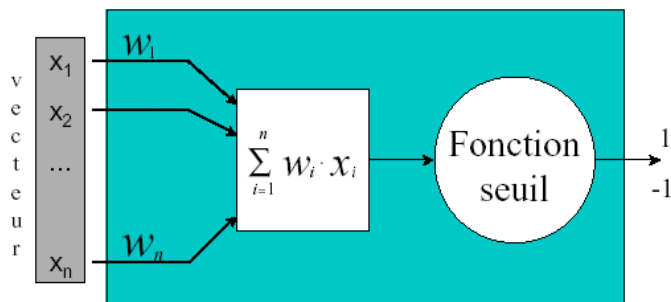


**Figure 2.7. Structure d'un neurone biologique**

Le neurone biologique reçoit des impulsions de ses neurones voisins avec lesquels il est connecté à travers les synapses. Les influx nerveux transmis par les dendrites sont sommés. Si la sommation dépasse un seuil, le neurone répond par un influx nerveux ou un potentiel d'action qui se propage le long de son axone. Si la sommation est inférieure au seuil, le neurone reste inactif. Les premières cellules qui alimentent le réseau peuvent être constituées par des capteurs (cellules sensorielles) comme les cellules de la rétine de l'oeil, par exemple.

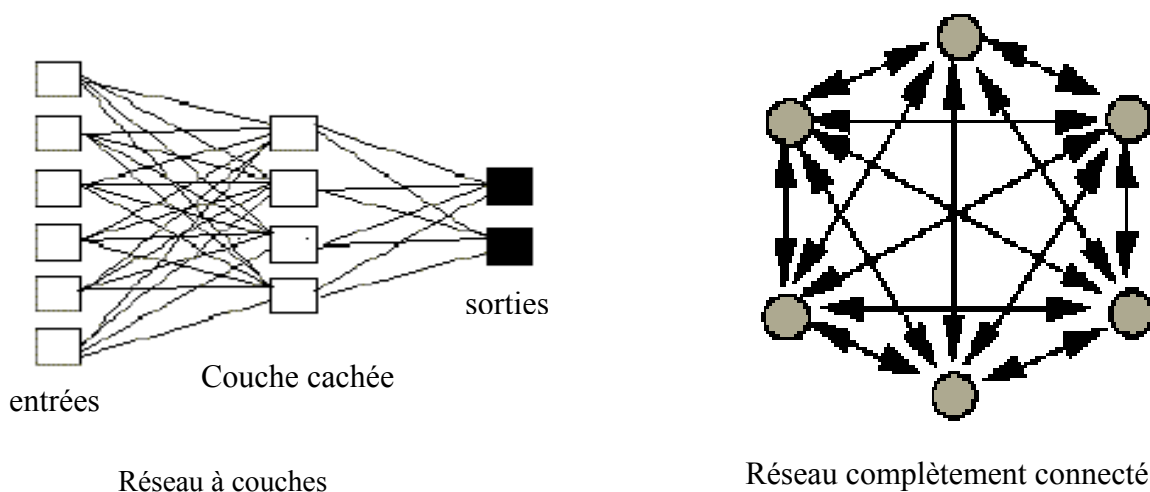
#### 2.5.2.2 Du neurone biologique au neurone formel

En 1943, Warren McCulloch et Walter Pitts en s'inspirant de leurs travaux sur le neurone biologique, ont proposé un des premiers modèles de neurone artificiel [McCulloch, 1943] qui deviendra la base des réseaux de neurones artificiels. Un RN est un réseau composé généralement d'un grand nombre d'unités de calcul très simples appelés *neurones formels*. Ces neurones formels sont interconnectés entre eux, de sorte que les signaux sortant (*outputs*) des neurones deviennent des signaux entrant (*inputs*) dans d'autres neurones. Chaque neurone artificiel est un automate binaire à deux états « actif » et « inactif ». Il reçoit un nombre variable de signaux  $x$  en provenance des autres neurones. A chacun de ces signaux est associé un poids  $w$  représentatif de la force de la connexion. Le neurone change d'état suivant la somme des connexions pondérées dépasse ou non un seuil d'activation  $\theta$  (figure 2.8).

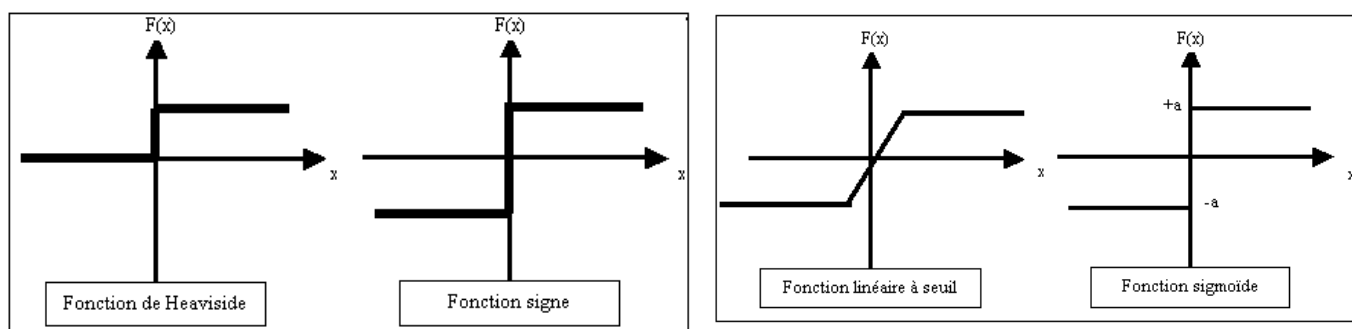


**Figure 2.8. le neurone formel de Mac Culloch et Pitts**

Un RN est caractérisé par sa topologie (figure 2.9.)qui dépend de la façon dont les neurones sont reliés (réseaux en couche, complètement connecté, récurrent ), par sa fonction d'activation (figure 2.10.) et par le mode d'apprentissage utilisé (supervisé, non supervisé).



**Figure 2.9. Quelques topologie de réseau de neurones**



**Figure 2.10. Quelques types de fonction d'activation**

D'une façon plus générale, on définit un neurone formel par les cinq paramètres suivants :

- 1 la *nature des entrées* (booléenne ou réelle)
- 2 la *fonction d'entrée totale*, définissant le prétraitement effectué sur les entrées,
- 3 la *fonction de seuillage* (appelée aussi *fonction d'activation*) du neurone définissant son état interne en fonction de la somme pondérée de ses entrées. .
- 4 la *fonction de sortie* calculant la sortie du neurone en fonction de son état d'activation,
- 5 la *nature des sorties* du neurone

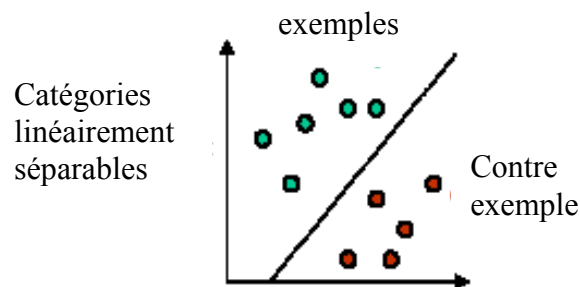
### 2.5.2.3 Références

Depuis le neurone de McCulloch et Pitts [McCulloch, 1943] et les travaux de Hebb [Hebb, 1949] sur les modèles d'apprentissage présentés dans son ouvrage « The Organization of Behavior », les réseaux de neurones ont commencé à évoluer. En 1958, Frank Rosenblatt proposa le Perceptron [Rosenblatt, 1958]. C'est un réseau de neurones inspiré du système visuel. Il possède deux couches de neurones: une couche de perception et une couche de sortie liée à la prise de décision. C'est le premier système artificiel capable d'apprendre par expérience. Dans la même période, L'Adaline (ADaptive LINar Element) [Widrow ? 1960] a été présenté par B. Widrow, chercheur américain à Stanford. En 1969, M. Minsky et S. Papert publient leur livre *Perceptrons* [Minsky 1969] qui était une critique des propriétés du Perceptron en montrant qu'il ne peut apprendre que des fonctions séparables linéairement (figure 2.11).

Cette limitation va avoir une grande incidence sur la recherche dans ce domaine, qui va fortement diminuer jusqu'en 1972, où T. Kohonen présente ses travaux sur un nouveau modèle de réseaux de neurones connu sous le nom « les cartes de kohonen ». Les cartes auto-organisatrices de Kohonen, appelées aussi cartes auto-adaptatives, sont des réseaux de neurones formels qui vont s'auto-organiser pour réaliser un objectif [Kohonen, 2001]. Ce type de réseaux tient compte de la proximité géographique des données d'entrées dans le fonctionnement des neurones. C'est à dire que l'algorithme permet d'assurer que les données projetées dans un même voisinage sont des données proches. Ces systèmes sont donc principalement utilisés pour des applications de clustering (regroupement), de topologie ou de classement de données. Dans une carte de kohonen, tous les neurones de la carte sont interconnectés. Le plus souvent une grille à deux dimensions est utilisée comme support des neurones. Les poids des arcs de connexion entre deux neurones dépendent de leur proximité. Plus deux neurones sont proches, plus leurs poids sont élevés. Le fonctionnement est simple : on présente un objet en entrée du réseau, et le réseau active en sortie le neurone dont le poids est le plus proche. Ensuite, les poids sont mis à jour non seulement pour le neurone activé

mais aussi pour les neurones voisins de ce dernier. Au cours des itérations, la configuration du voisinage peut décroître afin de stabiliser les poids.

En 1982 J. Hopfield présente son étude d'un réseau complètement rebouclé [Hopfield, 1982], dont il analyse la dynamique, suivis des travaux de Rumelhart qui introduisa en 1985 les réseaux multicouches avec la rétropropagation du gradient [Rumelhart, 1986] comme algorithme d'apprentissage. De nos jours, les réseaux multicouches et la rétropropagation de gradient reste le modèle le plus étudié et le plus productif au niveau des applications.



**Figure 2.11. limitation du Perceptron.**

Dans leur principe de base, les RN reposent entièrement sur les concepts d'auto-organisation et d'émergence : des règles locales de base d'interaction entre de nombreuses entités qui produisent au niveau global des comportements de plus haut niveau qui ne sont pas décrits au sein des règles. Si on leur rajoute en respectant toujours ces principes des capacités de création de neurones formels et d'apparition de nouveaux liens, ils présentent une technique intéressante pour la résolution de problèmes par émergence.

Grâce aux propriétés d'apprentissage et à la capacité puissante d'approximation des réseaux de neurones artificiels, ils ont trouvé des applications dans de nombreux domaines différents dont le contrôle de signaux, la robotique, le domaine médical (reconnaissance d'images de visages humains et d'objets) et en authentification (reconnaissance de signature, de caractères manuscrits, détection de fraudes). Ce paradigme s'avère souvent efficace pour résoudre des problèmes de classification et d'optimisation.

### 2.5.3 Algorithmes à essaims de particules

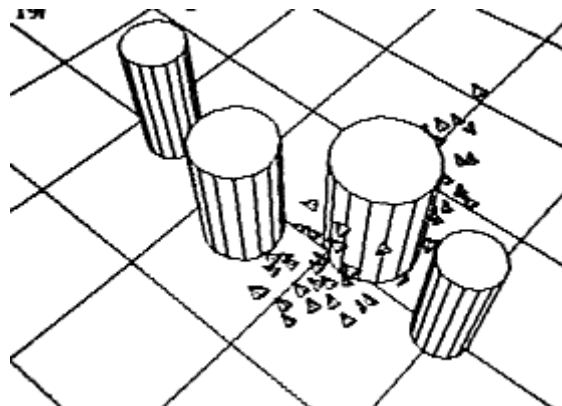
Les algorithmes « d'optimisation par essaim de particules » (Particle Swarm Optimization – PSO) introduits pour la première fois par Kennedy et Eberhart [Kennedy, 1995 ; Eberhart 2001] sont inspirés des déplacements collectifs observés chez certains animaux sociaux tels que les poissons et les oiseaux migrateurs. En effet, il est étonnant de voir comment ces animaux se déplacent en groupe dans une seule direction, se divisent parfois en plusieurs groupes afin d'éviter un obstacle ou un prédateur, puis reforment un groupe compact. Avec



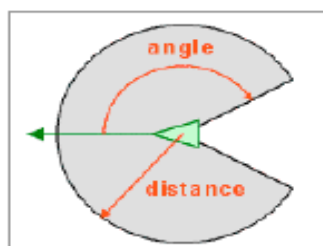
des règles locales très simples comme « rester proche des autres individus », « aller dans la même direction », « aller à la même vitesse », ces animaux sont capables d'éviter un prédateur par des mouvements d'explosion puis re-forment le groupe originel, tout en maintenant la cohésion du banc. Dans l'algorithme à essaim de particules, les individus de l'algorithme sont appelés particules et la population est appelée essaim.

L'approche par PSO trouve ses racines dans les travaux de l'infographiste Craig Reynolds qui fut le premier à modéliser ce genre de comportement collectif émergent par ses fameux boids [Reynolds, 1987] (Figure 2.12). Les boids sont des individus autonomes. Chaque individu n'a qu'une vision locale de son environnement (Figure 2.13) et son comportement va être le résultat d'une somme d'attractions et de répulsions engendrées par son environnement direct, codées par des règles très simples (Figure 2.14) :

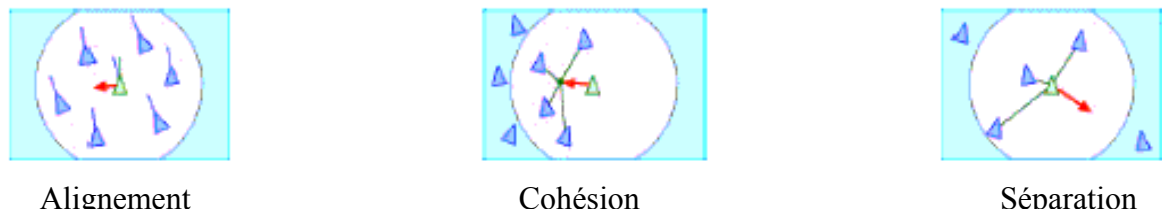
- 1 Chaque individu doit éviter de heurter ses voisins.
- 2 Chaque individu tend à s'approcher des vitesses et directions générales du groupe local, c'est-à-dire des voisins immédiats.
- 3 Chaque individu cherche à s'approcher du centre de gravité du groupe local.



**Figure 2.12. Les boids de Reynolds [Reynolds 2001]**

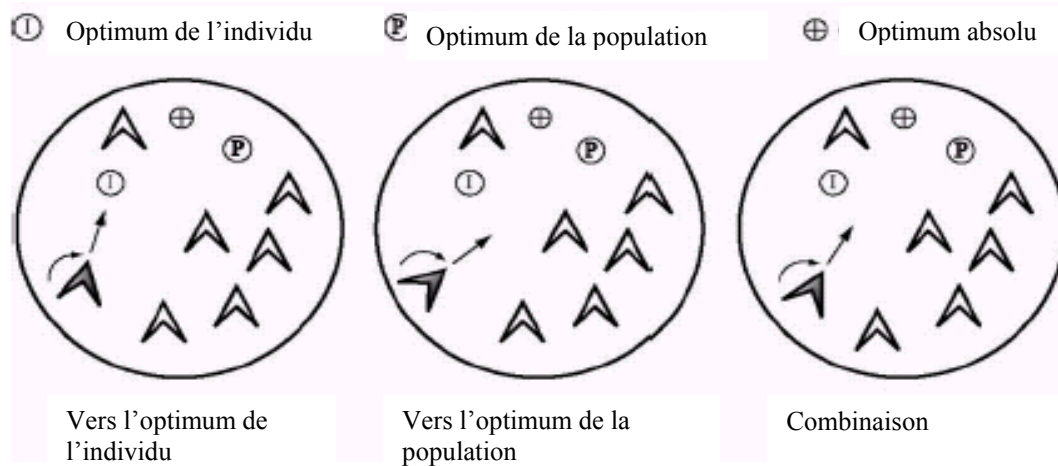


**Figure 2.13. Le voisinage local d'un boid.**



**Figure 2.14. Les trois règles trouvées par Reynolds pour programmer des mouvements collectifs complexes.**

La technique PSO utilise des règles de comportements locaux pour faire émerger à travers une dynamique collective auto-organisée la valeur optimale d'une fonction de fitness. Chaque particule est caractérisée par sa position (où la meilleure valeur pour sa fonction de fitness a été rencontrée) et un vecteur de changement de position (appelé vitesse). À chaque itération, une particule décide de son prochain mouvement en fonction de sa propre expérience, qui est dans ce cas la mémoire de la meilleure position qu'elle a rencontrée, et en fonction de son meilleur voisin. Ce voisinage peut être défini spatialement en prenant par exemple la distance euclidienne entre les positions de deux particules ou sociométriquement (position dans l'essaim de l'individu) (figure 2.15.).



**Figure 2.15. Règles simples d'application locale utilisées dans un essaim de particules.**

Cette méthode est facile à implémenter et donne de très bons résultats. Une présentation plus détaillée de l'approche par PSO se trouve dans la synthèse en français de Clerc [Clerc, 2002].

#### 2.5.4 Les algorithmes de fourmis artificielles

En plus des sociétés animales, les sociétés d'insectes telles que les fourmis, les termites, les araignées et certaines espèces d'abeilles et de guêpes, constituent-elles aussi des sources

d'inspiration pour le développement de systèmes artificiels « Les sociétés d'insectes nous proposent un modèle de fonctionnement bien différent du modèle humain: un modèle décentralisé, fondé sur la coopération d'unités autonomes au comportement relativement simple et probabiliste, qui sont distribuées dans l'environnement et ne disposent que d'informations locales. » [Deneubourg, 1991].

L'exemple des fourmis est le plus répandu dans la littérature à cause de leur capacité à réaliser des tâches hautement complexes à partir des interactions d'insectes simples à l'intelligence très rudimentaire. [Deneubourg, 1990; Coloni, 1992; Dorigo, 1996; Bonabeau, 1997; Van Dyke Parunak, 1997; Bonabeau, 1999; Topin, 1999a]. En particulier deux comportements collectifs ont été principalement étudiés chez les fourmis : l'optimisation de chemin et le tri des éléments du couvain.

Le premier comportement met en évidence la capacité des fourmis à optimiser la procédure de fourragement. En effet, au départ les fourmis se déplacent d'une manière aléatoire de la fourmilière vers une source de nourriture en laissant des traces de phéromones sur le chemin qu'elles empruntent. Cette trace tend à attirer les congénères qui, en la suivant, vont parvenir à la nourriture. Il a été constaté alors que les fourmis qui choisissent l'itinéraire le plus court, le terminent le plus vite. Les fourmis vont alors préférer progressivement cet itinéraire car il est marqué avec une plus grande quantité de phéromones [Goss, 1989]. De plus l'évaporation naturelle de la phéromone renforce encore plus ce choix des fourmis par l'affaiblissement des chemins plus longs.

Le second comportement collectif des fourmis concerne leur aptitude à nettoyer leur nid en organisant collectivement des cimetières composés de cadavres empilés les uns sur les autres. Le principe est le suivant : plus un cadavre est isolé, plus la fourmi a de chances de ramasser ce cadavre. La probabilité pour une fourmi porteuse de déposer ce qu'elle transporte suit une règle inverse : plus le monticule observé est important, plus la probabilité de déposer le corps au sol sera grande.

Les premières modélisations algorithmiques de ces comportements collectifs et auto-organisés sont dues à de Deneubourg et son équipe [Deneubourg, 1990] et furent après repris par d'autres chercheurs pour donner naissance à différents algorithmes pour la résolution de problèmes par stigmergie. Nous détaillerons plus amplement ces algorithmes de fourmis qui vont être la base de notre travail dans le chapitre suivant.

## 2.6 Conclusion

Intuitivement, Le phénomène d'émergence peut être défini comme étant l'apparition d'une propriété au niveau macroscopique d'un système sans qu'elle soit préalablement programmée d'une manière explicite ni qu'elle puisse déduite à partir des propriétés des niveaux microscopiques. Un moyen de mise en œuvre de l'émergence est l'auto-organisation qui fait référence à un processus au cours duquel le système se restructure, se maintient sans nécessiter une contrainte explicite qui provient de l'extérieur du système.

Avec la complexité croissante des systèmes informatiques et le besoin d'avoir des systèmes adaptatifs et dynamiques, les méthodes classiques de résolution de problèmes sont devenues inefficaces. Ce qui a conduit les chercheurs à explorer de nouvelles voies et de nouveaux outils de développement. D'où l'utilisation de la notion d'émergence pour la conception de nouveaux types de systèmes artificiels. L'objectif est de concevoir des systèmes complexes constitués de petites entités en interaction entre elles et avec leur environnement et dont le comportement global est émergent et peut être qualifié d'intelligent.

Ces systèmes s'inspirent en large partie de l'observation des systèmes naturels et en particulier du comportement de groupes et d'animaux sociaux. Parmi les systèmes artificiels collectifs à fonctionnalité émergente, on trouve les algorithmes évolutionnaires et leurs variantes, les réseaux de neurones formels ainsi que les algorithmes inspirés des comportements collectifs des animaux sociaux tels que les algorithmes d'optimisation par essaim particulaires qui simulent les comportements d'essaim observés dans les bancs de poissons ou les vols d'oiseaux et les algorithmes de fourmis artificielles qui miment les comportements des fourmis pour la résolution des problèmes complexes et à s'auto-organiser par stigmergie. Le chapitre suivant présentera en détail les différents algorithmes de fourmis artificielles dans la littérature ainsi que leurs domaines d'application.

# Chapitre 3

## Les Fourmis artificielles

<b>Chapitre 3. Les Fourmis artificielles .....</b>	<b>46</b>
3.1 Introduction .....	46
3.2 Généralités sur les fourmis .....	46
3.3 Quelques concepts de base .....	50
3.4 Les algorithmes de fourmis artificielles pour l'optimisation combinatoire .....	51
3.4.1 Inspiration biologique : le fourragement collectif par stigmergie .....	51
3.4.2 Les expériences .....	52
3.4.3 Modèles de fourragement pour la résolution des problèmes d'optimisation ...	54
3.4.4 Autres domaines d'application .....	58
3.4.5 La métaheuristique « optimisation par les colonies de fourmis » .....	59
3.5 Les algorithmes de fourmis artificielles pour la classification automatique .....	62
3.5.1 Inspiration biologique .....	62
3.5.2 Modèles du tri du couvain et application à la classification de données .....	62
3.5.3 Autres domaines d'application .....	67
3.5.4 Autres sources d'inspiration .....	67
3.6 Conclusion .....	68

---

## Chapitre 3. Les Fourmis artificielles

---

### 3.1 Introduction

Les études éthologistes ont montré que dans la nature, les petites créatures faibles que sont les fourmis, arrivent à résoudre collectivement des problèmes quotidiens nombreux et trop complexes pour une seule fourmi tels que : recherche de nourriture, construction du nid, division du travail et allocation des tâches entre les individus, avec une organisation excrément structurée et sans aucune supervision. Par les comportements simples de chacune des fourmis, des interactions limitées à travers une coopération inconsciente, émergent des comportements collectifs intelligents et des modèles d'auto-organisation [Bonabeau, 2000]. Les fourmis sont devenues dès lors une nouvelle source d'inspiration pour la conception de méthodes de résolution de problèmes complexes. De plus cette source d'inspiration n'est pas unique étant donné que les fourmis sont dotées d'une grande diversité de caractéristiques disjointes et de comportements collectifs variés. Une nouvelle classe d'algorithmes est alors apparue sous le nom « *algorithmes de fourmis artificielles* ». Leur popularité est due d'une part à la facilité de mise en œuvre et d'autre part à la complexité des fonctions réalisables [Deneubourg, 1990; Coloni, 1992; Dorigo, 1996; Bonabeau, 1997; Van Dyke Parunak, 1997; Bonabeau, 1999; Topin, 1999a]. Deux comportements collectifs ont été principalement étudiés chez les fourmis : l'optimisation de chemin et le tri des cadavres.

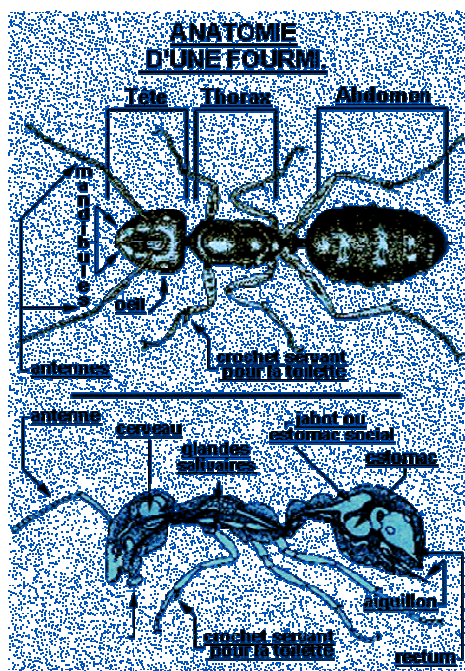
Le premier comportement appelé aussi fourrageage permet aux fourmis de retrouver le plus court chemin entre leur nid et une source de nourriture grâce à un système de marquage de phéromones. Ce comportement naturel a été modélisé et transposé à la résolution de nombreux problèmes d'optimisation combinatoires sous le nom d'une nouvelle métaheuristique « optimisation par les colonies de fourmis ou OCF ». Le deuxième comportement collectif des fourmis concerne la capacité de certaines espèces de fourmis à organiser collectivement des cimetières composés de cadavres empilés les uns sur les autres. Là aussi, les chercheurs ont exploité ce comportement pour fournir des algorithmes de classification pour lequel l'informatique classique n'a pas donné de solution satisfaisante.

Dans la suite nous présentons une brève introduction au monde des fourmis, ensuite nous décrirons en détail chacun des modèles de fourmis artificielles ainsi que les différents algorithmes qui lui sont associés.

### 3.2 Généralités sur les fourmis

Avec les abeilles et les guêpes, les fourmis font partie de la famille des Formicidae, incluse dans l'ordre des Hyménoptères, c'est à dire des insectes dont les deux paires d'ailes sont

membraneuses et fines. Les fourmis sont réparties en 11 sous-familles, et approximativement 10 000 espèces. La fourmi peut mesurer de quelques millimètres à plusieurs centimètres de long. Son corps qui se compose de 3 parties principales : la tête, le thorax et l'abdomen (figure 5.1). La tête est ornée d'une paire d'antennes qui abritent plusieurs organes des sens : le toucher, l'odorat et le goût. On trouve également à l'extrémité de la tête, la bouche qui sert à la fois à l'alimentation et très souvent d'outil de saisie. Les yeux, au nombre de 2, sont composés et formés d'une multitude de facettes. Quelques fourmis sont complètement aveugles, sans aucun organe visuel. Le thorax, ou corselet, porte 3 paires de pattes, très longues et terminées par 2 griffes. Les organes tactiles sont bien développés et sont constitués par des poils répartis sur tout le corps. Leur densité est plus forte sur les antennes et aux extrémités des pattes. L'abdomen est relié au thorax par un étroit pédoncule ou pétiole. L'abdomen renferme le jabot social, l'estomac, l'intestin et les conduits respiratoires, et chez les reines et les mâles, l'appareil reproducteur. La fourmi a un estomac un peu spécial, le "jabot social". Elle y garde en réserve de la nourriture, qui peut être régurgitée et transmise aux autres fourmis. La carapace (ou exosquelette) est composée d'une substance, résistante et imperméable que l'on nomme chitine. Elle renferme différents pigments, bruns, rouges, ou noirs qui sont responsables de la couleur de l'animal.



**Figure 3.1. Anatomie d'une fourmi**

Selon les espèces, les fourmis vivent dans le sol, sous un caillou, dans le bois mort ou dans les arbres. Certaines fourmis arboricoles construisent leurs nids dans les arbres, en cousant les feuilles entre elles ou en élaborant une architecture faite de grains de sable, de salive et de déjections. D'autres, comme les célèbres fourmis "légionnaires" d'Afrique, n'ont pas de nid et

se déplacent par centaines de milliers d'individus sur de vastes territoires, à la manière des nomades.

La fourmi est un insecte exclusivement social vivant en société dans des habitations collectives : la fourmilière. La fourmilière classique est constituée par un ensemble de cellules réunies entre elles par un réseau complexe de galeries qui peut être très important. Dans ces cellules, les fourmis déposent les oeufs, les larves et les nymphes. *La fourmilière* et ses environs constituent le centre de la vie communautaire. A l'intérieur de la fourmilière les tâches sont divisées entre les fourmis selon la spécialité de chacune d'elles. Les activités des communautés de fourmis sont caractérisées par un certain degré de division du travail souligné par une différenciation fonctionnelle et anatomique des individus. Une fourmilière peut abriter de 50.000 à plus de 1.000.000 individus bien différenciés tant au niveau physique qu'au niveau des comportements et des tâches à accomplir. On les sélectionne en castes :

**Les reines** : Dans une fourmilière on trouve une ou plusieurs reines. Les reines sont nettement plus grosses que les autres fourmis et peuvent vivre jusqu'à dix ou quinze ans. Leur rôle se résume essentiellement à pondre des œufs et sont donc les fondatrices de nouvelles colonies..

**Les ouvrières** : Elles forment la majorité des habitants de la cité et se chargent de la défense et de l'entretien de la colonie, qui comprend la construction des galeries, les soins apportés aux jeunes, la quête de la nourriture, etc...

**Les soldats** : Ils sont plus massifs que les ouvrières, et possèdent souvent de grosses mandibules. Leur rôle est de défendre la fourmilière, et de transporter des charges lourdes. Mais certains, comme chez les fourmis "Grand Galop" *Camponotus maculatus*, participent aux soins des larves, et donnent à manger aux fourmis qui le demandent. Chez cette espèce, qui est la plus grosse fourmi vivant à La Réunion, on peut distinguer des formes intermédiaires entre la petite ouvrière grêle et le puissant soldat.

**Les jeunes sexués** ce sont des fourmis femelles et mâles. Ils sont facilement reconnaissables par leur plus grande taille, par la présence de deux paires d'ailes membraneuses sur le thorax et par 3 ocelles disposés en triangle sur le dessus de la tête. Les femelles sont plus grosses que les mâles. Ce sont les futurs rois et reines qui iront fonder de nouvelles colonies.

**Le couvain** Il est constitué par les oeufs, les larves et les nymphes. Au bout de quelques jours les oeufs donnent naissance à des larves qui, bien nourries par les ouvrières pendant 15 jours à 3 semaines, se transforment en nymphes. Pendant la nymphose, la larve ne se nourrit plus.



Son corps tout entier subit de profondes mutations internes et externes, qui vont faire d'elle, petit à petit, une fourmi.

Les communications interindividuelles entre fourmis sont de plusieurs types et varient d'une espèce à l'autre. Les principaux moyens de communication sont :

*La communication tactile* : lorsque deux fourmis se rencontrent, elles procèdent à quelques attouchements d'antennes ou de pattes, et se reconnaissent aussitôt comme membres de la même fourmilière ;

*La communication sonore* : les fourmis peuvent également utiliser des stimuli vibratoire comme moyen de communication. Elles frappent les parois de la fourmilière avec leur abdomen pour prévenir les autres. Elles tapotent aussi leurs antennes pour se parler. Ce signal est utilisé en fonction de l'espèce comme signal de détresse en cas de danger, comme signal de qualité de l'alimentation pour le recrutement pour une source de nourriture, comme signal de demande d'aide en cas où la nourriture trouvée est de grande taille..

*La communication visuelle* : développée chez certaines espèces, permet aux fourmis de retrouver leur chemin, guidées par des marqueurs visuels.

*La communication chimique* : les fourmis sont équipées de glandes produisant des phéromones, substances chimiques volatiles et odorantes qu'elles peuvent sentir par leurs antennes. Ce signal chimique porte l'information à la fois sur l'espèce, la société mais aussi la caste et le stade de développement auxquelles appartiennent les fourmis rencontrées. C'est en sécrétant cette substance qu'une fourmi éclairceuse marque le chemin qu'elle a utilisé et revient avertir ses pairs de la présence de nourriture ou d'un danger, ainsi que de sa localisation. C'est le moyen de communication le plus efficace et plus utilisé par toutes les espèces de fourmis

La totalité des espèces de fourmis vit en effet en communautés<sup>49</sup> plus ou moins importantes, qui sont qualifiées d'"eusociales" par [Wilson, 1985], ce qui signifie qu'elles sont caractérisées par la présence simultanée et constante des trois critères suivants:

- La coopération des membres de la société pour élever les jeunes et le couvain ;
- L'existence d'une division des rôles, d'abord au niveau de la fonction reproductrice, ensuite au niveau des tâches à accomplir ;

- Le chevauchement d'au moins deux générations d'âge différent capables de contribuer aux travaux de la colonie.

### 3.3 Quelques concepts de base

Avant de s'intéresser aux algorithmes de fourmis artificielles, il convient tout d'abord de présenter quelques concepts de base qui seront utilisés tout au long de cette section.

#### 3.3.1 Problème d'optimisation

Un problème *d'optimisation* est tout problème défini par un espace de recherche des solutions, une fonction objectif qui associe un coût à chaque solution possible et un ensemble de contraintes. On cherche alors à trouver la solution optimale qui correspond à une solution de coût minimum ou maximum selon qu'il s'agit de minimiser ou de maximiser la fonction objectif.

Un problème *d'optimisation combinatoire* est tout problème d'optimisation pour lequel il faut trouver une solution optimale avec un espace de recherche de solutions fini mais extrêmement grand. Ce type de problème est dit « difficile ».

#### 3.3.2 Méthodes de résolution

Les méthodes de résolution des problèmes d'optimisation sont de deux types :

- Les méthodes exactes (déterministes) : elles fournissent une solution optimale au prix d'un temps de résolution qui risque d'être exponentiel en fonction de la taille des données du problème.
- Les méthodes approchées : pour un problème d'optimisation dit « difficile » aucune méthode exacte n'est capable de le résoudre exactement en un temps raisonnable. Dans ce cas on fait appel à ses méthodes permettant une optimisation approchée. Ce type de méthodes retourne une solution contenue dans un certain intervalle autour de la solution optimum avec un temps de calcul acceptable. Elles représentent un compromis entre la qualité de la solution trouvée et le temps de calcul nécessaire.

Parmi les méthodes de résolution approchées, on trouve :

#### ***a) Les heuristiques***

Une heuristique est une méthode approchée simple, rapide et dédiée à un problème donné. Elle exploite les propriétés structurelles d'une solution et tente de la rendre rapidement une

solution admissible par des critères de décision déduits de la connaissance du problème. Aucune garantie quant à l'optimalité de la solution trouvée ne peut être fournie.

### ***b) Les métaheuristiques***

Une métaheuristique est une méthode approchée générique dont le principe de fonctionnement repose sur des mécanismes généraux indépendants de tout problème. Les métaheuristiques sont stochastiques et donc peuvent éviter d'être piégés dans des minimums locaux. Elles sont principalement guidées par le hasard (exploration aléatoire de l'espace de recherche), cependant elles sont souvent alliées à d'autres algorithmes afin d'en accélérer la convergence.

#### 3.3.3 Les fourmis artificielles

Une fourmi artificielle est une entité simple dotée d'un comportement similaire ou étendu à celui de la fourmi réelle. Ce comportement doit être élémentaire, restreint et donc facile à programmer. A l'intérieur d'une colonie, les fourmis sont concurrentes et asynchrones, elles coopèrent inconsciemment ensemble pour la résolution du problème considéré. Les fourmis artificielles communiquent entre elles indirectement par stigmergie via des modifications de leur environnement (par exemple par dépôt de traces de phéromone artificielle) qui représente la mémoire collective de la colonie. Elles ont été de plus enrichies des contraintes et de comportements qu'on ne trouve pas dans leurs congénères réelles mais qui sont spécifiques au problème qu'elles résolvent.

### **3.4 Les algorithmes de fourmis artificielles pour l'optimisation combinatoire**

Dans la nature, les fourmis arrivent à résoudre différents problèmes d'optimisation liés à leur survie. Cette capacité naturelle a été simulée et transposée pour la résolution de divers problèmes d'optimisation combinatoire. Les premiers travaux dans ce sens sont ceux de Dorigo et ses collègues qui ont mimé le comportement collectif de fourrageage observé chez les fourmis réelles et l'ont appliqué pour traiter le problème du voyageur du commerce [Colomi, 1991]. Les bons résultats obtenus ont incité de nouveaux chercheurs à l'utiliser en lui ajoutant certaines extensions que nous allons présenter dans la suite.

#### **3.4.1 Inspiration biologique : le fourrageage collectif par stigmergie**

Les études éthologistes ont montré que dans la nature, les fourmis utilisent la stigmergie pour mener à bien la tâche de recherche de nourriture appelée aussi fourrageage. En se déplaçant de leur nid à la recherche d'une source de nourriture, la fourmi laisse sur le chemin qu'elle emprunte une substance chimique de nature volatile appelée « phéromone ». Par ce marquage naturel, elle incite ses congénères à suivre le même trajet. Au début, les fourmis explorent

différents chemins en effectuant des déplacements aléatoires. Une fois qu'un chemin intéressant (menant à une source de nourriture) est découvert, elles y déposent une quantité de phéromone renforçant ainsi son importance et la probabilité d'être choisi par d'autres fourmis de la colonie. D'un autre côté, les mauvais chemins auront tendance à être oubliés voir même disparaître avec l'évaporation de la phéromone. Ce procédé basé sur le mécanisme de rétroaction positive, assure que pendant le fourragement pour la nourriture, les fourmis utilisent la voie d'accès la plus courte car elle sera le plus imprégnée par la phéromone.

### 3.4.2 Les expériences

Pour bien comprendre ce comportement, des chercheurs tels que Deneubourg et ses collègues [Goss, 1989; Deneubourg, 1990] ont mené différentes expériences et ont permis l'élaboration d'un modèle comportemental mathématique qui a permis la naissance d'un panel d'algorithmes d'optimisation qui ont connu un grand succès.

#### 3.4.2.1 Le pont binaire

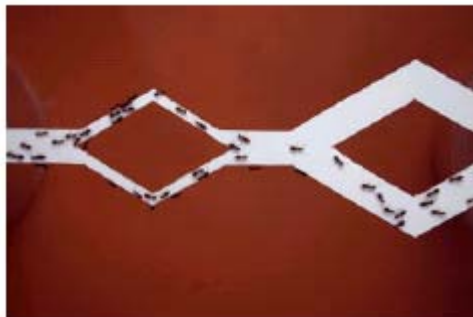
L'expérience la plus connue est celle du pont binaire de Deneubourg [Deneubourg, 1990], dans laquelle, le nid des fourmis et la source de nourriture sont séparés par un pont binaire constitué de deux branches identiques (figure 3.2). Comme au départ, il n'y a aucune trace de phéromone sur les deux branches, chacune peut être choisie avec la même probabilité (figure 3.3.). Après une phase de fluctuations, la quantité de phéromone a tendance à s'accumuler sur un chemin et il a été observé alors que les fourmis prenaient collectivement le même itinéraire après quelques minutes (figure 3.4.).



Figure 3.2. Pont Binaire de Deneubourg



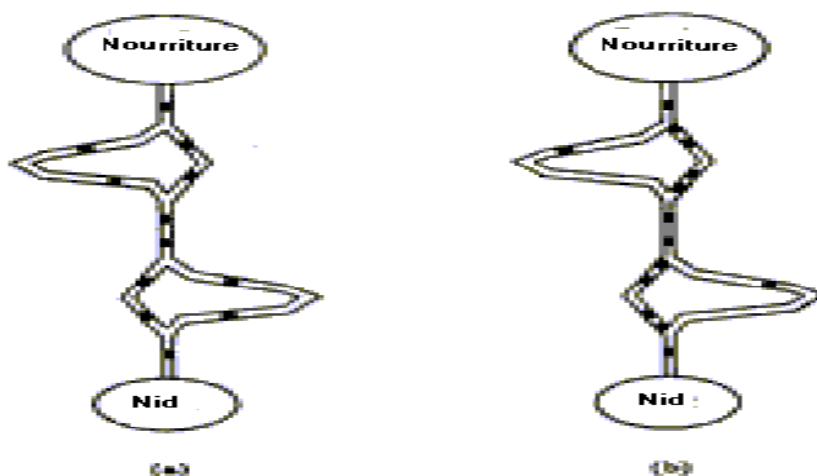
Figure 3.3. Les fourmis suivent indifféremment les deux branches du pont.



**Figure 3.4.** Les fourmis ont tendance à emprunter le même chemin (celui d'en bas).

#### 3.4.2.2 Le pont binaire à longueurs inégales

Dans une deuxième expérience, les chercheurs ont étudié l'effet de l'utilisation d'un pont avec des longueurs inégales (figure 3.5.). En raison du même mécanisme de dépôt de phéromone, les fourmis qui ont pris la branche la plus courte arrivent à la source de nourriture en premier, et retournent (le long du même chemin) en premier au nid augmentant de ce fait les quantités de phéromone sur ce plus court chemin, renforçant ainsi le choix des autres fourmis.

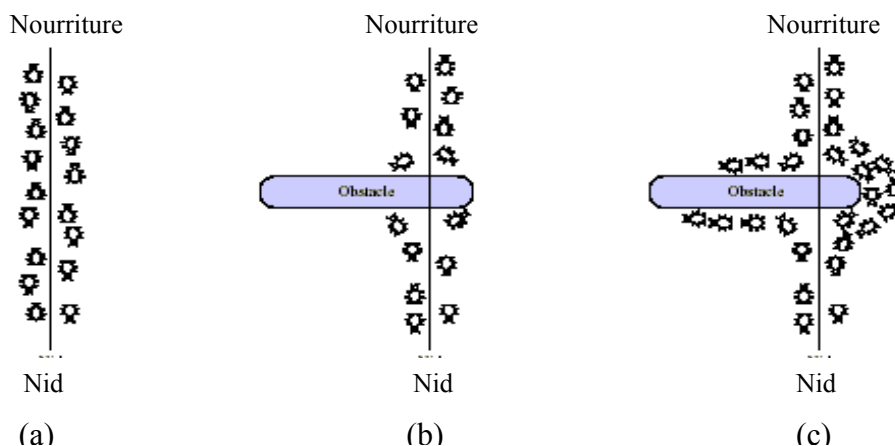


**Figure 3.5.** Expérience du pont avec des branches de taille différentes

#### 3.4.2.3 Le pont avec un obstacle

Dans une troisième expérience, alors que les fourmis suivent leur chemin (figure 3.6.a), un obstacle est placé afin de leur barrer la route entre le nid et la source de nourriture (figure 3.6b.). Les fourmis doivent alors contourner cet obstacle par la droite ou par la gauche afin de poursuivre leur chemin. Comme initialement, il n'y a aucune trace de phéromone le long de l'obstacle, la moitié des fourmis passent par le côté gauche et l'autre moitié par le côté droit. Il y'aura autant de fourmis (figure 3.6b). Après un certain temps, les chercheurs ont observé que les fourmis qui ont choisi le chemin le plus court, parviennent à reconstituer plus

rapidement la concentration en phéromones sur ce nouveau chemin que celles qui ont choisi le chemin le plus long. Ainsi la concentration de phéromone sur le plus court chemin va augmenter, incitant ainsi d'autres fourmis à choisir le chemin riche en phéromones. Du à ce processus auto-catalytique, les fourmis vont finalement choisir le chemin le plus court (figure 3.6c).



**Figure 3.6. Effet de dépôt d'un obstacle entre la source de nourriture et le nid**

### 3.4.3 Modèles de fourragement pour la résolution des problèmes d'optimisation

Se basant sur les travaux de Deneubourg, Marco Dorigo et ses collègues furent les premiers à modéliser le comportement de fourragement et à l'appliquer pour résoudre un problème d'optimisation classique : le problème du voyageur du commerce [Dorigo, 1991; Colomi 1992; Dorigo 1992]. Cette première modélisation s'est depuis très popularisée et a donné naissance à de nouveaux algorithmes appliqués à d'autres types de problèmes pour lesquels il est difficile de trouver une solution exacte. Cette section présente les différents algorithmes inspirés du comportement des stratégies de recherche de nourriture des fourmis réelles pour résoudre divers problèmes d'optimisation.

#### 3.4.3.1 Principe général : l'algorithme Ant System (AS)

Ant System (AS) est le premier algorithme de fourmi reposant sur le comportement de fourragement des fourmis et appliqué pour la résolution du problème du voyageur de commerce (PVC) (TSP en anglais) [Dorigo, 1996].

Le problème du voyageur de commerce, consiste à trouver un chemin Hamiltonien dans un graphe complètement connecté. Il s'agit pour un voyageur de commerce de trouver le chemin le plus court pour visiter une et une seule fois chacune des  $n$  villes dans lesquelles il doit se rendre. L'espace de recherche est l'ensemble des combinaisons possibles des  $n$  villes. Il s'agit sans doute du problème d'optimisation combinatoire NP-complet le plus utilisé comme test pour les nouvelles méthodes d'optimisation.

Le PVC est modélisé par un graphe  $G(V,E)$  où  $E$  est l'ensemble des nœuds représentant les villes à visiter, et  $V$  est l'ensemble des arêtes. Une arête existe entre deux nœuds du graphe s'il existe un chemin reliant les deux villes qu'ils représentent. On utilise alors un graphe valué dont les arêtes sont étiquetées par la distance séparant deux villes.

Initialement,  $m$  fourmis sont placées aléatoirement sur les nœuds du graphe. Ensuite chacune des fourmis se déplace d'un nœud à un autre en parcourant les arêtes du graphe. Ce déplacement dépend de la liste des villes déjà visitées représentant la mémoire de la fourmi et d'une probabilité fonction de la distance reliant les villes et de la quantité de phéromone présente sur les arêtes du graphe.

L'algorithme AS est constitué d'un nombre d'itération appelée « cycle ». A chaque cycle, chaque fourmi  $k$  ( $k = 1 \dots m$ ) parcourt le graphe en se déplaçant d'un nœud vers un autre. Le choix du passage d'un nœud  $i$  à un nœud  $j$  se fait aléatoirement en fonction d'une probabilité donnée par l'équation 3.1.

$$P_{ij}^k(t) = \begin{cases} \frac{(\tau_{ij}^k)^\alpha (\eta_{ij}^k)^\beta}{\sum_{l \in J_i^k} (\tau_{il}^k)^\alpha (\eta_{il}^k)^\beta} & \text{si } j \notin J_i^k \\ 0 & \text{sinon} \end{cases} \quad (3.1)$$

Où

- $\eta_{ij} = \frac{1}{d_{ij}}$  appelée « visibilité » représente l'heuristique de la désirabilité *innée* de choisir  $j$  à partir de la ville  $i$ .
- $\tau_{ij}(t)$  est la quantité de phéromone présente sur l'arête  $(i,j)$ . Cette information dynamique représente la désirabilité *acquise* de sélectionner  $j$  comme destination de  $i$ .
- $J_i^k$  est la liste de villes visitées par la fourmi  $k$  placée sur le nœud  $i$ . Cette mémoire joue un double rôle : d'une part elle empêche la fourmi de retourner sur des villes déjà visiter et d'autre part elle permet la sauvegarde le chemin parcouru par la fourmi afin de déposer les phéromones à la fin de chaque cycle.
- $\alpha$  et  $\beta$  sont des constantes qui servent à régler l'importance relative que l'on donne à l'heuristique et à la phéromone. Avec  $\alpha = 0$  seule la visibilité de la ville est prise en compte et la ville la plus proche est donc choisie à chaque déplacement. Avec  $\beta = 0$  seule les traces de phéromones sont prises en compte. Un compromis entre

ces deux types d'information est nécessaire et influence les comportements de diversification et d'intensification.

Après  $n$  itérations, lorsque toutes les fourmis ont construit un chemin complet, chacune d'elles calcule la longueur totale parcourue  $L^k(t)$  (somme des distances d'une ville à l'autre). Les quantités de phéromones sont mises à jour :

$$\tau_{ij}^{t+1} = \tau_{ij}^t (1 - \rho) + \sum_{k=1}^m \Delta \tau_{ij}^k(t) \quad (3.2)$$

où  $\rho$  est un coefficient d'évaporation des traces de phéromones présentes sur les arêtes.

$\Delta \tau_{ij}^k(t)$  représente le cumul de phéromone que chaque fourmi dépose sur l'arête qu'elle a suivie entre l'itération  $t$  et  $t+1$ . Cette valeur est inversement proportionnelle à la longueur du chemin construit par la fourmi: plus le chemin est court, plus il sera chargé en phéromones. Cette valeur est donnée par :

$$\Delta \tau_{ij}^k(t) = \begin{cases} \frac{Q}{L^k(t)} & \text{si } (i,j) \in T^k(t) \\ 0 & \text{sinon} \end{cases} \quad (3.3)$$

où  $T^k(t)$  est le trajet effectué par la fourmi  $k$  à l'itération  $t$  et  $Q$  est un paramètre constant.

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

#### Algorithme 3.1. L'Algorithme AS-TSP

Pour  $t = 1, \dots, t_{max}$

    Pour chaque fourmi  $k = 1, \dots, m$

        Choisir une ville au hasard

        Pour chaque ville non visitée  $i$

            Choisir une ville  $j$ , dans la liste  $J_i^k$  des villes restantes, selon la formule 3.1

        Fin Pour

        Déposer une piste  $\Delta_{ij}^k(t)$  sur le trajet  $T^k(t)$  selon l'équation 3.2

    Fin Pour

    Evaporer les pistes selon la formule 3.3

Fin Pour

Cette façon d'explorer l'espace de recherche a le mérite d'être efficace et d'offrir un compromis entre l'exploitation de l'information apportée par la collectivité (les phéromones



préalablement déposées) et l'utilisation des heuristiques locales pour l'exploration aléatoire de l'espace des solutions.

### 3.4.3.2 Variantes de Ant System

Les bons résultats obtenus par AS pour la résolution du PVC ont encouragé les chercheurs au développement d'autres algorithmes basés fourmis en introduisant un certain nombre d'extensions qui ont touché en particulier les aspects suivants :

- Le processus de génération de solution s'est vu intégrer une phase d'amélioration itérative afin de fournir des solutions de meilleure qualité. L'hybridation ainsi obtenue fournit souvent de bien meilleurs résultats [Dorigo, 1997a ; Stuzle, 1999] à condition de maîtriser la perte inévitable de diversité.
- L'actualisation des traces a évolué pour permettre l'application explicite de stratégies de fouille ou de prospection. Ainsi, dans le premier cas, seules les meilleures solutions donnent lieu à une mise à jour des traces, de façon à favoriser la réutilisation des attributs qu'elles contiennent. Dans le second cas, cette mise à jour est négative, afin de privilégier l'utilisation d'attributs jusque-là délaissés
- La coordination des différentes fourmis a été sensiblement améliorée par l'apparition d'une fourmi particulière, la reine [Taillard, 1999]. Cette dernière prend en charge la gestion de la mémoire collective de la colonie (les traces) et effectue les décisions stratégiques (fouille, prospection) en fonction de l'évolution globale de la recherche. Elle constitue de fait une sorte d'agent "intelligent"

Dans la suite de cette section, nous présentons les deux variantes de AS les plus connues.

#### ***Ant Colony System « ACS »***

L'algorithme « Ant Colony System » a été introduit par « Dorigo » et « Gambardella » en 1996 pour améliorer la performance de AS [Dorigo, 1997b]. ACS est basé essentiellement sur AS mais se distingue de lui par les points suivants :

- Le déplacement de la fourmi suit une autre règle de transition dite règle proportionnelle pseudo-aléatoire ;
- Deux méthodes sont utilisées pour la mise à jour :

- Une mise à jour locale est effectuée à chaque fin de cycle d'une fourmi.
- Une mise à jour globale est faite une fois que toutes les fourmis ont terminé leurs cycles. Seule la fourmi qui a trouvé la meilleure solution est autorisée à renforcer la phéromone sur tous les arcs constituant son tour.

La mise à jour globale évite de se bloquer dans des solutions sous optimal (minimums locaux). Tandis que la mise à jour locale a pour effet de réduire, de moins en moins, l'interactivité des arcs déjà visités par d'autres fourmis, et donc de favoriser l'émergence de d'autres solutions que celle déjà trouvées pendant les prochains cycles de l'algorithme.

### ***Max-Min Ant System***

Dans [Stutzle, 1999; Stützle, 2000] Stutzle et Hoos introduisent MMAS algorithm. Les modifications introduites concernent :

- L'utilisation de deux constantes  $\tau_{\min}$  et  $\tau_{\max}$  comme borne inférieure et supérieure à la quantité de phéromone présentent sur les arcs du graphe. Ces deux valeurs permettent de limiter les variations des taux de phéromone et éviter ainsi que certains arcs soient totalement délaissés au profit d'autres ce qui est reconnu comme un état de stagnation prématurée. Toutes les traces de phéromones sont initialisées à  $\tau_{\max}$ .
- Les quantités de phéromones sont initialisées à la valeur maximale ;
- La mise à jour des traces de phéromones n'est autorisée que par la fourmi ayant trouvé la meilleure solution.

### **3.4.4 Autres domaines d'application**

Les bonnes performances des algorithmes basés fourmis obtenues lors de leur application au problème du voyageur du commerce ont incité beaucoup de chercheurs à les utiliser dans d'autres domaines d'application. Sans vouloir dresser une liste exhaustive de toutes les applications et variantes qui ont été produites, on peut citer le problème d'affectation quadratique [Taillard, 1997; Maniezzo, 1994], le problème de coloration de graphe [Costa, 1997; Dudot, 2005], le problème de routage [Bullnheimer, 1997], les réseaux de communication [Di Caro, 1998], le problème d'ordonnancement [Gambardella, 1997; Merkloe, 2002], les problèmes de satisfaction de contraintes [Solnon, 2002], la fouille de données [Parpinelli, 2002], l'optimisation de site d'enseignement en ligne (notion d'« E-

Learning »)[Semet, 2003]. Une bonne synthèse de ces algorithmes est reportée dans [Dorigo, 2004].

### 3.4.5 La métaheuristique « optimisation par les colonies de fourmis »

Les algorithmes qu'on vient de décrire ont été regroupés sous le terme général de « métaheuristique par les colonies de fourmis (OCF) » ou « Ant Colony Optimization (ACO) » [Dorigo, 2004]. Dans OCF, le voisinage du nid des fourmis est associé à l'espace des solutions, telle que chaque solution s'apparente à une source de nourriture dont la qualité est fournie par une fonction objectif. Chaque fourmi est assimilée à un processus stochastique de construction de solutions itératif. La construction est biaisée par une phéromone artificielle et une information heuristique propre au problème à résoudre. Les traces de phéromones sont typiquement la mémoire des fourmis sur les éléments de la solution, et sont régulièrement mise à jour par les processus de construction (en fonction de la qualité de la solution) et par un mécanisme simulant l'évaporation de la phéromone.

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

#### 3.4.5.1 Représentation du problème

Un problème d'optimisation  $O = \{S, f, \Omega\}$  est représenté par un ensemble de solutions  $S$ , une fonction objectif  $f$  assignant une qualité à une solution et un ensemble de contraintes  $\Omega$ . L'objectif est de trouver une solution optimale  $f_{\min}(s), \forall s \in S$  qui satisfait les contraintes  $\Omega$ .

Suivant cette représentation, les fourmis construisent les solutions possibles en se déplaçant sur un graphe  $G = (C, L)$  associé au problème d'optimisation, tel que les nœuds sont les composantes  $C$  du problème qui forment les solutions et les arcs sont les transitions possibles  $L$  qui connectent les composantes de  $C$ , telle que à chaque transition on associe une fonction de coût. Les contraintes du problème sont implémentées directement dans les règles de déplacement des fourmis. Par conséquent, la solution optimale représente le chemin de Coût minimum.

#### 3.4.5.2 Construction des solutions

Les fourmis artificielles construisent les solutions en ajoutant selon un processus stochastique et itératif les séquences de composants à partir de  $C$ . Chaque fourmi construit une solution possible (même de mauvaise qualité) d'une manière stochastique en se déplaçant dans le graphe  $G = (C, L)$ . La solution optimale émergera de la coopération entre les fourmis de la colonie.

Chaque connexion  $(c_i, c_j) \in L(\text{arc})$  peut être pondérée par une valeur de phéromone artificielle  $\tau_{ij}$  représentant une mémoire collective de la colonie de fourmis et permettant les interactions entre les fourmis (stigmergie), et si elle existe une valeur heuristique  $\eta_{ij}$  qui représente une information a priori propre au problème à résoudre. Pour certains problèmes, les traces de phéromones et la valeur de l'heuristique peuvent aussi être associées aux composants du graphe.

A chaque fourmi  $k$  on associe un état de départ  $s_{\text{départ}}^k$  (nœud de départ du graphe) et des critères de terminaison  $e^k$ , en plus d'une mémoire  $T_k$  qui stocke les informations sur les composants qu'elle a ajouté à sa solution. Cette mémoire peut être utilisée

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

La construction de la solution par une fourmi  $k$  se fait d'une manière incrémentale en ajoutant à chaque déplacement un nouveau composant. Le processus de construction se termine pour la fourmi  $k$  si au moins un des critères de terminaison  $e^k$  est vérifié.

Quand la fourmi  $k$  est sur un nœud  $C_i$  elle choisit un prochain nœud  $C_j$  parmi les nœuds possibles dans un voisinage accessible  $N_i^k$ . Ce déplacement est choisi en appliquant une règle de décision probabiliste. Cette règle est soit fonction

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

Une fois la solution construite, les fourmis mettent à jour la trace de phéromone des transitions (ou composant) qu'elles ont utilisé. La mise à jour des traces de phéromone peut se faire de deux manières :

- A chaque déplacement d'un nœud  $C_i$  vers un nœud  $C_j$  la fourmi laisse une trace de phéromone sur l'arc de transition  $(C_i, C_j)$  on parle de *mise à jour de phéromone en ligne pas à pas*
- Quand la fourmi a terminé la construction d'une solution, elle refait son chemin à l'envers et mets à jour la trace de phéromone des arcs appartenant à sa solution. On parle de *mise à jour de phéromone en ligne retardée*.

En plus de ces règles de comportement des fourmis, un autre processus est introduit : l'évaporation des traces de phéromones qui permet de décroître les quantités des traces de phéromones et donc évite que certains arcs soient plus favorisés que d'autres à cause de la forte concentration de phéromone présente sur eux, ce qui peut amener à une convergence rapide vers une région sous-optimale de l'espace de recherche.

Le principe général de OCF est résumé dans l'encadré suivant :

<p><i>Algorithme 3.2. Principe général de la métaheuristique « OCF »</i></p> <p><i>Initialisation de la trace de phéromone</i></p> <p><i>Tant qu'un critère d'arrêt n'est pas atteint faire</i></p> <p style="padding-left: 40px;"><i>Construire une nouvelle solution à partir de la trace ;</i></p> <p style="padding-left: 40px;"><i>Mettre la trace à jour à partir de cette solution et de sa qualité.</i></p> <p><i>Fin tant que</i></p> <p><i>Retourner la meilleure solution trouvée.</i></p>
---

Les algorithmes de fourmis présentent les caractéristiques suivantes [Dorigo, 1996]:

- Ils sont flexibles : ils peuvent être facilement réutilisables pour des versions modifiées d'un même problème ;
- Ils sont robustes et nécessitent peu de changement pour leur application à de nouveaux problèmes d'optimisation combinatoires ;
- C'est des approches basées population : ce qui permet d'exploiter le caractère autocatalytique de la méthode en tant que mécanisme de recherche. D'un autre côté, une implémentation parallèle de la méthode est naturelle.

### **3.5 Les algorithmes de fourmis artificielles pour la classification automatique**

Les algorithmes de classification automatique sont un autre type d'algorithmes inspirés de comportements collectifs observés chez les fourmis : le tri collectif de couvains ou la constitution de cimetières. Les premiers travaux dans ce domaine ont été ceux de Deneubourg et son équipe [Deneubourg, 1990], se basant sur une colonie de fourmis artificielles qui se déplacent aléatoirement sur une grille rectangulaire et sont capables de ramasser et de déposer des objets présents sur une grille dans le but de les regrouper selon un critère de similarité. Ces travaux ont été par la suite améliorés et étendus à différents domaines d'application.

#### **3.5.1 Inspiration biologique**

Certains travaux ont montré que certaines espèces de fourmis parviennent à organiser divers éléments du couvain tels que les œufs, les larves [Deneubourg, 1990; Deneubourg, 1991]. Ce comportement est basé sur deux règles simples mais suffisantes pour former des tas d'éléments de même type et qui sont les suivantes :

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

Là aussi le tri du couvain est le résultat du processus de rétroaction positive et l'utilisation d'informations locales à chaque fourmi comme on le verra dans les prochaines sections.

#### **3.5.2 Modèles du tri du couvain et application à la classification de données**

Ces observations ont conduit à différentes modélisations qui avaient pour objectif de simuler et de reproduire ce comportement collectif et auto-organisé des fourmis afin de les appliquer par la suite à d'autres types d'applications..

##### 3.5.2.1 Les pionniers

L'origine des algorithmes de classification par les fourmis est due aux recherches entreprises par Deneubourg et son équipe [Deneubourg, 1990]. Lors des expériences de simulation, les objets à classer à rassembler sont placés aléatoirement sur une grille à deux dimensions. Les fourmis sont placées aussi aléatoirement sur la grille représentant l'environnement dans lequel elles évoluent.. Chaque fourmi n'a qu'une perception locale de son environnement et a pour

tâche de déplacer les objets en fonction de la concentration des objets de même type dans leur environnement proche appelé « voisinage »..

Le principe est de regrouper les objets similaires en des groupes sur une grille. Chaque fourmi peut prendre un objet avec une probabilité fonction de sa similarité avec les objets présents dans son voisinage et le déposer selon la même probabilité. Après un certain nombre d'itérations, des groupes d'objets similaires se forment sur la grille. La principale caractéristique de ces algorithmes est leur côté non supervisé qui permet de découvrir automatiquement le nombre de groupe adéquat sans intervention extérieure comme dans les algorithmes classiques de classification. Les opérations de dépôt et de ramassage des objets sont biaisées par les probabilités  $P_p$  et  $P_d$  représentées par

$$P_p = \left( \frac{k_1}{k_1 + f} \right) \quad (3.4)$$

$$P_d = \left( \frac{f}{k_2 + f} \right)^2 \quad (3.5)$$

$f$  est une estimation du nombre d'objets placés dans le voisinage de la fourmi.  $k_1$  et  $k_2$  sont des constantes positives. Quand  $f \ll k_1$  cela signifie qu'il y a peu d'objets dans le voisinage de l'objet et donc la probabilité de le prendre est élevée ( $P_d$  est proche de 1). Inversement quand  $f \gg k_1$  la probabilité de prendre l'objet est faible s'il est entouré de plusieurs objets. Deneubourg et al utilisent les valeurs de  $k_1 = 0.1$  et  $k_2 = 0.3$ .

Comme on le constate le mécanisme de rétroaction positive est présent mais agit d'une manière différentes par rapport à celui décrit dans les algorithmes de OCF. Les fourmis ne communiquent plus par trace de phéromones mais c'est la distribution des objets sur la grille qui est à la base de leur communication et coopération indirecte « stigmergie »

Cet algorithme de classification par les fourmis a trouvé ses premières applications en robotique collective [Beckers, 1994 ; Martinoli, 1999; Melhuish, 1999].

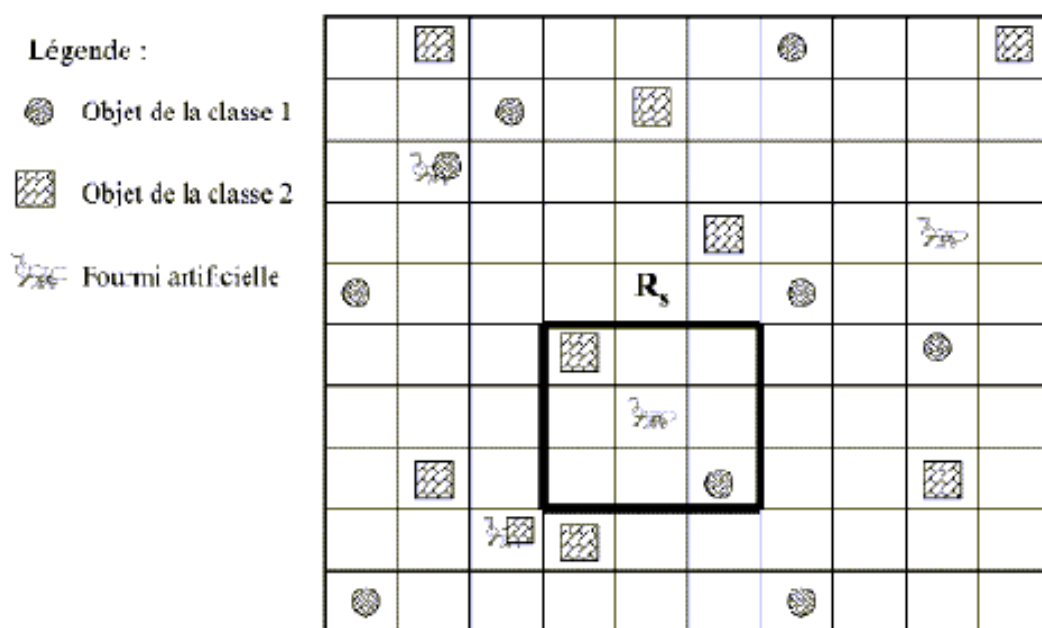
### 3.5.2.2 Modèle de Lumer et Faieta

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

- Les données sont représentées par des vecteurs de caractéristiques (numériques).

- La similarité entre deux données est mesurée comme une distance euclidienne entre leur vecteur de caractéristiques.
- La fourmi est capable de percevoir une région  $R_s$  de  $s \times s$  cases autour de sa position courante sur la grille.

La figure 3.7. représente un exemple de grille utilisé dans l'algorithme de Faieta et Lumer. Les objets sont représentés par des cases de deux motifs décrivant leurs types et le rectangle en trait épais est la région perçue par la fourmi.



**Figure 3.7. Grille de classification de Lumer et Faieta (d'après Labroche, 2003]**

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

$$P_p(i) = \left( \frac{k_1}{k_1 + f(i)} \right)^2 \quad (3.6)$$

$$P_d(i) = \begin{cases} 2f(i) & \text{si } f(i) < k_2 \\ 1 & \text{si } f(i) < k_2 s \end{cases} \quad (3.7)$$

avec



$$f(i) = \begin{cases} \frac{1}{s^2} \sum_{j \in R_s(r(i))} 1 - \frac{d(i,j)}{\alpha} & \text{si } f > 0 \\ 0 & \text{sinon} \end{cases} \quad (3.8)$$

$r(i)$  est la position de l'objet  $i$  sur la grille.  $f(i)$  est une mesure de similarité moyenne de l'objet  $i$  avec les objets  $j$  de son entourage. Le facteur  $\alpha$  contrôle la consistance de la fonction de dissimilarité entre les objets. Si  $\alpha$  est trop élevé les objets différents seront mis dans la même classe dans le cas contraire les objets similaires ne seront pas regroupés ensemble. Les probabilités de déplacement et de dépôt d'objets deviennent. Les tests ont été menés avec  $k_1=0.1$ ,  $k_2=0.15$ ,  $R=9$ ,  $\alpha = 0.5$

L'encadré 3.3. résume les étapes de l'algorithme de Lumer et Fata.  $A$  fourmis sont utilisées pour effectuer la tâche de classification de  $N$  objets.

---

*Algorithme 3.3 Algorithme de classification par les fourmis artificielles*

---

*Placer aléatoirement les  $N$  objets  $o_1, \dots, o_N$  sur la grille  $G$*

*Pour  $T = 1$  à  $T_{max}$  faire*

*Pour tout  $a_j \in \{a_1, \dots, a_A\}$  faire*

*Si la fourmi  $a_j$  ne transporte pas d'objets et  $r(o_i) = r(a_j)$  alors*

*Calculer  $f(o_i)$  et  $p_p(o_i)$*

*La fourmi  $a_j$  ramasse l'objet  $o_i$  suivant la probabilité  $p_p(o_i)$*

*Sinon*

*Si la fourmi  $a_j$  transporte l'objet  $o_i$  et la case  $r(a_j)$  est vide alors*

*Calculer  $f(o_i)$  et  $p_d(o_i)$*

*La fourmi  $a_j$  dépose l'objet  $o_i$  sur la case  $r(a_j)$  avec une probabilité  $p_d(o_i)$*

*Finsi*

*Finsi*

*Déplacer la fourmi  $a_j$  sur une case voisine non occupée par une autre fourmi*

*Finpour*

*Finpour*

*Retourner l'emplacement des objets sur la grille*

Les résultats obtenus ont montré que l'algorithme génère un nombre de classes qui est très souvent très éloigné du nombre réel de classes. Afin de remédier à cela, Lumer et Fata ont introduit trois extensions au comportement de base des fourmis artificielles :

- Les fourmis se déplacent sur la grille avec une vitesse propre à chacune d'elles comprise entre 1 et  $v_{max} = 6$ . Les fourmis les plus rapides sont moins sensibles aux

dissimilarités entre deux objets, ce qui permet de diminuer le nombre de classes générées.

- Chaque fourmi possède une mémoire à court terme lui permettant de se souvenir des positions des  $m$  derniers objets classés. Si une fourmi transporte un objet, elle cherche dans sa mémoire l'objet déjà classé qui est le plus proche de celui qu'elle transporte. Si elle le trouve, elle se déplacera (avec une certaine probabilité) vers ce nouvel emplacement pour y déposer son objet.
- Si au bout d'un certain nombre d'itérations la fourmi stagne (ne fait plus aucun déplacement) elle peut détruire un groupe en ramassant l'objet le plus éloigné du groupe.

### 3.5.2.3 L'algorithme AntClass et ses variantes

En se basant sur les travaux de Lumer et Faieta, Monmarché proposa un nouvel algorithme de classification AntClass [Monmarché, 1999]. AntClass utilise une colonie de fourmis artificielles qui se déplacent sur une grille toroïdale carrée afin d'éviter les effets de bords. La taille de la grille est calculée automatiquement en fonction de la taille des objets à classer. Chaque fourmi a la capacité de transporter plusieurs objets à la fois et de placer un tas d'objets sur une seule case de la grille. La localisation et l'extraction des classes est plus facile et plus fiable que dans le modèle de Lumer et Faieta. Cette façon de procéder permet une extraction plus facile et plus des classes. De plus, AntClass est une hybridation de l'algorithme stochastique de fourmis et de l'algorithme déterministe de classification « le K-means »(qu'on présentera dans le chapitre suivant) afin d'accélérer la convergence vers une partition stable. AntClass comporte deux exécutions successives d'un motif constitué d'une application de l'algorithme de fourmis suivi de l'algorithme K-means. Les fourmis génèrent une partition initiale de bonne qualité qui sera par la suite raffinée par K-means. AntClass a été testé sur des bases de données réelles et les résultats obtenus sont significatifs.

Dans [Kanade, 2003] Kanade et Hall reprennent les travaux de Monmarché et combinent l'algorithme de fourmis qu'il a proposé avec l'algorithme classique de classification FCM [Bezdek, 1981]. Dans une première étape, l'algorithme de fourmis est utilisé initialement pour fournir une première classification qui sera par la suite raffinée en utilisant l'algorithme FCM. Dans une seconde étape, les centres de classes obtenus par FCM sont considérés comme de nouveaux objets à classes. Ces derniers seront par la suite déplacés et fusionnés par les fourmis. A la fin, les classes obtenues par l'algorithme de fourmis sont une autre fois raffinée par l'algorithme FCM.

### 3.5.3 Autres domaines d'application

La simplicité de l'implantation des algorithmes de fourmis artificielles et les bons résultats obtenus pour la classification des données ont incité d'autres auteurs à les appliquer pour la résolution d'autres problèmes de classification. Parmi ces travaux, on peut citer ceux de Kuntz et Sneyer pour le partitionnement de graphe [Kuntz, 1997]. Les fourmis et les sommets du graphe sont placés aléatoirement sur la grille rectangulaire. Les fourmis regroupent les sommets en fonction de la topologie du graphe. Les résultats obtenus ont été évalués en calculant une mesure de l'entropie et le pourcentage de sommets mal classés. Pour la classification de documents, Ramos et al [Ramos, 2002] ont étendu les travaux de Deneubourg et al [Deneubourg, 1991] et Lumer and Faieta's [Lumer, 1994] en proposant leur algorithme Acluster pour la classification de mots dans des documents. Acluster utilise les traces de phéromones comme moyen de communication entre les fourmis et comme mémoire commune à toutes les fourmis du système de classification. Si une classe disparaît de la grille, la phéromone tend à s'évaporer dans sa direction, ce qui permet d'éviter que les fourmis se dirigent vers des directions non intéressantes. Ce travail a été repris par la suite par Abraham et Ramos [Abraham, 2003] et appliqué dans le domaine de l'internet. Dans le même ordre d'idée, Handl [Handl, 2003] propose une variante de l'algorithme de Lumer et Faieta pour la recherche visuelle de documents pour les moteurs de recherche sur internet. Les modifications introduites concernent en particulier l'adaptation de la valeur du paramètre  $\alpha$  en fonction du taux de ramassage et de dépôt d'objets par les fourmis et l'utilisation d'un processus de diversification nommé « stagnation control » afin d'accélérer la convergence. De plus les classes d'objets sont obtenues en appliquant un algorithme hiérarchique.

### 3.5.4 Autres sources d'inspiration

En plus du modèle du tri de couvain, il existe dans la littérature, d'autres modèles de classification automatique qui se placent aussi dans la lignée des précédents travaux sur les algorithmes de classification s'inspirant du comportement des fourmis réelles.

Dans sa thèse Labroche a introduit un nouveau modèle à base de fourmis pour la classification utilisant le système d'identification chimique des fourmis [Labroche, 2002]. Dans la nature, les éthologues ont montré que chaque fourmi arrive à reconnaître ses congénères qui appartiennent à sa colonie à partir d'une odeur coloniale qui est le fruit des apports génétiques, environnementaux et comportementaux. A partir de ce mécanisme d'identification un nouvel algorithme de classification a été proposé dans lequel chaque donnée est une fourmi dont l'odeur est déterminée par les valeurs prises par les attributs décrivant cette donnée. Les fourmis effectuent des rencontres aléatoires et décident qu'elles appartiennent ou non à la même classe en fonction de cette odeur.

En s'inspirant toujours du comportement des fourmis, Azzag proposa dans [Azzag, 2004] un nouvel algorithme de classification automatique qui trouve son origine dans la manière dont les fourmis réelles forment des structures vivantes est proposée. Il s'agit d'une méthode de classification hiérarchique distribuée qui simule le phénomène d'auto-assemblage observé chez les fourmis pour regrouper les données selon un arbre. Chaque fourmi représente une donnée à classer. A partir d'un point de support fixe sur lequel sont situées initialement les fourmis (répartition des données), ces dernières vont s'accrocher successivement au support, puis aux fourmis connectées au support, et ainsi de suite jusqu'à ce que, par exemple, une chaîne de passage soit construite entre deux points. Les fourmis se déplacent sur la structure vivante et s'accrochent sur celle-ci aux endroits les plus opportuns en fonction du but à atteindre (plus grande similarité avec les fourmis de la structure).

### **3.6 Conclusion**

Dans ce chapitre, nous avons présenté les algorithmes à base de fourmis artificielles inspirés de comportement collectif des fourmis pour la résolution des problèmes d'optimisation combinatoires et le problème de classification automatique. Ces systèmes auto-organisés et non centralisés utilisent une population de fourmis artificielles autonomes et d'une intelligence très réduite. Chaque fourmi n'a qu'une perception réduite de son environnement et ne fait qu'agir/réagir aux différents stimulus venant de son environnement. La coordination du comportement de la colonie se base sur un processus stigmergique.

Dans OCF, des agents fourmis coopèrent ensemble au sein de leur colonie, en cherchant en parallèle une solution optimale (ou sous optimale) au problème sous étude. Chaque fourmi construit une solution candidate même de mauvaise qualité. La solution optimale émerge de la coopération entre les fourmis. Lors de la construction de sa solution, la fourmi accumule des informations sur l'environnement parcouru. Ces informations sont stockées sous forme de taux de phéromones et seront échangées d'une manière indirecte avec les autres fourmis de la colonie. La construction des solutions est incrémentale selon un processus stochastique. A chaque étape du processus de construction, la fourmi choisit un élément de la solution parmi les éléments possibles d'une manière probabiliste guidée par deux mesures : une quantité de phéromone associée à l'élément et reflétant l'expérience accumulée des autres fourmis, et une information heuristique propre au problème à résoudre. Une fois qu'une fourmi a terminé la construction de sa solution, elle met à jour le taux de phéromones qui correspond à ses choix en fonction de la qualité de la solution qu'elle a trouvée. Cette mise à jour a pour but le renforcement de la probabilité de choisir les éléments qui conduisent à la bonne solution par les autres fourmis lors des prochaines itérations. Afin d'améliorer la qualité des solutions et accélérer la convergence, les algorithmes de fourmis sont très souvent hybridés avec une méthode de recherche locale.

Les algorithmes de fourmis pour la classification non supervisée sont aussi un autre modèle inspiré du comportement collectif observé chez les fourmis tels que le tri collectif de couvains, la reconnaissance chimique et l'auto-assemblage. Ces algorithmes peuvent bénéficier de propriétés intéressantes comme l'optimisation locale et globale de la classification, l'absence d'information sur une classification initiale des données, le parallélisme, etc.

# Chapitre 4

## La segmentation d'images

<b>Chapitre 4. La segmentation d'images</b> .....	<b>71</b>
4.1 Introduction .....	71
4.2 La vision artificielle .....	71
4.3 Architecture typique d'un système de vision .....	72
4.4 Techniques de segmentation .....	74
4.4.1 Détection de contours .....	75
4.4.2 Détection de régions homogènes.....	78
4.4.3 Les méthodes de classification.....	87
4.4.4 Les méthodes de segmentation biomimétiques.....	99
4.5 Méthodes d'évaluation des résultats de segmentation en régions.....	106
4.5.1 Evaluation par comparaison avec une segmentation de référence.....	107
4.5.2 L'évaluation se référant à l'image originale .....	111
4.6 Conclusion.....	116

---

## Chapitre 4. La segmentation d'images

---

### 4.1 Introduction

Nous présentons dans ce chapitre, le contexte général dans lequel ce travail s'inscrit ainsi que les différents problèmes abordés au cours de cette thèse. Dans un premier lieu, nous rappelons brièvement quelques notions attachées à la vie artificielle. Ensuite nous passons en revue les principales méthodes de segmentations développées dans le cadre de la segmentation d'images. De nombreux algorithmes ont été proposés dans la littérature durant ces dernières décennies [Zucker, 1976], [Haralick, 1985], [Pal, 1993], [Cocquerez, 1995a], [Freixenet, 2002], [Cho, 1997], [Heath, 1996], [Kara-Falah, 1995], [Palus, 1999], [Ramos, 1999], [Tang, 1999]. Une étude comparative des méthodes de segmentation peut aussi être retrouvée dans [Rouquet, 1998], [Salzenstein, 1998]. Ce chapitre porte un intérêt particulier pour les méthodes de segmentation par classification 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 biomimétiques. Cette présentation n'est pas exhaustive mais permet de présenter les bases de chacune des techniques de segmentation.

### 4.2 La vision artificielle

La richesse et la qualité des informations fournies par le système visuel humain ont depuis très longtemps suscité l'intérêt des chercheurs. Dès lors un nouveau paradigme est apparu sous le nom « vision artificielle ». L'idée est de doter les machines, d'un système visuel artificiel leur permettant à partir d'images de simuler des tâches complexes de perception et d'interprétation naturellement et facilement effectuées par un être humain.

Les applications des systèmes de vision se retrouvent d'ores et déjà dans de nombreuses sphères d'activités que nous pouvons citer sans être exhaustifs :

- Le domaine médical avec l'analyse d'images de radiographies ou d'échographies, l'étude cytologique des cellules ;
- La robotique avec la création de robots autonomes ;
- Le contrôle qualité,
- La cartographie aérienne,

· La police pour la reconnaissance des empreintes digitales.

Un système de vision artificiel peut être considéré comme un ensemble de tâches assurant le passage d'une image naturelle acquise par des capteurs à une description structurale puis sémantique en passant par la segmentation et l'extraction des caractéristiques des objets de la scène.

David Marr fut le premier chercheur à proposer une méthodologie pour la conception des systèmes de vision artificielle connu par la suite sous le nom *paradigme de Marr* [Marr, 1980]. Selon lui, tout système de vision doit contenir trois niveaux (Figure 4.1)

- La définition des moyens d'acquisition et de transformation permettant le passage d'une scène réelle à une image numérique traitable par un ordinateur.
- L'étude et le calcul des primitives présentes dans les images. Ce niveau nécessite, la définition de modèles mathématiques des différents phénomènes observés dans les images et servant à leur compréhension.
- L'interprétation

Guidées par l'approche de Marr, de nouvelles méthodologies de construction de systèmes visuels artificiels sont apparues dans la littérature telles que *la vision active*, *la théorie de Gestal*, *le smart sensing*, *la vision animée*, ..etc. Malgré leur diversité, la majorité des chercheurs soulignent que le problème actuel à étudier en vision artificielle est la nécessité de développer d'outils puissants pour l'extraction des primitives présentes.

### 4.3 Architecture typique d'un système de vision

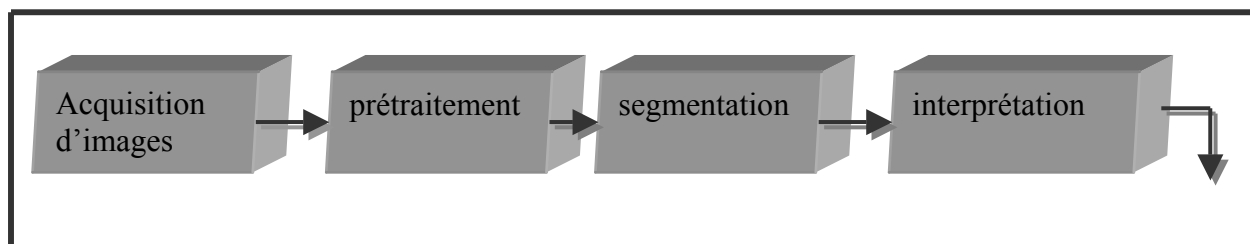
Le processus de vision artificielle se décompose en plusieurs étapes qui peuvent s'exécuter soit en séquence ou bien en parallèle [Boucher, ]. D'une manière générale, ces étapes concernent :

- L'acquisition d'une scène : elle permet de discrétiser l'image réelle continue afin d'être traitée par l'ordinateur. L'image obtenue est de nature bidimensionnelle discrète. A chaque élément (pixel) de l'image, correspond un vecteur de caractéristique qui se réduit le plus souvent au niveau de gris.
- Les prétraitements: Une image brute est toujours entachée de dégradations d'origine diverses, dont il va falloir essayer de minimiser l'influence afin d'obtenir l'image la



plus nette possible, c'est le rôle de la fonction prétraitements a qui incombe l'opération de clarification de l'image. Les prétraitements représentent des fois une étape incontournable, et peuvent être classés suivant deux catégories :

- Des prétraitements basés sur des transformations ponctuelles qui modifient ponctuellement les niveaux de gris des pixels de l'image.
- Des prétraitements basés sur des transformations globales où l'étendue de la transformation est toute l'image, pixel par pixel.
- La segmentation qui permet l'extraction des indices pertinents (primitives) présents dans l'images (régions, contours, etc.)
- Le calcul pour chaque indice ou primitive extraite, l'ensemble des attributs
- L'interprétation et la prise de décision



**Figure 4.1. Schéma d'un système de vision**

Le modèle présenté dans la figure 4.1. correspond à une architecture typique d'un système de vision, mais d'autres variantes peuvent exister dans la littérature [Boucher, 1999].

Les étapes de vision peuvent aussi être groupées en deux grandes classes d'algorithmes : les algorithmes bas-niveau qui concernent l'acquisition, les prétraitements, et la segmentation ; les algorithmes haut-niveau qui groupent l'extraction des mesures et la phase d'interprétation.

La segmentation des images constitue le cœur de tout système de vision et une étape importante dans le processus d'analyse des images [Cheng, 2001]. Elle a pour objectif de fournir une description des objets contenus dans l'image par l'extraction de différents indices visuels tels que les contours des objets, les régions homogènes, les objets 3 D. Ces indices visuels représentent des phénomènes photométriques et/ou géométrique présents dans l'image. Ils seront exploités ensuite par les traitements placés en aval pour une description symbolique de la scène permettant une interprétation et éventuellement une prise de décision.

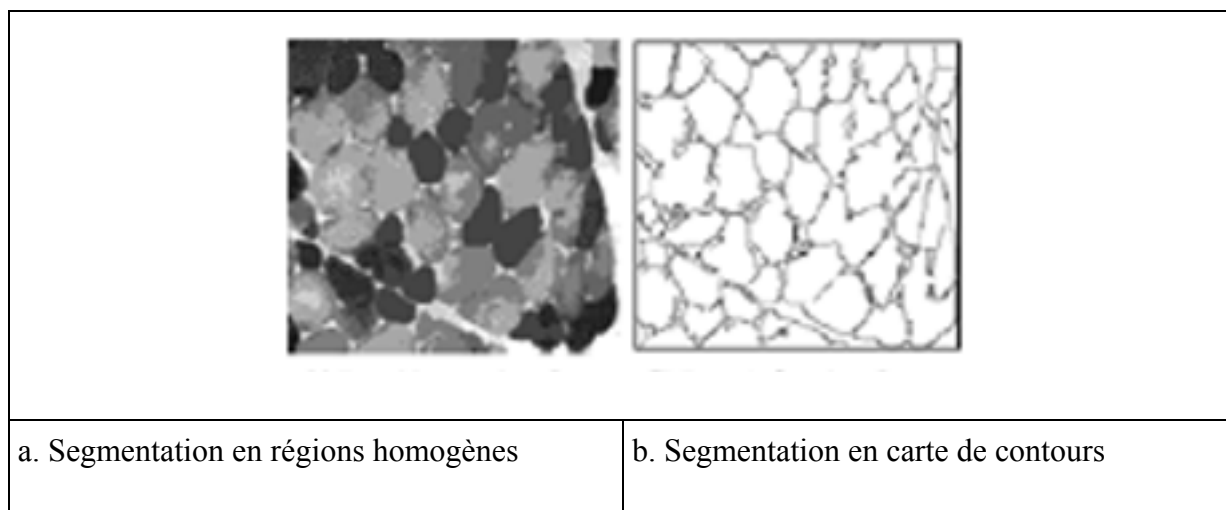
La segmentation peut être basée directement sur le niveau de gris ou la couleur de chaque pixel ou bien sur un attribut estimée dans le voisinage du pixel, tel que la valeur moyenne, la variance ou des paramètres de texture plus complexes. La phase de segmentation d'images n'est pas considérée comme un but en soi [Zhang, 1997 ; Baradez, 2004], mais dépend fortement aussi bien du type de traitement fixé par l'utilisateur sur les objets présents dans l'image, que de la nature de l'image (présence de bruit, présence de zones texturées, contours flous...), ainsi des primitives que l'on cherche à extraire de l'image et qui dépendent des opérations situées en aval de la segmentation (localisation, calcul 3D, reconnaissance de formes, interprétation).

Ces aspects, ainsi que les contraintes d'exploitation (complexité algorithmique, aspect temps réel), justifient la multiplicité des techniques développées dans le domaine de la segmentation d'image. Jusqu'à ce jour, il n'existe pas de méthode universelle de segmentation d'images. Toute technique n'est efficace que pour un type d'image donné, pour un type d'application donné, et dans un contexte informatique donné. En raison de ces contraintes, diverses stratégies de segmentation ont été proposées dans la littérature. La suite de ce chapitre présente les stratégies de segmentation les plus utilisées.

#### **4.4 Techniques de segmentation**

Les techniques de segmentations existantes sont nombreuses mais elles sont généralement regroupées en trois principales approches qui sont l'approche *contour*, l'approche *région* et l'approche *classification* [Tseng, 2001; Turi, 2001]. Les méthodes de l'approche contour consistent à identifier les discontinuités du niveau de gris qui séparent les régions. Les méthodes de l'approche région cherchent à regrouper des pixels qui présentent une similarité et une uniformité en niveau de gris. Ces deux approches sont duales l'une de l'autre : une région homogène définit un contour, alors qu'un contour fermé délimite une région homogène. Cependant elles mènent à des algorithmes différents et ne donnent pas les mêmes résultats. Les méthodes de classification partitionnent l'image en des classes homogènes en considérant les propriétés colorimétriques des pixels. Les informations spatiales peuvent ou non être pris en considération.

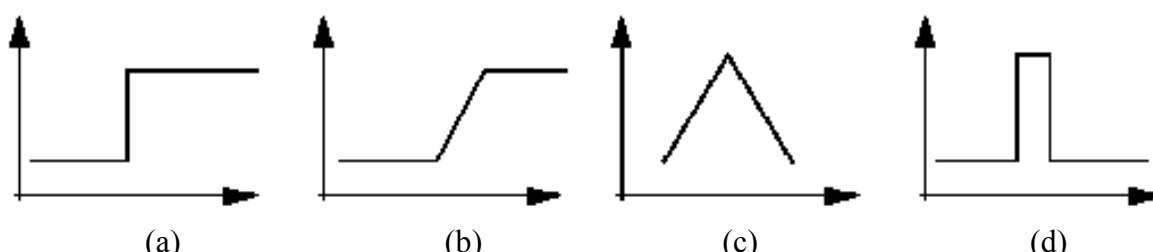
Dans la suite de ce chapitre, nous allons décrire quelques algorithmes de ces trois catégories sans pour autant prétendre à une présentation détaillée de l'ensemble des méthodes de segmentation. Nous présenterons de façon succincte l'approche « contour » de la segmentation, puis de façon un peu plus précise l'approche « région » et l'approche « classification » contexte de ce travail de thèse.



**Figure 4.2. Exemple de segmentation d'images**

#### 4.4.1 Détection de contours

Selon Pavlidis [Pavlidis, 1992], un contour est une zone de transition séparant deux régions homogènes due à des discontinuités de profondeur et de réflectance de la scène observée. Ainsi un contour peut séparer deux régions ayant des intensités moyennes différentes. Il s'agit alors de contours de type « saut d'amplitude », (voir figure 4.3 a) et b)). Ou bien il peut également correspondre à une variation locale d'intensité présentant un maximum ou un minimum (figure 4.3 c) et d)).



**Figure 4.3. Différents types de contours**

Nous présentons dans ce qui suit les principales méthodes de détection de contours.

##### 4.4.1.1 Méthodes dérivatives

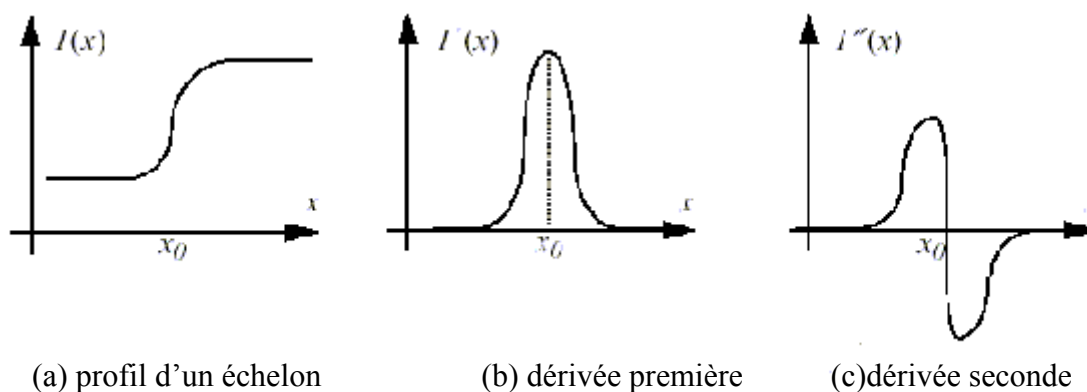
Les méthodes dérivatives sont les plus utilisées pour détecter des transitions d'intensité par différenciation numérique. Ce sont des méthodes locales qui balayent l'image avec un masque définissant la zone d'intérêt. A chaque position, un opérateur est appliqué afin de détecter les transitions significatives au niveau de l'attribut de discontinuité choisi. Le résultat est une image binaire constituée de points de contours et de points non-contours. Pour obtenir des régions homogènes il est nécessaire d'utiliser en aval de ces méthodes des traitements qui ferment les contours ouverts et remplissent les zones délimitées par ces contours. Etant donné

que les opérateurs de dérivation sont très sensibles au bruit, les images bruitées doivent être lissées au préalable. Le lissage et la dérivation sont en pratique réunies dans un seul filtre.

De nombreuses techniques d'extraction de contours existent dans la littérature. Elles peuvent être divisées en deux approches :

- Approche gradient : calcul des premières dérivées partielles ;
- Approche laplacien : calcul des deuxièmes dérivées partielles :

L'allure de la dérivée première et de la dérivée seconde est donnée par la figure 4.3. Une zone de transition dans le signal correspond à un maximum (ou un minimum) local de la dérivée première et un passage par zéro de la dérivée seconde. L'identification d'une zone de transition du signal peut être faite par seuillage de la norme de sa dérivée première ou du passage par zéro de sa dérivée seconde.



**Figure 4.4. Dérivées première et seconde d'un contour de type "saut d'amplitude"**

Parmi les opérateurs gradient, on trouve le masque de Robert [Roberts 1965], de Prewit [Prewit 1970], de Sobel [Sobel 1978], de Kirsh [Kirsh 1971]... Ces opérateurs sont faciles à implémenter, rapides en temps de calcul et donnent de bons résultats pour des images non bruitées. En cas de présence de bruit, une phase de seuillage est utilisée avant de procéder à l'application de ces opérateurs. Le problème majeur des techniques exploitant le Laplacien est sa grande sensibilité aux petites variations non significatives et qui sont dues essentiellement à la présence du bruit. Une manière de résoudre ce problème est d'utiliser un filtre gaussien comme proposé par Marr et Hildreth [Marr, 1980]. La méthode consiste à convoluer l'image par une gaussienne puis d'appliquer le filtre Laplacien.

Pour améliorer la qualité des contours et pallier aux problèmes de précision de localisation et d'efficacité de détection, sont apparus les opérateurs de dérivation avec filtrages optimaux. Le

filtre optimal est un dérivateur qui permet de détecter des contours en respectant les 3 critères suivants [Canny, 1986]:

1. *Une bonne détection* : l'opérateur donne une réponse au voisinage d'un contour ;
2. *Une bonne localisation* : optimisation de la précision avec laquelle le contour est détecté;
3. *Unicité de la réponse* : le contour doit provoquer une réponse unique de l'opérateur.

Plusieurs opérateurs optimaux sont apparus dans la littérature. Parmi eux on trouve le filtre de Canny [Canny, 1986], le filtre de Deriche [Deriche, 1987; Deriche, 1990] et le filtre de Shen et Castan [Shen, 1986 ; Castan, 1989 ; Shen, 1992].

#### 4.4.1.2 Autres méthodes

D'autres approches d'extraction de contours font appel à d'autres modèles de l'image. Nous en citons :

- Les approches surfaciques : L'image des intensités est considérée comme une surface. Dans ce cadre, la transition entre deux régions est modélisée par un modèle surfacique. Le contour est présent quand il y a une bonne corrélation entre le modèle surfacique et une zone de l'image [Hueckel, 1971].

- Les approches contours actifs : Les contours actifs ou « Snakes » ont été introduits par Kass et al. [Kass, 1988]. Ils se présentent comme un modèle pour l'extraction des caractéristiques visuelles dans une image comme les contours d'objet ou les éléments de frontières. L'idée de base est de positionner, au voisinage du contour à détecter, une courbe qui sera l'initialisation du contour actif et de la déformer successivement jusqu'à ce qu'elle coïncide avec la frontière de l'objet .

Une étude plus détaillée des méthodes de détection de contours, peut être trouvée dans [Hejmans, 1993].

#### 4.4.1.3 Post-traitements en détection de contours

D'autres étapes sont nécessaires pour passer des points de contours à une liste de contours exploitables par un processus d'interprétation. Très souvent la détection des points de contours est suivie d'une étape de prolongement et de chaînage de ces points et un codage des chaînes obtenues par des segments de droites [Giraudon, 1987].

#### 4.4.1.4 Conclusion

A partir de cette brève présentation de l'approche contour, nous pouvons tirer les conclusions suivantes :

- Les détecteurs de contours donnent de bons résultats quand l'image traitée n'est pas trop bruitée, dans le cas contraire ils nécessitent l'utilisation de filtres de lissage qui peuvent affecter les frontières entre les régions homogènes et donc délocaliser les points de contours. De plus le seuils sont difficiles à fixer sans une information a priori qui n'est pas toujours disponible. Les filtres optimaux présentent un bon compromis entre une bonne détection et une bonne localisation.
- La carte de points de contours obtenue à la fin du processus de segmentation, devra souvent être transformée en un ensemble de régions plus riches en information et plus facilement exploitable par le haut niveau. Une étape de fermeture des contours sera donc généralement nécessaire avant de commencer la phase d'interprétation [Haris, 1998].

#### 4.4.2 Détection de régions homogènes

Contrairement à la segmentation par contours dont le principe est la recherche des points essentiels qui donnent la forme des objets composant l'image, la segmentation en régions consiste à décomposer l'image en des régions homogènes [Gonzalez, 1992; Jain, 2000; Fuh, 2000]. Une région est composée de l'ensemble des pixels connexes possédant les mêmes propriétés au sens d'un prédicat d'homogénéité donné.

Il est courant de définir la segmentation d'une image  $I$  en terme d'un prédicat d'homogénéité  $P$  et d'un ensemble de « régions »  $R_i$  vérifiant les critères suivants [Zucker, 1976] :

1.  $I = \bigcup_i R_i$
2.  $R_i \neq \emptyset \quad \forall i$  ;
3.  $R_i \cap R_j = \emptyset \quad \forall i \neq j$  ;
4.  $R_i$  est connexe  $\forall i$  ;
5.  $P(R_i) = \text{vrai} \quad \forall i$  ;
6.  $P(R_i \cup R_j) = \text{faux} \quad \forall i \neq j$  et  $R_i$  et  $R_j$  sont adjacentes.

La vérification de ces conditions est une condition nécessaire et suffisante pour qu'une partition d'une image soit une segmentation. Rien, toutefois, n'implique l'unicité de cette segmentation. D'une part le nombre de régions obtenues est variable en fonction des prédicats d'homogénéité ainsi qu'aux seuils de tolérance contrôlant leur formation. D'autre part ces critères d'homogénéité doivent s'adapter, à la nature de l'image à segmenter.

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 prédicat prédéfini [Lopès, 1999], [Chang, 1994; Pavlidis, 1990]. Le contrôle géométrique peut être aussi utilisé pour contrôler le processus de segmentation [Chassery, 1991a]. 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.

On distingue dans cette classe trois grandes familles de techniques :

- Les méthodes de fusion de régions qui procèdent par agrégation itérative des pixels (ou de régions). L'algorithme s'arrête quand une segmentation optimale est atteinte au sens d'un prédicat d'homogénéité.
- Les méthodes de division qui procèdent par division des régions de base en régions plus petites et de plus en plus homogènes. La division s'arrête quand toutes les régions produites vérifient un certain critère d'homogénéité.
- Les méthodes de division/fusion qui combinent les deux méthodes précédentes.

Nous décrivons dans les paragraphes suivants les méthodes essentielles de chacune de ces classes.

#### 4.4.2.1 Fusion de régions

Dans cette catégorie, on distingue entre les méthodes qui agrègent des pixels adjacents de façon itérative et d'autres plus sophistiquées qui fusionnent de petites régions (obtenues par une simple technique de croissance de régions ou par une technique de division) en utilisant plusieurs critères de fusion portant sur les caractéristiques des régions adjacentes.

##### ***Agrégation de pixels***

Le point de départ est le choix d'un ensemble de pixels appelés « germes ». A 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 [Gonzalez, 1992 ; Trémeau, 1997]. Le processus d'agrégation est itératif de sorte que chaque région croît pixel par pixel. Divers vecteurs de caractéristiques et diverses mesures de similarité ont été proposés dans la littérature [Cocquerez, 1995a]. Généralement le vecteur de caractéristique est réduit à la valeur de niveau de gris et la distance euclidienne est utilisée comme mesure de similarité. Des propriétés statistiques telles que la moyenne ou la variance peuvent aussi être utilisées [Bryant, 1979].

Les régions ainsi construites dépendent fortement du choix des germes de départ ainsi que de l'ordre de parcours des pixels agrégés. La plupart des algorithmes de croissance de régions parcourent l'image de façon prédéterminée de haut en bas et de gauche à droite [Kara-Falah, 1995; Adams, 1994; Menhert, 1997]. Kara-Falah [Kara-Falah, 1995] propose une approche originale pour l'extraction des germes. A partir de la réalisation de  $n$  segmentations possibles, une segmentation consensus est obtenue par une mise en association des différentes régions des  $n$  segmentations. Les noyaux des régions de cette segmentation sont alors considérés comme des candidats pour être des germes de croissance de régions. Après un test de stationnarité, seuls les germes qui correspondent à des régions homogènes sont conservés. La croissance de régions est basée d'une part sur un critère d'homogénéité et sur une information de contours ce qui permet d'éviter d'éventuels conflits aux frontières des régions.

### ***Regroupement itératif d'ensemble de pixels***

Au départ, de petites régions-germes sont générées (elles peuvent se réduire à un pixel). Un graphe d'adjacence de régions est construit tel que chaque nœud représente une région et chaque arête représente une adjacence entre deux régions. Un coût de fusion est associé à chaque couple de régions adjacentes. La méthode consiste alors à déterminer à partir de ce graphe pondéré, les meilleures fusions qui minimisent ce coût. Une première utilisation de cette approche revient à [Muerle, 1968] qui ont suggéré de regrouper deux régions adjacentes si elles sont suffisamment proches en terme de niveau de gris.

Cette méthode est facile à implémenter mais pose certains problèmes tels que : 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 [Turi, 2001]. Monga propose une méthode de segmentation par croissance de régions, en utilisant plusieurs prédicats d'uniformité qui sont de plus en plus tolérants [Monga, 1987]. L'utilisation de prédicats emboîtés permet d'une part de limiter le problème du choix du prédicat de fusion et d'autre part de corriger des 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 le choix de l'ordre d'emboîtements des prédicats.



Dans [Regina, 2001] un algorithme de croissance de région est proposé tel que le critère d'homogénéité n'est pas fixé au départ mais est défini automatiquement à partir des caractéristiques de la région à segmenter. La méthode est basée sur un modèle qui décrit l'homogénéité et la forme de la région à segmenter. Les paramètres du critère d'homogénéité sont estimés à partir des zones d'échantillon dans la région. Ces endroits sont choisis séquentiellement à partir d'un point germe, et le critère d'homogénéité est mis à jour sans interruption.

R. Schettini propose une méthode de segmentation d'images couleurs en utilisant un critère prenant en compte la similarité colorimétrique de deux régions et la proximité spatiale [Schettini, 1993]. Il introduit une fonction de similarité qui repose sur une distance colorimétrique pondérée par la longueur relative de la frontière commune des deux régions à fusionner. K. Saarinen [Saarinen, 1994] propose de pondérer chaque arête d'un graphe d'adjacence de régions par la moitié de la distance colorimétrique entre les deux régions correspondant aux nœuds de l'arête. A chaque itération, les régions reliées par l'arête qui porte le poids minimal sont fusionnées. Les poids des arêtes sont mis à jour en fonction du nombre de pixels appartenant aux régions associées aux arêtes et de leur distance colorimétrique. Grâce à la pondération choisie, l'algorithme tend à privilégier les régions adjacentes de faible surface. L'algorithme de fusion s'arrête lorsqu'un nombre d'itérations fixé préalablement est atteint ou lorsque les poids des arêtes atteignent une valeur limite.

Afin d'affiner les résultats de segmentation obtenues, certains auteurs prennent en considération l'information frontière entre deux régions. Dans [Hiohan, ] Hiohan et al proposent un algorithme de croissance de régions par agrégation de pixels en utilisant un critère d'homogénéité qui est la somme pondérée du contraste entre une région et un pixel candidat et une information sur la valeur du gradient en ce pixel. Une valeur faible de cette fonction indique une grande adaptation du pixel à la région considérée. La même idée a été proposée par Karaet [Karaet, ] qui ne considère à chaque itération de l'algorithme, que les pixels dont le gradient est faible par rapport à un seuil fixé par l'utilisateur.

Dans [Kermod, 1995], une coopération entre une technique de seuillage et celle de fusion de régions est proposée. Une segmentation initiale est obtenue par seuillage des histogrammes locaux extraits sur des petites régions de l'image à traiter. Après, une fusion de régions étiquetées minimisant un critère de similarité est effectuée.

Il existe plusieurs autres représentations pyramidales qui sont utilisées pour la segmentation d'image. On peut mentionner la pyramide reliée [Burt, 1981], la pyramide reliée pondérée [Hong, 1984], la pyramide à seuillage dynamique [Home, 1990] et la segmentation par

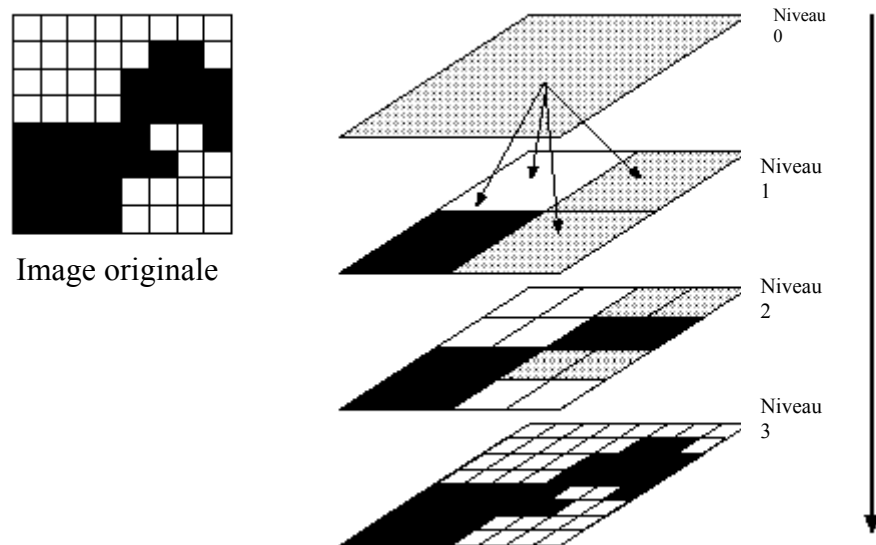
pyramide irrégulière [Bertolino, 1995]. Bertolino propose une segmentation pyramidale et utilise une information de gradient lors de l'évaluation de la fusion des régions du graphe d'adjacence. L'approche pyramidale s'inspire de la stratégie de fusion optimale des régions [Jolion 1989; Montanvert, 1991]. 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. Cette stratégie est optimale par rapport au critère de fusion des régions. Pour éviter la structure très séquentielle de cette méthode, l'approche pyramidale consiste à effectuer les meilleures fusions locales de façon simultanée, puis recommencer avec les régions formées au niveau suivant et ainsi de suite jusqu'à la stabilité de la carte des régions. L'optimalité est perdue mais les résultats restent du même ordre avec une parallélisation possible de l'algorithme. La difficulté est de maintenir la cohérence du graphe d'adjacence et de choisir une seule fusion locale en conservant l'aspect parallèle de la méthode. En mémorisant la qualité du voisinage des niveaux précédents (les nœuds pères transmettent les informations aux nœuds fils), la fusion peut exploiter des informations locales et globales.

#### 4.4.2.2 Les Méthodes de type division (Split)

Dans ce type de méthodes, l'image est divisée d'une manière récursive tant que le critère d'homogénéité n'est pas vérifié. Le critère d'homogénéité est validé d'une manière globale sur l'image originale. Si le critère est vérifié l'algorithme s'arrête. Sinon, on divise l'image en des zones. Chaque zone est testée et redivisée si elle ne valide pas le critère. L'algorithme se termine lorsque toutes les régions sont homogènes ou bien leur taille est en dessous d'un seuil de taille minimal fixé.

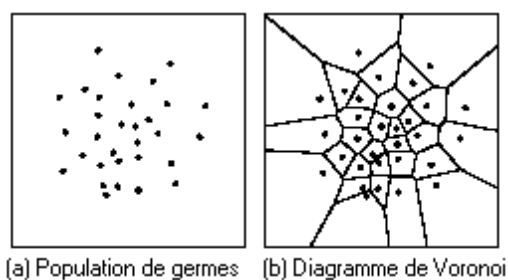
La division de l'image est réalisée selon une structure géométrique. Citons deux structures possibles :

- L'arbre quartenaire (structure « Quadtree ») : Le Quadtree est un tétra-arbre dont chaque nœud possède exactement quatre nœuds fils, excepté les nœuds terminaux. Chaque nœud correspond à un bloc, c'est à dire une zone de l'image de forme carrée, dont le nombre de lignes et de colonnes est une puissance de deux. La racine de l'arbre correspond à l'image entière qui doit être de forme carrée et dont le nombre de lignes et de colonnes est également une puissance de deux. A la base de la pyramide, chaque bloc correspond donc à un pixel de l'image. Un algorithme simple de division consiste à tester l'homogénéité de l'image originale. Si elle n'est pas homogène, elle est récursivement découpée en des zones carrées jusqu'à ce que chaque carré soit homogène (Figure 4.5). Malgré sa simplicité, cette méthode présente deux inconvénients majeurs qui sont la non invariance en translation et sa rigidité.



**Figure 4.5. Quad Tree d'une image 8\*8.**

- Structure de Voronoi: Au départ des germes initiaux sont choisis afin de construire les polygones de Voronoi. Ensuite de nouveaux germes sontensemencés d'une manière itérative à l'intérieur de chaque polygone non homogène, modifiant ainsi le diagramme de Voronoi. Le processus d'ensemencement de germes se poursuit jusqu'à ce que tous les polygones soient homogènes ou bien que leurs tailles atteignent un seuil préfixé. Une des différences par rapport à la structure précédente est que lorsqu'un polygone est partitionné, la topologie de polygones voisins change aussi, s'adaptant à ce nouveau partitionnement.



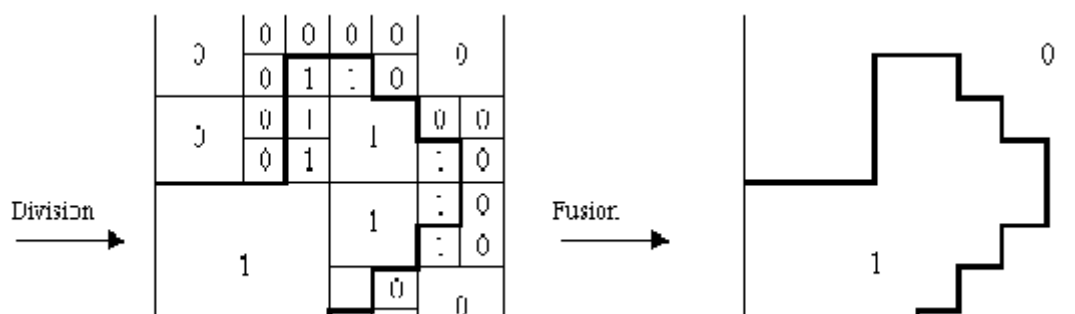
**Figure 4.6. Diagramme de Voronoi**

Beaucoup d'algorithmes de division consistent à tester l'homogénéité des régions en utilisant l'histogramme de niveau de gris. Les différents pics et vallées de l'histogramme sont recherchés et les régions sont alors définies à partir des intervalles entre les vallées. Dans la technique de Ohlander, Price et Reddy [Ohlander, 1979], l'histogramme de l'image originale est calculé et à partir des vallées et des pics extraits, l'image est décomposée successivement en des régions homogènes. Pour chacune des régions obtenues l'histogramme est recalculé. Si l'histogramme est uni-modal, la région reste inchangée, sinon elle est divisée et le processus

continue. La division récursive prend fin quand toutes les régions ont un histogramme unimodal.

#### 4.4.2.3 Les méthodes de type division/fusion

Ces méthodes combinent les deux méthodes décrites précédemment, la division de l'image en des petites régions homogènes, puis la fusion des régions connexes et similaires au sens d'un prédicat de regroupement. Horowitz et Pavlidis sont les premiers à avoir proposé une telle approche de segmentation [Horowitz, 1976] à partir de laquelle de nouvelles variantes sont apparues dans la littérature [Strasters 1991; Gonzalez 1992]. Le processus de segmentation utilise la structure pyramidale du Quadtree est peut être décrit comme suit: en premier lieu chaque bloc associé à un nœud du Quadtree est analysé de façon récursive afin de décider s'il doit être divisé en quatre sous-blocs. L'analyse récursive s'arrête lorsque chaque sous-bloc respecte un prédicat d'homogénéité. Ensuite, à chaque fois que 4 sous-bloc satisfont un critère d'homogénéité ils sont regroupés à un niveau supérieur du Quadtree. La fusion continue tant qu'il est possible de le faire, c'est à dire tant que le critère d'homogénéité est satisfait. Lorsqu'il n'est plus possible de fusionner, il reste encore une étape pour examiner les blocs adjacents qui n'étaient pas au même niveau dans le Quadtree et les fusionner s'ils satisfont le critère d'homogénéité (Figure 4.7)..



**Figure 4.7. La segmentation par division-Fusion.**

Chen et Pavlidis [Chen, 1978] reprennent cet algorithme pour segmenter des images au sens de la texture. Pour évaluer l'homogénéité d'une région, ils suggèrent l'utilisation d'un test statistique qui suppose que les moyennes des régions ne varient pas trop durant la phase de division. Cette méthode bien qu'elle soit rapide et de taille mémoire relativement simple, crée de faux contours entre les régions de l'image. De plus elle n'utilise aucun critère de forme pour contraindre la fusion, ce qui occasionne parfois des régions particulièrement découpées.

Dans [Brice, 1970], l'image est tout d'abord divisée en des régions d'intensités constantes. Des heuristiques sont alors utilisées afin d'obtenir des régions dont les frontières sont naturelles (faibles). Les régions dont les vecteurs de caractéristiques sont semblables sont

fusionnées. Ensuite, les frontières de faible niveau de gris sont supprimées, ce qui permet de fusionner de nouvelles régions. Des pré-conditions sont prises afin d'éviter de fusionner des régions de grandes tailles ne présentant que quelques pixels de frontières communes. Malgré son optimalité et l'indépendance par rapport à l'ordre d'évaluation des arêtes, c'est une approche séquentielle très coûteuse en temps de calcul.

Un autre exemple de division/fusion est l'algorithme développé par Beveridge et al [Beveridge, 1989]. L'image est d'abord découpée en de petites fenêtres rectangulaires qui se chevauchent afin d'éviter des problèmes dus aux frontières des régions. La division est ensuite appliquée d'une façon parallèle pour chacune de ces fenêtres en utilisant les modes et les vallées de leur histogramme de niveaux de gris. Une région est créée pour chaque mode de l'histogramme tout en tenant compte de la connexité des pixels. Les régions suffisamment proches au sens d'un critère appelé « merge score » sont ensuite fusionnées. Ce critère dépend de la moyenne des niveaux de gris des deux régions, de leur taille et de la longueur de leur contour commun. L'inconvénient de cet algorithme est qu'il introduit beaucoup de seuils pour effectuer la segmentation.

Beaulieu et Goldberg [Beaulieu, 1989] proposent un algorithme d'optimisation hiérarchique pour la croissance de régions et produisent une décomposition hiérarchique de l'image. La différence principale avec les autres méthodes réside dans l'ordre de fusion des régions adjacentes. A chaque étape de fusion, l'algorithme cherche la paire de régions à fusionner la plus optimale au sens du prédicat utilisé. Ce qui veut dire que les deux régions les plus similaires sont fusionnées en premier.

Dans le même ordre d'idée, O. Monga et B. Wrobel [Monga, 1990] proposent un algorithme de division/fusion dont la segmentation résultante peut être qualifiée de sous optimale. Pour chaque critère de fusion, ils introduisent une fonction caractérisant la qualité globale de la segmentation. L'optimisation se fait localement en déterminant à chaque itération le meilleur couple de régions adjacentes à fusionner (celui qui minimise un coût local de fusion). A cette fin, ils proposent d'utiliser la structure de graphe d'adjacence des régions qui évite de parcourir toute l'image à chaque itération et qui permet une remise à jour facile après chaque fusion. Dans une telle structure, les nœuds du graphe sont constitués par les régions, affectées de leurs divers attributs radiométriques, topologiques....., et les arcs entre nœuds définissent l'adjacence entre les régions : deux nœuds sont reliés par un arc si les deux régions sont adjacentes. A chacune des étapes de fusion, il faut choisir le meilleur arc, c'est-à-dire choisir parmi tous les couples de régions vérifiant le prédicat de fusion celui qui minimise la fonction de qualité locale. De plus, lors de la réalisation de la fusion de deux nœuds, il faut mettre à jour tous les attributs des nœuds et des arcs modifiés par cette fusion. Pour permettre

un accès rapide au meilleur arc, un arbre binaire de recherche est utilisé. La remise à jour du graphe après chaque fusion est facile. Les attributs du nouveau nœud créé et des arcs correspondants sont calculés directement à partir de ceux des deux nœuds fusionnés.

T. Pavlidis et Y. T. Liow [Pavlidis, 1990] présentent une méthode de segmentation en combinant les résultats obtenus par la méthode de croissance de régions et de contours actifs. L'algorithme se décompose en trois étapes. La première est une procédure de division-fusion (Split and Merge) de Pavlidis, qui permet d'obtenir une sur-segmentation de l'image. La seconde est une procédure de fusion des régions adjacentes contrôlée par une information de contraste le long de la frontière commune entre les deux régions. Dans [Bertolino,1995] Bertolino propose une segmentation pyramidale et utilise une information de gradient lors de l'évaluation de la fusion des régions du graphe d'adjacence. Bonnin [Bonnin, 1991] propose une méthode de segmentation incrémentale qui fait coopérer en parallèle des techniques de segmentation en régions homogènes et de détection de contours tout en s'échangeant des informations. Les deux cartes de segmentation obtenues sont alors exploitées pour une reconstruction 3D efficace.

L'inconvénient majeur du Quadtree réside dans la rigidité du découpage carré qu'il impose. Il conduit à un partitionnement global de l'image qui ne respecte pas toujours la forme des régions présentes dans l'image. Par ailleurs, la phase de regroupement des blocs est sensible à l'ordre du parcours du Quadtree.

Afin de résoudre ce problème le partitionnement de Voronoi est utilisé [Chassery, 1991] tout d'abord des germes initiaux sontensemencés sur l'image et servent de noyaux pour la détermination des polygones de Voronoi. Ensuite, de nouveaux germes sont introduits à l'intérieur des polygones non homogènes. Cette étape de division est itérée jusqu'à ce que tous les polygones du diagramme deviennent homogènes. A la fin du processus de segmentation, les polygones adjacents et similaires sont fusionnés. L'ajustement du seuil de fusion reste délicat et l'initialisation des germes peut mener à des résultats de segmentation différents.

#### 4.4.2.4 Conclusion

- Les méthodes de fusion de régions sont faciles à implémenter, mais introduisent un temps de calcul trop coûteux car elles impliquent à chaque itération, la recherche des couples de régions à fusionner. Elles se distinguent selon l'ordre de parcours des différents nœuds du graphe et selon les critères de fusion [Beveridge, 1989]. Un autre inconvénient des méthodes de fusion de régions est qu'elles sont très sensibles au bruit car reposent sur des petites

régions à fusionner (ou bien des pixels à agréger) et que ces dernières véhiculent peu d'informations. De plus, la segmentation dépend fortement des germes initiaux..

- Les méthodes de division sont efficaces surtout pour des images simples qui contiennent peu de régions, et dont l'histogramme présente une bonne séparation entre les pics afin de faciliter le choix de seuil. D'un autre côté, elles génèrent un effet de sursegmentation dès qu'une petite variation du critère d'homogénéité est détectée car elles reposent sur des statistiques globales.
- Les méthodes de division / fusion ont un temps de calcul moins important que les autres méthodes, étant donné que le nombre de régions à fusionner peut être réduit au cours de l'étape de division. Cependant elles sont peu robustes et sensibles aux résultats obtenus par la phase de division. D'un autre côté, cette classe de méthodes conduit à des algorithmes facilement parallélisables.

#### **4.4.3 Les méthodes de classification**

Les méthodes de segmentation d'images par classification sont issues du domaine de classification des données multidimensionnelles. Elles définissent une partition de 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 [Jain, 1999]. 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.

Formellement, une segmentation  $I$  en  $NC$  classes  $C_i$  composées chacune de  $R_{C_i}$  régions est définie comme suit :

$$I = \bigcup_i C_i$$

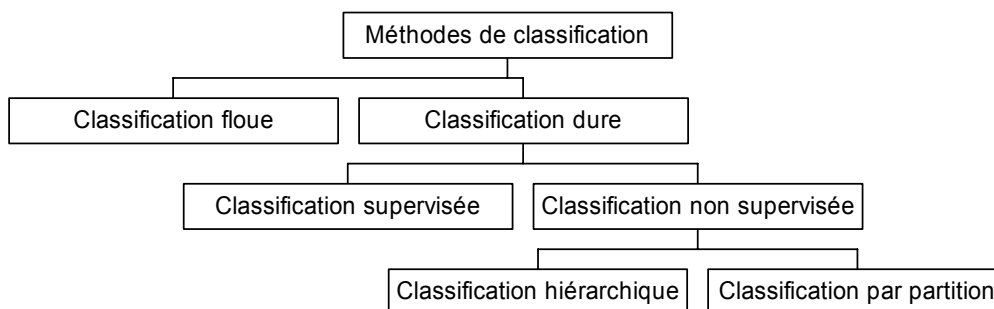
$$C_i = \bigcup_k R_{C_i}^k, \quad \forall i$$

$$C_i \neq \emptyset \quad \forall i ;$$

$$C_i \cap C_j \quad \forall i \neq j ;$$

Il n'existe pas une méthode de classification qui peut s'appliquer à tout type d'image et qui peut fournir un partitionnement optimal et le plus naturel possible. Ce qui explique la grande diversité de méthodes de classification qui existe dans la littérature. Le choix d'une méthode est déterminé par différents facteurs tels que le nombre de classes attendues, la forme des classes extraites ou encore le chevauchement ou non des classes [Jain, 2000].

Il est possible de regrouper les méthodes de classification sous la forme d'une hiérarchie de méthodes appelée taxonomie. Nous présentons ci-après la taxonomie inspirée de celle de Jain et al. dans [Jain, 1988 ; Jain, 1999] (voir Figure 4.8.).



**Figure 4.8. Taxonomie de la classification inspirée de celle de Jain et al.**

Les méthodes de classification peuvent être divisées en méthodes *dures* et méthodes *floues*. Dans une méthode de classification dure, un pixel ne peut être affecté qu'à une seule classe dans la partition de l'image. Dans une méthode de classification floue, on affecte au pixel un degré d'appartenance pour chacune des classes de la partition qui indique la probabilité que le pixel y appartienne. La classe finale du pixel sera celle pour laquelle son degré d'appartenance est le plus élevé.

Dans la *classification supervisée*, le nombre de classes est connu et on dispose d'un ensemble de pixels déjà étiquetés, servant d'ensemble d'apprentissage. Il s'agit alors de pouvoir associer chaque nouveau pixel à la classe la plus adaptée en se servant des pixels déjà étiquetés. Dans la *classification non supervisée*, aucune information sur le nombre et le



contenu des classes possible n'est fournit. L'objectif est alors de pouvoir regrouper automatiquement des pixels considérés similaires dans une même classe. Dans ce cas il s'agira de définir une fonction de similarité entre pixels qui sera maximum entre les pixels d'une même classe et minimum avec ceux des autres classes.

Les *méthodes hiérarchiques* produisent une hiérarchie de classes, telle que plus on descend dans la hiérarchie plus on trouve des classes homogènes. A l'opposé, les *méthodes de partitionnement* déterminent une seule partition.

Les méthodes de classification se différencient par la fonction de distance utilisée pour regrouper les pixels de l'image en des classes, la fonction objectif utilisée pour évaluer la qualité de la classification obtenue et par la stratégie de regroupement et de parcours de l'espace des classes possibles. Le processus de classification implique l'exécution des tâches suivantes [Jain,1988]:

- 1 Représentation des données de la classification ( inclut l'extraction des attributs des pixels servant à la classification tels que niveau de gris, paramètres de texture, couleur, etc) ;
- 2 Définition d'une distance de similarité entre les pixels (exemple distance euclidienne entre les niveaux de gris des pixels) ;
- 3 Regroupement des pixels en des classes ;
- 4 Evaluation de la classification obtenue ;

La stratégie de regroupement peut être :

**Ascendante versus descendante** : la méthode ascendante démarre avec un nombre de classes important (peut être égale au nombre de pixels de l'image) puis réduira ce nombre en fusionnant selon un processus récursif les classes proches jusqu'à ce qu'un critère d'arrêt soit vérifié. A l'inverse, la méthode descendante part d'une seule classe regroupant tous les pixels qu'elle divisera successivement en de nouvelles classes tant qu'un critère d'arrêt est vérifié.

**Déterministe versus stochastique** : Cette propriété concerne en particulier les méthodes de classification par partitionnement qui tentent d'optimiser la fonction objective mesurant la qualité de la partition obtenue. Cette optimisation peut être accomplie en utilisant des

techniques traditionnelles ou à travers une recherche aléatoire de l'espace de solutions contenant tous les étiquetages possibles.

Dans le paragraphe suivant, nous décrivons d'une manière détaillée le problème de partitionnement car c'est celui nous allons essayer de résoudre grâce à des approches stochastiques inspirées des comportements collectifs observés chez les fourmis.

#### 4.4.3.1 Le problème de partitionnement

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. Il a été montré dans [Ostrovsky, 2000; Leung, 2000] que c'est un problème d'optimisation NP-difficile. 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 [Jain, 1988]:

$$S(N,K) = \frac{1}{K!} \sum_{i=1}^K (-1)^{K-i} \binom{K}{i} i^N \quad (4.1)$$

On peut se 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.

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 :

$$I = I_{intra} + I_{inter} \quad (4.2)$$

$$I_{intra} = \sum_{j=1}^K \sum_{i=1}^{|c_j|} d^2(x_i^j, \mu_j) \quad (4.3)$$

$$I_{inter} = \sum_{j=1}^K |c_j| d^2(\mu_j, \mu) \quad (4.4)$$

avec :

$x_i^j$  :est le vecteur caractéristique du ième pixel de la classe  $c_j$  ;

$\mu_j$  :est le vecteur de caractéristiques moyen de la classe  $c_j$  ;

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

$\mu$  est le vecteur de caractéristiques moyen de l'image à segmenter ;

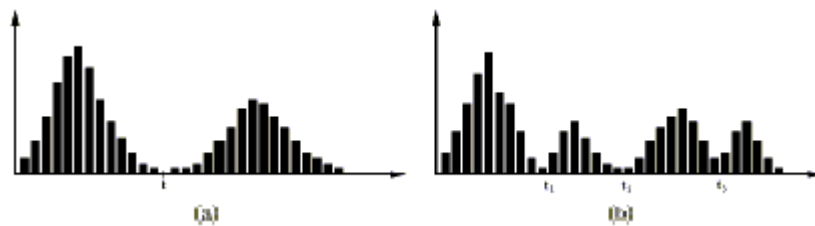
$d(.)$  est une mesure de distance, la plus connue est la distance euclidienne.

Si le vecteur de caractéristiques est réduit à l'information niveau de gris alors  $x_i^j$ ,  $\mu_j$  et  $\mu$  représentent resp. le niveau de gris du ième pixel de la classe  $c_j$ , le niveau de gris moyen de classe  $c_j$  et le niveau de gris moyen de l'image.

Les méthodes de classification par partitionnement sont fondées sur un processus itératif qui converge vers une partition qui optimise un critère prédéfini [Hamerly, 2002]. Le nombre de classes est spécifié a priori ou bien déterminé comme une partie de la méthode. Dans la suite du paragraphe nous passons en revue les méthodes de partitionnement les plus connues.

#### 4.4.3.2 Seuillage d'histogramme

La segmentation par le seuillage est la plus ancienne et la plus populaire de toutes les méthodes de classification [Gonzalez, 1992; Jain, 1995]. Le seuillage 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, l'histogramme est bimodal et il est possible de séparer les régions du fond par un simple seuil qui se situe sur la vallée entre les deux modes (figure 4.9 (a)). Si l'image est constituée de régions ayant des caractéristiques différentes, son histogramme est multimodal (figure 4.9 (b)), tels que chacun des modes correspond à une région. Le seuillage est alors défini comme le problème de trouver le minimum entre deux modes adjacents c'est-à-dire les vallées. Les pixels ayant des niveaux de gris entre deux vallées sont alors assignés à la même région.



**Figure 4.9. Exemples d'histogrammes. (a) Histogramme bimodal. (b) Histogramme multi-modal**

D'une manière générale, le choix de seuils d'histogramme peut se faire de deux manières différentes. Dans le premier procédé, les seuils sont déterminés globalement pour tous les points de l'image, tandis que dans le second procédé, les seuils sont calculés localement à partir de petites fenêtres de focalisation. Nous allons passer en revue les techniques les plus utilisées dans la littérature pour déterminer les seuils d'histogramme.

### *Méthodes globales*

Les méthodes de segmentation par seuillage global déterminent un seul seuil pour segmenter l'image entière. La valeur optimale du choix du seuil peut être calculée à partir des niveaux de gris de chaque pixel ou bien dépendre du voisinage immédiat de chacun d'eux [Kermad 1997]. Dans [Koller, 1995] les auteurs utilisent comme valeurs de seuils, celles qui minimisent la somme des variances de niveau de gris de chacune des régions dont le nombre est fixé par l'utilisateur. Quand les vallées sont difficiles à extraire à partir de l'histogramme de niveau de gris les seuils peuvent être déterminés en analysant la concavité de l'histogramme [Rosenfeld, 1983]. Pour améliorer la forme de l'histogramme et faciliter le choix du seuil, certaines techniques sélectionnent les pixels à utiliser pour le calcul de l'histogramme ou bien introduisent d'autres informations que le niveau de gris des pixels. Par exemple dans [Watanabe, 1974], Watanabe propose de sélectionner une valeur de seuil qui maximise la somme des gradients calculée sur tous les points dont le niveau de gris est égal à la valeur du seuil. Kohler s'est inspiré de cette idée et propose de sélectionner le seuil qui détecte le plus grand nombre de contours à fort contraste et le plus petit nombre de contours à faible contraste [Kohler, 1981]. Weszka reprend ces idées et utilise en plus des niveaux de gris des pixels, une information gradient de chacun d'eux [Weszka, 1974]. Dans son algorithme L l'histogramme n'est calculé que pour les pixels à faible gradient, ce qui permet de creuser les vallées entre les pics de l'histogramme. Ceci permet de tenir compte des pixels appartenant à la frontière entre les régions et le fond. De même Katz propose de sélectionner les pixels avec forte valeur de gradient et de choisir la valeur de gris moyenne de ces pixels [Haralich, 1985]. Toujours dans le même ordre d'idée Hertz et al. présentent dans [Hertz, 1988] une méthode de segmentation par coopération d'une technique de seuillage de l'histogramme et un opérateur de détection de contours. Le principe est d'extraire de l'histogramme de l'image, les seuils qui assurent le meilleur contraste sur les frontières des régions détectées ou une meilleure

coïncidence entre ces frontières et des contours préalablement extraits par un détecteur de contours. D'autres méthodes de seuillage se basent sur la théorie de l'information pour le calcul des seuils [Kapur, 1985 ; Pun, 1981]. Dans [Kermod, 1995], une coopération entre une technique de seuillage et celle de fusion de régions est proposée. Une segmentation initiale est obtenue par seuillage des histogrammes locaux extraits sur des petites régions de l'image à traiter. Après, une fusion de régions étiquetées minimisant un critère de similarité est effectuée.

### ***Méthodes locales***

À beaucoup d'occasions, quand une image est bruyante ou l'illumination n'est pas bonne, les seuils fixes ne sont pas capables de segmenter l'image sûrement. Les méthodes de seuillage adaptatives ont été développées pour traiter de tels problèmes [Haralick, 1985]. Une technique classique pour déterminer les seuils localement est présentée dans [Chow, 1972], l'algorithme partage l'image en de petites fenêtres de taille 33\*33 ou 65\*65 et calcule pour chacune d'elles la variance de niveau de gris et l'histogramme. Si la variance dépasse un certain seuil et si l'histogramme n'est pas uni-modal, la fenêtre est partagée en utilisant un seuil dans la vallée de celui ci. Puis par interpolation, ils déterminent un seuil pour chaque pixel de l'image. Dans [Ohlander, 1975 ; Ohlander, 1979], Ohlander propose une méthode de seuillage par division récursive de l'image. A partir de l'histogramme de l'image originale, il décompose l'image en régions telle que pour chacune d'elles un histogramme est calculé. Si l'histogramme d'une région n'est pas uni-modal, elle est redivisée sinon elle reste inchangée et ainsi de suite. Le processus se termine quand tous les histogrammes sont uni-modal. Cette technique a été adaptée par Ohta a adapté cet algorithme pour la segmentation d'images couleur [Ohta, 1980].

Pour plus de détails sur les méthodes de segmentation par seuillage, voir les surveys de Weszka [Weszka, 1978], Fu et Mui [Fu, 1980], Haralick et Shapiro [Haralick, 1985], Sahoo et al [Sahoo, 1988] ainsi que l'étude comparative de Lee et Chung [Lee, 1990].

#### 4.4.3.3 L'algorithme des centres mobiles et ses variantes

L'algorithme des centres mobiles [Forgy, 1965] est un algorithme de partitionnement des plus simples et des plus connus. Partant de centres de classes choisis arbitrairement parmi les pixels de l'image, le regroupement de pixels est itératif et est obtenu par minimisation de l'inertie intraclasse (voire Eq.4.1). L'algorithme 4.1 résume les étapes de classification par les centres mobiles tels que décrits par Tou et Gonzalez [Tou, 1974] :

*Algorithme 4.1 Algorithme des centres mobiles*

*Soit  $K$  le nombre de classes;*

*Initialiser chaque centre de classe  $c_k$   $k=1..K$  à une valeur initiale arbitraire*

***Tant que*** le critère d'arrêt n'est pas vérifié ***faire***

*Affecter chaque pixel  $i$  est affecté à la classe la plus proche :*

$$p \in c_k \text{ si } d^2(x_i^k, \mu_k) < d^2(x_i^j, \mu_j) \quad j=1..K \quad j \neq k$$

$$\text{mise à jour des centres des classes: } c_k = \frac{1}{|c_k|} \sum_{i \in c_k} x_i^k \quad k=1..K \quad ;$$

***Fin tantque***

*Retourner la partition générée*

L'algorithme s'arrête quand aucun changement significatif des centres des classes n'est observé d'une itération à une autre ou bien quand un maximum nombre d'itération est obtenue.

Une version légèrement différente des « centres mobiles » est connue sous le nom de « nuées dynamiques » [Diday, 1971] ou bien sous le nom de « k\_medoids ». Il s'agit d'une extension de l'algorithme des centres mobiles telle que une classe n'est plus définie par son centre, mais représentée par un groupe de pixels formant un « noyau » [Hamerly, 2003 ; Halkidi, 2001]. Le k-means [MacQueen, 1967] est différent de celui des centres mobiles par le fait qu'à chaque itération, chaque nouvelle affectation d'un pixel à une classe entraîne la remise à jour immédiate de son centre de gravité.

On notera que ce genre d'algorithmes exige la connaissance du nombre de classes [Lee 2000; Hamerly 2003]. Connaître le nombre « optimum » de classes présentes dans l'image est un problème complexe et nécessite une connaissance a priori sur les objets constituant l'image qui n'est pas toujours disponible. Afin de s'affranchir de cette contrainte, différents algorithmes de classification dynamique ont été proposés dans la littérature. Parmi ces algorithmes on cite l'algorithme ISODATA, (Iterative Self-Organizing Data Analysis Techniques A) [Ball, 1967] et l'algorithme Dynoc (Dynamic Optimal Cluster seek) [Tou, 1979] qui introduisent dans l'algorithme de classification une heuristique qui permet de modifier le nombre de classes au cours des itérations de la façon suivante:

– Lorsqu'une classe présente une inertie intraclasse supérieure à un seuil fixé par l'utilisateur, la classe est divisée en deux,

– Lorsque la distance entre les centres de gravité de deux classes devient inférieure à un autre seuil, les classes sont fusionnées.

Une autre variante du K-means est l'algorithme des moyennes floues (fuzzy c-means) ou *FCM*[Bezdeck, 1981]. La partition floue est basée sur la théorie des ensembles flous défini par Zadeh [Zadeh, 1965] qui décrit la manière dont sont structurés les ensembles flous et quelles sont les opérations permises sur ces ensembles. En traitement d'images, les méthodes de classification floue modélisent le degré d'appartenance d'un pixel à chaque classe, sous l'hypothèse que ce pixel puisse ne pas appartenir à une classe unique. Chacun des  $N$  pixels appartient à chacune des  $C$  classes avec un coefficient d'appartenance  $u$ ; L'ensemble des degrés d'appartenance est stocké dans la matrice de la C-partition floue  $U$ .

$$U = [u_{ik}]_{i=1..N, k=1..C} \quad (4.5.)$$

$u_{ik}$  étant le degré d'appartenance de pixel  $i$  à la classe  $k$  tel que

$$u_{ik} = \left( \sum_{j=1}^C \left( \frac{\|x_i - \mu_k\|}{\|x_i - \mu_j\|} \right)^{\frac{2}{m-1}} \right)^{-1} \quad (4.6.)$$

Le paramètre  $m(1 < m < 3)$  contrôle le degré de flou de la partition produite : plus  $m$  est grand plus la partition sera floue. Le choix du paramètre  $m$  reste une question délicate, mais en pratique la valeur 2 est souvent utilisée.

Trois contraintes sont définies pour la C-partition floue. La première impose que les valeurs des  $u_{ik}$  soient comprises entre 0 et 1 ce qui signifie que tout pixel appartient à la réunion des classes. La deuxième impose que l'appartenance d'un pixel soit répartie sur l'ensemble des classes, sans exclure l'existence de degrés d'appartenance nuls. Enfin, la troisième exprime que la classification ne produise pas de classe vide afin que tous les pixels soient caractérisés. Formellement, ces contraintes s'écrivent comme suit:

$$0 \leq u_{ik} \leq 1 \quad \forall i \in [1; c], \forall k \in [1; n] \quad (4.7.)$$

$$\sum_{i=1}^c u_{ik} = 1 \quad \forall k \in [1; n] \quad (4.8.)$$

$$0 < \sum_{k=1}^N u_{ik} < 1 \quad \forall i \in [1; C] \quad (4.9.)$$

L'objectif de l'algorithme *FCM* est de trouver  $U$  qui minimise la fonction objective suivante :

$$J_{FCM} = \sum_{k=1}^K \sum_{i=1}^N U_{ik} * \|x_i - \mu_k\|^2 \quad (4.10.)$$

Enfin, une ultime étape est nécessaire lorsque le résultat souhaité est une classification non floue. On parle alors de « défuzzification » telle que chacun des pixels est affecté à la classe pour laquelle son degré d'appartenance est le plus élevé.

Depuis son apparition la classification floue est devenue très populaire en traitement d'image en générale et en segmentation plus particulièrement. Dans [Lim, 1990], les auteurs proposent de segmenter des images selon une décomposition « coarse to fine » en effectuant une première segmentation par multiseuillage, suivie d'une segmentation plus fine par application de l'algorithme FCM. Chi et Yan [Chi, 1995] ont développé une méthode de segmentation à partir de règles de classification floues basés sur trois mesures : la différence d'intensité, l'écart type et une mesure de contraste locale entre le niveau de gris le plus foncé et le fond de l'image. Une extension de l'algorithme classique FCM est présenté dans [Liew, 2000] en prenant en compte l'information spatiale locale. L'utilisation de l'information spatiale locale permet d'améliorer les résultats de classification et fournit des résultats de segmentation satisfaisants.

Dans [Bouloudani, 2002] les auteurs traitent le problème de la segmentation automatique des images couleurs. L'algorithme proposé est similaire à ISODATA. Partant d'un résultat de classification obtenu à partir de l'algorithme FCM, l'algorithme appliqué d'une manière itérative, essaye de fusionner deux classes si leur distance est comparable à leur compacité. La signification quantitative du terme « comparable » est donnée par un paramètre  $\alpha$ . Ce paramètre fixe le niveau de fusion désiré, et par conséquent le niveau de simplification de la segmentation.



#### 4.4.3.4 Classification par les plus proches voisins

Depuis son introduction par Lu et Fu [Lu, 1978], l'algorithme de partitionnement par plus proches voisins a été largement utilisé pour la classification de données. Cette popularité est due en particulier à sa facilité de mise en oeuvre et à la bonne qualité des résultats obtenus [Theodoridis, 1999]. La règle d'affectation d'un pixel non étiqueté à une classe est la suivante :

- Déterminer les  $K$  plus proches pixels présents dans un voisinage immédiat sans tenir compte de leur classe;
- Relever ensuite le  $nn_k$  de ces pixels appartenant à chaque classe  $C_k$   $k=1\dots q$  du problème )  $q$  classes;
- Affecter le pixel à la classe  $C_i$  avec le nombre  $nn_i$   $i=1\dots q$  maximum.

Ce processus est répété jusqu'à ce que tous les pixels soient assignés à des classes [Jain, 1999].

#### 4.4.3.5 Classification par les champs de Markov

Contrairement aux méthodes de classification précédentes, les approches de classification par Champ de Markov permettent de faire évoluer une partition de l'image en intégrant deux types d'informations : le niveau de gris du pixel et la classe de ses voisins. Dans le cadre des champs de Markov, une image à segmenter  $Y$  est considérée comme une dégradation d'une image parfaite  $E$  contenant les étiquettes idéales des pixels. La segmentation est vue comme la restauration de l'image  $E$  à partir de l'image observée  $Y$ .

L'intérêt des champs de Markov est qu'ils offrent un cadre mathématique puissant qui permet une corrélation spatiale des étiquettes entre pixels voisins et donc de régulariser l'étiquetage [Chellappa, 1993]. De plus le théorème de Hameserly [Geman, 1984] a permis de simplifier l'écriture des probabilités conditionnelles et faciliter la détermination des paramètres qui spécifient le modèle Les interactions au niveau du point sont prises en compte par le biais d'énergies potentielles.

L'intérêt des champs de Markov est qu'ils offrent un cadre mathématique puissant qui permet une corrélation spatiale des étiquettes entre pixels voisins et donc de régulariser l'étiquetage. De plus le théorème de Hameserly [Geman, 1984] a permis de simplifier l'écriture des probabilités conditionnelles et faciliter la détermination des paramètres qui spécifient le

modèle Les interactions au niveau du point sont prises en compte par le biais d'énergies potentielles.

Dans le cadre de champs de Markov et de la théorie Bayésienne, on peut écrire :

$$P(E=e|Y=y) = \frac{P(E=e)P(Y=y|E=e)}{P(Y=y)} \quad (4.11.)$$

En éliminant la constante  $P(Y=y)$ , la probabilité a posteriori peut s'écrire sous la forme :

$$P(E=e|Y=y) \propto P(E=e)P(Y=y|E=e) \quad (4.12.)$$

En utilisant la distribution de Gibbs, il vient:

$$\frac{1}{Z} \exp[-U(E=e/Y=y)] \propto \frac{1}{Z_1} \exp[-U(E=e)] \times \frac{1}{Z_2} \exp[-U(Y=y/E=e)] \quad (4.13.)$$

Soit, en prenant le Logarithme de chaque terme :

$$U(E=e/Y=y) \propto U(E=e) + U(Y=y/E=e) \quad (4.14.)$$

La segmentation d'images est ramenée à un problème d'optimisation pour lequel on cherche à obtenir pour chaque pixel une étiquette optimale  $e^*$ . Plusieurs estimateurs existent dans la littérature, le plus communément utilisé est l'estimateur *MAP* : Maximum a posteriori [Geman, 1984]. Cette méthode revient à chercher l'étiquetage  $\hat{e}$  qui maximise la probabilité  $P(E=e|Y=y)$  et donc qui minimise l'énergie globale  $U(E=e/Y=y)$  qui représente l'erreur d'étiquetage.

C'est un problème d'optimiser difficile car d'une part la fonction d'énergie globale est loin d'être convexe, d'une autre part la minimisation doit se faire sur l'ensemble des configurations de toutes les étiquettes possibles dont le cardinal est très grand. Pour résoudre ce problème deux classes de méthodes ont été proposées, les algorithmes déterministes de type ICM [Besag, 1974] (Iterated Conditionnel Mode) et les algorithmes stochastiques de type Recuit Simulé [Geman, 1984; Hu, 1992] et Algorithmes Génétiques [Andrey, 1998; Kim, 2000]. Un aperçu de la modélisation markovienne est présenté en annexe .

Depuis leur apparition, les champs de Markov ont été largement utilisés en segmentation d'images. Nous pouvons citer les travaux de [Marroquin, 1985] [Lakshmanan, 1989; Derin, 1987; Descombe, 1999; Kervrann, 1995; Panjwani, 1995; Krishnamachari, 1997] [Yamazaki, 1995] [Jaggi, 1998]. Afin d'améliorer la qualité de la segmentation et accélérer la convergence du processus d'étiquetage, d'autres travaux utilisent une relaxation markovienne avec une approche multi-échelle [Perez, 1992; Bouman 1994] ou bien une approche hiérarchique [Kato, 1994; Chardin, 2000].

#### 4.4.3.6 Conclusion

- Les techniques de seuillage sont simples à mettre en œuvre et donnent de bons résultats pour les images bien contrastées. Elles exploitent exclusivement les informations de niveau de gris sans tenir compte de la position des pixels, ce qui, dans le cas d'images bruitées, crée des contours très chahutés et une sur-segmentation de l'image. La taille des régions extraites dépend du choix des seuils qui est souvent une opération délicate (les modes de l'histogramme ne sont toujours bien séparés par des vallées) ;
- Les méthodes de classification par partition recherchent pour chaque pixel de l'image la classe la plus proche en sens d'une distance et convergent vers un minimum local de l'inertie totale. Ce minimum ne correspond pas nécessairement au minimum global recherché et dépend de la position initiale des centres de gravité. De plus certaines méthodes impliquent un nombre  $k$  de classes à fixer a priori et distribuent les pixels sur l'ensemble de ces classes en ignorant l'information spatiale ;
- Les méthodes de classification par champ de markov sont très puissantes pour analyser les images car elles tiennent compte de la présence du bruit. Le modèle markovien permet une modélisation simple et pertinente des contextes et interactions locales de pixels, mais exige un nombre considérable d'itérations pour converger vers des solutions stables ce qui augmente le temps de calcul.

### **4.4.4 Les méthodes de segmentation biomimétiques**

#### 4.4.4.1 Segmentation par les réseaux de neurones

Depuis quelques années, les chercheurs informaticiens se sont inspirés du fonctionnement du cerveau humain pour proposer une nouvelle technique de segmentation d'images par des réseaux de neurones artificiels. Les cellules d'entrées du réseau correspondent chacune à un attribut des pixels (réduit au niveau de gris la plupart du temps). Les cellules de sortie donnent la classe du pixel.

Les réseaux de neurones ont été utilisés aussi bien pour la détection de contours que pour la classification de pixels. Par exemple on peut citer les travaux de Cortes [Cortes, 1989], ceux de Babaguchi [Babguchi,1990] qui utilise un réseau de neurones multicouches pour l'extraction des points de contour. Ces idées ont été reprises par Pinho [Pinho, 1996] qui utilise comme entrées du réseau les différences de niveau de gris entre les pixels adjacents. Dans [Pontecorvo, ], les auteurs utilisent un autre type de réseau de neurones, appelé Shunting Inhibitory Cellular Neural Network pour effectuer une détection de contours. Dans ce type de réseau, chaque cellule est connectée à celles situées à une certaine distance par rapport à elle.

Cette cellule peut être inhibée par celle de son voisinage. Dans [Yin, 1993], les auteurs présentent une série de tests effectués sur plusieurs types de réseaux de neurones effectuant le filtrage d'une image en éliminant le bruit qu'elle contient. Un état de l'art sur les méthodes basées réseaux de neurones pour la détection de contours peut être trouvée dans [Devaux, 1997].

Concernant le problème de la classification, différents modèles de réseaux de neurones ont été proposés dans la littérature. Dans [Wu, 1994] un algorithme de classification combinant l'algorithme FCM et les réseaux de neurones a été proposé pour la segmentation des images couleur. L'algorithme FCM est appliqué en premier pour fournir une partition initiale. Ensuite, un réseau de neurone avec un apprentissage supervisé est utilisé pour optimiser les centres de classes ainsi obtenus. Ces nouveaux centres sont alors utilisés pour segmenter l'image en utilisant la règle des plus proches voisins [Turi, 2001].

Dans le même ordre d'idée, Elkharraz propose d'utiliser un réseau de neurones à apprentissage compétitif flou pour segmenter des images couleurs de mosaïques marocaines [Elkharraz, 1999]. La structure d'un tel réseau comporte deux couches, une couche d'entrée et une de sortie. La couche d'entrée contient un nombre de neurones égal au nombre de composantes colorimétriques d'un pixel. Le rôle de chacun de ces neurones est de transmettre, à chaque itération de l'apprentissage, une composante du pixel à traiter vers l'ensemble des neurones de la couche de sortie. Le vecteur poids de chaque neurone de la couche de sortie sert à représenter une classe.. La phase d'apprentissage consiste à modifier ces vecteurs poids de façon itérative jusqu'à ce qu'ils convergent vers les centres des classes qu'ils représentent.

Les cartes de kohonen appelés aussi SOM (Self Organised Maps) [Kohonen, 1984], sont assez proches de l'algorithme K-means et peuvent être utilisées pour trouver automatiquement le nombre de classes sans aucune supervision [Pandya, 1996; Moreira, 1996]. Dans [Abrantes, 1996], les auteurs utilisent les cartes de Kohonen pour la détection de contours. Le réseau prend, comme données d'entrée, les vecteurs de position des arêtes préalablement déterminées à l'aide d'un opérateur de détection d'arêtes et fournit en sortie une image contours. Bélanger [Bélanger, 1998] remplace une information rigide que sont les arêtes par une information plus riche que sont les gradients de l'image.

#### 4.4.4.2 Segmentation d'images par les algorithmes génétiques

Le problème de partitionnement étant un problème d'optimisation combinatoire, il est naturel de l'aborder donc à travers des techniques dédiées pour la résolution de ce genre de problèmes tels que les algorithmes génétiques (voir chapitre 2).

Les premiers travaux utilisant les AG pour résoudre le problème de classification sont dus à [Raghavan, 1979]. L'algorithme démarre avec un nombre de classes  $K$  est fixé à l'avance et une population de chromosomes de longueur  $N$  (taille de l'image à segmenter). Chaque chromosome associe une classe à un pixel. Les opérateurs génétiques, utilisés pour générer à chaque génération une population de partitions possibles, sont une adaptation des opérateurs génétiques binaires. Le but de l'algorithme est de minimiser une fonction fitness représentant l'inverse de la variance intraclasse.

Plusieurs variantes de cet algorithme de base ont été introduit par d'autres auteurs dans la littérature. Bhandarkar et Zhang [Bhandarkar, 1999] utilisent l'algorithme génétique pour minimiser une fonction fitness qui sera utilisée pour évaluer le résultat de la segmentation. La population initiale est générée par un processus aléatoire. La représentation de chaque chromosome contient un tableau d'étiquette, une information contour et un graphe d'adjacence. Un crossover de deux points est appliqué sur le tableau d'étiquettes et une mutation sur les contours de pixels sont employés. Dans [Maulik 2003] un partitionnement flou est réalisé en utilisant un algorithme génétique de codage réel. Une classification automatique sans connaissance a priori du nombre de classes présentes dans l'image est obtenue dans [Murthy, 1996 ; Tseng, 2001; Bandyopadhyay, 2002]. Dans [Andrey, 1998], un algorithme génétique distribué est utilisé pour segmenter des images texturées dans le cadre des champs de Markov. Le système de segmentation repose sur un principe évolutionnaire tel que la segmentation optimale émerge progressivement des interactions mutuelles entre les chromosomes de la population sans aucune information fournit a priori. Le système d'apprentissage génétique proposé par [Bhanu, 1995] permet d'adapter le processus de segmentation en fonction du type d'image à segmenter et des caractéristiques des objets à extraire. Dans [Gong, 1999], les auteurs proposent un algorithme génétique pour la segmentation des images couleur en utilisant la structure Quadtree.

D'autres auteurs ont proposé des approches hybrides en utilisant conjointement les algorithmes génétiques et des algorithmes de classification classiques tels que K-Means et le FCM. Ces algorithmes sont utilisés soit en aval de l'algorithme génétique afin de leur fournir une partition initiale de bonne qualité [Babu, 1993], soit alternativement avec l'AG. Une autre possibilité est qu'ils servent d'opérateurs génétiques (opérateur de mutation par exemple) afin d'accélérer la convergence des AG [Krishna, 1999]. Dans [Chun, 1996] Chun et Yang utilisent les algorithmes génétiques pour segmenter des images médicales en minimisant une fonction fitness floue. L'algorithme FCM est utilisé pour générer une segmentation initiale. Chaque chromosome de la population est codé par  $n$  entiers ( $n$  étant le nombre de régions). Un opérateur de croisement de deux point et une mutation dynamique sont utilisés ensuite pour optimiser le résultat de l'algorithme FCM à travers un processus de division/fusion.

#### 4.4.4.3 Segmentation par les systèmes multi-agents

Les systèmes Multi-agents (SMA) sont nés au début des années 80 et se sont développés à partir de schémas de raisonnement ou d'organisations empruntés aux domaines de la vie et de la société. Un SMA est un ensemble d'agents autonomes qui travaillent ensemble (coopérativement ou parallèlement) pour résoudre un problème en s'appuyant sur les capacités et les connaissances individuelles de chaque agent. De ces actions individuelles, émerge la solution ou le comportement général du système.

Un agent peut être aussi bien un être physique évoluant dans le monde réel, qu'un être artificiel évoluant dans un monde virtuel. Tout agent (réel ou virtuel) ne se contente pas de raisonner, il agit. L'action est un comportement tendant à satisfaire au mieux les buts de l'agent, au vu de la perception et de la représentation de l'environnement dont il dispose, des ressources et des compétences qu'il possède et de la teneur des informations échangées. Etant guidés par des objectifs, les agents sont dits autonomes dans le sens où c'est eux les seuls pilotes de leur comportement, en décidant à chaque instant de l'action adéquate par rapport à un éventail de possibilités données.

Un système multi-agent (SMA) est défini dans [Ferber, 1995] comme un système composé des éléments suivants :

- un **environnement**  $E$ , c'est à dire un espace disposant généralement d'une métrique ;
- un ensemble d'**objets**  $O$ . Ces objets sont situés, c'est à dire que pour tout objet, il est possible, à un moment donné, d'associer une position dans  $E$ . Ces objets sont passifs, c'est à dire qu'ils peuvent être perçus, créés, détruits et modifiés par les agents ;
- un ensemble  $A$  d'**agents**, qui sont des objets particuliers ( $A \subseteq O$ ), lesquels représentent les entités actives du système. Ces agents possèdent les propriétés spécifiées plus haut. ;
- un ensemble de **relations**  $R$  qui unissent des objets (et donc des agents) entre eux ;
- un ensemble d'**opérations**  $Op$  permettant aux agents de  $A$  de percevoir, produire, consommer, transformer et manipuler des objets de  $O$  ;
- des opérateurs chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification, que l'on appellera les lois de l'univers ;

Les systèmes multi-agents offrent une architecture permettant de faire travailler plusieurs entités autonomes sur un même problème (en coopération ou en parallèle) à l'aide de protocoles de communication et de processus d'échange d'informations. Le paradigme Multi-agents a été largement utilisé avec succès ces dernières années dans le domaine de traitement d'images en général et dans la segmentation en particulier.

Bellet [Bellet, 1998] 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[Boucher, 1999] 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 agents 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.

L. Germond propose une méthode de segmentation coopérative pour la segmentation des images IRM [Germond, 1999]. Une segmentation en régions de la matière grise et de la matière blanche est effectuée par des agents spécialisés à partir d'un modèle déformable du cerveau construit au préalable. A l'issue de la segmentation en régions de ces tissus, le contour du cerveau plus précis est obtenu par des agents contours qui utilisent la méthode des contours actifs. A la fin, un retour complet sur le processus de segmentation est considéré cette fois initialisé par le contour reconstruit lors de la phase précédente.

Dans [Liu, 1999], la segmentation d'image est abordée sous l'angle des automates cellulaire. Il s'agit de générer une population d'agents réactifs et de l'adapter de génération en génération à la distribution de points rencontrés dans l'image. 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 un contour ou à un segment

homogène, 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. Ces idées sont reprises dans [Bureau, 2001], qui propose de combiner une approche par croissance de régions dans les zones faciles à segmenter et l'approche génétique dans les zones plus délicates. La coopération consiste ici à choisir automatiquement, en fonction de critères d'homogénéité locaux, laquelle des deux méthodes l'agent va appliquer.

Dans [Richard, 2001], des agents situés coopèrent pour segmenter des IRM cérébrales. On y trouve diverses catégories d'agents : un agent de contrôle global, des agents de contrôle locaux et au niveau le plus bas, les agents de segmentation, spécialisés dans la détection des trois types de tissus cérébraux (matière blanche, matière grise et liquide céphalorachidien). Un processus de négociation entre régions dans les zones litigieuses est évoqué dans les perspectives, sans toutefois que de solution soit apportée. Duchesnay [Duchesnay, 2001] s'appuie sur la structure de pyramide irrégulière pour gérer le processus de fusion de régions et assurer la convergence de la segmentation; une coopération région-région assez sophistiquée est mise en œuvre, mais qui ne tire pas suffisamment parti de l'information contour. Un des aspects intéressants de son approche est l'utilisation d'une procédure de décimation (récursive) pour le passage du niveau  $k$  au niveau  $k+1$ . La pyramide se construit en partant de la base qui représente l'image pré segmentée (par exemple avec l'algorithme Quadtree jusqu'au dernier niveau de la pyramide comportant le minimum d'information. Les niveaux de cette pyramide sont des graphes d'adjacence de régions. Dans [Veenman, 2003] un système multi\_agent est utilisé pour la segmentation d'images en régions homogènes. Initialement l'image est décomposée en de petites régions de la taille d'un pixel. Les agents sont placés sur une grille 2D représentant l'image à segmenter tel que chacun d'eux est affecté à un pixel. Les agents coopèrent ensemble à travers un processus de migration de pixels d'un agent à un autre dans le but d'améliorer un critère d'homogénéité des régions qu'ils représentent.

D'autres systèmes multi\_agents inspirés du comportement des animaux sociaux ont été proposés dans la littérature. On cite les travaux de Carden [Carden, 2002] qui propose une approche de segmentation d'images basée sur les idées de Reynolds [Reynolds, 1987 ; Reynolds, 2001], pour la détection des contours des régions dans une image simple. 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 [Bourjot, 1999 ; Bourjot, 2001]. 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.

D'autres auteurs s'inspirent du comportement de déplacement collectif des animaux sociaux pour résoudre le problème de classification [Chen, 2004]. Par exemple, dans [Omran, 2002] un algorithme basé sur l'optimisation par les essaims de particules (PSO) est proposé pour la classification d'images. L'algorithme utilise une population de particules chacune de taille  $K$  ( $K$  étant le nombre de classes). L'objectif est de minimiser la distance entre les pixels et les centres de leurs classes et de maximiser la distance entre les centres de tout couple de classes. Chaque particule  $x_i$  est représentée comme suit :

$$x_i = (m_{i,1}, \dots, m_{i,k}, \dots, m_{i,K}) \quad (4.15.)$$

Tel que  $m_{i,k}$  est le vecteur de caractéristiques moyen du  $k^{ieme}$  cluster de la  $i^{ieme}$  particule. La qualité d'une particule est définie par

$$f(x_i, Z_i) = w_1 \bar{d}_{\max}(Z_i, x_i) + w_2 (z_{\max} - d_{\min}(x_i)) \quad (4.16.)$$

Avec

$$\bar{d}_{\max}(Z_i, x_i) = \max_{k=1, \dots, K} \left\{ \sum_{\forall z_p \in c_{ik}} d(z_p, m_{ik}) / n_{ik} \right\} \quad (4.17.)$$

$$d_{\min}(x_i) = \min_{\forall k, l, j \neq 1} \{d(m_{i,k}, m_{i,j})\} \quad (4.18.)$$

Où  $z_{\max}$  est le niveau de gris maximum présent dans l'image ;  $Z_i$  est la matrice de correspondance représentant l'affectation des pixels aux classes de la particule  $i$ . Chaque élément  $z_{i,k,p}$  indique si le pixel  $z_p$  appartient au cluster  $c_k$  de la particule  $i$ .  $w_1$  et  $w_2$  sont des constantes fixées par l'utilisateur.  $n_{ik}$  est le nombre de pixels qui sont affectés au cluster  $c_k$  de la particule  $i$ .

Les résultats obtenus pour la segmentation des images IRM synthétiques et images de satellite sont de bonne qualité moyennant le contrôle des valeurs des paramètres  $w_1$  et  $w_2$ .

Dans d'autres travaux, l'algorithme PSO est hybridé avec un algorithme de classification déterministe tel que le K-means [Wu, 2002 ; Chen, 2005] et les cartes de Kohonen pour améliorer la convergence de l'algorithme [Xiao, 2003].

#### 4.4.4 Conclusion

Depuis quelques années les mécanismes biologiques sont devenues une nouvelle source d'inspiration appelée informatique biomimétique pour les informaticiens dans le domaine de l'imagerie et en particulier pour la résolution du problème de la segmentation d'images. L'approche biomimétique, copie et adapte les comportements concepts mis en œuvre par le monde du vivant pour proposer de nouvelles méthodes de résolution. Ces méthodes ont la particularité d'utiliser un ensemble d'entités simples capables d'accomplir des tâches complexes grâce au concept d'émergence et d'auto-organisation. Les réseaux de neurones, les algorithmes génétiques ou encore les méthodes basées sur les insectes sociaux apportent des solutions originales pour l'obtention d'une segmentation optimale, et peuvent être couplés ensemble pour pallier à leur insuffisances tout en réunissant leurs qualités.

### **4.5 Méthodes d'évaluation des résultats de segmentation en régions**

La mesure de performance est un facteur important dans le fonctionnement d'une méthode de segmentation. Malheureusement, il est difficile d'évaluer la qualité d'une technique de segmentation car ses résultats dépendent de l'objectif visé dès le départ et de la qualité de la décision obtenue en haut niveau [Cocquerez, 1995a]. C'est la raison pour laquelle, l'évaluation et la comparaison de procédures de segmentation restent, dans la majorité des cas, subjectives, fondées sur des considérations d'ordre qualitatif. Or la multiplicité des algorithmes de segmentation d'image, nécessite la définition d'un critère de mesure du degré de confiance que l'on peut avoir dans le résultat fourni par un algorithme de segmentation. Ceci étant, de nombreux chercheurs se sont penchés sur ce problème et ont proposé plusieurs méthodes d'évaluation d'une segmentation [Cocquerez, 1995b ; Zhang, 1996 ; Philipp-Foliguet, 2001].

L'évaluation des résultats de segmentation peut être effectuée par deux approches différentes. Dans la première approche, un résultat de segmentation est évalué en prenant en compte des informations a priori. L'évaluation consiste à mesurer l'écart entre le résultat fourni par un algorithme de segmentation et une segmentation de référence qui peut être fournie par un opérateur humain ou bien lorsqu'une segmentation manuelle peut facilement être réalisée. Ceci revient à effectuer une mesure de dissimilarité entre deux segmentations. La deuxième approche consiste à se référer à l'image originale en évitant de se rapporter à un résultat de référence. Dans le cas de la classification ceci revient à se fixer un modèle de l'erreur de mesure et à estimer la qualité du résultat par rapport à ce modèle. Par contre, dans le cas d'une

segmentation ceci revient à effectuer des mesures d'homogénéité des régions ou de contraste entre régions.

Nous présentons dans la suite de chapitre, les mesures d'évaluation de ces méthodes en présence ou non d'une vérité terrain. Nous détaillons leurs principes, leurs domaines d'application, leurs avantages et leurs limitations. Dans la mesure que la segmentation de contours n'est pas abordée dans cette thèse, nous nous focalisons sur l'évaluation des méthodes de segmentation en régions ou en classes homogènes

#### 4.5.1 Evaluation par comparaison avec une segmentation de référence

La performance des algorithmes de segmentation est évaluée en comparant le résultat fourni par l'algorithme avec une segmentation de référence. L'image de référence peut être une image synthétique ou une image obtenue par des mesures sur le terrain (la vérité de terrain). La comparaison entre deux résultats de segmentation revient à effectuer des mesures de dissimilarité entre ces deux segmentations. Ces mesures prennent en compte dans leur formulation une information sur la taille des régions mal classées et sur leur taux de recouvrement ainsi que l'information spatiale concernant la localisation des pixels qui la constituent [Chabrier, 2004]. Une grande valeur de ces mesures indique une grande erreur de l'algorithme de segmentation et donc une performance faible de l'algorithme.

Dans la suite de ce chapitre, nous utilisons les notations suivantes :

Soit  $I$  une image de  $N$  pixels. Soit  $I_{Seg}$  la segmentation de l'image  $I$  en segmentée en  $C_{Seg}$  régions/classes et  $I_{Ref}$  la segmentation de référence en  $C_{Ref}$  régions/classes que l'on souhaite obtenir. On note  $I_{seg}(i)$  et  $I_{ref}(i)$  l'étiquette de la classe/région du pixel  $p_i$  respectivement dans l'image segmentée et dans l'image de référence.

##### 4.5.1.2 Critères de mesures basées sur la distribution des étiquettes de classe

Différentes mesures sont utilisées dans la littérature pour mesurer la pureté d'une classe ou bien sa compacité en tenant compte d'un étiquetage de référence. Nous présentons deux principales mesures :

- *La Pureté d'une classe* : La pureté d'une classe  $c_k \in I_{Seg}$  est défini comme le pourcentage d'étiquettes prédominantes qu'elle contient en respect des étiquettes de classe  $l \in I_{ref}$ .

$$Pureté(c_k) = \max_l \frac{n_{lk}}{n_k} \quad (4.19.)$$

où  $n_k$  est le nombre de pixels de la classe  $c_k$  et  $n_{lk}$  est le nombre de pixels étiquetés  $l$  dans cette classe. La pureté de la partition  $P(I_{Seg})$  est alors la pureté moyenne de toutes ses classes. Elle prend ses valeurs dans l'intervalle  $[0,1]$  et doit être maximisée.

- *L'entropie d'une classe* L'entropie d'une classe est plus représentative que la mesure de pureté car elle tient compte de la distribution de toutes les étiquettes dans la classe considérée. L'entropie d'une classe est donnée par :

$$Entropie(c_k) = \frac{1}{\text{Log}(N)} \sum_l \frac{n_{lk}}{n_k} \log\left(\frac{n_{lk}}{n_k}\right) \quad (4.20.)$$

$N$  étant le nombre total de pixel de l'image  $I$ . L'entropie d'une classification est la moyenne des entropies de toutes les classes de la partition. La mesure de l'entropie d'une classe prend ses valeurs dans l'intervalle  $[0,1]$  et doit être minimisée.

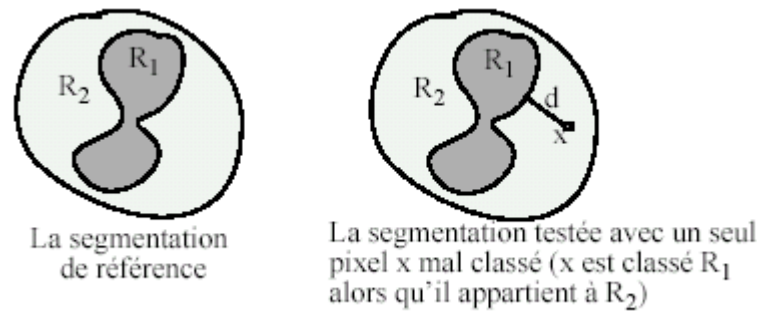
#### 4.5.1.2 Critères de mesures d'erreurs de classification

Les mesures d'écart entre une segmentation obtenue par un algorithme de segmentation et une segmentation de référence ont été fréquemment utilisées pour évaluer les performances des algorithmes de segmentation et de classification. On distinguera principalement entre deux catégories de mesures, celles qui sont plus sensibles aux erreurs de détection (ce qui veut dire ici que le pixel est soit bien classé soit mal classé), et celles qui sont plutôt sensibles aux erreurs de localisation (la localisation spatiale des pixels mal classés est prise en considération).

Yasnoff et al [Yasnoff, 1977] ont proposé des mesures d'évaluation qui quantifient les erreurs de classification quand les classes des pixels sont connues. La mesure de Yasnoff permet d'évaluer le taux d'erreur pour un pixel mal classé  $p$  proportionnel à la distance entre ce pixel et le plus proche pixel appartenant à la classe à laquelle  $p$  aurait dû être affecté. L'erreur de segmentation entre un résultat de segmentation  $I_{Seg}$  et sa référence  $I_{Ref}$  est alors définie comme suit :

$$YAS(I_{Seg}, I_{Ref}) = \frac{100}{N} \times \sqrt{\sum_{p \in I, p \notin R_p} \min_{p' \in R_p} d(p, p')} \quad (4.21.)$$

Où  $R_p \in I_{Ref}$  correspond à la région à laquelle devrait appartenir le pixel  $p$  et  $d(p, p')$  correspond à la distance entre un pixel  $p$  n'appartenant pas à  $R_p$  et le pixel  $p'$  le plus proche de  $R_p \in I_{Ref}$ .



**Figure 4.10.** La distance entre un pixel  $x$  et le plus proche pixel de  $R_1$

Une autre mesure pour la détection de similarité entre deux segmentation est celle de Vinet [Vinet, 1991]. Elle consiste à rechercher les couples de régions  $(R_i, R_j)$  les plus similaires dans les deux segmentation telle que  $R_i$  appartient à la segmentation résultat  $I_{Seg}$  et  $R_j$  à la segmentation de référence  $I_{Ref}$ . Pour cela une matrice de recouvrement  $T(I_{Seg}, I_{Ref})$  est construite tel que chaque élément  $t_{ij} = \text{card}(R_i \cap R_j)$ ,  $i=1 \dots C_{Seg}$  et  $j=1 \dots C_{Ref}$  mesure le taux de recouvrement entre deux classes  $R_i, R_j$ . Il s'agit alors de rechercher d'une manière récursive les couples de régions  $(R_i, R_j)$  ayant un recouvrement maximal. La mesure de Vinet est définie comme suit :

$$Vinet(I_{Seg}, I_{Ref}) = N - \sum_K \text{card}(R_i, R_j) \quad (4.22.)$$

tel que  $K$  est le nombre de couple de classes qui maximisent  $\text{card}(R_i \cap R_j)$ .

La mesure de Vinet est proportionnelle au nombre de pixels qui n'appartiennent pas au recouvrement entre deux classes. Cette mesure n'exige pas que les deux segmentation aient le même de classes et ne fait intervenir aucune information a priori sur la correspondance entre les régions. Cependant lors de l'évaluation, l'information spatiale sur la localisation des pixels n'intervenant pas dans la superposition n'est pas considérée. Cet inconvénient est résolu par la mesure de Kara-falah [Kara-Falah, 1995].

D'autres auteurs proposent d'autres types de critères qui s'intéressent en particulier à la bonne localisation de la frontière des classes obtenues. Pour une classe  $c_k$ , ces critères tiennent compte de quatre catégories de pixels (figure 4.11) :

- Les *Vrai Positifs* (VP) : nombre de pixels détectés à raison

$$VP = \left| \left\{ i \mid I_{ref}(i) = c_k \wedge I_{seg}(i) = c_k \right\} \right| \quad (4.23.)$$

- Les Faux Positifs (FP) : nombre de pixels détectés à tort

$$FP = \left| \left\{ i \mid I_{ref}(i) \neq c_k \wedge I_{seg}(i) = c_k \right\} \right| \quad (4.24.)$$

- Les Vrai Négatifs (VN) : nombre de pixels non détectés à raison

$$VN = \left| \left\{ i \mid I_{ref}(i) \neq c_k \wedge I_{seg}(i) \neq c_k \right\} \right| \quad (4.25.)$$

- Les Faux Négatifs (FN) : nombre de pixels non détectés à tort

$$FN = \left| \left\{ i \mid I_{ref}(i) = c_k \wedge I_{seg}(i) \neq c_k \right\} \right| \quad (4.26.)$$

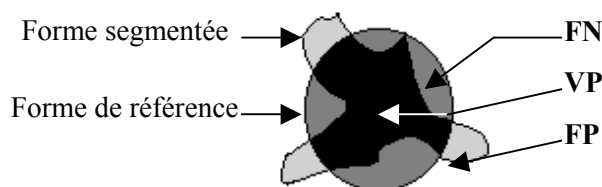


Figure 4.11. types de pixels classés

Sur la base de ces quatre catégories de pixels, Shufelt propose deux mesures pour évaluer la qualité de détection de bâtiments [Shufelt, 1999].

**Le pourcentage de bâtiments détectés ou Building detection percentage (BDP) :** il permet de quantifier le pourcentage de pixels qui ont été correctement classifiés par la méthode de segmentation. Dans le cas idéal, ce coefficient vaut 1. Il est calculé par la formule suivante :

$$BDP = \frac{VP}{VP+FN} \quad (4.27.)$$

**Le facteur d'embranchement ou Branching Factor (BF) :** ce coefficient quantifie la surdéréction de pixels n'appartenant pas à la bonne classe dans la segmentation de référence. Dans le cas idéal ce coefficient vaut 0. On le mesure par :

$$BF = \frac{FN}{VP} \quad (4.28.)$$

Enfin, d'autres classes de méthodes d'évaluation évaluent déterminent le taux de pixels bien classés en calculant la différence entre la segmentation de référence et celle obtenue par un algorithme de segmentation. pour cela elles comparent toutes les paires de pixels et vérifient à

chaque fois si ces pixels sont classés simultanément dans les deux segmentations.. L'index de Rand  $R$  est le plus généralement utilisé dans la littérature et est défini comme suit :

$$R = \frac{a+d}{a+b+c+d} \quad (4.29.)$$

avec  $a$ ,  $b$ ,  $c$  et  $d$  des paramètres calculés pour tous les couples de pixels  $p_i$  et  $p_j$  de la façon suivante

$$a = \left| \{i, j \mid I_{ref}(i) = I_{ref}(j) \wedge I_{seg}(i) = I_{seg}(j)\} \right| \quad b = \left| \{i, j \mid I_{ref}(i) = I_{ref}(j) \wedge I_{seg}(i) \neq I_{seg}(j)\} \right|$$

$$c = \left| \{i, j \mid I_{ref}(i) \neq I_{ref}(j) \wedge I_{seg}(i) = I_{seg}(j)\} \right| \quad d = \left| \{i, j \mid I_{ref}(i) \neq I_{ref}(j) \wedge I_{seg}(i) \neq I_{seg}(j)\} \right|$$

L'index de Rand prend ses valeurs dans l'intervalle  $[0,1]$ . La valeur élevée de  $R$  (tend vers 1) indique une parfaite correspondance entre la segmentation de référence et celle obtenue par un algorithme de segmentation.

Différentes variations de l'index de  $R$  existent dans la littérature. Parmi elles on cite le coefficient de Jaccard qui applique une stricte correspondance pour chaque classe entre les deux segmentations et est définit comme suit

$$J^k(I_{seg}, I_{ref}) = \frac{\left| V_{I_{seg}}^k \cap V_{I_{ref}}^k \right|}{\left| V_{I_{seg}}^k \cup V_{I_{ref}}^k \right|} \quad (4.30.)$$

où Soit  $V_g^k$  et  $V_s^k$  qui représentent respectivement le nombre total de pixels appartenant à la classe  $c_k$  dans les deux images  $I_{seg}$  et  $I_{ref}$ . Une bonne segmentation est obtenue quand  $J^k(I_{seg}, I_{ref})$  est proche de 1 ce qui signifie que la classe  $c_k$  a été bien extraite

Ce type de mesure présente l'inconvénient de ne compter que les pixels, sans prendre en considération les propriétés spatiales inhérentes aux pixels mal classés. De plus aucune information sur les pixels responsables de l'erreur de classification n'est fournie.

#### 4.5.2 L'évaluation se référant à l'image originale

Les méthodes présentées dans cette section permettent une évaluation non supervisée des résultats de segmentation, dans le cas où aucune vérité terrain ne serait disponible. Ces méthodes tiennent compte des points suivants: l'homogénéité intra-région, le contraste inter-régions, la forme et la taille des régions ou des classes.

#### 4.5.2.1 Critère d'uniformité intra-région

En pratique une bonne segmentation doit fournir des régions les plus homogènes possibles ou présentant de faibles fluctuations de niveau de gris. Dans [Levine, 1985] Levine et Nazif propose un critère qui mesure l'uniformité du niveau de gris d'une partie de l'image segmentée en se basant sur la variance de ce niveau de gris. Ce critère est défini comme suit :

$$Levine(I) = 1 - \frac{\sum_{k=1}^{NR} \left[ \sum_{p_i \in R_k} g_I(p_i) - \sum_{p_j \in R_k} g_I(p_j) \right]^2}{Card(I) \times \left( \max_{p_i \in R_K} (g_I(p_i)) - \min_{p_i \in R_K} (g_I(p_i)) \right)^2} \quad (4.31.)$$

où

- $I_{seg}$  l'image segmentée en un ensemble de  $R$  de  $NR$  régions  $\{R_1, \dots, R_{NR}\}$ ,
- $Card(I)$  correspond au nombre de pixels de l'image  $I$ ,
- $g_I(i)$  correspond au niveau de gris du pixel  $p_i$  de l'image  $I$ .

Plusieurs variations de cette mesure existent dans la littérature. On cite la mesure de Sahoo [Sahoo, 1988] défini comme suit :

$$Sahoo(I_{seg}) = 1 - \frac{Levine2(I_{seg})}{C} \quad (4.32.)$$

Avec

$$Levine2(I_{seg}) = \sum_{k=1}^{NR} \sum_{p_i \in R_k} \left[ g_I(p_i) - \frac{1}{Card(R_k)} \sum_{p_i \in R_k} g_I(p_i) \right]^2 \quad (4.33.)$$

où  $Card(R_k)$  correspond au nombre de pixels de la région  $R_k$

#### 4.5.2.2 Critère de contraste inter-régions

Ce type de mesures est complémentaire au type précédent. Il indique que dans bonne segmentation deux régions adjacentes doivent avoir des niveaux de gris différents. Levine et Nazif [Levine, 1988] ont formulé cette nécessité par une mesure de contraste entre tout couple de régions adjacentes  $R_i$  et  $R_j$  comme suit :



$$\text{Contraste}(R_i, R_j) = \frac{|\bar{g}_I(R_i) - \bar{g}_I(R_j)|}{|\bar{g}_I(R_i) + \bar{g}_I(R_j)|} \quad (4.34.)$$

De cette manière, le contraste d'une région par rapport à toutes les régions qui lui sont adjacentes se calcule comme suit :

$$\text{Contraste}(R_i) = \sum_{R_j \in W(R_i)} P_{R_i \setminus R_j} \text{Contraste}(R_i, R_j) \quad (4.35.)$$

où  $W(R_i)$  est le nombre de régions adjacentes à la région  $R_i$  et  $P_{R_i \setminus R_j}$  est égal au rapport des pixels situés sur la frontière des deux régions  $R_i$  et  $R_j$  sur le périmètre de la région  $R_i$ .

#### 4.5.2.3 Critère inter-intra régions

Afin de bénéficier de chacune des particularités que possède chacun des deux types de critères définis précédemment à savoir le critère intra-régions et celui inter-régions, des chercheurs ont proposé de les combiner en un seul critère qui évalue qu'une segmentation est bonne si les régions/classes obtenues sont homogènes et bien séparées. On cite le critère de Rosenberg Rosenberg [Rosenberger, 1999] qui permet d'estimer l'homogénéité des régions / classes d'un résultat de segmentation de manière à maximiser l'homogénéité intra-classe et le contraste inter-classes. Il repose sur deux des quatre critères heuristiques suggérés par Haralick et Shapiro [Haralick, 1985] pour évaluer les résultats de segmentation :

- les régions doivent être homogènes,
- les régions adjacentes doivent présenter des valeurs significativement différentes pour les caractéristiques utilisées.

Le critère de Rosenberg est donné par la formule suivante :

$$\text{Rosenberg}(I_{seg}) = \frac{\text{Disparité}_{intra}(I_{seg}) + 1 - \text{Disparité}_{inter}(I_{seg})}{2} \quad (4.36.)$$

La disparité intra-classe globale  $\text{Disparité}_{intra}(I_{seg})$  quantifie l'homogénéité de chacune des classes de l'image  $I_{seg}$ . Elle est calculée à partir de la disparité intra-classe  $\text{Disparité}_{intra}(R_i)$  des différentes régions/classes  $R_i$   $i = 1 \dots NR$  par la formule suivante :

$$\text{Disparité}_{intra}(I_{seg}) = \frac{1}{NR} \sum_{k=1}^{NR} \frac{\text{Card}(R_k)}{\text{Card}(I)} \text{Disparité}_{intra}(R_k) \quad (4.37.)$$

et

L'importance de la disparité intra-classe d'une région dans le calcul de la disparité globale est proportionnelle au nombre de pixels de cette région. Il est en effet souhaitable qu'une région de petite taille ait une influence moindre dans le calcul de la disparité intra-classe globale.

Suivant le même principe, la disparité inter-classes globale  $Disparité_{inter}(I_{seg})$  mesure la disparité globale de chacune des régions de l'image  $I_{seg}$ . La fonction de critère est définie comme suit :

$$Disparité_{inter}(I_{seg}) = \frac{1}{NR} \sum_{k=1}^{NR} \frac{Card(R_k)}{Card(I)} Disparité_{inter}(R_k) \quad (4.38.)$$

Dans le calcul des disparités intra et inter-classes, la nature des classes segmentées est prise en compte à savoir uniforme et texturée. Pour le cas d'une classe uniforme la disparité intra-classe est donnée par la formule suivante :

$$Disparité_{intra}(R_i) = \frac{2}{255} \sqrt{\frac{1}{Card(R_i)} \sum_{p_i \in R_i} g_i^2(p_i) - \frac{1}{Card(R_i)^2} \left( \sum_{p_i \in R_i} g_i(p_i) \right)^2} \quad (4.39.)$$

La disparité inter-classes d'une classe se base sur sa disparité avec ses voisines et se calcule comme suit :

$$Disparité_{inter}(R_i) = \frac{1}{q^{(i)}} \sum_{j=1}^{q^{(i)}} Disparité_{inter}(R_i, R_j) \quad (4.40.)$$

avec  $q^{(i)}$  le nombre de régions  $R_j$  voisine à  $R_i$ . La disparité entre deux régions uniformes se calcule de la manière suivante :

$$Disparité_{inter}(R_i, R_j) = \frac{|\bar{g}_I(R_i) - \bar{g}_I(R_j)|}{NG} \quad (4.41.)$$

Où  $NG$  est le nombre de niveaux de gris dans l'image. Cette mesure calcule la différence normalisée des niveaux de gris de deux régions uniformes.

#### 4.5.2.3 Critères hybrides

Les critères décrits dans les paragraphes précédents se focalisent sur l'aspect d'uniformité des régions obtenues par un algorithme de segmentation. Il existe dans la littérature d'autres critères qui évaluent une méthode de segmentation en tenant compte de la forme, de la taille et de l'homogénéité des régions.

Parmi ces critères, on cite la mesure d'évaluation proposée par Liu et Yang [Liu, 1994] qui incorpore les propriétés suivantes :

les régions doivent être uniformes et homogènes,

l'intérieur des régions doit être simple et sans trop de petits trous,

les régions adjacentes doivent présenter des valeurs significativement différentes pour les caractéristiques d'uniformité.

Leur mesure d'évaluation est empiriquement défini comme suit :

$$Liu(I_{seg}) = \frac{1}{1000(Card(I))} \sqrt{NR} \sum_{i=1}^{NR} \frac{e_i^2}{\sqrt{Card(R_i)}} \quad (4.42.)$$

où  $e_i$  correspond à l'erreur portant sur le niveau de gris moyen de la région  $R_i$ , défini 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 premier terme de l'équation est un facteur de normalisation qui tient compte de la taille de l'image. Le second terme pénalise les résultats avec plusieurs petites régions (sur-segmentation). Le troisième terme pénalise les résultats de segmentation contenant des régions non homogènes. Ce dernier terme est mesuré par le facteur d'aire parce que l'erreur de niveau de gris est plus élevée pour de grandes régions. Une bonne segmentation  $I_{seg}$  minimisera la mesure  $Liu(I_{seg})$ .

En étudiant de plus près la fonction de Liu et Yang, Borsotti et al [Borsotti, 1998] ont identifié deS limitations de cette fonction. En effet, dans le cas où l'image segmentée  $I_{seg}$  présente plusieurs petites régions (sur-segmentation), le nombre de régions serait très élevé. Cependant, l'erreur de niveau de gris de chaque région peut être proche de zéro et  $Liu(I_{seg})$  sera aussi proche de zéro, ce qui signifie incorrectement que le résultat de la segmentation est très bon. Le meilleur exemple de cette situation est de considérer chaque pixel comme une région. Borsotti et al [Borsotti, 1998] ont donc modifié la mesure d'évaluation de Liu et Yang de manière 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. la mesure de Borsotti et al est définie comme suit :

$$Borssoti(I_{seg}) = \frac{1}{10000(Card(I_{seg}))} \sqrt{NR} \sum_{i=1}^R \left[ \frac{e_i^2}{1+\log(Card(R_i))} + \left( \frac{f(Card(R_i))}{Card(R_i)} \right)^2 \right] \quad (4.43.)$$

où  $f(Card(R_i))$  correspond au nombre de régions ayant une aire égale à  $Card(R_i)$

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.

## 4.6 Conclusion

Dans ce chapitre, nous avons présenté les méthodes de segmentation d'images les plus connues ainsi que les mesures d'évaluation des résultats de segmentation. L'ensemble des techniques décrites dans les sections précédentes montre que les méthodes de segmentation bio-inspirée et en particulier celles basées sur les techniques d'intelligence en essaim constitue une voie de recherche très intéressante et mérite une étude approfondie en vue de résoudre des problèmes mal posés dans le domaine de la vision artificielle en général et celui de la segmentation d'images en particulier. Nous rappelons les principales remarques tirées des développements précédents.

- Les méthodes basées sur la détection de contours permettent l'extraction des points de brusque variation d'intensités et représentant les transitions locales entre régions homogènes dans l'image. Cependant la qualité de la segmentation obtenue dépend fortement de la présence du bruit dans l'image. Les détecteurs optimaux tendent à trouver le meilleur compromis entre une bonne détection et une bonne localisation des contours mais nécessitent l'utilisation de seuils qui peuvent être gênant dans le cas d'une segmentation non supervisée. D'un autre côté, les algorithmes de détecteurs de contours produisent des contours ouverts difficiles à fermer dans certaines configurations, ce qui rend leur exploitation difficile.
- Les méthodes basées sur la croissance de régions considèrent la propriété de la topologie de l'image en plus des caractéristiques photométriques des pixels lors de l'extraction des régions homogènes. Ces méthodes sont faciles à utiliser mais dont les résultats sont fortement dépendant de l'initialisation et de l'ordre d'agrégation de pixels ou bien de l'ordre de fusion de régions pour les méthodes de fusion. Certaines méthodes nécessitent le réglage de nombreux seuils qui sont difficiles à fixer en l'absence de connaissance a priori. Les méthodes de division génèrent un effet de sur-segmentation en introduisant des contours pas forcément présents dans l'image initiale. Les méthodes de division / fusion sont moins sensibles au bruit et réduisent le temps de croissance de régions.

- Les méthodes de classification classiques ne prennent en compte que la distribution de niveau de gris des pixels de l'image sans la cohérence spatiale. En effet, après la construction des classes les pixels sont assignés individuellement à l'une des classes sans tenir compte de leurs interactions spatiales. Les régions ne sont construites que par une analyse de connexité des pixels assignés aux différentes classes. Les méthodes de classification par relaxation permettent d'incorporer dans le processus de classification des informations a priori et un terme de régularisation qui tient compte de la présence du bruit dans l'image.
- Les méthodes de segmentation basées sur les systèmes multi\_agents permettent grâce aux interactions entre des entités complètement autonomes d'aboutir à une solution optimale.
- Les résultats obtenus à partir d'un algorithme de segmentation peuvent être évalués quantitativement selon deux approches. 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. En d'autres termes, il serait souhaitable, si l'on segmente à vue les images de l'échantillon, que chaque observateur fournisse un résultat identique.

# Chapitre 5

## Des fourmis pour la classification automatique des images

<b>Chapitre 5. Des fourmis pour la classification automatique des images.....</b>	<b>119</b>
5.1 Introduction .....	119
5.2 L'algorithme AntClust .....	120
5.2.2 L'environnement des fourmis .....	120
5.2.3 Les fourmis et leurs comportements .....	121
5.3 Etude expérimentale .....	126
5.3.1 Les images de test .....	126
5.3.3 Les mesures d'évaluation .....	127
5.3.4 Comparaison des résultats .....	128
5.3.5 Les résultats .....	128
5.4 Conclusion.....	132

---

## Chapitre 5. Des fourmis pour la classification automatique des images

---

### 5.1 Introduction

Les méthodes de segmentation par classification de pixels affectent chaque pixel à une classe, en fonction d'un ou plusieurs attributs de ce pixel. La classification est dite supervisée lorsque des informations a priori sont utilisées pour la construction de classes sous la forme d'un ensemble d'apprentissage. Dans le cas où aucune connaissance a priori n'est disponible, on parle de classification non supervisée. Dans ce type de méthodes, on trouve les méthodes hiérarchiques qui consistent une suite de partitions emboîtées et les méthodes par partitionnement qui fournissent une seule partition. De nombreux algorithmes de partitionnement déterministes existent dans la littérature tels que le K-means et le Fcm et leurs variantes. Ces algorithmes sont très simples à implémenter et convergent rapidement avec une solution localement optimale. Cependant leur majeur inconvénients est qu'ils nécessitent de fournir en entrée une partition initiale de bonne qualité ainsi que le nombre possible de classes. Ces contraintes rendent l'utilisation de ces algorithmes peu intéressante quand on veut segmenter automatiquement une image.

Dans le chapitre 3, nous avons montré que les fourmis arrivent à résoudre naturellement et d'une manière distribuée des problèmes de classification tels que l'organisation du couvain et le ramassage des cadavres. Ces comportements ont inspiré les informaticiens et ont permis l'introduction de nouvelles heuristiques pour le problème de la classification. Les premiers travaux dans ce domaine ont été ceux de Deneubourg et son équipe [Deneubourg, 1990], se basant sur une colonie de fourmis artificielles qui se déplacent aléatoirement sur une grille rectangulaire et sont capables de ramasser et de déposer des objets présents sur la grille dans le but de les regrouper selon un critère de similarité. Ces travaux ont été par la suite améliorés et étendus à différents domaines d'application comme nous avons constaté dans le chapitre 3.

En s'inspirant du comportement collectif d'agrégation/ségrégation des fourmis envers le couvain, et en se basant sur les travaux modélisant ce genre de comportement, nous proposons AntClust un nouvel algorithme de partitionnement non supervisé [Ouaïfel, 2005a]. AntClust utilise les principes d'exploration stochastique et distribuée d'une population de fourmis artificielles pour fournir une segmentation d'une image en des classes pertinentes sans disposer d'une partition de départ et sans connaître le nombre de classes qui seront nécessaires. Selon le phénomène de stigmergie la partition optimale émerge à partir de l'activité collective de l'ensemble des fourmis et des interactions locales entre les fourmis et leur environnement.

Dans la suite de ce chapitre, nous présentons tout d'abord les motivations qui nous amenés à proposer AntClust ainsi que les améliorations que nous avons introduites par rapport aux algorithmes de classification basés fourmis existants dans la littérature. A la suite de ça, nous décrivons en détail les différentes étapes de l'algorithme ainsi qu'une étude expérimentale effectuée sur des images synthétique afin de montrer l'efficacité de la méthode.

## **5.2 L'algorithme AntClust**

AntClust est une transposition du modèle comportemental de tri de couvain observé chez les fourmis réelles pour la résolution collective du problème de classification des images à niveau de gris. Lors du développement de l'algorithme AntClust, nous avons repris les travaux existants concernant le problème de classification non supervisé [Deneubourg, 1990 ; Lumer, 1994] et [Monmarché, 1999] pour y introduire un certain nombre de modification susceptibles d'améliorer les performances de l'algorithme en terme de qualité de classification et d'en accélérer la convergence vers un état stable. Dans la suite de cette section, nous détaillons les éléments de base de l'algorithme AntClust en terme d'environnement, de fourmis et leurs comportements.

### **5.2.2 L'environnement des fourmis**

Dans les travaux précédents utilisant les fourmis artificielles pour résoudre le problème de partitionnement, les objets à classer sont placés aléatoirement sur une grille à deux dimensions et les fourmis se déplacent sur la grille d'une case à une autre et utilisent une mesure de similarité locale pour regrouper des objets de même nature. Dans AntClust, nous avons vu que ce support ne semble pas adapté pour la recherche de classes dans une image et cela pour plusieurs raisons. Nous citons en particulier :

- Le nombre de pixels à classer est très important et nécessite une grille de grande taille afin de pouvoir trouver des cas vides pour placer les pixels déplacés par les fourmis ;
- Une taille de grande taille va rendre la convergence de l'algorithme extrêmement lente car les fourmis vont perdre du temps à chercher les pixels ;
- Le déplacement des fourmis sur la grille étant aléatoire, certaines cases peuvent ne pas être visitées par les fourmis et donc les pixels qui y sont placés ne seront pas ramassés en un nombre d'itérations acceptables.
- Le fait que chaque case de la grille ne puisse contenir qu'un seul objet à la fois, implique qu'une fourmi peut passer un certain temps à trouver une case libre sur la grille.



Pour ces différentes raisons, nous avons abandonné la grille car plusieurs paramètres s'y rattachent et il n'est pas facile de trouver le paramétrage adéquat.

Dans AntClust, l'environnement des fourmis est un tableau de  $N$  cellules reliées chacune à un emplacement représentant le nid de la colonie, afin de faciliter les déplacements des fourmis d'une cellule à une autre (Figure 5.1). Initialement les  $N$  pixels de l'image  $I$  à segmenter sont placés sur le tableau de telle sorte qu'une cellule ne contienne qu'un pixel à la fois. Durant le processus de classification une cellule peut correspondre à un ou plusieurs pixels de l'image d'origine. A la fin de l'algorithme le nombre de cellules non vides représente le nombre possible de classes présentes dans l'image.

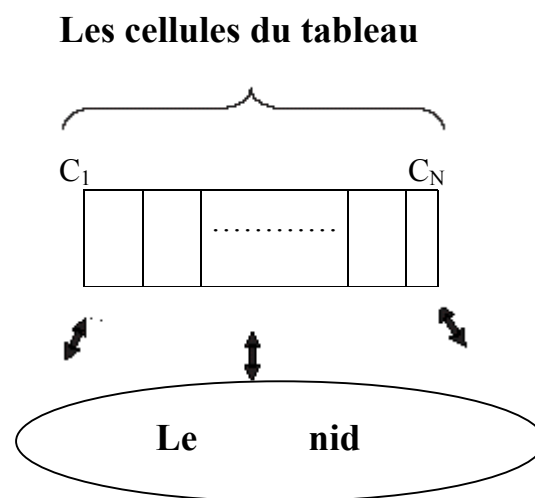


Figure 6.1.

Figure 5.1. L'environnement des fourmis artificielles

### 5.2.3 Les fourmis et leurs comportements

AntClust utilise une population  $A$  de  $K$  fourmis  $\{a_1, \dots, a_K\}$  qui coopèrent ensemble et communiquent par stigmergie pour fournir une classification optimale d'une image de  $N$  pixels. Les fourmis virtuelles déplacent les pixels de leurs cellules en fonction d'un critère de similarité entre un pixel donné et les pixels présents dans sa cellule pour former des classes homogènes. L'algorithme (voir l'algorithme 5.1.) commence avec une phase initiale dans laquelle (1) les  $N$  pixels sont placés aléatoirement sur les  $N$  cellules du tableau en vérifiant qu'une cellule ne peut contenir qu'un seul pixel à la fois ; (2) chaque fourmi  $a_i \in A$  choisit aléatoirement un pixel et retourne vers son nid. A la suite de cette étape, un processus de classification itératif commence : c'est un cycle simple exécuté pour chaque fourmi  $a_i \in A$  sélectionnée aléatoirement. Comme dans les précédents travaux, les items comportementaux d'une fourmi sont le déplacement d'un pixel, le dépôt d'un pixel et, en plus, un comportement de retour vers le nid pour modéliser le déplacement des fourmis d'une cellule du tableau à une autre. La fourmi porteuse évalue l'adéquation en terme de niveau de

**Algorithme 5.1. Algorithme AntClust**

*/\* Initialisation\*/*

*Pour chaque pixel  $p_i$  faire*

*Placer  $p_i$  dans une cellule du tableau*

*Finpour*

*Pour chaque fourmi  $a_1$  faire*

*Placer  $a_1$  dans une cellule choisie aléatoirement et lui affecter son pixel;*

*Etat [ $a_1$ ] := porteuse ;*

*Finpour*

*Déplacer toutes les fourmis vers le nid*

*/\* Boucle principale\*/*

*Pour  $t=1$  à  $t_{max}$  faire*

*Pour chaque fourmi  $a_1$  faire*

*Si état [ $a_1$ ] = porteuse alors*

*Déplacer  $a_1$  vers une cellule  $c_k$*

*Dépôt := faux ;*

*Dépôt := déposer le pixel  $p_i$  qu'elle transporte dans  $c_k$  avec une probabilité*

$$p_{\text{dépôt}}(p_i, c_k)$$

*Si Dépôt = vrai alors*

*Etat [ $a_1$ ] := libre ;*

*Finsi*

***Sinon***

*Choisir aléatoirement un pixel  $p_i$  ;*

*Déplacer  $a_1$  vers la cellule  $c_k$  contenant  $p_i$  ;*

*Porter := faux ;*

*Porter := Porter  $p_i$  de sa cellule avec une probabilité  $p_{\text{porter}}(p_i, c_k)$  ;*

*Si Porter = vrai alors*

*Etat [ $a_1$ ] := porteuse ;*

*Finsi*

*Déplacer les fourmis vers le nid*

*Finpour*

***Retourner la partition obtenue***

gris qu'il y a entre le pixel qu'elle porte et les pixels contenus dans la cellule du tableau où elle voudrait le déposer à l'aide d'un critère de similarité. Plus ce critère de dissimilarité est important, plus la probabilité de le déposer sera importante. Le déplacement d'un pixel obéit à un raisonnement inverse. La fourmi évalue le critère de similarité entre le pixel considéré et les pixels présents dans sa cellule. Plus ce critère est faible, plus la fourmi aura de la chance de le déplacer de sa cellule. De cette manière, les fourmis génèrent dynamiquement des classes de niveau de gris homogènes à travers une coopération inconsciente.

Il convient maintenant d'explicitier un critère de similarité entre un pixel et d'autres pixels d'une cellule du tableau. Pour cela, nous avons besoin d'évaluer une mesure de distance entre deux pixels  $p_i$  et  $p_j$ . C'est un élément important dans tout algorithme de partitionnement.

Dans notre travail, nous avons adopté la distance euclidienne des niveaux de gris des deux pixels comprise entre 0 et 1. Nous avons alors :

$$d(p_i, p_j) = \frac{ng_i - ng_j}{255} \quad (5.1)$$

Le critère d'adéquation entre un pixel  $p_i$  et les pixels d'une cellule  $c_k$  contenant  $n_k$  pixels est alors défini par la fonction de similarité suivante :

$$f(p_i, c_k) = \frac{1}{n_k} \sum_{p_j \in c_k} \frac{\alpha^2}{\alpha^2 + d(p_i, p_j)^2} \quad (5.2)$$

$$\text{avec } \alpha = \frac{1}{N(N-1)} \sum_{p_i \in I} \sum_{p_j \in I} d(p_i, p_j) \quad (5.3)$$

Où  $ng_i$  et  $ng_j$  sont les niveaux de gris respectifs des pixels  $p_i$  et  $p_j$ .  $\alpha$  est une constante représentant la distance moyenne entre deux pixels de l'image  $I$ .

La fonction de similarité  $f(\cdot)$  atteint son maximum quand le niveau de gris du pixel  $p_i$  est très proche de celui des pixels présents dans la cellule  $c_k$ .

Dans la suite, nous présentons les règles que les fourmis vont utiliser sur le tableau pour déplacer et déposer les pixels.

#### 5.2.3.1 Dépôt du pixel

Une fourmi porteuse effectue un certain nombre de déplacements entre son nid et le tableau afin d'explorer les cellules non vides et choisir d'une manière probabiliste la cellule où elle

déposera le pixel qu'elle transporte. La probabilité de déposer un pixel  $p_i$  dans une cellule  $c_k$  est donnée par la formule suivante :

$$p_{\text{dépôt}}(p_i, c_k) = 1 - \cos^2\left(\frac{\pi}{2} f(p_i, c_k)\right) \quad (5.4.)$$

Ainsi, plus la fonction de similarité entre le pixel  $p_i$  et les éléments de  $c_k$  est faible (tend vers 0), plus la probabilité de dépôt sera faible. Dans le cas contraire si la fonction de similarité est importante ( $f(p_i, c_k) \rightarrow 1$ ), alors la fourmi a une grande chance de déposer le pixel dans la cellule ( $p_{\text{dépôt}}(p_i, c_k) \rightarrow 1$ ).

### 5.2.3.2 Ramassage d'un pixel

Dès qu'une fourmi devient libre, elle recherche automatiquement le prochain pixel à transporter. Afin d'éviter les déplacements infructueux des fourmis comme dans les précédents travaux et accélérer la convergence de l'algorithme, cette recherche n'est pas aléatoire. Elle est guidée en utilisant un index commun contenant les pixels libres (non transportés par d'autres fourmis) ainsi que la position de leurs cellules dans le tableau. Initialement, l'index contient tous les pixels de l'image.

La fourmi choisit aléatoirement un pixel  $p_i$  de l'index et se dirige directement de son nid vers la position de sa cellule  $c_k$  dans le tableau. Elle évalue localement la fonction de similarité  $f(p_i, c_k)$  et décide avec une certaine probabilité de ramasser le pixel considéré de sa cellule. Que cette décision soit négative ou positive, la fourmi revient vers son nid pour terminer son cycle.

La probabilité de transporter un pixel  $p_i$  de sa cellule  $c_k$  est définie par la formule suivante :

$$p_{\text{porter}}(p_i, c_k) = \begin{cases} 1 & \text{si } |c_k|=1 \\ q & \text{si } |c_k|=2 \\ \cos^2\left(\frac{\pi}{2} f(p_i, c_k)\right) & \text{sinon} \end{cases} \quad (5.5.)$$

où  $|c_k|$  est le nombre de pixels dans la cellule  $c_k$ . Ainsi si la cellule  $c_k$  du pixel sélectionné ne contient que lui, il est systématiquement ramassé par la fourmi. Si la cellule contient deux pixels, la fourmi a une probabilité  $q$  de ramasser le pixel  $p_i$ . Enfin si la cellule contient plus de deux pixels la probabilité  $p_{\text{porter}}$  de transporter le pixel  $p_i$  est importante quand la fonction de similarité entre le pixel  $p_i$  et les pixels présents dans sa cellule est faible (tend vers 0). Dans

le cas contraire,  $f(p_i, c_k) \rightarrow 1$ ,  $p_{porter}(p_i, c_k) \rightarrow 0$  le pixel a une grande chance de rester dans sa cellule.

### 5.2.3.3 Mémoire des fourmis

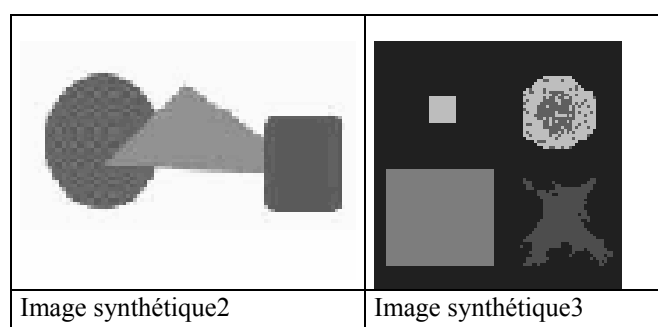
Durant le processus de classification, les fourmis vont se déplacer régulièrement entre leur nid et les cellules du tableau pour transporter ou bien déposer un pixel. Afin d'accélérer le processus de regroupement et donc la convergence de l'algorithme, la fourmi n'a pas un mouvement complètement désordonné. Pour cela une version modifiée du mécanisme de mémoire à court terme introduit dans [Lumer 1994] et [Monmarché 1999] est proposée. Dans l'approche de Lumer et Faieta, chaque fourmi mémorise les  $m$  derniers objets qu'elle a ramassés ainsi que leurs emplacements sur la grille. A chaque fois qu'elle ramasse un nouvel objet, il est comparé aux objets contenus dans sa mémoire afin de biaiser la direction qu'elle va prendre. La fourmi a tendance à se diriger vers l'emplacement où elle a déposé auparavant un objet similaire à celui qu'elle transporte actuellement) Ce mécanisme a été étendu dans [Handl, 2003] en remplaçant la distance euclidienne entre deux objets par la fonction de voisinage appliquée aux positions actuelles de tous les objets contenus dans la mémoire de la fourmi. Monmarché reprend les idées de Lumer et Faieta et remplace la comparaison des objets sur la distance les séparant par la distance entre le centre de gravité du tas transporté par la fourmi et les tas qu'elle a mémorisés( puisque dans son approche, les fourmis peuvent transporter plus qu'un objet à la fois) pour choisir le prochain emplacement de l'objet (ou du tas) qu'elle transporte. Dans les précédents travaux, la mémoire de la fourmi est statique, ce qui implique qu'elle ne reflète pas les changements des positions des objets qu'elle a transporté. Dans ce cas, il se peut que la fourmi revienne vers un objet qui a changé de position. Nous adaptons ces idées pour la classification des images et l'étendons pour le dépôt de pixels comme suit :

Chaque fourmi mémorise les  $m$  derniers objets qu'elle a ramassés ainsi que leurs emplacements sur tableau. Quand une fourmi transporte un pixel  $p_i$ , elle consulte sa mémoire et évalue la possibilité de le placeç è r dans la cellule d'un des pixels qu'elle a déjà transporté. Pour cela, elle calcule la fonction de similarité  $f(\ )$  pour chacune des cellules mémorisées. La cellule candidate à recevoir le pixel  $p_i$  sera celle pour laquelle la fonction  $f$  est maximum. La fourmi se dirige de son nid avec la cellule candidate et décide d'y déposer son pixel avec la probabilité  $p_{dépôt}$ . Si cette décision est négative, sa mémoire est désactivée et dans les prochaines itérations, elle tentera de déposer son pixel dans une autre cellule choisie aléatoirement, jusqu'à ce qu'elle y arrive.

## 5.3 Etude expérimentale

### 5.3.1 Les images de test

Afin de tester les performances de l'algorithme AntClust nous avons utilisé des images synthétiques et réelles présentées dans les figures 5.2 et 5.3.. L'intérêt des images synthétiques est qu'on de manière précise le nombre exact de classes ainsi que la qualité de la segmentation de référence. Ces informations ne sont évidemment pas transmises à AntClust et ne sont utilisées que pour tester la capacité de AntClust à retrouver la partition théorique des pixels. D'un autre côté, on peut comparer la qualité de la partition obtenue avec AntClust et celle de la vérité terrain. L'application de AntClust pour la segmentation d'images réelles conduit à une évaluation des résultats obtenus en l'absence d'informations sur la partition correcte. Cependant, elle permet l'analyse de la structure des classes générées en utilisant des mesures de validation

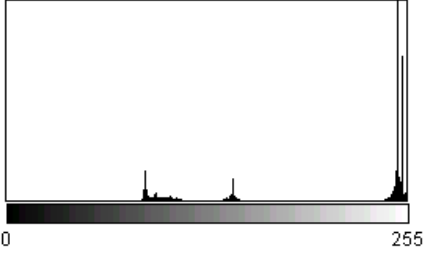
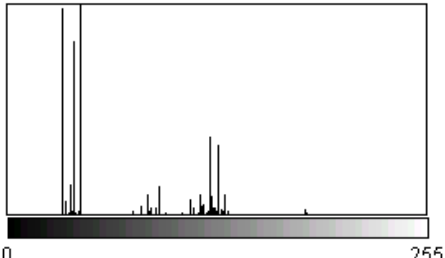
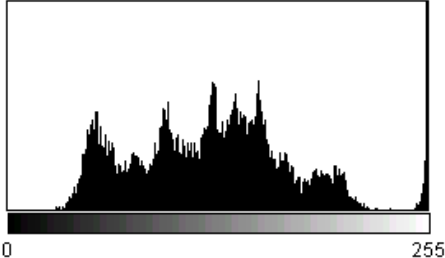
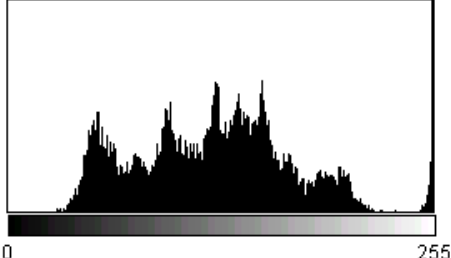


**Figure 5.2. Les images synthétiques de test**



**Figure 5.3. Les images réelles de test**

Le tableau 5.1. présente pour chacune des images, le nombre de classes( $K$ ), le nombre de niveau de gris présent dans l'image ainsi que l'histogramme associé. Pour les images réelles l'intervalle optimal pour le nombre des classes est obtenu de [Turi, 2001].

Les images	K	Nbre niveau de gris	Histogramme
Image synthétique 1	3	46	
Image synthétique 2	6	117	
Lenna	[5,10]	52	
Poivron	[6,10]	54	

**Figure 3.8. Caractéristiques des images de test**

### 5.3.3 Les mesures d'évaluation

Pour évaluer les résultats de classification obtenus par l'algorithme AntClust pour la segmentation des images synthétiques, nous avons choisi la mesure l'index de Rand dont l'utilisation est réponde pour valider la performance des algorithmes de classification. Pour étudier la performance de AntClust pour segmenter des images réelles, nous avons décidé d'utiliser la mesure d'intra-inter classes de Rosenberger. En plus de ces deux mesures, pour chaque expérimentation, nous donnons le nombre moyen de classes identifiées.

### 5.3.4 Comparaison des résultats

Les résultats obtenus avec AntClust, sont comparés avec ceux obtenus par l'algorithme classique de classification : l'algorithme KMEANS. Comme nous l'avons mentionné auparavant cet algorithme de classification converge rapidement vers un optimum local mais présente le principal inconvénient de nécessiter une partition de départ avec un nombre de classes fixé (ce qui n'est pas le cas de notre algorithme). Dans nos expériences, nous avons initialisés l'algorithme KMEANS avec une partition contenant un nombre de classes égal à la dimension de l'image et deux autres générées aléatoirement constituées de 20 et de 100 classes. Nous l'appellerons KMEANS, 20\_KMEANS et 100\_KMEANS dans la suite de cette section.

Le tableau 5.2 résume les valeurs des paramètres de AntClust et de KMEANS avec lesquels nous avons obtenu de bons résultats. Ces valeurs sont appliquées pour la segmentation de toutes les images de test.

<i>Paramètre</i>	Désignation	Valeur
<i>A</i>	Nombre de fourmis	10
<i>Q</i>	Contrôle la probabilité de choisir un pixel dans une cellule contenant deux pixels	0.7
<i>Tmax</i>	Nombre d'itérations maximum de AntClust	4500
<i>Nbre_iter</i>	Nombre d'itérations maximum de KMEANS	50

**Figure 3.9. Paramètres de AntClust et de KMEANS**

### 5.3.5 Les résultats

Cette section présente les résultats des évaluations conduites par les deux algorithmes de classification sur le jeu d'images de test utilisé. Pour chacune d'elles, 50 testes ont été menés afin d'avoir une estimation du comportement moyen de chacun des algorithmes. Pour chacune des mesures d'évaluation, nous avons calculé les valeurs moyennes ainsi que l'écart type.

Le tableau 5.3. montre les résultats de AntClust et de KMEANS pour la segmentation des images synthétique en utilisant l'index de Rand et la mesure de l'intra-inter classes. D'après les valeurs obtenues, on constate comme prévu l'incapacité de KMEANS, à extraire le bon nombre de classes quand le nombre de classes de départ est éloigné du nombre de classes originelles. Les résultats de 20-KMEANS sont meilleurs que ceux de 100-KMEANS qui sont bien sur mieux que ceux de KMEANS.

On constate aussi que l'algorithme KMEANS tend à fournir un nombre de classes égal au nombre de niveaux de gris présents dans les images à segmenter. Les résultats montrent clairement que l'algorithme AntClust bien qu'il ne nécessite pas une connaissance du nombre

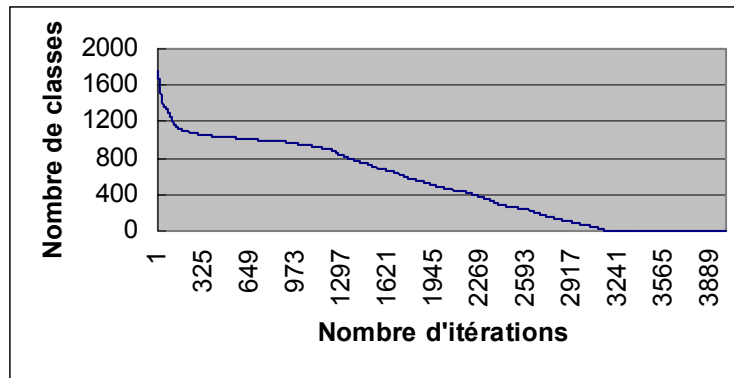


probable de classe, arrive à identifier un nombre égal (ou très proche) du nombre correct de classes avec un écart type presque nul. D'un autre coté, la qualité de la classification générée par l'algorithme AntClust est très proche de la qualité optimale comme l'atteste les valeurs de la valeur intra-inter classes réduite ce qui traduit une bonne homogénéité à l'intérieur des classes.

<b>Image Synthétique 1</b>	KMEANS	20-KMEANS	100-KMEANS	AntClust
Nb_classes	46 (0)	6 (0)	22 (0)	3.08 (0.284)
L'index de Rand	7,342 e-01 (0)	9,580 e-01 (0)	7,754 e-01 (0)	9,897 e-01 (0,433)
Valeur intra_inter classes	4,888 e-01 <b>(0)</b>	3,400 e-01 (0)	4.663e-01 (0)	5,082 e-02 (1.729)
<b>Image Synthétique 2</b>	KMEANS	20-KMEANS	100-KMEANS	AntClust
Nb_classes	117 (0)	10 (0)	99 (0)	6 (1.05)
L'index de Rand	6.199 e-01 (0)	6.843 e-01 (0)	6,266 e-01 (0)	8,984 e-01 (1.323)
Valeur intra_inter classes	4,981 e-01 (0)	4.864 e-01 (0)	4,979 e-01 (0)	4,500 e-02 (1.06)

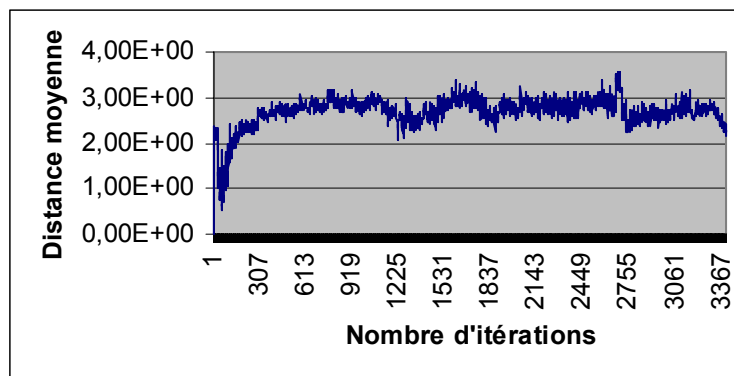
**Figure 3.10. Résultats de classification des images synthétiques par les algorithmes KMEANS et AntClust.**

La courbe de la figure 5.4. donne pour l'image synthétique 1 l'évolution du nombre de classes au cours des 3500 itérations.



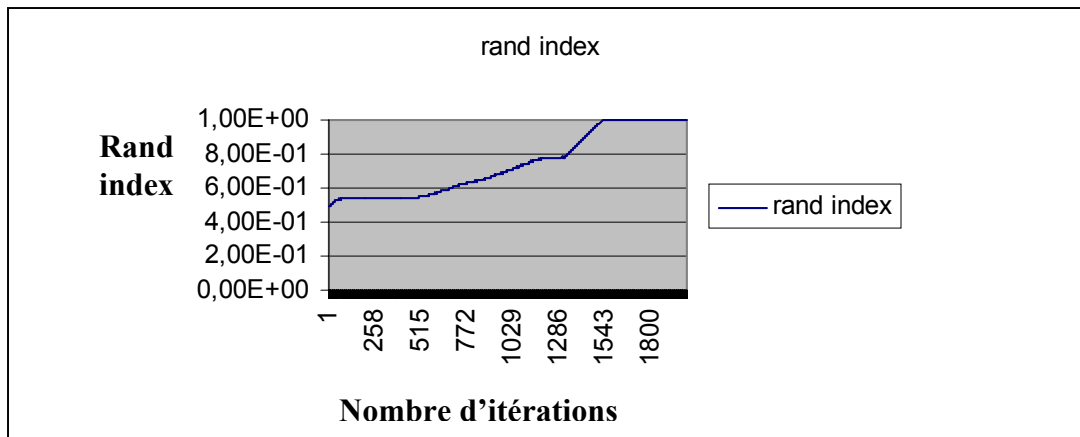
**Figure 5.4. Evolution du nombre de classes au cours des itérations de AntClust pour l'image synthétique 1.**

La figure 5.5. donne pour l'image synthétique 1 l'évolution de la distance moyenne des pixels par rapport au centre de la classe où ils ont été placés par les fourmis. La distance moyenne est nulle au départ car chaque pixel est placé dans une classe. Elle atteint son maximum de 3,50 au bout de 2700 itérations pour décroître au cours des prochaines itérations jusqu'à atteindre un minimum de 2,37 à partir duquel les fourmis améliorent la classification sans réduire le nombre de classes..



**Figure 5.5. Evolution de la distance moyenne des pixels au cours des itérations de AntClust pour l'image synthétique 1.**

Pour identifier l'évolution de l'erreur de classification, il est intéressant d'utiliser la courbe d'évolution de l'index de Rand à chaque itération (figure 5.6.). Ce critère donne une idée sur la qualité de la partition ainsi que l'adéquation de son nombre de classes avec le nombre de classes théorique. En effet plus l'écart entre le nombre de classes théorique et celui obtenu par l'algorithme de classification serait important plus l'index de Rand aurait une valeur faible.







**Figure 5.6. Evolution de l'index de Rand sur l'image synthtique1.**

Le tableau 5.4. résume les résultats de AntClust et de KMEANS pour la segmentation des images réelles en utilisant la mesure intra-inter régions. AntClust trouve toujours un nombre de classes proche de l'optimal. Les résultats obtenus confirment encore une fois la performance de AntClust pour l'extraction d'une classe optimale comparée à celle de l'algorithme KMEANS.

La figure 5.7 montre quelques exemples des images segmentées par AntClust avec différents nombre de classes extraits.

Lenna	KMEANS	20-KMEANS	100-KMEANS	AntClust
Nb_classes	52	19	52	10.09
	(0)	(0)	(0)	(1.789)
Valeur intra-inter classes	4,882 e-01	4,652 e-01	4,882 e-01	3,313 e-01
	(0)	(0)	(0)	(0.456)
Poivron	KMEANS	20-KMEANS	100-KMEANS	AntClust
Nb_classes	54	14	54	7.46
	(0)	(0)	(0)	(1.458)
Valeur intra-inter classes	4,887 e-01	4,334 e-01	4,887 e-01	2,462 e-01
	(0)	(0)	(0)	(0.433)

**Figure 3.11. Résultats de classification des images réelles par les algorithmes KMEANS et AntClust.**

Image	Image segmentée	Nombre de classes
Lenna		8
Lenna		10
Poivron		7
Poivron		8

**Figure 5.7. Exemples d'images segmentées par l'algorithme AntClust.**

## 5.4 Conclusion

AntClust est une nouvelle approche pour la classification non supervisée des images. Il est inspiré du comportement de tri de couvain observé chez les fourmis réelles. L'intérêt principal de cet algorithme est sa capacité à extraire automatiquement les classes présentes dans l'image sans connaître le nombre de classe a priori et sans partition de départ. Dans les précédents travaux les fourmis se déplacent sur une grille rectangulaire dont les paramètres sont difficiles à fixer et ont une grande influence sur la convergence de l'algorithme de fourmis. Dans AntClust, l'environnement des fourmis est un tableau de cases représentant les classes de l'image et reliée chacune à un nid collectif afin de faciliter le déplacement des fourmis. A travers cet environnement, les fourmis interagissent ensemble et construisent des partitions de bonne qualité en utilisant des règles simples de placement et de déplacement de

pixels. AntClust utilise une fonction de similarité locale qui mesure la similarité d'un pixel en terme de niveau de gris avec les autres pixels présents dans une classe. Les expériences effectuées montrent que AntClust fournit de très bons résultats sur des images synthétiques et réelles variées et s'avère comme prévu supérieur à l'algorithme classique KMEANS. De plus cet algorithme est par sa nature distribué donc facilement parallélisable.

# Chapitre 6

## Résolution collective de la segmentation par relaxation Markovienne

<b>Chapitre 6. Résolution collective de la segmentation par relaxation Markovienne.....</b>	<b>135</b>
6.1 Introduction .....	135
6.2 Motivations.....	135
6.3 Segmentation d'images par classification markovienne .....	136
6.3.1 Modélisation des observations .....	137
6.3.2 Modélisation des connaissances a priori .....	138
6.3.3 L'énergie totale .....	139
6.4 Transposition de OCF à la segmentation d'images par relaxation Markovienne ..	140
6.4.1 Le modèle de représentation .....	140
6.4.2 Comportement individuel de la fourmi .....	141
6.4.3 Le modèle de coopération entre les fourmis .....	141
6.4.4 La construction de partitions .....	142
6.4.4.1 Approche constructive.....	142
6.4.4.2 Approche perturbatrice.....	149
6.5 Résultats expérimentaux .....	153
6.5.1 Etude de la convergence.....	157
6.5.2 Etude de la robustesse .....	158
6.6 Conclusion.....	159

---

## Chapitre 6. Résolution collective de la segmentation par relaxation Markovienne.

---

### 6.1 Introduction

La segmentation par relaxation Markovienne, contrairement aux méthodes classiques de segmentation est robuste au bruit, et prend en considération la propriété de régularité spatiale en plus de la propriété d'homogénéité. La théorie de champs de Markov modélise le problème de segmentation en terme d'optimisation basé sur le critère du maximum A posteriori (MAP). La segmentation est alors considérée comme un problème d'étiquetage de pixels où on cherche à affecter à chaque pixel l'étiquette qui maximise l'estimateur MAP et donc minimise l'énergie globale. C'est un problème d'optimisation complexe qu'on se propose de résoudre en transposant le concept de OCF à la résolution collective du problème de segmentation d'images selon une approche des systèmes multi-agents réactifs

Ce chapitre est organisé comme suit : dans un premier temps nous présentons les motivations qui nous amènés à utiliser la métaphore des fourmis pour la résolution du problème de la segmentation dans le cadre des champs de Markov. Dans un deuxième temps, nous décrivons le principe général de notre méthode de segmentation,

### 6.2 Motivations

La fonction d'énergie globale est loin d'être convexe, et possède souvent un grand nombre de minima locaux. La segmentation par relaxation markovienne fait alors partie de la classes des problèmes d'optimisation NP-complet. Les algorithmes de relaxation sont souvent utilisé dans ce genre de problèmes tels que l'algorithme Iterated Conditionnel Mode.(ICM), l'algorithme Recuit Simulé (RS) et les algorithmes génétiques (AG). ICM est un algorithme de relaxation déterministe et nécessite qu'une partition initiale soit présentée en entrée et converge rapidement vers un optimum local qui peut être assez éloigné de la solution recherchée. Le seul moyen de contourner ce problème est de relancer la méthode avec une autre partition initial (méthodes multistart). Le RS et les AG sont des algorithmes de relaxation stochastiques qui permettent d'échapper à ces optimums locaux par une exploration stochastique de l'espace de solutions. Le problème de segmentation en tant qu'un problème d'optimisation complexe est un bon candidat pour les algorithmes basés OCF qui sont des algorithmes stochastiques construisant des solutions d'une manière incrémentale. L'idée de base est de représenter le problème de segmentation sous la forme d'une graphe et de rechercher un «meilleur» étiquetage de ces nœuds. Des fourmis artificielles circulent dans ce graphe de façon aléatoire et incomplète, à la recherche de «bonnes» étiquettes. Elles communiquent entre elles, à travers l'environnement, en déposant sur les arcs du graphe des traces de

phéromone qui tendent à attirer les fourmis artificielles dans une boucle de rétroaction positive, guidant de manière émergente la colonie vers une solution satisfaisante, si ce n'est la meilleure.

En plus de la difficulté de la tâche de segmentation, le problème même de l'extraction des classes homogènes dans l'image semble adapté au paradigme OCF basé sur l'intelligence artificielle distribuée qu'on a choisi de suivre. Ce domaine privilégie de par sa nature une approche locale de résolution. C'est à dire une approche où les décisions ne sont pas prises par un observateur global qui possède la synthèse du système et qui en manipule les composants, mais par chacun des composants. Ces composants, appelés agents sont très simples et n'ont qu'une vision partielle de l'univers dans lequel ils évoluent et avec lequel ils interagissent. Ils collaborent ensemble d'une manière inconsciente pour résoudre collectivement le problème de segmentation. Chaque agent possède un cycle d'exécution pendant lequel il commence par percevoir son environnement, ensuite, en fonction des informations provenant de l'environnement, il prend une ou plusieurs décisions. Chaque décision modifie l'état interne de l'agent, son comportement ainsi que l'environnement qui l'entoure orientant ainsi le comportement des autres agents.. Dans cette approche, les agents fourmis se déplacent sur l'environnement - image et construisent des solutions d'une manière concurrente et asynchrone. Ils coordonnent leurs activités et communiquent leurs résultats par des interactions stigmergique. La partition optimale émerge de la coopération entre les agents-fourmis.

### **6.3 Segmentation d'images par classification markovienne**

Ce paragraphe n'a pas pour objectif de présenter les bases théoriques de la modélisation markovienne, dont on pourra trouver un aperçu en annexe , mais de montrer comment segmenter une image dans un cadre stochastique markovien. Il concerne en premier lieu le formalisme de notre classification ainsi les différentes fonctions énergie utilisées dans notre méthode.

Dans le formalisme markovien, la segmentation d'image est vue comme un problème d'étiquetage caractérisé par :

(1) un ensemble de pixels; (2) un ensemble d'étiquettes; (3) une relation de voisinage entre les pixels, (4) une relation de compatibilité sur les paires d'étiquettes des pixels voisins et(5) une fonction objectif mesurant la qualité de chaque partition construite.

L'objectif est de trouver la partition optimale satisfaisant la contrainte d'homogénéisé et celle de la régularité spatiale.



Soit  $Y = \{Y_s, s \in S\}$  le champ des observations (dans notre cas niveau de gris) représentant l'ensemble des  $N$  pixels de l'image. Il est défini sur l'ensemble  $S$  des sites  $s$  tel que  $S = \{s_1, s_2, \dots, s_N\}$ . Nous cherchons à mettre en correspondance ce champ  $Y$  avec un champ des étiquettes  $E = \{e_s, s \in S\}$  prenant ces valeurs dans un ensemble fini  $\Lambda = \{1, \dots, L\}$  tel que  $L$  est le nombre total des classes présentes dans l'image et  $e_s = l$  indique que l'étiquette  $l$  est assignée au pixel  $s$ . L'espace  $\Omega$  représente la totalité des configurations possibles  $e = \{e_{s_1}, \dots, e_{s_N}\}$  sur l'image avec  $e_s \in \Lambda$ . Nous supposons que les champs  $Y$  et  $E$  sont des champs markoviens que nous intégrons dans un formalisme bayésien afin d'utiliser les contraintes issues d'informations a priori.

Le problème de la segmentation est vue comme l'estimation de la réalisation de  $E$ , connaissant la réalisation observée  $Y$ . On cherche en tout pixel de l'image, donc en tout site, une bonne estimation  $\hat{e}$  de la configuration du champ des étiquettes  $E$  connaissant les données  $Y$ . En se plaçant dans le contexte de la maximisation de la probabilité a posteriori MAP [Besag, 1974] on cherche à maximiser la probabilité  $P(E|Y)$ .

$$\hat{e} = \arg \max_{\Omega} \{P(E|Y)\} \quad (6.1.)$$

### 6.3.1 Modélisation des observations

Si on fait l'hypothèse que la distribution des intensités de pixels de chaque classe suit une loi normale de moyenne  $\mu$  et de variance  $\sigma$ , la probabilité conditionnelle  $P(Y|E)$  sera définie par :

$$P(Y|E) = \prod_{s \in S} \left( \frac{1}{\sqrt{2\pi}} \text{Exp} \left\{ -\frac{(y_s - \mu_{e_s})^2}{2\sigma_{e_s}^2} - \log(\sigma_{e_s}) \right\} \right) \quad (6.2.)$$

où  $e_s$  est l'étiquette assignée au pixel  $s$ .

$P(Y|E)$  peut se réécrire sous la forme suivante :

$$P(Y|E) = \frac{1}{(2\pi)^{\frac{|S|}{2}}} \text{Exp} \left[ -\sum_{s \in S} \left\{ \frac{(y_s - \mu_{e_s})^2}{2\sigma_{e_s}^2} + \log(\sigma_{e_s}) \right\} \right] \quad (6.3.)$$

L'équivalence entre la distribution de Gibbs et les champs de Markov permet de définir la fonction énergie  $U(Y/E)$  comme suit :

$$U(Y|E) = \sum_{s \in S} \left\{ \frac{(y_s - \mu_{e_s})^2}{2\sigma_{e_s}^2} + \ln(\sqrt{2\pi}\sigma_{e_s}) \right\} \quad (6.4.)$$

La seule définition de  $U(Y|E)$  permet en cherchant à la minimiser, d'effectuer une classification non contextuelle ou aveugle.

### 6.3.2 Modélisation des connaissances a priori

Les champs de Markov permettent l'introduction des connaissances a priori dans le processus de classification afin de favoriser la formation de régions homogènes plutôt qu'une multitude de petites régions. Ce processus de régularisation est réalisé par l'introduction des fonctions de potentiel afin d'influer sur l'affectation d'un pixel à une classe en fonction du voisinage de ce pixel.

En modélisant le champ des étiquettes  $E$  par un champ de Markov relativement au système de voisinage  $N_s$  tel que  $N_s$  est l'ensemble des pixels voisins de  $s$  et en vertu du théorème de Hammersley-Clifford [Geman 1984],  $E$  est un champ de Gibbs défini par la famille  $\{V_c, c \in C\}$  de potentiels sur les cliques engendrées par le système de voisinage  $N_s$ .

Soit  $e$  une réalisation particulière de  $E$ ,  $P(E)$  prend la forme suivante :

$$P(E) = \frac{1}{Z} \exp \left\{ - \sum_{c \in C} V_c(e) \right\} \quad (6.5.)$$

$Z$  est une fonction de normalisation.  $V_c(e)$  est appelée potentiel d'interactions pour une clique  $c$ . Un sous ensemble  $c$  de  $S$  est appelé une clique relative au système de voisinage  $N$  si tous les pixels de  $c$  sont voisins deux à deux.

Le potentiel d'interaction relatif à l'étiquette  $e_s$  relatif au pixel  $s$ , connaissant les étiquettes des pixels voisins à  $s$  (noté  $V(e_s/e_{v_s})$ ) est tel qu'il favorise l'appartenance de deux pixels voisins à la même étiquette. Ainsi l'étiquette la plus probable pour un pixel sera celle la plus présente dans son voisinage. Il existe dans la littérature de nombreuses modélisations du potentiel d'interaction

Le modèle de Potts est largement utilisé comme a priori à des fins de segmentation. C'est un modèle à interactions par paires telle que les cliques sont constituées de pixels voisins. La fonction potentielle utilisée pénalise alors les cliques non homogènes, c'est à dire les cliques dont les sites ont des étiquettes différentes. Elle est définie comme suit :

$$V(e_1, e_2) = -\beta \text{ si } e_1 = e_2 \text{ et } 0 \text{ sinon.} \quad (6.6.)$$

$\beta$  est un paramètre positif qui contrôle l'influence des pixels voisins.

Dans ce travail de thèse, nous avons modélisé les connaissances a priori par une fonction linéaire qui s'écrit comme suit :

$$V(e_s / e_{V_s}) = \frac{\sum_{s' \in V_s} V(e_s, e_{s'})}{\sum_{s' \in V_s} 1} \quad (6.7.)$$

Cette fonction est représentée figure pour un voisinage normalisé variant entre 0 et 1 ( $0 \leq \frac{\sum_{s' \in V_s} V(e_s, e_{s'})}{\sum_{s' \in V_s} 1} \leq 1$ ). Le voisinage est normalisé en divisant le nombre de voisins partageant la même étiquette par le nombre total des voisins.

L'énergie  $U(E)$  peut s'écrire alors :

$$U(E) = \sum_{c \in C} V_c(e) \quad (6.8.)$$

### 6.3.3 L'énergie totale

La fonction énergie totale à minimiser lors de la segmentation est la somme des énergies  $U(Y/E)$  et  $U(E)$ . L'énergie  $U(Y/E)$  est un terme d'attache aux données et représente la vraisemblance de l'observation étant donné un champ d'étiquette. L'énergie  $U(E)$  est un terme de régularisation lié au modèle a priori imposant une continuité spatiale. Le paramètre  $\beta$  a donc un rôle de régularisation et un rôle de pondération de l'énergie  $U(E)$  vis-à-vis de l'énergie  $U(Y/E)$ . Une valeur proche de 0 donne une plus grande importance au terme  $U(Y/E)$ , et donc à l'intensité du niveau de gris, et une valeur plus élevée donnera une plus grande importance à la formation de régions plus compactes. L'énergie totale s'écrit comme suit :

$$U(E/Y) = \left\{ \underbrace{\sum_{s \in S} \frac{(y_s - \mu_{e_s})^2}{2\sigma_{e_s}^2}}_{U(Y/E)} + \underbrace{\sum_{s \in S} \ln \sqrt{2\pi} (\sigma_{e_s}) + \sum_{c \in C} V_c(e)}_{U(E)} \right\} \quad (6.9.)$$

L'estimée  $\hat{e}$  est alors définie par :

$$\hat{e} = \arg \min_{\Omega} \left\{ \sum_{s \in S} \frac{(y_s - \mu_{e_s})^2}{2\sigma_{e_s}^2} + \sum_{s \in S} \log(\sigma_{e_s}) + \sum_{c \in C} V_c(e) \right\} \quad (6.10 .)$$

Cette minimisation se fait sur l'ensemble des configurations dont le cardinal est très important. Il n'est pas possible de tester chacune des configurations de l'ensemble  $\Omega$ . Plusieurs méthodes de relaxation existent dans la littérature pour pallier ce problème. Nous avons décidé de s'inspirer du comportement de fourragement observé chez les fourmis pour résoudre ce problème d'optimisation.

## 6.4 Transposition de OCF à la segmentation d'images par relaxation Markovienne

Nous décrivons maintenant comment le modèle de recueil de nourriture chez les fourmis a été exploité et adapté pour la résolution du problème de segmentation par relaxation markovienne. Les propriétés du modèle de construction de partition basé OCF sont :

- 1 Une perception limitée des agents-fourmis ;
- 2 Un mécanisme d'assignement des étiquettes aux pixels probabiliste ;
- 3 L'absence de représentation des autres agents-fourmis ;
- 4 L'interaction indirecte via l'environnement ;
- 5 Un mécanisme d'auto-organisation qui permet d'émerger la partition optimale.

Le problème de la segmentation peut être vu alors comme un maillage de nœuds interconnectés représentant les pixels qu'il faut étiqueter. L'action des fourmis se définit alors comme un déplacement de nœud en nœud. Le choix d'une étiquette pour chaque nœud est guidé par la phéromone et dans certains cas par une fonction heuristique. Il s'agit alors de définir un modèle de représentation adéquat, des règles de transition stochastique de chaque fourmi qui régissent son comportement qui doit notamment prendre en compte les différentes contraintes du problème ainsi que le modèle d'interaction entre les fourmis.

### 6.4.1 Le modèle de représentation

Le modèle de représentation repose sur la définition des éléments suivants :

- Modélisation de l'environnement des fourmis par un graphe sur lequel se déplacent les fourmis la recherche d'un étiquetage optimal pour les pixels ;

- Représentation d'une solution possible qui représente une partition acceptable ;
- Représentation de la phéromone artificielle ;
- Représentation de l'information heuristique si elle est disponible.

#### **6.4.2 Comportement individuel de la fourmi**

Le comportement individuel d'une fourmi est décrit par son cycle d'exécution durant lequel elle construit une partition possible. Le graphe-environnement est exploré par une colonie de  $K$  fourmis distribuées spatialement d'une manière aléatoire. Les fourmis n'ont qu'une vision partielle de l'environnement dans lequel elles évoluent. Chaque fourmi possède un cycle d'exécution pendant lequel elle se comporte au départ d'une manière aveugle ou aléatoire car aucune information n'est mise à sa disposition, ensuite, en fonction des informations provenant de l'environnement et stockées sous forme de trace de phéromone, elle raffine ses décisions et donc sa partition. A chaque décision et grâce à ses phéromones, la fourmi change son comportement ainsi que celui de ses congénères.

#### **6.4.3 Le modèle de coopération entre les fourmis**

Le modèle de coopération entre les fourmis est basé sur la communication indirecte (stigmergie). Cette coopération est mise en œuvre par un processus de renforcement positif des traces de phéromones associées aux meilleures partitions. Lors de la phase de renforcement, la mise à jour des traces de phéromones est fonction de la qualité de la partition obtenue. Une des propriétés de la phéromone est son caractère volatil. Dans le modèle virtuel, un mécanisme similaire sera utilisé afin d'éviter une convergence prématurée (stagnation) due à la découverte d'un optimum local.

Dans le cadre de cette thèse, nous avons développé deux approches pour la construction d'une partition possible

- Une approche « constructive » où les fourmis construisent en parallèle et selon un processus itératif et incrémental les partitions possibles de l'image.
- Une approche « perturbatrice » où les fourmis modifient ou proposent de nouvelles partitions à partir de partitions existantes.

## 6.4.4 La construction de partitions

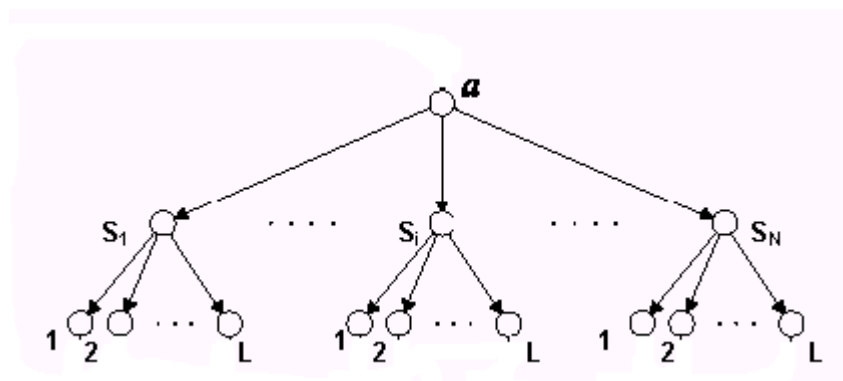
### 6.4.4.1 Approche constructive

#### 6.4.4.1.1 L'environnement

L'environnement dans lequel évoluent les fourmis est un graphe  $G = (N; A)$  tel que :

- $N$  est l'ensemble des nœuds du graphe tel que  $N = S\_noeuds \cup L\_noeuds$  où les  $S\_noeuds \in S$  sont les nœuds pixels de l'image, et les  $L\_noeuds \in \Lambda$  sont les nœuds étiquettes possibles. On notera que les  $L\_noeuds$  sont les nœuds terminaux du graphe  $G$ .
- $A$  est l'ensemble des arêtes du graphe. Chaque  $S\_noeud$  est relié par une arête à chaque  $L\_noeud$ .

Pour simuler le déplacement des fourmis réelles on introduit un nœud particulier  $a$  représentant le nœud de départ de toutes les fourmis.



**Figure 6.2. Représentation graphique du problème de segmentation**

Une partition construite par une fourmi  $k$  est notée par  $e^k$  et se présente sous la forme de  $N$  couples de la forme  $(s, l)$  correspondant à l'affectation de l'étiquette  $l$  au pixel  $s$ .

On associe à la partition  $e^k$  une fonction objectif définie par

$$U(e^l) = \sum_{s \in S} \frac{(y_s - \mu_{e_s})^2}{2\sigma_{e_s}^2} + \sum_{s \in S} \log(\sigma_{e_s}) + \sum_{s \in S} \beta \cdot V(e_s / e_{V_{ss}}) \quad (6.11 .)$$

$\beta$  est un paramètre positif qui contrôle l'influence des pixels voisins (il est mis à 1 dans nos expérimentations). C'est un terme de régularisation. Le problème revient alors à trouver les couples d'affectation qui correspondent au minimum de la fonction objectif.

### 6.4.4.1.2 Représentation de la phéromone artificielle

Chaque fourmi  $k$  lors de la construction de sa solution est guidée par la trace de phéromone artificielle  $\tau(s, l)$  associée à la paire  $(s, l)$  qui représente la désirabilité de mettre  $e_s = l$  dans sa solution. Cette information est rendue disponible par des tentatives précédentes d'autres fourmis. La trace de phéromone est une information numérique codée dans une matrice  $S \times \Lambda \rightarrow R^+$ .

	$0$	$1$	.....	$L-1$
$S1$				
$S2$				
.....				
$SN$				

Figure 6.3. Structure de la phéromone artificielle

### 6.4.4.1.3 Construction des partitions

La construction d'une partition est obtenue selon un processus incrémental et itératif. Durant une itération, les  $K$  fourmis construisent en parallèle  $K$  partitions en effectuant chacune  $N$  assignements possibles selon un processus stochastique. A chaque instant  $t$  une fourmi individuelle construit sa partition d'une manière incrémentale en lui ajoutant de nouvelles composantes. Une composante d'une partition est une paire  $(s, l)$  représentant l'attribution de l'étiquette  $l$  au pixel  $s$ .

A partir d'un nœud fictif  $a$  appelé nœud de départ, les fourmis explorent en parallèle l'environnement image et contribuent à l'émergence de la partition optimale. Chaque fourmi se déplace d'un nœud-pixel vers un autre jusqu'à ce que tous les pixels de l'image soient traités et exécute le même processus d'étiquetage pour tous les pixels de l'image. Ce processus est basé sur deux phases : à partir du nœud  $a$ , un pixel  $s$  est choisi parmi les  $N$  pixels de l'image représentés par  $s_1, s_2, \dots, s_N$ . Une fois le pixel  $s$  choisi, la fourmi doit lui affecter une étiquette  $l$  de l'ensemble  $\Lambda$ . Une fois le pixel étiqueté, la fourmi revient au nœud  $a$  encore, et le processus d'étiquetage est répété tant qu'il existe encore des pixels non encore étiquetés, ce qui complète la partition de l'image.

Pour une fourmi la base décisionnelle pour le choix d'une étiquette de classe est fournie par la trace de phéromone artificielle  $\tau(s, l)$  associée à la paire  $(s, l)$  qui représente la désirabilité de mettre  $e_s = l$  dans sa solution, cette information est rendue disponible par des tentatives précédentes d'autres fourmis. La trace de phéromone est une information numérique codées dans une matrice de dimension  $(N, L)$  tel que  $N$  est la dimension de l'image et  $L$  est le nombre total de classe présentes dans l'image.

Une fois que chaque fourmi a établi une solution complète représentant un marquage possible, ce dernier est amélioré en utilisant une méthode de recherche locale afin d'augmenter sa qualité. A la fin du processus de classification, les traces de phéromone sont mises à jour localement pendant la construction de solutions et globalement à la fin de chaque itération selon la qualité de la meilleure solution trouvée à partir du commencement du processus de segmentation. L'algorithme s'arrête quand un maximum du nombre d'itérations a été exécuté.

Pour situer les éléments essentiels de notre méthode, nous nous référons au squelette de l'application (Algorithme 1). On s'attardera sur les éléments suivants : initialisation de la phéromone, construction d'une partition, modification de la partition, la mise à jour de la phéromone et la recherche locale

### ***Initialisation***

$K$  fourmis sont placées sur le nœud  $a$  du graphe. Chaque fourmi  $k$  possède un vecteur  $e^k$  qui stocke la solution trouvée par la fourmi.

Toutes les valeurs de phéromone  $\tau$  pour les paires  $(s, l)$  sont initialisées avec une valeur  $\tau_0 = 1/(N \cdot U(e^0))$  où  $e^0$  est une première solution créée par une heuristique glouton. Les pixels de l'image sont traités dans l'ordre, et pour chacun d'eux est assignée une étiquette de classe qui réduit au minimum la fonction locale d'énergie donnée par :

$$U_s(e_s = l | y_s, x_{s'}, s' \in N_s \setminus s) = \frac{(y_s - \mu_l)^2}{2\sigma_l^2} + \log(\sigma_l) + \sum_{c: s \in c} V_c(x) \quad (6.12 .)$$

Les deux premiers termes de  $U_s$  représentent l'attache aux données et le dernier terme introduit une contrainte de régularisation dans le modèle.

### ***Construction de solution***

Pendant la construction de la solution, les  $K$  fourmis établissent en parallèle un marquage possible pour les  $N$  pixels de l'image. La construction d'une solution implique deux étapes. D'abord, la fourmi  $k$  se déplace du nœud de départ  $a$  vers un pixel choisi aléatoirement parmi les pixels non encore traités. Une fois le pixel choisi, une étiquette lui est assignée. Pour la deuxième étape la fourmi se base sur les traces de phéromone  $\tau(s, l)$  représentant le désir d'assigner l'étiquette de classe  $l$  au pixel  $s$ .



*Algorithme 6.1 Segmentation d'images par ACS-MRF*

*Générer une solution initiale  $e^0$  en utilisant un algorithme glouton  
soit  $\tau_0 = 1/(N \cdot U(e^0))$ .*

*Pour chaque paire  $(s, l)$  Soit  $\tau(s, l) = \tau_0$  Fin pour*

*Repeter*

*Pour chaque fourmi  $k=1, \dots, K$  Faire*

*Pour chaque pixel  $s \in S$  Faire*

*Avec probabilité  $q_0$  :*

$$l = \arg \max_{u \in \Lambda} \tau(s, u)$$

*Sinon*

*Choisir  $l$  aléatoirement à partir de l'ensemble des*

*étiquettes  $\Lambda$  avec une probabilité  $\frac{\tau(s, l)}{\sum_{u \in \Lambda} \tau(s, u)}$*

*Mettre  $e_s^k = l$*

*Fin pour*

*Fin pour*

*Pour chaque fourmi  $k=1, \dots, K$  Faire*

*Pour chaque pixel  $s \in S$  Faire*

$$\tau(s, l) = (1 - \rho) \cdot \tau(s, l) + \rho \cdot \tau_0, \text{ telle que } e_s^k = l$$

*Fin pour*

*Fin pour*

*Pour chaque fourmi  $k=1, \dots, K$  Faire*

*Améliorer localement  $e^k$*

*Evaluer  $e^k$  en calculant  $U(e^k)$*

*Fin pour*

$$e^{ib} = \operatorname{argmin}_{e=e^1, \dots, e^K} (U(e))$$

*Si  $U(e^{ib}) < U(e^{gb})$  Alors  $e^{gb} = e^{ib}$*

*Finsi*

*Pour chaque pixel  $s \in S$  Faire*

*Pour chaque étiquette  $l \in \Lambda$  Faire*

$$\tau(s, l) = (1 - \alpha) \cdot \tau(s, l) + \alpha \cdot \Delta\tau(s, l)$$

$$\Delta\tau(s, l) = \begin{cases} \frac{1}{U(e^{gb})} & \text{if } (s, l) \in e^{gb} \\ 0 & \text{otherwise} \end{cases}$$

*Fin pour*

*Fin pour*

*Jusqu'à  $M$  Itérations*

La décision d'assigner une étiquette de classe  $l$  au pixel  $s$  est basée sur la règle *pseudo-aléatoire* proposée par Dorigo et Gambardella dans [Dorigo 1997]. La transposition de cette règle à notre problème est modélisée comme suit [Ouadfel 2003a][ouadfel 2003b].

Soit  $q$  un nombre aléatoire uniformément distribué dans  $[0,1]$  et  $q_0$  une probabilité fixe.

- Si  $q \leq q_0$  la fourmi choisit d'assigner au pixel  $s$  l'étiquette de classe  $l$  pour laquelle la trace de phéromone  $\tau(s, l)$  est maximum .

$$l = \arg \max_{u \in \Lambda} \tau(s, u) \quad (6.13 .)$$

- Autrement on prend une décision aléatoire telle que une étiquette de classe  $l = 1 \dots L$  est choisie avec la probabilité  $P(s, l)$

$$P(s, l) = \frac{\tau(s, l)}{\sum_{u \in \Lambda} \tau(s, u)} \quad (6.14 .)$$

Le paramètre  $q_0$  contrôle l'importance relative de l'exploitation de la connaissance déjà accumulée et reflétée dans les traces de phéromone contre l'exploration des nouvelles solutions. Nous pouvons interpréter la colonie de fourmi comme un système d'apprentissage par renforcement, dans lequel les renforcements (les traces de phéromone) modifient la force de l'assignement Pixel - étiquette. Dans eq.(6.12) une fourmi exploite avec la probabilité  $q_0$  l'expérience qui s'est accumulée par la colonie de fourmis sous forme de trace de phéromone (la quantité de trace de phéromone tendra à augmenter pour les paires  $(s, l)$  qui appartiennent à la meilleure partition, les rendant plus souhaitables). Dans eq.(6.13) la fourmi applique avec la probabilité  $(1 - q_0)$  une exploration de nouveaux assignements en choisissant aléatoirement une étiquette pour à un pixel, avec une distribution de probabilité qui est une fonction des traces de phéromone accumulée par des fourmis.

Le choix de la valeur de  $q_0$  est important. Une valeur trop grande de  $q_0$  (proche de 1) rendrait le choix des étiquettes purement élitiste et le processus d'optimisation s'apparenterait à une démarche gloutonne. Les risques de stagnation sont alors importants. Une valeur trop petite de  $q_0$  conduirait à des temps de convergence trop importants.

### ***La mise à jour de la matrice de phéromones***

Le modèle d'interaction entre les fourmis est stigmergique. Les fourmis ne communiquent pas directement mais à travers le changement de la matrice de phéromones. En effet, les traces de

phéromones représentent une sorte de mémoire collective entre les fourmis. En modifiant ces traces, les fourmis modifient la façon dont le problème va être représenté et perçu par eux par leurs congénères. Le choix de la méthode de modification des traces de phéromones est donc important, pour obtenir les meilleurs résultats. On considère ici deux stratégies de modification :

#### *Stratégie de ACS*

La mise à jour des traces de phéromones est effectuée de manière locale (par chaque fourmi) et de manière globale (par la fourmi qui a trouvé la meilleure solution). La mise à jour locale est effectuée à chaque fois qu'une fourmi termine la construction de sa solution et a pour but de diminuer la quantité de traces de phéromones associés aux paires  $(s,l)$  sélectionnées par la fourmi afin de réduire son attraction pour les autres fourmis lors de la construction de leurs solutions et d'encourager ainsi l'exploration des autres choix possibles (d'autres assignments). La mise à jour locale de phéromone est appliquée selon la formule suivante:

$$\tau^{new}(s, l) = (1 - \rho) \cdot \tau^{old}(s, l) + \rho \cdot \tau_0 \quad (6.15 .)$$

$\rho$  est un paramètre dans  $]0,1[$ , et représente l'évaporation de la trace de phéromone comme dans le comportement des fourmis réelles.  $\tau_0$  est la valeur initiale de phéromone.

Une fois que toutes ont terminé la construction de leurs solutions, les traces de phéromone sont alors modifiées. Seule la fourmi qui a trouvé la meilleure partition  $e^{gb}$  telle que  $U(e^{gb}) = \min_{k=1, \dots, K} (U(e^k))$ , est autorisée à déposer les phéromones additionnelles sur les paires  $(s, l)$  appartenant à sa solution. La mise à jour globale est faite en appliquant la règle de mise à jour globale suivante:

$$\tau(s, l) = (1 - \alpha) \cdot \tau(s, l) + \alpha \cdot \Delta\tau(s, l) \quad (6.16 .)$$

là où

$$\Delta\tau(s, l) = \begin{cases} \frac{1}{U(e^{gb})} & \text{if } (s, l) \in e^{gb} \\ 0 & \text{otherwise} \end{cases} \quad (6.17 .)$$

avec  $0 < \alpha < 1$  l'évaporation de la trace de phéromone. Le processus d'évaporation permet d'éviter d'être piégé dans des solutions sous-optimales. L'augmentation des valeurs des traces de phéromones associées aux paires de la meilleure solution intensifie la recherche dans son voisinage par les autres fourmis pendant les prochaines itérations.

### Stratégie de MMAS

Selon cette stratégie, seule la fourmi ayant trouvé la meilleure partition est autorisée à effectuer une mise à jour de la matrice de phéromone. Cependant les quantités de phéromones sont bornées dans l'intervalle afin d'éviter la stagnation de la recherche. Elles sont initialisées au départ à la valeur maximale.

### Recherche locale

Les métaheuristiques de colonies de fourmis sont souvent plus efficaces quand elles sont hybridées avec des algorithmes de recherche locale. Ces derniers sont utilisés pour optimiser les solutions trouvées par les fourmis et accélérer la convergence de l'algorithme vers les meilleures solutions. Chaque solution créée par une fourmi est ainsi améliorée avant son évaluation. L'algorithme de recherche locale comporte deux composants: la définition d'un voisinage  $N(e)$  permettant d'obtenir les solutions voisines de la solution  $e$  et la donnée d'une stratégie de recherche permettant l'exploration de ce voisinage. L'ensemble  $N(e)$  est l'ensemble des partitions admissibles qui sont atteignables à partir de  $e$  en changeant les affectations des pixels. La stratégie d'exploration la plus simple consiste à aller d'une solution à une solution voisine qui soit meilleure que la solution courante jusqu'à ce que l'algorithme s'arrête sur un optimum local. Dans notre algorithme, pour chaque solution  $e$  nous considérons les solutions voisines  $e' \in N(e)$  et on sélectionne la première solution  $e'$  pour laquelle  $U(e') < U(e)$ , c'est à dire que la première solution qui diminue la fonction objective globale est retenue. Les solutions voisines d'une solution sont obtenues en permutant les étiquettes de deux pixels. Le squelette général de l'algorithme de recherche locale est donné dans l'encadré 6.2).

#### Algorithme 6.2. Algorithme de recherche locale

Soit  $e \in E$  une partition à améliorer

Soit  $w = \phi$

**Tant que** ( $N(e)/w \neq \phi$ ) **Faire**

$e' = \text{GénétrerSolution}(e)$

**Si**  $U(e') < U(e)$  **Alors**

    Retourner  $e'$

**Sinon**

$w = w \cup e'$

**Fin tant que**

**Figure 6.4.**

**Figure 6.5.**

#### 6.4.4.2 Approche perturbatrice

Cette approche se distingue principalement de l'approche précédente par (1) la manipulation des solutions, (2) la manière dont la phéromone est exploitée et mise à jour, (3) l'utilisation d'une fonction heuristique afin de guider les fourmis et (4) l'introduction d'une étape de diversification afin de mieux explorer l'espace de solutions [Ouadfel, 2005b].

##### 6.4.4.2.1 Représentation d'une partition possible

Une partition construite par une fourmi  $k$  est notée par  $e^k$  et se présente sous la forme de couples de la forme  $(s, s')$  correspondant à l'affectation des pixels  $s$  et  $s'$  dans la même classe.

Comme dans la méthode précédente, on associe à la partition  $e^k$  une fonction objectif définie par

$$U(e^k) = \sum_{s \in S} \frac{(y_s - \mu_{e_s})^2}{2\sigma_{e_s}^2} + \sum_{s \in S} \log(\sigma_{e_s}) + \sum_{s \in S} V(e_s / e_{v_{ss}}) \quad (6.18.)$$

##### 6.4.4.2.2 Représentation de la phéromone artificielle

Les traces de phéromone représentent l'information stigmergique et sont conservées dans une matrice  $\tau$  de dimension  $(N, M)$  tel que  $N$  est la dimension de l'image et  $M$  est le nombre total des voisins de chaque pixel (figure 6.3). Ici on ne considère que les 8 plus proches voisins.

	$S'_0$	$S'_1$	.....	$S'_7$
$S_0$				
$S_1$				
.....				
$S_{N-1}$				

Figure 6.6. Structure de la phéromone artificielle

##### 6.4.4.2.3 Représentation de l'information heuristique

Le but de l'heuristique est de favoriser le placement de pixels voisins ayant des niveaux de gris proches dans la même classe. Pour cela, l'heuristique attribue une valeur quantifiant la désirabilité d'affecter une étiquette à un pixel en tenant compte de son voisinage immédiat. La fonction heuristique utilisée est basée sur le coefficient de corrélation linéaire qui fournit une mesure sur la similarité en intensité dans le voisinage local de deux pixels voisins.

Les valeurs de la fonction heuristique sont calculées à partir de coefficient de corrélation linéaire une fois pour toute pour tous les pixels de l'image et sont stockées dans une matrice de dimension (N, M) ; tel que N est la dimension de l'image et M=8 si on considère pour chacun des pixels, seulement ses 8 plus proches voisins. La fonction heuristique est définie par :

$$\eta(s,s') = \frac{corr(s,s') + 1}{2} \quad \text{tel que } s' \in N_s \quad (6.19 .)$$

où  $corr(s,s')$  représente le coefficient de corrélation linéaire entre le voisinage immédiat des deux pixels s et s' .

Les valeurs de la fonction heuristique sont dans l'intervalle [0,1]. Elles indiquent de degré de similarité entre les pixels considérés en terme de niveau de gris. Elles tendent vers 1 pour une forte corrélation et vers 0 dans le cas contraire.

#### **6.4.4.2.4 Construction des partitions**

Dans cette méthode, les fourmis ne construisent pas totalement de nouvelles partitions, mais modifient celles déjà existantes en appliquant des changements d'étiquettes des pixels en se basant sur les traces de phéromones  $\tau$  et une information supplémentaire fournie par une fonction heuristique  $\eta$ . Dans la mise à jour, on retrouve également le fait comme dans la méthode précédente que seule la fourmi qui a trouvé la meilleure partition est autorisée à renforcer les traces de phéromones. De même, quand les fourmis terminent la modification de leur partition, on applique une recherche locale afin d'améliorer la performance de cet algorithme. Un mécanisme de diversification a été incorporé dans le processus de segmentation afin de déplacer les fourmis dans de nouvelles zones de l'espace de recherche.

Dans la section suivante, nous présentons en détail les différentes étapes de l'algorithme proposé (algorithme 6. 3).

### Algorithme 6.3. L'Algorithme HACSEG

*/\* Initialisation\*/*

*Génération aléatoire de K partitions aléatoires ;  
 Appliquer l'algorithme ICM à chacune des partitions ;  
 Choisir la meilleure partition  $e^{best}$  ;  
 Initialiser les traces de phéromones en utilisant la qualité de  $e^{best}$  .  
 Calculer les valeurs de la fonction heuristique ;*

*Pour  $t = 1, \dots, t_{max}$*

*/\*Modification des partitions\*/*

*Pour chacune des fourmis faire  
     *Modifier la partition en suivant les traces de phéromones et la fonction heuristique ;  
     Appliquer la recherche locale ;*  
 Fin pour*

*Mise à jour des traces de phéromones*

*Si nécessaire alors Diversifier*

*Fin Pour*

#### **Initialisation**

Initialement, on associe à chaque fourmi  $k$  une partition générée aléatoirement. A chaque pixel  $s \in S$ , on affecte une étiquette  $l$  choisie aléatoirement de l'ensemble  $\Lambda$ . Ces partitions sont ensuite améliorées en utilisant l'algorithme ICM. A partir des  $K$  partitions ainsi obtenues, on choisit la meilleure partition  $e^{best}$  qui minimise la fonction objectif. Chacune des paires  $(s, l)$  est alors initialisée avec la qualité de  $e^{best}$ .

#### **La modification des partitions**

Durant chaque itération, les fourmis modifient leurs partitions en déplaçant les pixels d'une classe vers une autre. Chaque fourmi applique un nombre  $m=N/2$  de modification de sa partition de départ en suivant les traces de phéromones et la fonction heuristique. Ce changement a pour but de regrouper dans une même classe des pixels voisins en tenant compte de leur voisinage immédiat. Pour cela, on choisit d'abord un pixel  $s \in S$  d'une manière aléatoire, ensuite un autre pixel  $s'$  appartenant à son voisinage immédiat  $N_s$  est sélectionné pour être placé avec lui dans sa classe selon l'information fournie par la phéromone et celle fournie par l'information heuristique. Plus précisément, soit  $q$  une valeur aléatoire uniformément distribuée dans  $[0,1]$  et  $q_0$  une probabilité fixe. Le choix du pixel  $s' \in N_s$  se fait selon la décision probabiliste suivante :

- Si  $q \leq q_0$  le pixel  $s'$  sélectionné pour être regrouper avec le pixel  $s$  dans sa classe est celui pour lequel le profil phéromone-heuristique est maximum :

$$\eta(s,s')^\beta \cdot \tau^k(s,s')^\gamma \text{ est maximum} \quad (6.20 .)$$

- Sinon le pixel  $s'$  est sélectionné aléatoirement parmi les 8 voisins de  $s$  selon la distribution de probabilité suivante :

$$P(s,s') = \frac{\eta(s,s')^\beta \cdot \tau^k(s,s')^\gamma}{\sum_{j \in N_s} \eta(s,j)^\beta \cdot \tau^k(s,j)^\gamma} \quad (6.21 .)$$

où

- $\beta$  et  $\gamma$  sont des paramètres qui mesurent l'importance relative de la trace de phéromone et la valeur de la fonction heuristique.
- $\eta(s,s')$  est la mesure de la corrélation entre le voisinage de  $s$  et celui de  $s'$  ;
- $\tau(s,s')$  est la désirabilité de regrouper les deux pixels  $s$  et  $s'$  dans la même classe dans la nouvelle partition.

Le choix de la valeur de  $q_0$  est important car il doit permettre de réaliser un équilibre entre le mécanisme d'exploration et celui de l'exploitation.

### La mise à jour des traces de phéromones

La mise à jour des traces de phéromones est effectuée une fois que toutes les fourmis ont modifié leurs partitions. Comme dans la méthode précédente, seule la fourmi qui a trouvé la meilleure partition est autorisée à renforcer les traces de phéromones. Tout d'abord toutes les valeurs de la matrice de phéromone sont diminuées par un processus d'évaporation comme suit :

$$\tau(s,s') = (1-\rho) \cdot \tau(s,s') \quad (6.22 .)$$

Ensuite, la fourmi ayant trouvé la meilleure partition renforce les valeurs des traces de phéromone associées à ses couples  $(s, l)$  selon la règle suivante

$$\tau(s,s') = \tau(s,s') + \rho \cdot \Delta\tau(s,s') \quad (6.23 .)$$



où

$$\Delta\tau(s, s') = \begin{cases} \frac{1}{U(x^{gb})} & \text{if } s, s' \text{ appartiennent à la classe dans } x^{gb} \\ 0 & \text{otherwise} \end{cases} \quad (6.24 .)$$

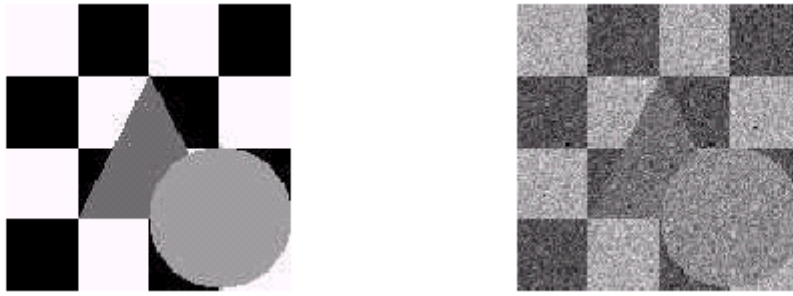
Avec  $x^{gb}$  est la meilleure partition trouvée depuis de début du processus d'étiquetage.

### La diversification

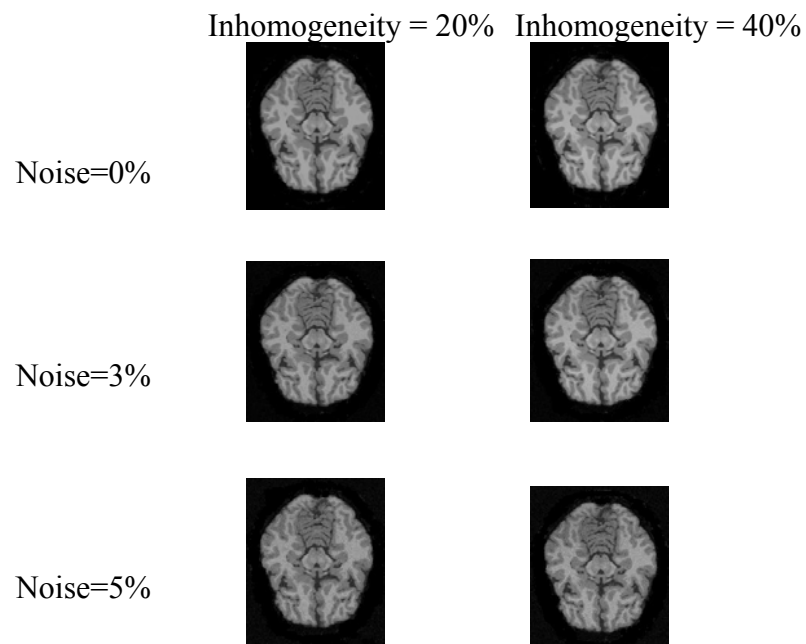
Dans le cas où la meilleure solution  $x^{gb}$  obtenue depuis le début de l'algorithme, n'a pas été améliorée depuis un nombre d'itération ce qui indique une stagnation ou une convergence prématurée, on active un processus de diversification qui force une exploration de nouvelles pistes dans l'espace de recherche. Toutes les valeurs de phéromones sont complètement re-initialisées à la même valeur et de nouvelles partitions sont générées pour  $k - 1$  fournis à l'exception de la keme fourni à qui on associe la meilleure global partition. Le processus de génération des nouvelles partitions est basé sur une matrice de fréquences  $Freq$  ( $Freq(s, s'), s \in S, s' \in N_s$ ) qui représente le nombre de fois où la paire  $(s, s')$  est apparue dans l'ensemble des solutions. Ces solutions ont été fournies par les fourmis lors des itérations précédentes. Cette matrice est utilisée lors de la diversification, pour générer des solutions comportant les affectations qui ont été les moins choisies.

## 6.5 Résultats expérimentaux

Nous présentons dans cette section quelques résultats expérimentaux que nous avons obtenus lors de l'application des approches proposées pour la segmentation d'images par relaxation Markovienne. Dans un premier temps, nous avons testé les algorithmes proposés sur des images synthétiques de nature différente. L'avantage de ce type d'images est que l'on connaît les caractéristiques des classes présentes dans l'image étant donné que la segmentation de référence est parfaitement connue et cela nous permet d'évaluer la performance des algorithmes de segmentation proposés. Le premier exemple des images synthétique, consiste en des objets de formes géométriques différentes (un cercle et un triangle) (voir figure 6.6). l'image est contaminée avec un bruit additif gaussien. Le second exemple concerne le domaine médical et en particulier le contexte de l'IRM cérébrale (Figure 6.7) Nous avons plus précisément utilisé des images « fantômes » disponibles sur le site : <http://www.bic.mni.mcgill.ca/brainweb/>. Ces images sont des images synthétiques réalistes auxquelles différents niveaux de bruits et différents degrés d'inhomogénéité peuvent être ajoutés. A chacune de ces images peut être associée une segmentation « exacte », connue par construction. Pour cette raison, ces images sont utilisées à des fins d'évaluation des algorithmes de segmentation.

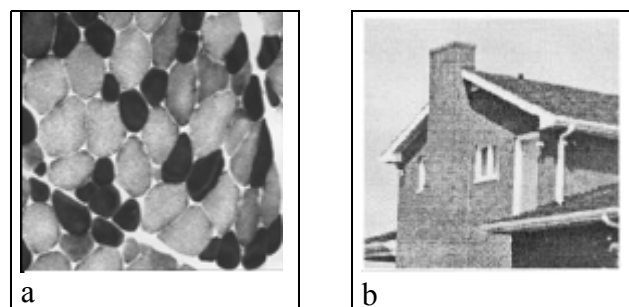


**Figure 6.7. Image synthétique. A) image originale, B) image bruitée SNB=10DB**



**Figure 6.8. Images IRM Fantômes**

Dans un second temps, nous avons utilisé quelques images réelles de la banque d'images du GdR 134. : une image de maison avec 3 classes et une image de muscle représentant une section transversale de fibres musculaires avec 4 classes.



**Figure 6.9. Images réelles: a) image de muscle , b) image de maison**

Afin de mieux situer les résultats obtenus nous les avons comparés avec ceux obtenus avec deux autres méthodes d'optimisation classiques qui sont le Recuit Simulé et les Algorithmes Génétiques.

Dans la validation des algorithmes présentés dans ce chapitre, nous nous sommes attachés au fait que l'objectif de tels algorithmes est de vérifier les deux critères suivants : une convergence vers une solution optimale et la robustesse vis à vis de la présence du bruit. La satisfaction de ces critères nécessite au préalable d'ajuster les paramètres de contrôles suivants :

1. Le nombre  $K$  de fourmi. Nous avons vu d'après nos expérimentations qu'un nombre trop petit de fourmis, réduisait considérablement la qualité de la segmentation construite et qu'un nombre trop grand conduisait à des temps de convergence trop important sans pour autant améliorer la qualité de la partition.
2. Les paramètres  $\rho$  et  $\alpha$  qui déterminent le degré d'évaporation des traces de phéromones et ont une grande influence sur l'exploration de l'espace de solutions. Une valeur élevée pour ces paramètres, réduirait la persistance de la phéromone et donc l'efficacité de l'algorithme : il prendra trop de temps pour trouver de bonnes solutions. Une valeur trop petite conduirait à un comportement déterministe de l'algorithme.
3. Le paramètre  $q_0$  est très important car il permet de moduler le degré d'exploration contre le degré d'exploitation. Réduire la valeur de  $q_0$  équivaut à de nouvelles zones dans l'espace de recherche. Augmenter  $q_0$  revient à choisir de concentrer la recherche dans la zone des meilleures solutions trouvées.
4. Le paramètre  $\gamma$  contrôle le degré de communication entre les fourmis. La recherche est plus efficace quand les fourmis communiquent entre elles, ce qui implique que les quantités de phéromones accumulées ont un grand impact sur le comportement de la fourmi.
5. Le paramètre  $\beta$  contrôle l'importance de l'information heuristique par rapport à la trace de phéromone ;.
6. Le paramètre  $\tau_{\min}$  qui fixe le taux maximum de trace de phéromone ;

7. Le nombre d'itération permet d'étudier le temps de convergence de l'algorithme.

Pour trouver les bonnes valeurs de ses différents paramètres, nous avons effectué plusieurs exécutions de nos algorithmes de segmentation sur les images de test présentés dans les figures 6.6. et 6.7 pour diverses valeurs des paramètres de contrôles. Le tableau présente les valeurs avec lesquelles de bons résultats ont été obtenus.

Paramètres	Valeurs		
	Approche constructive		Approche perturbatrice
	ACS	MMAS	HACSEG
$\tau_{\min}$	/	0.001	/
$q_0$	0.6	0.6	0.6
$\rho$	0.01	0.9	0.01
$\alpha$	0.01	/	/
$\gamma$	/	/	1
$\beta$	/	/	2
K	10	15	10
$N_{\max}$	2500	2500	2500

Tableau 6.1. Valeurs initiales des paramètres de nos algorithmes de segmentation

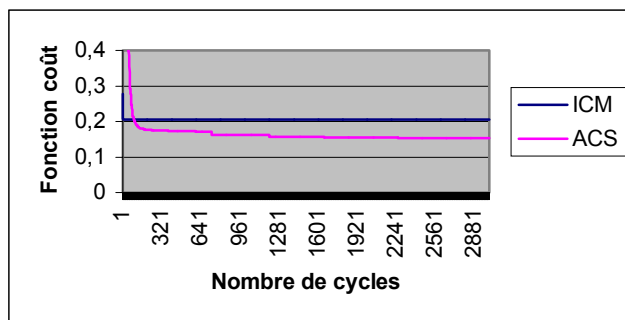
Les paramètres de l'algorithme Recuit Simulé et les Algorithmes Génétiques sont résumés dans le tableau 6.2.

Algorithme	Paramètre	Valeur
SA	$T_0$	2
	$T_m$	0.9
	$N_i$	100
	$N_{\max}$	3000
GA	$N$	30
	$P_c$	0.8
	$P_m$	0.01
	$P_m$	1000

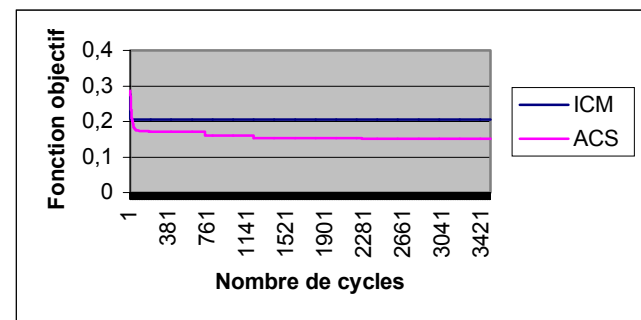
Tableau 6.2.. Paramètres des algorithmes: Recuit Simulé, Algorithme Génétique  $T_0$  : température initiale,  $T_m$  : multiplicateur de température,  $N_i$  : nombre d'itérations à partir desquelles la température est diminuée,  $N_{\max}$  : nombre maximum des itérations,  $N$  : taille de la population,  $P_c$  : probabilité du crossover,  $P_m$  : probabilité de mutation.

### 6.5.1 Etude de la convergence

L'étude de la convergence consiste à étudier le comportement de la fonction objectif durant le processus d'optimisation. Tout d'abord nous avons analysé ce comportement en exécutant l'algorithme de fourmis ACS avec ou sans l'utilisation d'une recherche locale et nous avons comparé les résultats obtenus avec ceux générés par l'algorithme d'optimisation déterministe ICM afin de montrer la capacité des algorithmes de fourmis à converger vers une solution de meilleure qualité. A partir de plusieurs exécution un comportement moyen de chacun des algorithmes a été dégagé. La figure 6.9 illustre pour l'image synthétique représentée dans la figure 6.6. l'évolution de la fonction de coût en fonction du nombre de cycle d'exécution pour chacun des deux algorithmes. Cette figure met en évidence la capacité de l'algorithmes de segmentation basé fourmis à éviter de rester piéger dans un minimum local et donc de trouver une solution de meilleure qualité que celle générer par ICM. D'un autre coté, on constate qu'une meilleure convergence est obtenue quand la recherche locale est utilisée.



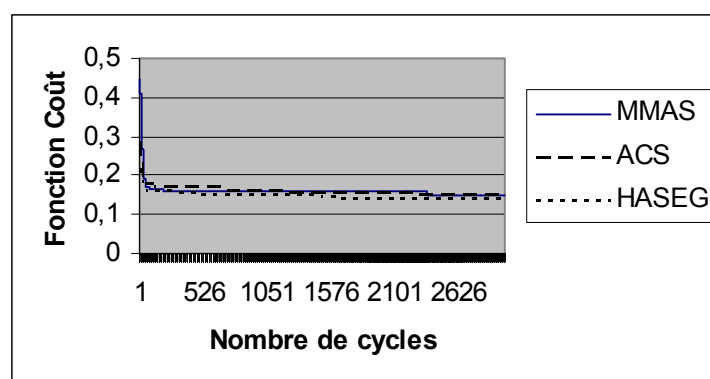
(a)



(b)

**Figure 6.10. Graphe de convergence des algorithmes d'optimisation ACS et ICM : (a) utilisation de ACS sans recherche locale, (b) utilisation de ACS avec recherche locale.**

Dans un second temps, nous avons étudié l'évolution de la fonction de coût en considérant l'approche de constructive de partition avec les deux stratégies de sélections des étiquettes ACS et MMAS et l'approche perturbatrice de partitions HACSEG. Le graphe 6.10. montre que les résultats de ACS et MMAS sont proches mais moins bons que ceux de HACSEG.



**Figure 6.11. Evolution de la qualité de la fonction objectif en utilisation les trois algorithmes de fourmis ACS ,MMAS et HASEG.**

### 6.5.2 Etude de la robustesse

La robustesse de nos algorithmes a été évaluée en calculant pour chacune des classes extraites les valeurs de la mesure de similarité de Jaccard pour les images dont la segmentation de référence est disponible et celles de la mesure de Borsotti pour les images réelles. Afin de mieux situer les résultats obtenus nous les avons comparés avec ceux obtenus avec d'autres méthodes d'optimisation telles que

		N= 0%		N = 3%		N = 5%	
		20%	40%	20%	40%	20%	40%
SA	LCR	0.86	0.87	0.86	0.85	0.82	0.80
	GM	0.91	0.89	0.88	0.88	0.86	0.84
	WM	0.90	0.86	0.88	0.84	0.83	0.81
GA	LCR	0.85	0.89	0.87	0.86	0.82	0.81
	GM	0.96	0.90	0.90	0.88	0.87	0.85
	WM	0.91	0.86	0.88	0.84	0.83	0.82
ACS	LCR	0.88	0.88	0.89	0.87	0.86	0.82
	GM	0.95	0.90	0.91	0.89	0.88	0.87
	WM	0.93	0.88	0.90	0.85	0.84	0.81
MMAS	LCR	0.89	0.87	0.87	0.85	0.86	0.82
	GM	0.94	0.89	0.90	0.89	0.88	0.86
	WM	0.93	0.88	0.89	0.86	0.84	0.82
HACSEG	LCR	0.89	0.89	0.88	0.86	0.87	0.82
	GM	0.95	0.90	0.91	0.90	0.88	0.87
	WM	0.94	0.89	0.91	0.87	0.85	0.83

Tableau 6.3. . Comparaison des performances des algorithmes (1) SA, (2) GA, (3)ACS-MRF, (4) HACSEG pour la segmentation des mages IRM.

Images	(a)	(b)
SA	1.130	0.980
GA	1.128	0.950
MMAS-MRF	1.1278	0.9479
ACS-MRF	1.128	0.9486
HACSEG	1.1272	0.9473

Tableau.6.4. Mesure de *Borsotti & al.* pour les images réelles

D'après les valeurs présentées dans le tableau 6.3. et 6.4., on constate que HACSEG obtient toujours de meilleurs résultats que ACS et MMAS. Cette performance s'explique par le fait que HACSEG utilise une fonction heuristique pour guider la recherche des fourmis et d'une autre part l'utilisation d'une fonction de diversification qui permet de mieux explorer l'espace de solutions.

## 6.6 Conclusion

La segmentation par relaxation markovienne, est un problème difficile pour qui plusieurs métaheuristiques ont été proposé. Dans ce chapitre, nous avons présenté une approche basée sur une colonie d'agents fourmis qui coopèrent ensemble par stigmergie pour la résolution collective du problème de segmentation L'idée est d'utiliser l'intelligence collective des fourmis a été utilisée pour la détection de classes dans des images à niveaux de gris tenant compte des contraintes d'homogénéité et de régularité spatiale. Les algorithmes proposés sont distribués et se basent sur une colonie de fourmis qui construisent/ modifient en parallèle des partitions candidates et communiquent via les traces de phéromones accumulées durant les itérations. La partition optimale émerge à partir de la coopération indirecte des fourmis. Un algorithme de recherche locale est ajouté afin d'améliorer la qualité des solutions trouvées et accélérer la convergence de l'algorithme. Une étape de diversification est introduite afin de diversifier la recherche dans l'espace de solutions.

Chapitre 7  
Conclusion générale



---

## Chapitre 7. Conclusion générale.

---

Ce travail de thèse a pour objectif de proposer quelques contributions pour la segmentation d'images en s'inspirant du comportement collectif et auto-organisé des fourmis. Les fourmis possèdent des comportements auto-organisés, à travers lesquels des interactions simples au niveau local permettent l'émergence d'un comportement global complexe. Les premiers travaux utilisant ces intéressantes caractéristiques ont donné naissance à de nouvelles approches pour la résolution des problèmes d'optimisations combinatoires dans différents domaines d'application.

Nous avons proposé de bénéficier du mécanisme de coopération inconsciente chez les fourmis pour mettre au point de nouvelles approches de segmentation d'images d'inspiration biomimétiques. La segmentation optimale émerge progressivement des interactions indirectes entre les fourmis selon un processus stigmergique. Ces fourmis sont très simples et ont une perception limitée de leur environnement ainsi qu'un mécanisme de sélection d'actions probabiliste en fonction de l'état de leur environnement. Nos contributions prennent deux formes :

Dans un premier temps, nous avons étudié et adapté la capacité des fourmis à rassembler et à trier les éléments de leur couvain, pour développer un nouvel algorithme de classification non supervisée a été proposé. L'objectif est de partitionner les pixels d'une image en des classes pertinentes, sans disposer d'une partition initiale ni d'information sur le nombre d'exacte de classes présentes dans l'image. Pour cela, nous utilisons une colonie de fourmis artificielles autonomes qui se déplacent d'une manière asynchrone sur un tableau de cases représentant leur environnement. Les pixels à classer sont placés initialement sur ce tableau de telle sorte qu'une case ne contienne qu'un seul pixel à la fois. Les fourmis déplacent les pixels d'une case à une autre en se basant sur une fonction de similarité entre les pixels d'une même classe. Grâce à des règles de comportement local très simples, et de la coopération inconsciente entre les fourmis, la partition optimale émerge.

Dans un second temps, nous nous sommes inspirés du comportement de fourragement chez les fourmis, modélisé par la métaheuristique OCF (ou ACO), pour proposer une nouvelle approche de segmentation par relaxation markovienne. Notre motivation est que la segmentation statistique étant définie comme un problème NP\_difficile, est ainsi une très bonne candidate pour les approches basées métaheuristicques y compris OCF. Les fourmis construisent des segmentations potentielles d'une manière incrémentale s'appuyant sur les informations collectées collectivement. Elles communiquent indirectement via des modifications de l'environnement par l'intermédiaire des traces de phéromones. L'idée

fondamentale de cette approche une recherche parallèle par les fourmis d'un étiquetage correcte des pixels de l'image. Cette recherche est basée sur une structure de mémoire dynamique (trace de phéromones) incorporant des informations sur l'efficacité des résultats précédemment.

Dans cette optique, trois algorithmes basés fourmis ont été développés. Les deux premiers algorithmes utilisent une approche constructive pour générer des segmentations potentielles. Sans aucune partition de départ, les fourmis construisent selon un processus incrémental des segmentations en utilisant des règles de sélection des étiquettes selon deux stratégies différentes : ACS et MMAS. Le troisième algorithme se distingue des aux premiers par le fait que les fourmis ne construisent pas totalement de nouvelles partitions, mais au contraire modifient celles déjà existantes. La procédure de modification se base sur l'échange d'étiquettes des pixels en fonction de leur voisinage immédiat. En plus une fonction de diversification a été introduite afin de mieux guider les fourmis lors de l'exploration de l'espace de solutions. La validation de ces algorithmes sur un jeu de test a permis de montrer leur robustesse vis à vis du bruit et leur capacité à converger vers une solution de bonne qualité.

La validation des ces algorithmes basés fourmis sur des images synthétiques et réelles a permis de dégager leurs intérêts en comparants les résultats obtenus avec ceux de méthodes alternatives de références dans la segmentation d'images.

## **Perspectives**

Les perspectives de cette contribution sont nombreuses, et concernent en particulier deux points de vue.

En premier lieu, il s'agit en premier de faire une étude théorique pour le choix des valeurs des paramètres des algorithmes afin de mieux contrôler la qualité des résultats obtenus. En second, d'étudier la possibilité d'une hybridation des algorithmes de fourmis avec d'autres algorithmes d'optimisation combinatoires tels que les algorithmes génétiques, les algorithmes d'optimisation par essaim de particules. Enfin, d'exploiter le parallélisme implicite des algorithmes basés fourmis afin d'améliorer les temps de convergence. Par exemple, on pourra associer aux fourmis des processeurs indépendants qui communiqueront entre eux à travers une mémoire commune qui contiendra les valeurs des phéromones.

En second lieu, la faisabilité des algorithmes proposés, leur simplicité de mise en œuvre ainsi que la qualité des solutions obtenues comparés aux méthodes classique, nous permettent de

penser à généraliser ce travail en modélisant d'autres comportements collectifs d'insectes sociaux pour proposer d'autres méthodes de segmentation en utilisant le concept d'auto-organisation et d'émergence.

## Annexe A

### Modélisation Markovienne pour la segmentation d'images

#### A.1 Introduction

Les modélisations Markoviennes ont connu ces dernières années un vif succès, notamment pour les problèmes de segmentation en régions et en contours. En effet, depuis les travaux de Geman et Geman [Geman, 1984], les champs de Markov ont été beaucoup utilisés pour la formalisation théorique des techniques de segmentation précédemment utilisées. Ils offrent un cadre mathématique cohérent pour l'extraction des régions de l'image, en utilisant conjointement les données de l'image et les contraintes contextuelles ce qui permet de fournir de bons résultats de segmentation. La segmentation associée aux champs de Markov se ramène comme on verra à un problème d'optimisation d'un étiquetage. Le but est alors de trouver le minimum d'une fonction représentant l'erreur de segmentation commise. Cette fonction appelée également « énergie » est constituée généralement de deux termes : un relatif au vecteur de caractéristiques de l'image, et l'autre relatif aux interactions entre les étiquettes des pixels. La minimisation globale de la fonction d'énergie est un problème NP-complet, plusieurs algorithmes d'optimisation ont été proposés pour le résoudre.

#### A.2 Champs de Markov et Distribution de Gibbs

Une image est représentée par une grille  $S$  rectangulaire finie bidimensionnelle de taille  $m \times n$  représentant sa résolution. Chacun des pixels est appelé site et est représenté par une valeur numérique qui représente le niveau de gris. Les sites d'une grille sont généralement désignés par leur numéro d'ordre :

$$S = (s_i, 1 \leq i \leq N).$$

A chaque site  $s$  est associée une variable aléatoire  $Y_s$ , dont la valeur  $y_s$  appartient à un ensemble  $\Omega = \{0, \dots, 255\}$ . L'image est alors considérée comme la réalisation d'un vecteur aléatoire  $Y = \{Y_s, s \in S\}$  représentant le champ des observations. La segmentation markovienne a pour objectif d'estimer le champ des étiquettes  $E = \{E_s, s \in S\}$ . Une réalisation des champs  $Y$  et  $E$  sont notées respectivement par  $y = \{y_s, s \in S\}$  et  $e = \{e_s, s \in S\}$ .

On appellera voisinage du site  $s$ , noté  $V_s$ ; l'ensemble des sites voisins de  $s$ . Le système de voisinage défini sur  $S$  est  $V = \{V_s, s \in S\}$  tel que :

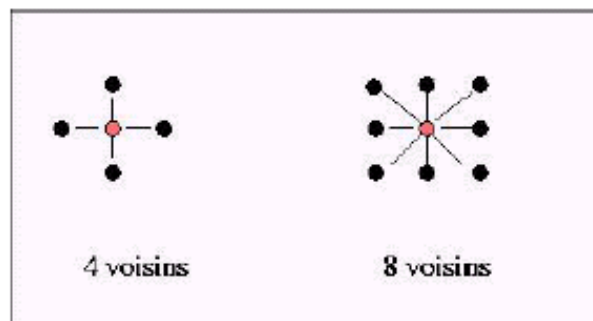
**Définition 1 : système de voisinage**

$V = \{V_s, s \in S\}$  est un système de voisinage défini sur  $S$  si :

- $\forall s \in S, s \notin V_s$
- $\forall (s, t) \in S, s \in V_t \Leftrightarrow t \in V_s$

La première condition signifie que le voisinage d'un pixel est un sous\_ensemble du support  $S$  qui ne contient pas le pixel lui même. La seconde condition est une réciprocity : si un pixel donné appartient au voisinage d'un second pixel, alors ce dernier doit aussi appartenir au voisinage du premier (on dit que les deux pixels sont voisins). De cette définition découle l'existence de plusieurs systèmes de voisinage d'ordre différent.

La figure (Figure A.1) illustre les deux voisinages les plus utilisés. Celui dit du premier ordre « système de voisinage de la 4-connexité » est celui pour lequel chaque site a quatre voisins qui sont ses quatre plus proches voisins. De même, celui du second ordre « le système de voisinage de la 8-connexité » est celui pour lequel chaque  $V_s$  est constitué des huit plus proches voisins de  $s$ .



**Figure 5.1. Système de voisinage 4-connexité et 8- connexité**

Les éléments de  $N_s(S)$  définissent les sites voisins de  $s$ . Une configuration de voisins de  $s$  est appelée clique. On note  $C$  l'ensemble des cliques de  $S$ . Une clique est dite d'ordre  $n$  si elle contient  $n$  éléments.

**Définition2 : Cliques**

Un sous-ensemble  $c$  de  $S$  est appelé une clique relative au système de voisinage  $V$  si

- $c$  est un singleton ou
- si tous les sites de  $c$  sont tous voisins deux à deux.

Généralement les cliques d'ordre deux sont les plus utilisées afin de réduire les temps de calcul.

**Définition2 : Champ de Markov**

Le vecteur aléatoire  $X$  est un champ de Markov relativement au système de voisinage  $N(S)$  si :

- $\forall x. P(E=e) > 0$  (contrainte de positivité)
- $\forall s \in S, P(E_s = e_s / E_r = e_r, r \in S - \{s\}) = P(E_s = e_s / E_r = e_r, r \in N_s(S))$

La première hypothèse signifie qu'aucune configuration n'est interdite a priori. La seconde propriété également appelée propriété de "markoviennité", exprime que le comportement de la variable aléatoire portée par un site est entièrement déterminé par les réalisations des variables aléatoires voisines. Ainsi dans la pratique, l'introduction des champs Markoviens permet la modélisation d'ensembles de variables aléatoires dont les interactions mutuelles résultent uniquement de la combinaison d'interactions locales. C'est cette localité des calculs qui rend les champs de Markov utilisables en traitement d'images.

Besag *et al.* [Besag, 1974] a rendu populaire la relation entre champs markoviens et distribution de Gibbs, initialement démontrée par le théorème de Hammersley-Clifford (1971). Ce théorème permet d'accéder à la probabilité globale d'une réalisation : si aucune réalisation du champ de Markov n'est de probabilité nulle, la probabilité d'une réalisation de ce champ suit une loi de probabilité de Gibbs.

### **Théorème de Hammersley-Clifford**

Un champ aléatoire  $E$  est un champ markovien associé au système de voisinage  $V$  si et seulement si sa distribution de probabilité  $P(E=e)$  est une distribution de Gibbs définie par :

$$P(E=e) = \frac{1}{Z} \exp\{-U(e)\}$$

avec  $Z$  est une constante de normalisation appelée fonction de partition de Gibbs et est donnée par :

$$Z = \sum_e \exp\{-U(e)\}$$

et la fonction d'énergie  $U(e)$  est une fonction d'énergie définie localement comme la somme des potentiels d'interactions  $V_c$  qui modélisent les interactions entre sites au sein des cliques:

$$U(e) = \sum_{c \in C} V_c(e)$$

D'un point de vue pratique, la spécification des propriétés locales d'un champ de Markov via la définition des fonctions potentiels des cliques, traduisant un certain a priori ou une certaine préférence concernant les interactions locales, se traduit de façon équivalente par la caractérisation globale du modèle probabiliste. De plus, on vérifie aisément que les distributions conditionnelles locales s'expriment à l'aide des potentiels associés aux cliques contenant le site  $s$

### **A.3 Segmentation d'images par les champs de Markov**

Cette approche permet de résoudre le problème de l'estimation d'un processus label à partir des observations dont on dispose. Elle relie un champ  $X$  de variables cachées, à estimer, à un champ  $Y$  de variables observées sur une grille de discrétisation  $S$ . Plusieurs techniques sont utilisables. La plus courante est l'approche bayésienne qui consiste à construire une loi a posteriori à partir d'une loi a priori sur le champ des labels, sous l'hypothèse d'indépendance des observations en chaque site.

### Théorème de BAYES

En terme d'analyse bayésienne, la probabilité a posteriori peut être exprimée comme suit :

$$P(E=e|Y=y) = \frac{P(E=e)P(Y=y|E=e)}{P(Y=y)}$$

Comme la réalisation  $y$  est connue,  $P(Y=y)$  est une constante vis à vis du problème, la distribution a posteriori  $P(E=e|Y=y)$  peut s'exprimer simplement en fonction de la distribution conditionnelle de  $Y$  sachant  $E$ , c'est-à-dire  $P(Y=y|E=e)$  la vraisemblance des observations et de la distribution a priori  $P(E=e)$  des étiquettes à estimer. Comme les champs  $X$  et  $Y$  sont supposés markoviens. On a alors :

$$P(E=e|Y=y) = \frac{\exp(-U_1(Y=y|E=e))}{Z_1}$$

$$P(E=e) = \frac{\exp(-U_2(E=e))}{Z}$$

La probabilité a postérieure peut alors s'écrire selon la forme suivante !

$$P(E=e|Y=y) = \frac{\exp(-U_1(Y=y|E=e) - U_2(E=e))}{Z_3}$$

où  $Z_1, Z_2, Z_3$  sont des constantes de normalisation.

On cherche à trouver un estimateur bayésien optimal  $\hat{e}$  parmi l'ensemble des configurations possibles. Plusieurs estimateurs existent dans la littérature, nous détaillons dans la suite trois principales estimateurs.

#### A.4 Estimation de la configuration des étiquettes

Un estimateur  $e^*$  est une application qui associe une configuration des étiquettes à une réalisation  $y$  du champ des observations. Afin d'obtenir la configuration optimale, nous cherchons à minimiser une fonction de coût  $C(e, e^*)$  pondérée par la probabilité d'apparition du champ  $e$  conditionnellement aux observations, soit

$$\hat{e} = \operatorname{argmin}_e \sum_e C(\hat{e}, e) P(E=e|Y=y)$$

Les estimateurs les plus généralement utilisés sont :



- a. L'estimateur MAP pour lequel la fonction de coût  $LMAP=C(\hat{e},e)=1-\delta(\hat{e},e)$  où  $\delta$  représente le symbole de Kronecker. Cette fonction de coût pénalise de la même façon toutes les configurations de  $e$ , différentes de la solution recherchée  $e$ . Autrement dit, elle pénalise de façon identique une erreur sur un site et une erreur sur plusieurs sites. L'estimateur du MAP fournit toutes les configurations les plus probables conditionnellement à la réalisation du champ des observations. On a

$$\hat{e}^{MAP}=\arg \max_{w \in \Omega} P(E=e/Y=y)$$

$$\hat{e}^{MAP}=\arg \min_e U_1(E=e/Y=y)+U_2(E=e)$$

- b. Mode des Marginales a Posteriori MPM estimateur permet de remédier à l'uniformité de l'estimateur du MAP, en utilisant une fonction de coût qui est proportionnel un nombre de sites sur lesquelles les deux configurations sont identiques.  $LMPM=C(\hat{e},e)=\sum_{s \in S} 1-\delta(\hat{e}_s,e_s)$ . L'estimateur du MPM minimise en chaque site la distribution marginale locale a posteriori :

$$\hat{e}_s^{MPM}=\arg \max_{s \in \Lambda_s} P(e_s=e'_s/Y=y)$$

- c. Champ moyen MF La fonction de coût utilisée a une forme quadratique :  $LMF=C(\hat{e},e)=\sum_{s \in S} (\hat{e}_s - e'_s)^2$ . L'estimateur est alors défini par :

$$\hat{e}_s^{MF}=E[e_s/Y=y]$$

L'estimation de la configuration optimale est un problème d'optimisation doublement difficile. car d'une part la fonction d'énergie à minimiser n'est pas convexe et donc possède des minima locaux, et d'une autre part l'espace des configurations est trop grand : pour une image de 512x512 pixels codés sur 256 niveaux de gris, l'espace des configurations =  $256^{512 \times 512}$ . Il n'est pas possible de tester chacune des configurations. Pour résoudre de problème il est alors nécessaire d'utiliser des techniques d'optimisation sophistiquées. Les algorithmes de relaxation stochastique et déterministe apporte une solution à ce problème d'optimisation.

## A.5 Algorithmes de Minimisation

Afin de résoudre ce problème d'optimisation classique, plusieurs méthodes de relaxation ont été proposées dans la littérature. On distingue entre les algorithmes de relaxation déterministe qui convergent vers une solution sous optimale en peu de temps et entre les algorithmes de relaxation stochastique qui fournissent une solution optimale au prix d'un temps de convergence important.

### A.5.1 Méthodes de relaxation stochastiques

Les algorithmes de relaxation stochastiques permettent théoriquement d'atteindre asymptotiquement un minimum global de la fonction d'énergie indépendamment de la configuration initiale.

L'algorithme le plus généralement utilisé est le recuit simulé. Le recuit simulé « simulated annealing » est un algorithme stochastique introduit par Kirkpatrick et al [Kirkpatrick, 1983] et appliqué à la segmentation d'images par [Geman, 1984]. C'est une méthode d'optimisation dérivée de la méthode de Monté Carlo qui est utilisée pour la simulation des systèmes thermodynamiques. Le principe de base repose sur le principe du modèle métallurgique qui consiste à faire chauffer le métal à une température assez élevée et puis à la faire décroître d'une manière graduelle jusqu'à ce que le métal prenne la forme désirée (refroidissement). Dans le cas de la segmentation d'images par les champs de Markov, le recuit simulé permet de minimiser la fonction d'énergie globale en acceptant des remontées d'énergie proportionnelles à la température, lui permettant ainsi de sortir des puits d'énergie assez profonds à une température élevée. Inversement, plus la température est faible, moins ces sauts d'énergie sont tolérés et par conséquent le système se stabilise.

L'objectif du recuit simulé est de trouver une solution optimale à la fonction objective. Partant d'une partition  $S$  donnée ayant une qualité  $C(s)$ , on modifie cette configuration en modifiant les étiquettes des pixels. La nouvelle partition  $S'$  est toujours acceptée si sa qualité  $C(S')$  est meilleure ; dans le cas contraire elle sera acceptée avec une probabilité fonction de la qualité et de la température. Avec ce procédé, des partitions moins bonnes seront acceptées, ce qui permet d'éviter d'être piégé dans des optima locaux.

### A.5.2 Méthodes de relaxation déterministes

Pour pallier les problèmes de lenteur de convergence, des algorithmes de relaxation déterministes ont été dérivés des méthodes de relaxation stochastique. Dans ces méthodes, seules les transitions de la configuration courante vers une configuration pour laquelle la fonction d'énergie décroît sont autorisées. Aucune augmentation de la valeur de la fonction d'énergie, et donc aucun changement de label, n'est admise par l'algorithme qui convergera

plus vite mais vers un minimum stable correspondant simplement au minimum le plus proche de la configuration initiale.

L'algorithme le plus connu est celui de ICM Cet algorithme a été introduit par Besag [Besag, 1974] et est largement utilisé dans tous les problèmes impliquant les champs de Markov introduit à cause de la simplicité de sa mise en œuvre. C'est un algorithme déterministe qui cherche à minimiser en chaque site l'énergie locale définie ne lui. Pour cela il faudra calculer cette énergie pour toutes les étiquettes possibles. Cet algorithme converge rapidement vers un minimum local mais est très sensible à la configuration de départ ainsi que de la stratégie de balayage des sites.

# Annexe B

## Publication de l'auteur

### Revue Internationale

**Ouadfel Salima**, Batouche Mohamed. MRF-based image segmentation using Ant Colony System. In *Electronic Letters on Computer Vision and Image Analysis* . ISSN 1577-5097. September 2003 - Volume 2, Number 1 pp 12-24.

**Ouadfel Salima**, Batouche Mohamed. A Hybrid MAX-MIN Ant System to MRF-based Image Segmentation. In a Special Issue of *Wseas Transactions in Systems* Issue 3, Volume 2, ISSN 1109-2777. pp734-741. Juillet 2003.

### Publications Internationales

**Ouadfel Salima**, Batouche Mohamed Ant Colonies For MRF-Based Image Segmentation, Setit 2005, 3rd International Conference on Sciences of Electronic, Technologies of Information and Telecommunications, Tunisie, 27-31 march, 2005.

**Ouadfel Salima**, Batouche Mohamed MRF-based image segmentation using autonomous agents, Setit 2004, 2nd International Conference on Sciences of Electronic, Technologies of Information and Telecommunications, Tunisie, 14-17 march, 2004.

**Ouadfel Salima**, Batouche Mohamed Multiresolution Image Segmentation using Ant Colony System and Markov Random Field . *icisp 2003* Juin 25-27 Agadir, Maroc .

**Ouadfel Salima**, Batouche Mohamed . Ant Colony System with local search for Markov Random Field Image Segmentation. *ICIP (1) 2003*: Septembre 14-18 pp. 133-136. Barcelone Espagne.

**Ouadfel Salima**, Batouche Mohamed Unsupervised image segmentation using a colony

of cooperating ants. 2nd Workshop on Biologically Motivated Computer Vision Ants'2002". Tübingen 2002. LNCS. ©Springer Verlag..

**Ouadfel Salima**, Batouche Mohamed Garbay Catherine. Une méthode d'optimisation globale pour la segmentation d'images » Le 6ème colloque Africain sur la recherche en informatique. CARI-2002. 14 – 17 octobre 2002. Yaoundé, Cameroun.

### **Publication Nationales**

**Ouadfel Salima**, Batouche Mohamed Garbay Catherine. Optimisation par les colonies de fourmis pour la segmentation d'images. 7<sup>ème</sup> Conférence Maghrébine sur la recherche en Informatique. Du 5-8 Mai 2002. Annaba. Pp : 227-235.

**Ouadfel Salima**, Hachouf Fella, Batouche Mohamed. Segmentation et Analyse des images cytologiques Acte de la conférence sur la vision Artificielle. CVA 2000. Page .126-133 TIZI-Ouzou du 18-20 Novembre 2000

# Bibliographie

## Bibliographie

- [Abraham, 2003] A. Abraham, et V. Ramos, Web Usage Mining Using Artificial Ant Colony Clustering and Genetic Programming. Proc. of the Congress on Evolutionary Computation (CEC 2003), Canberra, pp. 1384- 1391, IEEE Press.2003
- [Abrantes, 1996] A.J. Abrantes, et J. S. Marques, A Class of Constrained Clustering Algorithms for Object Boundary Extraction. IEEE Transactions on Image Processing, Vol. 5, no.11. 1996.
- [Adams, 1994] R. Adams, et L. Bischof, Seeded region growing, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, no. 6, 1994.
- [Ali, 1997] S-M.Ali, et R.M. Zimmer, The question concerning emergence, in CASYS'97, Abstract Book, First International Conference on Computing Anticipatory Systems, CHAOS asdl, 1997
- [Andrey, 1998] P. Andrey, et P. Tarroux,. Unsupervised segmentation of Markov random field modeled textured images using selectionist relaxation, IEEE Transactions on Pattern Analysis and Machine Intelligence, 3(20): 252–262, 1998.
- [Atlan, 2000] H. Atlan, *La finalité*, Hors série Science&Avenir, 2000.
- [Aupetit, 2003] S. Aupetit, V. Bordeau, N. Monmarché, M. Slimane and G. Venturini Interactive Evolution of Ant Paintings. IEEE Congress on Evolutionary Computation. 2003.
- [Azzag, 2004] H. Azzag, C. Guinot et G. Venturini, . AntTree: web document clustering using artificial ants. Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 04), p. 480-484, IOS Press, Valencia, Spain 2004.
- [Babguchi, 1990] N. Babaguchi, et Yamada, K. K. Kise, and T. Tezuka, Connectionist Model Binarization,” Proceedings of the 10th International Conference on Pattern Recognition, vol. 1, pp. 51-56, Atlantic City, New Jersey, USA, June 1990.
- [Babu, 1993] G.P. Babu et M.N. Murty. A near-optimal initial seed value selection in k-means algorithm using a genetic algorithm. 14:763–769, 1993.
- [Ball, 1967] G.H. Ball and D.J. Hall, A clustering technique for summarizing multi-variate data, Behavioral Science, Vol.12, pp. 153-155, 1967
- [Ballard, 1982] D. H. Ballard and C. M. Brown, Computer Vision, Prentice-Hall HJ, 1982.
- [Bandyopadhyay, 2002] S. Bandyopadhyay et U. Maulik Genetic clustering for automatic evolution of clusters and application to image classification, Pattern Recognition, vol.35, no. 6, pp. 1197-1208, 2002.
- [Baradez, 2004] M-O Baradez, C. P. McGuckin, N. F. R. Pettengell, et A.Hoppe. Thresholding based on linear diffusion for feature segmentation. Pattern Recognition, 37 (6) : 1131–1148, 2004.
- [Beaulieu 1989] J.M. Beaulieu and M. Goldberg, Hierarchy in picture segmentation: a stepwise optimisation approach, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, no. 2, pp. 150-163, 1989
- [Beaulieu, 1989] J.M. Beaulieu and M. Goldberg, Hierarchy in picture segmentation: a stepwise optimisation approach, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, no. 2, pp. 150-163, 1989.

- [Beckers, 1994] R. Beckers, O. Holland, and J. Deneubourg, From local actions to global tasks : stigmergy and collective robotics . In Proceedings of Artificial Life 4, pages 181–189. MIT Press. 1994
- [Bélanger, 1998] H. Bélanger, Réseau de Kohonen pour la détection des contours d'objets dans une image à niveau de gris . Maîtrise en technologie des systèmes M. Ing. 1998
- [Bellet, 1998] 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.
- [Bertolino, 1995] P. Bertolino. Contribution des pyramides irrégulières en segmentation d'images multiresolution. Thèse de doctorat. Institut National Polytechnique de Grenoble. France. Novembre 1995.
- [Besag, 1974] J. E. Besag. Spatial interaction and the statistical analysis of lattice system. Journal of the royal statistical society, Series B, 36 :192–236, 1974.
- [Beveridge, 1989] J.R. Beveridge, J.G. Riffith. R. R. Kohler A.R. Hanson et E. M. Riseman Segmenting images using localized histograms and region merging IJCV 2 pp. 311-347, Janvier, 1989
- [Bezdeck, 1981] J. C. Bezdeck. Pattern recognition with fuzzy objective function algorithms. Plenum Press Ed., New York, 1981.
- [Bhandarkar, 1999] S. M. Bhandarkar et H. Zhang, Image Segmentation Using Evolutionary Computation, IEEE Trans. On Evolutionary Comp., Vol. 3, No. 1, pp. 1-21, April, 1999
- [Bhanu, 1995] B. Bhanu, S. Lee, and J. Ming, Adaptive image segmentation using a genetic algorithm. IEEE Trans. Systems Man Cybernet. 25 (1995), pp. 1543–1567.
- [Bonabeau, 1994] E. Bonabeau et G. Theraulaz éditeurs. Intelligence Collective. Collection Systèmes Complexes. Hermès, Paris, France, novembre 1994.
- [Bonabeau, 1997] E. Bonabeau et G. Theraulaz G., Auto-organisation et comportements collectifs : la modélisation des sociétés d'insectes, Auto-organisation et comportement, Editions Hermès, 1997.
- [Bonabeau, 1999] E. Bonabeau, M. Dorigo, G. Theraulaz, Swarm Intelligence : From Natural to Artificial Systems, NEW York, Oxford University Press, 1999.
- [Bonabeau, 2000] E. Bonabeau, G. Theraulaz, L'intelligence en essaim, POUR LA SCIENCE, 282 (3): pp. 66-73, N° 271 mai 2000.
- [Bonnin, 1991] P. Bonnin, *Méthode systématique de conception et réalisation d'applications en vision par ordinateur*, Thèse de Doctorat, Décembre 1991.
- [Borsotti, 1998] M. Borsotti, P. Campadelli, et R. Schettini, Quantative evaluation of color image segmentation results , Patt. Rec Lett., vol.19, (1998)741-747.
- [Boucher, 1999] A. Boucher. *Une approche décentralisée et adaptative de la gestion d'informations en vision*. Thèse de doctorat de l'université Joseph Fourier Grenoble I. 15 Février 1999.
- [Bouloudani, 2002] N. Bouloudani, P. Lambert, G. Mauris G., *Color Image Processing Control Using Fuzzy Performance Indicators*, IEEE Int. Symp. on Virtual and Intelligent Measurements Systems (VIMS'2002), Anchorage, USA, May 2002, pp. 79-83.



- [Bouman, 1994] C.A Bouman, et M. Shapiro, Multiscale random field model for Bayesian image segmentation. IEEE Transaction . Img. Proc., 3(2): 162-177, March 1994.
- [Bourjot, 1999] 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.
- [Bourjot, 2001] 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.
- [Brice, 1970] C. R. Brice and C. L. Fennema. Scene analysis using regions. Artificial Intelligence1 :205-226, 1970
- [Bryant, 1979] J. Bryant On the clustering of multidimensional pictorial data. In Pattern Recognition, number 11, pages 115-125, 1979.
- [Bullnheimer, 1997] B. Bullnheimer, R.Hartl, et C. Strauss, Applying the Ant System to the Vehicle Routing Problems, 2nd Metaheuristic International Conference (MIC-97), Sophia-Antipolis, France.1997;
- [Bureau, 2001] A. Bureau, C. Garbay, M. Dodjat . Coopération entre deux populations d'agents pour la segmentation. ORASIS'2001 – p.281-290 (2001);
- [Burt, 1981] P.J. Burt, T. H. Hong, and A. Rosenfeld. Segmentation and estimation of image region properties through cooperative hierarchical computation. IEEE Trans. Systems Man and Cybernetics, 11: 802-809,1981.
- [Camazine, 2002] S. Camazine, J-L. Deneubourg , N.R. Franks , J. Sneyd , G. Theraulaz et E. Bonabeau?Self-Organization in Biological Systems, Princeton University Press, 2002.
- [Canny, 1986] J. Canny, A computational approach to edge detection, IEEE Trans. On Pattern Analysis and Machine Intelligence, vol. 8, n°6, p. 679-698, novembre 1986.
- [Carden, 2002] T.Carden, Image Processing with Artificiel Life, 2002.
- [Castan, 1989] S. .Castan, J. Zhao and J. Shen. Une famille de détecteurs de contours basée sur filtre exponentiel.optimal. In AFCET-RFIA, 1989
- [Chabrier, 2004] S. Chabrier, C. Rosenberger, H. Laurent, B. Emile et P. Marché : Evaluating the segmentation result of a gray-level image, European Signal Processing Conference (EUSIPCO), pages 953-956, Vienne, 2004.
- [Chang, 1994] M. Chang, I. Sezan. I et M. Tekalp. "Adaptative Bayesian Segmentation of Color Images". Journal of Electronic Imaging, vol. 3, no 4, pp. 404-414, 1994.
- [Chardin, 2000] A .Chardin, *Modèles énergétiques hiérarchiques pour la résolution des problèmes inverses en analyse d'images Application à la télédétection*. Thèse de Doctorat de l'Université de Rennes1 . 2000.
- [Chassery, 1991a] J.M. Chassery , Montonvert A., Géométrie discrète en analyse d'image",
- [Chassery, 1991b] J.M. Chassery et M. Melkemi. Diagramme de Voronoï appliqué à la segmentation d'images et à la détection d'événements en imagerie multi-sources. Traitement du signal, 8(3) :155–164, 1991.
- [Chellappa, 1993] R. Chellappa, et A.K Jain, Editors, 1993. Markov Random Fields: Theory and Application Academic Press, San Diego.

- [Chen, 1978] P.C Chen. et T.Pavildis. Segmentation by texture using a co-occurrence matrix and a split and merge algorithm. In ICPR78. PP 565-569. 1978
- [Chen, 2004] C-Y Chen,. et F Ye., Particle Swarm Optimization Algorithm and Its Application to Clustering Analysis, IEEE ICNSC 2004, Taipei, Taiwan, R.O.C., pp. 789-794. 2004.
- [Chen, 2005] C-Y Chen,. et F Ye, Alternative KPSO-Clustering Algorithm”, Tamang Journal of Science and Engineering, Vol. 8, No 2, pp. 165\_174 (2005).
- [Cheng, 2001] H. Cheng, X. Jaing, Y. Sun and J. Wang. Color Image Segmentation: Advances & Prospects. Pattern Recognition, vol.34, pp. 2259-2281, 2001.
- [Cho, 1997] K. Cho et P. Meer. Image Segmentation from Consensus Information. ComputerVision and Image Understanding (CVIU), 68 : 72–89, 1997.
- [Chow, 1972] C.K Chow et T. Kaneko, Boundary detection of radiographic images by a thresholding method, Frontiers of Pattern Recognition (S. Watanabe, Ed.), pp. 61-82, Academic Press, New York, 1972.
- [Chun, 1996] D.N. Chun, et H.S. Yang. Robust Image Segmentation Using Genetic Algorithm with a Fuzzy Measure, Pattern Recognition vol. 29, no. 7, pp. 1195-1211. 1996.
- [Clerc, 2002] M. Clerc, L'optimisation par essaim particulaire : principes, modèles et usage. RSTI-TSI, tome 21, p. 941-964, 2002.
- [Cocquerez, 1995a] J.P.Cocquerez and S. Philipp. Analyse d'images , filtrage et segmentation. Masson, Paris, 1995.
- [Cocquerez, 1995b] J.P. Cocquerez et S. Philipp, Comparaison de méthodes de segmentation d'images, conf. Invité, 15<sup>ème</sup> GRETSI, Juan-les Pins, sept. 1995.
- [Colormi, 1991] A. Colormi, M. Dorigo, V. Maniezzo Distributed optimisation by ant colonies, In Proceeding of ECAL-91, Paris, France 134-142, 1991.
- [Colormi, 1992] A. Colormi, M. Dorigo, V. Maniezzo. Distributed optimization by ant colonies, in Toward a practice of autonomous systems : proceedings of the first European Conference on Artificial Life (ECAL 91)”, MIT Press, 1992, p. 134-142.
- [Cortes, 1989] C. Cortes et A. Hertz, A Network System for Image Segmentation, Proceedings of the International Joint Conference on Neural Networks, vol. 1, pp. 121-125, Washington DC, USA, June 1989.
- [Costa, 1997] D. Costa. et A. Hertz., Ants Can Color Graphs, Journal of the Operational Research Society, (48): p. 295-305. 1997
- [Darwin, 1859] C. Darwin, On the origin of species (De l'origine des espèces). John Murrey, London, 1859.
- [Davis, 1991] L . Davis, The genetic algorithm handbook, chapter 17. Ed. New York : Van Nostrand Reinhold, 1991.
- [De Jong, 1975] De Jong., K. *An analysis of the behavior of a class of genetic adaptive systems*. PhD thesis, University of Michigan, 1975.
- [Deneubourg, 1987] J.L Deneubourg, S. Goss, J. Pasteels D. Fresneau et J.P. Lachaud J. P., Self-Organization Mechanisms in Ants Societies (II): Learning in Foraging and Division of Labour, in From Individual to Collective Behaviour in Social Insects, Birkhäuser Verlag, Basel, pp. 177-196, 1987.

- [Deneubourg, 1990] J.L Deneubourg, S Aron, S. Goss, et J.M Pasteels, The self-organizing exploratory pattern of the argentine ant. Dans *Journal on insect Behavior*, 3: 159-168, 1990.
- [Deneubourg, 1991] J.L Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, et L. Chretien,. The dynamic of collective sorting robot-like ants and ant-like robots, in J. A. Meyer and S. W. Wilson (eds), *SAB 90 - 1st Conference On Simulation of Adaptive Behavior: From Animals to Animats*, MIT Press. 1991.
- [Deriche, 1987] R. Deriche, Using Canny's criteria to derive a recursively implemented optimal edge detector, *International Journal of Computer Vision*, p. 167-187, 1987.
- [Deriche, 1990] R. Deriche. Fast algorithms for low level vision. *IEEE Trans. Pattern Analysis and Machine Intelligence* 12(1)78-87.1990.
- [Derin, 1987] H. Derin, et H. Elliot, Modeling and segmentation of noisy and textured images using gibbs random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):39-55. 1987.
- [Descombes, 1999] X. Descombes, R. Morris, J. Zerubia et M. Berthod. Estimation of Markov random field prior parameters using Markov chain Monte Carlo maximum likelihood. *IEEE Trans. on Image Processing*, 8(7) :954–963, 1999.
- [Devaux,1997] F. Devaux, *Filtrage d'images par réseaux de neurones*. Maîtrise en informatique Université paris 8, 1997.
- [Di Caro, 1998] Di Caro, G., Dorigo, M.(1998), AntNet: Distributed Stigmergetic Control for Communications Networks, *Journal of Artificial Intelligence Research (JAIR)*, (9): p. 317-365..
- [Diday, 1971] E. Diday. Une nouvelle méthode en classification automatique et reconnaissance des formes : la méthode des nuées dynamiques. *Revue de Statistique Appliquée*, 18(2) :20–33, 1971.
- [Dorigo, 1992] M. Dorigo, *Optimization, Learning and Natural Algorithms (in Italian)*. PhD thesis, Politecnico di Milano, Italy. 1992
- [Dorigo, 1991] M. Dorigo, V. Maniezzo, A. Colorni, Positive feedback as a search strategy, Technical Report 91-16 Politecnico di Milano, Italy.1991.
- [Dorigo, 1996] M. Dorigo, V. Maniezzo, A. Colorni,, The Ant System: Optimization by a Colony of Cooperating Agents, *IEEE Transactions on Systems, Man and Cybernetics-Part B*, 1(26): p. 29-41. 1996.
- [Dorigo, 1997a] M. Dorigo, M. Gambardella, M. Ant Colony for the Traveling Salesman Problem, *BioSystems*, (43): 73-81, 1997.
- [Dorigo, 1997b] M. Dorigo, M. Gambardella, Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem, *IEEE Transactions on Evolutionary Computation* Vol 1: p. 53-66 1997.
- [Dorigo, 2004] M. Dorigo, T. STUZLE,. *Ant Colony Optimization*. MIT Press, Cambridge, MA, 2004.
- [Duchesnay, 2001] E. Duchesnay *Agents situés dans l'image et organisés en pyramide irrégulière. Contribution à la segmentation par une approche d'agrégation coopérative et adaptative*. Thèse de Doctorat de l'Université de Rennes-1 2001.

- [Dudot, 2005] A. Dutot, A. Cardon, D. Olivier, et F. Guinand, Competing ants for organization detection, application to dynamic distribution. Proc. of the ECCS 05, European conference on complex systems. 2005
- [Eberhart, 2001] R. Eberhart, J. Kennedy, et Y. Shi, . Swarm Intelligence. Evolutionary Computation. Morgan Kaufmann. 2001
- [Elkharraz, 1999] Elkharraz *Classification automatique floue par réseaux de neurones. Application aux images couleur* , Thèse de 3eme cycle de la Faculté des Sciences de Rabat, Maroc, juillet 1999.
- [Ferber, 1995] Ferber Jacques, Les Systèmes Multi-Agents : vers une Intelligence Collective, Inter Editions, 1995.
- [Fogel, 1966] L. Fogel. & al., Artificial Intelligence through Simulated Evolution, John Wiley & Sons, New-York, 1966.
- [Forgey, 1965] E.M. Forgey. Cluster analysis of multivariate data : efficiency versus interpretability of classification. Biometrics, 21, 1965.
- [Forrest, 1990] S. Forrest, émergent computation: self-organizing, collective, and cooperative phenomena in natural and artificial computing networks, Proceedings of the Ninth annual CLNS conference, 1990..
- [Freixenet. 2002] J. Freixenet, X. Muñoz, D. Raba, J. Marti, et X. Cufi. Yet Another Survey on Image Segmentation : Region and Boundary Information Integration. ECCV, pages 408–422, 2002.
- [Fu, 1980] K.S. Fu and J.K. Mui, A survey on image segmentation, Pattern Recognition, vol. 13, p. 3-16, 1980.
- [Fuh, 2000] C. Fuh, S. Cho and K. Essig. Hierarchical Color Image Region Segmentation for Content-Based Image Retrieval Systems. IEEE Transactions on Image Processing, vol. 9, no. 1, pp. 156-162, 2000.
- [Gambardella, 1997] L. Gambardella, M. Dorigo, HAS-SOP: An hybrid ant system for the sequential ordering problem, Technical Report (11), IDSIA, Lugano, CH. 1997
- [Geman, 1984] S. Geman et D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Trans. on Pattern Analysis and Machine Intelligence, 6(6) :721–741, novembre 1984.
- [Georgé, 2004] J-P Georgé *Résolution de problèmes par émergence, Etude d'un Environnement de Programmation Emergente* , Thèse de l'Université Paul Sabatier, Toulouse, Juillet 2004.
- [Germond, 1999] L. Germond *Trois principes de coopération pour la segmentation en imagerie de résonance magnétiques*. Thèse de doctorat de l'université Joseph Fourier Grenoble I. 11 Octobre 1999.
- [Giraudon, 1987] G. Giraudon, Chaînage efficace de contours, Rapport interne INRIAn°605, janvier 1987.
- [Gleizes, 2004] M-P Gleizes *Vers la résolution de problèmes par émergence* Habilitation à diriger des recherches de l'Université Paul Sabatier Spécialité : Informatique, 2004.
- [Goldberg, 1989] D. Goldberg Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, 1989.

- [Goldstein, 1999] J. Goldstein, *émergence as a Construct : History and issues*, Emergence Volume 1, Issue 1, pp 49-71, 1999.
- [Gong, 1999] M. Gong, Y.Yong, Genetic-based multiresolution color image segmentation.
- [Gonzalez, 1992] R.C. Gonzalez And R.E, Woods. Digital image processing. Addison-Wesley, Reading, MA.
- [Goss, 1989] S. Goss, S. Aron, J.-L.Deneubourg , J.M. Pasteels, Self organized shortcuts in the argentine ant, *Natur wissenschaften*, Vol. 76, 1989, p. 579-581.
- [Grassé, 1959] P.P Grassé. La reconstruction du nid et les interactions inter-individuelles chez les bellicoitermes natalenis et cubitermes, la théorie de la stigmergie - essai d'interprétation des termites constructeurs. *Insectes Sociaux*, 6 :41–81, 1959.
- [Halkidi 2001] M. Halkidi, Y. Batistakis and M. Vazirgiannis. On Clustering Validation Techniques. *Intelligent Information Systems Journal*, Kluwer Pulishers, vol. 17, no. 2-3, pp.107-145, 2001.
- [Hamerly, 2002] G. Hamerly and C. Elkan. Alternatives to the K-means Algorithm that Find Better Clusterings. In *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM-2002)*, pp. 600-607, 2002
- [Hamerly, 2003] G. Hamerly. *Learning Structure and Concepts in Data using Data Clustering*, PhD Thesis. University of California, San Diego, 2003.
- [Handl, 2003] J. Handl, *Ant-based methods for tasks of clustering and topographic mapping: extensions, analysis and comparison with alternative techniques*. Masters Thesis, universität Erlangen-Nürnberg, Erlangen, Germany. 2003.
- [Haralick, 1985] R.M. Haralick and L.G. Shapiro, Survey : Image segmentation techniques, *Computer Vision, Graphics and Image Processing*, vol. 29, p.100-132, 1985.
- [Haris, 1998] K. Haris, S. Efstratiadis, N.Magiaveras et A. Katsaggelos. Hybrid image segmentation using watersheds and fast region merging. *IEEE Trans.Image. Processing*, vol.7, pp 1684-1699, 1998
- [Heath, 1996] M Heath, S Sarkar, T Sanocki, et K Bowyer. Comparison of Edge Detectors : A Methodology and Initial Study. *Computer vision and image understanding*,69(1) : 38–54, january 1996.
- [Hebb, 1949] D. Hebb, *The Organization of Behavior*, New York: Wiley, 1949.
- [Hejmans, 1993] H.J.A.M. Hejmans et L. Vincent. Graph morphology in image analysis. In E.R. Dougherty, editeur, *Mathematical morphology in image processing*, pages 171-203. Marcell Dekker, New York, 1993.
- [Herdy, 1991] M. Herdy Application of the Evolution Strategie to Discrete Optimization Problems, in *Proceedings of the 1st Workshop on Parallel Problem Solving from Nature*, H.P. Schwefel & R. Männer Eds., Springer, 1991, pages 188-192.
- [Hertz, 1988] L. Hertz, et R.W Schafer Multilevel thresholding using edge matching, *Computer Vision, Graphics, and Image Processing*, 44: 279-295, 1988.
- [Hoffmeister, 1991] F. Hoffmeister et T. Bäck Genetic Algorithms and Evolution Strategies : Similarities and differences, in *Proceedings of the 1st Workshop on Parallel Problem Solving from Nature*, H.P. Schwefel & R. Männer Eds., Springer, 1991, pages 455-469.
- [Holland, 1975] J. Holland *Adaptation in Natural and Artificial Systems*, MIT Press, 1975.

- [Holland, 1999] O.E Holland and Melhuish.C. Stigmergy, self-organization, and sorting in collective robotics. *Artificial Life*, 5 :173–202, 1999.
- [Hong, 1984] T. H. Hong et A. Rosenfeld. Compact region extraction using weighted pixel linking in a pyramid. *IEEE Trans. Pattern Analysis and Machine Intelligence* 6(2):222-229, 1984.
- [Hopfield, 1982] J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, *Proceedings of the National Academy of Sciences* 79 : 2554-2558,1982.
- [Heraud, 1995] R. Heraud and O. Monga, *Vision par ordinateur: outils fondamentaux*, edition Hermes, 1995
- [Horne, 1990] C. Horne. Unsupervised image segmentation. PhD thesis. Ecole Polytechnique Fédérale de Lausanne. Switzerland, 1990.
- [Horwitz, 1976] S.L. Horowitz, and T. Pavlidis. Picture segmentation by a tree traversal algorithm. *Journal of the association for computing machinery*, 23(2): 368-388.
- [Hu, 1992] Hu, Y., Dennis, T.J., 1992. Simulated annealing and iterated conditional modes with selective and confidence enhanced update schemes. In: *Proc. 5th Annual IEEE Symp. Computer-based Medical Systems*, pp. 257–264.
- [Hueckel, 1971] M.F Hueckel An operator which locates edges in digitized pictures, *J. Ass.Comput. Mach*, vol. 18, n° 1, pp. 113-125, 1971.
- [Jaggi, 1998] C. Jaggi *Segmentation par méthode Markovienne de l'encéphale humain en imagerie par résonance magnétique théorie, mise en œuvre et évaluation*. Thèse de Doctorat de l'Université de CAEN. 1998.
- [Jain, 1988] A.K Jain et R.C Dubes *Algorithms for clustering Data*. Prentice Hall Advanced Reference series, 1988.
- [Jain, 1999] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering : a review. *ACM Computing Surveys*, 31(3) :264\*322, 1999.
- [Jain, 2000] A. Jain, R.Duin et J.Mao. *Statistical Pattern Recognition : A review*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No.1, pp 4-37 , 2000.
- [Jolion, 1989] J.M. Jolion. *Analyse d'images : le modèle pyramidal*. *Traitement du Signal*. 1989.
- [Kapur, 1985] J.N. Kapur, P. K. Sahoo and A. K.C. Wong . A new method for gray-level picture thresholding using the entropy of histogram *CVGIP* 29 pp.273-2851985
- [Kara-Falah, 1995] R. Kara-Falah. *Segmentation d'images : Coopération, fusion, évaluation*. Thèse de doctorat, Université de Savoie, Juin 1995.
- [Kass, 1988] M. Kass A. Witkin , D. Terzopoulos Snakes: Active contour models, *Computer Vision, Graphics, and Image Processing*, : pp. 321-331, 1988.
- [Kato, 1994] Z. Kato, *Modélisation Markoviennes Multiresolutions en vision par ordinateur. Application à la segmentation d'images SPOT*, Thèse de l'Université de Nice Sophia Antipolis. 1994.
- [Kennedy 1995] J. Kennedy , R-C. Eberhart, Particle swarm optimization. In *Proceedings of the 1995 IEEE International Conference on Neural Networks*, volume IV, pages 1942-1948, Piscataway, NJ, 1995. IEEE Service Center.

- [Kermad, 1995] C. Kermad, K. Chehdi, et C. Cariou, segmentation d'images par multiseuillage et fusion de régions labellisées minimisant un critère de similarité. In Quinzième Colloque GRETSI. 2 :641-644, JUAN-LES-PINS, septembre, 1995.
- [Kervrann, 1995] C. Kervrann, F. Heitz, A Markov random field model based approach to unsupervised texture segmentation using local and global spatial statistics, IEEE trans. Image Process. 1995.
- [Kim, 2000] E.Y. Kim, S.H. Park, H. J. Kim, „A Genetic algorithm-based segmentation of Markov Random Field modeled images”, IEEE Signal processing letters 11(7): 301–303, 2000
- [Kirch, 1971] R.A. Kirsch, Computer determination of the constituent structure of biological images, Computers and Biomedical Research, vol. 4, n°3, p.315-328, 1971.
- [Kohler, 1981] R. Kohler. A segmentation system based on thresholding. Comput. Graphics Image Process. 15 pp319-338. 1981.
- [Kohonen, 1984] T.Kohonen Self-Organization and Associative Memory Springer-Verlag, 1984
- [Koller, 1995] Th.M. Koller, G. Gerig, G. Székely, and D. Dettwiler. Multiscale detection of curvilinear structures in 2-D and 3-D image data. In IEEE, editor, International Conference on Computer Vision (ICCV'95), pages 864-869, 1995.
- [Koza, 1992] J. Koza Genetic Programming : On the Programming of Computers by means of Natural Evolution, MIT Press, Massachusetts, 1992.
- [Koza, 2003] J.R Koza, M.A Keane, M.J Streeter, W. Mydlowec, J. Yu, , et G. Lanza Genetic Programming IV: Routine Human-Competitive Machine Intelligence ,Kluwer Academic Publishers, ISBN 1-4020-7446-8, 2003.
- [Krippendorff, 1997] K. Krippendorff Dictionary of Cybernetics” 27 janvier 1997. <http://pespmc1.vub.ac.be/SELFORG.html>
- [Krishna, 1999] K. Krishna et M. Murty. Genetic k-means algorithm. IEEE Transactions on Systems, Man and Cybernetics - Part B, 29(3):433–439, 1999.
- [Krishnamachari, 1997] S. Krishnamachari, et R. Chellappa, Multiresolution Gauss–Markov random field models for texture segmentation. IEEE Trans. Image Process. 2(6): p. 251–267. 1997.
- [Kuntz, 1997] P. Kuntz, P. Layzell et Snyers. D. A colony of ant- like agents for partitioning in vlsi technology. In P. Husbands et I. Harvey, \_editeurs, Proceedings of the Fourth European Conference on Arti\_cial Life, pages 417{424. MIT Press, 1997.
- [Labroche, 2002] N. Labroche, N. Monmarché, G. Venturini, . A new clustering algorithm based on the chemical recognition system of ants. Proc. of the 15th European Conference on Artificial Intelligence, France, pp. 345-349, IOS Press. 2002.
- [Lakshmanan, 1989] S.Z Lakshmanan, S.Z., Derin, Simultaneous parameter estimation and segmentation of Gibbs Random field using Simulated Annealing, IEEE Trans. On Pattern Analysis and Machine Intelligence. 8(11): p. 799-813. 1989.
- [Langton, 1991] C. Langton. Life at the Edge of Chaos, in Artificial Life II : SFI Studies in the Sciences of Complexity Vol. 10, C. Langton Ed., Addison-Weasley, 1991, pages 41-91.

- [Lee, 1990] S.U. Lee and S.Y. Chung, A comparative study of several global thresholding techniques for segmentation, *Computer Graphics and Image Processing*, vol. 52, 171-190, 1990.
- [Lee, 2000] C. Lee and E. Antonsson. Dynamic Partitional Clustering Using Evolution Strategies. In *The Third Asia-Pacific Conference on Simulated Evolution and Learning*, 2000.
- [Leung, 2000] Y. Leung, J. Zhang and Z. Xu. Clustering by Space-Space Filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no.12, pp. 1396-1410, 2000.
- [Kirkpatrick, 1985] S. Kirkpatrick, D.C. Gelatt, and M.P. Vechhi. Optimization by simulated annealing. *Science*, 220 :671–680, May 1983.
- [Levine, 1985] M. D. Levine et Ahmed M. Nazif. Dynamic Measurement of Computer Generated Image Segmentations. *IEEE Transaction on Pattern Analysis and Machine intelligence*, PAMI-7(2) : 155–164, March 85.
- [Liew, 2000] A.W.C. Liew, S.H. Leung, et W.H. Lau. Fuzzy image clustering incorporating spatial continuity. *IEE Proceedings on Vision Image Signal Processing*, 147(2) :185–192, April 2000.
- [Lim, 1990] Y. W. Lim et S. U. Lee. On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques. *Pattern Recognition*, 23(9) :935–952, 1990.
- [Liu, 1994] J. Liu, Y.H. Yang, Multiresolution color image segmentation, *IEEE Trans. On PAMI*, 16(7), (1994) 689-700
- [Liu., 1999] J. Liu , Y. T. Tang Adaptive Image Segmentation With Distributed Behavior-Based Agents . *IEEE Trans. On PAMI* 21(6) – p.544-551 1999.
- [Lopès, 1999] A. Lopès, R. Fjørtoft, D. Ducrot, P. Marthon, and C. Le maréchal. Segmentation of SAR images in homogeneous regions. In C. H. Chen, editor, *Information Processing for Remote Sensing*. World Scientific Publishing Co., Singapore, 1999.
- [Lumer, 1994] E.D Lumer, et B. Faieta, Diversity and adaptation in populations of clustering ants, in D. Cli®, P. Husbands, J. Meyer and S. Wilson (eds), *From Animals to Animats 3*, Proceedings of the 3rd International Conference on the Simulation of Adaptive Behavior, The MIT press/Bradford Books.1994.
- [Lutton, 2003] E. Lutton, Algorithmes génétiques et algorithmes évolutionnaires, *Traité d'Informatique Industrielle*, 2003.
- [MacQueen, 1967] L. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceeding 5th Berkeley Symp.*, pages 281–297, 1967.
- [Maniezzo, 1994] V. Maniezzo, A. Colomi, M. Dorigo,, The Ant System Applied to the Quadratic Assignment Problem”, Technical Report. IRIDIA. (28), Université Libre de Bruxelles, Belgium, 1994.
- [Marr, 1980] D. Marr, H. Hildreth, Theory of edge detection, *Proceedings of the Royal Society of London B207*, p 187-217, 1980.
- [Marroquin, 1985] J.L. Marroquin, Optimal Bayesian estimation for image segmentation and surface reconstruction, *MIT.AI.MEMO* 893, 1985
- [Martinoli, 1999] A. Martinoli, A. Ijspeert, et L. Gambardella, . . A Probabilistic Model for Understanding and Comparing Collective Aggregation Mechanisms In (Floreano et al., 1999), pages 575–584. 1999.



- [Maulik 2003] U. Maulik et S. Bandyopadhyay, Fuzzy partitioning using a real-coded variable-length genetic algorithm for pixel classification, *GeoRS(41)*, No. 5, pp. 1075-1081. 2003.
- [McCulloch, 1943] W. McCulloch, and W. Pitts, A logical calculus of the ideas immanent in nervous activity, *Bulletin of Mathematical Biophysics* 5: 115-133, 1943.
- [Melhuish, 1999] C. Melhuish, Exploiting Domain Physics : Using Stigmergy to Control Cluster Building with Real Robots . In (Floresano et al., 1999), pages 585–595. 1999
- [Mendel, 1865] G.J Mendel, *Experiments in Plant-hybridization*. Princeton University Press, 1865.
- [Menhert, 1997] Mehnert and P. Jackway, An improved seeded region growing algorithm, *Pattern Recognition Letters*, vol. 18, pp. 1065-1071, 1997.
- [Merkel, 2002] D. Merkle, M. Middendorf and H. Schmeck, Ant colony optimisation for resource constrained project scheduling, *IEEE Transaction on Evolutionary Computation*, 6(4): 333-346 August 2002.
- [Michalewicz, 1996] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. Springer Verlag, third - 1999 edition, 1996.
- [Minsky, 1969] M. Minsky and S. Papert, *Perceptrons*, Expanded Edition. MIT Press, 1969.
- [Monga, 1987] O. Monga et B. Wrobel B., Segmentation d'images: vers une méthodologie'', *Traitement du Signal*, vol. 4, n° 3, pp 169-193, 1987.
- [Monga, 1990] O. Monga, R. Deriche, et G.Malandain. Recursive filtering: a primary tool for 3d edge detection. In *First European Conference on Computer Vision. ECCV 90?* 1990.
- [Monmarché, 1999] N. Monmarché On Data Clustering with Artificial Ants. In:Freitas AA, editor, *Data Mining with Evolutionary Algorithms:Research Directions – Papers from the AAAI Workshop*, pages 23-26. AAAI Press, 1999..
- [Montanvert, 1991] A. Montanvert, P. Meer and A. Rosenfeld, Hierarchical image analysis using irregular tessellations, *IEEE Trans. PAMI*, vol. 13, p. 307-316, avril 1991.
- [Moreira, 1996] J. Moreira et L.D.F. Costa. Neural-based color image segmentation and classification using self-organizing maps. 47-54, 1996.
- [Murthy, 1996] C.A. Murthy, N. Chowdhury, In Search of Optimal Clusters Using Genetic Algorithms, *Pattern Recognition Letters*, vol.17, No. 8, , pp. 825-832, 1996.
- [Odell, 2001] J. Odell, H.V Parunak and B. Bauer Representing Agent Interaction Protocols in UML, *Oriented Software Engineering*, P. Ciancarini and M. Wooldridge Editors, Springer-Verlag, Berlin, pages 121-140, 2001
- [Ohlander, 1975] R. Ohlander, *Analysis of natural scenes*, PhD Thesis, Carnegie Institute of Technology, Dept. of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, 1975;
- [Ohlander, 1979] R. Ohlander, K. Price, and D.R.reddy. Picture segmentation using recursive region splitting method. *Computer Graphics and Image Processing*, 8: 313-333, 1979.
- [Ohta, 1980] Y. I. Ohta, T. Kanade, et T. Sakai. Color information for region segmentation. *Computer Graphics and Image Processing*, 13 :222–241, 1980.
- [Omran, 2002] M. Omran, A. Salman, and A.P. Engelbrecht. Image classification using particle swarm optimization. In *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning 2002 (SEAL 2002)*, pages 370–374, 2002.

- [Ostrovsky, 2000] R. Ostrovsky, et Y. Rabani. Polynomial time approximation schemes for geometric k-clustering. In IEEE Symposium on Foundations for computer science. Pages 349-358, 2000.
- [Ouadfel, 2005a] S. Ouadfel , M. Batouche des fourmis pour la segmentation des images. Journées d'étude en Informatique graphique JIG05, Biskra novembre 2005.
- [Ouadfel, 2005b] S. Ouadfel , M. Batouche Ant Colonies For MRF-Based Image Segmentation, Setit 2005, 3rd International Conference on Sciences of Electronic, Technologies of Information and Telecommunications, Tunisie,.27-31 march, 2005.
- [Ouadfel, 2004] S. Ouadfel , M. Batouche MRF-based image segmentation using autonomous agents, Setit 2004, 2rd International Conference on Sciences of Electronic, Technologies of Information and Telecommunications, Tunisie,.14-17 march, 2004.
- [Ouadfel, 2003a] S. Ouadfel , M. Batouche. MRF-based image segmentation using Ant Colony System. In Electronic Letters on Computer Vision and Image Analysis . ISSN 1577-5097. September 2003 - Volume 2, Number 1 pp 12-24.
- [Ouadfel, 2003b] S. Ouadfel , M. Batouche A Hybrid MAX-MIN Ant System to MRF-based Image Segmentation. In a Special Issue of Wseas Transactions in Systems Issue 3, Volume 2, ISSN 1109-2777. pp734-741. Juillet 2003.
- [Pal, 1993] N.R. Pal, S.K. Pal, A review on image segmentation techniques, Pattern Recognition 9(26): 1277–1294, 1993.
- [Palus, 1999] H Palus et D Bereska. Region-Based Colour Image Segmentation. 1999.
- [Pandya, 1996] A. Pandya et R.Macy . Pattern Recognition with Neural Networks in C++, CRC Press, 1996.
- [Panjwani, 1995] D.K Panjwani, G. Healey, G. Markov random field models for unsupervised segmentation of textured color images, IEEE Trans. Pattern Anal. Machine Intell. 10(17): p. 939–954. 1995.
- [Parpinelli, 2000] R.S. Parpinelli, H. S. Lopez and A. A. Freitas, data mining with ant colony optimisation algorithm, IEEE Transaction on Evolutionary Computation, 6(4): 321-332 August 2002.
- [Parunak, 1997] H. Parunak H Go to the Ant: Engineering Principles from Natural Multi-Agent Systems , Annals of Operations Research 75 69-101 Special Issue on Artificial Intelligence and Management Science.
- [Pavlidis, 1990] T. Pavlidis and Y.-T. Liow. Integrating region growing and edge detection. IEEE Trans. Pattern Analysis and Machine Intelligence, 12(3):225--233, 1990.
- [Pavlidis, 1992] T. Pavlidis, Why progress in machine vision is so slow, Pattern Recognition Letters, vol. 13, p. 221-225, 1992.
- [Perez, 1992] P. Perez, et F. Heitz, Une approche multi/echelle `a l'analyse d'images par champs markoviens', Traitement du Signal 9(6), 459–472. 1992.
- [Philipp-Foliguet , 2001]S. Philipp-Foliguet. Evaluation de la Segmentation. Technical report, ETIS, mars 2001.
- [Pinho, 1996] A. J. Pinho, L. B. Alemeida. Neural Network classifiers for edge detection, VII national symposium on pattern recognition and image analysis. Volume 1, PP. 371-376. 1996.

- [Pontecorvo, ] C. Pontecorvo, A. Bouzerdoum. Edge detection using a Shunting Inhibitory Cellular Neuronal Network.
- [Prewitt, 1970] J. M. S. Prewitt. Object enhancement and extraction. In B. S. Lipkin and A. Rosenfeld. Editors. *Picture Processing and Psychopictorics*. Pages 75-149, Academic Press. New York. 1970.
- [Prigogine 1977] I. Prigogine, G. Nicolis, *Self Organization in Non-Equilibrium Systems*, (Chaps. III and IV), J. Wiley and Sons, New York, 1977.
- [Pun, 1981] T. Pun Entropic thresholding: A new approach CVGIP.16, pp 210-239, 1981
- [Raghavan, 1979] V.V. Raghavan et K. Birchard. A clustering strategy based on a formalism of the reproductive process in natural systems. In *Information Implications into the Eighties*, Proceedings of the Second International Conference on Information Storage and Retrieval, pages 10–22. ACM, 1979
- [Ramos, 1999] V. Ramos et F Muge, editors. *Image Segmentation by Colour Cube Genetic K-Mean Clustering*. Third Workshop on Genetic Algorithms and Artificial Life, LASEEB, ISR, IST, 1999.
- [Ramos, 2002] V. Ramos, F. Muge, F. and P. Pina, Self-organized data and image retrieval as a consequence of inter-dynamic synergistic relationships in artificial ant colonies, Vol. 87, IOS Press.2002.
- [Ratle, 2003] A. Ratle, M. Sebag, *A Novel Approach to Machine Discovery: Genetic Programming and Stochastic Grammars*, Stan Matwin and Claude Sammut, eds, *Proceedings of Inductive Logic Programming*, p.207-222, Springer Verlag., 2003.
- [Rechenberg, 1973] I. Rechenberg *Evolutionstrategien-Optimierung technischer Systeme nach Prinzipien der biologischen Information*, Fromman Verlag, Freiburg, 1973.
- [Regina, 2001] P. Regina K.D. Toennies, Segmentation of medical images using adaptive region growing. *Proc. SPIE Vol. 4322*, p. 1337-1346, *Medical Imaging 2001: Image Processing*, Milan Sonka; Kenneth M. Hanson; Eds.
- [Reynolds, 1987] C-W.Reynolds. Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics: Proceedings of SIGGRAPH '87*, 21(4):25-34, July 1987 , URL: <http://www.red3d.com/cwr/papers/1987/boids.html> [13th March 2002].
- [Reynolds, 2001] C. W. Reynolds, Boids: Background and Update, in: *Boids (Flocks, Herds, and Schools: a Distributed Behavioral Model)*, URL: <http://www.red3d.com/cwr/boids> 13th March 2002.
- [Richard, 2001] N. Richard, M. Dodjat, C. Garbay . Dynamic adaptation of cooperative agents for MRI brain scans segmentation. *Artificial Intelligence in Medecine - AIME'01* – p. 349-358 2001.
- [Rosenberger, 1999] C. Rosenberger. *Mise en oeuvre d'un système adaptatif de segmentation d'images* Thèse de Doctorat de l'université de Rennes I, 1999.
- [Rosenblatt, 1958] F. Rosenblatt, The Perceptron: a probabilistic model for information storage and organization in the brain, *Psychological Review* 65 : 386-408, 1958.
- [Rosenfeld, 1983] A. Rosenfeld and P.D La Torre, Histogram concavity analysis as an aid in threshold selection, *T-SMC-13* pp. 231-235, 1983.

- [Rouquet, 1998] C. Rouquet, P. Bonton, and R. Tomczak, Etude comparative des stratégies de segmentation non supervisée en régions par champs de markov,” *Traitement du Signal*, 1998, vol. 15, n° 1, p. 39-55.
- [Rumelhart 1986] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representations by error propagation, in: vol. I, *Parallel Distributed Processing: Explorations in the microstructure of Cognition*, D.E. Rumelhart and J.L. McClelland, eds., MIT Press, Cambridge, pp.319-362, 1986.
- [Saarinen, 1994] K. Saarinen. Color image segmentation by a watershed algorithm and region adjacency graph processing. In *ICIP'94 : Int. Conf. on Image Processing*, pages 1021–1024, 1994.
- [Saarinen, 1994] K. Saarinen. Color image segmentation by a watershed algorithm and region adjacency graph processing. In *ICIP'94 : Int. Conf. on Image Processing*, pages 1021–1024, 1994.
- [Sahoo, 1988] P.K. Sahoo, S. Soltani, A.K.C. Wong and Y.C. Chen, A survey of thresholding techniques, *Computer Vision Graphics and Image Processing*, vol. 41, p. 233-260, 1988.
- [Sahoo. 1988] P. K. Sahoo, S. Soltani, A. K. C.Wong, et Y. C. Chen. A survey of thresholding techniques. *CVGIP*, 41 : 233–260, 1988.
- [Salzenstein , 1998] F. Salzenstein and W. Pieczynski, Sur le choix de la méthode de segmentation statistique d’images,” *Traitement du Signal*, 1998, vol. 15, n° 2, p. 119-127.
- [Schettini, 1993] R. Schettini. A segmentation algorithm for color images. *Pattern Recognition Letters*, 14 :499–506, 1993.
- [Schwefel, 1981] H. Schwefel *Numerical Optimisation of Computer Models*, Wiley, Chichester, 1981.
- [Semet, 2003] Y. Semet, E. Lutton and P.Collet. Ant Colony Optimization for e-learning : Observing the emergence of pedagogic suggestions. In *IEEE Swarm Intelligence Symposium 2003*, Indianapolis, Indiana, april 2003.
- [Shen, 1986] J. Shen and S. Castan. An optimal linear operator for edge detection. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'86)*, pages 109--114, Miami Beach, Florida, USA, June 22--26 1986.
- [Shen, 1992] [J. Shen and S. Castan\_ An optimal linear operator for step edge detection. *CVGIP. Graphics Models and Image Processing*.54(2). 112-133, Mars 1992.
- [Shufelt, 1999] J. A. Shufelt. Performance Evaluation and Analysis of monocular building extraction from aerial imagery. *Transaction on Pattern Analysis and Machine Intelligence*, 21(4) : 311–326, 1999.
- [Sobel, 1978] I. Sobel. Neighbourhood coding of binary images for fast contour following and general array binary processing. *Computer Graphics and Image Processing*.8:127-135,1978.
- [Solnon, 2000] C. Solnon. Solving permutation constraint satisfaction problems with artificial ants. In *Proceedings of ECAI'2000*, IOS Press, Amsterdam, The Netherlands, pages 118–122, 2000.
- [Strasters, 1991] K. C. Strasters and J. J. Gerbrands, Three-dimensional image segmentation using a split, merge and group approach,” *Pattern Recognition Letters*, 1991, vol. 12, p. 307-325.

- [Stützle, 1999] T. Stuzle, H.H. Hoos. The max-min ant system and local search for Combinatorial Problems, in *Meta-heuristics : advances and trends in local search paradigms for Optimization*” by S. VOSS, I.H. OSMAN and C. ROUCAIROL, Kluwer Academic Publishers, Boston, 1999, p. 313-329.
- [Stützle, 2000] T. Stützle, H. H. Hoos, MAX-MIN ant system, *Future Generation Computer Systems*, Vol. 16 (2000)
- [Syswerda, 1991] Syswerda, G. A study of reproduction in generational and steady-state genetic algorithms. In Rawlins, editor, *Foundation of Genetic Algorithms*. Morgan Kaufman, 1991.
- [Taillard, 1999] E. Taillard, Ant systems, `Handbook of Applied Optimization”, P. PARDALOS, M.G.C RESENDE.
- [Taillard, 1997] Taillard, L., Gambardella, L. (1997), An Ant Approach for Structured Quadratic Assignment Problems”, 2<sup>nd</sup> Metaheuristic International Conference (MIC-97), Sophia-Antipolis, France.
- [Tang, 1999] L. Tang, L. F. Tian, et B. L. Steward. Color Image Segmentation with Genetic Algorithm for In-Field Weed Sensing. ASAE, 1999.
- [Theraulaz, 1997] G. Theraulaz, F. Spitz, *Auto-organisation et comportement*, Hermès, Paris, 1997.
- [Tikhonov, 1974] A. N. Tikhonov, et V.Y. Arsenin, *méthodes de résolution de problèmes mal poses*, MIR, Moscou, 1974.
- [Topin, 1999] X. Topin *Etude de l'auto-organisation par coopération appliquée à l'apprentissage comportemental de robots fourmis* , DEA RCFR – Université Paul Sabatier, Toulouse Juin 1999.
- [Tou, 1974] J. Tou et R. Gonzalez. *Pattern Recognition Principles*. Addison-Wesley , Massachusetts, USA, 1974.
- [Tou, 1979] J. Tou, DYNOC- A Dynamic Optimal Cluster Seeking Technique. *International Journal of Computer and Information sciences*, vol.8, no.6, pp. 541-547, 1979.
- [Trémeau, 1997] A. Trémeau et N. Borel. A region growing and merging algorithm to color segmentation. *Pattern Recognition*, 30 :1191–1203, 1997.
- [Tseng, 2001] L.Y. Tseng, S.B. Yang, A genetic approach to the automatic clustering problem, *Pattern Recognition*, vol. 34, No. 2, pp. 415-424, 2001
- [Turi, 2001] R.H. Turi *Clustering-Based Color Image Segmentation*, Phd Thesis. Monash University, Australia 2001.
- [Ünsal, 1993] C. Ünsal *Self-organization in large populations of mobile robots* Master of Sciences in Electrical Engineering, May 1993, Blacksburg, Virginia. <http://armyant.ee.vt.edu/unsalWWW/cemthesis.html>.
- [Van de Vijver, 1997] G. Van De Vijver, *Emergence et explication*, *Intellectica : Emergence and explanation*, 1997/2 n°25, ISSN n°0984-0028 185-194, 1997.
- [Varela, 1988] F. Varela *Autonomie et connaissance : essai sur le vivant* Editions du seuil – 1988.
- [Veenman, 2003] C.J. Veenman\_, M.J.T. Reinders, and E. Backer. *IEEE Transactions on Image Processing*, Vol. 12, No. 3, pp. 304-316, March 2003.

- [Vinet 1991] L. Vinet. *Segmentation et mise en correspondance de régions de paires d'images stéréoscopiques* Thèse de Doctorat de l'université de Paris IX Dauphine, 1991.
- [Watanabe, 1974] S Watanabe and CYBEST Group. An automated apparatus for Cancer prescreening Cybest. In *Computer Graphics and Image Processing* number pages 350-358. 1974.
- [Weszka, 1974] J.S. Weszka , R.N.Nagel A. Rosenfeld A threshold selection technique, *IEEE Transactions on Computer*, vol. 23, pp. 1322-1326, 1974.
- [Widrow, 1960] B. Widrow and M. Hoff, Adaptive switching circuits, 1960 IRE WESCON Convention Record, New York: IRE, pp. 96-104, 1960.
- [Wilson, 1985] S. Wilson, Knowledge Growth in an Artificial Animal, in *Proceedings of the 1<sup>st</sup> International Conference on Genetic Algorithms and their Applications*, J.J. Grefenstette Ed., 1985, pages 16-23.
- [Wu, 1994] J. Wu, H. Yan et A. Chalmers. Color Image Segmentation Using Fuzzy Clustering and Supervised Learning. *Journal of Electronic Imaging*, Vol. 3, no. 4, pp. 397-403, 1994.
- [Wu, 2002] K.-L. Wu, et M.-S.Yang, , Alternative C-means Clustering Algorithms, *Pattern Recognition*, Vol. 35, pp. 2267. 2278 2002.
- [Xiao, 2003] X. Xiao, E. Dow, R. Eberhart, Z. Ben Miled, and R. J. Oppelt. Gene clustering using self-organizing maps and particle swarm optimization. In *Second IEEE International Workshop on High Performance Computational Biology (HICOMB 2003)*, 2003.
- [Yamazaki, 1995] T. Yamazaki, D. Gingras, Image classification using spectral and spatial information based on MRF models, *IEEE Trans. Image Processing* 9(4): p. 1333–1338. 1995.
- [Roberts, 1965] L.G. Roberts. Machine perception of three-dimensional solids, chaoter optical and electro-optical information processing. J-P.Tippet and al. Ed. MIT Press. Pp 159-197. Cambridge 1965.
- [Yasnoff, 1977] W. A. Yasnoff, J. K. Mui, et J. W. Bacus. Error measures for scene segmentation. *Pattern Recognition*, 9 : 217–231, 1977.
- [Yin, 1993] L. Yin, J astola, A new Class of Nonlinear Filters-Neural Filters, *IEEE Transaction on Signal Processing*, pp 1201-1222, 1993.
- [Zadeh, 1965] L.A. Zadeh., Fuzzy Sets, *Information and control*, Vol. 8, pp. 338-353, 1965.
- [Zhang, 1997] Y.J. Zhang, Evaluation and comparison of different segmentation algorithms. *Pattern Recognition Letters*, 18 : 963–974, 1997.
- [Zhang 1996] Y. J. Zhang. A survey on evaluation methods for image segmentation. *Pattern Recognition*, 29(8) : 1335–1346, 1996.
- [Zucker, 1976] S. Zucker. Region growing : childhood and adolescence. *Computer Graphics and Image Processing*, 5 :382–399, 1976.