



**HAL**  
open science

# Conception de métaheuristiques d'optimisation pour la segmentation d'images. Application aux images biomédicales

Amir Nakib

► **To cite this version:**

Amir Nakib. Conception de métaheuristiques d'optimisation pour la segmentation d'images. Application aux images biomédicales. Informatique [cs]. Université Paris XII Val de Marne, 2007. Français. NNT: . tel-00308789

**HAL Id: tel-00308789**

**<https://theses.hal.science/tel-00308789>**

Submitted on 1 Aug 2008

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

**THÈSE DE DOCTORAT**  
**DE L'UNIVERSITÉ PARIS 12-VAL DE MARNE**  
**UFR de Sciences et Technologie**

par

**Amir NAKIB**

pour obtenir le grade de

**DOCTEUR EN SCIENCES**

Spécialité : **SCIENCES DE L'INGÉNIEUR**

Option : Traitement d'images, optimisation

Équipe d'accueil : Laboratoire Images, Signaux et Systèmes Intelligents (LiSSi, E.A. 3956)

Ecole Doctorale : École Doctorale Sciences et Ingénierie : Matériaux - Modélisation - Environnement (SIMME)

Sujet de la thèse:

---

**Conception de métaheuristiques d'optimisation  
pour la segmentation d'images. Application à des  
images biomédicales**

---

soutenue le 5 décembre 2007 devant la commission d'examen composée de :

Pierre COLLET	Professeur à l'Université de Strasbourg	<b>Rapporteur</b>
Jean DEVARS	Professeur à l'Université de Paris 6	<b>Rapporteur</b>
Christian OLIVIER	Professeur à l'Université de Poitiers	<b>Examineur</b>
Jean LOUCHET	INRIA Rocquencourt	<b>Examineur</b>
Hammouche OULHADJ	Maître de conférences à l'Université de Paris 12	<b>Examineur</b>
Patrick SIARRY	Professeur à l'Université de Paris 12	<b>Directeur de thèse</b>

---

# RESUME

---

La segmentation des images est généralement l'étape la plus importante dans un système d'analyse d'images : dans l'aide au diagnostic en médecine, en navigation autonome des véhicules, etc. Toutes les tâches ultérieures à ces applications, comme l'extraction de primitives, la détection d'une position, ou la reconnaissance d'un objet dépendent fortement de la qualité de la segmentation. L'inconvénient majeur des algorithmes de segmentation actuels est leur incapacité à s'adapter aux différents types d'images.

L'apparition des "métaheuristiques" remonte aux années quatre-vingts. Ces algorithmes stochastiques d'optimisation globale peuvent être appliqués à tout problème, du moment qu'il est formulé sous la forme de l'optimisation de critère(s). Ces algorithmes sont inspirés par des analogies avec la physique (recuit simulé, recuit microcanonique), avec la biologie (algorithmes évolutionnaires) ou avec l'éthologie (colonies de fourmis, essais particuliers). Ils se prêtent aussi à toutes sortes d'extensions, notamment en optimisation multiobjectif.

Afin de concevoir un système de segmentation qui permet d'avoir une meilleure qualité de la segmentation sur une grande variété d'images, nous formulons la segmentation comme un problème d'optimisation, mono-objectif dans un premier temps, puis multiobjectif.

Dans l'approche mono-objectif, nous adaptons plusieurs métaheuristiques au problème de la segmentation. Une application sur des images par résonance magnétique (IRM) cérébrales est ensuite réalisée. Cette adaptation des différentes métaheuristiques pour la segmentation nous permet de les comparer sur les plans suivants : la complexité, la vitesse de convergence, l'adaptabilité, et la reproductibilité des solutions.

Nous proposons ensuite une approche multiobjectif pour résoudre le problème de la segmentation des images. Dans ce contexte, nous développons trois schémas de systèmes de segmentation adaptatifs : le premier est basé sur l'agrégation de critères, le second sur l'approche non-Pareto, et le troisième sur l'approche Pareto. Enfin, dans le cas particulier de la segmentation des espaces de ventricules cérébraux, nous appliquons différentes approches sur des IRM saines et d'autres pathologiques.

**Mots clés :** segmentation, seuillage, optimisation, optimisation multiobjectif, recuit simulé, recuit microcanonique, colonies de fourmis, algorithmes génétiques, algorithmes évolutionnaires, dérivation non-entière, entropie, entropie exponentielle, entropie survivante exponentielle, images IRM.

---

# ABSTRACT

---

**I**mage segmentation generally is the important step in any automated image analysis system, such as autonomous vehicle navigation, object recognition, etc. All these subsequent tasks, such as feature extraction, object detection, and object recognition, depend on the quality of segmentation. One of the fundamental weaknesses of current image segmentation algorithms is their inability to adapt the segmentation process to different kinds of images.

The metaheuristics appeared in the eighties. These global optimization algorithms are stochastic and can be applied to any problem, at the condition it is formulated as a mono-objective or multiobjective optimization problem. They are inspired from an analogy with physics (simulated annealing, microcanonical annealing), with biology (evolutionary algorithms) or with ethology (ant colonies, particle swarms). They also can be extended, particularly to multiobjective optimization.

In order to design a segmentation system that allows to have good segmentation results on different kinds of images, we formulate the segmentation as: a mono-objective then a multiobjective optimization problem.

In the mono-objective approach, we adapt several metaheuristics to the segmentation problem. Then, an application to the brain magnetic resonance images (MRI) is performed. This adaptation allows to compare the different metaheuristics in terms of complexity, convergence speed, adaptability and solution reproducibility.

Afterwards, we propose a multiobjective optimization approach to solve the image segmentation problem. In this context, we develop three adaptive methods: the first is based on aggregation of criteria, the second is based on a non-Pareto approach and the third is based on a Pareto approach. Finally, we consider the case of brain ventricle space segmentation and we apply the different approaches to sane and pathologic MRI.

**Keywords:** segmentation, thresholding, optimization, multiobjective optimization, simulated annealing, microcanonical annealing, ant colonies, genetic algorithms, evolutionary algorithms, fractional differentiation, entropy, exponential entropy, survival exponential entropy, MRI.

## TABLE DES MATIERES

---

<b>Introduction générale</b> .....	1
<b>Chapitre 1 : Etat de l'art des méthodes de segmentation d'images</b> .....	5
1.1 Introduction .....	5
1.2 Le seuillage .....	5
1.3 Les méthodes non paramétriques.....	7
1.3.1 La méthode d'Otsu et ses variantes .....	7
1.3.2 Le seuillage entropique .....	9
1.3.2.1 Seuillage à base de l'entropie à une dimension.....	9
1.3.2.2 Entropie à deux dimensions .....	11
1.4 Les méthodes paramétriques .....	13
1.4.1 Approximation de l'histogramme par fonctions de densité de probabilité.....	13
1.4.2 Calcul des seuils de segmentation.....	14
1.4.3 Algorithme espérance-maximisation .....	15
1.5 Seuillage adaptatif.....	16
1.6 Segmentation par classification .....	16
1.7 Les méthodes de segmentation "basées régions" .....	18
1.7.1 Approche par agrégation de pixels .....	19
1.7.2 Approche division-fusion.....	20
1.7.3 Les approches pyramidales .....	21
1.7.4 Approches basées sur la morphologie mathématique .....	21
1.8 Les méthodes de segmentation "basées contours" .....	22
1.9 Conclusion .....	22
<b>Chapitre 2 : Métaheuristiques d'optimisation et segmentation : état de l'art</b> .....	23
2.1 Introduction.....	23
2.2 Formulation du problème de la segmentation .....	23
2.3 L'optimisation multiobjectif .....	24
2.3.1 Transformation en un problème mono-objectif .....	24
2.3.1.1 Méthode d'agrégation.....	24
2.3.1.2 Autres méthodes de transformation.....	25
2.3.2 Approche non-Pareto .....	26
2.3.3 Approche Pareto .....	26
2.3.3.1 Optimum au sens de Pareto.....	27
2.3.3.2 La notion de dominance .....	27
2.3.3.3 La frontière de Pareto .....	28
2.3.3.4 La notion de dominance avec contrainte .....	28
2.4 Les métaheuristiques .....	28
2.4.1 Le recuit simulé .....	28
2.4.1.1 Algorithme de Metropolis .....	29
2.4.1.2 Algorithme du recuit simulé.....	29

2.4.2	Le recuit microcanonique.....	31
2.4.2.1	Algorithme de Creutz .....	31
2.4.2.2	Algorithme du recuit microcanonique .....	31
2.4.3	Algorithmes génétiques .....	32
2.4.3.1	Principe.....	32
2.4.3.2	Les algorithmes génétiques en optimisation multiobjectif.....	35
2.4.4	L'optimisation par essaim particulaire (OEP) .....	35
2.4.4.1	Principe.....	36
2.4.4.2	L'optimisation par essaims particuliers en optimisation multiobjectif.....	37
2.4.5	Les algorithmes de colonies de fourmis .....	38
2.5	Les métaheuristiques en segmentation d'images.....	40
2.6	Conclusion .....	42
<b>Chapitre 3 : Contribution à la segmentation par métaheuristiques en optimisation mono-</b>		
<b>objectif .....</b>		
3.1	Introduction.....	43
3.2	Application du recuit simulé continu à la segmentation.....	43
3.2.1	Algorithme de segmentation.....	44
3.2.2	Résultats .....	46
3.2.3	Bilan.....	46
3.3	Segmentation par entropie à deux dimensions (E2D) et recuit microcanonique.....	47
3.3.1	Algorithme de segmentation.....	47
3.3.2	Résultats .....	49
3.3.3	Bilan.....	50
3.4	Segmentation par entropie exponentielle à deux dimensions (eE2D) et recuit microcanonique hybride .....	50
3.4.1	Entropie exponentielle à deux dimensions .....	50
3.4.2	Algorithme de segmentation par recuit microcanonique hybridé.....	51
3.4.2.1	Algorithme du simplex .....	51
3.4.2.2	Hybridation du recuit microcanonique pour la segmentation.....	53
3.4.3	Résultats .....	54
3.4.4	Bilan.....	56
3.5	Segmentation par eE2D et OEP adaptatif (TRIBES).....	56
3.5.1	Description de TRIBES .....	57
3.5.2	Algorithme de segmentation via TRIBES .....	58
3.5.3	Résultats .....	58
3.5.4	Bilan.....	58
3.6	Segmentation par entropie survivante exponentielle à deux dimensions et OEP.....	59
3.6.1	L'entropie survivante exponentielle à deux dimensions.....	59
3.6.2	Méthode de segmentation.....	61
3.6.3	Résultats et discussion .....	61
3.6.4	Bilan.....	64
3.7	Segmentation par variance intraclasse biaisée et colonies de fourmis .....	64
3.7.1	Algorithme de segmentation.....	65
3.7.1.1	Le critère de variance intraclasse biaisée.....	65

3.7.1.2	L'algorithme ACO.....	65
3.7.2	Résultats .....	66
3.7.3	Bilan.....	66
3.8	Comparaison.....	67
3.9	Application de la 2DSEE et de l'OEP aux images IRM.....	70
3.10	Conclusion .....	70
<b>Chapitre 4 : Amélioration de la segmentation par agrégation de critères .....</b>		<b>72</b>
4.1	Introduction.....	72
4.2	Formulation en un problème multiobjectif.....	72
4.2.1	Critères utilisés .....	73
4.2.2	La fonction objectif équivalente.....	74
4.2.3	Segmentation par combinaison de critères (CCS).....	74
4.3	Description détaillée de l'algorithme CCS.....	75
4.3.1	Algorithme de détection de pics .....	75
4.3.2	Recuit simulé continu pour l'approximation de l'histogramme .....	76
4.3.2.1	Principe.....	76
4.3.2.2	Les critères d'arrêt utilisés .....	77
4.3.3	Complexité de l'algorithme CCS.....	78
4.4	Réglage des paramètres et résultats.....	78
4.4.1	Réglage des paramètres .....	78
4.4.2	Résultats de l'approximation d'un histogramme.....	78
4.4.3	Quelques exemples de segmentations .....	79
4.4.4	Exemples de segmentations d'images IRM.....	81
4.5	Conclusion .....	81
<b>Chapitre 5 : Application de la dérivation non-entière et de l'approche non-Pareto à la segmentation.....</b>		<b>82</b>
5.1	Introduction.....	82
5.2	Formalisme de la dérivation non-entière (DNE) .....	83
5.2.1	Formalisme de la DNE-1D.....	83
5.2.2	Extension de la dne-1D au cas 2D .....	84
5.3	Propriétés de l'histogramme et de l'image différenciée.....	84
5.3.1	Changement d'échelle dans le cas 1D.....	84
5.3.2	Interprétation géométrique dans le cas de la DNE-2D .....	85
5.3.3	Valeur moyenne et autocorrélation.....	87
5.4	Algorithmes de segmentation .....	87
5.4.1	Segmentation par DNE-1D .....	87
5.4.2	Segmentation par DNE-2D .....	88
5.4.3	Procédure d'optimisation de l'ordre de la DNE-1D et de la DNE-2D .....	88
5.4.3.1	Les critères de segmentation utilisés .....	88
5.4.3.2	Etapes d'optimisation.....	89
5.4.3.3	Critère de régularité géométrique .....	89
5.5	Résultats et discussion .....	91

---

5.5.1	Analyse de la sensibilité de l'ordre de la DNE .....	91
5.5.2	Comparaison.....	91
5.5.3	Exemples de segmentations.....	92
5.5.4	Analyse de la complexité algorithmique .....	93
5.5.4.1	Complexité de l'algorithme de segmentation DNE-1D .....	93
5.5.4.2	Complexité de l'algorithme de segmentation DNE-2D .....	94
5.5.4.3	Complexité de l'algorithme non-Pareto .....	94
5.5.5	Application aux images IRM .....	94
5.6	Conclusion .....	94
<b>Chapitre 6 : Adaptation de l'approche Pareto à la segmentation .....</b>		<b>97</b>
6.1	Introduction.....	97
6.2	Formulation du problème .....	97
6.3	L'algorithme NSGA-II .....	98
6.4	Critères de segmentation utilisés.....	99
6.5	Algorithme de segmentation (SMOP) .....	100
6.6	Résultats et discussion .....	101
6.7	Comparaison des performances.....	102
6.8	Application de la méthode SMOP aux images IRM .....	102
6.9	Conclusion .....	103
<b>Conclusions et perspectives .....</b>		<b>104</b>
<b>Annexes : Divers résultats de segmentations d'images à l'aide de nos divers algorithmes .....</b>		<b>107</b>
A1	Exemples de segmentations par recuit microcanonique hybridé .....	107
A2	Exemples de segmentations par la méthode 2DSEE .....	108
A3	Exemples de segmentations par la méthode CCS.....	119
A4	Exemples de segmentations par la méthode DNE-1D .....	110
A5	Exemple de segmentation par la méthode DNE-2D .....	112
A6	Exemples de segmentations par la méthode SMOP.....	113
<b>Références bibliographiques .....</b>		<b>114</b>



---

# INTRODUCTION GENERALE

---

Parmi les challenges du traitement d'images, celui de l'automatisation de l'interprétation des images biomédicales est certainement le plus étudié ces dernières années. De manière classique, un système de traitement automatique des images est constitué de deux niveaux. Le premier est consacré aux traitements numériques au sens large, tels que les opérations de codage, l'amélioration ou encore la restauration. Le second niveau est dédié aux opérations d'analyse symbolique des images, telles que la description, la reconnaissance ou encore l'interprétation, dans le but d'extraire une information.

La segmentation d'images est l'opération la plus importante dans un système de traitement des images, car elle est située à l'articulation entre le traitement et l'analyse des images. De ce fait, nombre de chercheurs ont travaillé sur le développement de méthodes et d'algorithmes dédiés. De plus, elle constitue, depuis quelques années, un axe important de recherche. Pour preuve, le nombre de travaux publiés traitant de ce problème est difficile à évaluer. C'est la conséquence de plusieurs éléments : la diversité des images, la complexité du problème, l'évolution des machines de calcul, et une évaluation des résultats assez empirique.

En effet, cette opération permet le passage d'une représentation en intensité lumineuse (niveaux de gris) à une représentation symbolique (appartenance des pixels à différentes classes de région). L'intérêt de la segmentation est de partitionner une image en plusieurs régions homogènes, au sens d'un critère fixé *a priori*. De nombreux critères de segmentation existent ; suivant le domaine d'application et le type d'images traitées, le critère prendra en compte le niveau de gris, la texture, la couleur, le mouvement, ou encore la distance. L'intérêt de disposer de régions homogènes est de fournir des données simplifiées, qui facilitent la tâche d'un système de reconnaissance de formes, ou autre système d'extractions de primitives, qui réalise une classification des images.

D'un point de vue algorithmique, la segmentation consiste à attribuer à chaque pixel de l'image un label d'appartenance à une région donnée. Cette classification des pixels peut être réalisée en mode supervisé ou en mode non supervisé. Dans le premier cas, le nombre de régions et leurs caractéristiques sont fournis par l'utilisateur, la segmentation est effectuée en déterminant, pour chaque site, la classe dont les propriétés se rapprochent le mieux de celles observées en ce site. Dans le second cas, le problème est plus complexe, car les informations nécessaires au processus de classification des pixels sont à fournir de manière complètement automatique. D'où la difficulté de ce type d'approche. La segmentation ainsi définie est un domaine vaste, pour lequel de nombreuses approches sont utilisées.

Toutes les approches de la segmentation visent à extraire des indices visuels. Après des années de recherche sur la méthode optimale, les chercheurs ont conclu à l'inexistence d'une segmentation idéale et d'un critère unique suffisant pour segmenter toutes les images. On peut même démontrer que le problème est mal posé, au sens de Hadamard. En effet, l'unicité de la segmentation d'une

image n'existe pas dans la plupart des cas. Une bonne méthode de segmentation est donc celle qui permettra d'arriver à une meilleure interprétation de l'image segmentée.

Les "métaheuristiques" d'optimisation sont des algorithmes généraux d'optimisation applicables à une grande variété de problèmes. Elles sont apparues à partir des années 80, dans le but de résoudre au mieux des problèmes d'optimisation. Dans la pratique, trois types de problèmes d'optimisation sont souvent rencontrés : les problèmes combinatoires (discrets), les problèmes continus (à variables continues), et les problèmes mixtes. L'exemple le plus répandu en optimisation combinatoire est celui du voyageur de commerce. En optimisation continue, un exemple simple est celui de la recherche des paramètres d'un modèle numérique pour approcher au mieux des données réelles. Le dernier type est celui des problèmes mixtes, qui comportent à la fois des variables discrètes et des variables continues.

Les métaheuristiques s'efforcent de résoudre tout type de problème d'optimisation. Elles sont caractérisées par leur caractère stochastique, ainsi que par leur origine discrète. Elles sont inspirées par des analogies avec la physique (recuit simulé, recuit microcanonique), avec la biologie (algorithmes évolutionnaires) ou encore l'éthologie (colonies de fourmis, essaims particulaires). Cependant, elles ont l'inconvénient d'avoir plusieurs paramètres à régler. Il est à souligner que les métaheuristiques se prêtent à toutes sortes d'extensions, notamment en optimisation multiobjectif.

L'application des métaheuristiques en traitement d'images a connu un engouement particulier ces dernières années, grâce aux avancées technologiques en matière de calcul des machines. L'ouvrage de Siarry *et al.* (Siarry, et al., 2007), paru récemment, regroupe de nombreuses applications des métaheuristiques en traitement des images. Plus particulièrement, les métaheuristiques sont très utilisées ces dernières années en segmentation d'images. A titre d'exemple, nous citons le recuit simulé dans le cas de l'approche markovienne, et les essaims particulaires pour l'approche par analyse d'histogramme.

Cette thèse a été préparée au sein du Laboratoire *Images, Signaux et Systèmes Intelligents* (LiSSi, E.A. 3956) de l'Université de Paris 12. Une des deux équipes de ce laboratoire est principalement orientée vers l'imagerie et le traitement du signal en génie biologique et médical. Ce travail a été effectué sous la direction du Professeur Patrick SIARRY, Directeur du LiSSi et responsable de la thématique "Optimisation", et co-encadré par Hamouche OULHADJ, maître de conférences et membre de l'équipe "imagerie". Ont participé à mon travail de doctorat des stagiaires : en dernière année d'école d'ingénieurs (Alexandre Minery), en DEA (Yasser Aidarous), ainsi qu'en deuxième année d'école d'ingénieurs (Sophie Roman).

L'objet de la présente thèse est d'évaluer l'adaptabilité des métaheuristiques et leurs avantages pour résoudre le problème de la segmentation d'images. En particulier, l'amélioration des performances des méthodes de segmentation, et la possibilité de les utiliser pour segmenter différents types d'images. Les objectifs que nous nous sommes fixés consistent à contribuer à l'élaboration de nouveaux critères de segmentation, et à l'évaluation de la qualité de la segmentation. Ensuite, nous avons reformulé le problème de la segmentation d'images comme un problème d'optimisation multiobjectif. Enfin, nous nous sommes attachés à adapter les métaheuristiques en utilisant plusieurs critères simultanément pour segmenter les images.

Les applications étudiées ont porté sur des images diverses (images infrarouges, images microscopiques, images par résonance magnétique (IRM) cérébrales, angiographies rétiniennes, etc.), dont la plupart intéressent le monde biomédical. Parmi ces images, l'accent est mis sur les IRM. Ce choix est motivé par le fait que les cliniciens ont souvent recours à ces images afin d'étudier les

maladies cérébrales, notamment la maladie d'Alzheimer, et par suite d'établir un diagnostic fiable, permettant de choisir une thérapeutique appropriée. Toutefois, l'observation uniquement qualitative de l'image est souvent insuffisante pour le diagnostic précis des pathologies, ainsi que pour le suivi de leur évolution. Afin de pallier à ces problèmes, nous nous sommes intéressés, plus particulièrement, à celui de la segmentation des ventricules cérébraux.

Ce rapport de thèse est scindé en trois parties. La première est consacrée à l'étude bibliographique de la segmentation d'images et des métaheuristiques. Elle fait l'objet des deux premiers chapitres. La deuxième partie, sujet du troisième chapitre, est une contribution à la segmentation d'images en optimisation mono-objectif. Enfin, la dernière partie regroupe le quatrième, cinquième et sixième chapitre. Elle est dédiée à l'approche multiobjectif du problème de la segmentation d'images.

Dans le premier chapitre, nous présentons un état de l'art des méthodes de segmentation d'images. L'accent est mis sur le seuillage d'images, notamment les méthodes avec lesquelles nous comparerons les performances de nos algorithmes.

Le deuxième chapitre porte sur les métaheuristiques et la segmentation d'images. Dans ce chapitre, nous reformulons le problème de segmentation d'image sous la forme d'un problème d'optimisation mono-objectif ou multiobjectif. Ensuite, nous présentons les différentes approches d'optimisation multiobjectif et les métaheuristiques que nous avons étudiées dans le cadre de cette thèse.

Le but du troisième chapitre est d'estimer l'adaptabilité des métaheuristiques pour résoudre la segmentation des images en optimisation mono-objectif. Nous exposons notre contribution à la segmentation d'images en mono-objectif via de nouveaux critères de segmentation, que nous avons développés. Ces critères sont au nombre de trois : *l'entropie exponentielle à deux dimensions*, *l'entropie survivante exponentielle à deux dimensions*, et *la variance intraclasse biaisée*. Les métaheuristiques utilisées dans ce chapitre sont le recuit microcanonique, le recuit simulé étendu aux problèmes à variables continues, le recuit microcanonique hybridé, l'optimisation par essaim particulaire (OEP), l'OEP adaptatif, et les algorithmes de colonies de fourmis. Une quantification de l'évaluation des résultats sera également traitée via la segmentation d'images synthétiques, dont l'image "vérité terrain" est connue *a priori*.

Dans le quatrième chapitre, nous présentons une nouvelle méthode de segmentation basée sur l'approche agrégation de fonctions objectifs, transformant le problème difficile de la segmentation multiobjectif en un problème mono-objectif. Dans cette approche, nous hybridons trois critères de segmentation, deux sont rencontrés dans la classe des méthodes paramétriques et un troisième dans la classe des méthodes non paramétriques. La métaheuristique utilisée est le recuit simulé, que nous avons utilisé sous deux formes : une forme adaptative combinatoire, et l'autre adaptée à l'optimisation continue. L'évaluation des performances, en fonction des types d'histogrammes, et l'utilisation d'images test synthétiques seront exposées.

Le cinquième chapitre est consacré à l'approche non-Pareto de la segmentation d'images. Le principe de base consiste à optimiser plusieurs critères simultanément, et par delà, à sélectionner la meilleure solution. Nous présentons la dérivation d'ordre non-entier, que nous avons appliquée pour réduire l'espace de recherche des solutions. Afin d'automatiser la sélection de la solution optimale, nous proposons un nouveau critère d'évaluation de la qualité de la segmentation, basé sur la régularité géométrique de l'image segmentée.

Le sixième chapitre porte sur l'application de l'optimisation multiobjectif selon l'approche Pareto à la segmentation d'images. Nous exploitons un algorithme génétique basé sur la notion de dominance,

appelé NGS-II. L'avantage de cet algorithme est son faible nombre de paramètres à régler et la possibilité d'utiliser un nombre élevé de critères, sans que ses performances ne soient dégradées.

L'approche que nous présentons est basée sur trois critères de segmentation ; la solution optimale est celle qui domine toutes les autres.

Enfin, dans la conclusion générale du manuscrit, nous récapitulons nos contributions et proposons des perspectives sur la base des travaux effectués.

---

# CHAPITRE 1

## ETAT DE L'ART DES METHODES DE SEGMENTATION D'IMAGES

---

### 1.1 INTRODUCTION

---

Ces dernières années, le problème de la segmentation a tant retenu l'attention des chercheurs, que l'on ne compte plus le nombre de publications dans ce domaine. Par conséquent, plusieurs classifications des méthodes de segmentation ont été proposées dans les différents ouvrages dédiés au traitement des images. Nous avons opté pour la classification suivante :

- méthodes de segmentation par seuillage,
- méthodes de segmentation par classification,
- méthodes de segmentation basées régions,
- méthodes de segmentation basées contours.

Dans ce chapitre, nous décrivons brièvement ces différentes classes de méthodes. Toutefois, l'accent est mis sur les méthodes de segmentation par seuillage et par classification.

### 1.2 LE SEUILLAGE

---

Le seuillage est la technique de segmentation la plus répandue pour extraire les objets du fond de l'image. Les avantages de cette technique sont sa facilité de mise en œuvre et son efficacité dans les systèmes en temps réel (Cocquerez, et al., 1995). Cette technique est basée sur l'hypothèse que les objets peuvent être distingués par leur niveau de gris. Le seuil optimal est celui qui permet de séparer les objets entre eux ou différents objets du fond. Le challenge de cette technique est de trouver ce seuil optimal de manière automatique.

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

Soit  $f$  l'image originale et  $g$  l'image segmentée, la classification de chaque pixel est définie par :

$$g(x, y) = k \text{ si } T_k \leq f(x, y) < T_{k+1} \text{ pour } k = 0, \dots, m$$

où  $x, y$  sont les coordonnées du pixel,  $m$  est le nombre de classes et  $T_0, \dots, T_{m+1}$  sont les différents seuils de segmentation. A titre d'exemple, dans le cas d'un seuillage simple (appelé aussi binarisation) ( $m=2$ ), les valeurs attribuées aux pixels sont 0 ou 1. Par ailleurs, un seuillage est dit

partiel si les valeurs attribuées aux pixels sont 0 ou  $f(x, y)$ , ce qui permet de voir sur l'image en niveaux de gris les valeurs des pixels appartenant à la classe A (voir figure 1.1).

Le seuillage manuel d'une image comporte quatre étapes :

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

Une segmentation optimale par seuillage nécessite donc une localisation optimale des seuils de segmentation dans l'histogramme. En partant de l'hypothèse du seuillage, si l'histogramme possède plusieurs modes (pics), les seuils optimaux sont situés entre ces derniers. Cependant, dans la pratique, ce cas de figure ne se présente pas fréquemment, ce qui rend l'opération de segmentation plus complexe (par exemple, dans le cas où l'histogramme possède un seul mode). Afin de résoudre ce problème de seuils optimaux, une multitude de méthodes ont été développées. Les plus répandues d'entre elles sont rassemblées dans un recueil récemment publié (Sezgin, et al., 2004).

La figure 1.1 illustre la méthode de segmentation par seuillage manuel appliquée à une image microscopique de bactéries.

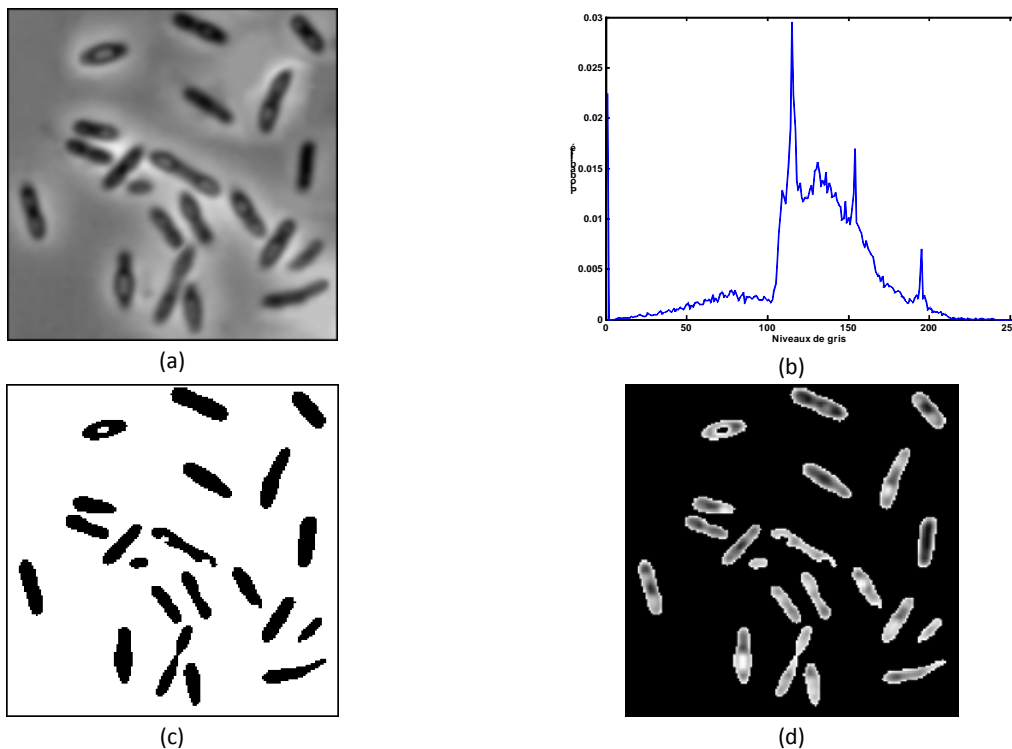


Figure 1. 1 : Exemple de seuillage manuel d'une image microscopique de bactéries. (a) Image originale de bactéries, (b) Histogramme de l'image, (c) Seuillage global de l'image ( $t=95$ ), (d) Seuillage partiel de l'image.

Dans la littérature, plusieurs classifications des méthodes de seuillage ont été proposées, mais la plupart des auteurs (Sezgin, et al., 2004; Otsu, 1979; Kittler, et al., 1986; Kapur, et al., 1985; Cheriet, et al., 1998; Pal, et al., 1991; Reddi, et al., 1984; Hou, et al., 2006; Ng, 2006; Lahfi, et al., 2006a) classent les méthodes de seuillage en deux classes : les méthodes paramétriques et les méthodes non paramétriques. Les méthodes non paramétriques ont la réputation d'être des méthodes robustes ; elles sont basées sur l'optimisation d'un ou plusieurs critères *a posteriori*, sans l'estimation de paramètre(s) ; la plus connue d'entre elles est la méthode d'Otsu (Otsu, 1979). Les méthodes

paramétriques sont basées sur l'hypothèse que les densités de probabilité des niveaux de gris des différentes classes sont gaussiennes, les seuils optimaux sont alors à l'intersection de celles-ci (Kittler, et al., 1986).

### 1.3 LES METHODES NON PARAMETRIQUES

Par définition, ces méthodes permettent de trouver le seuil optimal de segmentation sans aucune estimation de paramètres. Généralement, ces méthodes sont basées sur l'optimisation de critères statistiques. Les deux méthodes de référence dans cette classe sont la méthode d'Otsu (Otsu, 1979) et la méthode de Kapur *et al.* (Kapur, et al., 1985). La plupart des techniques parues ensuite sont basées sur l'un des deux principes.

#### 1.3.1 LA METHODE D'OTSU ET SES VARIANTES

Dans son papier original, Otsu décrit trois critères discriminants possibles : la variance intraclasse, la variance interclasse et la variance totale. Les trois critères sont équivalents et, suivant la situation, l'un d'eux peut être choisi. Du fait de la simplicité de la mise en œuvre de la variance interclasse, souvent les auteurs la maximisent pour segmenter des images.

Dans cette méthode, le calcul des différents critères est basé sur la densité de probabilité des différents pixels de l'image, qui est obtenue en normalisant l'histogramme de l'image :

$$p_i = \frac{h(i)}{\sum_{j=0}^{L-1} h(j)} \quad (1.1)$$

avec

$$p_i \geq 0, \text{ et } \sum_{i=0}^{L-1} p_i = 1 \quad (1.2)$$

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

Dans le cas d'une binarisation, le seuil optimal  $t^*$  est celui qui maximise le ratio de la variance interclasse à la variance totale :

$$t^* = \underset{th}{\text{Arg max}} \frac{\sigma_B^{2(t)}}{\sigma_T^2} \quad (1.3)$$

où la variance interclasse est définie par :

$$\sigma_B^{2(t)} = P_t(P_t - 1)(\mu_1 - \mu_0)^2 \quad (1.4)$$

et

$$\sigma_T^2 = \sum_{i=1}^{L-1} p_i (i - \mu_T)^2 \quad (1.5)$$

où

$$\mu_0 = \sum_{i=1}^t i \cdot \frac{p_i}{P_t} \quad \mu_1 = \sum_{i=t+1}^{L-1} i \cdot \frac{p_i}{1 - P_t} \quad \mu_T = \mu_0 + \mu_1$$

$$P_t = \sum_{i=0}^t p_i \quad 1 - P_t = \sum_{i=t+1}^{L-1} p_i \quad (1.6)$$

Comme la variance totale associée à un histogramme d'une image est constante, le problème se résume à maximiser la variance interclasse. La figure 1.2 montre un exemple d'application de la méthode sur une image de grains de riz dont l'histogramme est multimodal (figure 1.2 (b)), le seuil optimal obtenu est de 125.

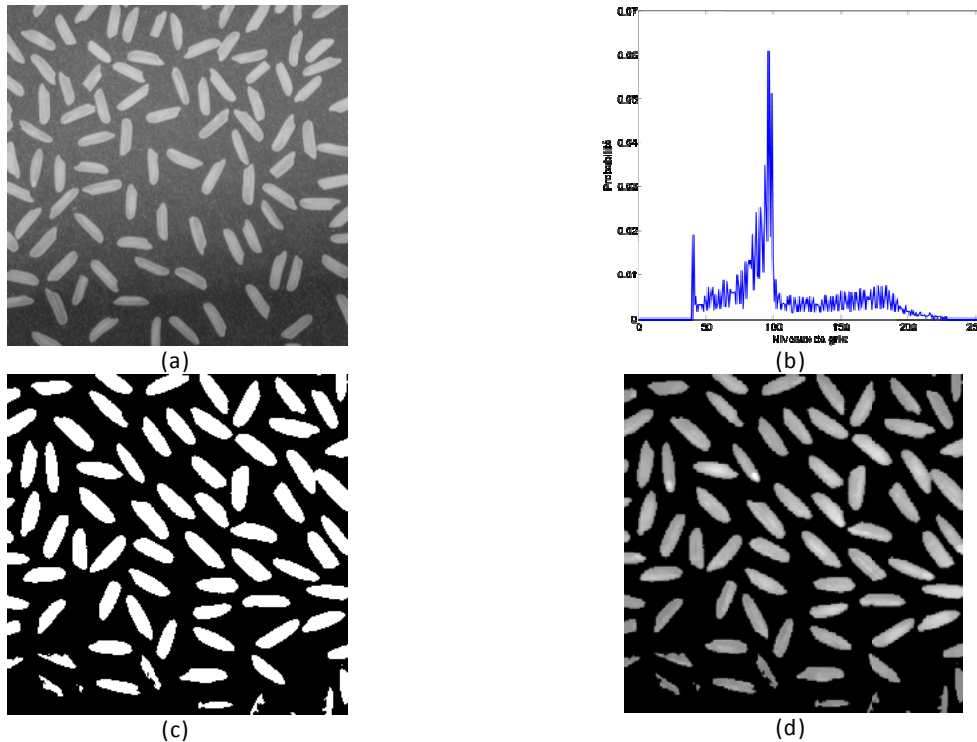


Figure 1. 2 : Illustration de la méthode d'Otsu. (a) Image originale de grains de riz. (b) Histogramme de l'image, (c) Seuillage global de l'image ( $t=125$ ), (d) Seuillage partiel de l'image.

Comme la méthode d'Otsu consiste à regrouper les pixels en deux classes, l'efficacité de cette dernière n'est prouvée que lorsqu'un seul type d'objet à segmenter est présent dans l'image. Afin de pallier à ce problème, Cheriet *et al.* (Cheriet, et al., 1998) ont proposé d'appliquer la méthode d'Otsu de façon récursive, de manière à segmenter un seul type d'objet à la fois dans l'image. A la fin, la segmentation complète se fait par regroupement de tous les objets segmentés. Parmi les problèmes de la méthode d'Otsu, il y a celui de l'unicité du seuil qui maximise le critère (1.3). Pour résoudre ce problème, Reddi *et al.* (Reddi, et al., 1984) ont établi la condition pour laquelle la variance interclasse a un seul maximum. Hou *et al.* (Hou, et al., 2006) ont publié récemment une analyse pertinente de la méthode d'Otsu. Dans (Ng, 2006), Ng a défini un nouveau critère inspiré de la méthode d'Otsu, appelé "*valley emphasis criterion*", où le seuil optimal est celui qui le maximise. Ce critère est défini par :

$$t^* = \underset{0 \leq t < L}{\text{Arg max}} \left\{ (1 - \lambda_t)(P_t \mu_1^2(t) + (1 - P_t) \mu_2^2(t)) \right\} \quad (1.7)$$

où  $\lambda_t$  est la plus faible probabilité d'occurrence dans l'histogramme. La modification apportée par rapport au critère d'Otsu (1.4) se résume dans l'introduction du coefficient  $\lambda_t$ .

Ces méthodes ont été appliquées dans beaucoup de domaines, notamment dans des applications biomédicales, comme la segmentation en temps réel des images radiologiques (Lahfi, et al., 2006a;



Lahfi, 2006b), la segmentation d'images microscopiques (Jiang, et al., 2002) et la segmentation des images IRM (cf. section 3. 9) (Qiao, et al., 2007).

### 1.3.2 LE SEUILLAGE ENTROPIQUE

#### 1.3.2.1 SEUILLAGE A BASE DE L'ENTROPIE A UNE DIMENSION

Le seuillage entropique, ou la méthode de Kapur *et al.* (Kapur, et al., 1985), est basé sur le principe de la maximisation de l'entropie totale de Shannon. Dans le cas de la segmentation d'une image en deux classes (A et B), la méthode suppose que les objets et le fond possèdent deux densités de probabilité indépendantes. Le seuil optimal de segmentation est celui qui permet de maximiser l'entropie totale de l'image partitionnée. Les densités de probabilité des deux classes (A et B) sont définies par :

$$\text{Classe A: } \frac{P_1}{P_t}, \frac{P_2}{P_t}, \dots, \frac{P_t}{P_t} \quad (1. 8)$$

$$\text{Classe B: } \frac{P_{t+1}}{1-P_t}, \frac{P_{t+2}}{1-P_t}, \dots, \frac{P_L}{1-P_t} \quad (1. 9)$$

où  $L$  est le nombre total de niveaux de gris et  $P_t$  la probabilité cumulée définie dans (1. 6).

Les entropies des deux classes A et B sont définies par :

$$H_A(t) = - \sum_{i=1}^t \frac{P_i}{P_t} \log_2 \frac{P_i}{P_t} \quad (1. 10)$$

$$H_B(t) = - \sum_{i=t+1}^L \frac{P_i}{1-P_t} \log_2 \frac{P_i}{1-P_t} \quad (1. 11)$$

et l'entropie totale s'écrit :

$$H_T(t) = H_A(t) + H_B(t) \quad (1. 12)$$

Le premier problème avec cette méthode, souligné par Pal *et al.* (Pal, et al., 1991), est que l'entropie de Shannon n'est pas définie pour les densités de probabilité incluant des probabilités nulles. Pal *et al.* ont alors proposé une nouvelle définition de l'entropie basée sur un gain exponentiel d'information :

$$H_T(t) = - \sum_{i=1}^t \frac{P_i}{P_t} e^{1-\frac{P_i}{P_t}} - \sum_{i=t+1}^L \frac{P_i}{1-P_t} e^{1-\frac{P_i}{1-P_t}} \quad (1. 13)$$

En résumé, le problème de seuillage entropique peut être formulé ainsi :

$$t^* = \underset{1 \leq t < L}{\text{Arg max}} \{H_T(t)\} \quad (1. 14)$$

La figure 1.3 montre l'application de la méthode de Kapur *et al.* sur l'image de grains de riz de la figure 1.2 (a). Le seuil optimal obtenu est de 118. D'un point de vue visuel, nous remarquons que, dans le cas de cette image, la méthode de Kapur *et al.* produit une segmentation de meilleure qualité que la méthode d'Otsu.

Cette approche a ouvert le champ à l'utilisation de nouvelles mesures d'informations pour mieux

segmenter les images. Une brève description des différentes variantes de l'entropie utilisées pour segmenter des images est présentée dans (Sezgin, et al., 2004). Ci-dessous, nous nous contenterons de présenter les expressions de l'entropie récemment proposées dans la littérature.



Figure 1. 3 : Segmentation de l'image originale (figure 1.2) de grains de riz. (a) Seuillage global de l'image ( $t=118$ ), (b) Seuillage partiel de l'image.

- **L'entropie de Tsallis**

L'entropie de Tsallis (Sahoo, et al., 2006) d'un système physique non extensif est définie par :

$$S_q = \frac{1 - \sum_{i=1}^k p_i^q}{q-1} \quad (1.15)$$

où  $q$  est un paramètre réel, appelé indice entropique et  $k$  est le nombre de possibilités du système. L'application de cette mesure d'entropie pour la segmentation d'images, dans le cas d'une segmentation à deux classes (A et B), consiste à maximiser l'entropie totale de Tsallis définie comme suit:

- l'entropie de Tsallis de la classe A:

$$S_A^q(t) = \frac{1 - \sum_{i=1}^t \left( \frac{p_i}{P_t} \right)^q}{q-1} \quad (1.16)$$

- l'entropie de Tsallis de la classe B :

$$S_B^q(t) = \frac{1 - \sum_{i=t+1}^L \left( \frac{p_i}{1-P_t} \right)^q}{q-1} \quad (1.17)$$

D'où l'entropie totale de Tsallis:

$$S_T^q(t) = \left[ S_A^q(t) + S_B^q(t) + (1-q)S_A^q(t)S_B^q(t) \right] \quad (1.18)$$

L'étude de la variation de la qualité de la segmentation, en fonction de  $q$ , a montré que l'obtention d'une bonne segmentation est conditionnée par la valeur de  $q$  (Portes, et al., 2004).

- **L'entropie de Renyi**

De la même manière, l'entropie de Renyi a été aussi utilisée pour la segmentation d'images (Prasanna, et al., 2004) (Prasanna, et al., 2004). Elle est définie pour une classe (A) et un seuil de segmentation ( $t$ ) par (1.18):

$$R_A^\alpha(t) = \frac{1}{1-\alpha} \log \sum_{i=1}^t (p_i)^\alpha \quad (1.19)$$

où  $\alpha \neq 1$  est un paramètre réel, appelé ordre de l'entropie. L'entropie totale à maximiser, dans le cas de deux classes A et B, s'écrit :

$$R_T^\alpha(t) = [R_A^\alpha(t) + R_B^\alpha(t)] \quad (1.20)$$

### 1.3.2.2 ENTROPIE A DEUX DIMENSIONS

L'idée de base de cette méthode revient à Abutaleb *et al.* (Abutaleb, 1989) et Pal *et al.* (Pal, et al., 1989), qui ont étendu l'entropie de Shannon au cas bidimensionnel et l'ont appliquée à la segmentation d'images. Pal *et al.* ont proposé d'ajouter de l'information à la distribution spatiale des pixels, afin de segmenter les images. Cette information spatiale consiste en un histogramme à deux dimensions, où sont recueillis soit la matrice de cooccurrence des pixels, soit une image filtrée, ou tout autre élément (Sezgin, et al., 2004; Beghdadi, et al., 1995; Brink, A.D., 1996).

#### 1.3.2.2.1 CALCUL D'UN HISTOGRAMME A DEUX DIMENSIONS

Afin d'expliquer le principe de calcul d'un histogramme à deux dimensions, nous considérons le cas où l'information spatiale exploitée est l'image moyennée.

Le calcul de l'histogramme 2D s'effectue en trois étapes. Dans la première étape, nous calculons la moyenne des niveaux de gris au voisinage de chaque pixel  $(x, y)$ . Soit  $g(x, y)$  cette valeur moyenne. Pour un voisinage 3x3,  $g(x, y)$  s'écrit :

$$g(x, y) = \frac{1}{9} \sum_{i=-1}^1 \sum_{j=-1}^1 f(x+i, y+j) \quad (1.21)$$

où  $f(x, y)$  représente l'image originale en niveaux de gris. Afin d'éviter le problème des bords, nous excluons des calculs les premières et dernières lignes et colonnes de l'image. Dans la deuxième étape, un histogramme  $h(i, j)$  est calculé à partir de  $f(x, y)$  et  $g(x, y)$  :

$$h(i, j) = \text{Card}\{f(x, y) = i \cap g(x, y) = j\} \quad (1.22)$$

où  $i, j \in \{0, 1, 2, \dots, 255\}$ .

Enfin, un histogramme normalisé à deux dimensions est calculé :

$$p_{ij} = \frac{h(i, j)}{\text{nombre de pixels dans l'image}} \quad (1.23)$$

Le seuil de segmentation est un vecteur  $(t, s)$ , où  $t$  est le seuil pour l'image originale de  $f(x, y)$  et  $s$  le seuil pour l'image  $g(x, y)$  des voisinages moyennés.

A partir de l'histogramme 2D normalisé, une surface à deux pics peut être dessinée. L'objet et le fond correspondent aux deux pics et peuvent être séparés par le vecteur  $(t, s)$ . Ce vecteur  $(t, s)$  nous permet de diviser l'histogramme en quatre quadrants ou classes de pixels (Figure 1.4). Le premier représente le fond de l'image, le deuxième les objets. Les quadrants 3 et 4 correspondent respectivement au bruit et aux contours, ils seront ignorés dans nos calculs des seuils optimaux.

Les densités de probabilité des quadrants 1 et 2 sont notées  $P1$  et  $P2$ . Pour un vecteur  $(t, s)$  donné,  $P1$  et  $P2$  sont définies comme suit:

$$P1 = \sum_{i=0}^s \sum_{j=0}^t p_{ij}, \quad P2 = \sum_{i=s+1}^{L-1} \sum_{j=t+1}^{L-1} p_{ij} \quad (1.24)$$

La figure 1.5 (b) présente un histogramme à deux dimensions de l'image microscopique de la figure 1.5 (a).

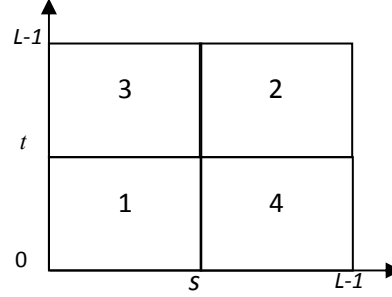


Figure 1.4. Les classes de pixels de l'histogramme à deux dimensions. 1 : fond, 2 : objets, 3 : bruit, 4 : contours.

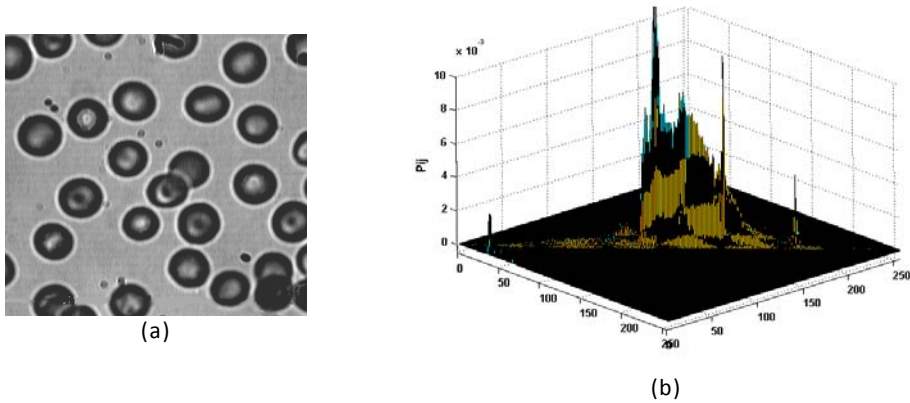


Figure 1.5 : Histogramme à deux dimensions. (a) Image microscopique, (b) Histogramme à deux dimensions.

### 1.3.2.2.2 EXTENSION DE L'ENTROPIE AU CAS BIDIMENSIONNEL

Dans le cas d'une segmentation en deux classes, les extensions des différentes expressions de mesure de l'entropie au cas bidimensionnel sont définies par les relations:

- L'entropie totale de Shannon à deux dimensions (Qiao, et al., 2007; Abutaleb, 1989) :

$$H_T(t,s) = \sum_{i=0}^s \sum_{j=0}^t \frac{p_{ij}}{P1} \log \frac{p_{ij}}{P1} + \sum_{i=s+1}^{L-1} \sum_{j=t+1}^{L-1} \frac{p_{ij}}{P2} \log \frac{p_{ij}}{P2} \quad (1.25)$$

- L'entropie de Tsallis à deux dimensions pour une image (Sahoo, et al., 2006):

$$S_T^q(t,s) = \frac{1}{q-1} \left[ 1 - \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} p(i,j)^q \right] \quad (1.26)$$

où  $q \neq 1$  est un paramètre réel positif.

L'expression de l'entropie totale de Tsallis à deux dimensions est alors la suivante:

$$S_T^q(t,s) = \left[ S_1^q(t,s) + S_2^q(t,s) + (1-q)S_1^q(t,s)S_2^q(t,s) \right] \quad (1.27)$$

Dans (Abutaleb, 1989), les auteurs ont étudié les variations de la qualité de la segmentation en fonction de l'ordre de l'entropie et sont parvenus à la conclusion que l'ordre  $q=0,8$  est l'ordre optimal de l'entropie de Tsallis.

- L'entropie de Renyi à deux dimensions de chaque classe (Prasanna, et al., 2004) (Prasanna, et al., 2004) est définie ainsi:

$$R_1^\alpha(t,s) = \frac{1}{1-\alpha} \log \sum_{i=0}^t \sum_{j=0}^s \left( \frac{P_{ij}}{P1} \right)^\alpha \quad (1.28)$$

$$R_2^\alpha(t,s) = \frac{1}{1-\alpha} \log \sum_{i=s+1}^{L-1} \sum_{j=t+1}^{L-1} \left( \frac{P_{ij}}{P2} \right)^\alpha \quad (1.29)$$

d'où l'entropie totale de Renyi à deux dimensions:

$$R_T^\alpha(t,s) = \left[ R_1^\alpha(t,s) + R_2^\alpha(t,s) \right] \quad (1.30)$$

Dans (Prasanna, et al., 2004), les auteurs ont aussi étudié les variations de la qualité de la segmentation en fonction de l'ordre de l'entropie et ont conclu qu'avec un ordre  $\alpha=0,5$  ou  $0,7$  la méthode permet, suivant le type d'image, d'avoir une segmentation correcte.

## 1.4 LES METHODES PARAMETRIQUES

Les méthodes paramétriques de segmentation d'image supposent que les différentes classes de l'image suivent une certaine fonction de densité de probabilité. Généralement, ces fonctions de densités de probabilité (fdp) sont supposées suivre un modèle gaussien. En partant d'une approximation de l'histogramme de l'image par une combinaison linéaire de gaussiennes, les seuils optimaux sont à l'intersection de ces dernières. La plupart de ces méthodes sont présentées dans (Sezgin, et al., 2004; Feng, et al., 2005; Sahoo, et al., 1988; Gonzalez, et al., 2002; Gonzalez, et al., 2004; Maitre, 2003; Cocquerez, et al., 1995). Dans cette partie, nous allons seulement présenter le principe de base de ces méthodes, qui s'exécutent en deux étapes : la première consiste à estimer les paramètres et la deuxième à trouver les seuils de segmentation.

### 1.4.1 APPROXIMATION DE L'HISTOGRAMME PAR FONCTIONS DE DENSITE DE PROBABILITE

Pour un histogramme multimodal  $h(x)$  d'une image donnée, où  $x$  prend sa valeur dans l'échelle des niveaux de gris, la technique la plus utilisée pour trouver les seuils optimaux de segmentation consiste à reconstruire l'histogramme de l'image sous la forme de la somme de  $d$  fonctions de densité de probabilité (fdp). Dans le cas où ces fdp sont gaussiennes, le modèle de l'histogramme peut alors s'écrire:

$$h(x) = \sum_{i=1}^d \frac{P_i}{\sqrt{2\pi}\sigma_i} e^{\left[ -\frac{(x-\mu_i)^2}{\sigma_i^2} \right]} \quad (1.31)$$

où  $P_i$  représente la probabilité a priori,  $\mu_i$  la moyenne et  $\sigma_i$  la variance du mode  $i$ .

Pour un histogramme donné, les paramètres optimaux sont ceux qui minimisent l'erreur quadratique moyenne de reconstruction :

$$J = \frac{1}{L} \sum_{i=0}^{L-1} [h(i) - h(\Theta, i)]^2 \quad (1.32)$$

où  $J$  est la fonction objectif à minimiser,

$$\Theta = \{P_i, \mu_i, \sigma_i; j = 1, 2, \dots, d\} \quad (1.33)$$

$i \in [0, L-1]$ ,  $L$  désignant le nombre total de niveaux de gris présents dans l'image originale.

#### 1.4.2 CALCUL DES SEUILS DE SEGMENTATION

Dans le cas bimodal  $d=2$ , le seuil optimal est déterminé par la minimisation de la probabilité d'erreur d'affectation d'un niveau de gris  $x$  à la première classe, sachant qu'il appartient à la deuxième classe:

$$E(t) = P_1 \int_{-\infty}^t p_2(x) dx + P_2 \int_t^{+\infty} p_1(x) dx \quad (1.34)$$

où  $t$  est le seuil,  $p_i(x)$  est la fonction de densité de probabilité du mode  $i$  et  $P_i$  sont les amplitudes des gaussiennes au point  $\mu_i$ . Pour un histogramme donné, l'algorithme qui permet de trouver le seuil optimal est détaillé dans (Cocquerez, et al., 1995).

Les méthodes présentées dans la littérature, qui sont basées sur le même principe, diffèrent selon la méthode d'optimisation utilisée, ou selon les critères utilisés pour trouver les seuils optimaux (Weszka, et al., 1979; Synder, et al., 1990; Yin, 1999; Zahara, et al., 2004; Ridler, et al., 1978; Kurita, et al., 1992). Un exemple illustrant ce type de méthode est donné sur la figure 1.6. L'image originale de l'avion (figure 1.6 (a)) est tirée de la base de données de Berkeley (Berkeley, 2007). La méthode utilisée est décrite dans (Synder, et al., 1990), où l'algorithme d'optimisation utilisé est l'algorithme des gradients conjugués, avec une initialisation manuelle.

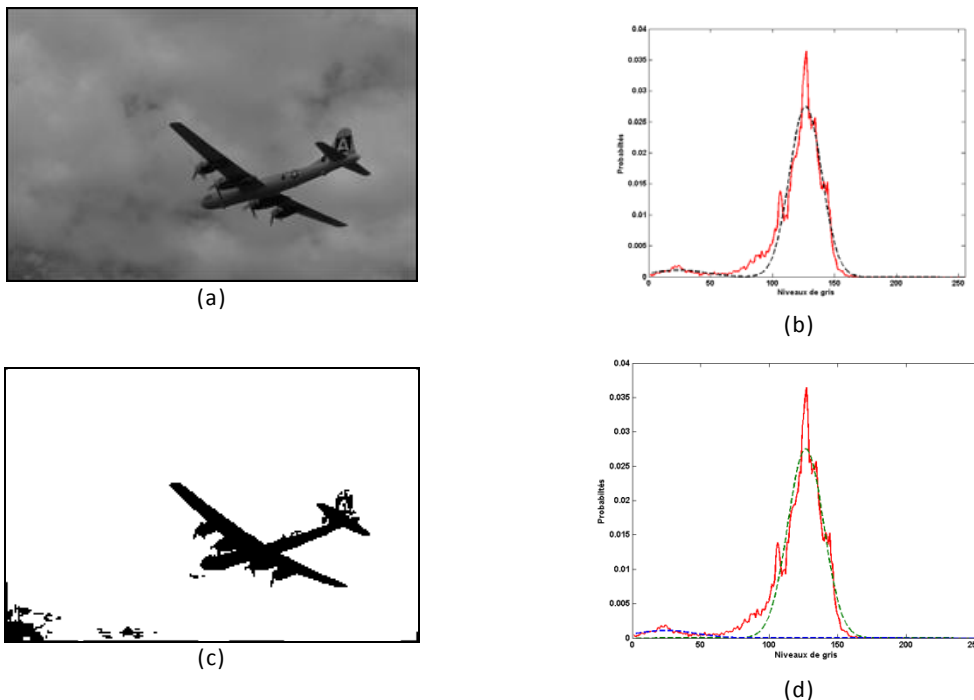


Figure 1.6 : Seuillage paramétrique. (a) Image originale, (b) Histogramme original et approché par deux gaussiennes, (c) Image seuillée ( $t=79$ ), (d) Approximation de l'histogramme par deux gaussiennes.

## 1.4.3 ALGORITHME ESPERANCE-MAXIMISATION (EM)

L'algorithme EM a été développé par Dempster *et al.* en 1977 (Dempster, et al., 1977) et a été utilisé dans de nombreux domaines, notamment en segmentation d'images (Fwu, et al., 1997; Bazi, et al., 2007). Nous allons présenter le principe de base de cet algorithme : pour plus de détails, voir (Dempster, et al., 1977).

- **Principe**

Nous nous plaçons dans le cas d'une approximation d'un histogramme d'une image par une combinaison de gaussiennes. Le problème est d'arriver à estimer les paramètres des différentes gaussiennes ( $\Theta$ ) qui approchent le mieux l'histogramme.

Par exemple, dans le cas de deux classes, c'est-à-dire d'un histogramme approché par deux gaussiennes (figure 1.7), le vecteur de paramètres à estimer est  $\Theta = \{P_1, \mu_1, \sigma_1, P_2, \mu_2, \sigma_2\}$  associé aux deux classes (objet et fond). Cette estimation est faite au sens de la maximisation du critère de maximum de vraisemblance  $L(X | \Theta)$ , défini par :

$$\Theta^* = \underset{\Theta}{\text{Arg max}} \{L(X | \Theta)\} \quad (1.35)$$

où

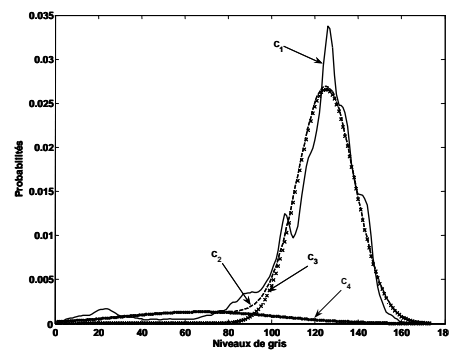
$$L(X | \Theta) = \sum_{x=0}^{L-1} h(x) \log [p(x | \Theta)] \quad (1.36)$$

La solution du problème ci-dessus ne peut pas être obtenue analytiquement, car le problème est intrinsèquement mal posé. L'algorithme EM va donc être utilisé afin d'y remédier. Cet algorithme est basé sur l'interprétation de  $X$  comme un vecteur de données, incomplet, où la partie qui manque est  $Y$  (l'image segmentée). En d'autres termes, la partie manquante peut être évaluée comme un ensemble de  $L$  étiquettes  $Z = \{z(x) : x \in [0, L-1]\}$ . Chaque étiquette  $z(x)$  de  $Z$  indique la composante (objet ou fond) qui est à l'origine de la réalisation  $x$ . Dans notre cas, l'ensemble  $Z$  est binaire :  $z(x) = \{z_1(x), z_2(x)\}$ , avec  $z_i(x) = 1$  si  $x$  appartient à la composante  $i$  et  $z_i(x) = 0$  dans le cas contraire. Le critère de segmentation final est :

$$L(X, Z, \Theta) = \sum_{i=1}^2 \sum_{x=0}^{L-1} z_i(x) h(x) \log [P_i p_i(x | (\mu_i, \sigma_i))] \quad (1.37)$$



(a)



(b)

Figure 1.7 : Seuillage par l'algorithme EM de l'image de la figure 1.6 (a). (a) Image seuillée ( $t=92$ ), (b) Histogramme original ( $C_1$ ) et approché ( $C_2$ ) par deux gaussiennes ( $C_3$  et  $C_4$ ).

L'algorithme EM consiste à itérer les deux étapes : Espérance et Maximisation, jusqu'à la convergence totale de l'algorithme. L'étape d'espérance va permettre de calculer les étiquettes

$z_i(x)$ , en utilisant les paramètres estimés en cours. L'étape de maximisation va permettre de mettre à jour l'estimation du vecteur  $\theta$ . La méthode est détaillée dans (Bazi, et al., 2007; Fwu, et al., 1997).

## 1.5 SEUILLAGE ADAPTATIF

Dans le cas où le fond de l'image n'est pas uniforme (variations de luminosité), le seuillage global ne permet pas d'en extraire les objets. Pour pallier à cette difficulté, la solution la plus utilisée en pratique est basée sur l'application d'un seuillage adaptatif, qui consiste à segmenter l'image avec différents seuils, qui varient en fonction de la position des pixels.

Le principe de base consiste à découper l'image en sous-images de tailles prédéfinies et à appliquer une technique de seuillage global à chacune de ces sous-images (Portes, et al., 2004; Gonzalez, et al., 2002).

La figure 1.8 montre un exemple d'une image synthétique avec un fond non uniforme. La figure 1.8 (b) présente le résultat de la segmentation de l'image dans la figure 1.8 (a) avec un seuil global. Comme nous pouvons le remarquer, l'objet ne peut être extrait avec un seul seuil. Le résultat de l'application d'un seuillage adaptatif, après un découpage de l'image originale en sous-images (figure 1.8 (c)), est présenté sur la figure 1.8 (d), où nous pouvons constater l'efficacité de cette méthode de seuillage dans ce cas de figure. Le problème avec cette approche est de définir le nombre optimal de sous-images.

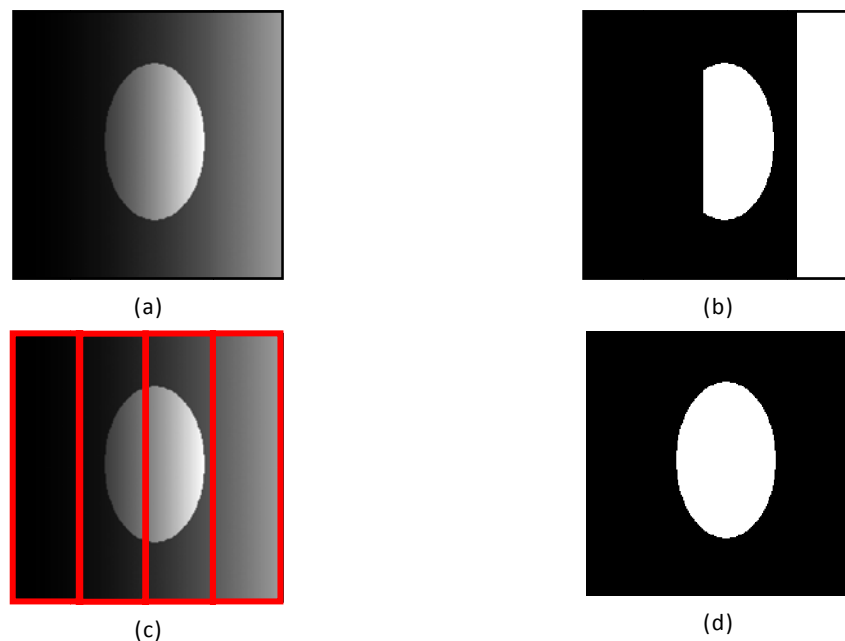


Figure 1.8 : Illustration du seuillage adaptatif. (a) Image originale avec un fond non uniforme, (b) Image segmentée avec un seuil global, (c) Image originale découpée en sous-images, (d) Image segmentée avec un seuillage adaptatif.

## 1.6 SEGMENTATION PAR CLASSIFICATION

Comme l'image est caractérisée essentiellement par son histogramme, la plupart des méthodes de classification ("*clustering*") s'appliquent sans difficulté majeure à la segmentation des images. Ces méthodes peuvent aussi être classées parmi les méthodes paramétriques de segmentation.

Dans le cas où le nombre de classes est connu *a priori*, nous parlons alors de classification supervisée ; dans le cas contraire, il s'agit de la classification non supervisée. Parmi les méthodes les



plus utilisées en segmentation d'images : l'algorithme des centres mobiles (nuées dynamiques ou *K-means*), les réseaux neuromimétiques et en particulier les cartes de Hopfield. Dans le paragraphe suivant, nous allons décrire seulement la méthode des centres mobiles (*K-means*) (Maitre, 2003; Monmarché, 2000).

L'algorithme des centres mobiles (*K-means*) est la technique de classification la plus simple (Algorithme 1.1). Cette technique utilise comme critère d'évaluation d'une partition la distance quadratique moyenne.

- **Principe**

Soit un nuage de points (une image) que nous voulons partitionner en  $K$  classes. Nous posons :

$x_j^{(i)}$  : le pixel  $j$  appartenant à la classe  $i$ ,

$y_i$  : le centroïde de la classe  $i$ ,

$L_i$  : le nombre de points de la classe  $i$ ,

$d(x_j^{(i)}, y_i)$  : la distance (mesure de distorsion) entre  $x_j^{(i)}$  et  $y_i$ ; dans le cas de l'algorithme *K-means*, c'est la distance Euclidienne.

$D_i$  : la distance totale de la classe  $i$ :

$$D_i = \sum_{j=1}^{L_i} d(x_j^{(i)}, y_i) \quad (1.38)$$

$D$  : la distorsion globale pour l'ensemble des vecteurs (inertie intraclasse):

$$D = \sum_{i=1}^K D_i \quad (1.39)$$

La classification optimale est celle qui permet de minimiser la distorsion totale  $D$  (1.39). La procédure d'optimisation doit tenir compte des hypothèses suivantes :

- Pour un ensemble donné de centroïdes, la classification qui minimise  $D$  (1.38) est celle pour laquelle chaque pixel est affecté à la classe dont le centroïde est le plus proche.
- Pour une classification donnée, il existe, pour chaque classe  $i$ , un pixel  $y_i$  qui minimise la distance totale de la classe  $D_i$ .

Algorithme 1. 1 : Algorithme des centres mobiles (*K-means*).

1. **Choisir** un nombre de classes  $k$ .
2. **Définir** une classification  $C$  aléatoire : Choisir les  $k$  centroïdes  $y_i$  de façon aléatoire dans l'espace  $D$  des niveaux de gris  $[0, L-1]$ .
3. **Tant que** l'inertie intraclasse n'est pas stable **faire**
  - 3.1. Affecter chaque niveau de gris à la classe dont le centre est le plus proche.
  - 3.2. Calculer les centres de gravité des classes de la nouvelle classification  $C'$  des classes.
  - 3.3.  $C \leftarrow C'$
- Fin tant que**
4. **Afficher** la classification obtenue.

Le principal inconvénient de cette méthode est que la classification finale dépend du choix de la partition initiale. Le minimum global n'est pas obligatoirement atteint, on est seulement certain d'obtenir la meilleure partition à partir de la partition de départ choisie. La convergence de

l'algorithme est due au fait que  $D_i$  décroît d'une itération à l'autre. De nombreuses variantes peuvent être rencontrées. Par exemple, au lieu de calculer le centre des classes, après avoir affecté tous les pixels, le centre de gravité peut être recalculé immédiatement après chaque affectation. La méthode des centres mobiles a été généralisée sous l'appellation de la "méthode des nuées dynamiques" (Monmarché, 2000). Au lieu de définir une classe par un seul point, son centre de gravité, elle est définie par un groupe de points (noyau de la classe).

Afin de déterminer le nombre de classes de façon automatique et au cours de la procédure de classification, Ball et *al.* ont proposé un algorithme basé sur le principe de division-fusion, connu aussi sous le nom ISODATA (Monmarché, 2000; Ball, et al., 1965):

- deux classes sont fusionnées, si la distance entre leurs centres est inférieure à un seuil prédéfini;
- une classe peut être divisée en deux, si la dispersion de ses points est supérieure à un seuil prédéfini.

La figure 1.9 présente un exemple de segmentation en plusieurs classes d'une image microscopique. Les résultats obtenus dans le cas d'une segmentation en deux, trois et quatre classes sont présentés sur les figures 1.9 (b), (c) et (d), respectivement. Il est à signaler que ces résultats sont obtenus en procédant à plusieurs initialisations différentes, jusqu'à l'obtention d'un bon résultat (jugé visuellement). Le nombre de méthodes de classification étant très important, nous invitons le lecteur à consulter les ouvrages (Fukunaga, 1972; Bres, et al., 2003) pour un exposé plus complet de ces méthodes.

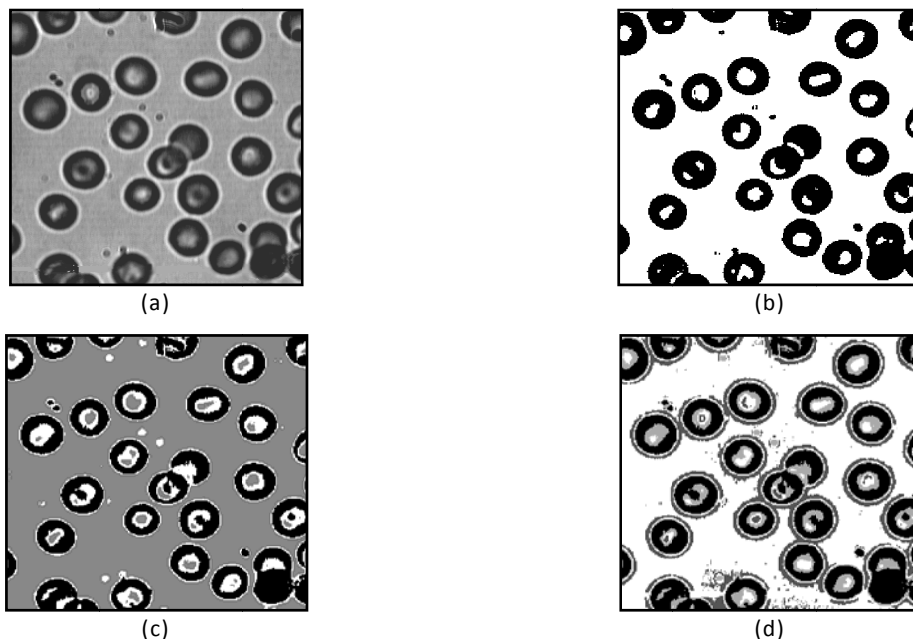


Figure 1.9. Exemple de segmentation par *K-means*. (a) Image originale, (b) Image segmentée en 2 classes, (c) Image segmentée en 3 classes, (d) Image segmentée en 4 classes.

## 1.7 LES METHODES DE SEGMENTATION "BASEES REGIONS"

Les méthodes de segmentation "basées régions" (ou par transformation de régions (Maitre, 2003)) ont pour but de décomposer l'image en un ensemble de régions connexes suivant un critère d'homogénéité. Ces méthodes sont utilisées, en particulier, dans le cas où l'histogramme de l'image est multimodal. Grâce à la notion de connectivité utilisée dans ces méthodes, la segmentation obtenue est meilleure que celle des méthodes de seuillage (Gonzalez, et al., 2002).

La formulation mathématique du problème de la segmentation d'une image en différentes régions homogènes selon un critère d'homogénéité  $H$  est présentée ci-dessous.

Soit  $X_1, \dots, X_n$  les régions de l'image  $I$ , celles-ci doivent respecter les hypothèses suivantes :

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

Le choix du critère d'homogénéité dans ces méthodes est primordial pour leur bon fonctionnement. Ce choix dépend du problème traité et du type d'images, d'où la large palette de critères d'homogénéité qui ont été utilisés (Fukunaga, 1972). A titre d'exemple, d'un point de vue statistique, la variance  $Var$  des niveaux de gris dans une région  $X$  peut être utilisée comme critère d'homogénéité ( $H$ ) :

$$Var = \frac{1}{N} \sum_{(x,y) \in X} \left[ I(x,y) - \frac{1}{N} \sum_{(x,y) \in X} [I(x,y)] \right]^2 \quad (1.40)$$

où  $N$  est le nombre de pixels de la région  $X$ .

La similarité statistique a aussi été utilisée pour tester l'union entre deux régions. Le calcul de cette similarité est effectué en assimilant chaque région à une population statistique dont la distribution est représentée par son histogramme. Ensuite, une distance statistique entre les deux distributions est calculée à l'aide de la métrique de Minkowski (Bres, et al., 2003) :

$$d(H_i, H_j) = \left[ \sum_{k=1}^n |H_i(k) - H_j(k)|^p \right]^{1/p} \quad (1.41)$$

où  $H_i$  et  $H_j$  sont les histogrammes des régions  $i$  et  $j$  respectivement et  $n$  est le nombre de classes. Dans le cas où  $p=2$ , cette distance est la distance Euclidienne, et dans le cas où  $p=1$ , c'est la distance de Manhattan (Bres, et al., 2003).

Une autre possibilité consiste à modéliser localement chaque région à l'aide d'une fonction paramétrique (par exemple par des fonctions polynomiales). Dans ce cas, la mesure de similarité est ramenée au calcul de la distance entre les paramètres des modèles (Canny, 1986) ou au calcul de l'erreur de modélisation. Il est aussi possible de combiner plusieurs critères de mesures de similarité, afin d'effectuer une segmentation hiérarchique où, à chaque niveau de segmentation, un critère de segmentation est alloué.

Les méthodes de segmentation, basées sur l'approche région, peuvent être divisées en trois classes : la première, dite locale, s'applique à une partition de l'image ; la deuxième, dite globale, s'applique à l'ensemble de l'image ; enfin, la troisième, dite hybride, s'applique localement et globalement à toute l'image.

### 1.7.1 APPROCHE PAR AGREGATION DE PIXELS

Cette méthode est appliquée aux pixels de l'image. Les pixels qui vérifient un critère prédéfini sont regroupés dans la même région. L'agrégation de pixels est initialisée avec un ensemble de pixels de départ, appelés germes, judicieusement choisis (dans les zones où la variance est faible, par

exemple). Chaque germe fusionne avec un premier pixel, pour donner une petite région. L'algorithme est itératif et chaque région croît pixel par pixel. Des heuristiques simples peuvent être utilisées : un pixel est agrégé à une région lorsque la différence entre son niveau de gris et la moyenne des niveaux de gris de la région est faible. Le choix de l'heuristique appliquée pour agréger un pixel à une région dépend de la nature de l'objet à segmenter dans l'image. Enfin, un critère d'arrêt de l'algorithme doit être déterminé. En général, l'algorithme s'arrête lorsqu'il ne reste plus de pixels satisfaisant l'heuristique choisie (Maitre, 2003; Lingrand, 2004).

### 1.7.2 APPROCHE DIVISION-FUSION

Horowitz et Pavlidis ont proposé cette méthode en 1976 (Horowitz, et al., 1976). Le processus de segmentation se déroule en deux étapes :

- **Division** : Chaque région  $X_i$  est analysée, puis l'heuristique d'homogénéité est évaluée. Si la région  $X_i$  ne vérifie pas l'heuristique, alors elle est divisée en plusieurs sous-régions. Cette opération est répétée sur chaque sous-région. Au départ, l'algorithme considère l'image entière comme étant une seule région.
- **Fusion** : dans cette étape, l'heuristique de l'homogénéité est évaluée pour tous les couples de régions voisines. Si la fusion de deux régions vérifie cette heuristique, elles sont fusionnées.

A titre d'exemple, la méthode de l'*arbre quaternaire* est souvent utilisée pour réaliser l'étape de division. Il s'agit d'une décomposition récursive en quatre sous-régions carrées, des régions ne vérifiant pas l'heuristique, en commençant par l'image entière, supposée être de taille  $N \times N$ , avec  $N = 2^n$ . Une structure pyramidale à l'envers est ainsi construite, où chaque nœud non terminal possède quatre liaisons vers le bas et une liaison vers le haut. Lors de l'étape de fusion, les blocs adjacents représentant des zones homogènes selon l'heuristique, mais qui ont été séparés par le découpage régulier quaternaire, sont regroupés (Gonzalez, et al., 2002; Maitre, 2003; Cocquerez, et al., 1995). La figure 1.10 illustre cette procédure de décomposition (division) de l'image en blocs (sous-régions).

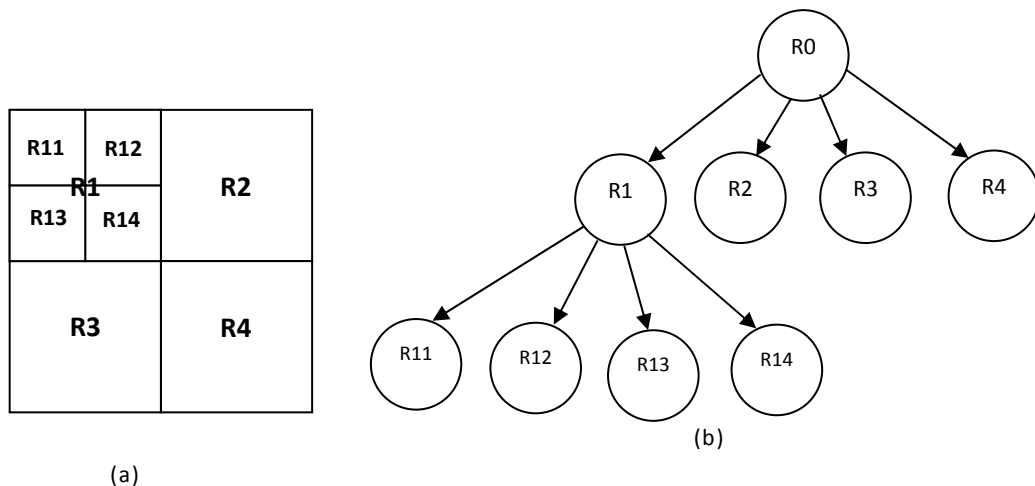


Figure 1.10 : Principe de la division de région. (a) Décomposition de l'image en blocs ; (b) Diagramme hiérarchique de la décomposition.

Le fait que cette méthode nécessite un temps de calcul important (elle parcourt l'ensemble de tous les couples de régions voisines au stade de la fusion) et que l'image segmentée présente des discontinuités, constitue un inconvénient majeur de son utilisation. Un exemple illustrant cette méthode est présenté sur la figure 1.11. L'image originale (figure 1.11 (a)) est segmentée, en utilisant

un seuil de tolérance de l'heuristique (critère) d'homogénéité de 0,2 (choisi arbitrairement). L'heuristique utilisée dans cet exemple veut que la différence, entre les valeurs maximales et minimales des pixels d'une région, soit inférieure à un seuil.

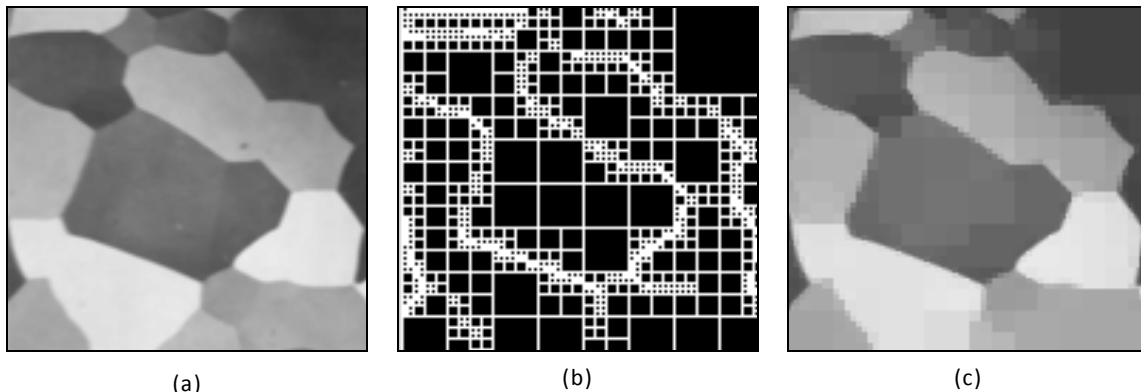


Figure 1.11 : Illustration de la méthode division-fusion. (a) Image originale, (b) Image de blocs déterminés par division, (c) Image segmentée.

### 1.7.3 LES APPROCHES PYRAMIDALES

Les pyramides irrégulières sont des approches visant à partitionner une image en régions par des techniques de type "*bottom-up region growing*". Il s'agit de représenter l'image par un graphe, dont les nœuds sont les régions et les arêtes sont les liens de voisinage (au départ, chaque pixel dans l'image est considéré comme une région). L'algorithme tente de réduire le graphe par un processus itératif, en supprimant le maximum de régions par agglomération de pixels. Suivant les règles de fusion, plusieurs techniques sont développées, telles que la pyramide stochastique (Meer, 1989), la pyramide adaptative (Jolion, et al., 1992) ou la pyramide duale (Kropatsch, 1995).

### 1.7.4 APPROCHES BASEES SUR LA MORPHOLOGIE MATHÉMATIQUE

A l'origine, la morphologie mathématique a été développée à l'Ecole des Mines de Paris. Elle repose essentiellement sur des travaux parus dans les années 1960-1970. Elle est basée sur les théories des ensembles, des treillis, de la topologie des fermés et des probabilités.

L'algorithme de segmentation le plus connu, basé sur cette approche, est l'algorithme de partage des eaux (LPE), proposé par Digabel et Lantuéjoul (Gonzalez, et al., 2002; Najman, et al., 2003). Cet algorithme permet de décomposer l'image en régions homogènes. Il repose sur une analogie avec la géographie, où une image est perçue comme un relief, en associant le niveau de gris de chaque point à une altitude. Ce qui permet de définir la ligne de partage des eaux comme étant la crête formant la limite entre deux bassins versants.

Une des difficultés de la mise en œuvre de cette analogie intuitive est qu'elle laisse beaucoup de liberté quant à sa formalisation. L'algorithme LPE peut être implémenté de deux manières : l'une est basée sur un algorithme récursif d'immersion et l'autre est basée sur l'utilisation de fonctions de distances géodésiques (Cocquerez, et al., 1995).

Il est à signaler que la méthode LPE ne s'applique pas à l'image originale, mais à son gradient (morphologique). Des lignes de partage des eaux aux points de discontinuités du niveau de gris sont alors créées, ce qui est intéressant pour les applications de segmentation. Nous avons donné seulement un petit aperçu de cette méthode : pour plus de détails voir (Cocquerez, et al., 1995; Najman, et al., 2003).

---

## 1.8 LES METHODES DE SEGMENTATION "BASEES CONTOURS"

---

La détection des contours dans une image est un problème très étudié dans le domaine du traitement et de l'analyse de l'image. De façon générale, un contour est défini comme le lieu de variation significative de l'information "niveau de gris". Par conséquent, trouver les contours dans une image revient à évaluer la variation du niveau de gris en chaque pixel de l'image (figure 1.15).



Figure 1.15 : Illustration de la détection de contours. (a) Image originale ; (b) Détection des contours par une méthode dérivative.

Les méthodes de segmentation basées sur l'approche contour ont donc pour objectif de trouver les lieux de fortes variations du niveau de gris. Un nombre important de méthodes a été développé. Ces méthodes s'appuient sur la détection des discontinuités dans l'image et peuvent être divisées en trois classes : les méthodes dérivatives (Marr, et al., 1980), les méthodes analytiques (Canny, 1986; Demigny, et al., 1997; Deriche, 1990; Shen, et al., 1986) et les méthodes basées sur les contours actifs (Terzopoulos, 1987; Terzopoulos, et al., 1988; Cohen, et al., 1993; Cohen, 1997; Montagnat, 1999; Wan, 1996; Le Guyader, 2004; Deléchelle, et al., 2000; Sum, et al., 2007; Xu, et al., 2000). Nous ne détaillerons pas cette approche de la segmentation ; toutefois, nous invitons le lecteur désirant avoir plus de détails à consulter les références suscitées.

---

## 1.9 CONCLUSION

---

Dans ce chapitre, nous avons présenté un panorama succinct des méthodes de segmentation. Par ailleurs, dans les approches par seuillage et par classification, nous avons montré que la segmentation peut être considérée comme un problème d'optimisation. L'accent a été mis sur les méthodes auxquelles nos algorithmes vont être comparés.

---

# CHAPITRE 2

## METAHEURISTIQUES D'OPTIMISATION ET SEGMENTATION : ETAT DE L'ART

---

### 2.1 INTRODUCTION

---

Dans le chapitre précédent, nous avons posé la problématique de la segmentation d'images : l'étendue de l'espace de recherche des solutions, la pertinence des paramètres caractéristiques, la complexité et la diversité des critères de segmentation (fonctions objectifs). Dans ce chapitre, nous reformulons le problème de la segmentation des images en l'examinant du point de vue des techniques d'optimisation de paramètres, et nous présentons quelques métaheuristiques d'optimisation pour le résoudre.

### 2.2 FORMULATION DU PROBLEME DE LA SEGMENTATION

---

Dans cette section, nous montrons que la segmentation d'une image par l'approche "basée région", ou par seuillage, peut se ramener à un problème d'optimisation, le plus souvent NP-difficile (Heraud, et al., 1995). D'où la nécessité d'utiliser une métaheuristique.

La segmentation d'une image  $I$  utilisant un attribut d'homogénéité  $A$ , est fréquemment définie comme une partition  $P = R_1, \dots, R_n$  de  $I$ , telle que :

1.  $I = \bigcup R_i, i \in [1, n]$
2.  $R_i$  est connexe,  $\forall i \in [1, n]$
3.  $A(R_i) = \text{vrai}, \forall i \in [1, n]$
4.  $A(R_i \cup R_j) = \text{faux}, \forall i \in [1, n],$  pour tout couple  $(R_i, R_j)$  de régions connexes.

Nous faisons observer que l'unicité de la segmentation n'est pas garantie par ces quatre conditions. Les résultats de segmentation dépendent non seulement de l'information contenue dans l'image, mais aussi de la méthode utilisée pour traiter ces informations. Généralement, pour réduire le problème de la non unicité de la solution, le problème de la segmentation est régularisé par une contrainte d'optimisation d'une fonction  $F$ , caractérisant la qualité d'une bonne segmentation. Donc, une cinquième condition est ajoutée aux quatre premières :

5.  $F(P^*) = \text{Min}_{P \in P_A(I)} F(P)$

où  $F$  est une fonction décroissante et  $P_A(I)$  est l'ensemble des partitions possibles de  $I$ .

Il est clair que la condition 5 ne résout pas entièrement le problème d'unicité de la segmentation. Il demeure des cas où plusieurs segmentations peuvent avoir la même valeur optimale. Ce qui explique la nécessité d'appliquer des algorithmes à base de métaheuristiques.

La détermination d'un vecteur de seuils optimaux (une configuration) dans l'espace des niveaux de gris rend la segmentation des images assimilable à un problème d'optimisation. D'où notre approche de la segmentation au travers des techniques destinées à résoudre ce type de problèmes. Un aperçu des différents critères de segmentation d'images a été présenté dans le chapitre précédent. Comme nous pouvons le constater, cet ensemble de critères n'est pas exhaustif, à l'instar des nombreux articles récents en traitement d'images.

La conclusion à laquelle nous sommes parvenus, est qu'il *n'existe pas de critère unique et suffisant pour segmenter de manière optimale toutes les images*. Ceci nous a conduits à des schémas de systèmes (algorithmes) de segmentation qui regroupent plusieurs critères. Pour résoudre un problème segmentation d'images, il faut donc optimiser plusieurs critères simultanément. C'est dans cet objectif que nous faisons appel à l'optimisation multiobjectif (multicritère).

Cette reformulation du problème de la segmentation d'images en un problème d'optimisation multiobjectif, nous conduit à la section suivante, où nous allons présenter les différentes approches dans ce domaine, dont certaines sont très peu utilisées en traitement d'images.

## 2.3 L'OPTIMISATION MULTIOBJECTIF

---

L'optimisation multiobjectif est née du besoin en industrie de satisfaire plusieurs critères contradictoires, simultanément. Les bases de cette optimisation ont été posées par Pareto et Edgeworth au 19<sup>ème</sup> siècle (Pareto, 1896). Ses théories trouvent leurs premières applications en économie et ces dernières années dans les sciences pour l'ingénieur (Talbi, 1999; Fonseca, et al., 1995; Gandibleux, et al., 2004; Hao, et al., 1999; Reeves, 1995; Reyes-Sierra, et al., 2006; Collette, 2002a).

Les approches de résolution des problèmes multiobjectif peuvent être réparties en trois classes : approches basées sur la transformation du problème en un problème mono-objectif (simple objectif), approches non-Pareto et approches Pareto. Ces dernières sont décrites dans les paragraphes suivants.

### 2.3.1 TRANSFORMATION EN UN PROBLEME MONO-OBJECTIF

---

Cette approche, qualifiée d'approche naïve de l'optimisation multiobjectif (MO) (Collette, et al., 2002b), consiste tout simplement à transformer un problème multiobjectif en un problème mono-objectif, dont il existe de nombreuses méthodes de résolution (Collette, et al., 2002b; Agrafiotis, 2001). Parmi les méthodes qui utilisent cette approche, nous pouvons citer les méthodes d'agrégation (Sugeno, 1974; Srinivas, et al., 1995), les méthodes  $\varepsilon$ -contraintes (Coello, et al., 1995), les méthodes de programmation par but et min-max (Coello, et al., 1995).

#### 2.3.1.1 MÉTHODE D'AGRÉGATION

---

Cette méthode est l'une des premières pour la génération de solutions à un problème multiobjectif (solutions Pareto optimales). Elle consiste à combiner, de façon linéaire, plusieurs critères  $f_i$  du problème en un critère unique  $F$ :



$$F(\bar{x}) = \sum_{i=1}^N a_i f_i(\bar{x}) \quad (2.1)$$

où les paramètres  $a_i \geq 0$  représentent les poids affectés aux critères,  $\bar{x}$  est le vecteur de paramètres de la fonction objectif,  $\sum_{i=1}^N a_i = 1$  et  $N$  désigne le nombre de critères. En faisant varier les paramètres, différentes solutions supportées sont produites. La même solution peut être aussi produite en utilisant des paramètres différents.

Les avantages de cette méthode sont sa simplicité de mise en œuvre et son efficacité, en ce sens qu'elle produit une seule solution, et qu'elle ne nécessite pas l'intervention d'un opérateur extérieur (un décideur). Toutefois, pour atteindre cet objectif, deux problèmes essentiels doivent être résolus: la détermination des valeurs des paramètres associés à chaque critère, et l'interaction entre les différents critères.

En ce qui concerne le premier problème, une approche très utilisée consiste à résoudre le problème avec plusieurs valeurs des paramètres  $a_i$ . Des stratégies "aveugles" ont été proposées dans la littérature pour engendrer, de façon aléatoire, les valeurs des paramètres (Ishibuchi, et al., 1998).

La figure 2.1 illustre le fonctionnement de la méthode d'agrégation linéaire dans un cas bi-objectif. Le calcul des paramètres de pondération revient à trouver un hyper-plan dans l'espace objectif (une ligne dans le cas bi-objectif) avec une orientation fixée. La solution Pareto optimale correspond au point où l'hyper-plan possède une tangente commune avec l'espace des solutions réalisables (point  $x$  dans la figure 2.1 (a)). La figure 2.1 (b) illustre les limites de cette méthode. Par exemple, dans le cas où le problème possède une frontière de Pareto non convexe, les solutions  $y$  et  $z$  peuvent être trouvées. Alors que les autres solutions entre les points  $y$  et  $z$  ne le seront jamais (Berro, 2001; Ishibuchi, et al., 1998).

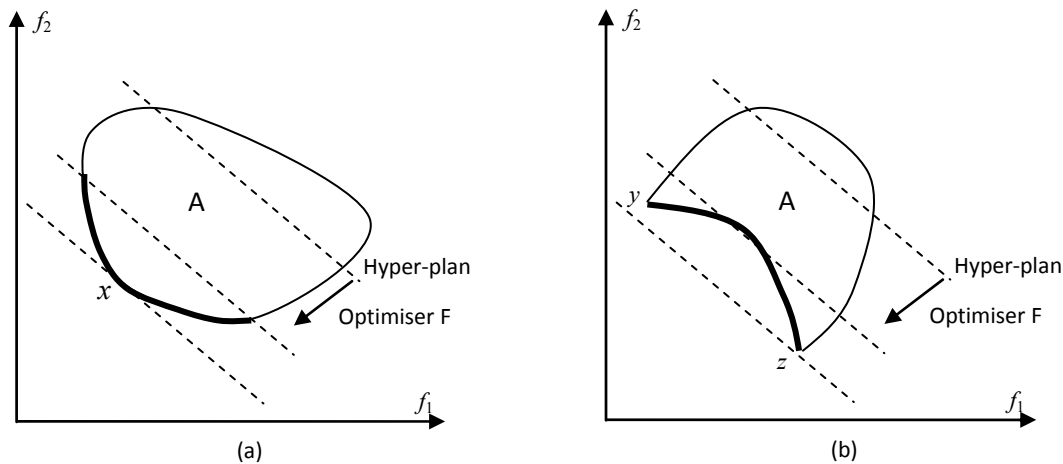


Figure 2.1 : Illustration de la méthode d'agrégation, dans un cas bi-objectif. (a) Une frontière de Pareto convexe, (b) Une frontière de Pareto non convexe. A : domaine réalisable,  $x$  : solution Pareto.

### 2.3.1.2 AUTRES METHODES DE TRANSFORMATION

La méthode, appelée  $\varepsilon$ -contraintes, de transformation d'un problème multiobjectif en problème mono-objectif est basée sur l'optimisation d'une fonction objectif  $f_j$  sujette à des contraintes sur les autres fonctions ( $j \neq i$ ). Par exemple, dans le cas d'une minimisation, le problème multiobjectif peut s'écrire sous la forme :

$$\begin{aligned} & \text{Minimiser } f_k(x) && (2.2) \\ \text{s.c. } & x \in A \text{ et } f_j(x) \leq \varepsilon_j \quad j=1, \dots, n \quad j \neq k \end{aligned}$$

Ce qui permet de transformer un problème multiobjectif en simple objectif sous contraintes et de le résoudre avec les méthodes d'optimisation mono-objectif. La connaissance *a priori* des intervalles appropriés pour les valeurs  $\varepsilon_j$  est nécessaire pour tous les objectifs (Talbi, 1999).

Une autre méthode très utilisée dans cette approche de résolution des problèmes multiobjectifs est la programmation par but et min-max. Dans cette méthode, les buts (valeurs) à atteindre pour chaque objectif doivent être définis *a priori* (Coello, et al., 1995). Ces valeurs servent par la suite à transformer le problème multiobjectif en un problème mono-objectif. A titre d'exemple, le critère à optimiser peut intégrer une norme pondérée, qui minimise l'écart par rapport aux buts. Le problème peut être formulé ainsi :

$$\begin{aligned} & \text{Minimiser } \left( \sum_{j=1}^n \lambda_j |f_j(x) - b_j|^p \right)^{1/p} && (2.3) \\ \text{s.c. } & x \in A \end{aligned}$$

où  $1 \leq p \leq \infty$ ,  $\lambda_i$  est le vecteur de pondérations, et  $b_j$  est le vecteur de buts. La norme utilisée est la métrique de Tchebycheff ( $L_p$ -métrique). Dans le cas d'une norme euclidienne  $p=2$ . La fonction (2.3) correspond à une fonction min-max, dans le cas où  $p=\infty$ . La sélection du vecteur de buts ne doit pas être aléatoire, car un mauvais vecteur peut aboutir à une solution qui n'est pas optimale au sens de Pareto (Collette, et al., 2002b).

### 2.3.2 APPROCHE NON-PARETO

---

Cette approche est appelée aussi approche non agrégée. Généralement, les méthodes basées sur cette approche traitent les objectifs séparément. Deux groupes de méthodes existent dans la littérature : sélection lexicographique et sélection parallèle.

Dans l'approche classique de la sélection lexicographique, la sélection est réalisée suivant un ordre défini *a priori*. Cet ordre permet de définir les poids des objectifs. Plusieurs métaheuristiques ont été utilisées pour la résolution des problèmes multiobjectifs avec sélection lexicographique (Talbi, 1999).

Dans l'approche par sélection parallèle, le premier travail a été publié par Schaffer (Schaffer, 1985), il est basé sur un algorithme génétique (voir section 3.3). Son algorithme, appelé VEGA ("*vector evaluated genetic algorithm*"), sélectionne les solutions courantes du front Pareto suivant chaque objectif, indépendamment des autres (sélection parallèle). L'analyse de cet algorithme a montré que son comportement est le même qu'un algorithme réalisant une agrégation linéaire. Plusieurs auteurs ont travaillé sur l'amélioration de cet algorithme (Richardson, et al., 1989). Pour plus de détails, nous invitons le lecteur à consulter (Collette, 2002a; Berro, 2001).

### 2.3.3 APPROCHE PARETO

---

Contrairement aux autres approches qui combinent les critères ou les traitent séparément, les approches Pareto utilisent la notion de dominance, pour sélectionner des solutions faisant converger la population vers un ensemble de solutions efficaces.

Cette idée, basée sur les algorithmes génétiques, a été proposée par Goldberg (Goldberg, 1989). Cette approche respecte l'intégrité de chaque critère, car les valeurs des différents critères ne sont

pas comparées *a priori*. Ces méthodes permettent d'avoir un ensemble de solutions possibles (solutions Pareto) au problème, mais ne permettent pas de choisir une alternative plutôt qu'une autre. Le choix de la solution finale revient à l'utilisateur (décideur) (Talbi, 1999). Dans les paragraphes suivants, nous définissons la notion de dominance au sens de Pareto, la frontière Pareto et la notion de "dominance contrainte".

### 2.3.3.1 OPTIMUM AU SENS DE PARETO

Vilfredo Pareto est un mathématicien italien du XIX<sup>ème</sup> siècle (Pareto, 1896). Il a posé les bases de la solution d'un problème économique multiobjectif : "Dans un problème multiobjectif, il existe un équilibre tel que l'on ne peut améliorer un critère sans détériorer au moins un des autres". Cet équilibre est appelé optimum Pareto. Donc, une solution  $x$  est dite **Pareto optimale** si elle n'est dominée par aucune autre solution appartenant à  $A$ . Ces solutions sont appelées **solutions non dominées**, ou non inférieures.

### 2.3.3.2 LA NOTION DE DOMINANCE

Soit  $x$  une solution potentielle au problème multiobjectif,  $x \in A$  domine  $x' \in A$  si :

$$\forall i, f_i(x) \leq f_i(x') \quad (2.4)$$

avec au moins un  $i$  tel que  $f_i(x) < f_i(x')$

Une solution  $x$  est dite **faiblement non dominée**, s'il n'existe pas de solution  $x' \in A$  telle que :

$$f_i(x) < f_i(x') \quad (2.5)$$

Une solution  $x$  est dite **fortement non dominée**, s'il n'existe pas de solution  $x' \in A$  telle que :

$$f_i(x') \leq f_i(x) \quad (2.6)$$

avec au moins un  $i$  tel que  $f_i(x) < f_i(x')$

où  $i = 1, \dots, n$  et  $n$  est le nombre d'objectifs.

La figure 2.2 illustre la définition de la notion de dominance. Dans cet exemple, le problème multiobjectif consiste à minimiser  $f_1$  et  $f_2$ . Les solutions représentées par les points 1, 4 et 5 ne sont dominées par aucune autre solution.

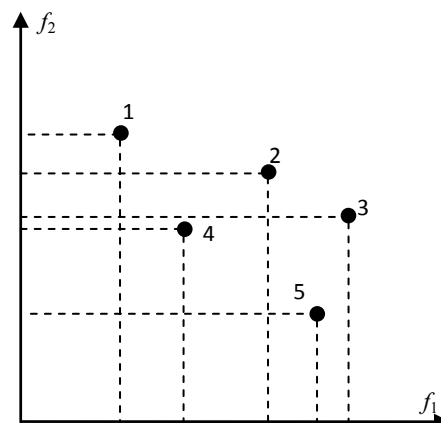


Figure 2.2 : Illustration de la notion de dominance.  $f_1$  et  $f_2$  sont deux fonctions objectifs à minimiser. La solution 4 domine les solutions 2 et 3.

### 2.3.3.3 LA FRONTIÈRE DE PARETO

La frontière, appelée aussi le front de Pareto, est l'ensemble des points Pareto optimaux. Les figures 2.3 (a) et (b) présentent deux formes, parmi une multitude d'autres, de fronts de Pareto. Sur l'exemple de la figure 2.2, le front de Pareto est composé des points 1, 4 et 5.

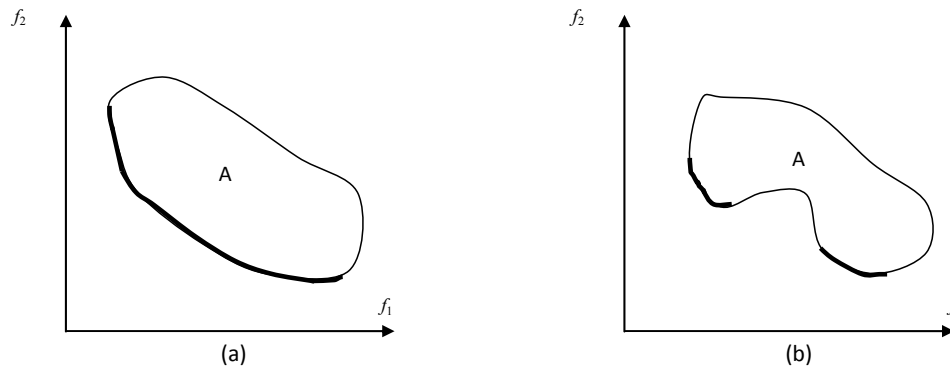


Figure 2.3 : Exemples de fronts de Pareto. A : domaine réalisable. (problème bi-objectif).

### 2.3.3.4 LA NOTION DE DOMINANCE AVEC CONTRAINTE

Une solution  $i$  domine avec contrainte une solution  $j$  lorsqu'une des situations suivantes a lieu :

- la solution  $i$  est réalisable alors que la solution  $j$  ne l'est pas ;
- les deux solutions sont réalisables et la solution  $i$  domine  $j$  ;
- les deux solutions ne sont pas réalisables, mais  $i$  a un nombre de violations de contraintes inférieur à celui de  $j$ .

Cette définition a été proposée par Deb *et al.* (Deb, et al., 2002), d'autres définitions existent dans la littérature, pour plus de détails consulter (Collette, 2002a).

## 2.4 LES METAHEURISTIQUES

Dans la littérature, les méthodes heuristiques sont réparties en deux classes : les algorithmes spécifiques à un problème donné, qui utilisent des connaissances du domaine (Talbi, 1999), et les algorithmes généraux applicables à une grande variété de problèmes multiobjectif : les *métaheuristiques* (Dréo, et al., 2003). Dans cette partie, notre intérêt va se porter sur la deuxième classe d'algorithmes. Pour résoudre des problèmes multiobjectif et déterminer des solutions Pareto optimales, plusieurs adaptations des métaheuristiques sont proposées dans la littérature. Les plus connues de ces adaptations peuvent être trouvées dans (Collette, et al., 2002b).

### 2.4.1 LE RECUIT SIMULE

L'origine de la méthode du recuit simulé vient de l'analogie avec la métallurgie, où la méthode, pour atteindre des états de basse énergie d'un solide, consiste à monter la température du solide à des valeurs élevées, puis à le laisser refroidir lentement. Ce processus est appelé "recuit".

L'algorithme d'optimisation appelé "recuit simulé" a été établi indépendamment par Kirkpatrick *et al.* en 1983 (Kirkpatrick, et al., 1983), et Cerny en 1985 (Cerny, 1985). L'idée de base est la suivante : à des paliers de températures décroissantes, l'algorithme utilise la procédure itérative de Metropolis (Metropolis, et al., 1953), pour atteindre un état de quasi-équilibre thermodynamique. Cette procédure permet de sortir des minima locaux avec une probabilité d'autant plus grande que la

température est élevée. Quand l'algorithme atteint les très basses températures, les états les plus probables constituent en principe d'excellentes solutions au problème d'optimisation.

### 2.4.1.1 ALGORITHME DE METROPOLIS

En 1953, Metropolis avait proposé un algorithme itératif qui permet d'atteindre l'état d'équilibre thermodynamique d'un système simulé à une température  $T$  (Metropolis, et al., 1953). Son principe consiste à itérer les deux étapes suivantes :

- évaluer la variation d'énergie associée à une transition élémentaire aléatoire de l'état courant  $i$ , d'énergie  $E_i$ , vers un nouvel état  $j$ , d'énergie  $E_j$  :  $\Delta E_{ij} = E_j - E_i$  ;
- accepter la transition vers le nouvel état avec une probabilité  $P_{ij}$

où

$$\begin{cases} P_{ij}(T) = 1 & \text{si } \Delta E_{ij} \leq 0 \\ P_{ij}(T) = \exp\left(-\frac{\Delta E_{ij}}{T}\right) & \text{si } \Delta E_{ij} > 0 \end{cases} \quad (2.7)$$

### 2.4.1.2 ALGORITHME DU RECUIT SIMULE

Le principe du recuit simulé, présenté dans l'Algorithme 2.1, consiste à démarrer l'algorithme avec une température initiale élevée et à contrôler la décroissance de la température (2.7) au sein de l'algorithme de Metropolis. Plusieurs lois de décroissance de la température peuvent être utilisées en pratique (Dréo, et al., 2003; Eglese, 1990).

Algorithme 2.1 : Algorithme du recuit simulé.

<ol style="list-style-type: none"> <li>1 <b>Définir</b> la fonction objectif (<math>f</math>).</li> <li>2 <b>Choix</b> des mécanismes de perturbation d'une configuration <math>\Delta S</math>.</li> <li>3 <b>Tirer</b> une configuration aléatoire <math>S</math>.</li> <li>4 <b>Calculer</b> l'énergie associée à cette configuration <math>E</math>.</li> <li>5 <b>Initialiser</b> la température (<math>T_0</math>).</li> <li>6 <b>Tant que</b> Conditions d'arrêts pas satisfaites <b>faire</b> <ol style="list-style-type: none"> <li>6.1. <b>Tant que</b> l'équilibre thermodynamique pas atteint <b>faire</b> <ol style="list-style-type: none"> <li>6.1.1. Tirer une nouvelle configuration <math>S'</math>.</li> <li>6.1.2. Appliquer la règle de Metropolis.</li> <li>6.1.3. <b>Si</b> <math>f(S') &lt; f(S)</math> <math display="block">f_{\min} = f(S')</math> <math display="block">S_{opt} = S'</math> </li> </ol> </li> <li><b>Fin de si</b></li> <li><b>Fin tant que</b></li> <li>6.2. <b>Décroître</b> la température.</li> </ol> </li> <li><b>Fin tant que</b></li> <li>7 <b>Afficher</b> la solution optimale.</li> </ol>
---

Nous distinguons deux types de recuit simulé, suivant la méthode de décroissance de la température :

- le recuit **homogène** : la température n'est abaissée que lorsque l'équilibre thermodynamique est atteint. Cet algorithme suppose que la procédure de Metropolis est itérée un nombre infini de fois et n'a donc qu'un intérêt purement théorique;

- le recuit **inhomogène** : le paramètre de température est diminué après un nombre fini d'évaluations de perturbations à une température donnée.

En résumé, les principaux paramètres de contrôle sont :

- la valeur initiale de la température ;
- la fonction de décroissance de la température ;
- le critère de changement de palier de température ;
- les critères d'arrêt.

L'efficacité du recuit simulé dépend fortement du choix de ses paramètres de contrôle, dont le réglage reste très empirique.

Pour le calcul de la température de départ, plusieurs méthodes ont été proposées (Dréo, et al., 2003). Une des méthodes est basée sur l'observation de la variation moyenne de la fonction  $f$ . A partir d'une configuration initiale  $S_0$ , un certain nombre de solutions  $S'_0$  (environ 50 à 100) telles que  $f(S'_0) > f(S_0)$  est tiré, ainsi la variation moyenne  $\langle \Delta f \rangle$  est calculée. La température initiale  $T_0$  est calculée de façon à accepter au départ un certain pourcentage ( $p$ ) de mouvements dégradant la fonction  $f$ . Pour une variation moyenne  $\langle \Delta f \rangle$  de la fonction  $f$  autour de la configuration initiale  $S_0$ , la valeur de  $p$  est de 50%. La valeur de  $T_0$  est déduite de la relation suivante :

$$p = \exp \frac{-\langle \Delta f \rangle}{T_0} \quad (2. 8)$$

Le réglage de la décroissance de la température est très important dans l'algorithme du recuit simulé. Une forte décroissance de température risque de piéger l'algorithme dans un minimum local, alors qu'une faible décroissance au début du processus entraîne une convergence lente de l'algorithme. D'un point de vue théorique, la convergence théorique du recuit simulé inhomogène est assurée avec une loi logarithmique (Hajek, 1988) :

$$T_k = \frac{\mu}{\log(1+k)} \quad (2. 9)$$

où  $k$  est le nombre de paliers de température effectués,  $\mu$  une constante positive égale à la profondeur maximale des minima locaux.

Malheureusement, cette règle induit un temps de calcul prohibitif. En pratique, la décroissance géométrique est souvent utilisée :

$$T_{k+1} = \alpha.T_k \quad (2. 10)$$

où  $\alpha$  est une constante dans l'intervalle  $[0, 1]$ .

En ce qui concerne le changement de température, le nombre de transformations au bout duquel la température est abaissée peut simplement être spécifié.

Le recuit a connu un grand succès dans différents domaines d'application (Chen, et al., 1999), grâce à ses deux atouts principaux : un comportement type *boîte noire* et une facilité de "réglage" des paramètres internes. D'un point de vue théorique, le recuit simulé permet d'approcher de près la solution optimale du problème plus rapidement qu'une exploration exhaustive dans l'espace de recherche. En pratique, un bon réglage des paramètres internes de l'algorithme permet d'accélérer la convergence vers une solution pseudo-optimale, avec une précision prédéfinie. Le recuit simulé est aussi adapté pour résoudre les problèmes d'optimisation continue (Siarry, et al., 1997).

L'inconvénient majeur de cet algorithme est sa lenteur. Toutefois, plusieurs tentatives de parallélisation de l'algorithme ont été proposées dans la littérature, au détriment de sa convergence théorique. Il est à signaler qu'il existe des approches rapides qui sont simples à mettre en œuvre et qui conservent les propriétés de convergence, comme par exemple dans (Roussel-Ragot, 1990), où le recuit simulé est réparti sur plusieurs calculateurs en parallèle.

En optimisation multiobjectif le recuit simulé (RS) a été appliqué après transformation du problème multiobjectif en un problème mono-objectif (Friesz, et al., 1993; Serafini, 1992). Les deux méthodes les plus populaires sont : la méthode MOSA ("*Multiple Objective Simulated Annealing*") proposée dans (Ulungu, et al., 1999), où le RS a été utilisé pour rechercher la surface de compromis, et la méthode PASA ("*Pareto Archived Simulated Annealing*") proposée dans (Engrand, 1997) qui utilise une fonction d'agrégation des fonctions objectifs couplées, avec un système d'archivage de solutions non dominées.

## 2.4.2 LE RECUIT MICROCANONIQUE

---

Cette méthode d'optimisation exploite des principes très proches de ceux du recuit simulé. Le recuit microcanonique effectue des paliers d'énergie totale décroissante en diminuant l'énergie cinétique entre deux paliers. Ainsi l'algorithme converge par réduction des énergies d'un ensemble de solutions autour de celles qui sont optimales (Ray, et al., 1997; Torreao, et al., 1995; Lee, 1994; Herault, et al., 1993; Woods, et al., 1998). Le recuit microcanonique met en œuvre l'algorithme de Creutz (Creutz, 1983), qui est basé sur l'évaluation d'une succession de transitions pour maximiser l'entropie pour une énergie totale constante. Cette énergie est fixée préalablement.

### 2.4.2.1 ALGORITHME DE CREUTZ

---

Pour une énergie totale  $E_t$ , un algorithme itératif permet de converger vers l'équilibre thermodynamique en répétant un grand nombre de fois les deux étapes suivantes :

- évaluation de la variation d'énergie associée à une transition élémentaire aléatoire de l'état courant  $i$ , d'énergie potentielle  $E_i$ , vers un nouvel état  $j$ , d'énergie  $E_j$  :  $\Delta E_{ij} = E_j - E_i$  ;
- acceptation de cette transition vers ce nouvel état si  $\Delta E_{ij} \leq E_t - E_i$ .

Dans l'article original de Creutz (Creutz, 1983), le terme  $E_t - E_i = D$  est appelé "*Demon*" et peut être interprété comme l'énergie cinétique du système à l'état  $i$ . Les transitions vers des états d'énergie potentielle plus élevée ne sont autorisées que s'il y a suffisamment d'énergie cinétique pour compenser l'augmentation d'énergie potentielle, et donc rester à énergie constante.

### 2.4.2.2 ALGORITHME DU RECUIT MICROCANONIQUE

---

L'algorithme du recuit microcanonique consiste à réduire l'énergie totale, à partir d'une énergie totale élevée, au sein de l'algorithme de Creutz. Plusieurs lois de décroissance de l'énergie peuvent être utilisées, comme dans le cas du recuit simulé. Les différentes étapes de l'algorithme sont résumées dans Algorithme 2.2.

L'algorithme de Creutz est beaucoup plus simple que celui de Metropolis, et nécessite bien moins de calculs. Comme il n'y a pas besoin d'un générateur de nombres aléatoires, sa mise en œuvre est simplifiée par rapport à d'autres algorithmes. En comparaison avec le recuit simulé, dans le cas des problèmes de grande taille, plusieurs études ont montré que les résultats obtenus sont très proches,

avec un avantage pour le recuit microcanonique en terme de calcul (Herault, et al., 1993). Par contre, la convergence de cet algorithme n'est toujours pas prouvée théoriquement.

Dans certains cas, le recuit microcanonique peut être moins performant, à cause des barrières d'énergie infranchissables qu'il construit au cours de sa recherche et qui peuvent le piéger dans un minimum local.

Cet algorithme peut éventuellement être appliqué pour résoudre les problèmes multiobjectifs dans le cas où ces derniers sont transformés en un problème mono-objectif.

Algorithme 2.2 : Algorithme du recuit microcanonique.

- 1. Initialisation,**
  - 1.1. Définir** le pourcentage minimal  $p$  de transitions acceptées dans le premier palier.
  - 1.2. Fixer**  $E_t$  initial tel que le taux  $p$  des transformations testées soient acceptées.
  - 1.3. Tirer** aléatoirement une solution acceptable et calculer son énergie  $E$ .
  - 1.4. Choisir** le nombre maximal de transitions testées à chaque palier de l'énergie totale.
- 2. Tant que** le nombre de solutions acceptées est non nul **Faire**
  - 2.1. Répéter** pour un nombre d'itérations donné, tirer aléatoirement une transition et calculer la variation d'énergie associée  $\Delta E$ .
    - 2.1.1. Accepter la transition si  $\Delta E \leq E_t - E$ .
    - 2.1.2. Si  $\Delta E \leq 0$ , comparer le nouvel état au meilleur état trouvé depuis le début de la recherche, et le mémoriser s'il est meilleur.
  - 2.2. Décroître**  $E_t$ :  $E_t = \alpha E_t$ .
- Fin tant que**
- 3. Afficher** le meilleur état rencontré au cours de la recherche.

### 2.4.3 ALGORITHMES GENETIQUES

#### 2.4.3.1 PRINCIPE

Les algorithmes génétiques appartiennent à la classe des algorithmes évolutionnaires. Ce sont des métaheuristiques inspirées de l'analogie entre un processus d'optimisation et l'évolution des êtres vivants. La simulation des mécanismes de variation et de sélection œuvrant dans les processus évolutifs naturels est exploitée pour résoudre des problèmes artificiels d'optimisation (Holland, 1975; Goldberg, 1989).

Dans les algorithmes évolutionnaires, l'analogie avec l'optimisation consiste à considérer les solutions potentielles au problème comme des chromosomes. Ceux-ci sont manipulés par des opérateurs de *sélection* et *mutation*. Dans le cas des algorithmes génétiques, un troisième opérateur est utilisé, appelé opérateur de *croisement*. La qualité de la solution correspondant à chaque chromosome est quantifiée via sa propre *fitness*. Les procédures de croisement et de mutation ont pour but de créer de manière permanente de nouveaux chromosomes.

Les algorithmes génétiques gardent la même taille de la *population* des solutions potentielles, appelées aussi *individus*. La population initiale est, généralement, choisie de manière aléatoire. Cependant d'autres mécanismes d'initialisation peuvent être utilisés suivant l'application (Bhanu, et al., 1995). Le nombre de fois qu'un individu est sélectionné pour la recombinaison est fonction de sa *fitness* par rapport au reste de la population. La procédure Algorithme 2.3 présente les différentes étapes de base d'un algorithme génétique. Dans cet exemple,  $P(t)$  représente une population de



solutions candidates pour un problème donné, à l'itération  $t$ . Dans les paragraphes suivants, nous décrivons les différentes procédures qui composent un algorithme génétique.

#### 2.4.3.1.1 LA REPRESENTATION DES SOLUTIONS

Dans un algorithme génétique, la représentation des individus est basée sur le codage binaire de l'information.

Le problème est qu'il existe plusieurs façons de coder l'information, et le choix optimal n'est pas évident à trouver. Il doit en effet permettre une recherche efficace par les opérateurs qui seront appliqués aux chromosomes. Afin de résoudre ce problème, une adaptation de la représentation et des opérateurs à l'espace de recherche, associée au problème traité, est essentielle. Dans une telle représentation, deux génotypes<sup>1</sup> dont les représentations diffèrent de peu doivent donner des solutions proches. Cette propriété n'est pas toujours vérifiée dans le cas d'un codage binaire classique. Par exemple, le passage du nombre entier 7 au nombre entier 8, représentés respectivement, par les chromosomes 0111 et 1000, se fait en 4 mutations (ce problème est connu sous le nom de barrière de Hamming). Différentes possibilités ont été proposées, comme le codage de Gray (Caruana, et al., 1988). Dans ce codage, le passage d'un nombre à l'autre se fait par l'inversion d'un seul bit de la chaîne. Cependant dans la pratique, il est rare que ces codages soient appliqués.

Algorithme 2.3 : Principe d'un algorithme génétique.

1. **Initialisation** de  $P(t)$ .
2. **Evaluer** chaque individu de  $P(t)$ .
3. **Tant que** le critère d'arrêt n'est pas satisfait **Faire**
  - 3.1.  $t = t + 1$ .
  - 3.2. **Sélectionner**  $P(t+1)$  de  $P(t)$ .
  - 3.3. **Croisement**  $P(t+1)$ .
  - 3.4. **Muter**  $P(t+1)$ .
  - 3.5. **Evaluer**  $P(t+1)$ .
- Fin tant que**
4. **Afficher** le meilleur état rencontré au cours de la recherche.

#### 2.4.3.1.2 LES COMPOSANTES DES ALGORITHMES GENETIQUES

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

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

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

<sup>1</sup> Génotype : un ensemble de gènes d'un individu. Dans la plupart des cas on utilise des génotypes formés d'un seul chromosome.

---

### 2.4.3.1.3 LES OPERATEURS UTILISES DANS LES ALGORITHMES GENETIQUES

---

#### 2.4.3.1.3.1 OPERATEUR DE SÉLECTION

Le principe de cet opérateur consiste à favoriser la propagation des meilleures solutions parmi la population, tout en préservant la diversité génétique, afin d'explorer de nouvelles régions de l'espace de recherche.

Plusieurs opérateurs de sélection ont été proposés dans la littérature (Dréo, et al., 2003), comme, par exemple, celui proposé par Goldberg (Goldberg, 1989) qui sélectionne les individus proportionnellement à leur performance.

#### 2.4.3.1.3.2 OPÉRATEUR DE CROISEMENT

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

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

$$A = 1\ 1\ 0 : 1\ 0$$

$$B = 0\ 1\ 1 : 0\ 1$$

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

$$A' = 1\ 1\ 0 : 0\ 1$$

$$B' = 0\ 1\ 1 : 1\ 0$$

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

#### 2.4.3.1.3.3 OPÉRATEUR DE MUTATION

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

#### 2.4.3.1.3.4 L'ÉLITISME

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

Vu le nombre important de méthodes de résolution des problèmes multiobjectif avec les AGs, nous consacrons le paragraphe suivant à leur application. L'accent sera notamment mis sur les AGs dans le cas de l'approche Pareto.

### 2.4.3.2 LES ALGORITHMES GENETIQUES EN OPTIMISATION MULTIOBJECTIF

---

Les algorithmes génétiques ont été très utilisés pour résoudre les problèmes multiobjectif, avec toutes les approches de l'optimisation multiobjectif (Talbi, 1999). Une étude comparative de quelques algorithmes est dans (Zitzler, et al., 1999).

#### ***Les AGs pour l'approche de transformation vers le mono-objectif***

Dans l'approche transformant un problème multiobjectif en un problème mono-objectif, plus de détails sur les applications des AGs sont fournis dans (Talbi, 1999). Dans le cas de l'approche  $\varepsilon$ -contraintes, les AGs ont été très utilisés pour résoudre cette classe de problèmes (Collette, 2002a).

#### ***Les AGs pour l'approche non-Pareto***

Dans le cas de l'approche non-Pareto, l'AG le plus répandu est celui proposé par Schaffer (Schaffer, 1985) connu sous le nom de VEGA. La particularité de cet AG est de créer des sous-populations dont les individus sont dédiés à un objectif en particulier.

#### ***Les AGs pour l'approche Pareto***

Dans ce type d'approche, deux familles de méthodes se dégagent : les méthodes non-élitistes et les méthodes élitistes.

Parmi les méthodes non élitistes, nous pouvons citer la méthode MOGA ("*Multiple Objective Genetic Algorithm*") proposée par Fonseca *et al.* (Fonseca, et al., 1995), où chaque individu est rangé suivant le nombre d'individus qu'il domine, la méthode NSGA ("*Non dominated Sorting Genetic Algorithm*") proposée par Srinivas *et al.* (Srinivas, et al., 1995), où le calcul de la *fitness* s'effectue en séparant la population en plusieurs groupes, en fonction du degré de domination au sens de Pareto de chaque individu. Pour plus de détails sur ces méthodes, ainsi que d'autres, nous invitons le lecteur à consulter (Berro, 2001; Collette, et al., 2002b).

Les techniques élitistes ne sauvegardant pas les individus Pareto optimaux trouvés au cours des itérations. Elles se distinguent de deux manières : la difficulté de maintenir la diversité et leur lenteur de convergence vers la frontière de Pareto.

Parmi les nombreux AGs adoptant une stratégie élitiste, nous citons l'algorithme SPEA ("*Strength Pareto Evolutionary Algorithm*") proposé par Zitzler *et al.* (Zitzler, et al., 1999), où le passage d'une génération à l'autre commence par la mise à jour des sauvegardes. Tous les individus non dominés sont sauvegardés et les individus dominés, déjà présents, sont éliminés. Après avoir amélioré cet algorithme, de meilleures performances ont été enregistrées, mais au prix de l'accentuation de sa complexité (Zitzler, et al., 2002). L'algorithme le plus connu et le plus privilégié, est NSGAII, une deuxième version de NSGA (Deb, et al., 2002). Dans cette version, l'auteur tente de résoudre toutes les critiques faites sur NSGA en termes de complexité, non élitisme et utilisation du partage. Cet algorithme sera détaillé ultérieurement. Nous invitons le lecteur à consulter (Collette, et al., 2002b; Berro, 2001), pour une vue plus complète des différents algorithmes.

### 2.4.4 L'OPTIMISATION PAR ESSAIM PARTICULAIRE (OEP)

---

La métaheuristique basée sur la méthode des essaims particuliers ("*Particle Swarm Optimization*", PSO) a été développée par Kennedy *et al.* en 1995 (Kennedy, et al., 1995). Le principe de la méthode provient analogiquement avec les comportements collectifs d'animaux. L'optimisation par essaims particuliers (OEP) est classée dans les techniques d'optimisation stochastiques à population. Une brève description de l'algorithme OEP est présentée dans les paragraphes suivants.

### 2.4.4.1 PRINCIPE

La méthode OEP s'identifie aux algorithmes génétiques. Cependant, elle ne possède pas d'opérateur d'évolution. En outre, une population de solutions potentielles est utilisée dans la recherche.

L'algorithme commence avec une initialisation aléatoire de l'essaim de particules dans l'espace de recherche. Chaque particule est modélisée par sa position dans l'espace de recherche et par sa vitesse. A chaque instant, toutes les particules ajustent leurs positions et vitesses, donc leurs trajectoires, par rapport (1) à leurs meilleures positions, (2) à la particule ayant la meilleure position dans l'essaim et (3) à leur position actuelle. En réalité, chaque particule est influencée, non seulement par sa propre expérience, mais aussi par celle des autres particules. La position et la vitesse d'une particule dans un espace de recherche à  $N$  dimensions sont définies par :  $P_i = (p_{i,1}, \dots, p_{i,N})$  et  $V_i = (v_{i,1}, \dots, v_{i,N})$ , respectivement. Chaque particule est caractérisée par sa meilleure position  $L_i = (l_{i,1}, \dots, l_{i,N})$  à l'itération  $t$ . La meilleure position qu'atteint l'essaim est sauvegardée dans le vecteur  $G = (g_1, \dots, g_N)$ . La vitesse de chaque particule est mise à jour selon l'expression suivante (Clerc, et al., 2002):

$$v_{ij}(t+1) = K \left[ w \cdot v_{ij}(t) + c_1 \cdot r_1 \cdot (l_{ij} - v_{ij}(t)) + c_2 \cdot r_2 \cdot (g_j - v_{ij}(t)) \right] \quad (2.11)$$

et

$$K = \frac{2}{2 - \varphi - \sqrt{\varphi^2 - 4\varphi}} \quad (2.12)$$

où  $\varphi = c_1 + c_2$  et  $\varphi > 4$ . Dans (Clerc, et al., 2002), les auteurs suggèrent les valeurs suivantes  $j = 1, \dots, N$ ,  $w$  est une constante appelée *facteur d'inertie*,  $c_1$  et  $c_2$  sont des constantes appelées coefficients d'accélération,  $r_1$  et  $r_2$  sont des nombres aléatoires uniformément distribués dans l'intervalle [0 1]. Si la vitesse calculée fait sortir une particule de l'espace de recherche, sa *fitness* n'est pas calculée. Compte tenu de la nouvelle vitesse, obtenue à partir de (2.11) et (2.12), la position à l'itération  $t+1$  est alors calculée :

$$p_{ij}(t+1) = p_{ij}(t) + v_{ij}(t+1) \quad (2.13)$$

pour  $j=1, \dots, N$ .

Le paramètre d'inertie  $w$  contrôle l'influence de l'ancienne vitesse sur la vitesse courante, afin de permettre aux particules d'éviter les minima locaux. De la même façon,  $c_1$  contrôle le comportement de la particule dans sa recherche autour de sa meilleure position et  $c_2$  contrôle l'influence de l'essaim sur le comportement de la particule. Les différentes étapes de l'algorithme sont présentées sur la figure Algorithme 2.4. La méthode est analysée avec plus de détails dans (Clerc, et al., 2002).

Cette méthode a connu beaucoup de succès auprès de la communauté d'optimisation. Ses bonnes performances dans différentes applications, et la possibilité d'hybridation avec d'autres métaheuristiques ont contribué à cet engouement. Malgré son "jeune âge" en comparaison des autres métaheuristiques, un nombre important de travaux ont été publiés ; ce qui a conduit à la parution d'un article récent, en deux parties, (Banks, et al., 2007a; Banks, et al., 2007b), présentant un état de l'art complet sur cette méthode. La plupart des variantes de la méthode, ainsi que les différentes hybridations avec d'autres métaheuristiques, y sont, brièvement, présentées.

Algorithme 2.4 : Principe de l'algorithme OEP.

1. **Initialiser** la population de particules avec des positions et vitesses aléatoires.
2. **Evaluer** la fonction objectif pour chaque particule et calculer  $g$ .
3. Pour chaque individu  $i$ ,  $L_i$  est initialisée à  $P_i$
4. **Répéter** jusqu'au critère d'arrêt
  - 4.1. Mettre à jour les vitesses et les positions des particules.
  - 4.2. Evaluer la fonction objectif pour chaque individu.
  - 4.3. Calculer les nouveaux  $L_i$  et  $g$ .
5. **Afficher** le meilleur état rencontré au cours de la recherche.

Des travaux récents ont montré la possibilité d'utiliser l'algorithme OEP comme une boîte noire à l'image du recuit simulé (Clerc, 2003; Zhang, et al., 2003 ; Ye, et al., 2002; Yasuda, et al., 2004), ce qui a encouragé l'application de la méthode dans différents domaines (Banks, et al., 2007a).

---

#### 2.4.4.2 L'OPTIMISATION PAR ESSAIMS PARTICULAIRES EN OPTIMISATION MULTIOBJECTIF

---

Afin d'étendre l'OEP à l'optimisation multiobjectif, il a fallu modifier la formulation originale du schéma d'un problème multiobjectif. Pour cela il a fallu répondre aux trois questions suivantes :

1. Comment sélectionner les particules pour donner la préférence à celle qui est non dominée par rapport aux autres ?
2. Comment retenir les solutions Pareto optimales par rapport à celles déjà vues et rencontrées au cours de la recherche ?
3. Comment maintenir la diversité dans l'essaim pour éviter la convergence vers une seule solution ?

La première application des OEP en multiobjectif a été proposée par Moore *et al.* (Moore, et al., 1999). Par la suite, il a fallu attendre quelques années (2002) pour voir apparaître le premier article. Cependant, le nombre d'articles reste très faible comparé à celui des AGs en multiobjectif (Reyes-Sierra, et al., 2006).

##### ***L'OEP pour l'approche de transformation vers le mono-objectif***

Dans cette approche, quelques algorithmes dédiés à la résolution d'un problème multiobjectif, en le transformant en un problème mono-objectif, ont été proposés dans la littérature ; la plupart sont recensés dans (Reyes-Sierra, et al., 2006). L'une des approches, utilisant l'agrégation linéaire d'objectifs, consiste à diviser l'essaim en sous-essaims, de taille égale. Chaque sous-essaim utilise un vecteur de paramètres différents et évolue dans sa propre direction. Enfin, les solutions Pareto optimales sont choisies en utilisant la méthode du gradient.

##### ***L'OEP pour l'approche non-Pareto***

Plusieurs algorithmes ont été développés. Nombreux sont inspirés des AGs, à l'image de l'algorithme VEPSO dérivé de l'algorithme VEGA (Schaffer, 1985). VEPSO utilise une approche dite multi-essaims, où chaque essaim est évalué en utilisant une des fonctions objectifs du problème. Les différents essaims communiquent entre eux à travers l'échange de leur meilleure position (Reyes-Sierra, et al., 2006).

##### ***L'OEP pour l'approche Pareto***

C'est dans cette approche d'optimisation multiobjectif que le nombre de travaux publiés est le plus important. Ces approches utilisent les méthodes de sélection de "leader" basées sur la dominance au

sens de Pareto. Le principe de base de toutes ces approches consiste à considérer des particules comme étant des leaders, si elles sont non dominées dans l'essaim. Plusieurs schémas de sélection des leaders sont possibles (Reyes-Sierra, et al., 2006).

#### 2.4.5 LES ALGORITHMES DE COLONIES DE FOURMIS

Cette métaheuristique est inspirée des comportements collectifs de dépôt et de suivi de pistes observés dans les colonies de fourmis. Les fourmis communiquent indirectement à travers des modifications dynamiques de leur environnement (les pistes de phéromones) et construisent ainsi une solution à un problème en tenant compte de leur expérience collective.

L'illustration de la capacité des fourmis à chercher le chemin le plus court est donnée via l'exemple de l'apparition d'un obstacle sur un chemin entre la nourriture et le nid. La figure 2.4 présente une illustration de la procédure suivie par les fourmis pour aller chercher de la nourriture.

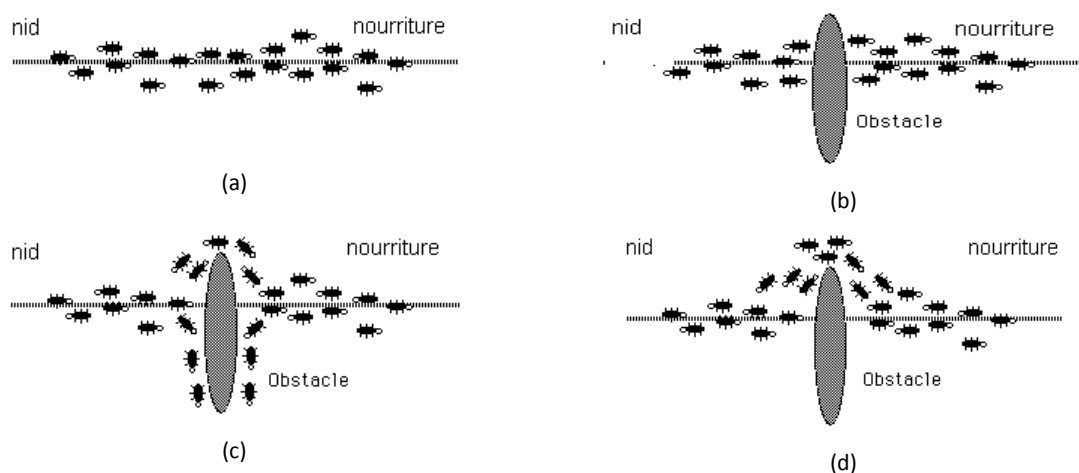


Figure 2.4 : Illustration de la capacité des fourmis à chercher de la nourriture en minimisant leur parcours. (a) Recherche sans obstacle, (b) Apparition d'un obstacle, (c) Recherche du chemin optimal, (d) Chemin optimal trouvé (Dorigo, 2006).

Au départ, les fourmis partent de leur nid en direction de la nourriture (figure 2.4 (a)). Dès l'apparition de l'obstacle, les fourmis modifient leur trajectoire pour le contourner (figure 2.4 (b) et (c)). La colonie se divise alors en deux groupes, le premier groupe choisit le plus long chemin, tandis que le second choisit le plus court. Puisque toutes les fourmis déposent leurs phéromones sur leur trajectoire, le chemin le plus court finira par avoir la densité de phéromones la plus élevée. Par conséquent, le nombre de fourmis suivant cette trajectoire augmente. Au fil du temps, la quantité de phéromones déposée sur le plus long chemin diminue et finit par disparaître ; toutes les fourmis suivent alors le chemin le plus court (figure 2.4 (d)).

Le premier algorithme d'optimisation par colonies de fourmis ("*Ant colony optimization*", ACO) a été proposé par Dorigo *et al.* vers le début des années quatre-vingt-dix (Dorigo, et al., 1996). Le premier algorithme, appelé "*Ant System*" (AS), a été développé spécialement pour résoudre le problème du voyageur de commerce (Blum, 2005; Dréo, et al., 2003). Depuis, cette approche a connu un succès important et le nombre de travaux publiés augmente d'une année à l'autre (Blum, 2005). Toute la théorie et les plus intéressants travaux récents sont dans (Blum, 2005; Dorigo, et al., 2005).

L'adaptation du comportement et des caractéristiques des fourmis réelles pour l'optimisation a nécessité quelques modifications, d'où la naissance des fourmis artificielles. Elles possèdent une mémoire, ne sont jamais aveugles et le temps est discret (Monmarché, 2000).

La modélisation des fourmis artificielles pour la résolution du problème du voyageur de commerce est la suivante : à chaque itération  $t$ , chaque fourmi  $k$  parcourt le graphe et construit un trajet complet de  $n$  étapes ( $n = \text{Card}(N)$ ). Pour chaque fourmi, le trajet entre une ville  $i$  et une ville  $j$  dépend de :

1. La liste des villes déjà visitées. Elle définit les mouvements possibles à chaque pas, quand la fourmi  $k$  est sur la ville  $i$  :  $J_i^k$
2. La **visibilité** de chaque fourmi est définie par l'inverse de la distance entre les villes :  $\eta_{ij} = \frac{1}{d_{ij}}$ . Ce qui permet aux fourmis de se déplacer vers les villes les plus proches.
3. L'**intensité de la piste** correspond à la quantité de phéromone déposée sur le chemin reliant deux villes. Ce qui définit une pseudo-mémoire globale du système.

L'heuristique du déplacement, appelée la *probabilité de transition* (Dorigo, et al., 2005), est définie par :

$$p(c_i^j | s^p) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in J_i^k} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta} & \text{si } j \in J_i^k \\ 0 & \text{si } j \notin J_i^k \end{cases} \quad (2.14)$$

où  $\alpha$  et  $\beta$  sont deux paramètres contrôlant l'importance de l'intensité de la piste  $\tau_{ij}(t)$ , et la visibilité  $\eta_{ij}$ . Une étude sur l'influence de ces deux paramètres figure dans (Dorigo, et al., 2005). A la fin d'une tournée complète, la quantité de phéromone  $\Delta \tau_{ij}^k(t)$  laissée par chaque fourmi dépend de la qualité de la solution trouvée, ce qui est modélisé par la relation suivante :

$$\Delta \tau_{ij}^k(t) = \begin{cases} \frac{Q}{L^k(t)} & \text{si } (i, j) \in T^k(t) \\ 0 & \text{si } (i, j) \notin T^k(t) \end{cases} \quad (2.15)$$

où  $T^k(t)$  est le trajet effectué par la fourmi  $k$  à l'itération  $t$ ,  $L^k(t)$  est la longueur du tour et  $Q$  un paramètre fixé. Pour que l'algorithme évite les pièges des solutions locales, une mise à jour des pistes est effectuée :

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \Delta \tau_{ij}(t) \quad (2.16)$$

où  $\Delta \tau_{ij}(t) = \sum_{k=1}^m \Delta \tau_{ij}^k(t)$  et  $m$  est le nombre de fourmis. Au départ, la phéromone est initialisée par une petite quantité  $\tau_0 \geq 0$ .

La figure 2.5 illustre le principe de base d'un algorithme d'optimisation par la méthode des colonies de fourmis.

En mono-objectif, la métaheuristique des colonies de fourmis est privilégiée pour résoudre des problèmes d'optimisation combinatoire. Cependant, en optimisation multiobjectif par l'approche Pareto, le nombre de travaux publiés est faible comparé aux autres métaheuristiques, seulement trois références sont citées dans l'article résumant l'état de l'art (Blum, 2005). Quelques travaux récents d'application de ces algorithmes ont été publiés, comme par exemple (Benlian, et al., 2007)

pour la poursuite des multi-trajets. L'algorithme proposé par Doerner (Doerner, et al., 2004), amélioré deux ans après (Doerner, et al., 2006), P-ACO, est dédié à la résolution du problème multiobjectif d'allocations de portefeuilles, où les résultats obtenus sont supérieurs à ceux obtenus avec NSGA et le recuit simulé. Toutefois, nous regrettons l'absence d'une comparaison avec NSGAI. L'algorithme OCF proposé par Gagné (Gagné, et al., 2004) repose sur le même principe que celui de P-ACO. Il consiste à optimiser séparément chaque objectif. En effet, à chaque cycle une fourmi change de fonction objectif à optimiser. A la fin de chaque cycle, la fourmi ayant la meilleure qualité de solution met à jour la trace de phéromone, en fonction de la solution trouvée.

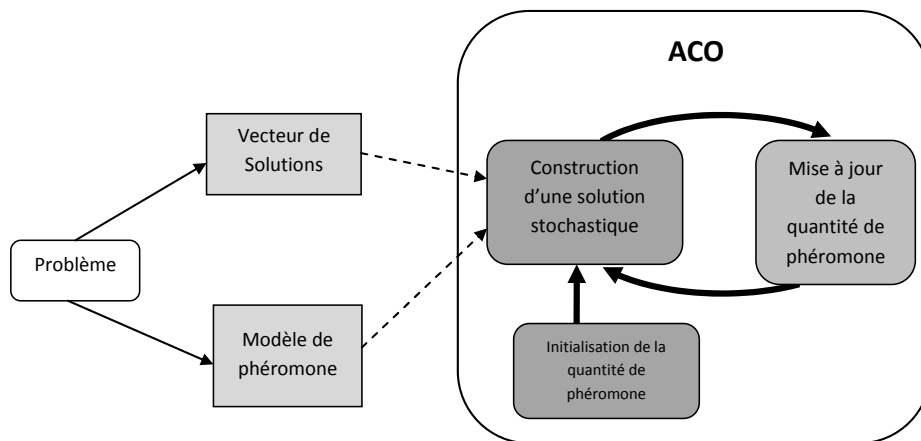


Figure 2.5 : Principe de base de l'algorithme ACO (Blum, 2005).

## 2.5 LES METAHEURISTIQUES EN SEGMENTATION D'IMAGES

Dans cette partie, nous allons décrire brièvement les travaux d'application des métaheuristiques au problème de la segmentation d'images. L'ouvrage de Siarry *et al.* (Siarry, et al., 2007) consacré à l'application des métaheuristiques en traitement d'images en général, montre l'importance de ces méthodes dans cette discipline. Nous invitons le lecteur à le consulter pour plus de détails sur les applications en général dans ce domaine.

### **Le recuit simulé**

Le recuit simulé adapté aux problèmes continus a été utilisé pour résoudre le problème de segmentation par l'approche paramétrique (Chap. 1, section 1.4). Le but est d'approcher l'histogramme à l'aide d'une combinaison de gaussiennes, dont il faut trouver les paramètres (Synder, et al., 1990).

L'algorithme du recuit simulé a été appliqué à la segmentation d'images par l'approche markovienne pour chercher les configurations les plus probables, correspondant aux états d'énergie minimale (Maitre, 2003).

Pour la détection de contours, le recuit simulé a été utilisé par Jamieson *et al.* pour trouver les paramètres optimaux d'un modèle de contours basé sur les B-splines, (Jamieson, et al., 2003).

### **Les algorithmes génétiques**

L'application des algorithmes génétiques à la segmentation d'images a commencé dans les années quatre-vingt-dix. A titre d'exemple, la thèse de Andrey (Andrey, 1997) et l'ouvrage de Bhanu *et al.* (Bhanu, 1994) sont entièrement dédiés à ce sujet.



Dans (Andrey, 1997), l'auteur a proposé une méthode de segmentation non supervisée basée sur "la relaxation sélectionniste", qui consiste à faire évoluer des populations d'unités capables de reconnaître des caractéristiques locales dans l'image. Au cours de cette évolution induite par un algorithme génétique, les populations envahissent progressivement, et de manière spécifique, les régions à segmenter.

Dans (Bhanu, et al., 1995), les auteurs ont proposé un système de segmentation de scènes qui optimise la qualité de l'image segmentée par le biais d'un algorithme génétique.

Dans (Majdi, et al., 2006) les auteurs proposent un algorithme de segmentation basé sur l'hybridation d'un algorithme génétique avec un recuit simulé, pour initialiser l'algorithme EM.

Récemment, Bazi *et al.* (Bazi, et al., 2007) ont présenté un algorithme de segmentation basé sur l'algorithme EM (Chap. 1, section 1.4.3), initialisé par un algorithme génétique de base de Goldberg (Goldberg, 1989).

### ***L'optimisation par essaim particulaire***

L'optimisation par essaim particulaire a été appliquée à la segmentation d'images en 2005 grâce aux travaux de Du Feng *et al.* (Feng, et al., 2005). La métaheuristique a été utilisée pour maximiser l'entropie de Shannon à deux dimensions (Chap. 1, section 1.3.2.2) afin de segmenter des images infrarouges. L'algorithme OEP mis en œuvre est celui développé par Kennedy (Kennedy, et al., 1995).

Récemment, dans (Yin, 2007), la métaheuristique OEP a été utilisée pour minimiser la distance de Kullback entre les distributions, représentant chacune une région, pour segmenter de manière supervisée des images. La version de l'algorithme OEP appliqué est celle proposée par Clerc *et al.* (Clerc, et al., 2002).

Zahara *et al.* (Zahara, et al., 2004) ont proposé une approche basée sur l'algorithme OEP hybridé avec un algorithme de recherche locale Simplex (Dréo, et al., 2003), appelé NM-PSO. L'algorithme d'optimisation est utilisé pour déterminer les paramètres des gaussiennes permettant d'approcher au mieux l'histogramme de l'image pour la segmentation (Chap. 1, section 1.4.1). Le schéma de cette hybridation est dans (Fan, et al., 2007). Cependant, le problème d'initialisation n'a pas été résolu et l'algorithme est curieusement très lent.

Dans (Bazi, et al., 2007) l'OEP est utilisé pour initialiser l'algorithme EM, décrit précédemment (Chap. 1, section 1.4.3), pour segmenter des images de circuits. Les résultats obtenus sont comparés à ceux de NM-PSO.

### ***Les algorithmes de colonies de fourmis***

Lors de notre recherche bibliographique, nous n'avons trouvé qu'un seul article qui traite de la méthode des colonies de fourmis pour la segmentation des images. Dans (Han, et al., 2007), l'approche utilisée pour segmenter les images est l'approche par classification, l'algorithme d'optimisation utilisé est celui proposé par Dorigo *et al.* (Dorigo, et al., 2005) ACO. La seule modification, par rapport à l'algorithme initial, est l'introduction d'un paramètre pour modifier la visibilité des fourmis. Cependant, les résultats obtenus ne sont pas très satisfaisants.

### ***L'optimisation multiobjectif en segmentation d'images***

L'optimisation multiobjectif a été utilisée par Bhanu *et al.* (Bhanu, 1994; Bhanu, et al., 1995), pour évaluer la qualité de la segmentation. Ils ont proposé un algorithme génétique non élitiste d'optimisation multiobjectif.

## 2.6 CONCLUSION

---

Dans ce chapitre, nous avons présenté une formulation de la segmentation des images en tant que problème d'optimisation en mono-objectif. Ensuite, nous avons proposé une reformulation du problème en optimisation multiobjectif.

Les principes de base de l'optimisation multiobjectif, ainsi que ses différentes approches, ont été exposés. Les trois approches décrites seront appliquées à la segmentation des images dans les chapitres suivants.

Comme nous l'avons souligné dans la section 2.2, les métaheuristiques, en optimisation mono-objectif, jouent un rôle important dans le domaine de la segmentation des images. De ce fait, le prochain chapitre sera consacré à notre contribution pour la segmentation des images par métaheuristiques en mono-objectif.

---

# CHAPITRE 3

## CONTRIBUTION A LA SEGMENTATION PAR METAHEURISTIQUES EN OPTIMISATION MONO-OBJECTIF

---

### 3.1 INTRODUCTION

---

Dans ce chapitre, nous présentons six méthodes de segmentation d'images basées sur des métaheuristiques. Notre but est de proposer de nouveaux critères de segmentation et d'adapter les métaheuristiques suivantes : recuit simulé (Nakib, et al., 2006a) , recuit microcanonique (Nakib, et al., 2007b), OEP (Nakib, et al., 2007c; Nakib, et al., 2007a), colonies de fourmis, et non pas de faire une étude comparative de ces différentes métaheuristiques. Le chapitre sera terminé par une étude comparative de résultats de segmentation sur des images synthétiques.

Les images réelles de test que nous avons choisies pour illustrer les différentes méthodes, sont présentées sur la figure 3.1. Ces images sont issues de la base de données de l'université de Berkeley (Berkeley, 2007). Nous faisons remarquer que le nombre de classes, dans ce chapitre, est supposé connu *a priori*.

Il est à signaler que nos algorithmes ont été codés sous Matlab, et exécutés sur un processeur Intel Pentium 4 à 2,26GHz.



(a)



(b)

Figure 3. 1: Images test en niveaux de gris. (a) *Maison*, (b) *Avion*.

### 3.2 APPLICATION DU RECUIT SIMULE CONTINU A LA SEGMENTATION

---

Comme nous l'avons signalé dans le premier chapitre, la segmentation d'image par seuillage s'attache à segmenter l'image en plusieurs classes, en n'utilisant que l'histogramme. A chaque pic de l'histogramme est associée une classe. Le seuillage d'un histogramme en  $n$  classes consiste à déterminer  $n-1$  seuils tels que chaque classe soit associée à un intervalle de niveaux de gris distinct.

Ces techniques sont très performantes dans le cas d'images simples, dont l'histogramme est multimodal.

En pratique, le problème d'approximation d'histogramme est un problème d'optimisation non linéaire avec des minima locaux. La figure 3.2 illustre cette problématique. Lors de l'utilisation d'un algorithme de recherche local (Gauss-Newton) pour la segmentation (Synder, et al., 1990), sur les figures 3.2 (a) et (c), la méthode permet d'avoir un bon résultat de segmentation et une bonne approximation de l'histogramme, respectivement, grâce à une bonne initialisation de l'algorithme de Gauss-Newton. Cependant, sur les figures 3.2. (b) et (d), les résultats obtenus sont moins bons, du fait d'une mauvaise initialisation de l'algorithme. L'objet de cette section est d'introduire la métaheuristique du recuit simulé, adaptée au cas continu (RSC), afin d'améliorer les performances du seuillage d'images à partir de l'histogramme, tout en évitant la phase critique d'initialisation de l'algorithme d'optimisation. Ce problème a déjà été traité dans la littérature avec l'approche OEP (Chap. 2, section 2.4). Cependant, nous avons voulu adapter la métaheuristique du recuit simulé, notamment pour améliorer les performances de la segmentation et garder l'avantage de l'utiliser comme une boîte noire, notamment pour les biologistes.

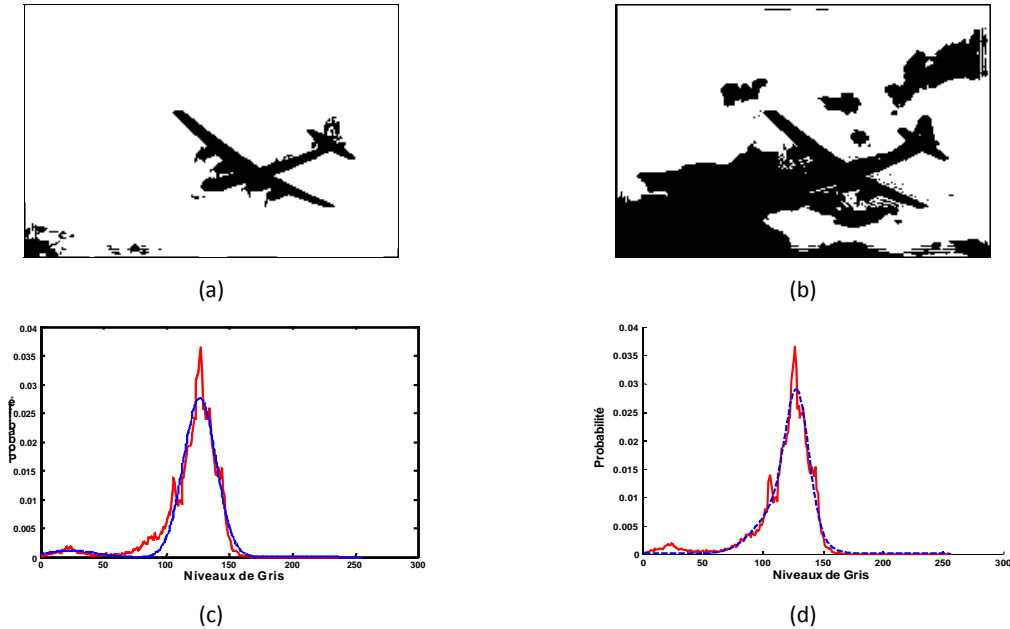


Figure 3. 2: Seuillage paramétrique avec un algorithme de descente de gradient. (a) Image segmentée ( $t=79$ ), (b) Image segmentée ( $t=116$ ), (c) Histogramme original et approché par la somme de deux gaussiennes avec une initialisation manuelle, (d) Approximation de l'histogramme par deux gaussiennes avec une initialisation aléatoire. L'image originale est présentée sur la figure 3.1 (b).

### 3.2.1 ALGORITHME DE SEGMENTATION

Nous rappelons que, pour un histogramme multimodal  $h(x)$  d'une image donnée, où  $x$  représente la valeur du niveau de gris, le problème consiste à déterminer les seuils optimaux qui séparent les différentes classes de l'image. La technique la plus utilisée pour trouver ces seuils optimaux, dans le but de segmenter l'image, consiste à reconstruire l'histogramme de l'image sous la forme de la somme de  $d$  fonctions de densité de probabilité (fdp). Dans le cas où ces fdp sont gaussiennes, le modèle de l'histogramme peut alors s'écrire:

$$h(\Theta, x_i) = \sum_{i=1}^d \frac{P_i}{\sqrt{2\pi}\sigma_i} \exp\left[-\frac{(x - \mu_i)^2}{\sigma_i^2}\right] \quad (3.1)$$

où  $P_i$  représente la probabilité *a priori*,  $\mu_i$  la moyenne et  $\sigma_i^2$  la variance du mode  $i$ . Dans le cas bimodal ( $d=2$ ), le seuil optimal est déterminé par la minimisation de la probabilité d'erreur d'affectation d'un niveau de gris  $x$  à la première classe, sachant qu'il appartient à la deuxième classe:

$$E(t) = P_1 \int_{-\infty}^t p_2(x) dx + P_2 \int_t^{+\infty} p_1(x) dx \quad (3. 2)$$

où  $t$  est le seuil et  $p_i(x)$  est la fonction de densité de probabilité du mode  $i$ ,  $\Theta$  est le vecteur de paramètres des fdp des gaussiennes:

$$\Theta = \{P_j, \mu_j, \sigma_j; \quad j = 1, 2, \dots, d\} \quad (3. 3)$$

et  $i \in [0 \ L-1]$ , en désignant par  $L$  le nombre total de niveaux de gris présents dans l'image originale.

Pour un histogramme donné, l'algorithme qui permet de trouver le seuil optimal est détaillé dans (section 1.4.2) Avant de rechercher les seuils optimaux, il faut déterminer les paramètres des gaussiennes qui permettent de reconstruire l'histogramme de l'image d'origine. Les deux critères les plus utilisés sont le critère du maximum de vraisemblance et celui de l'erreur quadratique moyenne. Pour un histogramme donné, l'erreur de reconstruction est définie par l'expression suivante:

$$J = \frac{\sum_i |h_i - h(\Theta, x_i)|}{\sum_i h_i} \quad (3. 4)$$

où  $J$  est la fonction objectif à minimiser. Pour minimiser la fonction objectif  $J$ , nous avons adapté le recuit simulé en vue d'une recherche dans un espace continu (RSC). Après avoir développé les principes du recuit simulé dans le chapitre précédent (section 2.4.1), nous présentons seulement les modifications que nous avons apportées.

Plusieurs travaux ont été publiés, portant sur l'adaptation du recuit simulé afin de résoudre des problèmes d'optimisation en continu (Dréo, et al., 2003), dans différents domaines d'applications (électronique (Siarry, et al., 1997), chimie (Agrafiotis, 2001), ...). Pour plus d'informations, on pourra consulter (Dréo, et al., 2003).

Notre approche consiste à ajuster la valeur de la constante de Boltzmann durant la simulation, de telle sorte que la probabilité d'accepter les solutions dégradantes à la température finale soit un nombre "faible" prédéfini (0,5%). Pour adapter le recuit simulé au cas continu, nous définissons un paramètre de contrôle de la discrétisation de l'espace de recherche. Nous ajustons automatiquement ce paramètre, en fonction du taux d'acceptation des modifications tentées, tout au long de la descente de température. Enfin, nous avons défini un pas de discrétisation de l'espace de recherche (un pas différent pour chaque paramètre).

L'adaptation que nous avons faite se résume en ces points :

- Pour chaque valeur du pas de discrétisation: définir la loi de calcul du mouvement d'une variable.
- Ajuster le pas de discrétisation afin de maintenir constante l'efficacité de l'optimisation.
- A la fin de chaque transition:
  - o Mise à jour de la moyenne des énergies de transition ( $\overline{\Delta E}$ ).

- Ajustement de la valeur de  $K_B$ :  $K_B = -\left(\overline{\Delta E} / T\right) \cdot \ln(a)$ , de manière à ce que la probabilité d'accepter les solutions dégradantes à la température finale soit un petit nombre  $a$  prédéfini ( $0 < a < 1$ ).

En ce qui concerne les critères d'arrêt, nous avons utilisé les mêmes critères que ceux d'un recuit simulé classique.

### 3.2.2 RESULTATS

A titre d'exemple, la segmentation de l'image d'*avion*, en niveaux de gris, de la figure 3.1 (b) dont l'histogramme est bimodal, est donnée sur la figure 3.3 (a). Cet exemple illustre les performances de la méthode basée sur le RSC, en comparaison avec celles procurées par la technique d'optimisation de Gauss-Newton (figure 3.2 (a)). Le RSC a permis de reconstruire l'histogramme avec des pics bien positionnés, comme le montre la figure 3.3 (b). Le tableau 3.1 contient les valeurs des paramètres, obtenues avec le RSC, avec une initialisation aléatoire de l'algorithme.

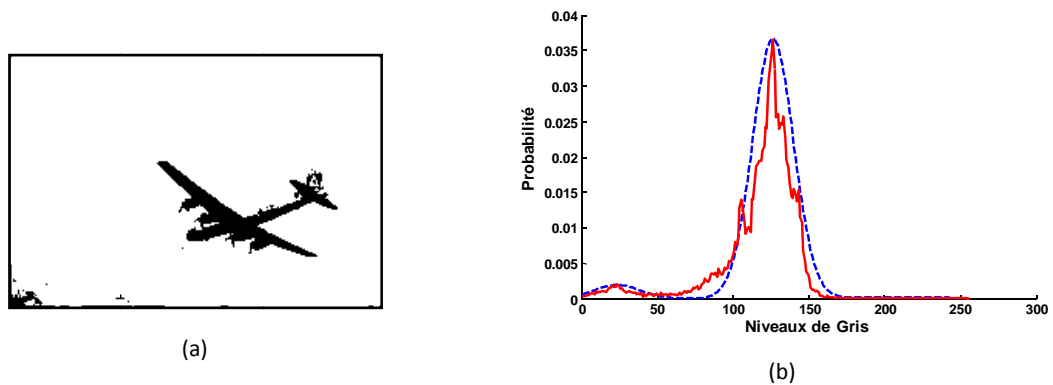


Figure 3. 3: Illustration de la méthode basée sur le RSC. (a) Image segmentée ( $t=72$ ), (b) Histogramme approché par deux gaussiennes. L'image originale est présentée sur la figure 3.1 (b).

Tableau 3. 1: Résultats d'approximation des histogrammes par RSC de la figure 3.4.

Paramètre	Intervalle	Valeur finale	
		Figure 3.2	Figure 3.3
$P_1$	[0 1]	0,0019	0,0254
$P_2$	[0 1]	0,0365	0,0933
$\mu_1$	[0 255]	24	110
$\mu_2$	[0 255]	127	201
$\sigma_1$	[0 100]	21,32	18,82
$\sigma_2$	[0 100]	19,14	16,74
$J$	-	4,99 e-4	3,34 e-3

Un deuxième exemple, est présenté sur la figure 3.4, où le but est de segmenter la *maison* présentée sur la figure 3.1 (a). L'histogramme n'est pas correctement approché, ceci est dû au nombre insuffisant de gaussiennes utilisées. Toutefois, le résultat nous paraît acceptable. Il est à signaler que les résultats présentés sont la moyenne de 20 exécutions successives.

### 3.2.3 BILAN

Dans cette section, nous avons introduit le recuit simulé continu (RSC) pour améliorer les performances de la technique de seuillage, basée sur la minimisation de l'erreur d'affectation et

l'approximation de l'histogramme. Les résultats obtenus montrent que la méthode permet d'avoir une modélisation d'histogramme plus favorable, en vue d'une segmentation optimale de l'image sans aucune initialisation particulière de l'algorithme. Toutefois, l'algorithme est assez lent (en moyenne 25 sec). Pour résoudre ce problème, une hybridation avec un algorithme de recherche locale pourrait accélérer la convergence de l'algorithme. Enfin, plus d'exemples d'illustrations des performances de cette méthode sont présentés dans notre article (Campbell, 1996).

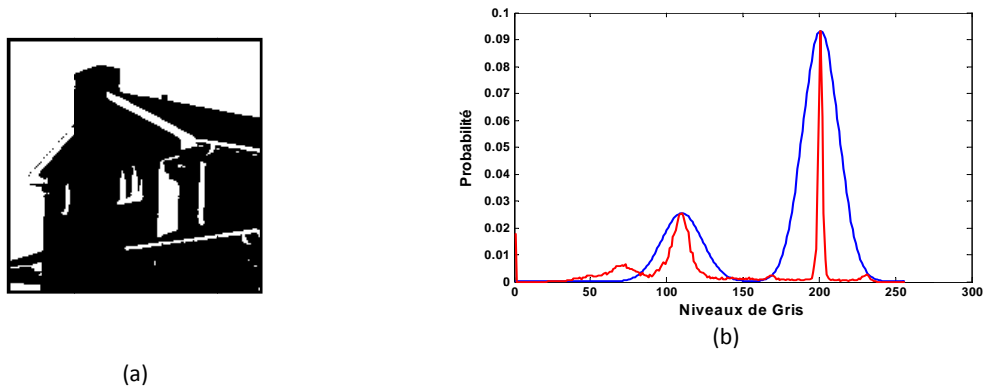


Figure 3. 4: Exemple de segmentation par RSC, (a) Image segmentée ( $t=150$ ), (b) Approximation de l'histogramme avec deux gaussiennes. L'image originale est présentée sur la figure 3.1 (a).

### 3.3 SEGMENTATION PAR ENTROPIE A DEUX DIMENSIONS (E2D) ET RECUIT MICROCANONIQUE

Dans cette section, nous présentons la méthode de segmentation d'images par analyse d'histogramme, et seuillage par maximisation de l'entropie de Shannon à deux dimensions (section 1.3.2.2). Afin de remédier au défaut des algorithmes de recherche locale, qui peuvent s'arrêter au premier maximum d'entropie de Shannon rencontré, et au défaut des métaheuristiques, qui ne garantissent pas la reproduction des solutions, nous mettons en œuvre la métaheuristique basée sur le recuit microcanonique. Dans l'espace exploré, la recherche des seuils de segmentation optimums s'effectue par paliers d'énergie, en gravitant autour des meilleures solutions potentielles. Le temps de convergence s'en trouve amélioré et la reproductibilité des résultats est mieux garantie.

#### 3.3.1 ALGORITHME DE SEGMENTATION

Notre algorithme est directement inspiré du recuit microcanonique, en raison du succès rencontré par ce dernier dans la résolution des problèmes de grande taille. Ceci justifie notre choix de l'appliquer à la segmentation par seuillage multiniveau, du fait de l'étendue de l'espace des solutions, notamment lorsque le nombre de classes croît. La recherche de la solution optimale est basée sur la maximisation de l'entropie à deux dimensions, que nous avons définie dans la section 1.3.2.2.

Le principe de l'entropie 2D, décrit précédemment, peut facilement être généralisé à la segmentation multiniveau, compte tenu de l'expression de l'entropie totale pour un ensemble de seuils :

$$H^T[(t_1, s_1), (t_2, s_2), \dots, (t_n, s_n)] = \sum_{i=1}^n H(t_i, s_i) \quad (3.5)$$

où

$$H(s_n, t_n) = - \sum_{i=s_n}^{s_{n+1}-1} \sum_{j=t_n}^{t_{n+1}-1} \frac{p_{ij}}{P_1} \log \frac{p_{ij}}{P_1} \quad (3.6)$$

Sachant que l'échelle des niveaux de gris s'étale de 0 à 255, nous ajoutons à l'ensemble des seuils de segmentation, deux seuils fictifs  $(s_0, t_0) = (1, 1)$  et  $(s_{n+1}, t_{n+1}) = (255, 255)$ , afin de représenter toutes les frontières des classes de l'histogramme.

Nous rappelons que le vecteur des seuils de segmentation potentiels est une suite ordonnée croissante:  $t_0 < t_1 < \dots < t_{n+1}$  (idem pour les seuils  $s_i$  dérivant de l'image moyennée). Pour résumer, le problème de la segmentation revient à chercher dans l'espace des vecteurs de segmentation potentiels celui qui maximise l'entropie totale donnée par l'expression (3.5) :

$$(t^*, s^*) = \text{Max } H_T(t, s) \quad (3.7)$$

$$\text{s. c. } s_0 < s_1 < s_2 < \dots < s_{n+1} \text{ et } t_0 < t_1 < t_2 < \dots < t_{n+1}$$

La procédure Algorithme 3.1 décrit le processus général de segmentation basé sur le recuit microcanonique maximisant l'entropie 2D.

Algorithme 3 1: Algorithme de segmentation par recuit microcanonique.

- 1. Initialisation**
  - 1.1.** Définir le pourcentage minimal  $p$  de transitions acceptées au cours du premier palier.
  - 1.2.** Fixer  $E_t$  initiale telle que les transformations testées soient acceptées dans la proportion  $p$ .
  - 1.3.** Tirer aléatoirement une solution acceptable et calculer son énergie  $E$  en utilisant l'expression (3.5).
  - 1.4.** Choisir le nombre maximal de transitions testées à chaque palier de l'énergie totale.
- 2. Tant que** le nombre de solutions acceptées est non nul.
  - 2.1. Répéter** pour un nombre d'itérations donné,
    - 2.1.1. Tirer aléatoirement une transition, qui ***n'est pas présente dans la liste tabou*** et qui ***satisfait les contraintes***.
    - 2.1.2. Calculer la variation  $\Delta E$  d'énergie associée.
    - 2.1.3. Accepter la transition si :  $\Delta E \leq E_t - E$
    - 2.1.4. Si  $\Delta E \leq 0$ , comparer le nouvel état au meilleur état trouvé depuis le début de la recherche, et le mémoriser s'il est meilleur.
  - 2.2. Décroître**  $E_t$ :  $E_t = \alpha E_t$
- 3. Afficher** le meilleur état rencontré au cours de la recherche.

### **Initialisation de l'algorithme**

- définir le pourcentage minimal  $p$  (70%) de transitions acceptées au cours du premier palier ;
- fixer  $E_t$  initiale telle que  $p\%$  des solutions testées soient acceptées ;
- tirer aléatoirement une solution acceptable et calculer son énergie  $E$  ;
- choisir le nombre maximal de transitions testées à chaque palier d'énergie totale. Nous avons opté pour 1000 transitions ;
- choisir le paramètre de décroissance de la température  $\alpha$  (0,9).

Notre contribution dans cet algorithme consiste à ajouter une "liste tabou", qui contient toutes les configurations déjà évaluées. Cette modification nous permet d'éviter de revisiter les configurations déjà évaluées, afin de réduire le nombre d'évaluations de la fonction objectif.



### 3.3.2 RESULTATS

Nous présentons sur les figures 3.5 (a) et (b), les allures des *fitness* avec une recherche exhaustive des valeurs optimales de  $t$  et  $s$ , pour les images originales de la *maison* et de l'*avion*.

Comme nous pouvons le remarquer, les allures des fonctions varient d'une image à l'autre, et avec des maxima locaux. Les résultats de segmentation des images test, *maison* et *avion*, sont représentés sur les figures 3.5 (c) et (d), respectivement. D'un point de vue visuel, la qualité de la segmentation obtenue montre que le critère du maximum de l'entropie 2D de Shannon est insuffisant pour avoir de bons résultats, notamment sur l'image de la *maison*. Cependant, une métaheuristique peut trouver une solution sous-optimale qui pourrait donner de meilleurs résultats lorsque le critère n'est pas bien adapté aux images traitées.

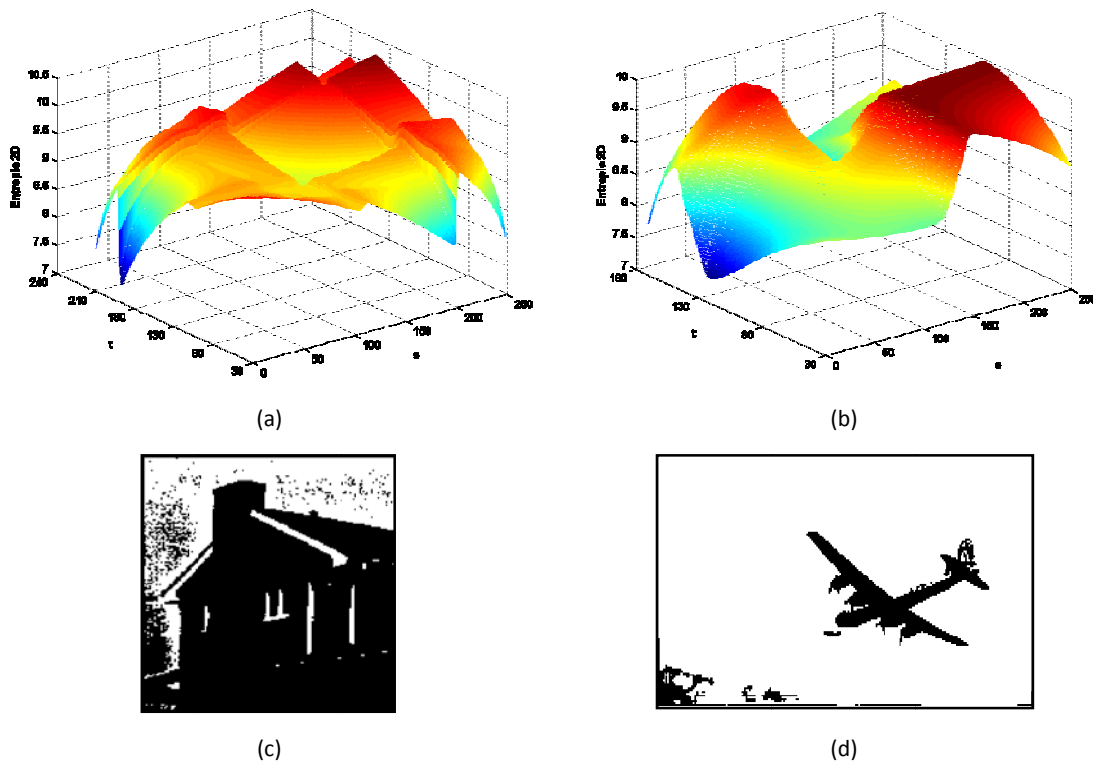


Figure 3. 5: *Fitness* et résultats de la segmentation lors d'une recherche exhaustive du maximum de l'entropie à 2D. (a) Image *maison* ( $t=198$ ;  $s=250$ ), (b) Image *avion* ( $t=81$ ;  $s=156$ ), (c) Image *maison* segmentée avec  $t=198$ , (d) Image *avion* segmentée avec  $t=81$ .

La figure 3.6 présente les résultats de segmentation en utilisant le recuit microcanonique. Les solutions présentées sont une moyenne de 20 exécutions successives de l'algorithme. Les résultats de la segmentation en deux classes, des images *maison* et *avion*, sont présentés respectivement sur les figures 3.6 (a) et (b). Dans ce cas, nous pouvons remarquer que, sur l'image *maison*, le résultat est meilleur que sur la figure 3.5. (a), même s'il n'est pas encore parfait. Concernant l'image *avion*, les deux seuils optimaux trouvés sont presque identiques: 80 et 81. Dans le cas d'une segmentation multiniveau (3 classes), les résultats obtenus sont représentés sur les figures 3.6 (c) et (d).

En terme de temps de calcul, nous pouvons noter que la méthode produit un résultat de segmentation en deux classes, en moyenne, au bout de moins de 1000 évaluations de la fonction objectif. Tandis que, dans le cas de trois classes, la méthode converge vers un résultat, en moyenne, au bout de moins de 4000 évaluations. En revanche, cet algorithme est ralenti par la liste tabou. Il est à signaler que la recherche est effectuée de façon aléatoire.

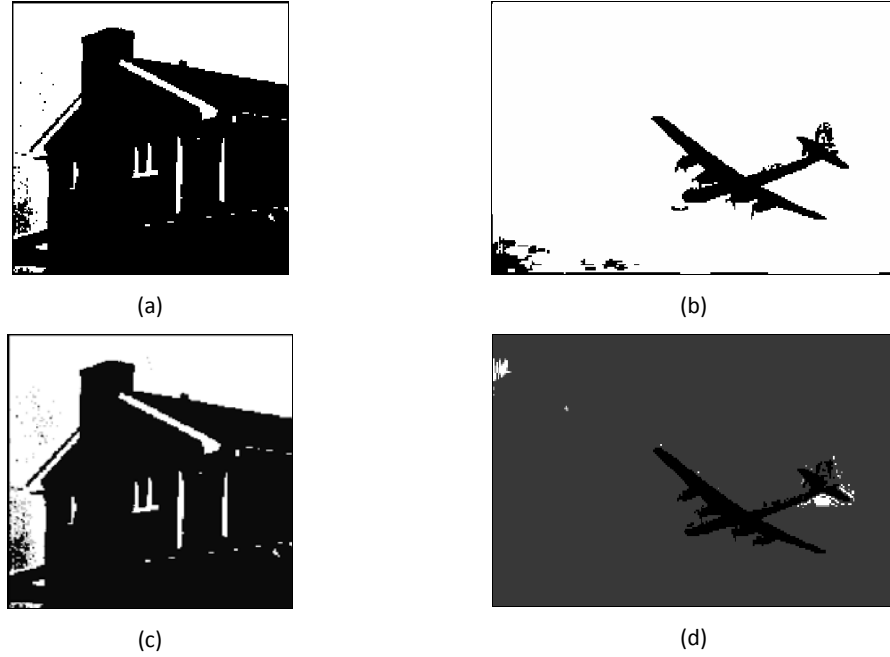


Figure 3. 6: Résultats de la segmentation par recuit microcanonique. (a) Image segmentée  $t=196$ , (b) Image *avion* segmentée  $t=80$ ;  $s=152$ , (c) Image *maison* segmentée en 3 classes  $t=(11; 197)$ , (d) Image *avion* segmentée en 3 classes  $t=(56; 153)$ .

### 3.3.3 BILAN

L'utilisation du recuit microcanonique permet d'avoir de bons résultats de segmentation et une reproductibilité meilleure, du fait de l'absence de caractère stochastique dans la recherche de la solution optimale. Le seul caractère probabiliste est celui de la recherche dans le voisinage. Du point de vue vitesse de convergence, cet algorithme est plus rapide que le recuit simulé, ce qui est en accord avec la comparaison faite dans (Herault, et al., 1993).

## 3.4 SEGMENTATION PAR ENTROPIE EXPONENTIELLE A DEUX DIMENSIONS (eE2D) ET RECUIT MICROCANONIQUE HYBRIDE

Dans cette section, nous présentons un algorithme basé sur une mesure d'information appelée l'"entropie exponentielle". Cet algorithme permet de pallier aux problèmes de l'entropie de Shannon, celle-ci n'est pas définie pour les probabilités nulles. Afin d'accélérer la convergence de l'algorithme du recuit microcanonique (MC), nous avons hybridé avec ce dernier l'algorithme du simplexe de Nelder-Mead (NM). Nous allons présenter l'entropie exponentielle et l'étendre au cas bidimensionnel, afin de prendre en compte l'information spatiale (Nakib, et al., 2007b).

### 3.4.1 ENTROPIE EXPONENTIELLE A DEUX DIMENSIONS

L'entropie exponentielle, proposée par Campbell (Campbell, 1996), est donnée par :

$$B^{(\alpha)} = \left( \int_{-\infty}^{+\infty} f^{\alpha}(x) dx \right)^{1/(1-\alpha)} \quad (3.8)$$

où  $\alpha \in \mathbb{R}^{*+}$  et  $\alpha \neq 1$ .

L'entropie exponentielle est liée à l'entropie classique de Shannon par la relation suivante :

$$H_{sh}(X) = \lim_{\alpha \rightarrow 1} \{ B^{(\alpha)}(X) \} \quad (3.9)$$

où  $H_{sh}$  est l'entropie de Shannon et  $B^{(\alpha)}$  est l'entropie exponentielle proposée par Campbell. Après discrétisation et extension au cas bidimensionnel, l'expression ci-dessus de l'entropie exponentielle d'une image s'écrit:

$$B^{(\alpha)} = \left( \sum_{i=0}^{255} \sum_{j=0}^{255} P_{ij}^{\alpha} \right)^{1/(1-\alpha)} \quad (3.10)$$

où les  $p_{ij}$  sont les probabilités des pixels dans l'histogramme à deux dimensions (voir section 1.3.2.2.1).

D'où l'entropie exponentielle à deux dimensions (eE2D), associée aux distributions des différentes classes de l'image:

- L'entropie 2D de la classe  $m-1$  :

$$B^{(m-1)}[a_{n-1}, a_n] = \left( \sum_{i=s_{n-1}}^{s_n-1} \sum_{j=t_{n-1}}^{t_n-1} \left( \frac{P_{ij}}{P_m[a_{n-1}, a_n]} \right)^{\alpha} \right)^{1/(1-\alpha)} \quad (3.11)$$

où  $a_n \equiv (s_n, t_n)$ ,  $n=1, \dots, N-1$ ,  $m=2, \dots, N$  et  $N$  est le nombre de classes.

- L'entropie 2D de la classe  $m$  :

$$B^{(m)}[a_n, a_{n+1}] = \left( \sum_{i=s_n}^{s_{n+1}-1} \sum_{j=t_n}^{t_{n+1}-1} \left( \frac{P_{ij}}{P_m[a_n, a_{n+1}]} \right)^{\alpha} \right)^{1/(1-\alpha)} \quad (3.12)$$

L'entropie totale s'écrit :

$$B_{\alpha}^T[a_1, \dots, a_{N-1}] = \sum_{i=0}^{N-1} B_{\alpha}^{(i+1)}[a_i, a_{i+1}] \quad (3.13)$$

En appliquant le principe du maximum de l'entropie, les vecteurs de seuils optimaux  $(a_{1, \dots, N-1}^*) \equiv ((s_1^*, t_1^*), \dots, (s_{N-1}^*, t_{N-1}^*))$  doivent vérifier le critère suivant :

$$B_{\alpha}^T(a_{1, \dots, N-1}^*) = \max \{ B_{\alpha}^T(a_{1, \dots, N-1}) \} \quad (3.14)$$

s. c.  $1 < s_1 < s_2 < \dots < s_{N-1} < 255$  et  $1 < t_1 < t_2 < \dots < t_{N-1} < 255$

Comme nous l'avons précisé dans la section précédente, ce problème d'optimisation requiert l'utilisation d'une métaheuristique. Nous avons aussi constaté que l'algorithme du recuit microcanonique permet d'avoir de bons résultats mais, malheureusement, il est relativement long. Afin de résoudre ce problème, nous avons hybridé le recuit microcanonique avec l'algorithme simplex (NM-MC), que nous allons détailler dans la section suivante.

### 3.4.2 ALGORITHME DE SEGMENTATION PAR RECUIR MICROCANONIQUE HYBRIDE

Dans cette section, nous allons décrire l'algorithme du simplex que nous avons hybridé avec l'algorithme du recuit microcanonique. Les différentes étapes de l'algorithme sont résumées dans l'Algorithme 3.2.

#### 3.4.2.1 ALGORITHME DU SIMPLEX

C'est un algorithme d'optimisation locale, très utilisé en optimisation, notamment en hybridation avec une métaheuristique, comme présentés dans (Dréo, et al., 2003; Zahara, et al., 2004) où les auteurs hybrident respectivement l'OEP et les colonies de fourmis avec le simplex. Proposé par Nelder et Mead en 1965 (Nelder, et al., 1965), il est très simple, efficace et ne dépend pas des dérivées de la fonction à optimiser. Il manipule un ensemble de points sous la forme d'un "simplex"

non dégénéré. Un "simplex" est une figure géométrique dont le volume, de  $n$  dimensions, est non nul, et représente une enveloppe convexe de  $n+1$  dimensions. La méthode du simplex consiste à déformer la figure géométrique de quatre manières différentes : réflexion (paramètre  $\delta$ ), expansion ( $\gamma$ ), contraction ( $\beta$ ), et rétrécissement (multicontraction) ( $\beta$ ). Ces paramètres ont généralement les valeurs suivantes :

$$\delta = 1; \quad \gamma = 0,5; \quad \beta = 0,5 \quad (3.15)$$

Les différents mouvements des sommets du simplex sont engendrés selon par les formules suivantes :

$$\text{réflexion: } x_r = (1 - \delta)\bar{x} - x_{n+1}$$

$$\text{expansion: } x_e = \gamma x_r + (1 - \gamma)\bar{x}$$

$$\text{contraction interne: } x_c = \beta x_{n+1} + (1 - \beta)\bar{x}$$

$$\text{contraction externe : } x_c = \beta x_r + (1 - \beta)\bar{x}$$

où  $\bar{x}$  est le barycentre du simplex, défini comme suit :

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (3.16)$$

Algorithme 3.2: Algorithme de segmentation par recuit microcanonique hybridé (NM-MC).

- 1. Initialisation**
  - 1.1.** Définir le pourcentage minimal  $p$  de transitions acceptées dans le premier palier.
  - 1.2.** Fixer  $E_i$  initiale telle que les transformations testées soient acceptées dans la proportion  $p$ .
  - 1.3.** Tirer aléatoirement une solution acceptable et calculer son énergie  $E$  en utilisant l'expression (3.5).
  - 1.4.** Choisir le nombre maximal de transitions testées à chaque palier de l'énergie totale.
- 2. Tant que** le nombre de solutions acceptées est non nul.
  - 2.1. Trouver une solution locale au problème (3.14) avec la méthode du Simplex.**
  - 2.2. Répéter** jusqu'à  $Nbmaxtest$ 
    - 2.2.1.** Tirer aléatoirement une transition, qui n'existe pas dans la liste tabou et qui satisfait les contraintes.
    - 2.2.2.** Calculer la variation  $\Delta E$  d'énergie associée.
    - 2.2.3.** Accepter la transition si :  $\Delta E \leq E_t - E$
    - 2.2.4.** Si  $\Delta E \leq 0$ , comparer le nouvel état au meilleur état trouvé depuis le début de la recherche, et le mémoriser s'il est meilleur.
  - 2.3. Décroître**  $E_t$ :  $E_t = \varphi E_t$
- 3. Afficher** le meilleur état rencontré au cours de la recherche.

L'algorithme commence par modifier seulement le point où la fonction objectif est grande en déterminant un autre point à partir du plus mauvais des points du simplex (opération de réflexion). Si le nouveau point est meilleur que tous les autres, le simplex est étendu dans sa direction. En revanche, s'il est au moins meilleur que le plus mauvais point, l'opération de réflexion est répétée, mais cette fois avec le nouveau mauvais point. L'opération de contraction est effectuée dans le cas où le plus mauvais point est au moins aussi bon que le point réfléchi, de manière à ce que le simplex

s'adapte à la forme de la fonction et entoure l'optimum. Si le plus mauvais point est meilleur que le point contracté (contraction rejetée), une multicontraction est effectuée. Tous les sommets du simplex sont remplacés par  $(x_j - x_1)/2$ , un rétrécissement du simplex est ainsi obtenu. La méthode est illustrée sur la figure 3.7, dans le cas d'une recherche dans  $\mathfrak{R}^3$ . Les quatre types de mouvements sont conçus pour que le simplex puisse suivre le gradient de la function objectif.

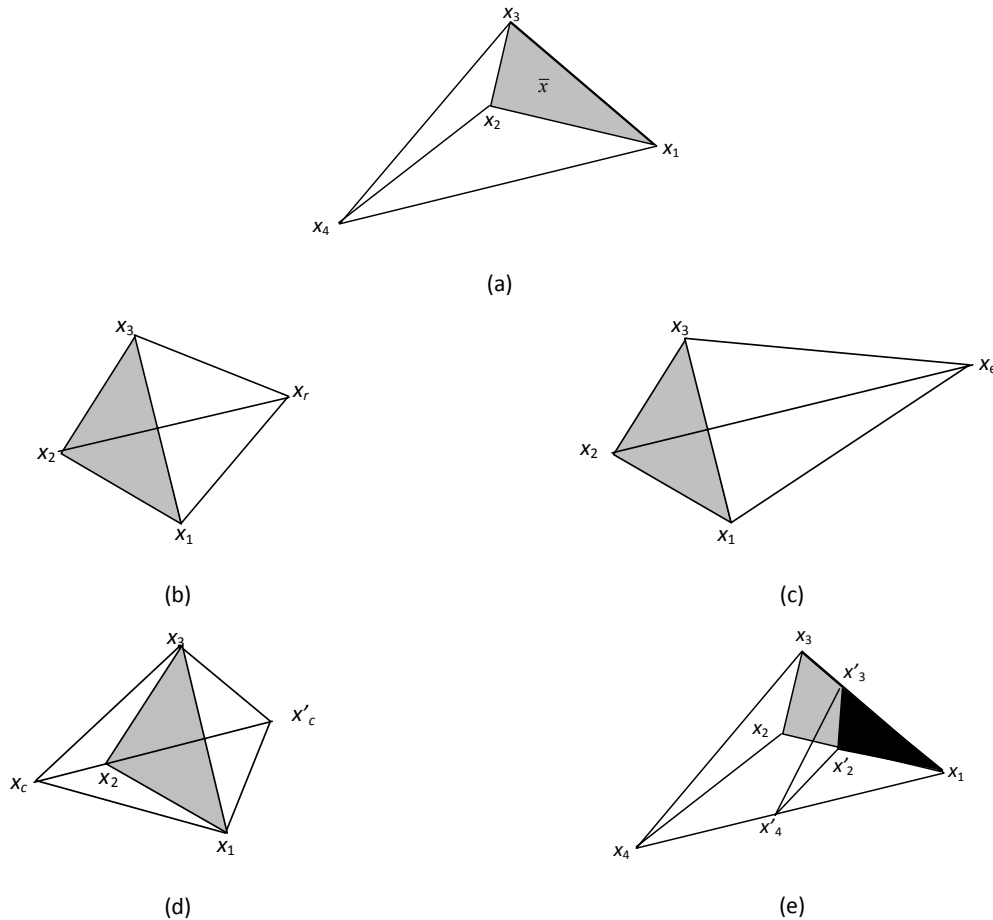


Figure 3. 7: Les mouvements possibles dans la méthode de Nelder-Mead dans un espace de recherche à 3 dimensions ( $n=3$ ). (a) Simplex au départ, (b) Réflexion, (c) Expansion, (d) Contraction ( $x_c$  interne,  $x'_c$  externe), (e) Multicontraction.  $x_j$  et  $x_4$  représentent les sommets où la valeur de la fonction est la plus basse et la plus haute, respectivement.

### 3.4.2.2 HYBRIDATION DU RECUIT MICROCANONIQUE POUR LA SEGMENTATION

L'algorithme de segmentation consiste en l'hybridation du recuit microcanonique avec le simplex (NM-MC). Cette hybridation consiste à effectuer une recherche locale de la solution optimale à chaque changement de palier d'énergie. La solution trouvée par le simplex est ensuite perturbée de façon aléatoire. Afin de respecter les contraintes, nous donnons le plus mauvais score aux résultats qui ne les respectent pas. Par exemple, si le résultat d'une expansion est un paramètre négatif, l'évaluation de la fonction objectif n'est pas effectuée, et la valeur 0 lui est attribuée. Il est à noter que nous avons transformé le problème de maximisation en un problème de minimisation.

L'algorithme d'optimisation ne nécessite aucune initialisation particulière. La fonction objectif, résumant le problème de segmentation d'image par l'entropie exponentielle, est donnée par l'expression (3.14).

Le tableau (3.2) récapitule les différents paramètres de l'algorithme et les valeurs que nous leur avons attribuées sur la base des différents tests que nous avons effectués.

### 3.4.3 RESULTATS

Tout d'abord, nous allons présenter les résultats obtenus avec l'entropie de Shannon ( $H_{Sh}$ ) et l'algorithme hybride NM-MC. Ceci nous permettra de voir l'utilité de l'hybridation du recuit microcanonique. Ces résultats sont présentés sur la figure 3.8. Dans le cas de la segmentation en deux classes, la qualité de la segmentation de l'algorithme NM-MC (figure 3.8(a)) est, d'un point de vue visuel, meilleure que celle de l'algorithme MC (figure 3.6 (a)).

Tableau 3. 2: Valeurs des paramètres de NM-MC.

Paramètres	Valeurs
$Nbmaxtest$	10
$\rho$	90%
$\varphi$	0,99
$\alpha$	0,5
$\delta, \gamma, \beta$	1,2; 0,5; 0,5



Figure 3. 8: Segmentation de l'image *maison* par  $H_{Sh}$  en : (a) Deux classes  $t=197$ ;  $s=254$ , (b) Trois classes  $t=(30;194)$  ;  $s=(8;251)$ .

Dans le cas d'une segmentation en trois classes, le résultat obtenu par NM-MC est également supérieur, mais pas encore assez bon, car il ne restitue pas toute l'information utile au regard de l'image d'origine. Ce dernier résultat montre les limites de la méthode basée sur le critère  $H_{Sh}$ , quand le nombre de classes augmente.

Nous décrivons les résultats de la segmentation, basés sur l'entropie exponentielle à deux dimensions d'ordre  $\alpha$ . Dans un premier temps, nous allons examiner les variations de la qualité de la segmentation en fonction de l'ordre de l'entropie exponentielle ( $\alpha$ ). Sur la figure 3.9, nous présentons les variations de la fonction objectif et les résultats optimaux de segmentation, obtenus par une recherche exhaustive.

Pour évaluer la qualité de la segmentation, un critère basé sur la mesure d'homogénéité des classes de l'image segmentée est utilisé (Sezgin, et al., 2004). Ce critère est défini par :

$$u = 1 - 2(N-1) \sum_{j=1}^N \sum_{i \in C_j} (I_i - \mu_i)^2 / M \cdot (I_{max} - I_{min})^2 \quad (3.17)$$

où  $N$  est le nombre de classes,  $C_j$  la classe  $j$ ,  $M$  le nombre de pixels dans l'image,  $I_i$  le niveau de gris du pixel  $i$ ,  $\mu_i$  le niveau de gris moyen des pixels dans l'image,  $I_{min}$  est le niveau de gris le plus faible dans l'image originale et  $I_{max}$  le niveau le plus élevé. L'uniformité est dite bonne si  $u$  est proche de 1 et vice versa.

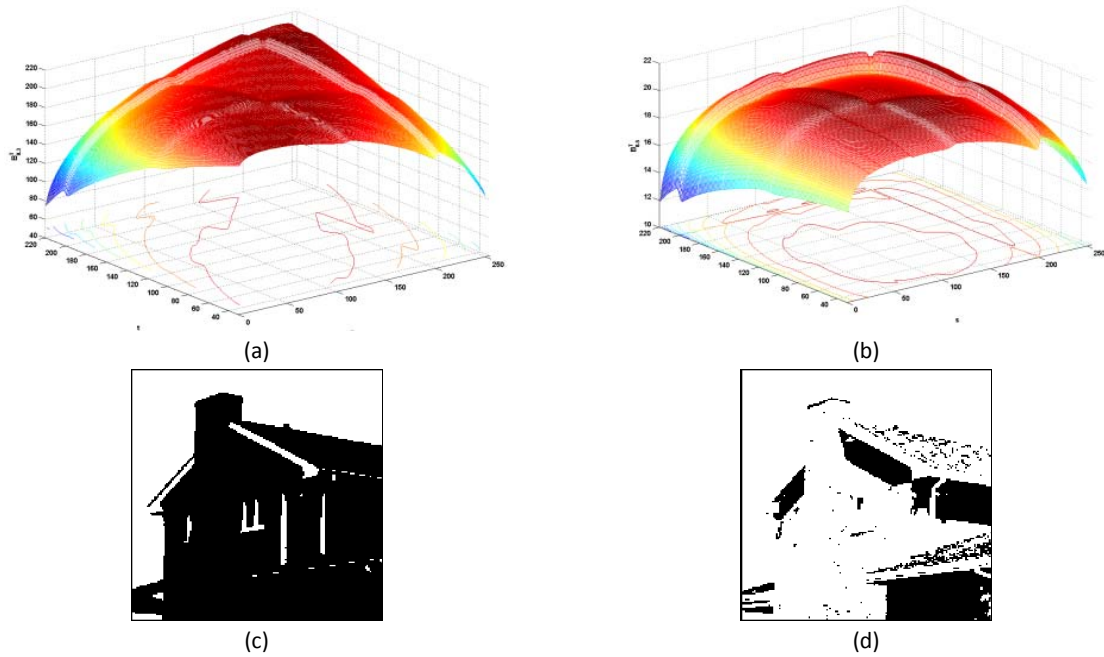


Figure 3. 9 Allure de l'eE2D pour différentes valeurs de  $\alpha$  et le résultat de segmentation correspondante avec une recherche exhaustive pour l'image *maison*. (a) Allure de l'eE2D pour  $\alpha=0.3$ , (c) Image segmentée avec le seuil ( $t=192$ ;  $s=206$ ), (b) Allure de l'eE2D pour  $\alpha=0.5$ , (d) Image segmentée avec le seuil ( $t=96$ ;  $s=118$ ).

La figure 3.10 présente les variations de l'ordre de l'eE en fonction des seuils optimaux de segmentation de l'image *maison*, pour différentes valeurs de  $\alpha$ . Comme nous pouvons le remarquer sur l'exemple de la figure 3.10, les valeurs optimales de l'eE2D sont sensibles aux changements de valeurs de  $\alpha$ . D'après la courbe, la valeur optimale de  $\alpha$  est de 0,3 ( $u=0,68$ ), ce qui correspond bien à une bonne segmentation, d'un point de vue visuel. Les valeurs optimales procurées par l'eE2D sont sensibles aux variations  $\alpha$ , quand ce dernier varie avec un pas d'au moins 0,2.

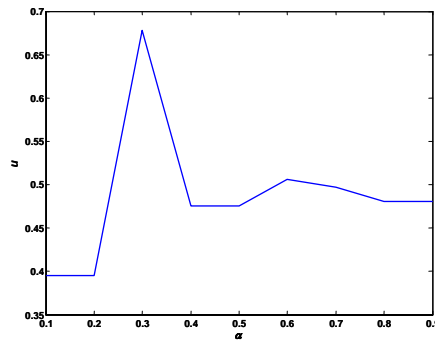


Figure 3. 10: Variations du critère  $u$  en fonction des seuils optimaux.

A partir de cette analyse, nous avons développé un nouvel algorithme, basé sur le schéma de la figure 3.11.

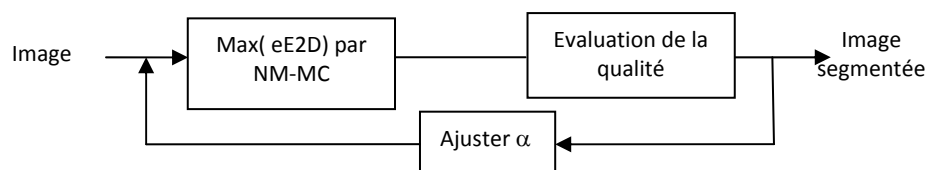


Figure 3. 11: Schéma-bloc final de la méthode de segmentation par l'eE2D et NM-MC.

Dans le cas de la segmentation en deux et trois classes des images *avion* et *maison*, les résultats de cet algorithme sont présentés sur la figure 3.12. Comme nous pouvons le constater, en comparant

par exemple les arrière-plans des figures 3.6 (c) et 3.12 (c), la segmentation obtenue par l'algorithme NM-MC est de bien meilleure qualité. D'autres résultats sont présentés en Annexe dans la section A1.

#### 3.4.4 BILAN

Dans cette section, nous avons présenté une méthode de segmentation basée sur l'entropie exponentielle à deux dimensions d'ordre  $\alpha$ . L'analyse de la fonction objectif, qui modélise le problème posé, a montré la nécessité de faire appel à une métaheuristique puissante, afin de produire des solutions de bien meilleure qualité. L'adaptation du recuit microcanonique et son hybridation avec l'algorithme du simplexe a, dans une large mesure, contribué à l'obtention de ces bons résultats. Par ailleurs, l'analyse de la sensibilité des résultats de la segmentation par rapport aux variations de l'ordre de l'eE2D a montré l'utilité d'optimiser cet ordre, afin d'avoir une segmentation optimale. Toutefois, ces résultats ne sont obtenus qu'au prix d'une augmentation de la complexité de la méthode de segmentation développée.

En conclusion, cette méthode a été appliquée avec succès pour segmenter des images microscopiques, les résultats obtenus sont présentés dans (Nakib, et al., 2007b).

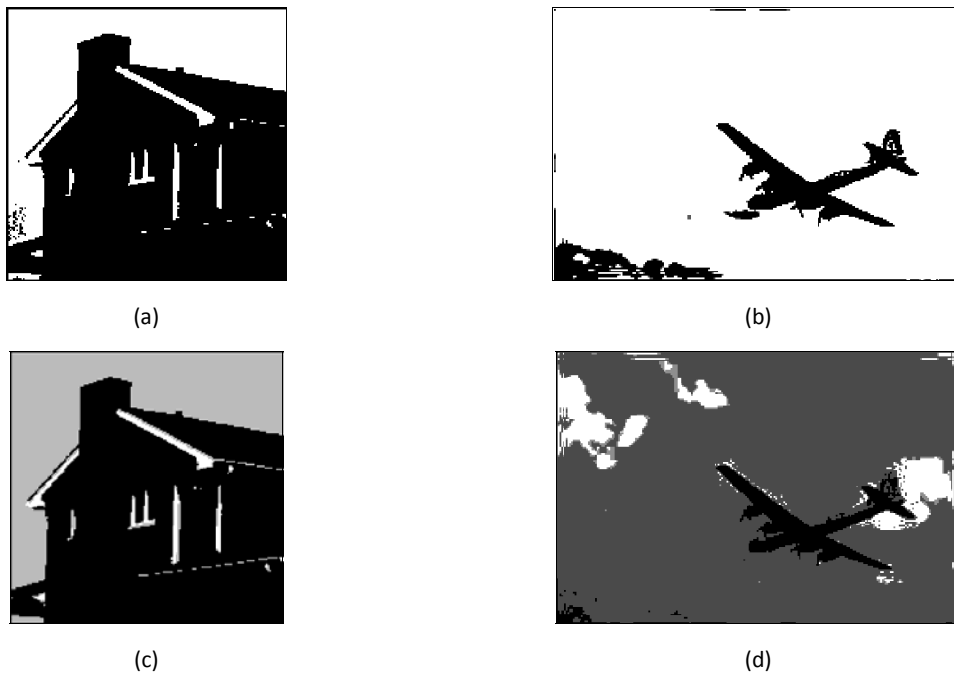


Figure 3. 12: Résultats de la segmentation produite par NM-MC avec  $\alpha$  optimal. (a) Image *maison* segmentée en 2 classes avec ( $\alpha=0,3$ ;  $t=188$ ;  $s=208$ ), (b) Image *avion* segmentée en 2 classes avec ( $\alpha=0,3$ ;  $t=89$ ;  $s=151$ ), (c) Image *maison* segmentée en 3 classes avec ( $\alpha=0,7$ ;  $t=(181; 218)$ ;  $s=(35; 181)$ ), (d) Image *avion* segmentée en 3 classes avec ( $\alpha=0,7$ ;  $t=(74; 142)$ ;  $s=(6; 75)$ ).

### 3.5 SEGMENTATION PAR eE2D ET OEP ADAPTATIF (TRIBES)

Afin de pallier au problème de la complexité algorithmique, nous utilisons, dans cette section, un algorithme d'optimisation par l'approche OEP adaptatif, pour maximiser l'entropie exponentielle à deux dimensions (Nakib, et al., 2007a).

Disposer d'une métaheuristique qui permettrait de résoudre n'importe quel problème d'optimisation, sans passer par une étape préalable de réglage, reste un vœu pieux. Malgré la



difficulté de ce challenge, quelques travaux, basés sur l'optimisation par essaim particulaire (OEP) et traitant ce problème, sont parus récemment. M. Clerc, notamment, a mis en œuvre un algorithme totalement autonome (dit aussi adaptatif), appelé TRIBES (Clerc, 2003). Cet algorithme incorpore des règles, définissant la manière selon laquelle la structure de l'essaim, et une particule donnée doivent être modifiées à chaque instant. Ces règles dépendent des informations recueillies progressivement durant le processus d'optimisation. Les paragraphes suivants sont dédiés à la description de TRIBES et à son application en segmentation d'images.

### 3.5.1 DESCRIPTION DE TRIBES

---

Dans TRIBES, l'utilisateur doit seulement définir la fonction objectif et le critère d'arrêt. Cependant, il est à signaler que TRIBES ne peut pas résoudre tous les problèmes avec exactitude. De plus, ses résultats sont probabilistes à cause de son caractère stochastique. Le but de TRIBES, d'après son auteur M. Clerc, est d'être efficace dans la plupart des cas et de permettre à ses utilisateurs de gagner du temps, en évitant l'étape de réglage de la métaheuristique.

#### ***Les tribus***

Les tribus sont des sous-ensembles de particules, dans lesquels chaque particule informe les autres de sa meilleure position atteinte. Les tailles des différentes tribus sont différentes. Cette appellation vient d'une analogie avec un groupe d'individus évoluant dans un environnement inconnu, à la recherche d'un "bon" emplacement.

#### ***Structure et relation tribales***

L'espace de recherche est exploré par toutes les tribus, simultanément, et toutes les tribus s'échangent leurs résultats pour trouver l'optimum global. L'algorithme comprend deux types de communications : intra-tribu et inter-tribu. Plus de détails sur ces types de communication sont dans (Clerc, 2003).

Afin de définir des règles de modification de la structure de l'essaim, des mesures de qualité de tribus et de particules sont définies. Ces mesures ont permis de définir les deux règles suivantes : suppression d'une particule et génération de particules. Dans le but de laisser le temps à l'information de se propager entre les particules, ces adaptations de structure ne sont pas effectuées à chaque itération.

En pratique, la fréquence des adaptations est liée au nombre total de liens d'information : si, après une adaptation structurelle, ce nombre est  $L$ , alors la prochaine aura lieu dans  $L/2$  itérations.

#### ***Evolution de l'essaim***

Au départ, l'essaim est composé seulement de trois particules. Dans le cas où aucune particule n'améliore sa situation, de nouvelles particules sont créées; leur nombre est proportionnel à la taille de la tribu génératrice, formant ainsi une nouvelle tribu.

La taille de l'essaim devient de plus en plus grande, au fil des itérations, jusqu'à ce que des zones "prometteuses" soient trouvées. Plus la taille de l'essaim augmente, plus les adaptations sont espacées dans le temps. Cette augmentation de la taille permet d'améliorer la capacité d'exploration de l'essaim. La chance de trouver une bonne solution, entre deux adaptations, successives devient ainsi plus grande. A l'inverse, dans le cas où une zone "prometteuse" est trouvée, chaque tribu va progressivement retirer sa plus mauvaise particule, ce qui peut aller jusqu'à sa disparition. Idéalement, quand la convergence est confirmée, chaque tribu est réduite à une particule.

### Stratégie de déplacement

La seconde stratégie d'adaptation de l'essaim afin de trouver les solutions optimales, consiste à sélectionner différentes stratégies de déplacement pour chaque particule, en fonction de son passé récent.

L'algorithme choisit alors de faire appel à la meilleure stratégie pour déplacer la particule vers la meilleure position possible pouvant être atteinte.

### 3.5.2 ALGORITHME DE SEGMENTATION VIA TRIBES

L'algorithme de segmentation proposé est basé sur la maximisation de l'entropie exponentielle à deux dimensions (eE2D) (section 3.4.1). La formulation du problème est donnée par l'expression (3.14). L'algorithme ne nécessite aucune initialisation particulière, si ce n'est le nombre initial de particules que nous avons fixé à 3. Le nombre maximum d'évaluations de la fonction objectif est utilisé comme critère d'arrêt, que nous avons fixé à  $2 \times 10^{N+1}$  évaluations de la fonction objectif, où  $N$  est le nombre de seuils.

En ce qui concerne la gestion des contraintes, si la particule prend une valeur en dehors de l'intervalle, elle est ramenée à la frontière la plus proche, et une vitesse nulle lui est attribuée. Le schéma-bloc de l'algorithme de segmentation est présenté sur la figure 3.13.

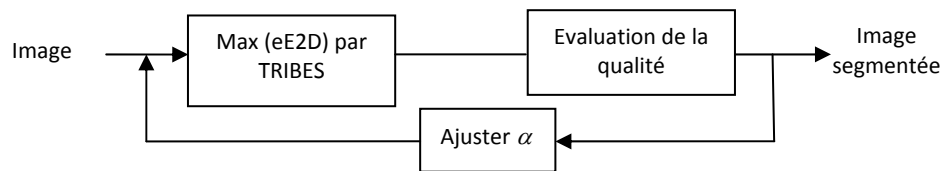


Figure 3. 13: Schéma-bloc de la méthode de segmentation avec TRIBES et eE2D.

### 3.5.3 RESULTATS

Les résultats de segmentation avec TRIBES sont présentés sur la figure 3.14 et proviennent de la moyenne de 25 exécutions. Ceux de la segmentation de l'image *maison* en deux classes (figure 3.14 (a)) sont exactement les mêmes qu'avec NM-MC (figure 3.12 (a)), mais avec une plus grande vitesse de convergence.

En ce qui concerne la reproductibilité de la solution, l'écart type est de l'ordre de  $10^{-3}$ . Cependant, le résultat obtenu sur l'image *avion* (figure 3.14 (b)) et celui de la figure 3.12(b) sont "équivalents" malgré la différence entre les seuils de segmentation : 89 avec NM-MC et 84 avec TRIBES, ceci est dû à la nature de l'image. Dans le cas de plusieurs classes, la segmentation des images *maison* et *avion* en 3 classes est présentée sur les figures 3.14(c) et 3.14 (d), respectivement. D'un point de vue visuel, les résultats obtenus sont de bonne qualité, même s'ils sont relativement moins bons que ceux de la méthode NM-MC. En terme de vitesse de convergence, TRIBES est en revanche nettement meilleur : les résultats présentés sont obtenus à l'issue de 1000 évaluations de la fonction objectif, contre 4000 évaluations avec NM-MC.

### 3.5.4 BILAN

Dans cette section, nous avons présenté une méthode de segmentation basée sur la maximisation de l'eE2D avec un algorithme d'optimisation par OEP adaptatif que nous avons ajusté au cas combinatoire.

La rapidité et l'efficacité de cet algorithme nous ont permis de l'appliquer pour segmenter des images IRM. Plus de détails sur ce travail sont dans (Nakib, et al., 2007a).

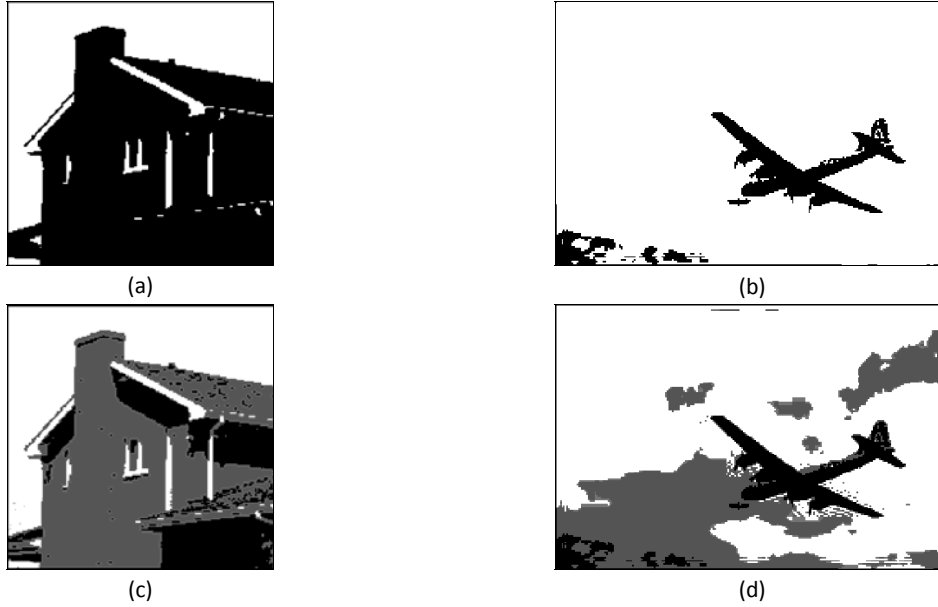


Figure 3. 14: Résultats de la segmentation avec TRIBES, avec  $\alpha$  optimal. (a) Image *maison* segmentée en 2 classes avec ( $\alpha=0,3$ ;  $t=195$ ;  $s=208$ ), (b) Image *avion* segmentée en 2 classes avec ( $\alpha=0,8$ ;  $t=84$ ;  $s=151$ ), (c) Image *maison* segmentée en 3 classes avec ( $\alpha=0,5$ ;  $t=(91; 180)$ ;  $s=(119;204)$ ), (d) Image *avion* segmentée en 3 classes avec ( $\alpha=0,3$ ;  $t=(84; 116)$ ;  $s=(120; 151)$ ).

### 3.6 SEGMENTATION PAR ENTROPIE SURVIVANTE EXPONENTIELLE A DEUX DIMENSIONS ET OEP

Dans cette partie, nous présentons une nouvelle méthode pour la segmentation des images basée sur une nouvelle mesure de l'information, appelée "entropie survivante exponentielle" ("*Survival Exponential Entropy*", SEE) (Nakib, et al., 2007c). Afin d'inclure l'information sur la distribution spatiale des niveaux de gris, nous avons défini un histogramme à deux dimensions, dont la procédure de construction est décrite dans le premier chapitre (section 1.3.2.2.1).

Nous avons aussi généralisé la SEE au cas bi-dimensionnel et l'avons étendue au cas multiniveau. Le problème posé présente une complexité algorithmique qui croît de manière exponentielle avec le nombre de classes. Ce qui nous a motivés à mettre en œuvre la métaheuristique rapide d'optimisation par essaim particulaire (OEP), afin de chercher la solution optimale.

#### 3.6.1 L'ENTROPIE SURVIVANTE EXPONENTIELLE A DEUX DIMENSIONS

Dans cette partie, nous présentons la nouvelle mesure d'information des variables aléatoires, appelée entropie survivante exponentielle ("*Survival Exponential Entropy*", SEE) (Zografos, et al., 2005) et son extension au cas bi-dimensionnel.

La fonction survivante multivariable  $\bar{F}_{|X|}(x)$  d'un vecteur aléatoire  $|X|$  est définie par:

$$\bar{F}_{|X|}(x) = P(|X| > x) = P(|X_1| > x_1, \dots, |X_m| > x_m) \quad (3. 18)$$

pour  $x \in R_+^m$ . Pour une distribution discrète, la fonction de distribution cumulée peut être définie par:

$$\bar{F}(x) = 1 - \sum_{i=0}^x p(i) \quad (3.19)$$

pour le vecteur aléatoire  $X$  dans  $R^m$ , l'entropie survivante d'ordre  $\alpha$  est définie par:

$$M_\alpha(X) = \left( \int_{R^m} \bar{F}_{|X|}^\alpha(x) dx \right)^{1/1-\alpha} \quad (3.20)$$

Pour  $\alpha \geq 0$  et  $\alpha \neq 1$ , où  $m$  représente la dimension de  $X$ . La notation  $|X| > x$  est équivalente à  $|X_i| > x_i$  pour  $x_i \geq 0, i = 1, \dots, m$ .

La SEE utilise la fonction de densité avec la distribution cumulée, celle-ci étant plus régulière que la fonction de densité. La SEE a plusieurs avantages par rapport à l'entropie classique de Shannon et à l'entropie différentielle (extension de l'entropie de Shannon au cas continu): elle est compatible dans les deux cas continu et discret; elle est toujours positive et simple à calculer à partir des données. Alors que l'entropie de Shannon est basée sur la densité des variables aléatoires, qui parfois n'existe pas et, quand elle existe, doit être estimée. Pour la segmentation d'images, nous utilisons une propriété bien connue des mesures de l'entropie: si les distributions marginales d'une distribution jointe sont indépendantes, alors l'entropie totale de la distribution jointe est égale à la somme des entropies des distributions marginales.

Nous définissons l'entropie survivante exponentielle à deux dimensions discrète (Nakib, et al., 2007c) ("*Two-dimensional Survival Exponential Entropy*", 2DSEE) par la relation:

$$M_\alpha(X) = \left( \sum_i \sum_j \left( \bar{F}_{|X|}(i, j) \right)^\alpha \right)^{1/1-\alpha} \quad (3.21)$$

où  $\alpha \in \mathfrak{R}^+$  et  $\alpha \neq 1$ .

Ainsi, l'entropie survivante exponentielle à deux dimensions discrète, associée aux différentes distributions de classes d'une image, est calculée comme suit:

- la 2DSEE de la classe  $m-1$ :

$$M_\alpha^{(m-1)} [a_{n-1}, a_n] = \left( \sum_{i=s_{n-1}}^{s_n-1} \sum_{j=t_{n-1}}^{t_n-1} \left( \bar{F}(i, j) \right)^\alpha \right)^{1/(1-\alpha)} \quad (3.22)$$

- la 2DSEE de la classe  $m$ :

$$M_\alpha^{(m)} [a_n, a_{n+1}] = \left( \sum_{i=s_n}^{s_{n+1}-1} \sum_{j=t_n}^{t_{n+1}-1} \left( \bar{F}(i, j) \right)^\alpha \right)^{1/(1-\alpha)} \quad (3.23)$$

La 2DSEE totale est alors :

$$M_\alpha^{(T)} [a_0, \dots, a_N] = \sum_{i=0}^{N-1} M_\alpha^{(i+1)} [a_i, a_{i+1}] \quad (3.24)$$

Comme dans le cas de la eE2D (section 3.4.1), nous ajoutons les deux vecteurs  $(s_0, t_0) = (1, 1)$  et  $(s_{N+1}, t_{N+1}) = (255, 255)$ , où  $t_0 < t_1 < t_2 < \dots < t_{N+1}$  et  $s_0 < s_1 < s_2 < \dots < s_{N+1}$ .

En appliquant le principe du maximum d'entropie, le vecteur optimal  $(a_{1, \dots, N-1}^*) \equiv ((s_1^*, t_1^*), \dots, (s_N^*, t_N^*))$  doit vérifier:

$$M_{\alpha}^{(T)}(a_{1,\dots,N}^*) = \text{Max}\{M_{\alpha}^{(T)}(a_{1,\dots,N})\} \quad (3.25)$$

$$\text{s. c. : } t_0 < t_1 < t_2 < \dots < t_{N+1} \text{ et } s_0 < s_1 < s_2 < \dots < s_{N+1}$$

Dans le cas d'un seul seuil ( $N=2$ ), la complexité réside dans le calcul du vecteur optimal  $(s^*, t^*)$ , elle vaut  $O(L^4)$ . Cependant, dans le cas multiniveau, la recherche de l'optimum nécessite  $O(L^{2n+2})$ , où  $n$  est le nombre de seuils, ce qui nécessite un temps de calcul relativement important. Dans cette section, nous utilisons l'optimisation par OEP pour résoudre efficacement le problème décrit par (3.24).

### 3.6.2 METHODE DE SEGMENTATION

Cette méthode est basée sur l'optimisation de la 2DSEE totale (3.24). La procédure Algorithme 3.3 montre les différentes étapes de l'algorithme. La méthode exploite l'optimisation par essaim particulière pour résoudre le problème de segmentation défini par l'expression (3.24). L'algorithme ne nécessite aucune initialisation particulière. La taille de l'essaim utilisée pour notre approche est de  $3N+1$ , pour un problème de segmentation de dimension  $N+1$ .

En ce qui concerne les critères d'arrêt, nous avons utilisé deux conditions :

- le nombre maximum d'itérations :  $Nbitmax = \frac{1}{N-1}10^N$ , où  $N$  est le nombre de classes;
- le nombre maximum d'itérations n'induisant pas d'amélioration du dernier optimum enregistré : 200.

Algorithme 3.3: Maximisation de la 2DSEE par PSO.  $g$  est la meilleure position que l'essaim a atteint,  $L$  la meilleure position de la particule  $i$ ,  $P_i$  la position de la particule  $i$ .

- 1. Initialisation** de la population de particules avec des positions  $P$  et vitesses  $v$  aléatoires.
- 2. Evaluer** la fonction objectif (3.24) pour chaque particule et calculer  $g$ .
- 3. Pour** chaque individu  $i$ ,  $L_i$  est initialisée à  $P_i$ .
- 4. Répéter** jusqu'au critère d'arrêt
  - 4.1. Mettre à jour les vitesses et les positions des particules ( $v_i$  et  $P_i$ ).
  - 4.2. Evaluer la fonction objectif pour chaque individu définie par (3.24).
  - 4.3. Calculer les nouveaux  $L_i$  et  $g$ .
- 5. Afficher** la meilleure solution.

### 3.6.3 RESULTATS ET DISCUSSION

La première discussion porte sur le choix de l'ordre ( $\alpha$ ) de la 2DSEE. Celle-ci est illustrée sur des images synthétiques. La seconde discussion concerne la sélection des seuils optimaux au sens de la maximisation de la 2DSEE, ainsi que l'évaluation de la qualité de la segmentation sur quelques IRM. Nous ne présentons ici que des résultats de segmentation dans le cas de quatre et cinq classes.

Nous avons utilisé une image synthétique que nous avons bruitée avec différents degrés de bruit, mesurés par le rapport signal sur bruit (RSB) (figure 3.15).

Le rapport signal sur bruit (RSB) d'une image  $I$  est défini par :

$$RSB(\text{dB}) = 10 \cdot \log_{10} \frac{V_I}{V_B} \quad (3.26)$$

où  $V_I$  est la variance de l'image et  $V_B$  est la variance du bruit.

Pour mesurer les performances de la segmentation, nous avons utilisé le critère de l'erreur de classification ( $ME$ ).  $ME$  est défini en terme de corrélation des images avec l'observation humaine. Il

correspond au rapport du nombre de pixels de l'arrière plan faussement classés au premier plan, et vice versa. L'expression analytique de  $ME$  est donnée par :

$$ME = \left( 1 - \left( \frac{|B_O \cap B_T| + |F_O \cap F_T|}{(|B_O| + |F_O|)} \right) \right) \cdot 100 \quad (3.27)$$

où  $B_O$  est l'arrière plan,  $F_O$  est le premier plan de l'image originale et  $B_T$  et  $F_T$  sont, respectivement, l'arrière plan et le premier plan de l'image test segmentée.

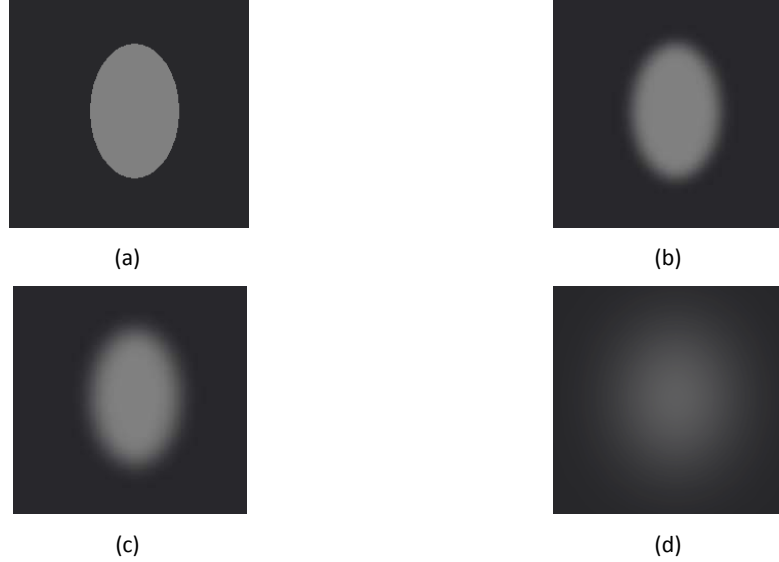


Figure 3. 15: Images test synthétiques. (a) Image originale, (b) Image bruitée (RSB = 17,77 dB), (c) Image bruitée (RSB=15,44 dB), (d) Image bruitée (RSB = 5,84 dB).

Dans le cas idéal  $ME$  est égale à 0% et dans le plus mauvais cas,  $ME$  est égale à 100%. La figure 3.16 illustre les variations de  $ME$  (%) en fonction de l'ordre de la 2DSEE dans l'intervalle 0 à 0,9.

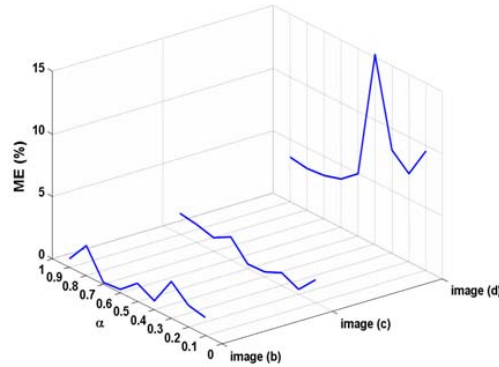


Figure 3. 16: Variations de la qualité de la segmentation en fonction de l'ordre de la 2DSEE.

Comme nous pouvons le constater, la qualité de la segmentation dépend de la valeur de  $\alpha$ . Les valeurs optimales obtenues dans le cas des images des figures 3.15 (b) à (d), sont : 0,12% ( $\alpha=0,7$ ), 0,33% ( $\alpha=0,5$ ) et 3,30% ( $\alpha=0,8$ ), respectivement. L'analyse de ces résultats montre qu'une segmentation optimale peut être obtenue avec un ordre de la 2DSEE dans l'intervalle  $[0,4 \ 0,7]$ , et un ordre de grandeur de variation de  $\alpha$  égale à 0,1.

Afin de trouver l'ordre optimal du processus de segmentation, nous allons réutiliser la procédure décrite dans la section 3.4.2.2, basée sur l'utilisation du critère d'homogénéité défini par l'expression (3.17). Le schéma complet du système de segmentation est donné sur la figure 3.17.

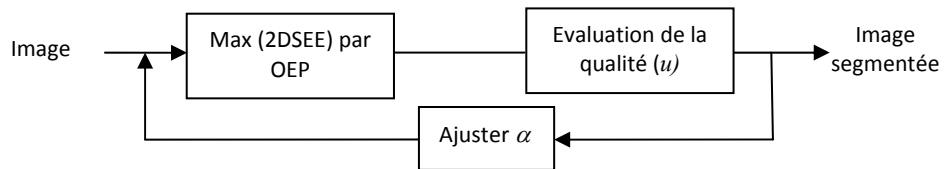


Figure 3. 17: Schéma-bloc de la méthode de segmentation avec 2DSEE et OEP.

Les résultats de la segmentation de deux images (test, *maison* et *avion*) sont présentés sur la figure 3.18. Ces résultats représentent une moyenne de 20 exécutions consécutives de l'algorithme. Les figures 3.18. (b) et (d) présentent les résultats de la segmentation en deux et trois classes de l'image *avion*, respectivement. Le résultat de la figure 3.18 (b) coïncide avec l'optimum global de la fonction  $M_{(0.7)}^T$  trouvé via une exploration exhaustive de l'espace des solutions. Avec notre méthode, ce résultat est obtenu en moins de 500 itérations, ce qui équivaut à un facteur de gain de 130, par rapport à la recherche exhaustive. La qualité de la segmentation en deux classes de l'image *maison* (figure 3.18 (a)) est inférieure à celle trouvée par la méthode NM-MC (figure 3.14(a)). En revanche, le résultat de la segmentation de cette image en 3 classes est très bon (figure 3.18(c)).

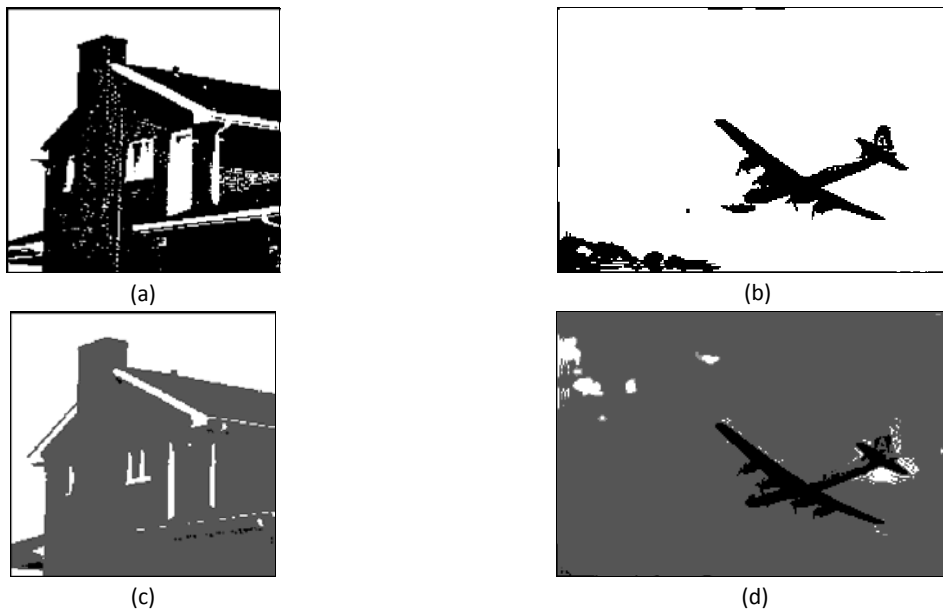


Figure 3. 18: Résultats de la segmentation avec 2DSEE et  $\alpha$  optimal. (a) Image *maison* segmentée en 2 classes avec ( $\alpha=0,5$ ;  $t=172$ ;  $s=142$ ), (b) Image *avion* segmentée en 2 classes avec ( $\alpha=0,7$ ;  $t=89$ ;  $s=129$ ), (c) Image *maison* segmentée en 3 classes avec ( $\alpha=0,8$ ;  $t=(37; 187)$ ,  $s=(48; 205)$ ), (d) Image *avion* segmentée en 3 classes avec ( $\alpha=0,7$ ;  $t=(61; 147)$ ;  $s=(34; 175)$ ).

Pour vérifier si les critères d'arrêt permettent à l'algorithme de converger vers la solution optimale, nous présenterons sur la figure 3.19 les variations de la meilleure solution de l'essai en fonction du nombre d'itérations. Comme nous pouvons le remarquer, l'OEP met moins de 100 itérations pour trouver l'optimum de la fonction objectif, dans le cas d'une segmentation en deux classes, et moins de 500 itérations dans le cas d'une segmentation en trois classes. Nos critères d'arrêt prévoyaient 100 itérations, dans le premier cas, et 500 dans le deuxième cas, ce qui est suffisant pour la convergence de l'algorithme.

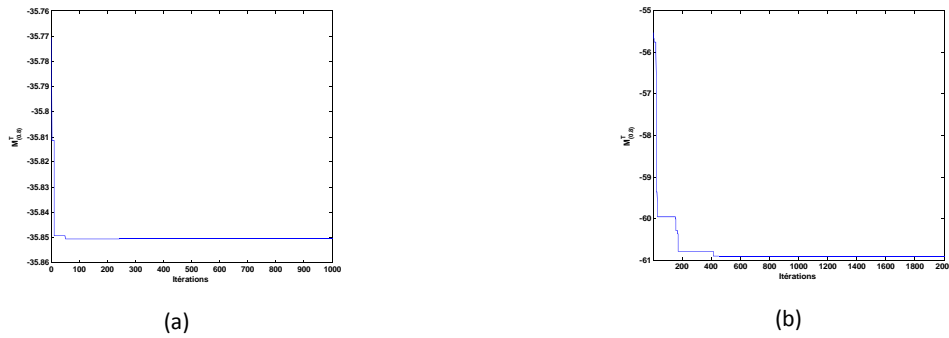


Figure 3. 19: Evolution de l'optimum dans le cas de la segmentation de l'image *maison*. (a) Segmentation en 2 classes, (b) Segmentation en 3 classes.

### 3.6.4 BILAN

Dans cette section, nous avons tenté de résoudre deux problèmes distincts: réduire le nombre de minima locaux et trouver un algorithme rapide de segmentation. L'utilisation de la 2DSEE a permis de résoudre le problème des minima locaux, comme nous pouvons le constater sur la figure 3.20, où sont tracées les variations de la 2DSEE en fonction de  $s$  et  $t$ , dans le cas d'une segmentation en 2 classes, pour  $\alpha=0,5$  et  $0,8$  de l'image test *avion*. L'introduction de l'OEP, que nous avons adaptée au problème combinatoire, a permis d'avoir, en un nombre d'itérations relativement faible, des solutions acceptables d'un point de vue visuel.

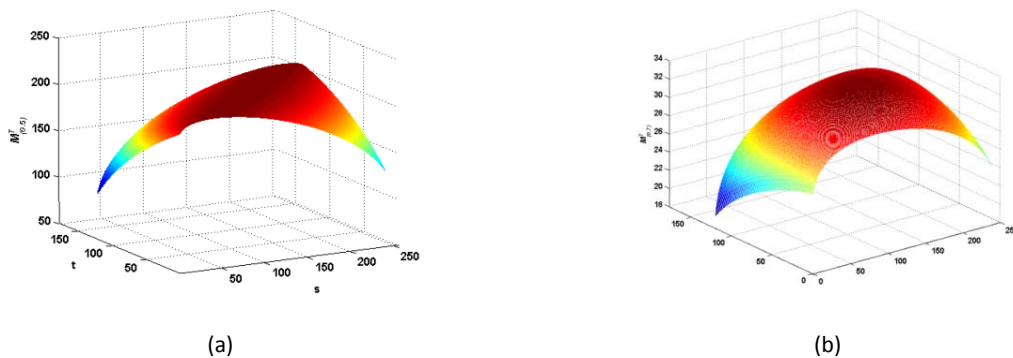


Figure 3. 20: Variations de la 2DSEE en fonction des variations des seuils  $s$  et  $t$ , exemple de l'image *avion*. (a)  $\alpha=0,5$ , (b)  $\alpha=0,8$ .

## 3.7 SEGMENTATION PAR VARIANCE INTRACLASSE BIAISEE ET COLONIES DE FOURMIS

Dans cette section, nous proposons d'améliorer la méthode de segmentation d'image par maximisation de la variance intraclasse. Pour ce faire, nous faisons appel à la métaheuristique des colonies de fourmis (ACO). Celle-ci nous permet de nous rapprocher au mieux de la solution optimale (pseudo-solution).

Dans cette étude, nous considérons le problème de la méthode d'Otsu dont l'efficacité n'est prouvée que lorsqu'un seul type d'objet à segmenter est présent dans l'image. Afin de pallier à ce problème, Cheriet *et al.* (Cheriet, et al., 1998) ont proposé d'appliquer la méthode d'Otsu de façon récursive, de manière à segmenter un seul type d'objet à la fois dans l'image. A la fin, la segmentation complète se fait par regroupement de tous les objets segmentés. Parmi les problèmes de la méthode d'Otsu, il y a



celui de l'unicité du seuil qui maximise le critère (1.3). Dans cette section, nous utilisons le critère de la variance intraclasse auquel nous apportons des modifications (Nakib, et al., 2005) . Ces dernières consistent en trois paramètres permettant de mieux caractériser une segmentation de bonne qualité. Ensuite, les seuils de segmentation maximisant le critère sont recherchés via l'algorithme ACO.

### 3.7.1 ALGORITHME DE SEGMENTATION

#### 3.7.1.1 LE CRITERE DE VARIANCE INTRACLASSE BIAISEE

Les modifications consistent en l'ajout de paramètres qui caractérisent les propriétés d'une segmentation parfaite. L'expression initiale du critère de la variance intraclasse (méthode d'Otsu) est présentée dans la section 1.3.1. La nouvelle expression du critère, pour une image  $I$ , s'écrit :

$$MVar(t_1^*, \dots, t_{N-1}^*) = \alpha \cdot \sum_{j=1}^N \left( \frac{\sigma_{int}^2(j)}{\beta_j} + \gamma_j \right) \quad (3.28)$$

où :  $\sigma_{int}^2$  est définie dans (1.4).  $N$  est le nombre de classes dans l'image à segmenter, il est supposé connu *a priori*.

$\alpha$  est égal à  $\frac{\sqrt{NR}}{(10000 \times M)}$ , où  $M$  est le nombre total de pixels dans l'image et  $NR$  le nombre de régions, c'est-à-dire le nombre de composantes connexes, dans l'image segmentée. Ce terme est utilisé pour normaliser et pénaliser les images sur-segmentées.

Le terme  $\beta_j = \frac{1}{(1 + \log N_j)}$  est choisi de telle sorte que le terme  $\frac{\sigma_{int}^2(j)}{\beta_j}$  soit faible pour les grandes classes.  $N_j$  est le nombre de pixels dans la classe  $j$ .

Le terme  $\gamma_j = (C(N_j)/N_j)^2$  est élevé lorsque l'image segmentée a beaucoup de régions de même taille, surtout si elles sont petites.  $C(N_j)$  est le nombre de régions dont le cardinal est égal à  $N_j$ . Pour les grandes régions  $C(N_j)$  est, dans la plupart des cas, égal à 1, alors que, pour les régions de petite taille, il devient supérieur à 1. Dans tous les cas, le dénominateur  $N_j$  force  $\gamma_j$  à être proche de zéro pour les grandes régions. Ces paramètres sont inspirés du critère de Borsotti, utilisé pour l'évaluation de la qualité de la segmentation des images en couleur (Borsotti, et al., 1998).

Le problème de la segmentation d'une image revient à trouver les seuils optimaux maximisant le critère défini par l'expression (3.28) :

$$MVar(t_1^*, \dots, t_{N-1}^*) = \text{Max} \left\{ MVar(t_1^*, \dots, t_{N-1}^*) \right\} \quad (3.29)$$

s. c. :  $1 < t_1 < t_2 < \dots < 255$

#### 3.7.1.2 L'ALGORITHME ACO

La métaheuristique ACO a été développée par Dorigo (Dorigo, et al., 1996). A la base, l'algorithme est destiné à résoudre le problème combinatoire du voyageur du commerce (Dréo, et al., 2003) Par conséquent, nous n'avons pas eu de modifications majeures à lui apporter, mis à part les intervalles de variation des variables, afin de respecter les contraintes : si une fourmi sort de son intervalle, sa densité de phéromone est remise à zéro.

La procédure Algorithme 3.4 résume les différentes étapes de l'algorithme de segmentation (*MVar-ACO*), basé sur la maximisation du critère défini par l'expression (3.28).

A l'instar de l'algorithme OEP, nous avons utilisé deux critères d'arrêt :

- le nombre maximum d'itérations :  $N_{bitmax} = 100^{(N-1)}$ , où  $N$  est le nombre de classes;
- le nombre maximum d'itérations n'induisant pas d'amélioration du dernier optimum enregistré : 200.

Le réglage des paramètres de l'algorithme a été fait de manière empirique. Les valeurs utilisées sont résumées dans le tableau 3.3.

Algorithme 3.1 : Algorithme de segmentation par *MVar-ACO*.  $m$  est le nombre de fourmis.

1. Tirer  $m$  fourmis aléatoirement.
2. Tant que le critère d'arrêt n'est pas satisfait
  - 2.1. Pour chaque fourmi  $k=1, \dots, m$ 
    - 2.1.1. Evaluer la fonction objectif donnée par la formule (3.27).
    - 2.1.2. Attribuer une probabilité de transition selon l'expression (2.15).
    - 2.1.3. Evaporer les pistes selon l'expression (2.17).
  - Fin pour
3. Afficher le meilleur état rencontré au cours de la recherche.

Tableau 3.1: Valeurs des paramètres de ACO.

Paramètre	Valeur
Nombre de fourmis $m$	$50 \times N$
Coefficient d'évaporation	0,1
$\alpha$	1
$\beta$	5

### 3.7.2. RESULTATS

Les résultats de la segmentation par l'algorithme *MVar-ACO* sont présentés sur la figure 3.21. Les segmentations des images *maison* et *avion* en deux classes sont présentées respectivement sur les figures 3.21 (a) et (b).

Par rapport aux autres méthodes, les résultats obtenus sont de moindre qualité. En revanche, dans le cas de la segmentation en trois classes, l'algorithme *MVar-ACO* produit de bons résultats, notamment sur la figure 3.21 (c), où toutes les classes sont bien séparées et homogènes. Ces améliorations sont obtenues grâce aux paramètres que nous avons ajoutés à la variance intraclasse. Dans le cas de la segmentation en trois classes, nous pouvons donc estimer que nous avons réussi à trouver les seuils optimaux qui produisent une segmentation acceptable, en maximisant le critère *MVar* à l'aide de l'algorithme *ACO*.

### 3.7.3 BILAN

Dans cette section, nous avons présenté un nouveau critère de segmentation basé sur la variance intraclasse biaisée.

Pour résoudre le problème de la complexité des calculs dans le cas de seuillage multiniveau, nous avons adapté l'algorithme *ACO* à ce problème. Ce dernier s'applique bien à la segmentation, mais son réglage est assez compliqué et son temps de convergence est assez lent.

En comparaison des autres méthodes de segmentation développées, les performances de l'algorithme *MVar-ACO* seront réexaminées dans le prochain chapitre. Enfin, nous avons appliqué ce critère (*MVar*) pour segmenter des images d'angiographie rétinienne, les résultats sont présentés dans (Nakib, et al., 2005).

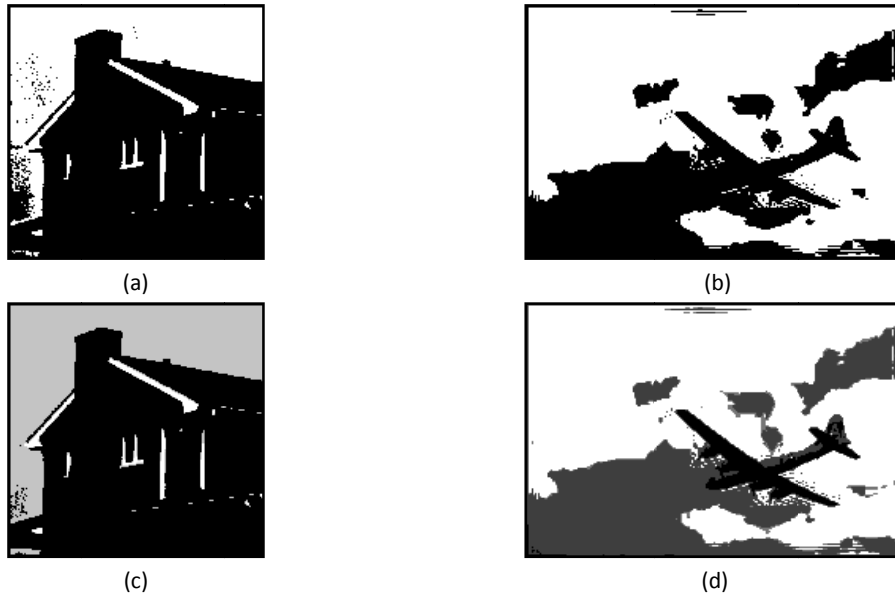


Figure 3.21: Résultats de la segmentation avec *MVar-ACO* via  $\alpha$  optimal. (a) Image *maison* segmentée en 2 classes avec ( $t=197$ ), (b) Image *avion* segmentée en 2 classes avec ( $t=117$ ), (c) Image *maison* segmentée en 3 classes avec  $t=(197 ; 210)$ , (d) Image *avion* segmentée en 3 classes avec  $t=(62 ; 118)$ .

### 3.8 COMPARAISON

Nous allons comparer les méthodes développées, du point de vue de la qualité de la segmentation obtenue. Pour cela, nous utilisons des images synthétiques dont le résultat optimal de segmentation est connu *a priori* (image "vérité terrain" connue). Le critère d'évaluation utilisé est défini par l'expression (3.26). Parmi les méthodes que nous avons développées, les deux méthodes eE2D et 2DSEE vont être comparées à six autres méthodes : la méthode classique d'Otsu, la méthode de Kapur, la méthode EM, la méthode VE ("*valley emphasis*"), la méthode basée sur l'entropie 2D de Tsallis (TE) et la méthode *K-means* basée sur la distance euclidienne. Toutes ces méthodes ont déjà été présentées dans le premier chapitre : pour les méthodes d'Otsu et VE, se reporter à la section 1.3.1; pour les méthodes de Kapur et TE, se reporter à la section 1.3.2; pour la méthode EM, se reporter à la section 1.4.3 et pour la méthode *K-means*, se reporter à la section 1.6. Les performances des autres méthodes, développées dans les sections 3.2 et 3.7, seront traitées dans le chapitre suivant.

Pour notre expérience, l'image synthétique de la figure 3.22 (A) est utilisée. Dans un premier temps, cette image est dégradée avec un bruit multiplicatif de plus en plus fort (figure 3.15. (B) à (D)). Ensuite, un bruit blanc gaussien (BBG) additif est ajouté aux images déjà bruitées. Nous présentons ces images sur la figure 3.22.

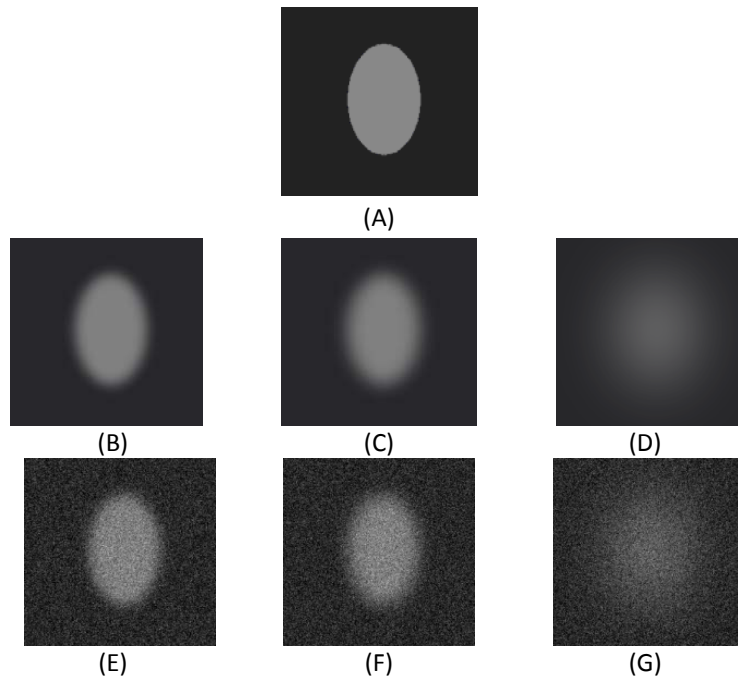


Figure 3. 22: Image originale et images bruitées synthétiques: (B) Image bruitée (RSB = 17,77 dB), (C) Image bruitée (RSB=15,44 dB), (D) Image bruitée (RSB = 5,84 dB), (E) Image bruitée (RSB = 15,91 dB), (F) Image bruitée (RSB = 13,91 dB), (G) Image bruitée (RSB = 5,44 dB).

Les résultats de segmentation obtenus via les méthodes comparées sont présentés sur la figure 3.23. Pour chaque méthode, les résultats sont présentés sur une colonne de la figure. La quantification de ces résultats en fonction des valeurs du critère  $ME$  est résumée dans le tableau 3.4. En examinant le tableau 3.4, nous pouvons constater que les deux méthodes de segmentation 2DSEE et eE2D permettent d'avoir de faibles valeurs du critère  $ME$ , c'est-à-dire une bonne qualité de segmentation, dans le cas d'un bruit multiplicatif (figures 3.22 B à D), avec un léger avantage pour la méthode 2DSEE. Lorsque le BBG est ajouté, cet avantage devient plus important.

Par rapport aux autres méthodes, la 2DSEE est la plus performante, sauf dans le cas des images E et F, où la méthode EM est meilleure. Ces résultats s'expliquent par le fait, que dans le cas des images E et F, la modélisation de l'histogramme avec deux gaussiennes était encore possible, malgré la densité du bruit, ce qui a permis d'avoir une bonne détection. Malheureusement, dans le dernier cas (image G), le bruit est tellement important que l'histogramme est fortement dégradé. En d'autres termes, il devient unimodal, ce qui rend la méthode EM peu performante par rapport à la 2DSEE.

D'un point de vue visuel, les résultats présentés sur la figure 3.23 montrent la supériorité de la méthode 2DSEE par rapport aux autres. En ce qui concerne les six autres méthodes concurrentes, la méthode *K-means* permet d'avoir un résultat optimal, dans le cas des images faiblement bruitées. Malheureusement, l'augmentation du bruit a conduit à une baisse considérable de ses performances. Les résultats de la méthode de Kapur ne sont pas loin des résultats des autres méthodes, dans le cas des images E à G, mais ne sont jamais les meilleurs. Quant à la méthode d'Otsu, ses performances sont médiocres dans le cas des images fortement bruitées. Les méthodes VE et TE permettent d'avoir des résultats comparables à ceux des autres méthodes, sans atteindre le résultat optimal.

Cette étude comparative nous a permis d'avoir une idée précise sur les performances des deux critères proposés (eE2D et 2DSEE). La méthode que nous avons jugée la plus performante, grâce à cette étude, est la 2DSEE. Dans la section qui suit, nous allons présenter l'application de cette méthode à la segmentation des images à résonance magnétique (IRM).

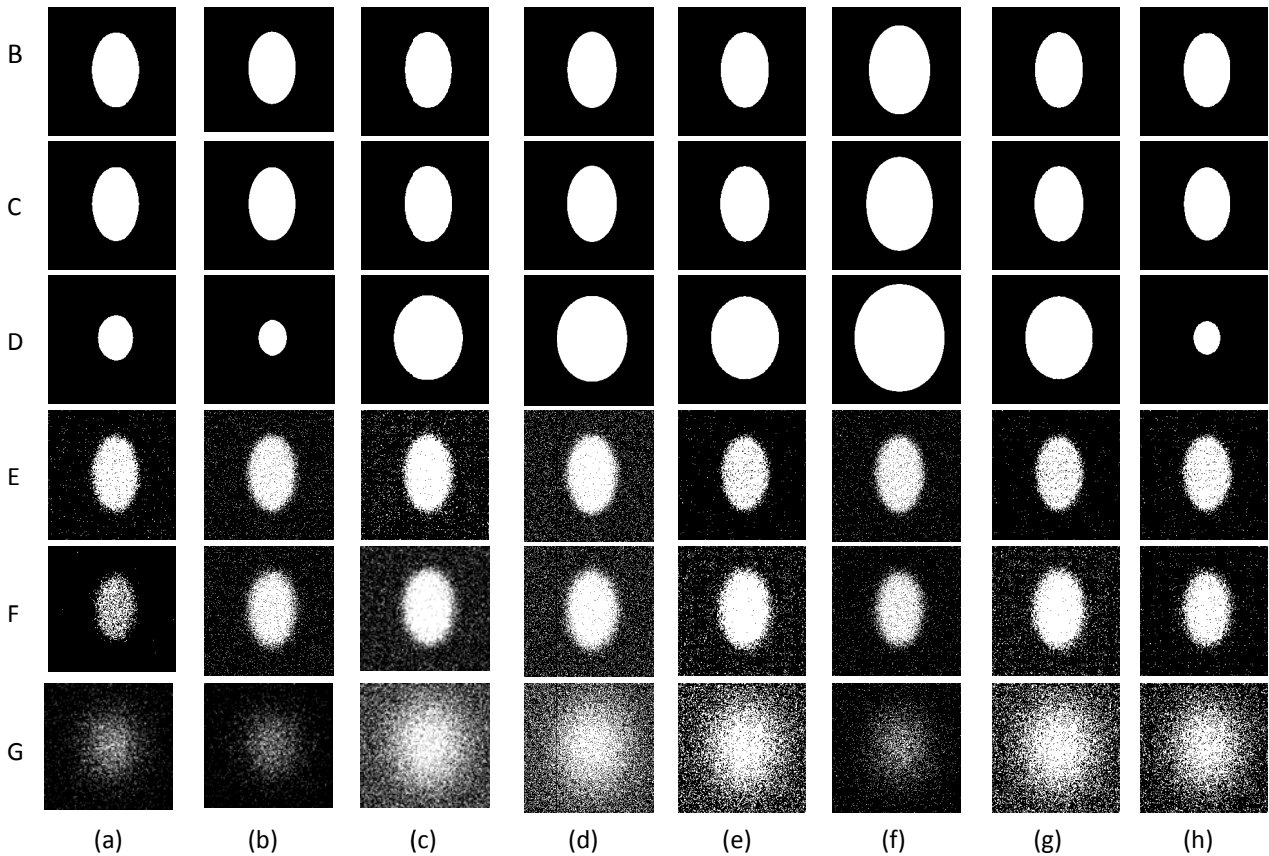


Figure 3. 23: Performances des différentes méthodes. (colonne a) Images segmentées par eE2D, (colonne b) Images segmentées par 2DSEE, (colonne c) Images segmentées par *K-means*, (colonne d) Images segmentées par Otsu, (colonne e) Images segmentées par Kapur, (colonne f) Images segmentées par la méthode *EM*, (colonne g) Images segmentées par *VE*, (colonne h) Images segmentées par la méthode *TE*.

Tableau 3.4: Evaluation des performances des différentes méthodes de segmentation. Les faibles valeurs de *ME* sont en gras.

Images test de la Figure 3.22	Méthodes de segmentation									
	Otsu	Kapur	<i>K-means</i>	EM	VE	TE	eE2D		2DSEE	
	<i>ME</i> (%)	<i>ME</i> (%)	<i>ME</i> (%)	<i>ME</i> (%)	<i>ME</i> (%)	<i>ME</i> (%)	<i>ME</i> (%)	$\alpha$	<i>ME</i> (%)	$\alpha$
Image B	0,45	5,61	<b>0,17</b>	8,68	0,34	0,64	<b>0,23</b>	0,4	<b>0,17</b>	0,7
Image C	0,88	4,50	0,61	12,50	0,63	1,12	<b>0,56</b>	0,4	<b>0,37</b>	0,5
Image D	12,22	4,97	10,86	28,87	11,59	12,90	<b>3,57</b>	0,6	<b>3,46</b>	0,8
Image E	11,92	5,77	11,84	<b>5,74</b>	5,90	5,92	6,78	0,5	6,31	0,5
Image F	13,82	6,74	14,92	<b>6,60</b>	13,22	6,68	9,84	0,6	7,45	0,6
Image G	36,63	16,66	36,45	16,63	35,56	29,98	16,09	0,5	<b>15,72</b>	0,5

### 3.9 APPLICATION DE LA 2DSEE ET DE L'OEP AUX IMAGES IRM

Le besoin croissant en analyse d'images IRM a permis à la segmentation des images IRM de s'imposer comme un important axe de recherche. Elle permet, par exemple, de faciliter l'évaluation des évolutions des espaces ventriculaires, par une approche multiniveau. Nous nous sommes intéressés au problème de la détection des espaces ventriculaires sur des images IRM du cerveau. La difficulté de ce problème réside dans la présence d'un liquide blanc entourant les ventricules,

notamment chez les sujets âgés, même lorsqu'ils ne sont pas atteints d'une atrophie; ce qui rend difficile la détection des lésions péri-ventriculaires, lors d'une image IRM.

Les méthodes paramétriques utilisées pour la segmentation des images IRM du cerveau sont résumées dans (Qiao, et al., 2007; Drapaca, et al., 2005; Warfield, et al., 2000; Murgasova, et al., 2006; Song, et al., 2006; Kamber, et al., 2000; Cocosco, et al., 2003). La classe des méthodes paramétriques la plus répandue est basée sur l'algorithme espérance-maximisation. Cette classe de méthodes repose sur l'hypothèse que les densités de probabilité des différents tissus du cerveau sont une combinaison de gaussiennes. Malheureusement, dans la plupart des situations, ces distributions sont loin d'être gaussiennes. Afin de résoudre ce problème, de nombreux auteurs régularisent l'erreur de classification des pixels par des contraintes spatiales sur la segmentation, traduisant ainsi une information *a priori* à partir d'un atlas probabiliste (Murgasova, et al., 2006). Toutefois, cette méthode est très sensible à l'alignement correct de l'atlas avec l'image. De plus, elle est très lente en exécution.

Dans la pratique, les médecins ne souhaitent pas attendre trop longtemps un résultat de segmentation, en raison du nombre important de patients, ce qui justifie le besoin d'un algorithme rapide de segmentation. Nombre d'auteurs ont appliqué des méthodes classiques de segmentation aux images IRM du cerveau, ces méthodes sont détaillées dans (Warfield, et al., 2000), (Kamber, et al., 2000), et (Cocosco, et al., 2003). Pour améliorer ces méthodes, des algorithmes de post-classification ont été proposés (Cocosco, et al., 2003). Pour illustrer les résultats obtenus par notre algorithme (Nakib, et al., 2007c), deux images IRM cérébrales (coupes axiales T2-niveau 11) avec leur classification multiniveau pour  $N=4$  et 5 sont présentées sur la figure 3.24. Dans le cas d'un sujet sain, les résultats sont illustrés sur les figures 3.24 (c) et (e), dans le cas d'une atrophie par les figures 3.24 (d) et (f). Le but est de détecter rapidement les différents espaces et la matière blanche (liquide encéphalo-rachidien) entourant les ventricules.

Pour quantifier les performances de l'algorithme d'optimisation OEP, nous définissons un facteur de gain en vitesse qui correspond au ratio du nombre d'évaluations dans le cas d'une recherche exhaustive et du nombre d'évaluations effectuées par notre algorithme. Le tableau 3.5 montre les résultats expérimentaux obtenus sur l'image de la figure 3.24 (a). Les performances en terme de vitesse et de temps d'exécution montrent l'efficacité de l'OEP. Cela confirme la rapidité de notre méthode en comparaison avec celles exposées dans (Qiao, et al., 2007; Drapaca, et al., 2005; Warfield, et al., 2000; Murgasova, et al., 2006; Song, et al., 2006; Kamber, et al., 2000; Cocosco, et al., 2003). D'autres résultats sont présentés dans la section A2.

### 3.10 CONCLUSION

Dans ce chapitre, nous avons présenté trois nouveaux critères de segmentation et nous avons adapté des métaheuristiques afin de les optimiser. A travers cette étude, nous avons mesuré l'adaptabilité des métaheuristiques pour optimiser des critères de segmentation :

#### ***Le recuit simulé***

L'adaptation du recuit simulé permet d'avoir de bons résultats proches de l'optimum global. Son adaptation requiert beaucoup de temps, mais elle est très simple.

#### ***Le recuit microcanonique***

Le recuit microcanonique permet de résoudre un problème de segmentation, mais nécessite un bon réglage, car il peut être piégé dans des minima locaux. L'augmentation du nombre d'itérations

permet d'améliorer les résultats, mais alourdit fortement l'algorithme. Son hybridation est nécessaire pour avoir de bonnes performances en terme de vitesse.

**L'optimisation par essaim particulaire**

Cette approche s'adapte bien à la résolution des problèmes de segmentation en continu, comme par exemple l'approximation de l'histogramme. Son adaptation en combinatoire n'est pas une tâche rude, et l'algorithme produit de bons résultats, avec un temps de convergence très court.

**Les colonies de fourmis**

Cette approche procure de bons résultats, du fait que l'algorithme a été conçu pour résoudre des problèmes d'optimisation combinatoire. Cependant, son paramétrage est assez difficile et sa complexité est supérieure à celle des autres métaheuristiques.

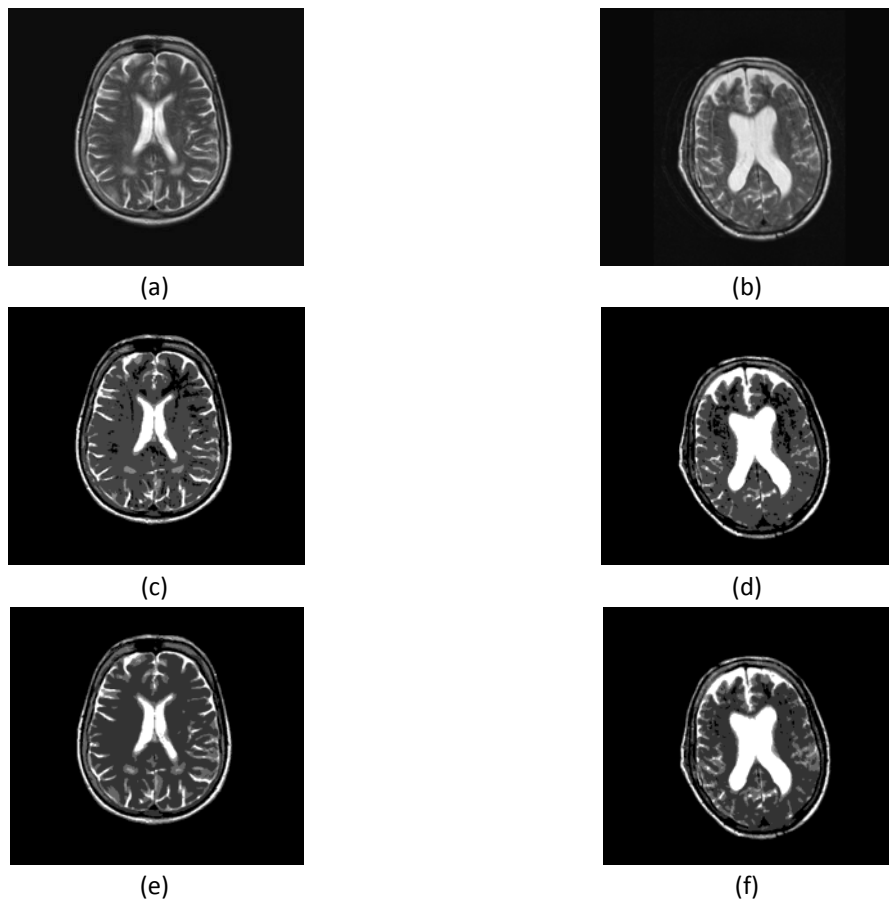


Figure 3.24: Segmentation d'images IRM saine et pathologique. (a) Image saine originale, (b) Image pathologique originale, (c) Segmentation de l'image en 4 classes  $t=(68 ; 128 ; 192)$ , (d) Segmentation de l'image en 4 classes  $t=(70 ; 129 ; 191)$ , (e) Segmentation de l'image en 5 classes  $t=(57 ; 105 ; 155 ; 202)$ , (f) Segmentation de l'image en 5 classes  $t=(58 ; 104 ; 154 ; 200)$ .  $t$  est le vecteur de seuils et  $\alpha = 0,6$ .

Tableau 3. 5: Evaluation du facteur de gain en vitesse. Résultats expérimentaux dans le cas de la figure 3.24 (a).

Nombre de classes	Temps (s)	Facteur de gain en vitesse
3	2,9	210
4	3,6	301
5	4,0	390

---

# CHAPITRE 4

## AMELIORATION DE LA SEGMENTATION PAR AGREGATION DE CRITERES

---

### 4.1 INTRODUCTION

---

Dans ce chapitre, nous présentons une méthode de segmentation basée sur l'agrégation de critères de segmentation (Nakib, et al., 2007d; Nakib, et al., 2007e). La métaheuristique que nous avons utilisée dans cette approche est celle du recuit simulé. Ce chapitre est scindé en quatre sections, la première est consacrée à un rappel de la formulation du problème de segmentation, en tant que problème d'optimisation multiobjectif. Les hypothèses sur lesquelles repose cette approche sont exposées aussi dans cette section. Ensuite, une description détaillée de l'algorithme est traitée dans la deuxième section. Elle comprend la présentation d'un algorithme de détection de pics, permettant d'estimer le nombre de classes dans une image, dans le cas d'une approche non supervisée de la segmentation. L'adaptation de la métaheuristique, et l'analyse de la complexité de calcul de la méthode, sont aussi présentées dans cette section. Dans la troisième section, nous présentons le réglage des paramètres de la métaheuristique, les résultats intermédiaires et une étude comparative avec, d'une part, des méthodes de référence, et d'autre part d'autres méthodes, parues récemment dans la littérature. Cette section se termine par quelques exemples de segmentations d'images IRM.

### 4.2 FORMULATION EN UN PROBLEME MULTIOBJECTIF

---

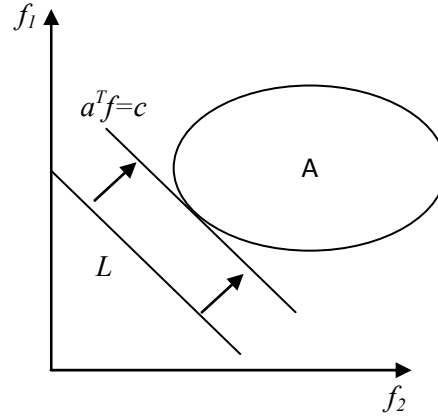
Dans le premier chapitre, nous avons présenté les méthodes de segmentation d'images paramétriques et non-paramétriques. Dans le deuxième chapitre, nous avons évoqué le fait qu'un seul critère de segmentation ne permet pas de segmenter tous les types d'images de façon correcte. Dans les paragraphes suivants, nous mettons en œuvre l'approche multiobjectif basée sur l'agrégation de fonctions, afin d'améliorer la qualité de la segmentation par un critère hybride : paramétrique et non-paramétrique.

Cette approche, qui consiste à transformer le problème multiobjectif en un problème mono-objectif, a l'avantage de produire une solution de compromis; ce qui ne nécessite pas l'intervention d'un expert pour le choix de la solution finale. La formulation mathématique de la nouvelle fonction objectif est donnée par l'expression (2. 2) définie dans le chapitre 2.

Les paramètres  $a_i$  sont appelés *facteurs d'importance*, et peuvent être considérés comme une mesure de l'apport de chaque fonction objectif dans le système multiobjectif. Le front de Pareto convexe peut être tracé en relançant l'algorithme plusieurs fois, avec différentes valeurs de  $a_i$ . La figure 4.1 présente l'illustration géométrique de cette approche dans le cas bi-objectif. Dans l'espace de recherche de la fonction objectif, une droite ( $L$ ), d'équation  $a^T f(\bar{x}) = c$ , est tracée. La minimisation de la nouvelle fonction objectif peut être interprétée comme la recherche de la valeur



de  $c$  pour laquelle  $L$  rejoint les bornes de l'ensemble des réalisations (A). Nous précisons que les valeurs de  $a_1$  et  $a_2$  définissent la pente de  $L$ .



$a^T$  est le vecteur de paramètres de pondération.

$c$  est une constante.

A est le domaine des réalisations.

La droite  $L$  correspond à :  $a_1 f_1 + a_2 f_2 = c$ .

$f_1$  et  $f_2$  sont les fonctions objectifs.

Figure 4. 1: Représentation géométrique de l'approche d'agrégation, dans un cas bi-objectif.

#### 4.2.1 CRITERES UTILISES

Nous avons expérimenté des combinaisons de plusieurs critères. A l'issue des nombreux tests effectués et des résultats obtenus, nous avons retenu les trois critères suivants:

##### **La variance intraclasse biaisée**

$$MVar(T) = \alpha \cdot \sum_{j=1}^{N_c} \left( \frac{\sigma_{Int}^2(j)}{\beta_j} + \gamma_j \right) \quad (4. 1)$$

où  $T \equiv (t_1, \dots, t_{N-1})$  est le vecteur de seuils. Ce critère est présenté dans la section 3.7.1.1.

##### **Critère de minimisation de la probabilité de l'erreur (PE) de classification**

Une fois l'approximation de l'histogramme effectuée par une combinaison de gaussiennes (section 3.2), le critère, défini par l'expression (4.2), peut être utilisé.

$$e(t_i) = P_i \int_{-\infty}^{t_i} p_i(x) dx + P_{i+1} \int_{t_i}^{+\infty} p_{i+1}(x) dx \quad (4. 2)$$

pour  $i = 1, 2, \dots, d-1$ , où  $d$  est le nombre de gaussiennes, déterminé par l'algorithme de détection de pics (voir section 2.1),  $t_i$  sont les seuils de segmentation,  $p_i(x)$  est la fonction de densité de probabilité, et  $P_i$  sont les amplitudes des gaussiennes. La probabilité d'erreur totale de classification est donnée par :

$$PE(T) = \sum_{i=1}^{d-1} e(t_i) \quad (4. 3)$$

où  $T$  est le vecteur de seuils, tel que:  $0 < t_1 < t_2 < \dots < t_{d-1} < 255$ .

##### **Critère de l'entropie de Shannon**

Nous rappelons que l'entropie d'un histogramme d'une image est donnée par:

$$H = -\sum_{i=0}^{255} p_i \log_2 p_i \quad (4.4)$$

Ce critère est plus détaillé dans la section 1.3.2.1.

#### 4.2.2 LA FONCTION OBJECTIF EQUIVALENTE

La fonction objectif équivalente, résultant de l'agrégation des critères *MVar* et *PE* décrits respectivement par les expressions (4.1) et (4.3), s'écrit :

$$MOBJ(T) = a_1.MVar(T) + a_2.PE(T) \quad (4.5)$$

où  $T$  est le vecteur de seuils de segmentation, tel que :  $0 < t_1 < t_2 < \dots < t_{d-1} < 255$ , ce qui correspond au vecteur  $\vec{x}$  dans (2.1). Les paramètres de pondération sont définis par :

$$a_1 = 1 - a_2 \text{ et } a_2 = \hat{\sigma}^2 / \sigma_{Histogramme}^2$$

où  $\hat{\sigma}^2$  est la variance de l'histogramme estimé et  $\sigma_{Histogramme}^2$  la variance de l'histogramme original.

Dans cette approche, nous supposons que les paramètres de pondération ( $a_1$  et  $a_2$ ) permettent d'atteindre la borne de l'ensemble des réalisations possibles. La différence entre notre approche, et celle utilisée par [ (Bhanu, et al., 1995), tient dans l'approche multiobjectif pour évaluer la qualité de la segmentation des images.

Le nombre d'évaluations, dans le cas d'une recherche exhaustive, peut être calculé par l'expression

suivante :  $\frac{L!}{(L+1-d)!(d-1)!}$ , où  $L$  est le nombre de niveaux de gris dans l'image (généralement

256). Par exemple, quand  $L=256$  et  $d=2$ , le nombre d'évaluations de la fonction objectif est de 256 et, quand  $L=256$  et  $d=3$ , il est de 32640.

#### 4.2.3 SEGMENTATION PAR COMBINAISON DE CRITERES (CCS)

Comme nous l'avons déjà signalé, la nature combinatoire du problème ne permet pas d'explorer de manière exhaustive l'espace des solutions, en particulier quand le nombre de classes augmente. Les courbes des fonctions objectifs de segmentation varient d'une image à une autre, par conséquent le nombre de minima locaux varie aussi. D'où le besoin d'une métaheuristique robuste, capable d'identifier les minima globaux. Dans cette partie, nous avons expérimenté le recuit simulé (RS) détaillé dans la section 3.2. L'initialisation de la méthode et sa manière de parcourir, par des petits pas, l'espace de recherche sont effectuées de manière aléatoire.

L'énergie de la fonction objectif correspond à la valeur de la fonction *MOBJ* pour chaque solution candidate. Les solutions qui permettent de diminuer cette énergie (bonnes solutions) sont toujours acceptées, tandis que celles qui augmentent l'énergie (solutions dégradantes) peuvent être acceptées, avec une probabilité qui dépend conjointement de la différence entre les deux états et de la température du palier. Pour calculer cette probabilité, nous avons utilisé l'algorithme de Metropolis (section 2.4.1.1).

Nous avons adapté le recuit simulé de manière à ce que la valeur de la constante de Boltzmann  $K_B$  soit ajustée durant la simulation, dans le but d'avoir une "faible" probabilité, prédéfinie (0,5%), d'accepter les solutions dégradantes à la température finale. La température est réduite suivant une loi géométrique :  $Temp_{k+1} = a_{dec} \times Temp_k$ , où  $a_{dec}$  est une constante réelle dans l'intervalle  $[0, 1]$  (voir les valeurs utilisées dans le tableau 4.1).

Tableau 4.1: Valeurs des paramètres de RS et RSC.

Paramètre	Valeur
$R_{acc}$	0,05
$a_{dec}$	0,9
$c_r$	$10^{-6}$
$Temp_f$	0,1
$p$	autour de $n/3$

D'autres lois de réduction ont été testées, comme la loi gaussienne, la loi exponentielle et la loi de Cauchy, mais dans notre cas la loi géométrique nous a permis d'avoir de bonnes performances en terme de vitesse de convergence. Les valeurs initiales des seuils de segmentation sont situées aux centres des intervalles séparant les gaussiennes successives. La procédure est présentée dans l'Algorithme 4.1.

Algorithme 4.1: Algorithme de segmentation CCS.

1. **Détecter** les pics pour déterminer le nombre de classes *a priori* et les valeurs initiales de l'algorithme RSC.
2. **Approcher** l'histogramme de l'image par une combinaison de gaussiennes avec RSC (PE-RSC).
3. Début de la procédure de segmentation
  - 3.1. **Initialiser** la température.
  - 3.2. **Perturber** aléatoirement la solution courante.
  - 3.3. **Calculer** la variation correspondante de *MOBJ*.
  - 3.4. **Appliquer** le critère de Metropolis pour accepter ou rejeter la nouvelle solution.
  - 3.5. **Si** la fin du palier de température n'est pas atteinte, aller à l'étape 3.2.
  - 3.6. **Si** aucun des critères d'arrêt n'est vérifié :
    - **Ajuster** la température.
    - **Aller** à l'étape 3.2.
4. Fin de la procédure de seuillage.
5. **Afficher** les résultats: seuils de segmentation, image segmentée.

Dans le cas d'un histogramme bimodal, l'espace de recherche est étendu en dehors des moyennes des gaussiennes, s'il permet de maximiser l'entropie. Ce procédé permet, en particulier, d'améliorer la segmentation des images dont la scène présente un éclairage non uniforme. En effet, il peut y avoir un fort chevauchement des deux classes de pixels détectées. Certains pixels censés appartenir à l'arrière-plan vont être classés parmi les pixels du premier-plan, et vice versa. Dans ce cas, les pics de l'histogramme ne représentent pas le premier et l'arrière-plan de l'image. Dans le cas d'un histogramme unimodal, la segmentation se fera en deux classes et le seuil optimal est celui qui maximisera le critère *MVar*, car le critère *PE* ne peut pas être calculé.

## 4.3 DESCRIPTION DETAILLÉE DE L'ALGORITHME CCS

### 4.3.1 ALGORITHME DE DETECTION DE PICS

La méthode que nous proposons est totalement non-supervisée. Par conséquent, aucune intervention d'expert n'est envisageable, pour ajuster les paramètres de segmentation, par exemple le nombre de classes à identifier.

Ainsi, nous avons développé un algorithme efficace, pour calculer automatiquement le nombre initial de classes dans l'image. Il est basé sur une nouvelle méthode de détection et d'identification des pics les plus significatifs dans l'histogramme de l'image. Le nombre de ces derniers correspond directement au nombre de classes de pixels dans l'image.

Dans la littérature, la plupart des algorithmes ont un temps d'exécution long, ils choisissent les pics en évaluant leurs aires, ou ils sont trop complexes (Cheng, et al., 2000; Chernov, et al., 2004), ou encore ils ignorent un pic s'il n'est pas assez large. Leurs résultats expérimentaux montrent la nécessité d'un prétraitement de l'histogramme, pour que l'algorithme fonctionne correctement.

Le "cahier des charges" de notre algorithme de détection tient compte de trois propriétés : simplicité, rapidité et non-supervision par un expert. Celui-ci fonctionne de façon itérative. Il a été testé sur 101 images, dont 42% ont un histogramme unimodal et 58% ont un histogramme multimodal. Les différentes étapes de l'algorithme, sont présentées dans l'Algorithme 4.2. Pour plus de détails, voir (Nakib, et al., 2007d).

Algorithme 4.2: Algorithme de détection de pics.

1. Détection de l'ensemble S1 des maxima locaux.
2. Détection de l'ensemble des maxima significatifs S2 à partir de S1.
3. A partir de S2, faire un seuillage :
  - 3.1. Suppression des "petits" pics en les comparant au pic le plus haut.
  - 3.2. Détermination de l'ensemble S3 en réduisant le nombre de pics en mesurant leurs inter-distances en abscisses.
  - 3.3. Affinage du nombre de pics en mesurant les inter-distances en coordonnées polaires.
4. Afficher le nombre final de pics, leurs amplitudes ainsi que leurs emplacements.

#### 4.3.2 RECUIT SIMULE CONTINU POUR L'APPROXIMATION DE L'HISTOGRAMME

Afin d'approcher l'histogramme de l'image, nous utilisons l'algorithme RSC, décrit dans le chapitre précédent (section 3.2), pour optimiser la fonction objectif décrite par l'expression (3.3). Cet algorithme est inspiré de l'algorithme ESA ("*Enhanced Simulated Annealing*") décrit dans (Siarry, et al., 1997). Nous ne présentons ici que ce qui le différencie de l'algorithme classique du recuit simulé: l'initialisation et les critères d'arrêt.

Le principe d'ESA consiste à discrétiser l'espace de recherche avec un pas adaptatif, et à remplacer l'algorithme de Metropolis, qui est généralement utilisé dans l'espace des variables  $\mathfrak{R}^n$ , par un autre algorithme de recherche similaire dans des sous-espaces  $\mathfrak{R}^p$ , avec  $p \ll n$ . Par rapport à l'algorithme recuit simulé classique, nous avons donc introduit trois améliorations : une méthode adaptative, pour sélectionner le pas de discrétisation de l'espace des variables, une transformation locale, pour explorer l'espace de recherche et un ajustement adaptatif de la constante de Boltzmann (voir section 3.2).

##### 4.3.2.1 PRINCIPE

Soit  $V$  un vecteur dans un espace à  $n$ -dimensions  $\mathfrak{R}^n$  dont les composantes sont  $(x_1, x_2, \dots, x_n)$  et  $J$  la fonction objectif. En partant d'un vecteur initial  $V_{init}$ , le recuit simulé engendre une suite de vecteurs  $V_1, V_2, \dots$  dans un sous-espace  $\mathfrak{R}^p$  de  $\mathfrak{R}^n$ . Un nouveau vecteur est déterminé à partir du vecteur  $V$

courant, en perturbant uniformément les  $p$  composantes choisies par l'algorithme de partitionnement :

- Pour chaque variable  $x_i$ , un pas de discrétisation  $\Delta_i$  différent est utilisé.
- Les pas  $\Delta_i$  sont initialisés à 20% de la largeur de l'intervalle de recherche des variables  $x_i$ .
- Pour chaque  $\Delta_i$ , nous spécifions une loi  $F$  de calcul du mouvement de  $x_i$  :  $x'_i = F(x_i, \Delta_i)$ .
- $\Delta_i$  est mis à jour à la fin de chaque palier de température.
- Nous utilisons un taux d'acceptation  $R_i$  des vecteurs tirés pour mettre à jour  $\Delta_i$  :
  - Si  $R_i > 20\%$ ,  $\Delta_i$  est doublé.
  - Si  $R_i < 5\%$ ,  $\Delta_i$  est divisé par 2,
- Pour chaque variable  $x_i$  appartenant à l'ensemble des  $p$  variables sélectionnées auparavant, la valeur perturbée est calculée via la formule suivante :

$$x'_i = x_i \pm rand_x \cdot \Delta_i$$

où  $rand_x$  est un nombre aléatoire dont la densité de probabilité est uniforme, dans l'intervalle  $[0, 1]$ . Le signe  $\pm$  est tiré de manière aléatoire.

- Les  $n$  variables du problème sont modifiées par groupe de  $p$  variables, où  $p \cong n/3$ . Les fréquences de mouvement des différentes variables sont égalisées.
- Les critères d'arrêt : la procédure peut s'arrêter quand au moins un des quatre critères d'arrêt est vérifié : ces critères concernent la température, le vecteur de pas, le nombre d'évaluations de la fonction objectif, et le nombre de mouvements acceptés. Les deux critères que nous avons utilisés sont détaillés plus bas.

#### 4.3.2.2 LES CRITERES D'ARRET UTILISES

##### 4.3.2.2.1 TEMPERATURE

La température initiale est fonction de la moyenne des variations de la fonction objectif  $\Delta J_0$  et du taux initial d'acceptation des perturbations dégradantes ( $R_{acc}$ ), selon la relation suivante :

$$R_{acc} = \exp(-\Delta J_0 / Temp_0) \quad (4.6)$$

d'où

$$Temp_0 = -\Delta J_0 / \log(R_{acc}) \quad (4.7)$$

Cette expression de  $Temp_0$  montre clairement qu'une valeur élevée de  $\Delta J_0$  produit une température initiale élevée, ce qui montre que la procédure d'initialisation prend en compte la nature des variations de  $J$ . Les perturbations initiales sont calculées avec un pas de discrétisation de l'espace élevé, ce qui produit une variation  $\Delta J_0$  élevée de  $J$ . Nous avons choisi pour  $R_{acc}$  la valeur 0,05 (réglée empiriquement).

Comme la température est abaissée progressivement, nous devons définir une température limite ( $Temp_f$ ) à laquelle la procédure doit être interrompue. Nous l'avons fixée à 0,1.

Il est à signaler que nos algorithmes ont été codés sous Matlab, et exécutés sur un processeur Intel Pentium 4 à 2,26GHz.

##### 4.3.2.2.2 LE VECTEUR DE PAS

Comme la température  $Temp$  est abaissée progressivement, chaque composante du vecteur des pas de discrétisation de l'espace de recherche est aussi diminuée. Ainsi, nous pouvons considérer cette

diminution comme un nouveau critère d'arrêt local, quand au moins une des composantes a diminué son pas  $\Delta_i$  d'un facteur  $c_r$  par rapport à sa valeur initiale. Nous avons opté pour la valeur  $10^{-6}$ .

### 4.3.3 COMPLEXITE DE L'ALGORITHME CCS

Soient  $k_1 = (Temp_0 - Temp_f) / R_{acc}$  et  $k_2 = (Temp_0 - Temp_f) / a_{dec}$  les nombres d'itérations de RSC et RS, respectivement.

1. La complexité de l'algorithme de détection de pics est  $O(N)$ , où  $O(.)$  est la fonction de Landau et  $N = \text{Card}(\arg \max(hist \neq 0))$ .
2. La complexité de l'algorithme d'approximation de l'histogramme par RSC (GCF) est égale à  $O(g(d))$ , où  $g(d) = O(d.k_1)$  et  $d$  est le nombre de gaussiennes.
3. Le critère  $PE$  nécessite  $O((d-1)N)$  opérations.
4. Le critère  $MVar$  requiert  $O((d-1)(N^2 + S^2))$  opérations, où  $S$  est le nombre de pixels dans l'image.

D'où l'approximation de la complexité totale de l'algorithme de segmentation :

$$C_{CCS} = O(N + g(d) + k_2((d-1)N + d)) \quad (4.8)$$

## 4.4 REGLAGE DES PARAMETRES ET RESULTATS

Dans cette partie, nous présentons des résultats de segmentations d'images test (présentées sur la figure 4.1), obtenus avec l'algorithme CCS. Les images test utilisées ont trois types d'histogrammes, présentés sur la figure 4.2. Cette partie est scindée en trois sous-parties : dans la première, nous détaillons les réglages des deux algorithmes d'optimisation (RS et RSC). Dans la deuxième, nous présentons les résultats de l'algorithme d'approximation d'histogramme (PE-RSC), dans le cas des histogrammes unimodal et multimodal. Enfin, la dernière sous-partie est consacrée aux résultats de segmentation des images test. Une analyse plus détaillée figure dans (Nakib, et al., 2007d; Nakib, et al., 2007e).

### 4.4.1 REGLAGE DES PARAMETRES

La nécessité de l'étape de réglage vient des inconvénients majeurs des métaheuristiques : plusieurs paramètres à régler et l'inexistence de réglages par défaut. En outre, chaque problème traité a ses propres réglages. Dans notre cas, les valeurs des paramètres de RS et RSC sont récapitulées dans le tableau 4.1. Ces valeurs ont été déterminées de manière empirique sur 101 images. Les images test sont : Lena, moon, etc. de *Mathworks* et autres images téléchargées de la base de données de l'Université de Berkeley (Berkeley, 2007). Plus précisément, les images sont à 40% des images infrarouges, 25% des images classiques, 25 % des images IRM et angiographiques et 10% des images synthétiques.

### 4.4.2 RESULTATS DE L'APPROXIMATION D'UN HISTOGRAMME

Les figures 4.3 (a) à (c) présentent respectivement les histogrammes originaux et approchés par l'algorithme PE-RSC des images des figures 4.2 (a) à (c).

Nous précisons que le nombre et les valeurs initiales des paramètres des gaussiennes ( $P_i$  et  $\mu_i$ ) ont été détectés et déterminés par l'algorithme de détection de pics; ce qui accélère la convergence de la procédure d'optimisation RSC. Les valeurs optimales trouvées sont présentées dans le tableau 4.2.

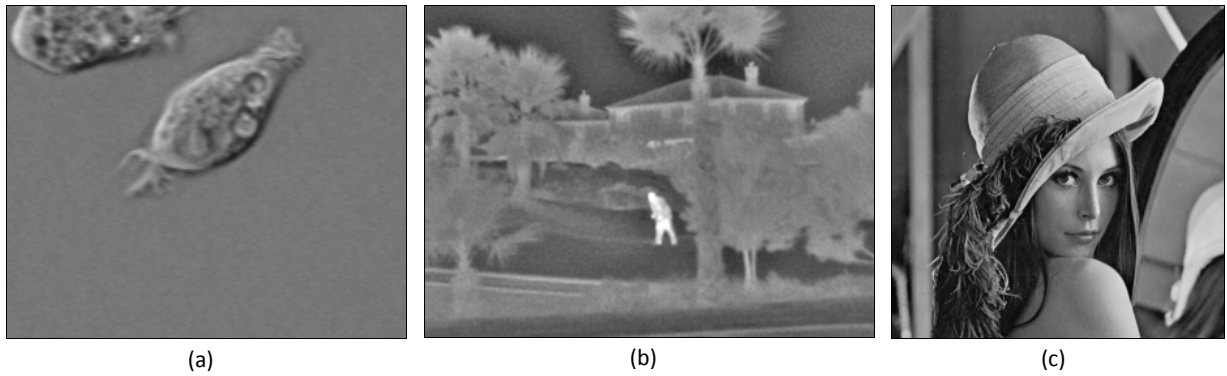


Figure 4. 2: Images test originales. (a) Cell, (b) Infrarouge, (c) Lena.

Tableau 4. 2: Résultats expérimentaux de CCS.

Image	Taille	Nombre de classes	Paramètres des gaussiennes	Temps (s)
Figure 4.2 (a)	191x159	2	$P(0,1325)$ $\mu(120)$ $\sigma(3,70)$	6,20
Figure 4.2 (b)	303x224	2	$P(0,0147 ; 0,0216)$ $\mu(90 ; 131)$ $\sigma(21,2913 ; 20,6369)$	10,62
Figure 4.2 (c)	256x256	5	$P(0,0089 ; 0,0084 ; 0,0084 ; 0,0084 ; 0,0034)$ $\mu(24 ; 66 ; 96 ; 135 ; 201)$ $\sigma(16,916 ; 15,115 ; 16,567 ; 22,236 ; 20,2157)$	38,95

Les tests effectués sur les différentes formes d'histogrammes montrent l'efficacité de RSC pour optimiser la fonction objectif  $PE$  : le cas unimodal est présenté sur la figure 4.3 (a), le cas bimodal sur la figure 4.3 (b) et le cas multimodal sur la figure 4.3 (c). Les temps d'exécution, pour les différents histogrammes, sont indiqués dans le tableau 4.2. Comparés à ceux de (Zahara, et al., 2004), ces temps sont tout à fait acceptables.

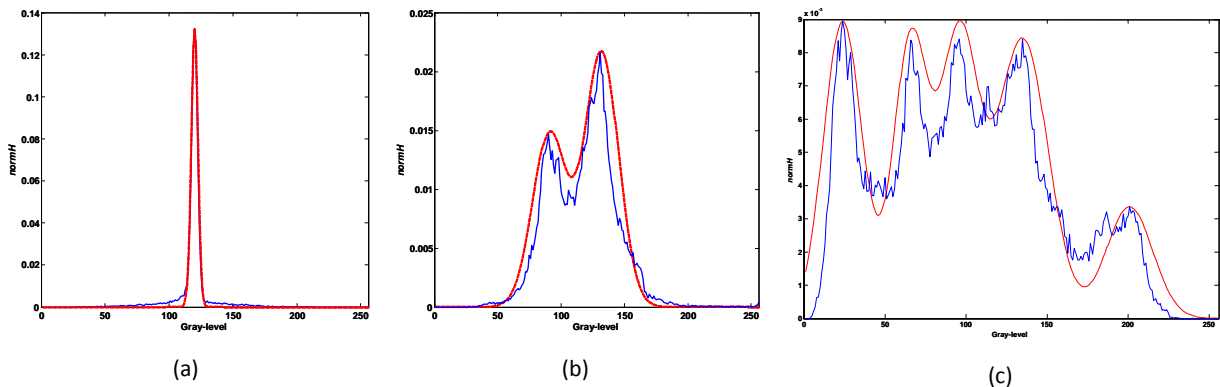


Figure 4. 3: Histogrammes originaux et approchés pour les images : (a) Cell, (b) Infrarouge, (c) Lena.

#### 4.4.3 QUELQUES EXEMPLES DE SEGMENTATIONS

Une illustration de nos résultats sur des images unimodales est présentée sur la figure 4.4 (a). Comme nous pouvons le constater, bien que l'histogramme de l'image soit unimodal, la méthode CCS produit un résultat de segmentation acceptable, grâce à l'hybridation. En effet, si le critère  $PE$  avait été utilisé seul, il n'y aurait pas eu de résultat de segmentation, car une seule gaussienne est insuffisante pour segmenter une image. Une comparaison avec la méthode d'Otsu est présentée dans (Nakib, et al., 2007e).

Dans le cas d'un histogramme bimodal (figure 4.3 (b)), la figure 4.4 (b) présente un exemple de segmentation d'une image infrarouge qui, au vu de son histogramme, peut paraître facile à segmenter. Le problème posé est que le seuil optimal n'est pas situé dans la vallée, ce qui induit en erreur la plupart des méthodes paramétriques et non-paramétriques. Nous n'avons pu obtenir une segmentation acceptable (le but est de détecter la personne sans l'arrière plan), que grâce à la dernière partie de l'algorithme (voir section 4.2.3). Une comparaison avec d'autres méthodes figure dans (Nakib, et al., 2007d).

Concernant les images dont l'histogramme est multimodal, l'exemple de la figure 4.4 (c) illustre les performances de l'algorithme CCS. Le nombre de classes est détecté de manière non-supervisée et une bonne séparation entre les classes de l'image est aussi observée. La comparaison avec la méthode basée sur l'optimisation du critère  $PE$  isolé est présentée dans (Nakib, et al., 2007e).

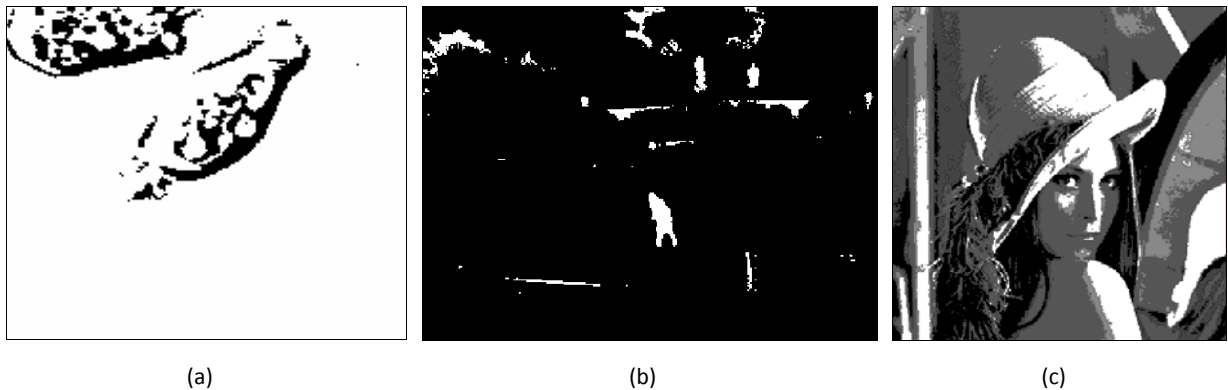


Figure 4.4: Résultats de segmentation par CCS : (a) Cell ( $t=108$ ), (b) Infrarouge ( $t=151$ ), (c) Lena  $t=(44 ; 71 ; 129 ; 147)$ .

Dans (Nakib, et al., 2007d), nous avons comparé, sur la base d'images synthétiques, les performances de la méthode CCS avec d'autres méthodes de référence : Otsu et la méthode PE-RSC et d'autres méthodes récemment publiées : la méthode de segmentation par maximisation de l'entropie de Tsallis ( $TE$ ) (Sahoo, et al., 2006) (voir section 1.3.2.2.2), la méthode VE (Ng, 2006) (voir section 1.3.1). Les résultats obtenus montrent que CCS est supérieur à ces méthodes, notamment lorsque les images test sont bruitées avec un bruit blanc additif et un bruit multiplicatif (figure 4.5, image F). Les images originales sur lesquelles les tests ont été effectués sont présentées sur la figure 3.23. D'autres résultats sont présentés dans l'Annexe (section A3).

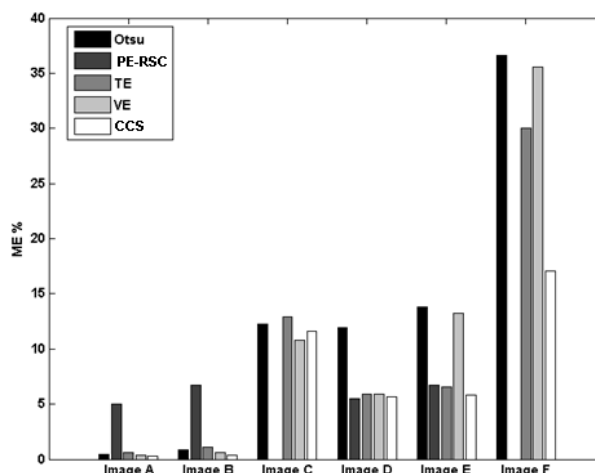


Figure 4. 5: Performances de l'algorithme CCS comparées à Otsu, PE-RSC, TE, et VE pour les images de la figure 3.23 (chapitre 3). La meilleure méthode est celle qui minimise le critère  $ME$ .



#### 4.4.4 EXEMPLES DE SEGMENTATIONS D'IMAGES IRM

Nous présentons sur la figure 4.6 une illustration des résultats obtenus sur des images IRM. Sur la figure 4.6 (a) est présenté le résultat de la segmentation de l'image IRM de la figure 3.25 (a). Dans ce cas, l'algorithme a détecté la présence de trois classes, ce qui nous permet d'évaluer le volume du liquide encéphalo-rachidien. Nous pouvons aussi remarquer que le cortex a aussi été bien détecté. La figure 4.6 (b) illustre les résultats obtenus dans le cas d'une image IRM pathologique (atrophie). Le résultat obtenu est une segmentation en quatre classes. Comme nous pouvons le constater, le volume du liquide encéphalo-rachidien est bien segmenté, mais le cortex n'est pas bien détecté sur la partie inférieure de l'image segmentée, du fait du faible contraste dans cette zone et des bruits d'acquisition de l'image. D'autres résultats sont présentés dans l'Annexe A3.



Figure 4. 6: Résultats de segmentations d'images IRM de la figure 3.25. (a) Segmentation de l'IRM du cerveau d'un sujet sain en 3 classes  $t=(94; 154)$ , (b) Segmentation de l'IRM en 4 classes  $t=(19; 123; 136)$ , dans le cas d'une pathologie du cerveau.

## 4.5 CONCLUSION

Dans ce chapitre, nous avons présenté une méthode de segmentation basée sur l'approche d'agrégation de critères. Pour cette approche, nous avons élaboré un nouveau critère de segmentation hybride, combinant deux critères paramétriques et un critère non-paramétrique.

La comparaison avec d'autres méthodes de segmentation (tableau 3.4) a montré que notre approche permet d'améliorer considérablement les résultats de segmentation.

Cependant, la méthode ne permet pas d'avoir de bons résultats sur des images fortement bruitées, dans le cas d'un bruit blanc gaussien. Dans ce cas, en effet, l'histogramme est unimodal et par conséquent un seul critère de segmentation est utilisé par défaut.

Dans le chapitre suivant, nous explorerons l'approche non Pareto de l'optimisation multiobjectif pour segmentation, qui pourrait donner de meilleurs résultats.

---

# CHAPITRE 5

## APPLICATION DE LA DERIVATION NON-ENTIERE ET DE L'APPROCHE NON-PARETO A LA SEGMENTATION

---

### 5.1 INTRODUCTION

---

Dans ce chapitre, nous présentons une méthode de segmentation basée sur une approche multiobjectif non-Pareto (Nakib, et al., 2006b; Nakib, et al., 2007h; Nakib, et al., 2007g; Nakib, et al., 2007f). Le principe d'un système de segmentation d'images par approche non-Pareto consiste à segmenter l'image avec plusieurs critères, les uns indépendamment des autres. Ce système de segmentation produit un ensemble de solutions optimales selon chaque critère, dont le cardinal est égal au nombre de critères utilisés (figure 5.1).

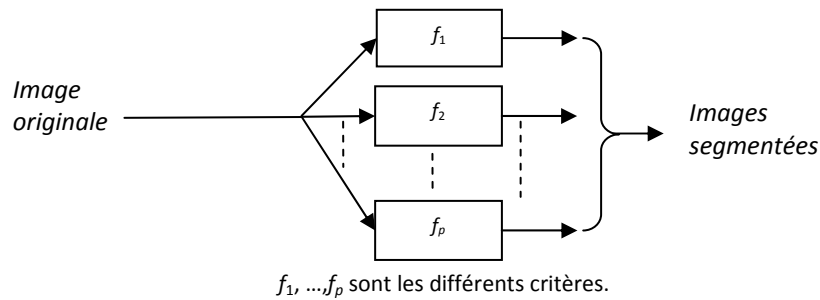


Figure 5. 1: Principe d'un algorithme de segmentation par l'approche non-Pareto.

Afin de réduire l'espace de recherche des solutions, de manière à limiter le nombre d'évaluations des critères utilisés, nous introduisons en amont un bloc de dérivation non entière. Il nous faut aussi un bloc de sélection de la meilleure solution parmi les  $p$  solutions proposées, où  $p$  est le nombre de critères. Pour résoudre ce problème et rendre l'algorithme non-supervisé, nous avons introduit en aval un algorithme de sélection, permettant d'élire la meilleure solution, au sens de la régularité géométrique des images segmentées. Le schéma de l'algorithme complet de segmentation par l'approche non-Pareto est présenté sur la figure 5.2.

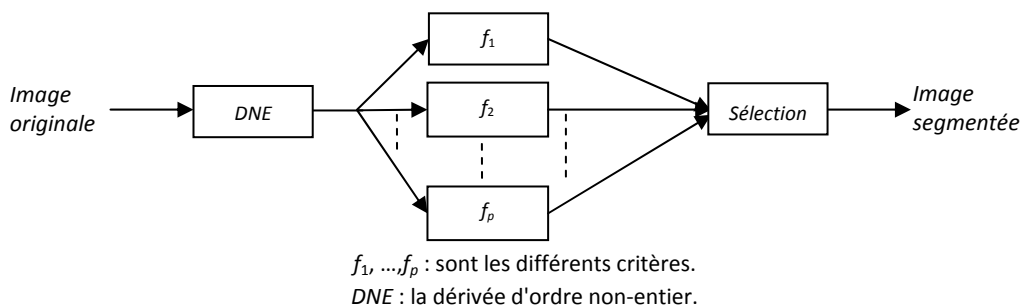


Figure 5. 2: Schéma-bloc de l'algorithme de segmentation par approche non-Pareto et DNE.

Dans l'algorithme, nous avons utilisé la dérivation non-entière (DNE) sous deux formes discrètes : le cas unidimensionnel défini par Grünwald (Oldham, et al., 1974), et le cas bi-dimensionnel que nous avons proposé. Dans les sections suivantes, nous allons décrire chaque bloc de cet algorithme.

La première section est dédiée au formalisme de la DNE à une dimension (1D), et à son extension aux espaces à deux dimensions (2D). Dans la deuxième section, nous présentons les propriétés d'un histogramme et d'une image dérivés avec un ordre non-entier. Les deux variantes de l'algorithme, basées sur la DNE-1D et DNE-2D, sont détaillées dans la troisième section. L'analyse de l'algorithme, sa comparaison avec d'autres méthodes, ainsi que des exemples de segmentations font l'objet de la quatrième section. Le chapitre se termine par une conclusion.

## 5.2 FORMALISME DE LA DERIVATION NON-ENTIERE (DNE)

La théorie de la dérivation non-entière (dérivée fractionnaire) date des travaux de Leibniz et L'Hospital en 1665. La dérivation d'ordre non-entier généralise la notion de dérivée d'ordre entier  $\alpha$  d'une fonction  $f(x)$  par rapport à la variable  $x$  à des valeurs non-entières de  $\alpha$ . Lorsque  $\alpha$  est négatif, nous sommes en présence d'une intégration non-entière. Nous sommes en présence d'une dérivation non-entière lorsque l'ordre  $\alpha$  est positif. Toutefois, nous avons utilisé indifféremment le terme dérivation, que l'ordre soit positif ou négatif. La propriété remarquable de la DNE est qu'elle peut être appliquée à des fonctions qui ne sont pas différentiables au sens classique. Contrairement à la dérivée d'ordre entier, la DNE en un point  $x$  n'est pas déterminée par un "petit" voisinage arbitraire de  $x$ . En d'autres termes, la DNE n'est pas une propriété locale de la fonction. Un recueil des différentes approches pour définir la DNE figure dans (Oldham, et al., 1974). Au départ, la théorie des dérivées fractionnaires a été considérée comme une branche relevant des mathématiques. Récemment, la DNE a été appliquée dans différents domaines : en automatique, où elle est utilisée pour le calcul d'une commande robuste (Oustaloup, 1996), dans la résolution de problèmes inverses mal posés de transfert thermique (Battaglia, 2001), et dans de nombreux autres domaines, notamment en réseaux de neurones (Ramus, et al., 2002), en traitement d'images pour la détection de contours (Mathieu, et al., 2003) et en traitement de signal (Ferdjani, et al., 2000; Nakib, et al., 2002).

### 5.2.1 FORMALISME DE LA DNE-1D

Le formalisme de la DNE de Riemann-Liouville est défini par (Oldham, et al., 1974) :

$$D^{-\alpha} f(x) = \frac{1}{\Gamma(\alpha)} \int_c^x (x - \xi)^{\alpha-1} f(\xi) d\xi \quad (5.1)$$

où  $f(x)$  est une fonction réelle et causale,  $x > 0$ ,  $\alpha$  est l'ordre de la DNE,  $\text{Re}(\alpha) > 0$  ( $\alpha$  peut être un nombre complexe),  $c$  est la référence de l'intégrale et  $\Gamma$  est la fonction Euler-gamma.

L'approximation de la forme discrète de DNE d'ordre  $\alpha$  est donnée par (Oldham, et al., 1974) :

$$g(x) = D^{\alpha} f(x) \approx \frac{1}{h^{\alpha}} \sum_{k=0}^M \omega_k(\alpha) f(x - kh) \quad (5.2)$$

où  $h$  est le pas d'échantillonnage,  $M$  est le nombre d'échantillons,  $x = M.h$ , et  $\omega_k(\alpha)$  sont définis par :

$$\omega_0(\alpha) = 1, \quad \omega_{k+1}(\alpha) = \frac{(k+1) - \alpha - 1}{(k+1)} \omega_k(\alpha), \quad k = 0, 1, 2, \dots, M-1 \quad (5.3)$$

L'expression (5.2) est équivalente à celle de Riemann-Liouville, quand  $h$  tend vers zéro. A partir de l'expression (5.2), la fonction  $g(x)$  peut être interprétée comme la sortie d'un filtre discret dont l'entrée est  $f(x)$ . Sa réponse impulsionnelle est donnée par :

$$h(k) = \begin{cases} -\omega_k(\alpha)/h^\alpha & k = 1, 2, \dots, M \\ 0 & k = 0 \end{cases} \quad (5.4)$$

La définition (5.2) montre que la DNE prend en compte tout le passé de la fonction  $f$ . Plus de détails sur la définition de la DNE figurent dans (Oldham, et al., 1974; Prodlubny, 2002).

### 5.2.2 EXTENSION DE LA DNE-1D AU CAS 2D

Comme une image peut être considérée comme une fonction à deux dimensions, réelle et bornée  $f(x, y)$ , nous avons proposé d'étendre l'approximation de la DNE1D au cas 2D (Nakib, et al., 2007f) :

$$D^\alpha f(x, y) = \left(\frac{\partial}{\partial x}\right)^\alpha \left(\frac{\partial}{\partial y}\right)^\alpha f(x, y) \approx \frac{1}{h^{2\alpha}} \sum_{k=-\lfloor M/2 \rfloor}^{\lfloor M/2 \rfloor} \sum_{l=-\lfloor N/2 \rfloor}^{\lfloor N/2 \rfloor} p(k, l) \cdot f(x - kh, y - lh) \quad (5.5)$$

où  $M$  et  $N$  représentent le nombre d'éléments de  $f$  qui sont pris en compte pour le calcul de l'image différenciée.  $M \times N$  est la taille du masque, et  $\lfloor x \rfloor$  représente la partie entière du nombre  $x$ .

$p(k, l) = \omega_k(\alpha) \times \omega_l(\alpha)$  sont les éléments de la matrice  $P_M^{(\alpha)}(p(k, l))$ , calculés à partir de

l'expression (5.3), qui correspondent aux composantes horizontales et verticales, respectivement.

L'expression (5.5) peut être vue comme un produit de convolution en 2D d'une image  $f$  avec un filtre 2D défini par  $P_M^{(\alpha)}$  :

$$g(x, y) = D^\alpha f(x, y) = f(x, y) \otimes P_M^{(\alpha)}(x, y) \quad (5.6)$$

où  $\otimes$  est l'opérateur de convolution en 2D. Les réponses en fréquence (amplitudes et phases), ainsi que les réponses impulsionnelles du filtre DNE-2D (de taille 3x3) pour différentes valeurs de  $\alpha$ , sont présentées sur la figure 5.3.

## 5.3 PROPRIETES DE L'HISTOGRAMME ET DE L'IMAGE DIFFERENCIEE

Nous présentons brièvement les propriétés d'un histogramme d'une image et d'une image différenciée avec un ordre  $\alpha$ . Ces propriétés sont plus détaillées dans (Nakib, et al., 2007g) pour le cas 1D, et dans (Nakib, et al., 2007f) pour le cas 2D.

### 5.3.1 CHANGEMENT D'ECHELLE DANS LE CAS 1D

Lorsqu'un histogramme est différencié avec un ordre  $\alpha$  positif, son amplitude est comprimée. Nous avons remarqué qu'avec l'augmentation de l'ordre de 0 vers 1, l'intervalle de variations diminue considérablement. En revanche, dans le cas où la valeur de  $\alpha$  s'étend de 0 à -1, l'intervalle de variations de l'amplitude augmente. Afin d'illustrer cette propriété, la figure 5.4 présente les formes d'un histogramme différencié avec un ordre positif.

Sur la figure 5.4, les courbes correspondent à :  $\text{Max}(0, D^\alpha f(x))$ . En effet, si  $\forall x \in ]a, b[$   $f(x) > 0$  et  $\forall x \notin ]a, b[$   $f(x) = 0$ , et si  $\alpha \in ]0, 1[$  (dérivation), alors  $\exists c < b$  tel que  $\forall x \in ]a, c[$ ,  $D^\alpha f(x) > 0$  et,  $\forall x > c$ ,  $D^\alpha f(x) < 0$ . Si  $\alpha \in ]1, 2[$ , alors  $D^\alpha f(x) > 0$  dans les deux intervalles.

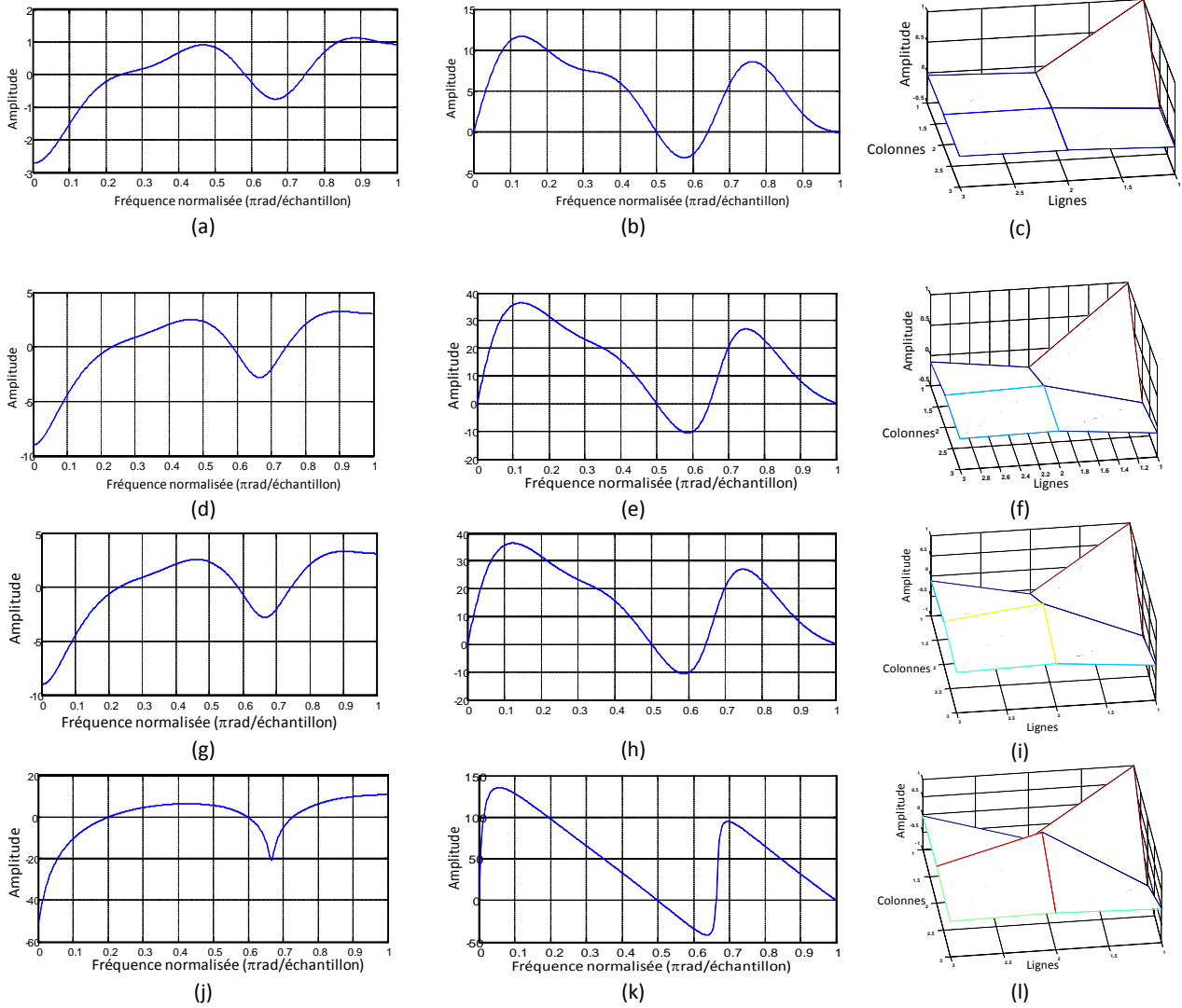


Figure 5. 3: Les réponses en fréquence (amplitudes et phases) et réponses impulsionnelles des filtres DNE-2D (de taille 3x3) pour différentes valeurs de  $\alpha$ . (a), (b) et (c):  $\alpha=0,1$ , (d), (e) et (f):  $\alpha=0,3$ , (g), (h) et (i):  $\alpha=0,6$ , (j), (k) et (l):  $\alpha=0,9$ .

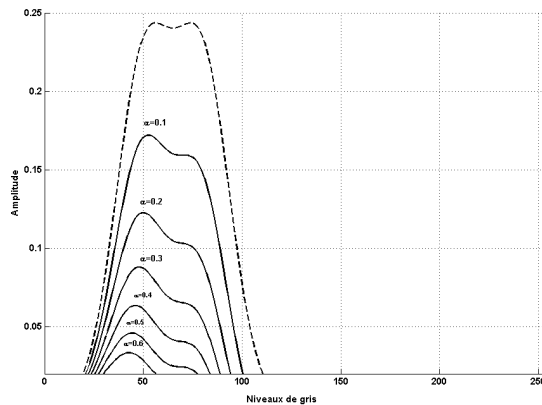


Figure 5. 4: Diminution de l'amplitude de l'histogramme avec  $\alpha > 0$ . Fonction originale causale et non négative différenciée, avec  $\alpha$  variant de 0,1 à 0,6, avec un pas de 0,1.

### 5.3.2 INTERPRETATION GEOMETRIQUE DANS LE CAS DE LA DNE-2D

Afin d'illustrer l'effet de l'application de la DNE-2D sur une image, nous considérons l'image test *avion* de la figure 5.5 (a). Nous représentons les variations des niveaux de gris en les traçant comme une fonction 2D (figure 5.5 (b), (d), (f) et (h)). L'interprétation géométrique est obtenue par l'analyse des variations de niveaux de gris de l'image.

Sur la figure (5.5), les résultats du filtrage sont bornés :

$$g(x,y) = \begin{cases} L-1 & \text{si } g(x,y) \geq L-1 \\ 0 & \text{si } g(x,y) \leq 0 \end{cases} \quad (5.7)$$

où  $L$  est le nombre total de niveaux de gris, généralement 256.

Comme nous pouvons le remarquer sur la figure 5.5, la DNE-2D permet d'augmenter le contraste dans l'image, sans toutefois augmenter le bruit. Cette remarque est en accord avec la conclusion des travaux présentés dans (Mathieu, et al., 2003).

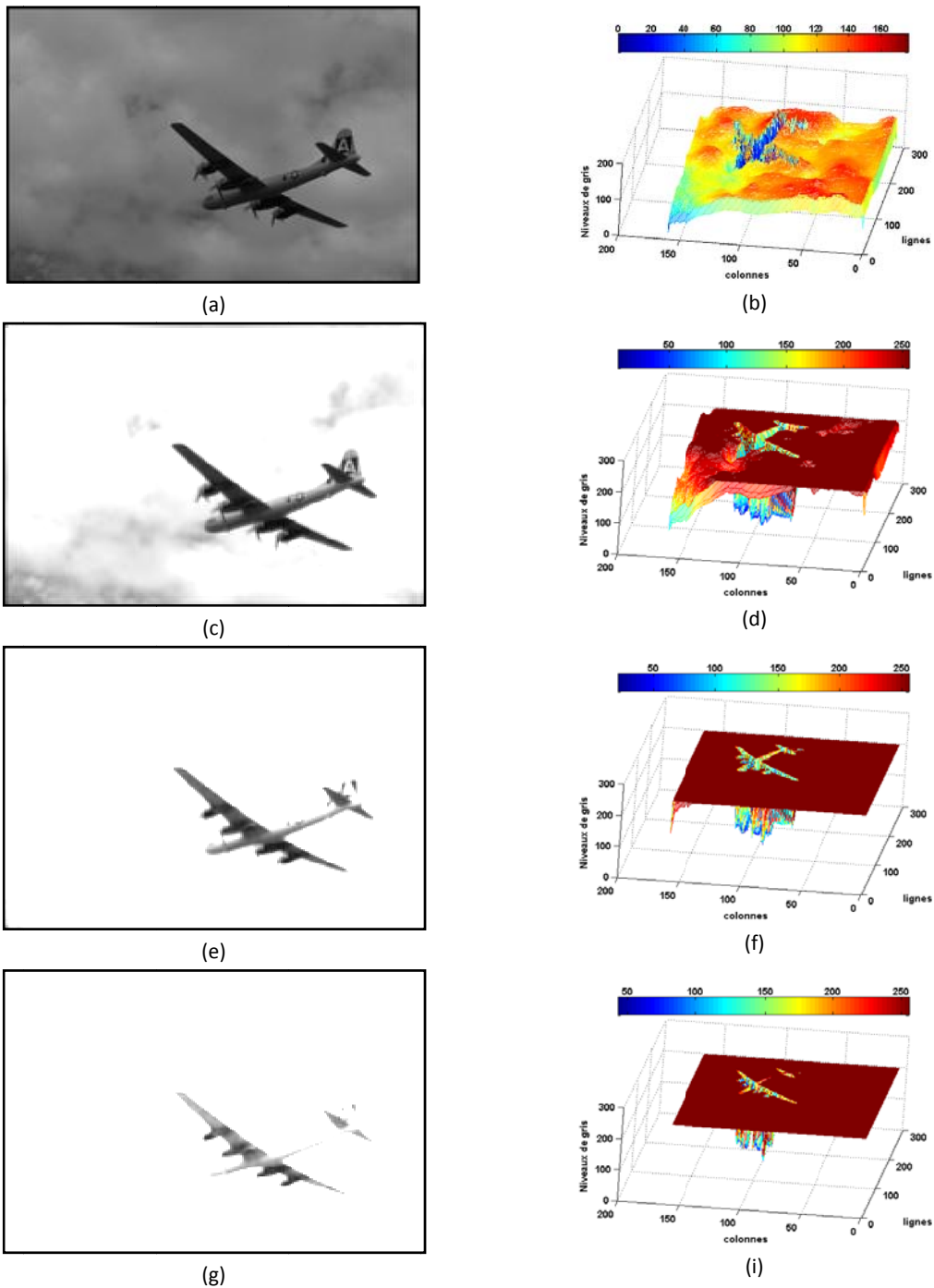


Figure 5. 5: Variations de la dynamique de l'image en fonction de l'ordre de la DNE-2D. (a) Image originale, (b) Variation de la dynamique originale, (c) Image filtrée avec  $\alpha=0,3$ , (d) Variation de la dynamique pour  $\alpha=0,3$ , (e) Image filtrée avec  $\alpha=0,6$ , (f) Variation de la dynamique pour  $\alpha=0,6$ , (g) Image filtrée avec  $\alpha=0,9$ , (h) Variation de la dynamique pour  $\alpha=0,6$ , (g) Image filtrée avec  $\alpha=0,9$ , (d) Variation de la dynamique pour  $\alpha=0,9$ .

### 5.3.3 VALEUR MOYENNE ET AUTOCORRELATION

Les filtres DNE-1D et DNE-2D n'ont pas de valeurs moyennes nulles (Nakib, et al., 2007f; Nakib, et al., 2007g), ce qui nécessite un moyennage de l'histogramme dans le cas de la DNE-1D, et de l'image dans le cas de la DNE-2D.

Dans (Nakib, et al., 2007g) nous avons étudié les variations de l'autocorrélation de l'histogramme quand les valeurs de l'ordre de la DNE-1D varient de -1 à +1. Nous avons remarqué que celle-ci diminue quand les valeurs de  $\alpha$  sont positives. Par contre, elle augmente lorsque les valeurs de  $\alpha$  sont négatives. La même remarque a été constatée lorsque la DNE-2D a été appliquée.

## 5.4 ALGORITHMES DE SEGMENTATION

Nous allons décrire les deux algorithmes de segmentation basés sur la DNE-1D et la DNE-2D. Ces deux algorithmes reposent sur l'hypothèse d'existence d'une dépendance entre la valeur d'un pixel et les valeurs de ses voisins.

### 5.4.1 SEGMENTATION PAR DNE-1D

Dans cette approche, nous appliquons la DNE-1D sur l'histogramme de l'image, avec différents ordres de dérivation. Nous partons de l'hypothèse que les différentes classes de niveaux de gris se traduisent par des pics dans l'histogramme. A partir de là, chaque intervalle entre deux pics de l'histogramme est différencié par un ordre (optimal) qui permet de séparer deux classes successives. Les différentes étapes de l'algorithme sont présentées dans l'Algorithme 5.1.

Algorithme 5.1: Algorithme de segmentation par DNE-1D, dans le cas d'une binarisation.

1. **Calculer** l'histogramme  $h(x)$  de l'image à segmenter.
2. **Calculer** la DNE-1D de  $h(x)$  pour un ordre  $\alpha$ , défini par :  $h^{(\alpha)} = \text{Max}(0, D^\alpha h)$ .
3. **Calculer** le seuil de segmentation :  
 Soit l'ensemble:  $\Omega_\alpha = \{i : i \in [p_1, p_2] \text{ and } h^{(\alpha)}(i) > 0\}$ , où  $p_1$  et  $p_2$  sont les arguments du premier et du dernier pic significatif dans l'histogramme ( $h^{(\alpha)}$ ). Alors le seuil de segmentation  $t_\alpha$  est calculé:  

$$t_\alpha = \mathbb{1}_{\{\alpha \geq 0\}} \times \max \Omega_\alpha + \mathbb{1}_{\{\alpha < 0\}} \times \min \Omega_\alpha$$
 où  $\mathbb{1}_{\{\alpha \geq 0\}}$  est la fonction de Heaviside: égale à 1 si la condition  $\alpha \geq 0$  est satisfaite.
4. **Afficher** les résultats: ordre ( $\alpha$ ), seuil ( $t$ ) et image segmentée.

Dans le cas de la segmentation multiniveau à  $N$  classes, qui correspond à  $N-1$  seuils, nous généralisons l'expression du calcul du seuil, donnée précédemment :

$$\Omega_\alpha^{(j)} = \{i : i \in [p_j, p_{j+1}] \text{ and } h^{(\alpha)}(i) > 0\} \quad (5.8)$$

$$t_\alpha^{(j)} = \mathbb{1}_{\{\alpha \geq 0\}} \times \max \Omega_\alpha^{(j)} + \mathbb{1}_{\{\alpha < 0\}} \times \min \Omega_\alpha^{(j)} \quad (5.9)$$

où les  $p_j$  sont les arguments des pics les plus significatifs dans l'histogramme original.  $j=1, 2, \dots, N-1$  et  $N$  est le nombre de classes. L'algorithme de détection de pics utilisé est celui présenté dans le chapitre précédent (section 4.3.1).

Dans la procédure de segmentation globale (Figure 5.2), l'Algorithme 5.1 est exécuté plusieurs fois, et à chaque fois avec un ordre  $\alpha$  différent. L'objectif est de trouver l'ordre  $\alpha$  optimal, et par là la binarisation optimale recherchée. Il est à préciser que chacun des ordres ne produira pas, certainement, un seuil de segmentation différent. L'ensemble des seuils de segmentation ainsi créés est ensuite traité et réduit par les étages suivants de l'algorithme global. L'étape finale de *sélection* fournira alors le seuil de segmentation optimal recherché en correspondance avec l'ordre optimal de  $\alpha$ .

Pour séparer deux classes, il faut rechercher le seuil de segmentation optimal dans l'intervalle  $[p_j, p_{j+1}]$ . La zone de recherche potentielle est celle où  $D^\alpha h = 0$ . Pour cela, la DNE est appliquée dans l'intervalle  $[p_j, p_{j+1}]$  avec plusieurs ordres  $\alpha$ . Les règles de détermination des seuils ont été élaborées en se basant sur nos expériences.

#### 5.4.2 SEGMENTATION PAR DNE-2D

L'algorithme de segmentation par DNE-2D est très simple et consiste en deux étapes (voir Algorithme 5.2) :

Algorithme 5.2: Algorithme de segmentation par DNE-2D, dans le cas d'une binarisation.

1. **Calculer** le masque de la DNE-2D pour un ordre  $\alpha$ .
2. **Filtrer** l'image originale par le masque de la DNE-2D en tenant compte de l'expression (5.7).
3. **Afficher** les résultats: ordre ( $\alpha$ ), le seuil de segmentation  $t=L-2$  et l'image segmentée.

Le seuil de segmentation est fixé à  $L-2$ , où  $L$  est le nombre total de niveaux de gris. Comme la DNE-2D augmente le contraste sans augmenter le bruit, lorsque l'ordre est optimal, l'objet devient facilement détectable (voir figure 5.5). De par les expérimentations que nous avons effectuées, l'ordre optimal de la DNE-2D se situe dans l'intervalle  $[-1, 0[$ .

#### 5.4.3 PROCEDURE D'OPTIMISATION DE L'ORDRE DE LA DNE-1D ET DE LA DNE-2D

Afin de trouver l'ordre optimal de la DNE et par la suite la segmentation optimale de l'image, nous faisons appel à l'optimisation multiobjectif par l'approche non-Pareto. En nous appuyant sur le schéma de la figure 5.2, nous avons utilisé trois critères : l'entropie de Shannon, la variance interclasse biaisée et la régularité géométrique des images segmentées (figure 5.6).

##### 5.4.3.1 LES CRITERES DE SEGMENTATION UTILISES

Nous avons utilisé les mêmes critères de segmentation que ceux du chapitre précédent:

**La variance interclasse modifiée**

$$MVar(t) = \alpha \cdot \sum_{j=1}^{N_c} \left( \frac{\sigma_{Bet}^2(j)}{\beta_j} + \gamma_j \right) \quad (5.10)$$

où  $t \equiv (t_1, \dots, t_{N-1})$  est le vecteur de seuils. Les paramètres  $\alpha$ ,  $\beta_j$  et  $\gamma_j$  associés à la variance interclasse sont présentés dans la section 3.7.1.1.



### Critère de l'entropie de Shannon

Nous rappelons que l'entropie d'un histogramme d'une image est donnée par:

$$H = -\sum_{i=0}^{255} p_i \log_2 p_i \quad (5.11)$$

Ce critère est plus détaillé dans la section 1.3.2.1.

#### 5.4.3.2 ETAPES D'OPTIMISATION

La procédure de segmentation globale est composée de trois étapes (cf. figure 5.6). Dans la première, nous faisons varier l'ordre de la DNE-1D ou la DNE-2D, afin de produire un ensemble initial de seuils candidats ( $t(\alpha_1, \dots, \alpha_n) = \Psi$ ). Les étapes 2 et 3 permettent de rechercher l'ordre optimal de la DNE et, par là, le seuil de segmentation optimale de l'image. Dans la deuxième étape, nous évaluons les critères de segmentation  $MVar$  et  $H$  séparément, deux nouveaux sous-ensembles de seuils optimaux, inclus dans le premier, sont alors constitués :  $t(\beta_1^*, \dots, \beta_m^*)$  et  $t(\delta_1^*, \dots, \delta_l^*)$  optimisant le premier et le second critère, respectivement, où  $(\delta_i^*, \beta_j^*) \in \Psi \times \Psi$ , pour  $i = 1, \dots, l, j = 1, \dots, m, l \leq n$  et  $m \leq n$ . Finalement, à partir de ces deux ensembles, le seuil optimal final  $t_{(\alpha_{opt})}$  est celui qui optimise le critère de régularité géométrique.

Nous faisons remarquer que, dans le cas de la segmentation multiniveau, les étapes 1 à 3 sont répétées autant de fois qu'il y a de classes à séparer dans l'image.

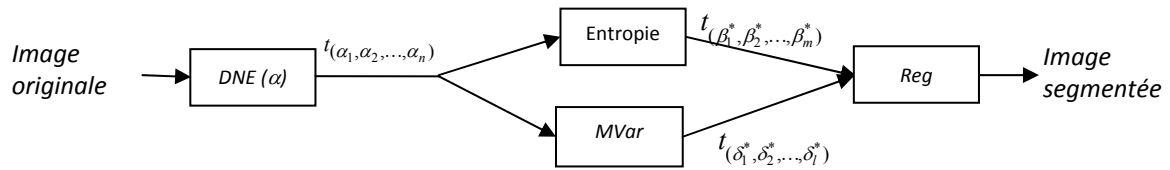


Figure 5. 6: Schéma-bloc de l'algorithme de segmentation proposé basé sur l'approche non-Pareto et la DNE.  $\alpha$  est l'ordre de la DNE-1D ou 2D,  $Reg$  : le critère de régularité géométrique,  $MVar$  : le critère de variance interclasse biaisée.

#### 5.4.3.3 CRITERE DE REGULARITE GEOMETRIQUE

Une nouvelle mesure de la qualité de la segmentation basée sur la mesure de la régularité géométrique d'une image est utilisée pour sélectionner la meilleure image segmentée.

Soit  $f(x)$  une fonction discrète, la mesure de la régularité géométrique de cette fonction sur l'intervalle  $w$  est définie par :

$$g(w) = \|\delta\varphi(x, w)\|_2 \quad w = w_1, w_2, \dots, w_Y \quad (5.12)$$

- $\delta\varphi(x, w) = \varphi(x + w) - \varphi(x - w)$
- $\varphi$  est la dérivée de premier ordre de  $f$ .
- $w$  est la largeur de l'intervalle (fenêtre d'observation) sur lequel l'observation de la régularité est effectuée.
- $Y$  est le nombre de fenêtres d'observation, généralement inférieur à 10.

–  $\|\cdot\|_2$  est l'opérateur de la norme  $L^2$ .

La fonction  $f$  est dite géométriquement régulière si, pour différentes valeurs de  $w$ , la valeur de  $g(w)$  est "faible". Dans notre cas,  $f$  est une ligne de l'image binaire.

La régularité géométrique d'une image, pour différentes valeurs de la fenêtre  $w_i$  et un seuil de segmentation, est donnée par :

$$Reg(t_\alpha) = \|\delta g(w)\|_2 \quad (5.13)$$

où  $\delta$  représente la variation de la fonction.

Nous avons utilisé cette définition de la régularité géométrique afin d'évaluer la qualité de l'image segmentée. Finalement, le seuil optimal est celui qui minimise le critère :

$$t_{\alpha_{opt}} = \underset{t(\alpha_1, \alpha_2, \dots, \alpha_n)}{\text{Min}} Reg(t_\alpha) \quad (5.14)$$

Le calcul de ce critère comporte deux étapes :

Dans la première étape, nous évaluons la régularité des différentes lignes de l'image. Le nombre de lignes dépend de la largeur des objets dans l'image en niveaux de gris, et définit le pas entre deux lignes successives. Généralement, ce nombre est compris entre 2 et 5. L'illustration de cette étape est présentée sur la figure 5.7. Les variations de chaque ligne sont calculées en utilisant l'expression (5.12). Les figures 5.7 (c) et (e) illustrent la forme de deux lignes de l'image binarisée, dans deux cas de segmentation.

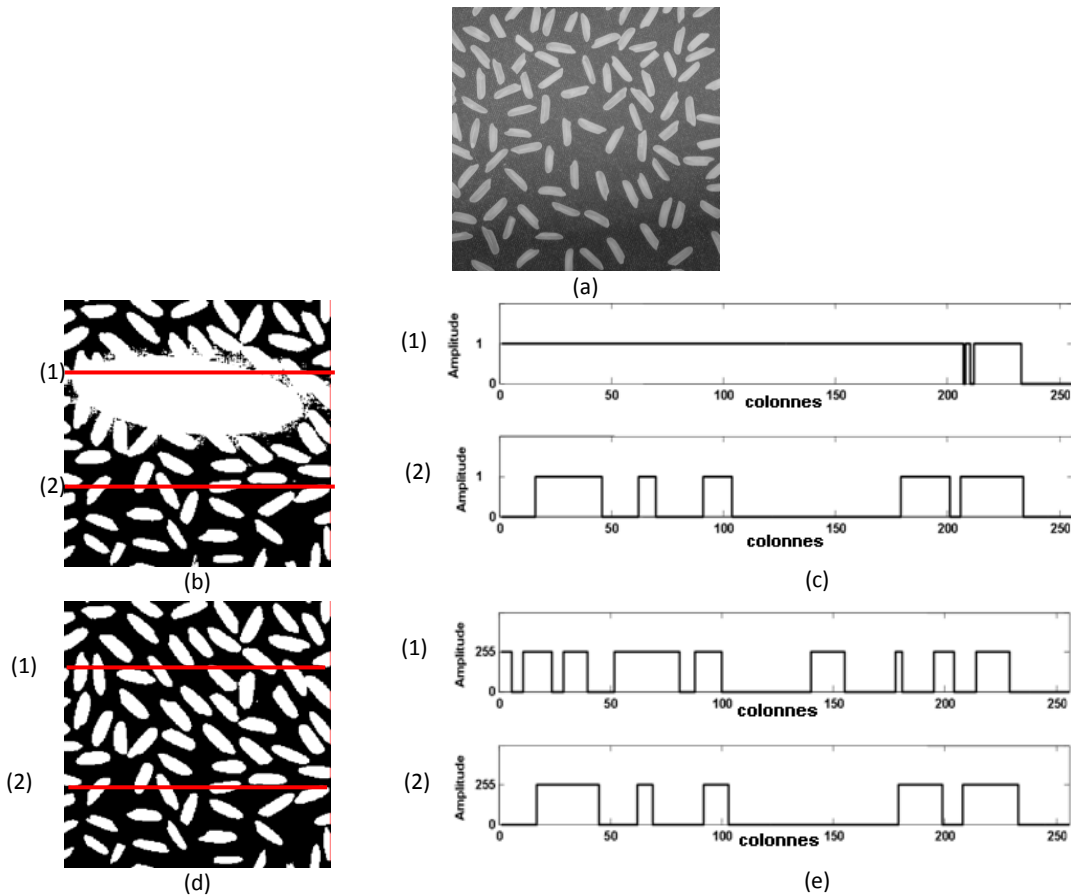


Figure 5.7: Illustration du critère de régularité géométrique. (a) image originale (256x256) pixels, (b) image segmentée avec  $(t=95)$ , correspond à  $Reg = 3,45$ , (c) Représentation 1D des lignes (1) et (2) de la figure (b), (d) image segmentée avec  $(t=106)$ , correspond à  $Reg= 3,05$ , (e) représentation 1D des lignes (1) and (2) de la figure (d).

Dans la seconde étape, nous évaluons les variations de régularité entre les lignes de l'image. La régularité géométrique totale est alors celle définie par l'expression (5.13). Les figures 5.7 (b) et (d) présentent les résultats de segmentation de l'image de la figure 5.7 (a), les valeurs de  $Reg$  sont 3,45 et 3,05. Comme le but est de minimiser le critère ( $Reg$ ), l'image de la figure 5.7 (d) est alors choisie, ce qui correspond à la meilleure segmentation, d'un point de vue visuel.

## 5.5 RESULTATS ET DISCUSSION

---

Dans cette section, nous allons analyser les sensibilités des deux algorithmes par rapport aux variations de l'ordre de la DNE (1D et 2D). Ensuite, nous présentons une comparaison des performances des deux algorithmes sur la segmentation des images synthétiques. Une analyse de la complexité de calcul des deux algorithmes est aussi présentée. Enfin, nous terminons cette section par deux exemples de segmentations.

### 5.5.1 ANALYSE DE LA SENSIBILITE DE L'ORDRE DE LA DNE

---

Dans cette section, nous étudions la sensibilité des deux algorithmes par rapport aux variations de l'ordre de DNE-1D et 2D. Pour cela, nous considérons les images synthétiques de la figure 3.22 (section 3.8).

- **DNE-1D**

Pour analyser la sensibilité de l'algorithme, nous faisons varier l'ordre de la DNE-1D de -1 à +1 avec un pas de 0,01, et traçons les variations du critère  $ME$  défini par l'expression (3.27) (section 3.6.3). Les figures 5.8 (a) à (f) présentent les variations de  $ME$  en fonction des variations de l'ordre de la DNE pour les images des figures 3.22 (B) à (G) (section 3.6.3), respectivement. Il est à signaler que les images synthétiques (B) à (D) sont bruitées avec un bruit multiplicatif, dans ce cas l'ordre optimal de la DNE est dans l'intervalle  $[0, 1]$ , et lorsqu'un bruit blanc gaussien est ajouté (les images (E) à (G)), l'ordre optimal de la DNE-1D se situe dans l'intervalle  $[-1, 0]$ . Nous pouvons remarquer que la qualité de la segmentation est sensible aux variations de l'ordre de  $10^{-2}$  et  $10^{-1}$  lorsque  $\alpha$  est positif et négatif, respectivement.

- **DNE-2D**

Nous avons effectué la même étude dans le cas de la DNE-2D, et avons trouvé une sensibilité de l'ordre de  $10^{-2}$ .

### 5.5.2 COMPARAISON

---

Dans cette section, nous comparons les deux méthodes proposées par rapport aux autres méthodes déjà proposées ainsi qu'aux méthodes de la littérature. Les performances des deux méthodes basées sur la DNE-1D sont présentées dans le tableau 5.1. Ces résultats montrent que la méthode DNE-2D permet d'avoir de meilleures performances que la méthode DNE-1D, surtout lorsque les images sont fortement bruitées. Cependant, la méthode DNE-1D permet d'avoir des performances légèrement supérieures à celles de la DNE-2D, dans le cas d'images faiblement bruitées (image A). En comparant aux autres méthodes de segmentation présentées au tableau 3.4 (section 3.8) et dans la figure 4.5 (section 4.4.3), les performances de la méthode DNE-2D sont supérieures à celles des autres méthodes : la méthode classique d'Otsu, la méthode de Kapur, la méthode EM, la méthode VE, la méthode basée sur l'entropie 2D de Tsallis (TE) et la méthode K-means basée sur la distance euclidienne. Par exemple, dans le cas de l'image G, le score de la DNE-2D est de **8.36%** alors que pour

les autres méthodes : *Otsu* 36,63%, *Kapur* 16,66%, *EM* 36,45%, *VE* 16,63%, *TE* 35,56%, *K-means* 29,98%, *eE2D* 16,09%, *SEE2D* 15,72%, *MO\_seg* 17,07%.

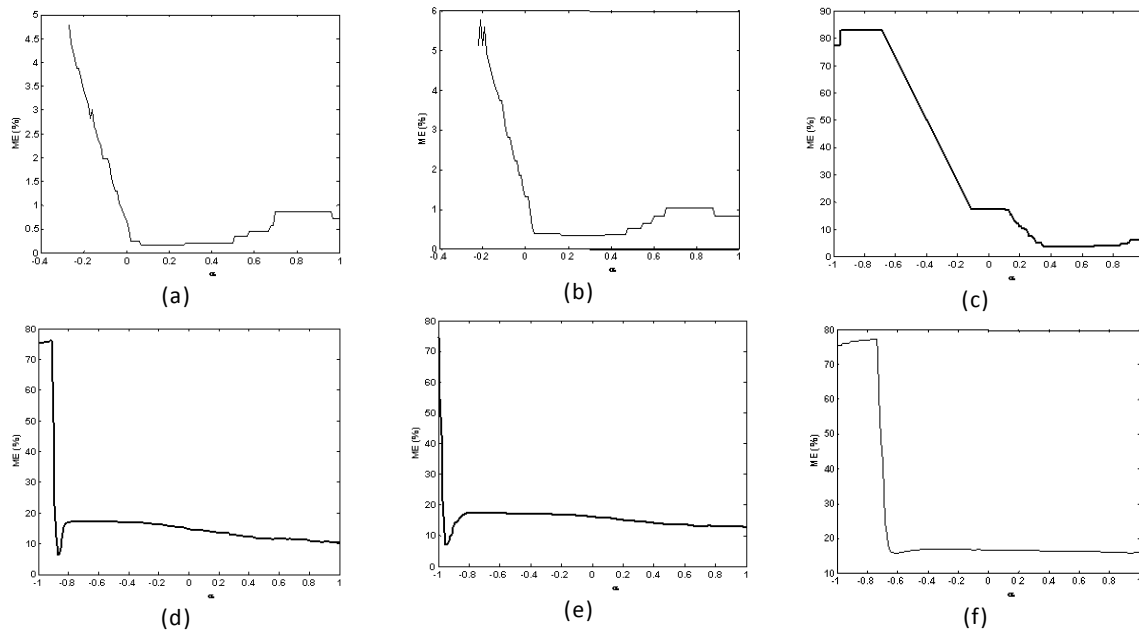


Figure 5. 8: Sensibilité de DNE-1D quand  $\alpha$  varie de -1 à +1. (a) Variations de *ME* dans le cas de l'image synthétique (b), (b) Variations de *ME* dans le cas de l'image synthétique (c), (c) Variations de *ME* dans le cas de l'image synthétique (d), (d) Variations de *ME* dans le cas de l'image synthétique (e), (e) Variations de *ME* dans le cas de l'image synthétique (f), (f) Variations de *ME* dans le cas de l'image synthétique (g). Le pas est 0,01.

Tableau 5. 1: Evaluation des performances des méthodes de segmentation par DNE-1D et DNE-2D.

Images test de la figure 3.22	Méthodes de segmentation			
	Méthode DNE-1D		Méthode DNE-2D	
	$\alpha$	<i>ME</i> (%)	$\alpha$	<i>ME</i> (%)
Image B	0,63	<b>0,17</b>	-0,44	0,26
Image C	0,03	<b>0,39</b>	-0,45	<b>0,32</b>
Image D	0,39	<b>3,30</b>	-0,51	<b>2,84</b>
Image E	-0,83	<b>5,71</b>	-0,44	<b>1,28</b>
Image F	-0,82	<b>6,51</b>	-0,44	<b>1,77</b>
Image G	-0,60	<b>15,78</b>	-0,5	<b>8,36</b>

### 5.5.3 EXEMPLES DE SEGMENTATIONS

- **DNE-1D**

La figure 5.9 présente les résultats de segmentations des images test *maison* et *avion*. Les segmentations en deux classes de ces images sont présentées respectivement sur les figures 5.9 (a) et (b).

Le résultat de la segmentation de l'image *maison* n'est pas optimal, d'un point de vue visuel, nous remarquons quelques points qui apparaissent au coin gauche de l'image ; cependant les contours de la *maison* sont bien préservés. Dans le cas d'une segmentation en 3 classes, les résultats sont présentés sur les figures 5.9 (c) et (d). Les segmentations obtenues de l'image *avion* en deux classes (figure 5.9 (b)) et en trois classes (figure 5.9 (d)) sont de bonne qualité par rapport à celles des méthodes présentées dans le troisième chapitre. D'autres résultats sont présentés dans l'Annexe (section A4).

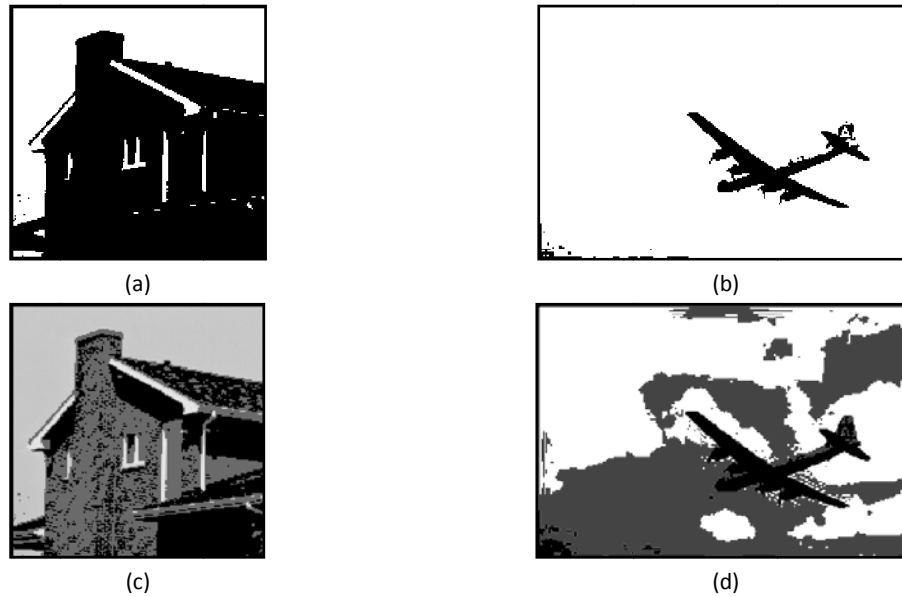


Figure 5. 9: Segmentation des images test *maison* et *avion* par DNE-1D. (a) Segmentation en 2 classes de l'image *maison*,  $t=194$ , avec  $\alpha=-0,08$ , (b) Segmentation en 2 classes de l'image *avion*, avec  $t=67$  ( $\alpha=0,1$ ), (c) Segmentation de l'image *maison* en 3 classes, avec  $t= (106 ; 195)$  ( $\alpha= (0,5 ; -0,1)$ ), (d) Segmentation de l'image *avion* en 3 classes  $t= (77 ; 124)$  ( $\alpha= (-0,1 ; -0,9)$ ).

- **DNE-2D**

Les résultats de la segmentation des images précédentes sont présentés sur la figure 5.10. La segmentation obtenue de l'image *maison* (figure 5. 10 (a)) est de meilleure qualité, comparée à celle de l'approche DNE-1D. La même remarque peut être formulée concernant l'image *avion*. Le cas de la segmentation multiniveau utilisant l'approche DNE-2D est en cours de développement.

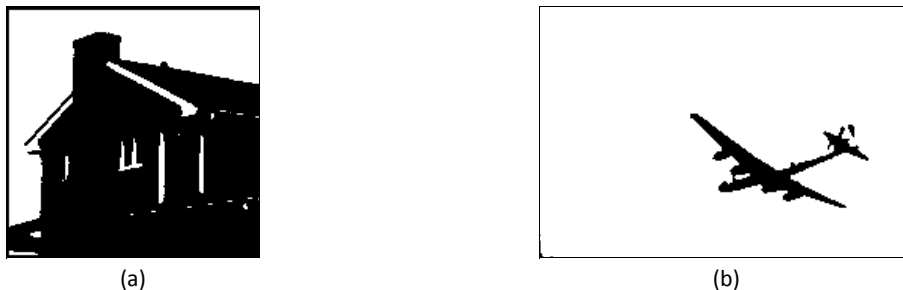


Figure 5. 10: Segmentation des images test *maison* et *avion* par DNE-1D. (a) Segmentation en 2 classes de l'image *maison* avec  $\alpha=-0,1$ , (b) Segmentation en 2 classes de l'image *avion* avec ( $\alpha=-0,62$ ).

#### 5.5.4 ANALYSE DE LA COMPLEXITE ALGORITHMIQUE

##### 5.5.4.1 COMPLEXITE DE L'ALGORITHME DE SEGMENTATION DNE-1D

- La complexité de calcul de l'algorithme de détection de pics est donnée par (Nakib, et al., 2007d) :  $O(N)$ , où  $O(\cdot)$  est la fonction de Landau et  $N = \text{Card}(\arg \max(h \neq 0))$ , où  $h$  est l'histogramme de l'image originale.
- La complexité de calcul de la DNE-1D est donnée par:  $O(c_1.M^2)$ , où  $M \leq 255$  est le nombre de composantes non nulles de  $h$ . Dans le cas d'un seuillage simple:  $M = p_2 - p_1$ , où  $p_2$  et  $p_1$  sont définis dans Algorithme 5.1 et  $c_1$  est le nombre d'ordres de la DNE-1D à évaluer; dans le plus mauvais cas, il est égal à 111.

### 5.5.4.2 COMPLEXITE DE L'ALGORITHME DE SEGMENTATION DNE-2D

La complexité de calcul de la DNE-2D pour un ordre  $\alpha$  est donnée par :  $O(c_2 \cdot S_1)$ , où  $S_1$  est la taille du masque, généralement 3x3, et  $c_2 = S_1 / S$ ,  $S$  est la taille de l'image originale.

### 5.5.4.3 COMPLEXITE DE L'ALGORITHME NON-PARETO

- Le critère de la variance interclasse biaisée nécessite  $O((d-1)(N^2 + S^2))$  opérations, où  $S$  est la taille de l'image et  $d$  le nombre de classes. Dans le cas d'un seuillage simple,  $d=2$ .
- Le critère de l'entropie nécessite:  $O(M)$  opérations.
- La complexité de calcul du critère de régularité géométrique est donnée par:  $O(X \cdot Y)$ , où  $X$  est le nombre de lignes, prises en compte pour évaluer la régularité géométrique de l'image et  $Y$  est le nombre d'intervalles (voir la section 5.3.3.3).

Enfin, la complexité de calcul totale des algorithmes de segmentation par DNE-1D et DNE-2D sont, respectivement:

$$C_{\text{DNE-1D}} = O(c_1 \cdot M^2) + O(N^2 + S^2) + O(M) + O(X \cdot Y) \quad (5. 15)$$

et

$$C_{\text{DNE-2D}} = O(c_1 \cdot c_2 \cdot S_1) + O(N^2 + S^2) + O(N) + O(X \cdot Y) \quad (5. 16)$$

Comme nous pouvons le remarquer, l'algorithme DNE-1D est moins complexe que l'algorithme DNE-2D.

## 5.5.5 APPLICATION AUX IMAGES IRM

Nous avons testé cette approche pour segmenter des images IRM. Les images originales sont présentées sur les figures 3.25 (a) et (b), qui représentent une coupe axiale niveau 11 du cerveau. Les résultats de la segmentation de ces images sont sur les figures 5.11 et 5.12 pour les algorithmes DNE-1D et DNE-2D, respectivement.

Les segmentations des images IRM saine et pathologique, par l'algorithme DNE-1D, dans le cas d'une segmentation en deux classes, sont respectivement sur la figure 5.11 (a) et (b). La segmentation de l'IRM sain en deux classes avec la DNE-1D permet de bien extraire le cortex, contrairement aux ventricules qui ne le sont pas. Cependant, en segmentant cette image en trois classes, ces derniers sont bien extraits. L'IRM pathologique est très bruitée, de ce fait, le cortex est moins bien extrait, seulement les ventricules le sont. La modification de la taille du volume liquide encéphalo-rachidien due à l'atrophie est bien détectée (d'autres résultats sont présentés dans l'Annexe A4).

Dans le cas de la DNE-2D, pour une IRM saine, la segmentation obtenue montre l'efficacité de cet algorithme : le cortex et le volume du liquide encéphalo-rachidien sont bien détectés. Dans le cas de l'image IRM pathologique, seules la taille du liquide encéphalo-rachidien a été extraite, mais avec une plus grande précision que dans le cas de la DNE-1D. Une illustration de l'application de la DNE-2D sur des images microscopiques est présentée dans l'Annexe A5. La segmentation en plusieurs classes, avec cette méthode, est en perspective.

## 5.6 CONCLUSION

Dans ce chapitre, nous avons présenté une approche d'optimisation de critère de segmentation basée sur l'approche multiobjectif non-Pareto. Dans notre approche, nous avons utilisé trois critères

de segmentation : l'entropie de Shannon, la variance interclasse biaisée et un nouveau critère basé sur la régularité géométrique d'une fonction.

Nous avons aussi introduit les DNE-1D et 2D, afin de segmenter des images, tout en réduisant l'espace de recherche des solutions. La première approche est basée sur la dérivation avec un ordre non entier de l'histogramme, tandis que la deuxième est basée sur la dérivation avec un ordre non entier de l'image.

L'application des deux approches sur des images synthétiques a permis d'obtenir de bons résultats, notamment lorsque les images sont fortement bruitées : la DNE-2D surclasse alors toutes les méthodes auxquelles nous l'avons comparée.

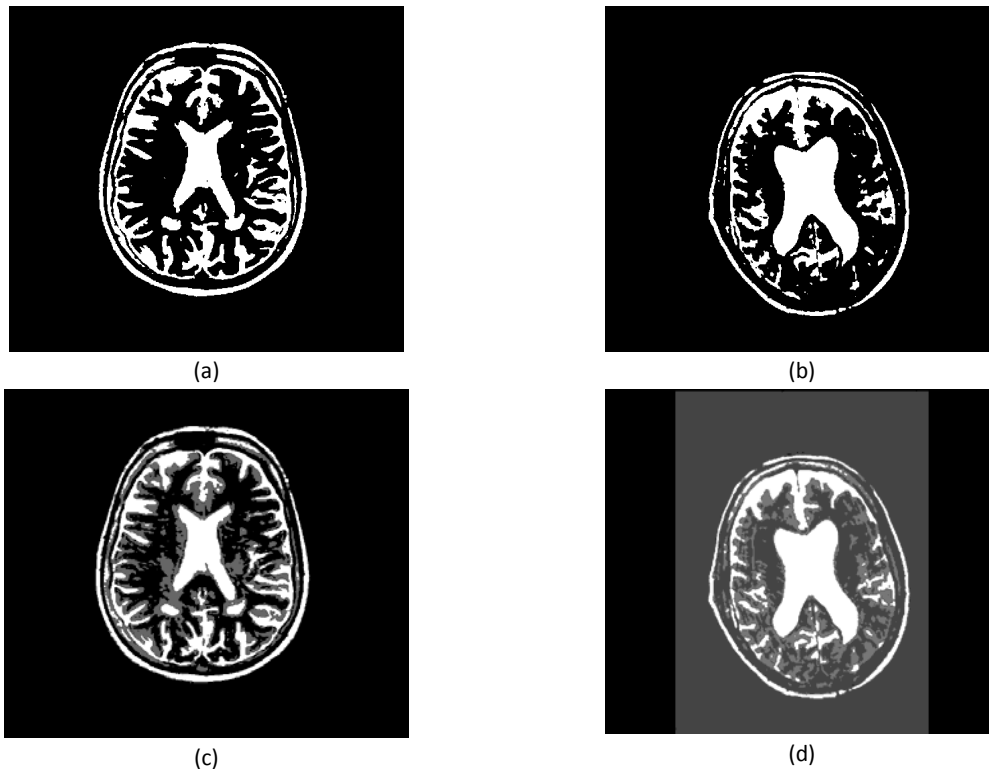


Figure 5. 11:Résultats de segmentations d'images IRM de la figure 3.25 par DNE-1D. (a) Segmentation de l'IRM sain en 2 classes,  $t=78$   $\alpha=-0,5$ , (b) Segmentation de l'IRM pathologique en 2 classes  $t=98$  et  $\alpha=-0,5$ , (c) Segmentation de l'IRM sain en 3 classes  $t= (62; 94)$  et  $\alpha= (-0,5; -0,4)$ , (d) Segmentation de l'IRM pathologique en 3 classes  $t= (74; 115)$  et  $\alpha= (-0,5; -0,1)$ .



Figure 5. 12:Résultats de segmentations d'images IRM de la figure 3.25 par DNE-2D. (a) Segmentation de l'IRM sain en 2 classes,  $\alpha=-0,37$ , (b) Segmentation de l'IRM pathologique en 2 classes, avec  $\alpha=-0,12$ . L'application sur les images IRM a permis d'avoir de bons résultats, dans le cas d'une segmentation en deux classes. Cependant, lorsque le nombre de classes augmente, les performances de notre méthode diminuent.

A l'issue de l'étude de cette approche de segmentation par optimisation multiobjectif non-Pareto, nous avons constaté une augmentation de la complexité de la méthode. De plus, le choix des critères n'est pas facile. Augmenter le nombre de critères permettrait d'augmenter les performances de l'approche. Toutefois, il s'agit de trouver des critères de segmentation qui ne favorisent pas un type particulier d'images.



---

# CHAPITRE 6

## ADAPTATION DE L'APPROCHE PARETO A LA SEGMENTATION

---

### 6.1 INTRODUCTION

---

Ce chapitre est consacré au développement d'une méthode de segmentation basée sur l'optimisation multiobjectif par l'approche Pareto (**SMOP**). Ce type d'approche est déjà présenté dans la section 2.3.3. Le principe consiste à optimiser plusieurs critères de segmentation, et à trouver la solution qui permet d'avoir un compromis entre les différents critères. Pour résoudre ce problème, nous utilisons l'algorithme de référence en optimisation multiobjectif : NSGA-II. C'est la deuxième version de l'algorithme NSGA ("*Non dominated Sorting Genetic Algorithm*") (Deb, et al., 2002). La figure 6.1 illustre le principe de base de la segmentation par optimisation multiobjectif au sens de Pareto.

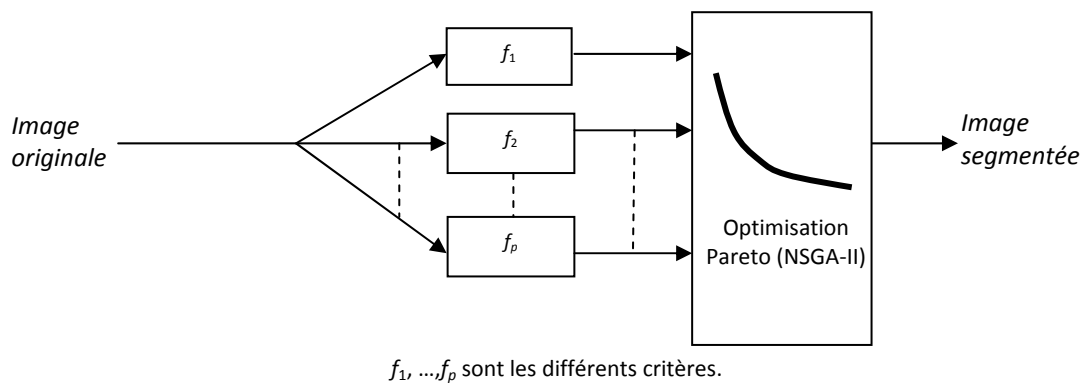


Figure 6. 1: Principe de l'algorithme de segmentation par l'approche Pareto.

Dans la première section de ce chapitre, nous formulons le problème de segmentation d'images par optimisation multiobjectif suivant l'approche Pareto. La deuxième section est consacrée aux différents critères utilisés dans l'algorithme. La métaheuristique utilisée décrite est dans la troisième section. Dans la quatrième section, nous présentons l'algorithme de segmentation. Des résultats de segmentation des images test *maison* et *avion* sont présentés dans la cinquième section. La sixième section est réservée à l'étude comparative de notre méthode avec celles déjà développées et celles issues de la littérature. Enfin, nous présentons, dans la septième section, les résultats de segmentations d'images IRM.

### 6.2 FORMULATION DU PROBLEME

---

Dans le deuxième chapitre, nous évoquons le problème du critère unique de segmentation, ne permettant pas de segmenter tous les types d'images de façon correcte. Dans la suite, nous faisons appel à l'optimisation multiobjectif basée sur l'approche Pareto, afin de contourner ce problème.

Cette approche consiste à utiliser la notion de dominance pour faire converger la population vers un ensemble de solutions efficaces, et sélectionner, en fin d'analyse, la solution optimale, au sens d'un compromis entre les différentes fonctions objectif. Ce concept permet donc d'avoir un ensemble de solutions, dans lequel nous ne retiendrons que la solution de compromis, afin de ne pas favoriser un critère par rapport à un autre.

### 6.3 L'ALGORITHME NSGA-II

Dans NSGA-II, la complexité de calcul a été réduite grâce à la modification de la procédure de tri de la population, en la répartissant en plusieurs fronts. La complexité totale de l'algorithme est de l'ordre  $O(m \cdot N^2)$ , où  $N$  est la taille de la population des solutions et  $m$  le nombre d'objectifs.

Dans la première version de l'algorithme NSGA, une fonction de partage des performances ("*sharing*") était utilisée pour maintenir la diversité. Cependant, cette procédure est fortement consommatrice en temps de calcul. De plus, elle exige le réglage de plusieurs paramètres. Dans NSGA-II, les auteurs ont remplacé la fonction de partage de performance par une fonction de remplacement ("*crowding*"). Le principe consiste à définir deux attributs à chaque individu :

- $i_{rang}$  représente le rang de non-dominance de l'individu. La valeur attribuée à chaque individu dépend de la frontière à laquelle il appartient. Par exemple, les individus du premier front auront la valeur 0, car ils sont non dominés et les individus du second front auront la valeur 1, car ils ne sont dominés que par les individus du premier front, et ainsi de suite.
- $i_{distance}$  définit la situation de l'individu par rapport aux autres et permet d'estimer la densité de population autour de lui.

L'estimation de la densité des solutions au voisinage d'un point  $i$  passe par le calcul de la distance moyenne sur chaque objectif, entre les deux points les plus proches situés de part et d'autre du point  $i$  (figure 6.2). Cet attribut est utilisé comme estimation de la taille du plus large hyper-cube n'incluant que le point  $i$  de la population. La complexité de cette procédure est de  $O(m \cdot N \cdot \log N)$  opérations.

En ce qui concerne l'élitisme, NSGA-II utilise une sélection par tournoi et une nouvelle procédure de passage d'une génération à l'autre. L'attribut  $i_{rang}$  sert à sélectionner, parmi les deux solutions choisies pour le tournoi, celle qui a la plus faible valeur. S'il y a égalité, celle qui a l'attribut  $i_{distance}$  le plus élevé est privilégiée. En d'autres termes, lorsque deux solutions en compétition ont le même rang, celle qui est située dans une région où il y a peu de points est choisie. Cette procédure permet d'avoir un front de Pareto bien étalé.

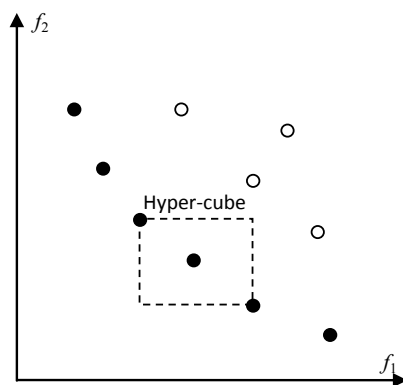


Figure 6. 2: Illustration de la notion de dominance.  $f_1$  et  $f_2$  sont deux fonctions objectifs à minimiser.

La figure 6.3 illustre le principe de fonctionnement de NSGA-II. A chaque itération, une population d'enfants  $Q_t$  de taille  $N$  est engendrée à partir d'une population  $P_t$  de parents de même taille. Les deux populations sont alors combinées dans une population  $R_t$  de taille  $2xN$ . Cette nouvelle population est triée en plusieurs fronts, selon le degré de non-dominance. Une fois la procédure de tri terminée, une nouvelle population  $P_{t+1}$  de taille  $N$  est alors créée à partir des solutions des différents fronts de  $R_t$ . Comme la taille de  $R_t$  est le double de  $P_{t+1}$ , tous les fronts qui ne pourront pas être contenus dans  $P_{t+1}$  sont supprimés. Dans le cas où le nombre de solutions est supérieur au nombre de places, les individus ayant l'attribut  $i_{distance}$  élevé sont choisis, tandis que les autres sont éliminés.

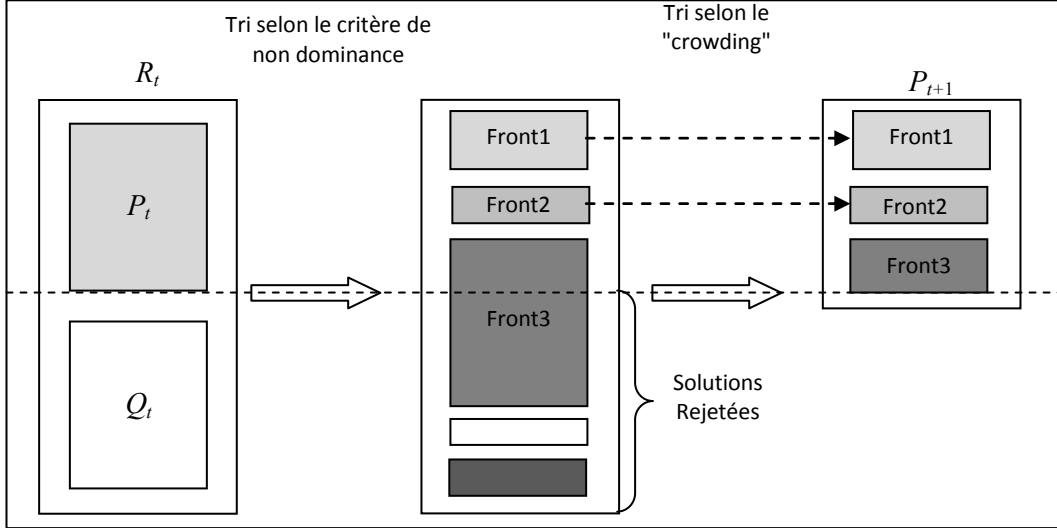


Figure 6. 3: Principe de fonctionnement de NSGA-II.

## 6.4 CRITERES DE SEGMENTATION UTILISES

Nous avons utilisé les trois critères de segmentation suivants :

- **La variance interclasse biaisée**

$$MVar(T) = \alpha \cdot \sum_{j=1}^{N_c} \left( \frac{\sigma_{Bet}^2(j)}{\beta_j} + \gamma_j \right) \quad (6. 1)$$

où  $T \equiv (t_1, \dots, t_{N-1})$  est le vecteur de seuils. Ce critère est présenté dans le chapitre 3, section 7.

- **Le critère de l'entropie de Shannon**

L'entropie, à une dimension, d'un histogramme uni-dimensionnel d'une image est donnée par l'expression suivante:

$$H = - \sum_{i=0}^{255} p_i \log_2 p_i \quad (6. 2)$$

Ce critère est plus détaillé dans la section 1.3.2.1.

- **Le critère de l'entropie à deux dimensions de Shannon et la matrice de cooccurrence**

Le principe de l'entropie 2D est déjà détaillé dans le premier chapitre (section 1.3.2.2), et le calcul du critère de segmentation sous-jacent est défini dans le troisième chapitre (section 3).

Par conséquent, nous ne rappelons ici que la définition de la fonction objectif :

$$H^T[(t_1, s_1), (t_2, s_2), \dots, (t_n, s_n)] = \sum_{i=1}^n H(t_i, s_i) \quad (6.3)$$

où

$$H(s_n, t_n) = - \sum_{i=s_n}^{s_{n+1}-1} \sum_{j=t_n}^{t_{n+1}-1} \frac{P_{ij}}{P_1} \log \frac{P_{ij}}{P_1} \quad (6.4)$$

L'entropie 2D est appliquée à la matrice de cooccurrence des pixels de l'image. Cette matrice est définie comme suit :

Soit  $COO$  une matrice de cooccurrence d'une image  $I$ , un élément de la matrice  $COO(i, j)$  représente le nombre de fois que le niveau de gris  $i$  est adjacent dans le sens horizontal au pixel de niveau de gris  $j$  (Cocquerez, et al., 1995).

Dans l'expression (6.4), les  $p_{ij}$  sont les valeurs normalisées de la matrice  $COO$ . Cette matrice permet de caractériser les textures présentes dans l'image. La figure 6.4 illustre respectivement, en trois dimensions, les matrices de cooccurrence des images tests *maison* et *avion*.

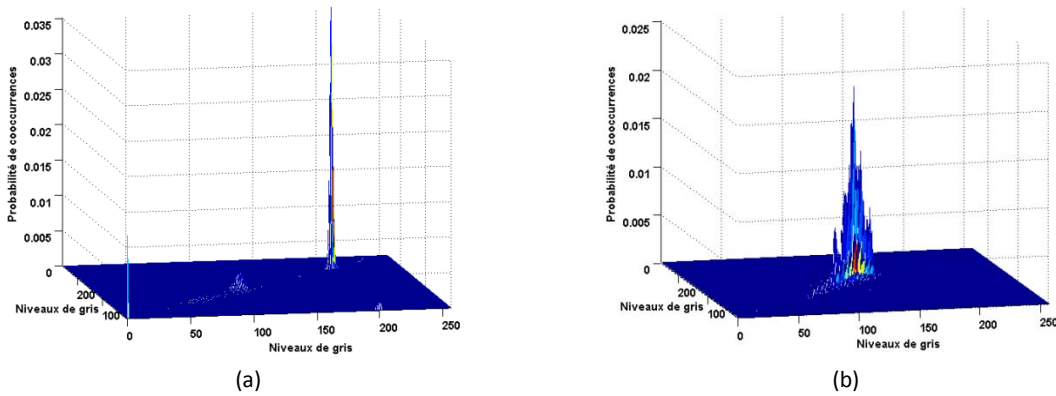


Figure 6. 4: Matrices de cooccurrence des images test : (a) *Maison*, (b) *Avion*.

## 6.5 ALGORITHME DE SEGMENTATION (SMOP)

Tout d'abord, il faut reproduire NSGA-II et l'adapter au problème de la segmentation d'images (Algorithme 6.1). Le choix de cet algorithme de référence s'explique par sa réputation dans le domaine de l'optimisation multiobjectif. De plus, il nécessite peu de réglages de paramètres. Une étude récente (Shukla, et al., 2007) a montré qu'il obtient de bonnes performances sur différents problèmes en multiobjectif.

Le critère d'arrêt que nous avons appliqué repose sur le nombre de générations. Ce nombre, noté  $N_g$ , est calculé ainsi :  $N_g = 5 \cdot N_c$ , où  $N_c$  est le nombre de classes, supposé connu *a priori*. Nous précisons que nous n'avons pas utilisé de croisement.

Nous utilisons un opérateur de sélection par tournoi. Son principe est le suivant : à partir de la population totale (parents et enfants), une moitié de la population est prise de manière aléatoire pour participer au tournoi. La moitié la plus faible des individus participant à ce tournoi est à son tour éliminée, tandis que l'autre est gardée. Nous avons fixé le nombre de tournois à 5.

Algorithme 6.1 : Algorithme SMOP pour la segmentation d'images. Card est la fonction cardinale.

- 1 **Donner** le nombre de classes :  $k$ .
- 2 **Initialiser** aléatoirement la population  $P_0$  de taille  $N = 10 \cdot k$ .
- 3 **Calculer** les différents fronts  $F_i$  de la population  $P_0$ .
- 4 **Sélectionner** dans  $P_0$  et créer  $Q_0$  par l'application des opérateurs génétiques
- 5 **Tant que** critère d'arrêt non satisfait **faire**
  - 5.1 **Créer**  $R_t = P_t \cup Q_t$
  - 5.2 **Calculer** les différents fronts  $F_i$  de la population  $R_t$
  - 5.3 **Mettre**  $P_{t+1} = \emptyset$  et  $i = 1$
  - 5.4 **Tant que**  $Card(P_{t+1}) < N$  **faire**
    - 5.4.1 **Inclure** tous les individus de  $F_i$  dans  $P_{t+1}$
    - 5.4.2 **Incrémenter**  $i : i = i + 1$
  - 5.5 **Fin Tant que**
- 6 **Inclure** les  $(N - Card(P_{t+1}))$  individus de  $F_i$  les mieux répartis dans le de  $i_{distance}$ .
- 7 **Créer**  $Q_{t+1}$  à partir de par  $P_{t+1}$  l'application des opérateurs génétiques.
- 8 **Fin Tant que**
- 9 **Afficher** les résultats: seuil ( $t$ ) et image segmentée.

## 6.6 RESULTATS ET DISCUSSION

La figure 6.5 présente les résultats de segmentations des images test *maison* et *avion*. Les segmentations en deux classes de ces images sont présentées sur les figures 6.5 (a) et (b), respectivement. D'un point de vue visuel, la segmentation de l'image *maison* est de bonne qualité. De plus, les contours de la *maison* sont bien préservés. Dans le cas de l'image *avion*, le résultat de la segmentation est équivalent à celui de la méthode DNE-1D. Les segmentations en trois classes sont sur les figures 6.5 (c) et (d). Dans ce cas, la segmentation de l'image *maison* est également d'assez bonne qualité, toujours du point de vue visuel. Pour l'image *avion*, en revanche, la segmentation est moins bonne : certains points censés appartenir au fond, au coin gauche de l'image, sont rangés dans la même classe que les pixels qui composent l'*avion*.

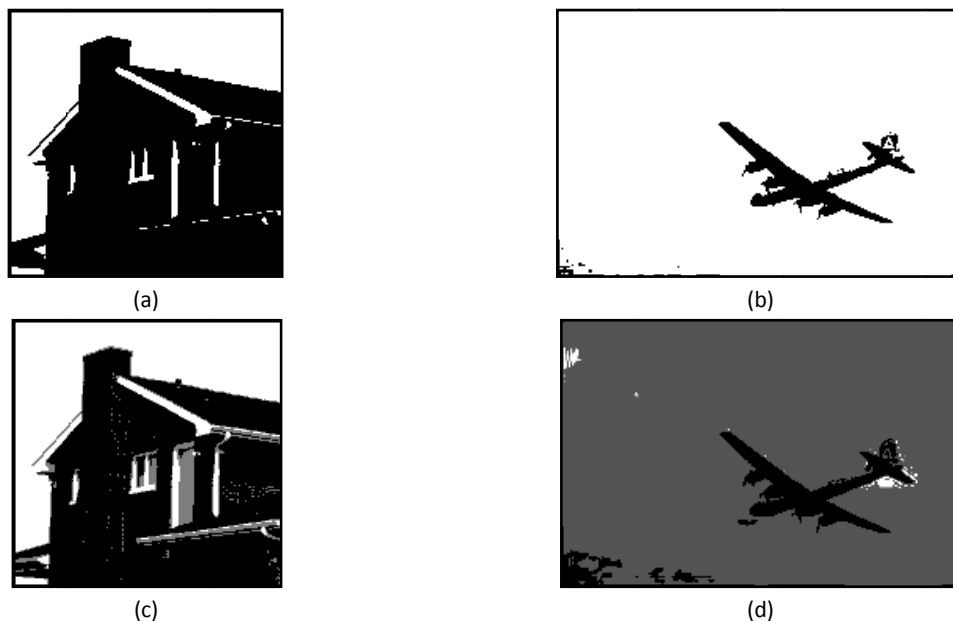


Figure 6. 5: Segmentation des images test *maison* et *avion* par SMOP. (a) Segmentation en 2 classes de l'image *maison*,  $t=185$ , (b) Segmentation en 2 classes de l'image *avion*, avec  $t=67$ , (c) Segmentation de l'image *maison* en 3 classes avec  $t=(125 ; 185)$ , (d) Segmentation de l'image *avion* en 3 classes,  $t=(77 ; 124)$ .

Les figures 6.6 (a) et (b) illustrent les fronts de Pareto en trois dimensions, dans le cas de la segmentation en trois classes des images *maison* et *avion*. Ces résultats sont la moyenne de 20 exécutions successives de l'algorithme SMOP.

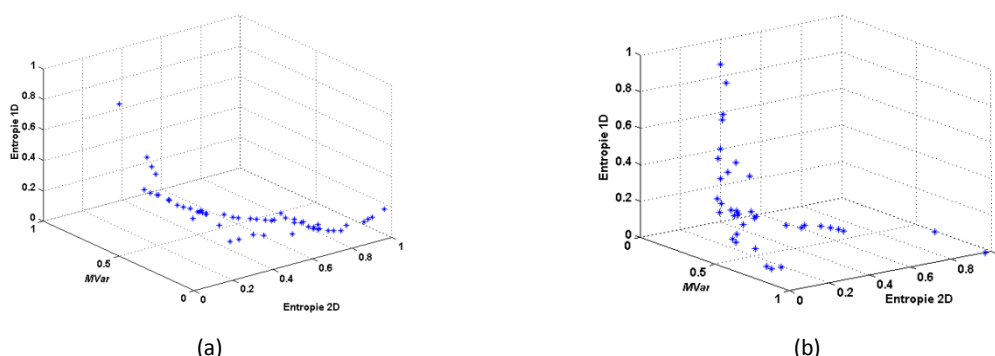


Figure 6. 6: Fronts de Pareto trouvés par l'algorithme SMOP pour la segmentation en 3 classes des images : (a) *Maison*, (b) *Avion*.

## 6.7 COMPARAISON DES PERFORMANCES

Pour évaluer les performances des méthodes, nous utilisons la méthode "vérité terrain" basée sur le taux de pixels bien classés. Cette méthode implique l'utilisation d'images test bruitées dont l'originale est connue *a priori*. Nos images test sont celles de la figure 3.20. Pour quantifier les résultats de la segmentation, le critère défini par l'expression (3.26) est utilisé.

Les performances obtenues (tableau 6.1) montrent que l'algorithme SMOP converge vers le meilleur résultat lorsque les images sont faiblement bruitées. En revanche, dans le cas des images fortement bruitées, SMOP ne produit pas de bons résultats, à cause des critères utilisés. Cependant, ces résultats restent tout de même assez proches de ceux des autres algorithmes.

Tableau 6. 1: Performances de l'algorithme SMOP. Les faibles valeurs de *ME* sont en gras.

Images test de la figure 3.22	Méthodes de segmentation						
	Otsu <i>ME</i> (%)	Kapur <i>ME</i> (%)	<i>K-means</i> <i>ME</i> (%)	EM <i>ME</i> (%)	VE <i>ME</i> (%)	TE <i>ME</i> (%)	SMOP <i>ME</i> (%)
Image B	0,45	5,61	<b>0,17</b>	8,68	0,34	0,64	<b>0,17</b>
Image C	0,88	4,50	0,61	12,50	0,63	1,12	<b>0,51</b>
Image D	12,22	4,97	10,86	28,87	11,59	12,90	<b>3,42</b>
Image E	11,92	5,77	11,84	<b>5,74</b>	5,90	5,92	6,93
Image F	13,82	6,74	14,92	<b>6,60</b>	13,22	6,68	6,66
Image G	36,63	16,66	36,45	16,63	35,56	29,98	16,68

## 6.8 APPLICATION DE LA METHODE SMOP AUX IMAGES IRM

Après avoir évalué les performances de SMOP sur des images synthétiques, nous l'avons appliqué sur des images IRM, présentées sur les figures 3.25 (a) et (b). Elles représentent une coupe axiale (niveau 11) du cerveau. Les résultats de la segmentation de ces images sont sur les figures 6.7 (a) à (d).

Les segmentations en deux et trois classes de l'image IRM saine sont sur les figures 6.7 (a) et (c), respectivement. Dans les deux cas, on peut remarquer que le cortex et le volume du liquide encéphalo-rachidien sont bien détectés.

Pour l'IRM pathologique, la modification de la taille du volume du liquide encéphalo-rachidien due à l'atrophie est bien détectée, aussi bien avec la segmentation en deux classes qu'en trois classes. En revanche, le cortex est moins bien extrait. D'autres résultats sont présentés dans l'Annexe A6.

## 6.9 CONCLUSION

Dans ce chapitre, nous avons présenté une méthode d'optimisation multiobjectif par l'approche Pareto, et l'avons appliquée à la segmentation des images. Afin de résoudre ce problème multiobjectif, nous avons fait appel aux algorithmes génétiques, via l'algorithme NSGA-II.

L'adaptation de NSGA-II au problème de la segmentation d'images est très aisée, du fait du faible nombre de paramètres à régler. Nous avons adapté l'algorithme en modifiant le critère d'arrêt et en automatisant le choix de la taille de la population de départ et le nombre de générations.

Les difficultés de cette approche sont, d'une part, la détermination des critères contradictoires et, d'autre part, le choix final de la solution Pareto optimale, parmi toutes les solutions présentées.

Cependant, cette approche multiobjectif nous a permis de penser à un schéma regroupant plusieurs critères de segmentation et faisant appel à un système expert, afin de sélectionner les critères à utiliser parmi ceux disponibles, compte tenu du type d'image à segmenter.



Figure 6. 7: Résultats de segmentation par SMOP d'IRM de la figure 3.25. (a) Segmentation de l'IRM saine en 2 classes,  $t=90$ , (b) Segmentation de l'IRM pathologique en 2 classes,  $t=101$ , (c) Segmentation de l'IRM saine en 3 classes,  $t=(43; 95)$ , (d) Segmentation de l'IRM pathologique en 3 classes,  $t=(35; 100)$ .

---

## CONCLUSIONS ET PERSPECTIVES

---

Dans cette thèse, nous avons abordé la problématique de la segmentation des images au moyen des métaheuristiques; celles-ci présentant l'avantage de s'adresser à n'importe quel problème d'optimisation. La première étape de notre étude a donc porté sur la reformulation du problème de la segmentation d'images en un problème d'optimisation. Deux solutions ont été proposées : la première relève de l'optimisation mono-objectif, et la seconde de l'optimisation multiobjectif. Pour mieux segmenter les images, nous avons aussi proposé de nouveaux critères de segmentation, dont le critère de mesure de la régularité géométrique d'une segmentation, dans le cas de l'approche non Pareto. L'évaluation des résultats d'une segmentation est difficile, du fait que sa qualité dépend des traitements en amont, car il peut exister plusieurs façons de segmenter une image. Donc, pour avoir une idée aussi précise que possible de la performance de nos algorithmes, nous avons opté pour une méthode "vérité terrain", basée sur le calcul de l'erreur de classification. On obtient cette erreur en comparant une image synthétique originale non bruitée et connue *a priori* à l'image segmentée d'une copie plus au moins dégradée. Enfin, L'ensemble de nos algorithmes a aussi été testé sur des images diverses, bien qu'un accent particulier ait été mis sur des images IRM du cerveau. L'objectif était de segmenter ces images de manière à bien extraire les ventricules d'un sujet atteint d'une atrophie, et par suite de permettre d'évaluer leurs variations de volume par rapport à un sujet sain.

Dans la première approche de la segmentation d'images, basée sur l'optimisation en mono-objectif, nous avons proposé plusieurs solutions pour adapter les métaheuristiques au problème de la segmentation. Nous précisons que nous avons choisi de ne pas appliquer ces métaheuristiques sur les critères qui ont déjà été abordés dans la littérature.

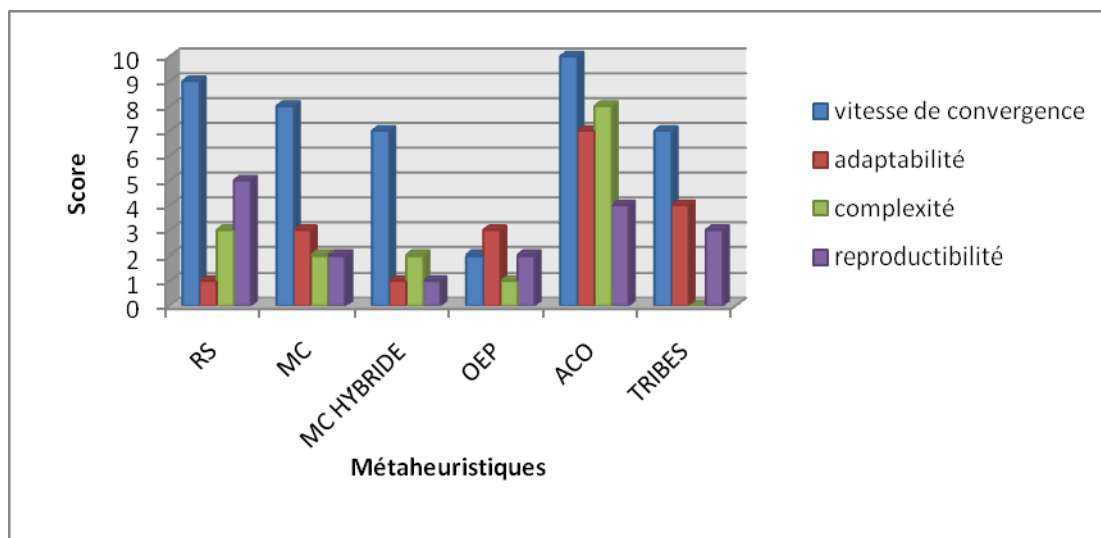
La première métaheuristique expérimentée a été le recuit simulé. Celui-ci a été adapté aux problèmes à variables continues, afin d'approcher les histogrammes des images à l'aide de courbes continues suivant un modèle gaussien. La seconde métaheuristique que nous avons adaptée est le recuit microcanonique. Après les réglages et modifications nécessaires, cette métaheuristique a été hybridée avec un simplex (NM-MC), de sorte à maximiser un nouveau critère de segmentation, appelé entropie exponentielle à deux dimensions (eE2D). Ce critère a aussi été utilisé pour segmenter des images avec un algorithme d'optimisation par essaim particulaire (OEP) sans paramètres, appelé TRIBES. Nous avons aussi révisé l'algorithme OEP, afin de l'appliquer à un deuxième nouveau critère de segmentation, appelé entropie survivante exponentielle à deux dimensions (2DSEE). Enfin, nous avons adapté l'algorithme d'optimisation par colonies de fourmis pour maximiser un troisième nouveau critère, appelé la variance intraclasse biaisée.

Afin d'avoir une idée aussi claire que possible de la pertinence de toutes ces méthodes, nous avons effectué une étude comparative à partir de quatre paramètres : l'adaptabilité de la méthode, la vitesse de convergence, la complexité, et la reproductibilité des résultats. Sur une échelle s'étendant de 0 à 10, nous avons affecté un score à chaque méthode. La figure ci-dessous illustre les résultats de



cette étude comparative. Ces résultats montrent que l'OEP permet d'avoir le meilleur compromis en termes de vitesse de convergence, d'adaptabilité, de complexité, et de reproductibilité. En conséquence, l'OEP est la métaheuristique d'optimisation la mieux adaptée pour résoudre le problème de la segmentation d'images en mono-objectif. Contrairement à l'algorithme d'optimisation par colonies de fourmis, qui s'avère le moins performant. Quant à l'algorithme recuit microcanonique hybridé avec l'algorithme du simplexe, il peut être utile dans le cas d'une application qui nécessite une grande précision. L'algorithme TRIBES a une faible complexité; cependant nous regrettons sa mauvaise reproductibilité des solutions et sa convergence lente. Le recuit simulé possède une complexité plus élevée que TRIBES; cependant, ses performances sont comparables à ce dernier.

Toutefois, nous conseillons de juger ces résultats avec prudence, car ils peuvent être entachés d'une part de subjectivité, du fait qu'ils découlent essentiellement de notre propre expérience de l'adaptabilité de toutes ces métaheuristicues au problème de la segmentation d'image.



Comparaison des métaheuristicues en mono-objectif pour résoudre le problème de la segmentation des images.

Dans la seconde approche, basée sur l'optimisation multiobjectif, nous avons exploré trois directions: l'approche par agrégation de critères, l'approche non-Pareto et l'approche Pareto. Dans l'approche par agrégation de critères, nous avons testé un critère de segmentation paramétrique et deux critères non-paramétriques, optimisés par un recuit simulé continu et un recuit simulé adaptatif. Cette agrégation nous a permis d'élargir le champ d'application de la méthode paramétrique et d'améliorer la qualité de la segmentation.

Dans la seconde direction, nous avons utilisé la dérivation d'ordre non-entier, afin de réduire l'espace de recherche des solutions. Dans cette approche, nous avons expérimenté parallèlement deux critères de segmentation non-paramétriques, suivis d'un nouveau critère d'évaluation de la qualité de la segmentation. Ce dernier est basé sur la régularité géométrique des images segmentées. Cette approche produit des résultats de segmentation beaucoup plus intéressants, en grande partie grâce à la dérivation non-entière.

Dans la troisième direction, nous avons utilisé l'algorithme génétique de référence en optimisation multiobjectif, NSGA-II, afin d'optimiser plusieurs critères simultanément. Dans ce cas, nous avons utilisé trois critères de segmentation, ce qui permet d'obtenir de bons résultats de segmentation. A travers cette méthode, nous avons montré que l'approche multiobjectif au sens de Pareto permet

d'avoir les meilleurs résultats. Cependant, le choix des critères de segmentation n'est pas évident, du fait de leur nombre et de leur diversité.

Les perspectives de ce travail sont nombreuses, aussi bien en pratique qu'en théorie. A titre d'exemple, il serait intéressant d'étendre l'algorithme de segmentation via la dérivation non-entière en deux dimensions au problème de la segmentation en plusieurs classes.

Aussi, il est possible d'améliorer les résultats de la segmentation en hybridant nos algorithmes au sein d'une approche coopérative globale contours-régions. Une piste possible pour étendre nos algorithmes à l'approche contours consiste à initialiser un algorithme basé sur les courbes de niveaux avec un algorithme de seuillage, afin de bien placer les contours de départ et, par ricochets, accélérer la convergence.

En ce qui concerne la segmentation d'images par l'approche Pareto, il serait intéressant de concevoir un système expert qui permettrait de sélectionner des critères de segmentation, parmi un ensemble prédéfini, en fonction de leur type et de l'application en amont. Cette procédure suppose l'existence d'une base de critères bien classés, en relation avec une banque indexée d'images prototypes, regroupées de préférence en familles.

Enfin, une perspective intéressante reste l'application des métaheuristiques à la segmentation de séquences d'images. Ceci revient à résoudre un problème d'optimisation dynamique.

---

# ANNEXES

## DIVERS RESULTATS DE SEGMENTATIONS D'IMAGES A L'AIDE DE NOS DIVERS ALGORITHMES

---

### A1. EXEMPLES DE SEGMENTATIONS PAR RECUIT MICROCANONIQUE HYBRIDE

---

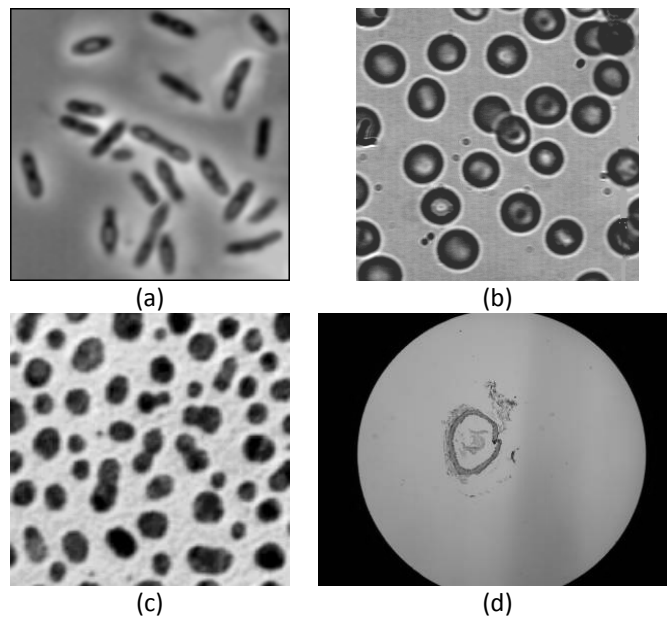


Figure A1. 1 images test originales : (a) Bactéries, (b) Sang, (c) cellules, (d) Vaisseau.

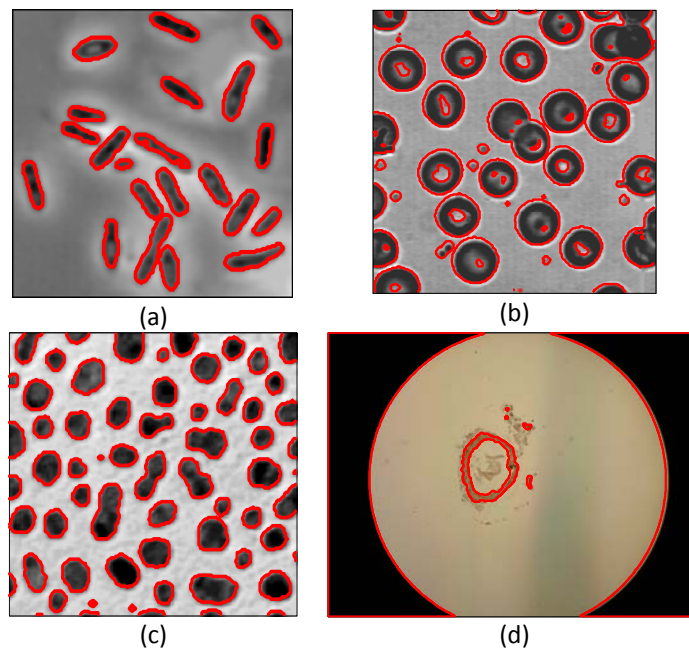


Figure A1. 2: Résultats de segmentation obtenus par la méthode NM-MC: (a) bactéries (t=98), (b) sang (t=131), (c) cellules (t=141), (d) Artère (t=70)

## A2. EXEMPLES DE SEGMENTATIONS PAR LA METHODE 2DSEE

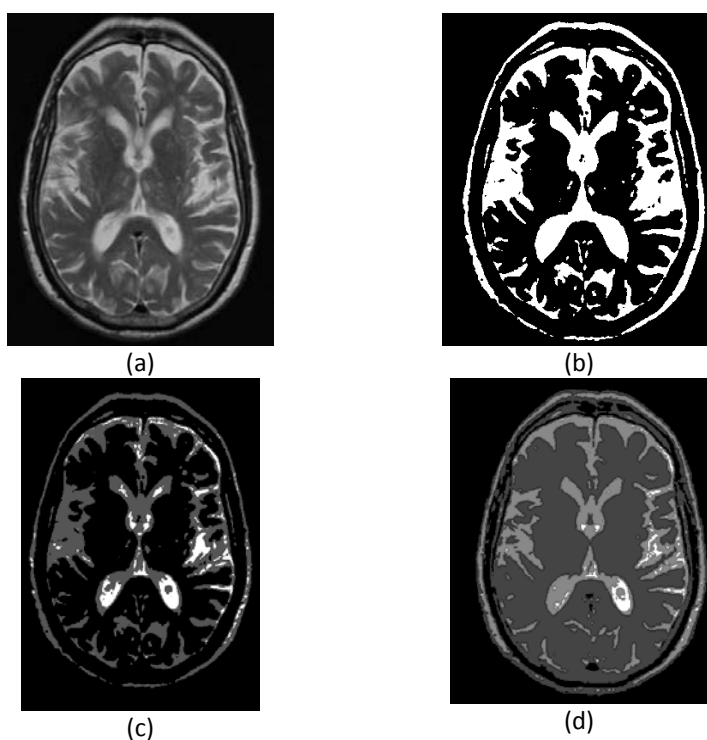


Figure A2.1: Segmentation d'une image IRM d'un sujet atteint d'une atrophie au niveau des ventricules et au niveau du cortex. (a) Image IRM originale pathologique, (b) Segmentation en 2 classes:  $\alpha = 0,6$ ;  $t=98$ , (c) Segmentation en 3 classes  $\alpha=0,8$  ;  $t=(99 ; 169)$ , (d) Segmentation en 4 classes  $\alpha=0,8$   $t=(35 ; 120 ; 179)$ .

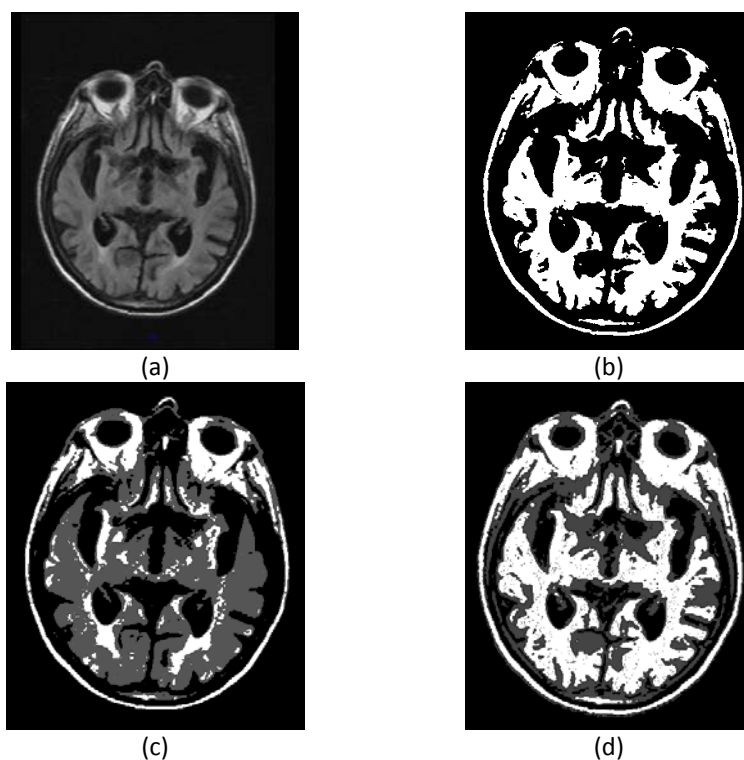


Figure A2.2: Segmentation d'une image IRM présentant une atrophie des ventriculaire. (a) Image IRM originale pathologique, (b) Segmentation en 2 classes:  $\alpha = 0,8$ ;  $t=87$ , (c) Segmentation en 3 classes  $\alpha=0,9$  ;  $t=(64 ; 110)$ , (d) Segmentation en 4 classes  $\alpha=0,8$  ;  $t=(37 ; 90 ; 110)$ .

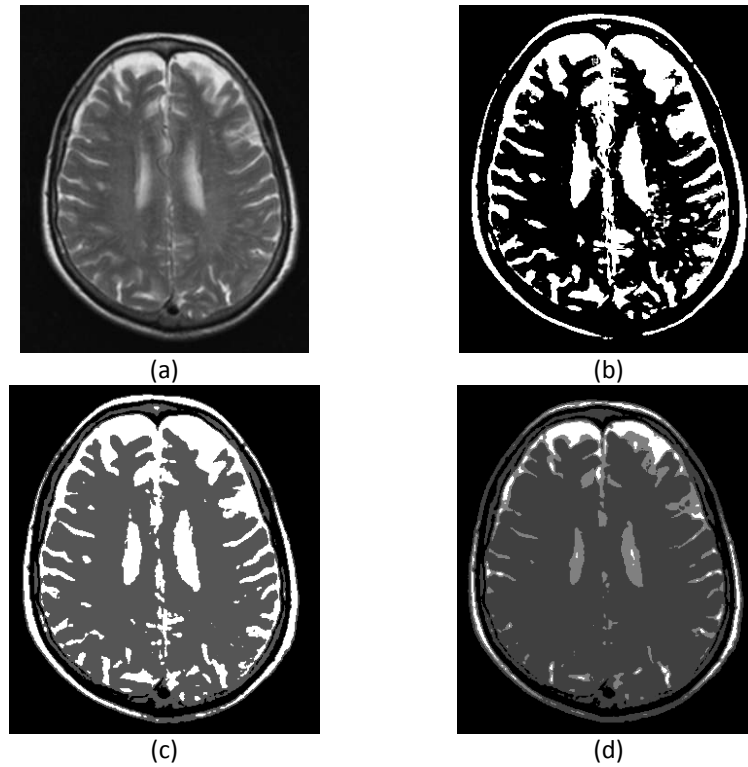


Figure A2.3: Segmentation d'une image IRM présentant une atrophie faciale. (a) Image IRM originale pathologique, (b) Segmentation en 2 classes:  $\alpha=0,8$ ;  $t=97$ , (c) Segmentation en 3 classes  $\alpha=0,8$  ;  $t=(57 ; 169)$ , (d) Segmentation en 4 classes  $\alpha=0,8$  ;  $t=(35 ; 120 ; 179)$ .

### A3. EXEMPLES DE SEGMENTATIONS PAR LA METHODE CCS

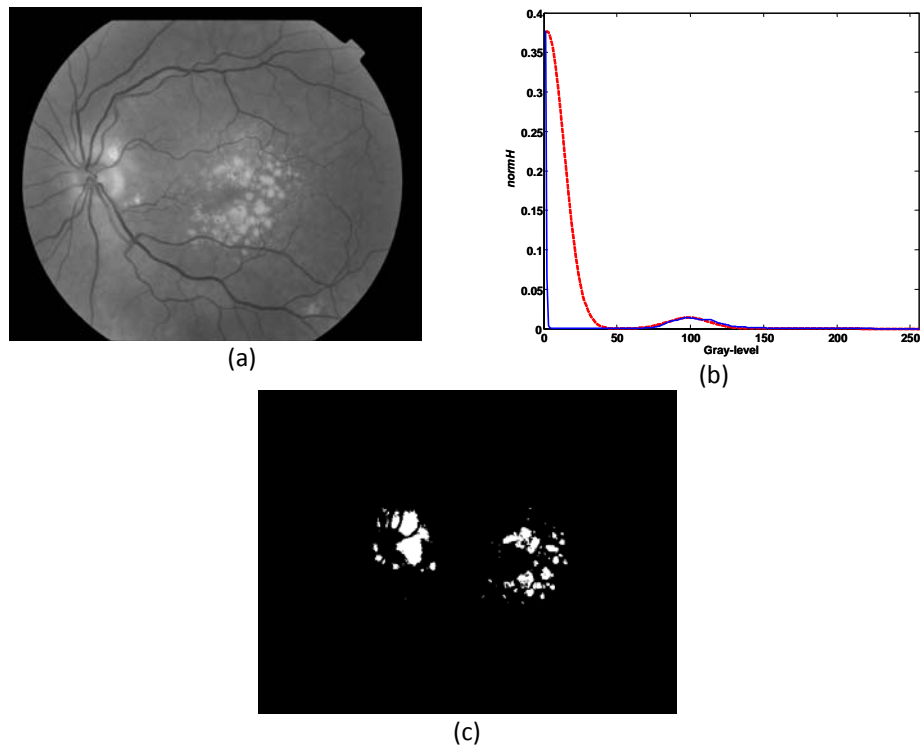


Figure A3. 1: Segmentation d'une image d'angiographie retienne présentant la pathologie des drusens. (a) Image originale pathologique, (b) Histogramme original et son approximation, (c) Segmentation en 2 classes,  $t=150$ .



Figure A3.2: Segmentation d'une image IRM cérébrale de ventricules. (a) Image originale, (b) Image segmentée en 3 classes,  $t=(134 ; 187)$ .

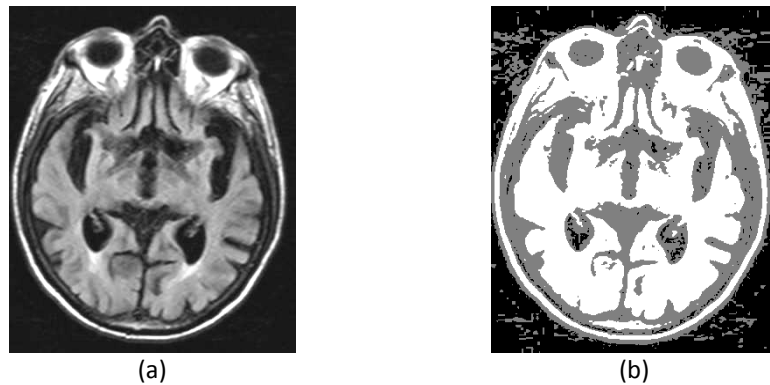


Figure A3.3: Segmentation d'une image IRM cérébrale. (a) Image originale, (b) Image segmentée en 3 classes  $t=(28 ; 108)$ .

#### A4. EXEMPLES DE SEGMENTATIONS PAR LA METHODE DNE-1D

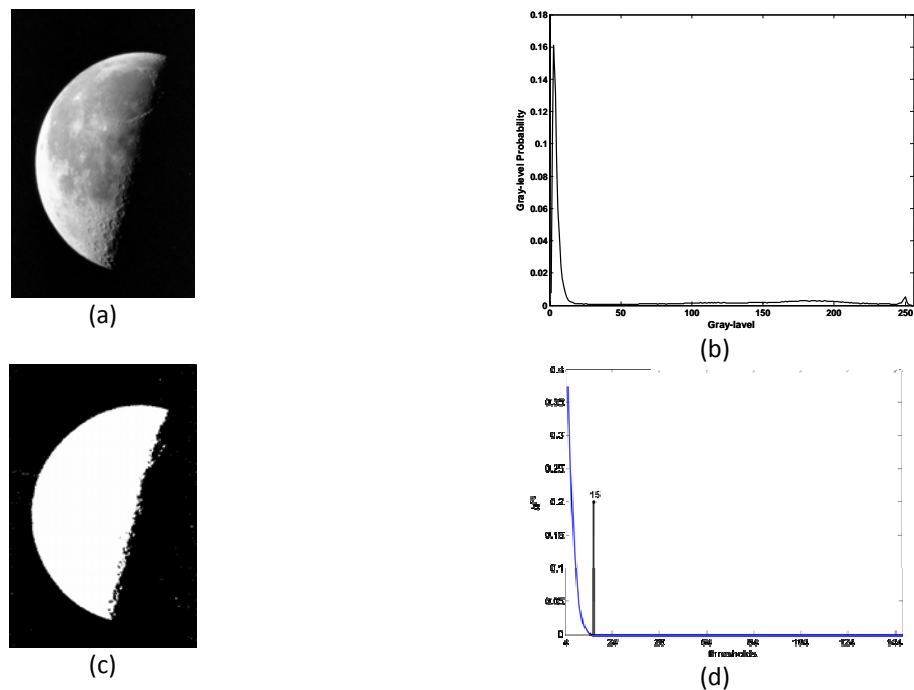


Figure A4.1: Illustration de la performance de l'algorithme DNE-1D avec un ordre optimal. (a) Image originale *moon*, (b) Histogramme original, (c) Image segmentée par DNE-1D ( $t=15$ ,  $\alpha=0.1$ ), (d) Histogramme différencié, avec  $\alpha=0,1$ .

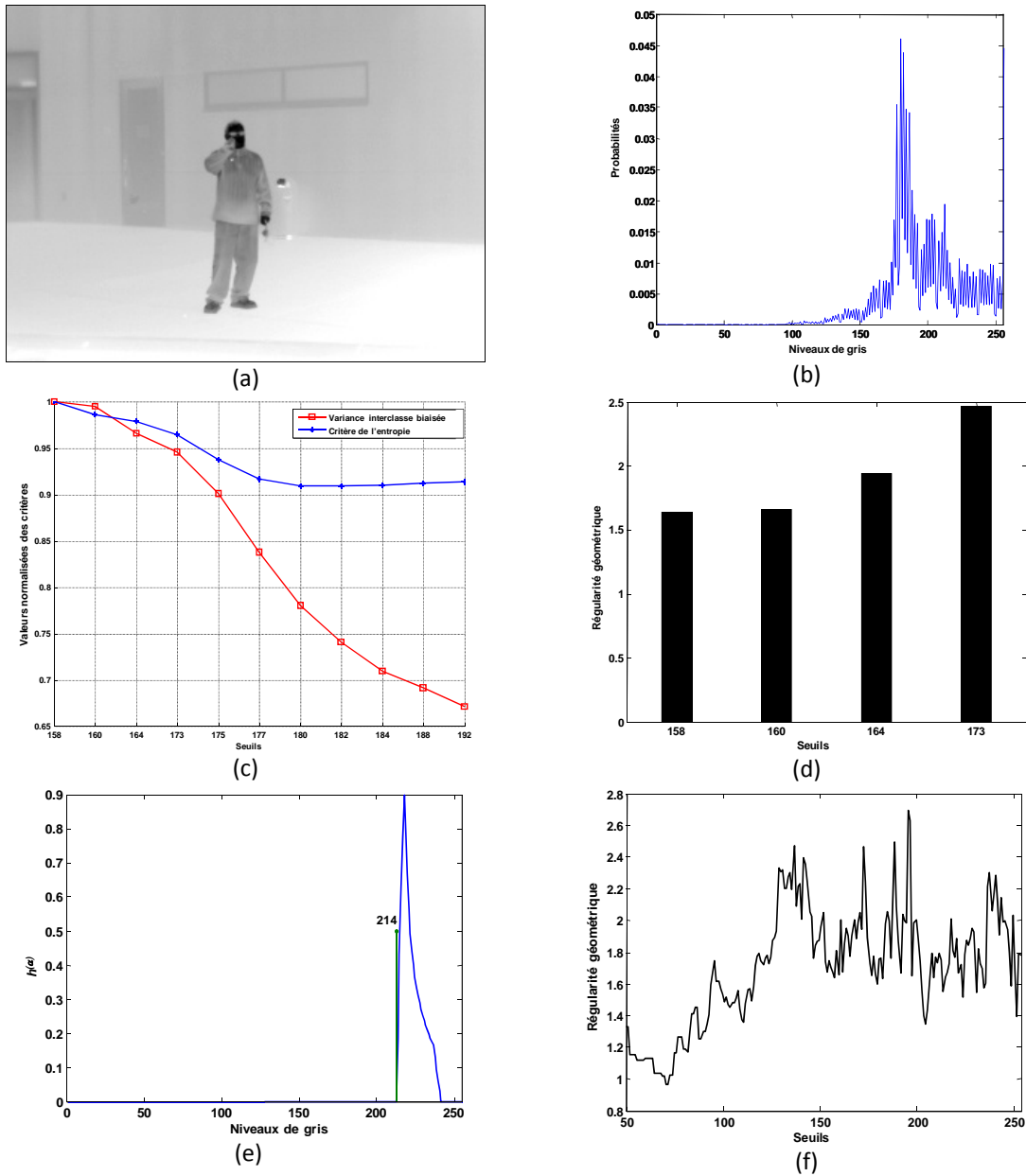


Figure A4.2: Illustration du principe de l'algorithme DNE-1D sur une image infrarouge, (a) Image originale, (b) Histogramme original, (c) Variations des critères  $H$  et  $J$  pour les seuils du sous-ensemble  $\Psi$ , (d) Valeurs de  $Reg$  pour les seuils des sous-ensembles  $\Psi'$  et  $\Psi''$ , (e) Seuil optimal et histogramme différencié pour  $\alpha=0,1$ , (f) Variations du critère de régularité géométrique ( $Reg$ ) pour toutes les solutions possibles.

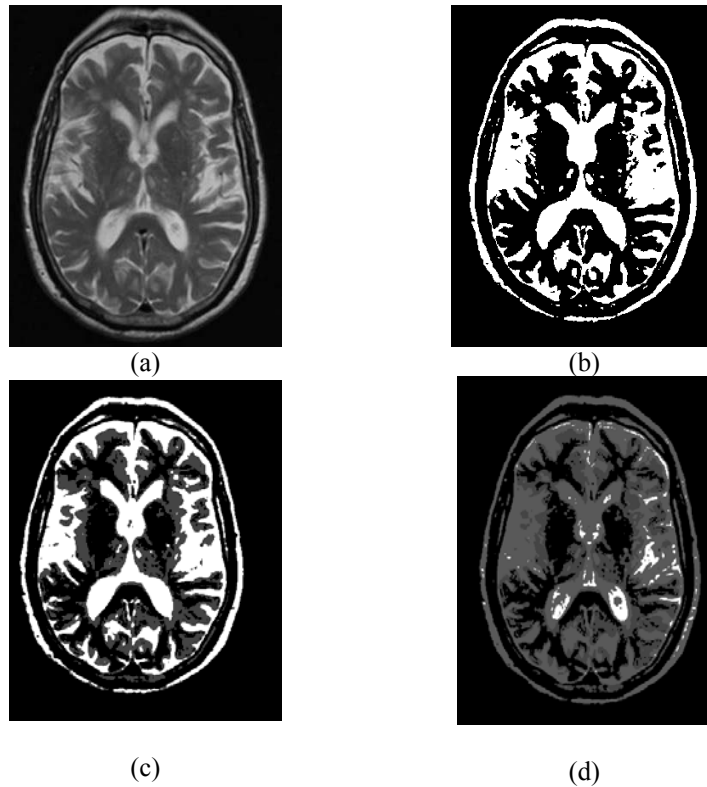


Figure A4.3: Segmentation d'une image IRM d'un sujet atteint d'une atrophie au niveau des ventricules et sur le cortex. (a) Image IRM originale pathologique, (b) Segmentation par DNE-1D en 2 classes:  $\alpha = -0,6$ ;  $t=85$ , (c) Segmentation en 3 classes  $\alpha=(-0,1; 0,5)$ ,  $t=(72; 153)$ , (d) Segmentation en 4 classes  $\alpha=(-0,1;0,1;0,5)$ ;  $t=(69; 89; 172)$ .

## A5. EXEMPLE DE SEGMENTATION PAR LA METHODE DNE-2D

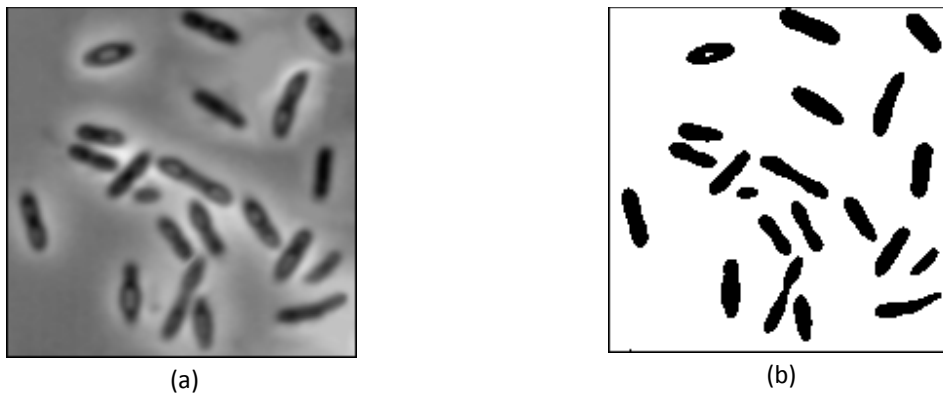


Figure A5. 1: Exemple de segmentation d'une image microscopique de bactéries par l'algorithme DNE-2D. (a) Image originale, (b) Image segmentée ( $t=102$ ).



## A6. EXEMPLES DE SEGMENTATIONS PAR LA METHODE SMOP

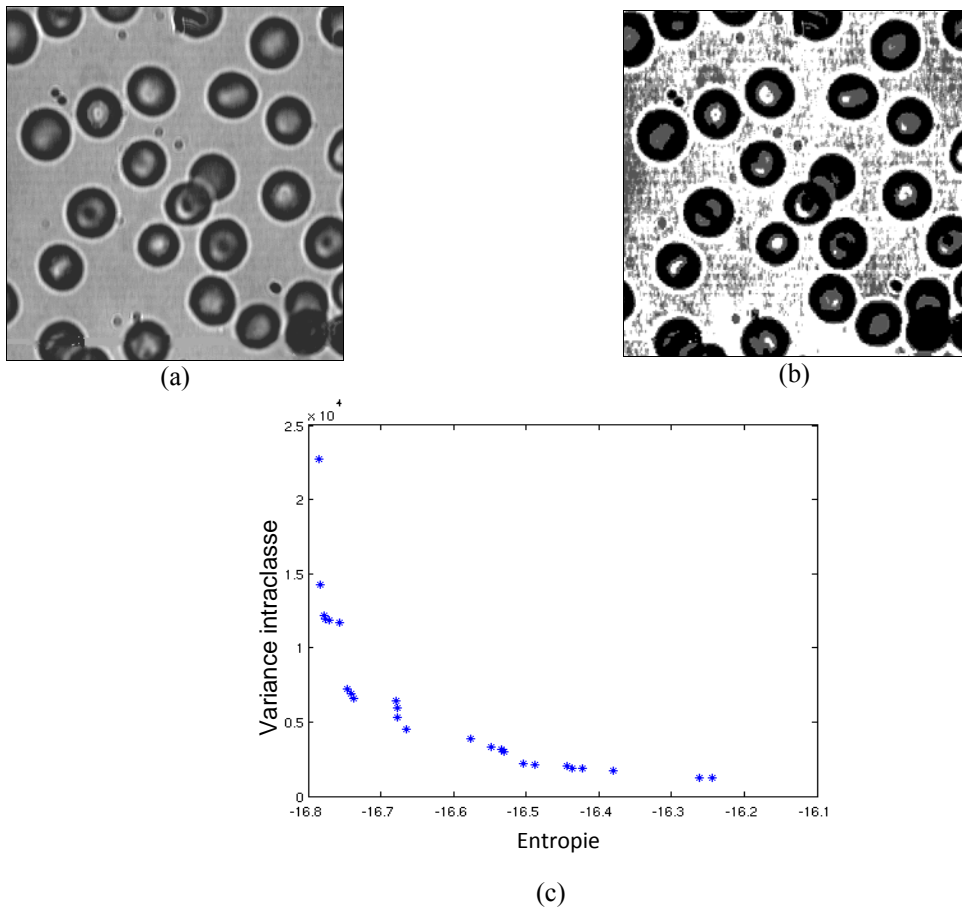


Figure A6. 1: Exemple de segmentation d'une image microscopique de sang. (a) Image originale, (b) Image segmentée en 3 classes, (c) Front de Pareto obtenu après la moyenne de 10 exécutions indépendantes.

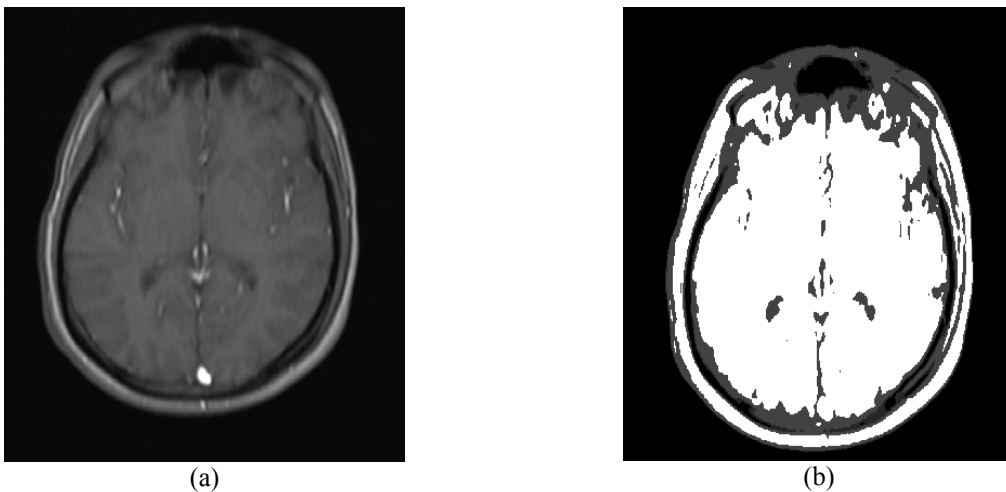


Figure A6.2: Exemple de segmentation d'une image IRM saine. (a) Image originale, (b) image segmentée en 4 classes.

---

## REFERENCES BIBLIOGRAPHIQUES

---

- (Abutaleb, 1989) Abutaleb, A.S., Automatic Thresholding of gray-Level pictures using two-dimensional entropy. *Computer Vision, Graphics and Image Processing*. 1989, Vol. 47, pp. 22-32.
- (Agrafiotis, 2001) Agrafiotis, D. K., Multiobjective optimization of combinatorial libraries. *IBM Journal of Research and Dev.* May/July 2001, Vol. 45, 3/4, pp. 545-566.
- (Andrey, 1997) Andrey, P., Segmentation d'images par algorithmes génétiques. Thèse de doctorat. Université Paris7. Décembre 1997, Paris.
- (Ball, et al., 1965) Ball, G. H. and Hall, D. J., ISODATA, a novel method of data analysis and pattern classification. *Stanford Research Inst. Tech. Rep.* 5RI, project 5533, Standfor (USA), 1965.
- (Banks, et al., 2007a) Banks, A., Vincent, J. and Anyakoha, C., A review of particle swarm optimization. Part I: background and development. *Nat. comput.*. 2007, pp. 1-18.
- (Banks, et al., 2007b) Banks, A., Vincent, J. and Anyakoha, C., A review of particle swarm optimization. Part II: hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications. *Nat. comput.*, 2007, pp. 1-16.
- (Battaglia, 2001) Battaglia, J. L., Cois, O. , Puigsegus, L., and Oustaloup, A., Solving an inverse heat conduction problem using a non-integer identified model. *Inter. J. of Heat and Mass Trans.*. 2001, Vol. 44, pp. 2671-2680.
- (Bazi, et al., 2007) Bazi, Y. and Bruzzone, Melgani, F., Image thresholding based on the EM algorithm and the generalized Gaussian distribution. *Pattern Recognition*. 2007, Vol. 40, pp. 619-634.
- (Beghdadi, et al., 1995) Beghdadi, A., Negrata, A.L. and De Lesegno, P.V., Entropic thresholding using a block source model. *Graph. Models Image Processing*. 1995, Vol. 57, pp. 197-205.
- (Benlian, et al., 2007) Benlian, X. and Zhiquan, W., A multi-objective-ACO-based data association method for bearings-only multi-target tracking. *Communications in Nonlinear Science and Numerical Simulation*. 2007, Vol. 12, pp. 1360-1369.
- (Berkeley, 2007) Berkeley, university. The Berkeley Segmentation Dataset and Benchmark. *Grouping and Ecological Statistics*. June 2007.  
<http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/>.
- (Berro, 2001) Berro, A., Optimisation multiobjectif et stratégies d'évolution en environnement dynamique. Thèse de doctorat. Université des Sciences Sociales Toulouse 1. Décembre 2001, Toulouse.
- (Bhanu, 1994) Bhanu, B., *Genetic Learning for adaptive image segmentation*. USA : Kluwer Academic Publishers, 1994.
- (Bhanu, et al., 1995) Bhanu, B., Lee, S. and Das, S., Adaptive image segmentation using genetic and hybrid search methods. *IEEE Trans. on Aerospace and electronic syst.* 1995, Vol. 31, 4, pp. 1268-1291.
- (Blum, 2005) Blum, C., Ant colony optimization: Introduction and recent trends. *Physics of Life Reviews*. 2005, Vol. 2, pp. 353-373.

- (Borsotti, et al., 1998) Borsotti, M., Campadelli, P. and Schettini, R., Quantitative evaluation of color image segmentation results. *Pattern Recognition Letters*. 1998, Vol. 19, pp. 741-747.
- (Bres, et al., 2003) Bres, S., Jolion, J.M. and Lebourgeois, F., *Traitement et analyse des images numériques*. Lavoisier, 2003.
- (Brink, A.D., 1996) Brink, A.D., Using spatial information as an aid to maximum entropy image threshold. *Recognition Letters*. 1996, Vol. 17, pp. 29-36.
- (Campbell, 1996) Campbell, L.L., Exponential entropy as a measure of extent of a distribution. *Probability Th. and Related Fields*. 1996, Vol. 5, 3, pp. 217-225.
- (Canny, 1986) Canny, J. A., Computational approach to edge detection. *IEEE Pattern Ana. and Mach. Intelli.* 1986, Vol. 8, 6, pp. 679-698.
- (Caruana, et al., 1988) Caruana, R. A. and Schaffer, J. D., Representation and hidden bias: Gray vs. binary coding for genetic algorithms. In *Proc. of Fifth Int. Conf. on Mach. Learning*, pp. 153-161, 1988, San Mateo (Canada).
- (Cerny, 1985) Cerny, V., Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. *J. of Optimization th. and applications*. 1985, Vol. 45, 1, pp. 41-51.
- (Chen, et al., 1999) Chen, S and Luk, B. L., Adaptive simulated annealing for optimization in signal processing applications. *Signal Processing*. 1999, Vol. 79, pp. 117-128.
- (Cheng, et al., 2000) Cheng, H. and Sun, Y., A hierarchical approach to color image segmentation using homogeneity. *IEEE Trans. on Image Processing*. 2000, Vol. 9, 12, pp. 2071-2082.
- (Cheriet, et al., 1998) Cheriet, M., Said, J.N. and Suen, C. Y., A recursive thresholding technique for image segmentation. *IEEE Trans. on Image Processing*. 1998, Vol. 7, 6, pp. 918-921.
- (Chernov, et al., 2004) Chernov, N., Lesort, C. and Simayi, N., On the complexity of curve fitting algorithms. *Journal of complexity*. 2004, Vol. 20, pp. 484-492.
- (Clerc, et al., 2002) Clerc, M. and Kennedy, J., The particle swarm: explosion, stability, and convergence in multi-dimensional complex space. *IEEE Trans. on Evolutionary Compt.*, 2002, Vol. 6, pp. 58-73.
- (Clerc, 2003) Clerc, M., TRIBES - Un exemple d'optimisation par essaim particulaire sans paramètres de contrôle. OEP'03 Paris. Octobre 2003.
- (Cocosco, et al., 2003) Cocosco, C. A. and Evans, A. C., A fully automatic and robust brain MRI tissue classification. *Medical Image Analysis*. 2003, Vol. 7, 4, pp. 513-527.
- (Cocquerez, et al., 1995) Cocquerez, J. P. and Philipp, S., *Analyse d'images et segmentation*. Masson, 1995.
- (Coello, et al., 1995) Coello, C. A. C. et al., Multiobjective design optimization of counterweight balancing of a robot arm using genetic algorithm. *Seventh Int. Conf. on Tools with Arti. Intelli. (TAI'95)*, pp. 20-23, November 5-8, 1995, Herndon (USA).
- (Cohen, et al., 1993) Cohen, L. and Cohen, I., A finite element method applied to new active contours models and balloons for 2D and 3D images. *IEEE Trans. on Pattern Ana. and Mach. Intelli.* 1993, Vol. 15, 11, pp. 1131-1147.
- (Cohen, 1997) Cohen, L., Modèles déformables. Conférence invitée, Ecole thématique ISIS, pp. 1-20, April 4, 1997, Marly le Roy.
- (Collette, et al., 2002b) Collette, Y. and Siarry, P., *Optimisation multiobjectif*. Eyrolles, 2002.
- (Collette, 2002a) Collette, Y., Contribution à l'évaluation et au perfectionnement des méthodes d'optimisation multiobjectif. Application à l'optimisation des plans de rechargement de combustible nucléaire. Thèse de doctotat. Université de Paris 12. Décembre 2002, Créteil.

- (Creutz, 1983) Creutz, M., Microcanonical Monte Carlo simulation. *Physical Review Letters*. May 9, 1983, Vol. 50, 19, pp. 1411-1414.
- (Deb, et al., 2002) Deb, K., et al., A fast elitist non-dominated sorting genetic algorithm for multiobjective optimization: NSGA II. *IEEE Trans. on Evolutionary comput.* April 2002, Vol. 5, 3, pp. 115-148.
- (Deléchelle, et al., 2000) Deléchelle, E. and Lemoine, J., La trajectoire déformable: un modèle de variété active fondé sur le principe de Fermat. *Traitement du signal*. 2000, Vol. 17, 1, pp. 47-58.
- (Demigny, et al., 1997) Demigny, D. and Kamlé, T., A discrete expression of Canny's criteria for step edge detector performances evaluation. *IEEE Pattern Ana. and Mach. Intelli.*, 1997, Vol. 19, 11, pp. 1199-1211.
- (Dempster, et al., 1977) Dempster, A. P., Laird, N. M. and Rubin, D. B., Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*. 1977, Vol. 39, 1, pp. 1-38 .
- (Deriche, 1990) Deriche, R., Using Canny's criteria to derive recursively implemented optimal edge detector. In *Proceedings of Int. Conf. on Pattern Recognition*. pp. 712-716, June 17-21 1990, Atlantic City (USA).
- (Doerner, et al., 2004) Doerner, K., et al., Pareto Ant colony optimization: A metaheuristic approach to multiobjective portfolio selection. *Annals of Operations Research*. 2004, Vol. 131, pp. 79-99.
- (Doerner, et al., 2006) Doerner, K., et al., Pareto ant colony optimization with ILP preprocessing in multiobjective project portfolio selection. *Euro. J. of Operational Research*. 2006, Vol. 171, pp. 830-841.
- (Dorigo, 2006) Dorigo, M., About ACO: Behavior of real ants. *Ant Colony optimization*. November 2006. <http://iridia.ulb.ac.be/~mdorigo/ACO/index.html>.
- (Dorigo, et al., 2005) Dorigo, M. and Blum, C., Ant colony optimization theory: A survey. *Theoretical Computer Sc.* 2005, Vol. 344, pp. 243-278.
- (Dorigo, et al., 1996) Dorigo, M., Maniezzo, V. and colorni, A., Ant System: optimization by a colony of cooperating agents. *IEEE Trans. on Man. Cyber. Part B*. 1996, Vol. 26, 1, pp. 29-41.
- (Drapaca, et al., 2005) Drapaca, C. S. and Cardenas, S., Segmentation of tissue boundary evolution from brain MR image sequences using multi-phase level sets. *Comput. Vision and Image Underst.* 2005, Vol. 100, pp. 312-329.
- (Dréo, et al., 2003) Dréo, J., Pérowski, A., Siarry, P., and Taillard, E., *Métaheuristiques pour l'optimisation difficile*. Eyrolles, 2003.
- (Eglese, 1990) Eglese, R. W., Simulated annealing: A tool for operational research. *European J. of Operational Research*. 1990, Vol. 46, pp. 271-281.
- (Engrand, 1997) Engrand, P., Approche d'optimisation multiobjective basée sur l'adoucissement simulé, et son application à la gestion des combustibles nucléaires. *Fifth Int. Conf. on Nuclear Engineering (ICONE 5)*. pp. 1-8, Vol.10, May 26-30 1997, Nice.
- (Fan, et al., 2007) Fan, S. and Zahara, E., A hybrid simplex search and particle swarm optimization for unconstrained optimization. *Euro. J. of Operational Research*. 2007, Vol. 181, 2, pp. 527-548.
- (Feng, et al., 2005) Feng, D., et al., Infrared image segmentation with 2-D maximum entropy method based on particle swarm optimization (PSO). *Pattern Recognition Letters*. 2005, Vol. 26, 5, pp. 597-603.

- (Ferdin, et al., 2000) Ferdin, Y., Herbeuval, J. P. and Charel, A., Un filtre numérique basé sur la dérivation non-entière pour l'analyse du signal électrocardiographique. *Journal de l'Innovation et Technologie en Biologie et Médecine*. 2000, Vol. 21, pp. 205-209.
- (Fonseca, et al., 1995) Fonseca, C. M. and Fleming, P. J., An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary computation*. 1995, Vol. 3, 1, pp. 1-16.
- (Friesz, et al., 1993) Friesz, T. L., et al., The multiobjective equilibrium network design problem revisited: A simulated annealing approach. *European J. of Operational Research*. 1993, Vol. 65, pp. 44-57.
- (Fukunaga, 1972) Fukunaga, K., *Introduction to statistical pattern recognition*. Academic Press, 1972.
- (Fwu, et al., 1997) Fwu, J.K. and Djuric, P.M., EM algorithm for image segmentation initialized by tree structure scheme. *IEEE Trans. on Image Processing*. 1997, Vol. 6, pp. 349-352.
- (Gagné, et al., 2004) Gagné, C., Gravel, M. and Price, W.L., Optimisation multiobjectifs à l'aide d'un algorithme de colonies de fourmis. *Inf. syst. and Operational Research*. 2004, Vol. 42, 1, pp. 23-42.
- (Gandibleux, et al., 2004) Gandibleux, X., et al., *Metaheuristics for multiobjective optimization*. Springer, 2004.
- (Goldberg, 1989) Goldberg, D. E., *Genetic algorithms in search, optimization and machine learning*. Massachusetts: Addison-Wesley, Reading, 1989.
- (Gonzalez, et al., 2002) Gonzalez, R. C. and Woods, R. E., *Digital image processing*. Pearson Prentice Hall, 2002.
- (Gonzalez, et al., 2004) Gonzalez, R. C., Woods, R. E., *Digital image processing using Matlab*. Pearson Prentice Hall, 2004.
- (Hajek, 1988) Hajek, B., Cooling schedules for optimal annealing. *Mathematics of operations research*. 1988, Vol. 13, 2, pp. 311-329.
- (Han, et al., 2007) Han, Y. and Shi, P., An improved ant colony algorithm for fuzzy clustering in image segmentation. *Neurocomputing*. 2007, Vol. 70, 4, pp. 665-671.
- (Hao, et al., 1999) Hao, J. K., Galinier, P. and Habib, M., Métaheuristiques pour l'optimisation combinatoire et l'affectation sous contraintes. *Revue d'intelli. arti*. 1999, Vol. 13, 22, pp. 283-324.
- (Herault, et al., 1993) Herault, L. and Horaud, R., Figure-Ground Discrimination: A combinatorial optimization approach. *IEEE Trans. on Pattern Ana. and Mach. Intelli.* 1993, Vol. 15, 9, pp. 899-914.
- (Holland, 1975) Holland, J., *Adptation in natural and artificial systems*. University of Michigan Press. 1975.
- (Horaud, et al., 1995) Horaud, R. P. and Monga, O., *Vision par ordinateur: outils fondamentaux*. Paris : Hermes, 1995.
- (Horowitz, et al., 1976) Horowitz, S.L. and Pavlidis, T., Picture segmentation by tree transversal algorithm. *J. ACM* . 1976, Vol. 32, 2, pp. 368-388.
- (Hou, et al., 2006) Hou, Z., Hu, Q. and Nowinski, W. L., On minimum variance thresholding. *Pattern Recognition Letters*. 2006, Vol. 27, pp. 1732-1743.
- (Ishibuchi, et al., 1998) Ishibuchi, H. and Murata, T., A multiobjective genetic local search algorithm and its application to flowshop scheduling. *IEEE Trans on Syst., Man., and Cyber. - Part C: App. and Reviews*. 1998, Vol. 28, 3, pp. 392-403.
- (Jamieson, et al., 2003) Jamieson, M., Fieguth, P. and Lee, L. J., Parametric Contour Estimation by Simulated Annealing. *Proc. of Int. Conf. on Image Process. (ICIP03)*. pp. 449-452, Vol. 3, September 14-17 2003, Barcelona (Spain).

- (Jiang, et al., 2002) Jiang, T. and Yang, F., An evolutionary Tabu search for cell image segmentation. IEEE Trans. on Syst., Man and Cybernetics-Part B. 2002, Vol. 32, 5, pp. 675-678.
- (Jolion, et al., 1992) Jolion, J.M. and Montanvert, A., The adaptive pyramid: framework for 2D image analysis. CVGIP: Image Underst. 1992, Vol. 55, 3, pp. 339-348.
- (Kamber, et al., 2000) Kamber, M., et al., Model-based segmentation of multiple sclerosis lesions in magnetic resonance brain images. IEEE Trans. on Med. Imaging. 2000, Vol. 14, 3, pp. 442-453.
- (Kapur, et al., 1985) Kapur, J. N., Sahoo, P. K. and Wong, A. C. K., A new method for gray-level picture thresholding using the entropy of the histogram. Computer Vision, Graphics and Image Processing. 1985, Vol. 29, pp. 273-285.
- (Kennedy, et al., 1995) Kennedy, J. and Eberhart, R. C., Particle swarm optimization. IEEE Int. Conf. on Neuronal networks, pp. 1942-1948, Vol. 4, October 1995, Piscataway (Japan).
- (Kirkpatrick, et al., 1983) Kirkpatrick, S., Gelatt, C. and Vecchi, M. P., Optimization by simulated annealing. Science. 1983, Vol. 220, 4598, pp. 671-680.
- (Kittler, et al., 1986) Kittler, J. and Illingworth, J., Minimum error thresholding. Pattern Recognition. 1986, Vol. 19, 1, pp. 38-52.
- (Kropatsch, 1995) Kropatsch, W., Building irregulars pyramids by dual graph contraction. In IEEE Proc. Vision, Image and Signal Processing. 1995, Vol. 142, pp. 366-374.
- (Kurita, et al., 1992) Kurita, T., Otsu, N. and Abdelmalek, N., Maximum likelihood thresholding based on population of mixture models. Pattern Recognition. 1992, Vol. 25, pp. 1231-1240.
- (Lahfi, 2006b) Lahfi, Y. and Lemoine, J., Automatically real time region of interest detection method applied to radiological images. In proceedings of IEEE Int. Conf. on information and communication technologies. pp. 1125-1129, April 19-23, 2006, Damascus (Syria).
- (Lahfi, et al., 2006a) Lahfi, Y., Segmentation d'images radiologiques en temps réel dans le cadre d'un système d'aide à la pose d'une prothèse endovasculaire. Thèse de doctorat, Université Paris 12, Octobre 2006, Créteil.
- (Le Guyader, 2004) Le Guyader, C., Imagerie mathématique: segmentation sous contraintes géométriques, théorie et application. Thèse de doctorat. INSA Rouen. Décembre 2004, Rouen.
- (Lee, 1994) Lee, Jooyoung., Optimization by multicanonical annealing and the travelling salesman problem. Physical Review E. August 1994, Vol. 50, 2, pp. R651-R654.
- (Lingrand, 2004) Lingrand, D., Introduction au traitement des images. Vuibert, 2004.
- (Maitre, 2003) Maitre, H. (coordinateur), Le traitement des images. Paris : Hermes, 2003.
- (Majdi, et al., 2006) Majdi-Nasab, N., Analoui, M. and Delp, E. J., Decomposition parameters of mixture gaussian model using genetic and maximum likelihood algorithms on dental images. Pattern Recognition Letters. 2006, Vol. 27, pp. 1522-1536.
- (Marr, et al., 1980) Marr, D. and Hildreth, E., A theory of edge detection. In Proc. of the Royal Society of London. 1980, pp. 187-217.
- (Mathieu, et al., 2003) Mathieu, B., et al., Fractional differentiation for edge detection. Signal Processing. 2003, Vol. 83, pp. 2421-2432.
- (Meer, 1989) Meer, P., Stochastic image pyramids. Comput. Vision Graph. Image Process. 1989, Vol. 45, 5, pp. 269-294.
- (Metropolis, et al., 1953) Metropolis, N., Rosenbluth, M. and Teller, A., Teller, E., Equation of state calculation by fast computing machines. J. of Chemical Physics. 1953, Vol. 21, pp. 1087-1092.

- (Monmarché, 2000) Monmarché, N., Algorithmes des fourmis artificielles: applications à la classification et à l'optimisation. Thèse de doctorat. Université de Tours. Décembre 2000, Tours.
- (Montagnat, 1999) Montagnat, J., Modèles déformables pour la segmentation et la modélisation d'images médicales en 3D et 4D. Thèse de doctorat. INRIA Sophia-Antipolis. Décembre 1999, Nice.
- (Moore, et al., 1999) Moore, J. and Chapman, R., Application of particle swarm to multiobjective optimization. Depart. of Comp. Sc. and Soft. Eng. Auburn University, 1999.
- (Murgasova, et al., 2006) Murgasova, M., et al., Segmentation of brain MRI in young children. 9th Int. Conf. on Medical Image Comput. and Computer-Assisted Intervention (MICCAI) 2006. pp. 687-694, October 2006, Copenhagen (Denmark).
- (Najman, et al., 2003) Najman, L. and Couprie, M., Watershed algorithms and contrast preservation. Discrete geometry for computer imagery. 2003, Vol. 2886, pp. 62-71.
- (Nakib, et al., 2007a) Nakib, A., Cooren, Y., Oulhadj, H. and Siarry, P., Magnetic resonance image segmentation based on two-dimensional exponential entropy and a parameter free PSO. In Proc. of Int. Conf. on Evolution Artificial EA'07. October 29-31, 2007, Tours.
- (Nakib, et al., 2002) Nakib, A., Ferdi, Y., Entropy reduction in fractionally differentiated electrocardiogram signals. European Medical and Biological Engineering Conference (EMBE'02). December 4-8, 2002, Vol. 3, 1, pp. 562-563, Vienna (Austria).
- (Nakib, et al., 2005) Nakib, A., Minery, A., Oulhadj, H. and Siarry, P., Modified intra-class variance minimization for segmentation of retinal angiographic images. Conférence de la Société Française de Génie Biologique et Médical (SFGBM). pp. 134-135, Février 2005, Nancy.
- (Nakib, et al., 2007b) Nakib, A., Oulhadj, H. and Siarry, P., Microscopic image segmentation with two-dimensional exponential entropy based on hybrid microcanonical annealing. IAPR Conference on Machine Vision Applications (MVA2007). pp. 420-423, May 16-18 2007, Tokyo (Japan).
- (Nakib, et al., 2007c) Nakib, A., Roman, S., Oulhadj, H. and Siarry, P., Fast MRI segmentation based on two dimensional survival exponential entropy and particle swarm optimization. In proc. of the 29th Annual Int. Conf. on the IEEE EMBS. pp. 5563-5565, August 22-26 2007, Lyon.
- (Nakib, et al., 2007d) Nakib, A., Oulhadj, H. and Siarry, P., Image histogram thresholding based on multiobjective optimization. Signal Processing. 2007, Vol. 87, pp. 2516-2534.
- (Nakib, et al., 2007e) Nakib, A., Oulhadj, H. and Siarry, P., Non-supervised image segmentation based on multiobjective optimization. To appear in Pattern Recognition Letters. Accepted September 2007.
- (Nakib, et al., 2007f) Nakib, A., Oulhadj, H. and Siarry, P., A thresholding method based on two-dimensional fractional differentiation. Image and Vision Computing. 2007. In revision.
- (Nakib, et al., 2007g) Nakib, A., Oulhadj, H. and Siarry, P., Image histogram thresholding based on fractional differentiation. Under submission (2007).
- (Nakib, et al., 2007h) Nakib, A., Oulhadj, H. and Siarry, P., Segmentation optimale d'images par optimisation multicritère et dérivation non-entière. Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF'07). pp. 337-338, Février 20-23 2007, Grenoble.

- (Nakib, et al., 2006b) Nakib, A., Oulhadj, H. and Siarry, P., Image thresholding based on fractional differentiation. Présentation. Journée du GDR Systèmes à dérivées non-entières (SDNE), Octobre 30 2006, Paris.
- (Nakib, et al., 2006a) Nakib, A., Oulhadj, H. and Siarry, P., Segmentation optimale d'image par recuit simulé. Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF'06), Février 6-8, 2006, Lille.
- (Nelder, et al., 1965) Nelder, J. A. and Mead, R., A simplex method for function minimization. *Comput. J.* 1965, Vol. 7, pp. 308-313.
- (Ng, 2006) Ng, H., Automatic thresholding for defect detection. *Pattern Recognition Letters*. 2006, Vol. 27, pp. 1644-1649.
- (Oldham, et al., 1974) Oldham, K. and Spanier, J., *The fractional calculus*. Academic Press, 1974.
- (Otsu, 1979) Otsu, N., A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans. on Syst., Man and Cyb.* 1979, Vol. 9, 1, pp. 62-66.
- (Oustaloup, 1996) Oustaloup, A. and Linares, H., The CRONE path planning. *Math. and Compu. in Simu.* 1996, Vol. 41, pp. 209-217.
- (Pal, et al., 1991) Pal, N. K. and Pal, S. K., Entropy: A new definition and its applications. *IEEE Trans. Syst. Man. Cybern.* 1991, Vol. 21, pp. 1260-1270.
- (Pal, et al., 1989) Pal, N.K. and Pal, S.K., Object-background segmentation using new definitions of entropy. *IEE Proceedings on Computer and Digital Techniques*. 1989, Vol. 136, pp. 284-295.
- (Pareto, 1896) Pareto, V., *Cours d'économie politique*. Rouge, 1896.
- (Portes, et al., 2004) Portes de Albuquerque, M., Esquef, I. and Gesualdi Mello, A., Image thresholding using Tsallis entropy. *Pattern Recognition Letters*. 2004, Vol. 25, pp. 1059-1065.
- (Prasanna, et al., 2004) Prasanna, P.K., Sahoo, K. and Arora, G., A thresholding method based on two-dimensional Renyi's entropy. *Pattern Recognition*. 2004, Vol. 37, pp. 1149-1161.
- (Prodlubny, 2002) Prodlubny, I., Geometric and physical interpretation of fractional integration and fractional differentiation. *Fractional Calculus and Applied Analysis*. 2002, Vol. 5, 4, pp. 367-386.
- (Qiao, et al., 2007) Qiao, Y., et al., Thresholding based on variance and intensity contrast. *Pattern Recognition Letters*. 2007, Vol. 40, pp. 596-608.
- (Ramus, et al., 2002) Ramus-Serment, C., et al., Generalised approach on fractional response of fractal networks. *Chaos, Solitons and fractals*. 2002, Vol. 14, pp. 479-488.
- (Ray, et al., 1997) Ray, John R. and Harris, Richard W., Simulated annealing in the microcanonical ensemble. *Physical Review E*. May 1997, Vol. 55, 5, pp. 5270-5274.
- (Reddi, et al., 1984) Reddi, S. S., Rudin, S. F. and Keshavan, H. R., An optimal multiple threshold scheme for image segmentation. *IEEE Trans. on Syst., Man., and cybern.* 1984, Vols. SMC-14, pp. 661-665.
- (Reeves, 1995) Reeves, C. R., *Modern heuristic techniques for combinatorial problems*. McGraw-Hill, 1995.
- (Reyes-Sierra, et al., 2006) Reyes-Sierra, M. and Coello, C. A. C., Multi-objective Particle Swarm Optimizers: A survey of the State-of-the-Art. *Int. Journal of Comput. Intelligence Research*. 2006, Vol. 2, 3, pp. 287-308.
- (Richardson, et al., 1989) Richardson, J. T., et al., Some guidelines for genetic algorithms with penalty functions. In the third Int. Conf. on Genetic Algorithms (ICGA'03), pp. 191-197, June 1989, Fairfax (USA).



- (Ridler, et al., 1978) Ridler, T. W. and Calvard, S., Picture thresholding using an iterative selection method. *IEEE Trans. on Syst. Man. Cybern.*. 1978, Vols. SMC-8, pp. 630-632.
- (Roussel-Ragot, 1990) Roussel-Ragot, Dreyfus, G., A problem-Independent parallel implementation of simulated annealing: Models and experiments. *IEEE Trans. on Computed Aided Design*. 1990, Vol. 9, pp. 827-837.
- (Sahoo, et al., 2006) Sahoo, K. P. and Arora, G., Image thresholding using two dimensional Tsallis-Havrda-Charvat entropy. *Pattern Recognition Letters*. 2006, Vol. 27, pp. 520-528.
- (Sahoo, et al., 1988) Sahoo, P. K., et al., A survey of thresholding techniques. *Computer Vision, Graphics, and Image Processing*. 1988, Vol. 41, pp. 233-260.
- (Schaffer, 1985) Schaffer, J. D., Multiple objective optimization with vector evaluated genetic algorithm. In the proceedings of the first Int. Conf. on Genetic Algorithms, pp. 93-100, July 1985, Pittsburgh (USA).
- (Serafini, 1992) Serafini, P., Simulated annealing for multiple objective optimization problems. Making, In Tenth Int. Conf. on Multiple Criteria Decision, pp. 87-96, July 1992, Taipei (Taiwan).
- (Sezgin, et al., 2004) Sezgin, M. and Sankur, B., Survey over image thresholding techniques and quantitative performance evaluation. *J. of Electronic Imaging*. 2004, Vol. 13, 1, pp. 146-165.
- (Shen, et al., 1986) Shen, J. and Castan, S., An optimal linear operator for edge detection. In proceedings of Conference on Vision and Pattern Recognition (CVPR). pp. 109-114, June 1986, Miami (USA).
- (Shukla, et al., 2007) Shukla, P. K., Deb, K., On finding multiple Pareto-optimal solutions using classical and evolutionary generating methods. *Euro. J. of Operational Research*. 2007, Vol. 81, pp. 1630-1652.
- (Siarry, et al., 2007) Siarry, P. (coordinateur), *Optimisation en traitement du signal et de l'image*. Hermes, 2007.
- (Siarry, et al., 1997) Siarry, Patrick, et al., Enhanced simulated annealing for globally minimizing functions of many continuous variables. *ACM Trans. on Math. Soft.*. June 1997, Vol. 23, 2, pp. 209-228.
- (Song, et al., 2006) Song, Z., et al., Integrated Graph Cuts for Brain MRI Segmentation. 9th Int. Conf. on Medical Image Comput. and Computer-Assisted Intervention (MICCAI) 2006. pp. 831 -838, October 2006, Copenhagen (Denmark).
- (Srinivas, et al., 1995) Srinivas, N. and Deb, K., Multiobjective optimization using non-dominated sorting in genetic algorithms. *Evolutionary Computation*. 1995, Vol. 2, 8, pp. 221-248.
- (Sugeno, 1974) Sugeno, M., *Theory of fuzzy integrals and its applications*. Tokyo : institute of technology, 1974.
- (Sum, et al., 2007) Sum, K.W. and Cheungn, P. Y. S., Boundary vector field for parametric active contours. *Pattern Recognition*. 2007, Vol. 40, pp. 1635-1645.
- (Synder, et al., 1990) Synder, W., Bilbro, G., Logenthiran, A. and Rajala, S., Optimal thresholding– A new approach. *Pattern Recognition Letters*. 1990, Vol. 11, pp. 803-810.
- (Talbi, 1999) Talbi, E.G., *Métaheuristiques pour l'optimisation combinatoire multi-objectif: Etat de l'art*. CNET, 1999. PE: 98-757.33.
- (Terzopoulos, et al., 1988) Terzopoulos, D., On matching deformable models to image. Technical report 60, Shlumberger Palo Alto research, 1986. Reprinted in *Topical Meeting on Machine Vision, Technical Digest Series*. 1987, Vol. 12, pp. 160-167.

- (Terzopoulos, 1987) Terzopoulos, D., Witkin, A. and Kass, M., Constraints on deformable models: recovering 3D shape and nongrid motion. *Artificial Intelligence*. 1988, Vol. 22, 1, pp. 91-123.
- (Torreao, et al., 1995) Torreao, J. R. A. and Roe, E., Microcanonical optimization applied to visual processing. *Physics Letters*. 1995, Vol. 205, A, pp. 377-382.
- (Ulungu, et al., 1999) Ulungu, E. L., Teghem, J., Fortemps, P. and Tuytens, D., MOSA method: A tool for solving multi-objective combinatorial optimization problems. *Journal of Multicriteria Decision Ana.*, 1999, Vol. 20, 2, pp. 221-236.
- (Wan, 1996) Wan, Y., Un opérateur hyperbolique pour la détection de contours dans les images numériques. Thèse de doctorat Université de Lille 1. Décembre 1996, Lille.
- (Warfield, et al., 2000) Warfield, S. K., et al., Adaptive template moderate spatially varying statistical classification. *Medical Image Ana.* 2000, Vol. 4, 1, pp. 43-55.
- (Weszka, et al., 1979) Weszka, J. S. and Rosenfeld, A., Histogram modification for threshold selection. *IEEE Trans. Syst. Man Cybernet.* 1979, Vol. 9, pp. 38-52.
- (Woods, et al., 1998) Woods, I. and Downs, T., Demon algorithms and their application to optimization problems. In *Proc. of Int. Joint Conf. on Neural Networks 2 and IEEE World Congress on Computational Intelli.*, pp. 1661-1666, May 4-9 1998, Anchorage (USA).
- (Xu, et al., 2000) Xu, C., Pham, D.L. and Prince, J.L., Image segmentation using deformable models. In *Handbook of Med. Imag.*, 2000.
- (Yasuda, et al., 2004) Yasuda, K. and Iwasaki, N., Adaptive particle swarm optimization using velocity information of swarm. *IEEE Conf. on Syst. Man. and Cyber.*, pp. 3475-3481, October 10-13 2004, The Hague (Netherlands).
- (Ye, et al., 2002) Ye, X.F., Zhang, W.J. and Yang, Z.L., Adaptive particle swarm optimization on individual level. *Int. Conf. on Signal Processing (ICSP)*. pp. 1215-1218, August 26-30 2002, Beijing (China).
- (Yin, 2007) Yin, P., Multilevel minimum cross entropy threshold selection based on particle swarm optimization. *Applied Mathematics and Computation*. 2007, Vol. 184, 2, pp. 503-513.
- (Yin, 1999) Yin, P. Y., A fast scheme for optimal thresholding using genetic algorithms. *Signal Processing*. 1999, Vol. 72, pp. 85-95.
- (Zahara, et al., 2004) Zahara, E., Fan, S. and Tsai, D., Optimal multi-thresholding using a hybrid optimization approach. *Pattern Recognition Letters*. 2005, Vol. 26, 8, pp. 1082-1095.
- (Zhang, et al., 2003 ) Zhang, W., Liu, Y. and Clerc, M., An adaptive PSO algorithm for real power optimization. *APSCOM (Advances in Power System Control Operation and Management). Application of Artificial Intelligence Technique (part I)*. pp. 302-307, November 11-14 2003, Hong Kong (China).
- (Zitzler, et al., 1999) Zitzler, E. and Thiele, L., Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Trans. on Evol. Comput.* November 1999, Vol. 3, 4, pp. 257-271.
- (Zitzler, et al., 2002) Zitzler, E., Laumanns, M. and Thiele, L., SPEA2: Improving the Strength Pareto Evolutionary Algorithm for multiobjective optimization. In *Proc. of Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems (EUROGEN 2001)*, pp. 95-100, September 12-21 2001, Athens (Greece).
- (Zografos, et al., 2005) Zografos, K. and Nadarajah, S., Survival Exponential Entropies. *IEEE Trans. on Inf. Th.* 2005, Vol. 51, 3, pp. 1239-1246.