

République Algérienne Démocratique et Populaire
Ministre de l'enseignement supérieur et de la recherche scientifique

UNIVERSITE MOHAMED KHIDER BISKRA
FACULTE DES SCIENCES EXACES, DES SCIENCES DE LA NATURE
ET DE LA VIE
DEPARTEMENT D'INFORMATIQUE

N d'ordre :.....
Série :.....

THESE

Présentée pour obtenir le grade de
DOCTORAT EN SCIENCES EN INFORMATIQUE

Algorithmes Evolutionnaires pour la Segmentation d'Images basée sur les Automates Cellulaires

Présentée par :
Melle. SLATNIA Sihem

Dirigée par :
Dr. KAZAR Okba
Université Mohamed Khider Biskra

Soutenue le :

Devant le jury :

- Président : **Pr.** KOUDIL Mouloud (Professeur, Université d'Alger)
- Rapporteur : **Dr.** KAZAR Okba (Maître de conférences, Université de Biskra)
- Examineurs : **Pr.** BENATCHBA Karima (Professeur, Université d'Alger)
Dr. CHERIF Fodil (Maître de conférences, Université de Biskra)
Dr. MOUSSAOUI Abdelwahab (Maître de conférences, Université de Sétif)

RÉSUMÉ

L'objectif de notre travail vise à s'inspirer des phénomènes naturels afin d'utiliser leurs puissances dans la résolution des problèmes difficiles. Ce travail consiste à explorer le domaine des algorithmes génétiques et des systèmes complexes (**SCs**). Dans cette thèse, nous nous sommes intéressés à l'exploitation du phénomène de l'émergence dans le problème de détection de contours des images en couleurs. Nous utilisons les Automates Cellulaires (**ACs**) qui sont un outil puissant de modélisation basée sur l'émergence. Nous avons choisis les Algorithmes Génétiques (**AGs**) pour déterminer les meilleures règles données pour la détection de contours des images en couleurs. Nous visons une implémentation de cette approche pour valider et détecter les meilleures règles (paquets). Nous avons déterminé les rôles de chaque règle surtout dans la détection de contours.

Mots clés : Vie Artificielle, Systèmes Complexes, Emergence, Automates Cellulaires, Algorithmes Génétiques, Détection de Contours.

ABSTRACT

In this thesis, we are interested in Cellular Automata (CA) model-based approaches for image segmentation and Evolutionary Algorithms.

In our first approach, we combine Cellular Automata model and Genetic Algorithms to present a new image segmentation method.

We seek specialized powerful rules of Cellular Automata (CA) for extracting edges in a given colour images. This set of local rules given as packet format determines the future state of CA in an asynchronous way, based on the state of its neighbourhood. The optimal packet (set of transitions) reassuring the best result which is computed by a Genetic Algorithm (GA).

Keyword :- Artificial Life, Complex Systems, Emergence, Cellular automata, Genetic Algorithms, Edge Detection.

BIBLIOGRAPHIE DE L'AUTEUR

Revue International :

1. S. SLATNIA, O. KAZAR, Generalised Evolutionary Cellular Automata based approach for edge detection, In International Review on Computers and Software (I.RE.CO.S.) Praise Worthy Prize 2008, ISSN 1828-6003, Vol. 3 N.4, July 2008, Manuscript received and revised June 2008, Accepted July 2008 pp 424-428.
2. S. SLATNIA, O. KAZAR, Evolutionary Cellular Automata for image segmentation and noise filtering using genetic algorithms, to appear in Journal Applied Computer Science and Mathematics, Issues No. 10, March 2011, Editor in chief : Doru E Tiliute, ROMANIA, 2011.

Revue national :

S. SLATNIA, O. KAZAR, IMAGES SEGMENTATION BASED CONTOUR USING EvCA APPROACH, EVOLUTIONARY CELLULAR AUTOMATONS, COURIER DU SAVOIR, On revue Périodique de l'Université Mohamed Khider Biskra, Algérie 2009, ISSN 1112-3338, Vol. N°09, Mars 2009, pp 113-118.

Symposium international :

S. SLATNIA, O. KAZAR, Evolutionary Cellular Automata based approach for Region Detection, 5ème symposium international, Image Multimédia Application Graphique et Environnement IMAGE, 03, 04-05 Novembre 2009, Papiers Choisis Révisés pp. 75-82, 2009. Biskra, Algérie.

Conférences internationales :

1. S. SLATNIA, M.C. Batouche et K.E. Melkemi, Evolutionary Cellular Automata Based-Approach for Edge Detection. WILF 2007 SCIP special session, In F. Masulli, S. Mitra, and G. Pasi (Eds.) : Workshop on Fuzzy logic and Application, LNAI 4578, Revised Selected Papers pp. 404-411, 2007. Springer-Verlag Berlin Heidelberg 2007.
2. S. SLATNIA, O. KAZAR, Images Segmentation based contour using EvCA approach, Evolutionary Cellular Automats. 18 th International Conference on Computer Theory Application, ICCTA 2008, 11-13 October 2008, Papiers Choisis Révisés pp123-128. Alexandria, Egypt.

DÉDICACE

A ma mère

REMERCIEMENTS

Je tiens à présenter mes remerciements et ma reconnaissance très chaleureuse à mon encadreur le Docteur Okba kazar de l'Université Mohamed khider de Biskra, pour m'avoir proposé un aussi important sujet. J'ai trouvé dans son esprit positif, ses conseils, ses orientations et sa grande expérience, un soutien dans les moments les plus difficiles.

Je remercie chaleureusement le professeur KOUDIL Mouloud, pour m'avoir fait le grand honneur de présider le jury.

J'exprime mes remerciements à Madame le Professeur BENATCHBA Karima, pour sa participation à ce jury malgré ses occupations.

Mes remerciements vont également à Monsieur le Docteur CHERIF Fodil, pour sa participation à ce jury durant cette période sociale.

J'adresse tous mes remerciement à Monsieur le Docteur MOUSSAOUI Abdelwahab, pour sa disponibilité à juger ce travail.

Enfin, que toutes les personnes qui ont oeuvré de près ou de loin à la réalisation de ce travail, trouvent ici l'expression de mes sincères remerciements.

Table des matières

1	INTRODUCTION GÉNÉRALE	1
1.1	Contexte du travail de recherche	1
1.2	Problématique	2
1.3	Contribution et objectifs du travail	2
1.4	Structure et contenu de document	3
2	COMPLEXITÉ ET SYSTÈMES COMPLEXES	4
2.1	Introduction	5
2.2	Généralités et définitions	5
2.2.1	Définition d'un système	5
2.2.2	Système simple	6
2.2.3	Système compliqué	6
2.2.4	Système complexe	6
2.3	Niveaux de complexité	7
2.4	Définitions de la complexité	7
2.4.1	La complexité brute (aléatoire)	7
2.4.2	La complexité effective (organisée)	9
2.5	Systèmes complexes	10
2.5.1	Définitions de système complexe	10
2.5.2	Propriétés d'un système complexe	10
2.5.3	Modélisations des systèmes complexes	12
2.6	Classification des SCs	13
2.6.1	Type 1 - Complexité Statique	14
2.6.2	Type 2 - Complexité Dynamique	14
2.6.3	Type 3 - Complexité Evolutive	14
2.6.4	Type 4 - Complexité Auto-Organisatrice	14
2.7	Caractéristiques des SCs	15
2.7.1	Emergence	16
2.7.2	Changement du niveau d'intégration	16
2.7.3	Boucles de rétroaction et Non-Linearité	16
2.7.4	Indéterminisme	17
2.8	Exemples de SCs naturels et/ou artificiels	17
2.8.1	Systèmes Complexes Naturels (SCNs)	18
2.8.2	Systèmes Complexes Artificiels (SCAs)	22
2.9	Conclusion	34

3	LE PHÉNOMÈNE D'ÉMERGENCE	36
3.1	Introduction	37
3.2	Phénomène d'émergence	37
3.2.1	Définition de l'émergence	38
3.3	Les propriétés d'un système émergent	39
3.3.1	La non-linéarité	39
3.3.2	Proche/Loin de l'équilibre	39
3.3.3	L'auto-organisation	40
3.3.4	Les attracteurs	40
3.3.5	La dynamique	40
3.4	La résolution de problèmes par émergence	40
3.4.1	Etudes des systèmes complexes par émergence	41
3.4.2	Construction de systèmes complexes artificiels par émergence	41
3.5	Les approches émergentes	42
3.5.1	Les Automates Cellulaires Evolutionnaires (ACEv)	42
3.5.2	Les systèmes adaptatifs à fonctionnalité émergente : la théorie des SMAA	44
3.5.3	La programmation émergente : le Calcul Emergent (CE)	44
3.6	Conclusion	45
4	SEGMENTATION D'IMAGE	46
4.1	Introduction	47
4.2	Image à niveau de gris	47
4.3	Image en couleur	48
4.3.1	Images 24 bits (couleurs vraies)	49
4.3.2	Images à palettes, images en 256 couleurs (8 bits)	50
4.4	Les caractéristiques d'une image numérique	50
4.4.1	La résolution	50
4.4.2	Pixel écran-Pixel numérique	51
4.4.3	La luminance	51
4.4.4	Le contraste	52
4.4.5	Histogramme	52
4.5	Propriété d'une image numérique	53
4.6	Le seuillage	54
4.7	La segmentation d'images et détection de contours	54
4.7.1	Présentation du problème	54
4.7.2	Définition	55
4.7.3	Les types de segmentation d'image	56
4.7.4	Les différentes approches de la segmentation	57
4.7.5	Critère de choix de la technique de segmentation	70
4.8	Conclusion	71
5	APPROCHE PROPOSÉE : Généralisation des Automates Cellulaires Evolutionnaires pour la Détection de Contours (GACEv-DC)	72
5.1	Introduction	73
5.2	La segmentation comme fonctionnalité émergente	73
5.3	L'approche proposée : la Généralisation des Automates Cellulaires Evolutionnaires pour la Détection de Contours (GACEv-DC)	74
5.3.1	Phase d'apprentissage	75

5.3.2	Phase de construction de paquet optimal	76
5.3.3	Phase d'adaptation	84
5.3.4	Architecture	84
5.4	Conclusion	85
6	IMPLÉMENTATION ET RÉSULTATS EXPÉRIMENTAUX	86
6.1	Introduction	87
6.2	Paquet de prétraitement	87
6.3	Paquet optimal GACEv-DC	87
6.4	Paquets d'adaptation	88
6.4.1	Paquet de Double Contour Coloré (PDCC)	89
6.4.2	Paquet d'un Seul Contour Coloré (PSCC)	92
6.4.3	Paquet d'un Seul Contour Blanc (PSCB)	93
6.5	Présentation de l'interface de logiciel GACEv-DC	93
6.6	Résultats expérimentaux	94
6.6.1	Résultats de phase d'apprentissage	94
6.6.2	Résultats de phase de teste	98
6.6.3	Discussion	99
6.7	Conclusion	106
7	CONCLUSION GÉNÉRALE	107

Liste des tableaux

4.1	Image en niveau de gris.	48
6.1	Les valeurs d'évaluation des expériences sont rapportées dans les figures 6.14, A, B, C, D.	96
6.2	Les valeurs d'évaluation des expériences sont contenues dans les figures 6.23, 6.18, 6.22 et 6.24	104

Table des figures

2.1	Typologie des systèmes.	6
2.2	Types d'interactions entre les éléments d'un système.	7
2.3	Entités en interaction pour un dynamique et structure globale, adapte de [Amma2007].	11
2.4	Fractale de Mandelbrot.	14
2.5	Différents types d'attracteurs : Point fixe, Périodique et Chaotique.	15
2.6	Schéma de boucle de rétroaction simple et double, adapte de [Amma2007].	17
2.7	Le tas de sable réel.	18
2.8	Les trois états granulaires.	18
2.9	Avalanches et affaissements du tas de sable.	19
2.10	Réseau de cellules de Bénard.	21
2.11	Déplacements collectifs d'oiseaux migrateurs et de bancs de poissons.	21
2.12	Les Boids de Reynolds.	23
2.13	Le voisinage local d'un boid,	23
2.14	Comportements de direction : alignement, cohésion et séparation, Adapté de [Reyn2001].	23
2.15	Schéma d'un Algorithme Evolutionnaire.	24
2.16	L'opérateur de croisement.	25
2.17	L'opérateur de mutation.	26
2.18	Principe de base d'un AG [Roma2003].	27
2.19	La sélection par clonage des lymphocytes, présentant des récepteurs spécifiques d'un antigène, se différencient en cellule mémoire ou en cellule participant à la défense active de l'organisme par le biais d'anticorps.	28
2.20	Mode d'arrangement des cellules.	34
2.21	Les quatre classes de ACs, Adaptées de Wolfram [Wolf1983]	34
3.1	Travaux sur l'émergence, adapte de [Géor2003].	37
3.2	Illustration d'un AC unidimensionnel à états binaires, avec un Voisinage de $r = 1$ et $n = 11$,	43
4.1	Représentation d'une image à niveau de gris du chromosome et des zooms d'une partie de cette image.	48
4.2	Image en niveau de couleur.	49
4.3	Image en couleur.	49
4.4	Les couleurs varient.	50
4.5	Différentes résolution.	51
4.6	Echange de luminance.	51
4.7	Echange de contraste.	52
4.8	Histogramme d'une image.	53

4.9	Segmentation en imagerie satellitaires et aériennes.	56
4.10	Segmentation en imagerie médicale.	56
4.11	Type de segmentation d'images.	57
4.12	Les différentes approches de la segmentation.	58
4.13	Exemple type contour.	58
4.14	Filtrage de Sobel.	61
4.15	Filtrage Laplacien.	62
4.16	Les différentes méthodes de détection de contours.	63
4.17	Exemple type région	65
4.18	Les différentes régions de l'image.	66
4.19	Décomposition de l'image.	67
4.20	Division par Quad-Tree et regroupement feuilles similaires.	67
4.21	Résultat de segmentation par décomposition/fusion.	67
4.22	Segmentation par agrégation des pixels.	68
4.23	Résultat de segmentation par agrégation de pixels.	68
4.24	Classification de pixels.	69
4.25	Méthode structurelle.	70
5.1	Générique pattern, Ci : classe i, i : numéro de classe.	75
5.2	Patterns de base pour la phase d'apprentissage.	76
5.3	Règle de AC, voisinage et état futur.	77
5.4	Equivalence rotationnelle des règles.	77
5.5	Construction du modèle de voisinage.	79
5.6	Construction de séquence d'exécution de règle.	80
5.7	Opérateur de croisement horizontal.	81
5.8	Opérateur de mutation.	81
5.9	Sélection des chromosomes pour des opérations génétiques.	83
5.10	La Généralisation des Automates Cellulaires Evolutionnaires (GACEv-DC) pour la Détection de Contours dans des images en couleur.	84
6.1	Paquet d'apprentissage, un meilleur paquet des règles de AC trouvent par AG. (1, 2, 3 : sont les index de couleur).	87
6.2	Paquet optimal GACEv-DC avec 1057 Règles.	88
6.3	Image d'apprentissage collectif.	89
6.4	L'évolution de Fitness par générations pour des images de la FIG.6.3.	91
6.5	Paquet de Double Contour Coloré (PDCC).	92
6.6	Paquet d'un Seul Contour Coloré (PSCC).	92
6.7	Paquet d'un Seul Contour Blanc (PSCB).	93
6.8	Interface principale de logiciel GACEv-DC.	93
6.9	Présentation des sous interfaces de logiciel GACEv-DC.	94
6.10	Détection de contour de la phase d'apprentissage de 1er ensemble d'images.	95
6.11	L'évolution de la fonction Fitness par générations pour les images de la figures 6.14, A, B, C, D.	95
6.12	L'évolution de GEN-Résultats et MAX-GEN par générations pour des images de la Table 6.1 (GEN-Résultats : numéro de la génération de meilleur résultat, MAX-GEN : numéro de maximum de générations générées).	96
6.13	L'évolution de la fonction Fitness par Etalon pour des images de la Table 6.1.	96
6.14	Détection de contour de la phase d'apprentissage de 2ème ensemble d'images.	97

6.15	Détection de contour de la phase de test de l'ensemble d'images.	98
6.16	Evolution de la fonction Fitness par génération pour les images de la figures 6.15. A, B, C.	99
6.17	Résultats de classification de l'image 'Apr2'.	100
6.18	Les images étalon de 'Apr2'.	100
6.19	Les images étalon de 'Apr2'.	101
6.20	Diagrammes de détection de contour descriptifs des trois composants pour l'image 'Apr2'.	101
6.21	Histogramme de la fonction Fitness des trois paquets pour l'image 'Apr2'. .	102
6.22	Détection de contour par les paquets d'adaptation a) 'Apr3' : image Ori- ginale en NG de 46*46 pixels, b) Image Classifier, c) Image Etalon Canny, d) PDCC, e) PDCB, f) PSCC, g) PSCB, k) paquet optimal GACEv-DC. .	102
6.23	Détection de contour par les paquets d'adaptation a) 'Apr1' : image Ori- ginale en NG de 32*31 pixels, b) Image Classifier, c) Image Etalon Canny, d) PDCC, e) PSCB, f) PSCC, g) PSCB, k) paquet optimal GACEv-DC. .	103
6.24	Détection de contour par les paquets d'adaptation a) 'Fleur' : image Ori- ginale en NG de 98*108 pixels, b) Image Classifier, c) Image Etalon Canny, d) PDCC, e) PDCB, f) PSCC, g) PSCB.	103
6.25	Evaluation comparative de la fonction Fitness de chaque image en niveau de gris et en couleur.	104
6.26	Résultats de détection de contour par les paquets d'adaptation dans des images en couleur.	105

Liste des abréviations

(abrev.)	Description
(AC)	Automate Cellulaire
(AG)	Algorithmes Génétique
(SCNs)	Systèmes Complexes Naturels
(SCAs)	Systèmes Complexes Artificiels
(SAR)	Système autoreproducteur
(CU)	Constructeur universel
(DESC)	Description
(PixEls)	PICTure Eléments
(PPP)	Point Par Pouces
(DPI)	Dots Per Inch
(PES)	Postscript Encapsulated
(PCT)	Pict Macintosh
(TIFF)	Tag Image File Format
(RVB)	Rouge, Vert et Bleu
(ACE _{vs})	Automates Cellulaires Evolutionnaires
(SMAA)	Les Systèmes Multi-Agents Adaptatifs
(SMA)	Les Systèmes Multi-Agents
(IA)	Intelligence Artificielle
(IAD)	Intelligence Artificielle Distribuée
(CE)	Les Calculs Emergents
(RNA)	Les Réseaux Neurones Artificiels
(SIN)	Les Systèmes Immunitaires Naturel
(SIA)	Les Systèmes Immunitaires Artificiels
(ACE _v -DC)	Les Automates Cellulaires Evolutionnaires pour la Détection de Contours
(GACE _v -DC)	Généralisation des Automates Cellulaires Evolutionnaires basée-approche pour la Détection de Contours
(PDCC)	Paquet de Double Contour Coloré
(PSCC)	Paquet d'un Seul Contour Coloré
(PDCB)	Paquet de Double Contour Blanc
(PSCB)	Paquet d'un Seul Contour Blanc
(SEC)	Seconde
(PSO)	Optimisation par essaim de particules

Liste des équations

(N.)	Description	Page
(2.1)	Nombre de bits nécessaires pour décrire un système à Ω états	8
(2.2)	Entropie d'un système	8
(2.3)	Entropie mesure du désordre d'un système	8
(2.4)	Formalisme par les équations différentielles	13
(2.5)	Formalisme par les équations aux dérivées partielles	13
(2.6)	Formalisme par les équations maîtresses	13
(2.7)	Formalisme par les équations de Langevin	14
(2.8)	Mise à jour de l'état d'une cellule d'un AC	32
(4.1)	Luminance	52
(4.2)	Contraste local	53
(4.3)	Contraste global	53
(4.4)	Diffinition d'une région	56
(4.5)	Dérivée par gradient (première dérivée)	61
(4.6)	Dérivée par le Laplacien (Dérivée seconde)	62
(4.7)	Les deux dérivées (gradient et le Laplacien)	63
(5.1)	Critères de la fonction de Fitness, la non égalité [SLKa2008]	82
(5.2)	Critères de la fonction de fitness, la distance entre Observations	82
(5.3)	La fonction Métric	82
(5.4)	Critères de la fonction de Fitness de Rosin	82
(5.5)	Critères de la fonction de Fitness, déférence de distance	83
(5.6)	La fonction d'adaptation (Fitness)	83

Chapitre 1

INTRODUCTION GÉNÉRALE

1.1 Contexte du travail de recherche

Le phénomène d'émergence dans les systèmes complexes est l'une de ces notions clefs qui commence à être une des solutions pour la résolution des problèmes difficiles à résoudre.

La propriété d'émergence est la résultante directe de la complexité des interactions à l'intérieur du système. Les éléments (composants) du système interagissent de façon locale. Les interactions en elles même sont simples, mais le nombre important des éléments et le phénomène de rétroaction (feed-back) qu'a le système avec son environnement, produisent un comportement complexe et intéressant [Géor2003]. La notion d'émergence est définie de manière intuitive comme une propriété macroscopique d'un système qui ne peut pas être inféré à partir de son fonctionnement microscopique.

Ces interactions locales se produisent donc à un niveau microscopique du système. Mais les phénomènes que l'on observe au niveau macroscopique (phénomènes émergents) semblent ne présenter aucun rapport direct avec les phénomènes locaux.

Dans ce travail, l'accent sera mis sur l'application de cette notion d'émergence dans le traitement d'images. Particulièrement, nous nous sommes intéressés à utiliser cette technique pour le domaine de la détection de contours. L'idée principale est de modéliser l'image par un système complexe (SC). Un système complexe est défini grossièrement comme étant construit par un grand nombre de parties simples, interagissant mutuellement, capables d'échanger des stimulus avec leur environnement et à même d'adapter leur structure interne comme une conséquence d'une telle interaction. Ces interactions peuvent provoquer des comportements complexes, cohérents, émergents et ne peuvent pas être attribués aux sous-systèmes simples séparés, mais plutôt à un effet collectif. La plupart de ces systèmes ne sont pas susceptibles de la discussion mathématique, analytique et

peuvent seulement être explorés au moyen "des expériences numériques".

1.2 Problématique

Un contour est une frontière entre deux zones homogènes de l'image. Une région homogène est une partie de l'image contenant des pixels qui exhibent des caractéristiques semblables (intensité, couleur, texture, motif...). Le contraste inter-zones doit être relativement marqué. La frontière entre deux régions homogènes constitue ce que l'on appelle un "contour" [Fabr1998].

Dans la littérature relative au domaine du traitement d'image, on trouve une multitude de méthodes de détection de contours s'appuyant sur divers algorithmes [Cocq1995]. La plupart de ces méthodes se basent sur le calcul des dérivées directionnelles. D'autres méthodes abolissant le calcul des dérivées se basent plutôt sur des connaissances préalables de la nature et des caractéristiques de l'image, mais ces exigences limitent le champ d'application de la méthode [Cocq1995]. On se propose ici d'échafauder une méthode alternative. Faisant appel à l'évolution d'une population d'AC (qui utilise une règle locale simple) [Rosi2006], on obtient par émergence, après ce processus d'évolution guidé par un critère de la qualité de la détection de contours, l'AC capable de segmenter une image en contours.

Une possibilité pour résoudre le problème inverse [TBRW2005] est d'utiliser une stratégie de recherche stochastique, afin d'optimiser le résultat obtenu. Lors de l'utilisation d'un AC pour résoudre une tâche donnée.

La combinaison des ACs et des Algorithmes Génétiques (AGs), nous donne un outil puissant de résolution de problèmes, connu sous le nom d'Automates Cellulaires Evolutifs (ACEvs) [Mitt1996, Thom2005]. L'espace de recherche sera l'espace des règles d'interaction locales. La recherche sera guidée par un critère de qualité calculé au niveau global, qui reflèterait l'adéquation du système au problème posé [BarY1997].

1.3 Contribution et objectifs du travail

Dans cette thèse, nous proposons une projection de la résolution de problème par émergence pour le problème de détection de contours [SLBM2007, SLKa2008, SLKa2011], qui est une tâche primordiale dans le domaine de traitement d'images. Nous allons découvrir les meilleures règles (paquets) qui assurent la détection optimale de contours dans les images en couleurs, en représentant l'image comme étant un ensemble de ACs adjacents.

Pour atteindre cet objectif, nous utilisons les ACEvs dans les règles locales de AC émergent (versus trouvées à la main) par l'évolution d'une population d'AC candidats, par le biais d'une stratégie d'optimisation basée sur les AGs [SLKa2008A, SLKa2009]. Cet ensemble de règles locales de AC détermine l'état futur de chaque AC de façon synchrone (indépendante) en se basant seulement sur l'état de son voisinage en cours. L'ensemble optimal de transitions est recherché par l'AG, assurant la meilleure détection de contours. cette approche présente un algorithme simple et intrinsèquement parallèle [SLKa2008, SLKa2011] pour la détection de contours.

1.4 Structure et contenu de document

Le reste de la thèse est organisé en sept chapitres. Le second chapitre est une description détaillée des systèmes complexes ainsi que ses sources d'inspiration (naturelle). Il présente un ensemble d'exemples du modèle de calcul inspiré de la nature.

Le troisième chapitre est l'étude du phénomène d'émergence et son utilisation pour la résolution des problèmes dans le cadre informatique en particulier, les propriétés d'un système émergent, puis de diverses approches émergentes en rapport avec ces notions : les Automates Cellulaires Evolutionnaires (ACEvs), les Systèmes Multi-Agents Adaptatifs (SMAA)pproches (SMAA), les calculs émergents (CE).

Le quatrième chapitre présente les notions de base de segmentation d'images, ensuite les techniques utilisées pour la segmentation et la détection de contours. L'accent sera mis sur les approches de détection de contours, d'où nous allons utiliser leur principe pour arriver à l'objectif de notre approche proposée.

Le Cinquième et le dernier chapitre sont consacrés respectivement à la description détaillée de l'approche proposée GACEv-DC pour la détection de contours par émergence, ainsi que les résultats obtenus sur des images de synthèses sont réeles. L'analyse de ces résultats expérimentaux sera également discutée.

Nous essayons de présenter un ensemble d'idées visant à comprendre le phénomène d'émergence dans la résolution de problèmes difficiles en général, et à la détection de contour en particulier.

Enfin, nous clôturons notre thèse par une conclusion générale récapitulative, porteuse de perspectives très prometteuses.

Chapitre 2

COMPLEXITÉ ET SYSTÈMES COMPLEXES

Ce chapitre est une description détaillée des systèmes complexes ainsi que ses sources d'inspiration (nature). Il présente un ensemble d'exemples du modèle de calcul inspiré de la nature. Dans une première partie, nous allons définir les systèmes complexes. Nous décrivons leurs propriétés et nous présentons les différents types de modélisations des systèmes complexes. Puis, nous introduisons les concepts de base de la vie artificielle et des systèmes multi-agents (SMA). Dans la deuxième partie, nous exposerons quelques exemples du modèle de calcul naturel.

2.1 Introduction

De nombreux systèmes constitués d'un grand nombre d'entités en interaction sont qualifiés de complexes lorsqu'un observateur ne peut prévoir leur comportement ou leur évolution par un raccourci de calcul. Une réaction chimique, comme la dissolution d'un grain de sucre dans du café, est simple car on connaît à l'avance le résultat : quelques équations permettent de décrire exactement l'évolution, sans avoir besoin d'attendre la fin de l'expérience pour savoir comment cela se passe. Au contraire, les cellules nerveuses du cerveau, une colonie de fourmis ou les agents qui peuplent un marché économique sont autant des systèmes complexes car le seul moyen de connaître l'évolution du système est de faire l'expérience, éventuellement sur un modèle réduit.

De par la diversité des systèmes complexes, leur étude est interdisciplinaire. Deux approches complémentaires sont utilisées : certains scientifiques analysent un système complexe naturel particulier et cherchent à le comprendre, d'autres cherchent des méthodes et définitions générales applicables à de nombreux systèmes différents [Moig1999, Heno2003]. Dans ce chapitre, nous nous intéressons aux systèmes complexes de quelques composants élémentaires, interagissant localement l'un avec l'autre. Il est impossible de prédire son comportement. Cette impossibilité vient de nos limites de calcul : dans certains cas les calculs deviennent rapidement trop imprécis, quelle que soit la précision des données initiales, comme pour la prédiction météo.

Dans une première partie, nous allons définir les systèmes complexes. Nous décrivons leurs propriétés et nous présentons les différents types de modélisation des systèmes complexes. Puis, nous introduisons les concepts de base de la vie artificielle et les systèmes multi-agents (SMA). Dans la deuxième partie, nous exposerons quelques exemples du modèle de calcul naturel.

2.2 Généralités et définitions

Plusieurs définitions d'un système sont cités dans la littérature [Horn2001], nous citons ici la plus importante

2.2.1 Définition d'un système

Un système est un ensemble constitutif et des propriétés le structurant en tant que :

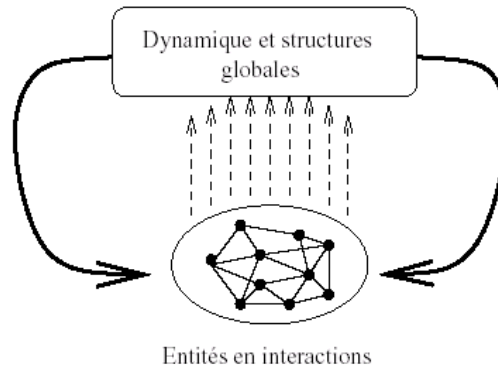


FIG. 2.1 – Typologie des systèmes.

- Ensemble constitutif d'entités en interactions mutuelles et en interaction avec un milieu extérieur ou environnement ;
- Propriétés caractéristiques conférant la structure de système [Mela2006] ;

La théorie des systèmes classe les systèmes en trois types :

2.2.2 Système simple

Un système est dit simple [Horn2001], s'il offre un comportement réductible à la somme des comportements de ses éléments constitutifs et de nature d'interaction linéaire.

2.2.3 Système compliqué

Les systèmes compliqués sont des systèmes que l'on peut réduire en éléments plus simples que l'on peut analyser séparément pour comprendre le système global. En outre, un système compliqué a un comportement non trivial, résultant de la composition linéaire des comportements non-triviaux de ses composants [Moig1999].

2.2.4 Système complexe

Un système complexe [Sloo1997], est une population d'un élément unique avec des attributs bien définis. De plus, si ces éléments ont des interactions non-linéaires, il se peut que l'on assiste à l'émergence d'un comportement global. Le comportement d'un système complexe ne peut être réduit à la composition linéaire des comportements de ses éléments constitutifs. Nous disons que "le tout est plus que la somme de ses parties". Un système complexe montre un comportement au niveau macroscopique, qui ne peut être expliqué par la simple composition des comportements de ses parties au niveau microscopique.

2.3 Niveaux de complexité

La nature des interactions entre les éléments du système permet de distinguer trois niveaux de complexité [Bato2005] (voir la figure ci-dessous) :

- Interactions hiérarchiques : des symétries qui permettent de l'appréhender facilement.
- Interactions en partie hiérarchisées : il faut dresser la liste de tous les interactions autorisées l'une après l'autre.

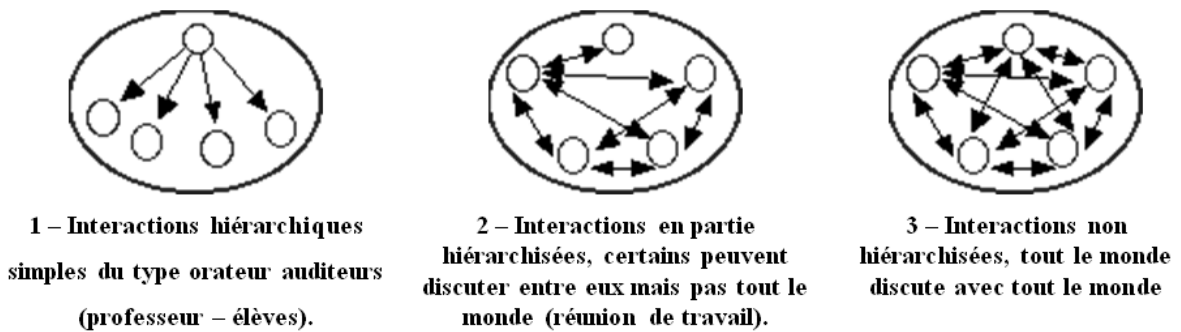


FIG. 2.2 – Types d'interactions entre les éléments d'un système.

2.4 Définitions de la complexité

On distingue deux définitions principales de la complexité [Bato2005, Abb2006] :

- *Complexité brute* "aléatoire", qui signifie : sans ordre, sans régularités.
- *Complexité effective* "organisée", qui signifie : structurée, riche en information.

2.4.1 La complexité brute (aléatoire)

La complexité brute d'un système est relative à la quantité d'information nécessaire pour le décrire [GeMa1997]. La complexité dépend donc du niveau de détail nécessaire à cette description.

Plus un phénomène est non prédictible et aléatoire, plus sa complexité brute augmente. La quantité d'information d'un message est liée à sa taille et à sa rareté. Par exemple on obtient une information plus importante en connaissant à l'avance les numéros gagnants du loto, qu'en connaissant à l'avance le résultat du tirage d'une pièce de monnaie. "La quantité d'information est donc liée à son volume et à sa probabilité d'occurrence".

Soit un système qui a plusieurs états possibles. On veut identifier l'un des états courants

de ce système. Alors, le nombre de bits nécessaires pour le spécifier est relatif au nombre d'états possibles Ω .

Le nombre de bits d'information nécessaires I sera :

$$I = \text{Log}_2(\Omega) \quad (2.1)$$

Cette équation ressemble beaucoup à celle de l'entropie¹ d'un système :

$$S = k \ln(\Omega) = k \ln(2)I \rightarrow S \approx I, \quad (2.2)$$

Constante de Boltzmann, $k=1.381023$ Joule/deg *Kelvin*.

L'entropie H d'une source d'informations, est la moyenne de la quantité d'information délivrée par celle-ci, tel que p_i est la probabilité d'apparition du symbole i de l'alphabet de la source. L'entropie est donc une mesure du désordre d'un système en thermodynamique.

$$H = - \sum_i p_i \log_2(p_i) \text{ (enbits/symbole)} \quad (2.3)$$

Le premier principe de la thermodynamique dit que l'énergie d'un système isolé reste constante. Le second principe de la thermodynamique, plus subtil, nous indique que l'entropie d'un système isolé ne peut qu'être constante, ou augmenter.

Le second principe indique que le désordre d'un système isolé ne peut qu'augmenter, ce que l'on voit à l'oeuvre tous les jours. L'état ordonné d'un système est, soit unique, soit assez peu probable, tandis que l'état désordonné est très probable car possible dans beaucoup plus de combinaisons.

La quantité d'information est liée à sa rareté : $I = - \log_2(p_i)$; plus la probabilité est faible, plus l'information est grande.

L'entropie, elle est également liée à la probabilité : plus un état macroscopique est probable plus son entropie est forte. Ainsi en termes d'information, l'entropie est une mesure de l'ignorance : au fur et à mesure qu'elle s'aggrandit, le désordre augmente et la reconnaissance des informations précises devient rare plus, elle faiblit, plus le nombre d'informations augmente.

¹La tendance du système à perdre de l'énergie et de l'ordre et à rester dans des états homogènes. 2ème loi de la thermodynamique.

La complexité d'un système est proportionnelle à son entropie, que l'entropie est la quantité d'information nécessaire pour décrire son micro-état (états microscopiques de ses composants). Celle-ci est maximale pour un système dans un état d'équilibre². Alors que la complexité d'un système caractérise la quantité d'information nécessaire pour décrire son macro-état. Plus il y a de corrélations entre les micro-états des éléments du système, plus celui-ci est complexe. L'entropie du système doit donc être très réduite.

2.4.2 La complexité effective (organisée)

La complexité effective d'un système, est liée à la description des régularités de celui-ci, par un observateur intelligent (être humain) [GeMa1997, GeMa2003]. C'est à dire qu'un système apparemment très complexe, peut être résumé dans ces grandes lignes par un ensemble de règles simples (régularités du système).

L'approche de la complexité organisée, se base sur la description des régularités d'un système, comme observées par un système adaptatif complexe qui l'examine.

Dans la complexité organisée, on peut distinguer des systèmes tels les corps célestes, les montagnes, un tas de sable, et autres systèmes qui sont longs à décrire, mais dont on a pu extraire la loi sous-jacente. Ces systèmes sont en évolution sous l'effet d'une dynamique assez lente et ne restent donc pas figés dans un état d'immobilité.

On doit les distinguer des systèmes adaptatifs complexes qui eux sont capables d'apprendre quelque chose sous la pression de l'environnement : l'évolution biologique, le fonctionnement du système immunitaire, l'apprentissage et la pensée chez les animaux et le comportement des investisseurs sur les marchés financiers.

Le point commun entre tous ces processus consiste à décrire de manière générale ce qu'est un système adaptatif complexe.

²La tendance du système de rester dans un état stable qui n'est pas aisément modifiable, un attracteur. Traditionnellement, les systèmes en équilibre, en physique, ne reçoivent pas d'apport énergétique et ont une entropie maximale. Ils sont sujets à un attracteur ergodique (passent par tout les point de l'espace des états de façon équiprobable. Caractérisant une entropie forte, et en total opposition avec le comportement d'un SC).

2.5 Systèmes complexes

Dans cette section, nous présentons une description détaillée des systèmes complexes. Nous commençons par un bref historique des systèmes complexes et nous continuons par un ensemble de définitions du système complexe. Ensuite, des propriétés d'un système complexe seront exposées.

2.5.1 Définitions de système complexe

Dans cette section, nous reprenons quelques définitions trouvées dans la littérature scientifique.

Jean-Louis Le Moigne [Moig1999], fait la distinction entre systèmes complexes et systèmes compliqués qui ne sont que des systèmes que l'on peut réduire en éléments plus simples que l'on peut analyser séparément pour comprendre le système global. Tandis que les systèmes complexes sont la somme des éléments fait émerger de nouvelles propriétés qui ne sont pas dans les éléments eux-même. Slood [Sloo1997] a défini un système complexe comme est une population d'un élément unique avec des attributs bien définis. De plus, si ces éléments ont des interactions non-linéaires, il se peut que l'on assiste à l'émergence d'un comportement global.

Nous considérons que les systèmes complexes sont des ensembles d'éléments en relation. Lorsqu'on fait la somme de ces éléments et de leurs relations, de nouvelles propriétés émergent sans qu'ils soient observables à un niveau macroscopique, alors qu'ils sont produits par l'interaction des éléments du système à un niveau microscopique [GMPG2003, Mois2001]. Ce qui montre qu'il n'existe pas de définition formelle largement acceptée de ce qu'est un système complexe [Moig1999].

2.5.2 Propriétés d'un système complexe

Le comportement des systèmes complexes représenté par un ensemble de propriétés considérées comme des caractéristiques des systèmes complexes sont :

Emergence : plutôt que de prétendre donner une définition exacte et exhaustive de

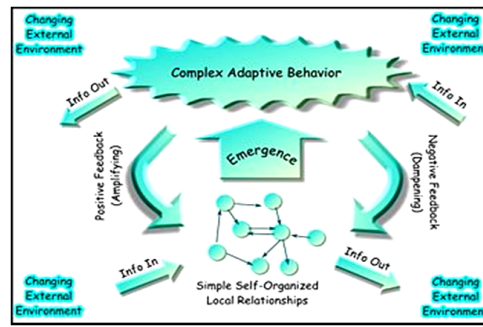


FIG. 2.3 – Entités en interaction pour un dynamique et structure globale, adapte de [Amma2007].

l'émergence, nous donnons, à partir de définitions tirées de la littérature, les particularités suivantes semblent fondamentales pour cerner de façon précise la notion d'émergence [Géor2003].

Nouveauté : d'une part, l'émergence présuppose qu'il y a apparition de nouveauté - propriétés, structures, formes ou fonctions -, et d'autre part, elle implique qu'il est impossible de décrire, d'expliquer ou de prédire ces nouveaux phénomènes en termes physiques à partir des conditions de base définies aux niveaux inférieurs [Vijv1997].

Interdépendance des niveaux : des structures et/ou des fonctions apparaissent à un niveau macroscopique sans que la seule observation des propriétés des constituants permette de les prédire. Langton [Lang1990] définit l'émergence en termes de relation de feedback entre les niveaux dans un système dynamique. Les micro-dynamiques locales causent les macro dynamiques et les macro-dynamiques globales contraignent les locales.

Auto-organisation : l'auto-organisation est généralement définie comme l'ensemble des mécanismes permettant à un système d'exhiber des structures de formes ou de fonctionnements sans contrainte explicite ni intervention quelconque de l'extérieur du système [Géor2003, Marc1997].

La non linéarité et l'irréductibilité : [Ali1997] définit l'émergence en termes d'irréductibilité des propriétés associées à une théorie de haut niveau à des propriétés associées à des composants dans une théorie de plus bas niveau.

L'ordre et le chaos : plusieurs comportements possibles sont en compétition, certains sont simples, d'autres sont chaotiques (désordonnés). Le système est souvent à la frontière

entre les deux et alterne ces deux types de comportement [Iani1995, Moig1999].

2.5.3 Modélisations des systèmes complexes

Ils existent de nombreuses manières d'étudier et de modéliser les systèmes complexes. [Mois2001] a essayé de répertorier les différents formalismes utilisés actuellement. Le choix d'un formalisme dépend du problème à étudier.

2.5.3.1 Formalisme par les équations différentielles

Les équations différentielles sont de la forme :

$$\frac{dx}{dt} = f(x, t) \quad (2.4)$$

où x est une grandeur scalaire ou vectorielle, et f est une fonction suffisamment régulière. La fonction f est généralement non linéaire.

2.5.3.2 Formalisme par les équations aux dérivées partielles

Lorsque nous désirons modéliser un système en préservant ses caractères continus, on fait appel aux équations aux dérivées partielles de la forme :

$$\frac{x}{t} = f(x, t) \quad (2.5)$$

La différence par rapport aux équations différentielles est que le temps n'est plus découpé en intervalles mais il est continu.

2.5.3.3 Formalisme par les équations maîtresses

L'équation maîtresse est une équation qui décrit l'évolution temporelle de la probabilité p_n que le système soit dans l'état n à l'instant t .

$$\frac{dp_n(t)}{dt} = n' (W_{nn'} p_{n'}(t) - W_{n'n} p_n(t)) \quad (2.6)$$

où les états du système sont indexés par n et n' , où W est une matrice de transition : $W_{nn'}$ est la probabilité qu'une transition de l'état n' vers l'état n . La plupart des cas pouvant être résolus analytiquement ne concernent que le taux de transition indépendant

du temps et dépendant uniquement de n et n' .

2.5.3.4 Formalisme par les équations de Langevin

Nous utilisons ce type d'équation lorsque l'on veut intégrer la notion de bruit à une équation différentielle. Dans sa formulation la plus générale, l'équation de Langevin s'écrit :

$$\frac{dy}{dt} = A(y) + B(y)L(t) \quad (2.7)$$

Dans cette équation, y est une variable continue supposée pouvoir varier entre $-\infty$ et $+\infty$, et les fonctions-coefficients $A(y)$ et $B(y)$ sont des fonctions réelles différentiables avec $B(y) > 0$ pour tout y . $L(t)$ est un processus stochastique associé au bruit blanc Gaussien.

En arrivant au dernier type de modélisation de système complexe qui devient l'objet de notre étude d'application.

2.5.3.5 Formalisme par les Automates Cellulaires

Les AC [Wolf1984] sont des réseaux de cellules à une, deux, trois ou n dimensions. Dans un automate cellulaire, le temps et l'espace sont représentés de manière discrète. Chacune des cellules contient un automate à états finis qui calcule son état en fonction d'un voisinage, par exemple ses deux cellules adjacentes dans le cas d'un automate à une dimension, ou bien ses huit cellules adjacentes dans le cas d'un automate à deux dimensions. Chacun des automates est défini par un ensemble de règles locales qui vont spécifier son comportement, en fonction de son voisinage. Des voisinages différents peuvent conduire à des états différents de l'automate considéré. Les sections suivantes consacrées donnent une vue plus illustrée sur ces concepts de base.

2.6 Classification des SCs

Plusieurs niveaux de complexité sont cités dans la littérature de classification des SCs [Luca2000, Abb2006], la plus importante :

2.6.1 Type 1 - Complexité Statique

L'attracteur point fixe Caractérise les systèmes statiques qui affichent des structures hiérarchiques répétitives et des motifs récurrents. C'est le cas des fractales qui sont des systèmes clos en équilibre (Voir la figure ci-dessous).

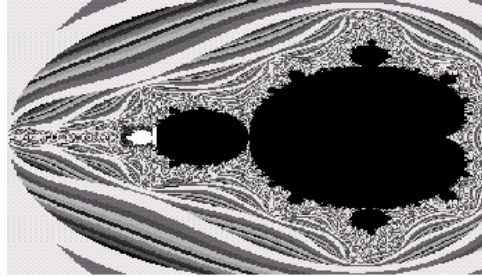


FIG. 2.4 – Fractale de Mandelbrot.

2.6.2 Type 2 - Complexité Dynamique

Certaines séquences cycliques sont observées dans le comportement du système. L'étude de ces séquences peut aider à qualifier et à prédire le comportement futur du système. Caractérisés par l'attracteur cyclique, plusieurs cycles superposés peuvent être identifiés.

2.6.3 Type 3 - Complexité Evolutive

Elle qualifie la capacité de changement et d'adaptation des systèmes caractérisés par l'attracteur chaotique (ergodique). Ces systèmes effectuent une recherche dans l'espace des états. Plus encore, ils étendent l'espace de recherche en ajoutant de nouvelles possibilités (création de nouveaux composants dans le système). Inversement ils peuvent réduire l'espace de recherche en éliminant les cas de défaillance.

2.6.4 Type 4 - Complexité Auto-Organisatrice

C'est la forme la plus intéressante de complexité. Elle combine les contraintes internes des systèmes clos et l'évolution créative des systèmes ouverts. Le système est en co-évolution avec son environnement. Toute la question est de comprendre comment les composantes du système s'auto-organisent pour atteindre un objectif de haut niveau, dans le cadre des contraintes environnementales.

Ces systèmes opèrent aux frontières du chaos. Ils sont caractérisés par des boucles de rétroaction, qui leur donnent une dynamique non-linéaire. En combinant les trois types d'attracteurs, ils ont une structure plus riche et plus complexe.

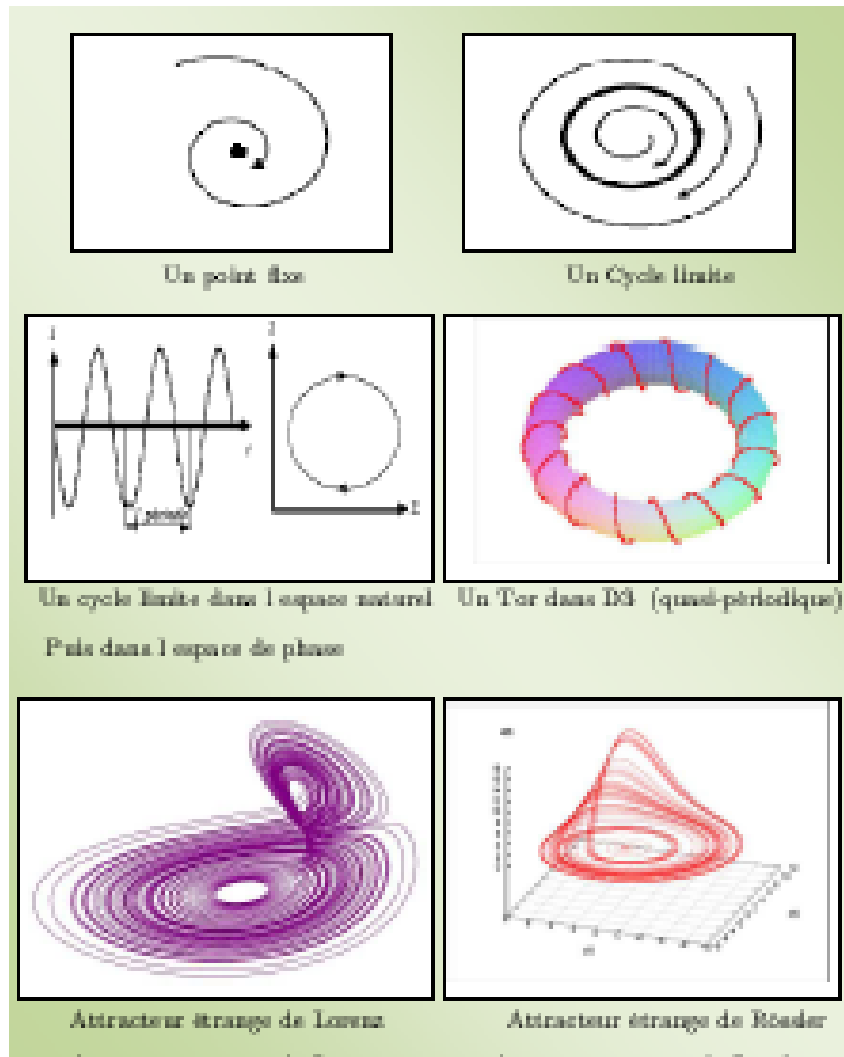


FIG. 2.5 – Différents types d'attracteurs : Point fixe, Périodique et Chaotique.

2.7 Caractéristiques des SCs

Voici les propriétés les plus courantes attribuées aux SCs qui se distinguent selon des règles récurrentes qui apparaissent dans la plupart d'entre eux.

2.7.1 Emergence

Un SC est un groupe ou une organisation composée d'une multitude d'entités en interaction. Les interactions entre elles, produisent un comportement global à l'échelle du système, qui ne peut être prédit simplement à partir des comportements individuels de ses éléments. De tels effets collectifs sont appelés des **comportements émergents** [MiNe2001].

2.7.2 Changement du niveau d'intégration

Lorsque l'on change de niveau de description, il se produit des phénomènes de réorganisation et de simplification. L'étude des SCs nous permet de remarquer une sorte de mutation du comportement observé lors du changement des différents types d'attracteurs : attracteurs réguliers et attracteurs étranges [GeMa1997].

2.7.3 Boucles de rétroaction et Non-Linearité

"Un SC est vu comme un système non linéaire, dont le résultat des interactions entre composants excède les contributions individuelles de chaque composant".

"Les propriétés du système ne sont pas une composition linéaire des propriétés de ses composants. Les interactions épistatiques³ entre les parties du système, imposent une étude globale et non-réductionniste (analytique) de celui-ci" [Luca2000].

Le retour des sorties du système sur ses entrées définit le phénomène de Feed-Back (rétroaction). Ce phénomène produit une force non-linéaire. En systémique, la boucle de rétroaction double réfère à la boucle de rétroaction des mécanismes classiques [Amma2007].

- *Schéma de boucle de rétroaction classique* : on "joue un jeu" et on apprend à agir sur les variables.

- *Schéma de boucle de rétroaction simple et double* : on projette un jeu en intervenant à la fois sur les variables et sur le processus (voir la figure ci-dessous).

³Epistasie : l'effet d'une variable sur une autre, une interdépendance des composants plutôt qu'une indépendance

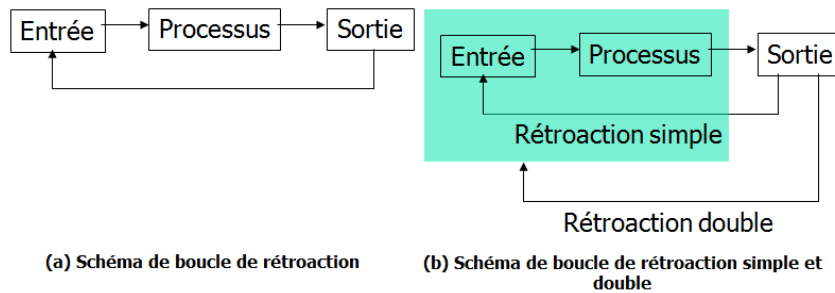


FIG. 2.6 – Schéma de boucle de rétroaction simple et double, adapte de [Amma2007].

2.7.4 Indéterminisme

Poincaré en 1908 écrivait dans son livre : "Science et méthodes" : *"Si nous connaissions exactement les lois de la nature et la situation de l'univers à l'instant initial, nous pourrions prédire exactement la situation de ce même univers à un instant ultérieur. Mais, alors même que les lois naturelles n'auraient plus de secret pour nous, nous ne pourrions connaître la situation initiale qu'approximativement. Si cela nous permet de prévoir la situation ultérieure avec la même approximation, c'est tout ce qu'il nous faut, nous disons que le phénomène a été prévu, qu'il est régit par des lois ; mais il n'en est pas toujours ainsi, il peut arriver que de petites différences dans les conditions initiales en engendrent de très grandes dans les phénomènes finaux ; une petite erreur sur les premières produirait une erreur énorme sur les derniers. La prédiction elle devient impossible et nous avons le phénomène fortuit."*

Ce phénomène est connu sous le nom de *"chaos déterministe"* qu'il est essentiellement causé par le phénomène de Feed-Back.

"Les boucles de rétroaction induisent des divergences de comportement du système, à partir de données d'entrée très similaires. En plus, des convergences peuvent être observées pour des entrées sensiblement distinctes. Ceci est dû au fait que le système est dans un état proche du chaos, caractérisé par un mélange d'attracteurs" [Luca2000].

2.8 Exemples de SCs naturels et/ou artificiels

Pour illustrer les définitions précédentes, on présente une liste d'exemples de SCs naturels et/ou artificiels.

2.8.1 Systèmes Complexes Naturels (SCNs)

Plusieurs systèmes Complexes Naturels sont cités dans la littérature du domaine, nous citons ici les plus importants.

2.8.1.1 Le tas de sable

D'après le physicien danois Per Bak adapte de [Peba1996] dans sa proposition en 1987 théorique de la complexité modélisée à l'aide d'un tas de sable réel qui se traduit par un modèle évolutif pour toute une classe de phénomènes dits "auto-organisés" comme par exemple le scénario de l'évolution phylogénique des espèces vivantes.

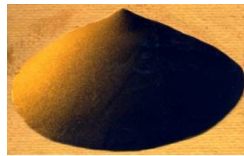


FIG. 2.7 – Le tas de sable réel.

A. Caractéristique de la complexité du tas de sable

Si l'on en juge par ses propriétés physiques, le sable sec est en quelque sorte à mi-chemin entre les fluides et les solides. Comme un liquide, il épouse les formes du récipient qui le contient et se déverse en masse lorsqu'on retourne énergiquement ce dernier sur une table.



FIG. 2.8 – Les trois états granulaires.

En revanche, sur cette surface, le sable forme un tas immobile comme un solide (voir la figure ci-dessus).

Le frottement des grains de sable de forme quelconques cause des interactions non linéaires entre ces grains, ce qui définit la configuration d'un SC (voir la figure ci-dessous).

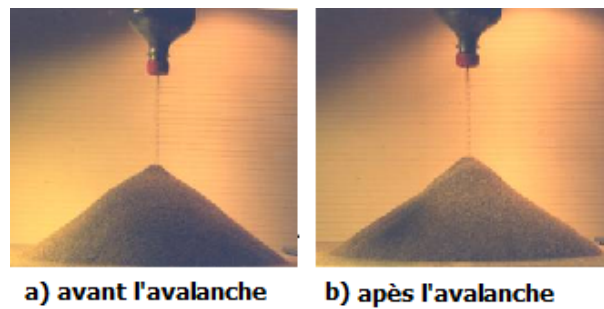


FIG. 2.9 – Avalanches et affaissements du tas de sable.

B. Le contenu d'un silo à grains

Pour mesurer le contenu d'un silo à grains en mesurant le poids sur le fond et lorsque l'on effectue des mesures de pression sous un tas de sable, on retrouve les conséquences de ces effets de voûte. Si le tas est construit avec un entonnoir, les voûtes rejettent le poids des grains vers l'extérieur et la pression au centre.

En revanche, lorsque le tas est construit en couches horizontales successives, ce minimum disparaît, et la répartition du poids des grains est à peu près uniforme dans toute la partie centrale.

Avec ce deuxième protocole, les voûtes ne privilégient ni l'intérieur ni l'extérieur du tas. La configuration des voûtes au sein d'un tas de sable dépend donc de la manière dont on l'a construit et sollicité.

2.8.1.2 Auto-organisation chez les insectes sociaux

Il existe de nombreux phénomènes émergents issus du comportement des insectes sociaux tels que les fourmis ou les termites. Cela n'a rien de surprenant car si l'on prend par exemple une colonie de fourmis, ce système, bien évident complexe, contient tous les "ingrédients" pour des phénomènes émergents [Géor2003] :

- Grand nombre d'entités autonomes,
- Nombreuses interactions,
- Fonctionnement simple des entités,
- Capacités proprioceptives et cognitives limitées,

- Système confronté à un environnement dynamique,
- Fonctionnements de haut niveau remarquable et nécessaires à la survie.

On interprète ainsi les phénomènes émergents chez les insectes sociaux à la lumière de quatre mécanismes principaux :

- L'existence d'interactions multiples,
- L'amplification par la rétroaction positive,
- La rétroaction négative,
- L'amplification des fluctuations.

A. Fourmis

Au départ, un grand nombre de fourmis se déplacent à l'extérieur du nid, plus ou moins au hasard. Elles recherchent de la nourriture. Tout au long de leur chemin, elles déposent une légère trace de phéromones. Si l'une d'entre elles découvre une ressource quelconque, elle retourne au nid en déposant une trace beaucoup plus intense (cette intensité dépend éventuellement de la richesse de la ressource). Cette trace tend à attirer les congénères qui, en la suivant, vont parvenir à la nourriture. Ils vont alors retourner au nid et renforcer la trace à leur tour. On assiste ainsi à la mise en place d'une boucle de rétroaction positive : la trace attire des individus qui renforcent la trace qui attire donc plus d'individus.

Les fourmis ne communiquent pas directement entre elles. Les échanges d'informations passent par une modification de l'environnement (en l'occurrence les concentrations locales de phéromones) [Renn2002].

B. Fourragement

Le fourragement de nourriture par une colonie de fourmis est un exemple d'application illustrant traditionnellement les systèmes multi-agents. Cette application montre que des entités autonomes au comportement primaire peuvent effectuer collectivement une tâche complexe : collecter de la nourriture de manière efficace.

Le fourragement par une colonie de fourmis met en jeu des comportements locaux : chaque fourmi a une connaissance limitée de son environnement. Elle ne sait pas où se

trouve la fourmilière ni où se trouve la nourriture, elle sait uniquement analyser son voisinage direct.

L'environnement lui aussi évolue de manière locale. Les phéromones qui sont des molécules présentes dans l'environnement se répandent localement.

2.8.1.3 Autres exemples naturels

Pour finir de clarifier l'émergence, on peut citer quelques exemples de phénomènes qui semblent être émergents à la vue de ce qui précède.

Les interactions entre des éléments simples donnent naissance à des structures complexes. Ces systèmes naturels composés d'individus autonomes exhibent des aptitudes à effectuer des tâches qualifiées de complexes, sans contrôle global. De plus, ils peuvent s'adapter à leur milieu, soit pour y survivre, soit pour améliorer le fonctionnement du collectif [Géor2003, Bato2005B].

- *Le système de Bénard* : une auto-organisation du système qui quitte son état chaotique et produit un réseau de courants hexagonaux appelés cellules de Bénard (voir la figure ci-dessous).

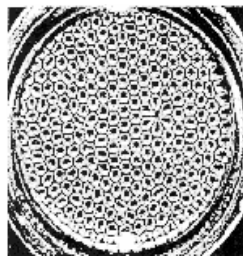


FIG. 2.10 – Réseau de cellules de Bénard.

- *Déplacement collectif* : le déplacement d'oiseaux migrateurs ou de bancs de poissons exhibe (voir la figure ci-dessous), le fait que la tâche collective est le résultat des interactions d'individus autonomes.

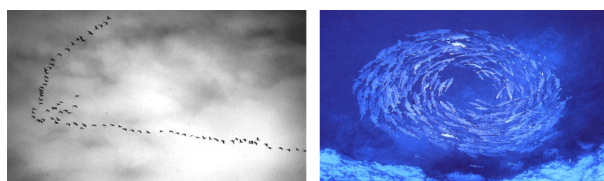


FIG. 2.11 – Déplacements collectifs d'oiseaux migrateurs et de bancs de poissons.

- *Système immunitaire* : l'ensemble de cellules dont le rôle est de défendre l'organisme contre les infections, lorsqu'il est envahi par un organisme étranger (virus , bactérie). Le système immunitaire réagit très vite pour s'en débarrasser.

2.8.2 Systèmes Complexes Artificiels (SCAs)

La vie artificielle est un regroupement de travaux hétéroclites, ayant pour point commun de s'inspirer directement et explicitement des caractéristiques du vivant ou de la nature [Renn2002]. Christopher Langton [Lang1990] définit la vie artificielle comme étant :

"L'étude de systèmes construits par l'homme qui présente des comportements caractéristiques des systèmes vivants".

Des processus comme la reproduction, la mort, la compétition pour des ressources, l'apprentissage, le développement et bien d'autres encore peuvent faire partie d'un système ALife. Ce domaine de recherche complète l'approche traditionnelle de la biologie, où le mode de fonctionnement est l'analyse des êtres vivants, tout en essayant de synthétiser des comportements vivants sur ordinateur.

La vie artificielle est généralement approchée de deux manières. D'une part, il s'agit de concevoir de nouveaux types de calculs avec des techniques inspirées de la biologie, ainsi ont développé des outils tels que : les réseaux neurones artificiels (RNA), les AGs et le calcul basé sur les essaims. D'autre part, nous utilisons les méthodes et les outils informatiques pour les recherches en Biologie, nous analysons les systèmes naturels et nous essayons de produire des modèles informatiques les représentant.

2.8.2.1 Boids de Reynolds

L'exemple célèbre d'un système de vie distribuée artificielle est bien le modèle du comportement de meutes. L'entité individuelle est appelée "Boid". Il représente un oiseau virtuel avec des capacités de vol de base. La meute (flocking), est une collection de Boids dans un monde simulé [Reyn1987].

Chaque Boid a une perception locale de son environnement local. Il peut seulement connaître ses voisins qu'ils l'entourent ou les obstacles. Le voisinage local d'un Boid est caractérisé par une distance (mesurée depuis le centre du boid) et un angle (mesuré de la

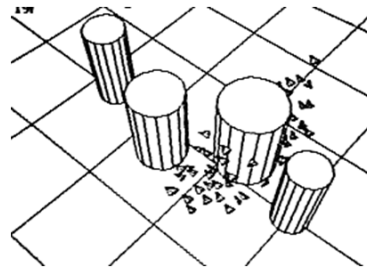


FIG. 2.12 – Les Boids de Reynolds.

direction de vol du boid). Les voisins à l'extérieur de ce voisinage sont ignorés.

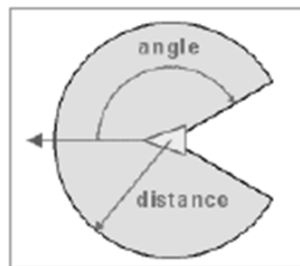


FIG. 2.13 – Le voisinage local d'un boid,

Le comportement de déplacement de Boids dans l'espace virtuel est un phénomène émergent qui résulte de leur interaction, chacun d'eux se contentant de suivre trois règles comportementales simples suivantes :

1. Alignement : adapter sa vitesse à la moyenne de celle de ses voisins.
2. Cohésion : aller vers le centre de gravité des boids voisins.
3. Séparation : maintenir une distance minimale par rapport aux autres objets dans l'environnement et en particulier les autres boids.



FIG. 2.14 – Comportements de direction : alignement, cohésion et séparation, Adapté de [Reyn2001].

Ces règles sont suffisantes pour que les boids adoptent un comportement semblable à ceux d'un vol d'oiseaux migrateurs. Les boids évoluent sans leader et sans contrôle global. Leur comportement est particulièrement fluide et semble très naturel lorsqu'ils évitent tous ensemble des obstacles en restant groupés. Parfois la meute se divise et produit deux

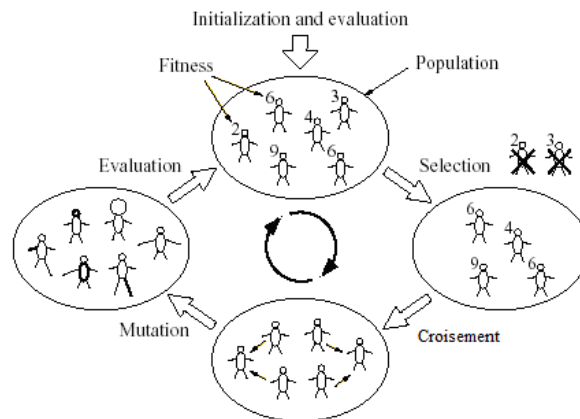


FIG. 2.15 – Schéma d'un Algorithme Evolutionnaire.

groupes qui se meuvent indépendamment l'un de l'autre. Parfois aussi, deux sous-groupes se réunissent pour produire un nouveau groupe.

Ces boïds ont apparu dans des films d'animation (par exemple "Batman" et "Le roi Lion"), et leurs concepts ont été appliqués pour travailler dans domaines aussi divers que l'analyse géographique l'exploration spatiale, et le traitement d'images.

2.8.2.2 Les Algorithmes Evolutionnaires

Les AEs [Gold1989] font partie du champ de l'IA. Il s'agit d'IA de bas niveau, inspiré par "*l'intelligence*" de la Nature. Trois types d'AE ont été développés dans les années 59 [Holl1975] : les Algorithmes Génétiques, les Stratégies d'Evolution, et la *Programmation Evolutionnaire*.

Dans les années 90, ces trois champs ont été regroupés sous le terme anglais *d'Evolutionary Computation*. Parmi les AE que nous venons de citer, nous avons choisi de traiter les Algorithmes Génétiques (AG) [Mitc1994].

2.8.2.3 Les Algorithmes Génétiques

Les AGs forment une famille très intéressante d'algorithmes d'optimisation. Ils ont été développés en première par John Holland [Holl1975] à l'université de Michigan ("*Adaptation in Natural and Artificial Systems*", 1975). Le principe des AGs [Gold1989] est de coder chaque solution potentielle d'un problème par un "*chromosome*". L'ensemble des chro-

mosomes ou "*individus*" forme alors la "*population*" qui va être amenée à évoluer au cours du temps. Une "*génération*" est l'état de la population à un instant t . La population évolue au cours des générations en suivant des lois de sélection, de croisement et de mutation. En informatique, nous parlons d'opérateur génétique. Dans AG de base, il existe trois opérateurs qui sont la sélection, le croisement et la mutation [Gold1989, Roma2003, Rous2003] dont les rôles sont représentés sur la Fig. 2.14.

A. La sélection : l'opérateur de sélection donne plus de chance aux "*bons*" individus de participer à la génération suivante en fonction d'un certain critère, la *fitness*.

B. Le croisement : l'objectif de croisement est de combiner des chromosomes à partir de parents présentant des qualités pour obtenir de meilleurs individus. Le croisement consiste à mélanger, au niveau du génome, des gènes de deux individus parents pour générer deux enfants. La taille de la population reste constante.

- **Le croisement par coupure :** le génome des deux parents est sectionné en un même endroit, choisi le plus souvent aléatoirement. Le nombre de points de coupure est un paramètre de l'opérateur. Les fragments sont recombinaés pour constituer les enfants comme l'illustre la Fig. 2.16.
- **Le croisement uniforme :** dans le cas du croisement uniforme, un masque binaire aléatoire de même longueur que le génome d'un chromosome utilisé. Si le bit i du masque est à 1, alors l'enfant 1 prend le bit i du parent 1, et l'enfant 2 prend le bit i du parent 2. Si par contre c'est un 0, c'est l'inverse qui est effectué. Le principe est décrit la Fig. 2.16.

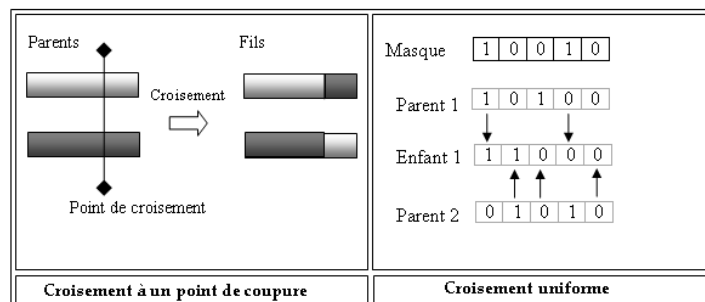


FIG. 2.16 – L'opérateur de croisement.

C. La mutation : elle effectue une modification des gènes des chromosomes des enfants. En fonction du problème à résoudre, à chaque individu est associé un '*degré d'adaptation à son environnement*', appelé *fitness*. Après croisement et mutation, une nouvelle génération est construite en conservant les individus de la population précédente ayant une propriété de fitness particulière, jusqu'à la convergence vers une solution optimale.

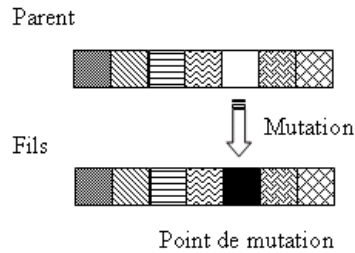


FIG. 2.17 – L'opérateur de mutation.

D. Le processus d'optimisation

1. Initialisation : générer un ensemble de solutions initiales.
2. Phase de mise à jour.

A. Evolution :

- Sélection : choisir avec une probabilité proportionnelle à leur qualité une liste d'individus.
- Reproduction : générer à partir de cette liste de nouveaux individus à l'aide des opérateurs génétiques.
- Remplacement : éliminer avec une probabilité inversement proportionnelle à leur qualité certains individus.

B. réactualisation de la meilleure solution.

C. allez en A tant que Nombre de Générations \leq Valeur pré-déterminée.

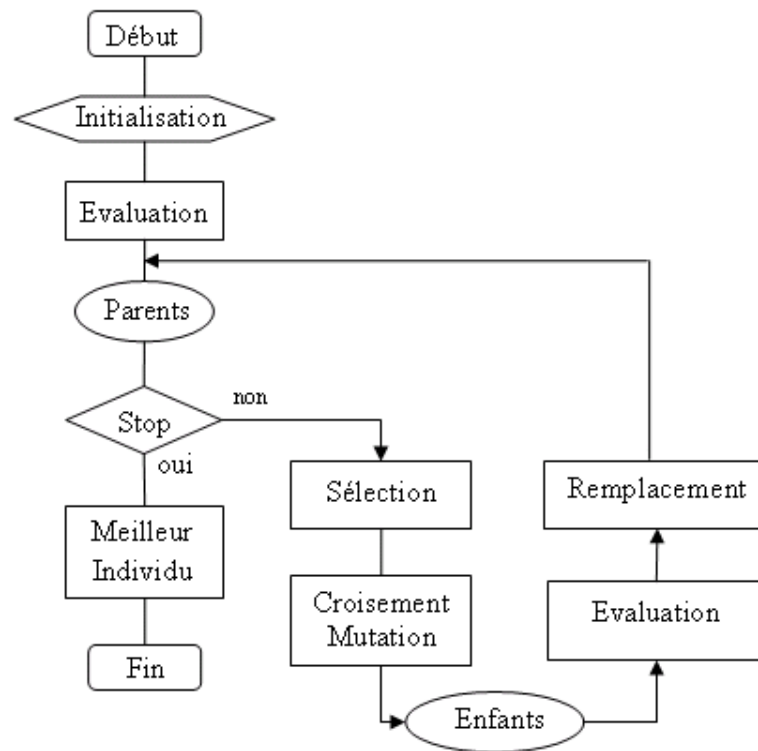


FIG. 2.18 – Principe de base d'un AG [Roma2003].

2.8.2.4 Les Systèmes Immunitaires

Plusieurs systèmes Immunitaires Naturels sont cités dans la littérature [DCVZ2000], nous citons ici les plus importants :

A. Les Systèmes Immunitaires Naturels

Le Système Immunitaire Naturel (**SIN**) est la collection de cellules et d'organes qui représente un mécanisme d'identification capable de percevoir et de combattre aussi bien le dysfonctionnement de nos propres cellules que les agents exogènes, appelés : antigènes.

B. Les Systèmes Immunitaires Artificiels

Les systèmes immunitaires artificiels (SIA) sont des systèmes adaptatifs inspirés du système immunitaire biologique pour résoudre certains problèmes.

Les ISAs sont composés de méthodologies intelligentes, inspirées du système immunitaire

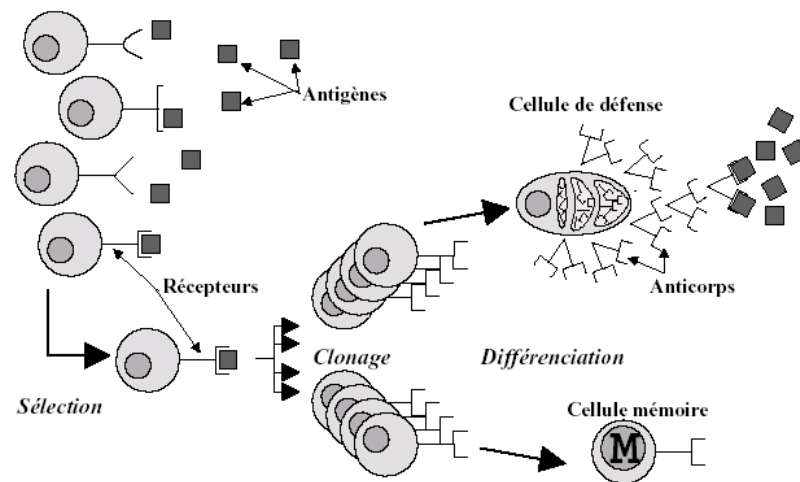


FIG. 2.19 – La sélection par clonage des lymphocytes, présentant des récepteurs spécifiques d'un antigène, se différencie en cellule mémoire ou en cellule participant à la défense active de l'organisme par le biais d'anticorps.

naturel, pour la résolution de problèmes du monde réel.

L'utilisation des capacités naturelles du système immunitaire pour accroître la convergence des algorithmes génétiques classiques. Et cela par :

- Des mécanismes évitant la convergence prématurée.
- Ainsi que le maintien de la diversité.

Dans les SIAs, le problème est traité comme étant l'antigène, quant aux solutions elles sont modélisées par les anticorps.

2.8.2.5 Les réseaux de Neurones

- Le réseau de neurones artificiels est un réseau fortement connecté de processeur élémentaire fonctionnant en parallèle.
- Le Neurone formel est une modélisation mathématique qui hérite le principe du fonctionnement du neurone biologique [Russ1995].

2.8.2.6 Intelligence d'essaims de particules

Les algorithmes *d'optimisation par essaim de particules* "*Particle Swarm Optimization*" (*PSO*), trouvent leur origine dans les bois de l'infographiste Craig REYNOLDS [Renn2002]. Analysant les nuées d'oiseaux, REYNOLDS [Reyn2001] a proposé de les si-

muler à partir de trois règles simples *d'application locale* :

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

Il est possible d'y ajouter une règle de propension à rejoindre un point donné dans l'espace (*perchoir*). Les rassemblements animaux présentent divers avantages adaptatifs relevant au moins pour partie de l'échange d'informations. De la même manière que les AEs s'inspirent des mécanismes évolutifs pour mettre en oeuvre des procédures d'optimisation, PSO s'inspire des phénomènes de rassemblement et de nuée, considérant qu'en tant que processus adaptatifs, ils sont potentiellement porteurs de capacités d'optimisation.

L'approche PSO repose sur deux règles simples [Renn2002] :

1. Chaque individu se souvient du meilleur point (le plus proche de l'objectif) par lequel il est passé au cours de ses évolutions et tend à y retourner.
2. Chaque individu est informé du meilleur point connu au sein de la population prise dans son ensemble et tend à s'y rendre.

L'une des premières applications a été l'entraînement d'un réseau de neurones devant réaliser la fonction booléenne XOR (*ou exclusif*) et utilisant treize paramètres. L'essaim devait ainsi se déplacer au sein d'un hyperespace à treize dimensions. Chacune de ces dimensions représente alors l'espace des poids possibles pour une connexion neurale donnée.

2.8.2.7 Les Automates Cellulaires

Les Automates Cellulaires (ACs) sont un outil mathématique puissant, pour modéliser les SCs. Inventés par deux des plus renommés mathématiciens, Ulam et Von Neumann [Ulam1974, Neum1966].

2.8.2.7.1. Une brève histoire des ACs

On distingue trois périodes essentielles dans l'histoire des ACs : leur création, qui fut une retombée des travaux de Von Neumann et d'Ulam [Ulam1974, Neum1966] sur l'auto-

reproduction ; le développement de l'AC 'Jeu de la Vie' et enfin les explorations d'espaces d'automates cellulaires, qui visent à une compréhension globale du monde des AC.

A. La naissance des automates cellulaires

Les ACs ont été inventés par Stanislas Ulam (1909-1984) [Ulam1974] et John Von Neumann (1903-1957) [Neum1966] à la fin des années 40 au Los Alamos National Laboratory.

En 1948 Von Neumann, proposa un article intitulé "*Théorie générale et logique des automates*". En 1949, il donna une série de cours sur le thème "*Théorie et organisation des automates complexes*". Une des questions centrales abordées était de savoir s'il était possible de concevoir une machine capable de s'auto-reproduire. La solution à ce problème vint de son collègue Stanislaw Ulam, qui s'intéressait aux "*objets géométriques définis de façon récursive*". Ces objets provenaient de jeux aux règles simples dans lesquels on pouvait voir des figures (patterns) se développer, se reproduire, combattre pour une portion de territoire et mourir [Fate2001].

Ces jeux se déroulaient dans un univers "cellulaire", composé d'une matrice infinie où les cellules, régulièrement réparties, peuvent être dans un état passif ou actif. Les figures de cet univers étaient composées des cellules actives, et à tout moment, le devenir de chaque cellule était dicté par l'état des cellules avoisinantes. Ulam s'aperçut que l'analyse de ces figures défiait le bon sens : elles semblaient évoluer dans un monde qui leur était propre avec des lois bien spécifiques. Il suggéra alors à von Neumann d'utiliser un tel monde abstrait pour pallier les difficultés pratiques qui se posaient pour la construction de l'automate auto-reproducteur. Ce monde serait suffisamment complexe pour pouvoir simuler les opérations élémentaires des machines et en même temps construit de façon à ce que les lois de la physique qui gouvernent ce monde se réduisent à quelques règles simples.

Le problème de l'autoreproduction : Von Neumann aboutit à l'idée qu'un automate autoreproducteur devrait comporter un "constructeur universel" capable de fabriquer n'importe quelle machine (cellulaire) à partir d'une "description" qui lui serait fournie. Dans le cas particulier où l'on fournirait la description du constructeur universel au constructeur

universel lui-même, il y aurait autoreproduction. Le système autoreproducteur (SAR) est constitué du constructeur universel (CU) et de sa propre description (DESC). Or cette description ne peut être la description du constructeur universel seulement, elle doit être la description de tout le système, et donc être en particulier une description de la description.

Le système autoreproducteur est défini par l'équation : $SAR = CU + DESC(SAR)$, ce qui paraît a priori insoluble étant donné l'existence d'une régression à l'infini (récursivité à gauche : quand on remplace SAR à droite par le contenu de l'équation).

Le problème fut résolu par Von Neumann [Neum1966] en rajoutant une troisième unité : une machine de Turing, appelée superviseur, qui orchestre le processus. L'utilisation du superviseur évitait la régression à l'infini en distinguant deux phases :

- L'ensemble (constructeur universel + superviseur), le "copieur-superviseur" réalise une copie de lui-même dans une région vide de l'espace en lisant la description, c'est la phase d'interprétation.

- Le superviseur comprend qu'il ne faut plus que la description soit interprétée ; celle-ci est considérée comme un ensemble de données et recopiée littéralement pour rebâtir le système initial.

En 1952, Neumann proposait une version qui utilise 29 types de cellules différentes. L'état de chaque cellule au temps t était déterminé uniquement par l'état des quatre cellules adjacentes et celui de la cellule centrale au temps $t - 1$. Ce voisinage est d'ailleurs nommé voisinage de Von Neumann.

B. L'Automate Cellulaire "Jeu de la Vie"

Dans le numéro d'Octobre 1970 de Scientific American, Martin Gardner publie un article intitulé *"Les combinaisons fantastiques du Jeu de la Vie de John Conway"*. Conway a inventé un automate cellulaire qui a la particularité suivante : des figures peuvent croître et atteindre une grande taille. Cet automate cellulaire se nomme "Game of Life".

Les règles du Jeu de la Vie sont simples. Les cellules peuvent se trouver dans deux états qui sont : vivant / mort. L'espace cellulaire est composé de cellules qui se trouvent dans l'état mort au départ, sauf pour un nombre fini d'entre elles. L'évolution de chaque cellule est déterminée en fonction du nombre Nv de cellules vivantes se trouvant dans les huit cases adjacentes à une cellule. Les règles sont :

- " Une cellule vivante meurt (devient vide) pour $Nv = 1$: isolement de cellule.
- " Une cellule vivante meurt pour $Nv = 4$: surpeuplement autour de la cellule.
- " Une cellule morte peut devenir vivante pour $Nv = 3$: reproduction.

L'évolution des objets manipulés, bien que totalement déterminée par les fonctions de transition, est hautement imprévisible. On remarque ainsi qu'il n'y a pas de correspondance apparente entre la taille d'une configuration initiale et le temps qu'elle met pour se stabiliser. Par ailleurs, le simple fait d'ajouter ou d'enlever une cellule dans une configuration change son évolution de façon radicale.

2.8.2.7.2. Les automates cellulaires : Etude formelle

Un AC se définit à l'aide de deux types de caractéristiques : structurelles et fonctionnelles. Les premières concernent l'aspect topologique du réseau cellulaire, les secondes concernent l'aspect dynamique de l'évolution du réseau au cours du temps. Le réseau cellulaire peut en effet prendre corps dans des espaces différents, ayant une, deux ou trois dimensions ; l'arrangement des cellules en réseau pouvant aussi être variable (orthogonal, hexagonal).

Définition

Un AC est défini par la donnée d'un quadruplet $A = (C, S, V, F)$. $C = (c_1, \dots, c_{NCell})$ représente un ensemble de cellules où $NCell$ est constant au cours du temps. $S = (s_1, \dots, s_k)$ est l'ensemble fini d'états que va pouvoir prendre chaque cellule. L'état de la cellule c_i est noté $c_i(t)$. V est le voisinage entre cellules qui va structurer l'ensemble des cellules. Pour chaque cellule c_i , $V(c_i)$ est défini comme l'ensemble des cellules voisines de c_i . La fonction de transition locale F détermine le nouvel état d'une cellule en fonction des états perçus. La configuration de l'AC à l'instant t est le vecteur d'états $A(t) = (c_1(t), \dots, c_{NCell}(t))$.

Un AC évolue de $A(t)$ à $A(t + 1)$ en appliquant F à chacune des cellules en parallèle.

2.8.2.7.3. Caractéristiques des ACs

Plusieurs caractéristiques de l'automate cellulaire sont cités dans la littérature du domaine [Wolf1994], nous citons ici les plus importantes.

Caractéristiques fonctionnelles :

- Le voisinage : le nouvel état de chaque cellule est déterminé à partir de sa position spatiale dans l'univers de l'automate, en examinant les états des cellules voisines (transitions locales).
- Le parallélisme : mises à jour des cellules de l'univers de l'automate sont effectuées de manière simultanée et synchrone (en pratique, recours à un buer pour simuler le pseudo-parallélisme).
- Le déterminisme : pour une cellule, la donnée des états des cellule voisines détermine à elle seule le nouvel état.
- L'homogénéité : toutes les cellules de l'automate utilisent les même règles de transition pour déterminer leurs état suivant.
- La discrétisation : un CA se déroule dans le temps de manière discrète, génération après génération [GMPG2003, Wolf1983, Neum1966].

Caractéristiques techniques :

- Sa dimension : Le plus général 1 (CA élémentaire) ou 2 (CA type Life).
- Voisinage d'une cellule : L'ensemble des cellules qui auront une inuence sur la cellule étudiée.
- Son espace d'états : Ensemble des états que peut prendre une cellule.
- Sa fonction de transition : Ensemble des règles pour déterminer le nouvel état d'une C en fonction de son état précédent et de l'état précédent de ces voisines. Pour un automate à n états et avec un voisinage de k cellules, il peut y avoir n congurations de voisinage, Exemple : CA de Wolfram ($n = 2, k = 3$), Life Game : ($n = 2, k = 9$) [Wolf1983, Wolf1994, Maro2003].

2.8.2.7.4. Mode d'arrangement des cellules

Le type de grille sur laquelle l'automate cellulaire se déploie peut également varier : les cellules peuvent être carrées, triangulaires, hexagonales.

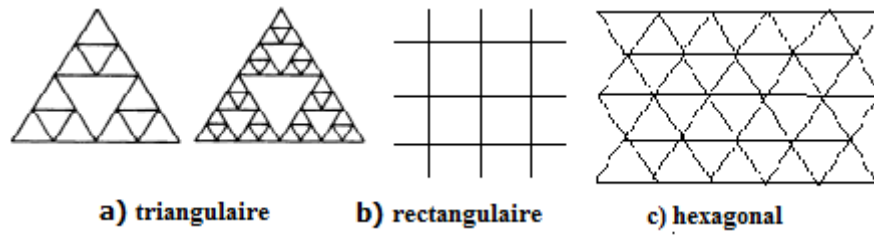
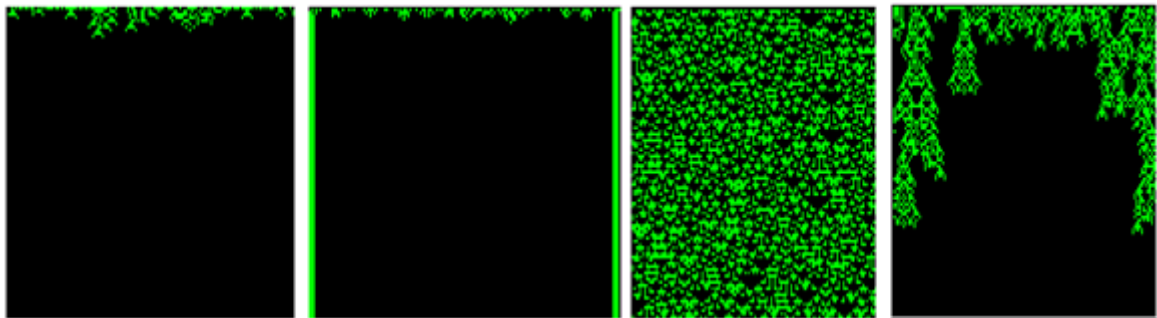


FIG. 2.20 – Mode d’arrangement des cellules.

2.8.2.7.5. Classes d’universalité dans les automates cellulaires

Parmi les recherches les plus intéressantes de Wolfram [Wolf1983] une idée forte est posée, Wolfram réalise une étude, montre que de nombreux automates cellulaires (peut-être tous) s’intègrent dans quatre classes principales :



<p>Classe I- L’évolution conduit à des configurations homogènes (systèmes statiques),</p>	<p>Classe II - L’évolution conduit à des structures simples ou périodiques (systèmes cycliques),</p>	<p>Classe III- L’évolution conduit à des configurations chaotiques (systèmes chaotiques),</p>	<p>Dans le cas de la classe IV, Les automates de cette classe évoluent vers des configurations globales complexes (système complexe).</p>
--	---	--	--

FIG. 2.21 – Les quatre classes de ACs, Adaptées de Wolfram [Wolf1983]

2.9 Conclusion

Dans ce chapitre, nous avons donné un aperçu général sur les concepts de base liés aux systèmes complexes et leurs propriétés objectives qui donnent un concept très intéressant : c’est l’émergence.

L’émergence est souvent invoquée les discussions des particularités globales des systèmes distribués. D’où le chapitre suivant sera consacré au concept d’émergence, ainsi

que la manière de l'exploiter dans le cadre informatique. En outre ce chapitre aborde les notions fondamentales de la vie artificielle où il présente des modèles de calcul inspirés de systèmes naturels.

Nous allons sélectionner parmi ces exemples de modèle d'intelligence, les algorithmes évolutionnaires et les algorithmes génétiques modélisés par les automates cellulaires comme une technique de travail dans le chapitre consacré à l'approche proposée.

Chapitre 3

LE PHÉNOMÈNE D'ÉMERGENCE

Dans ce chapitre, nous présentons le phénomène d'émergence. Nous donnons un historique, des définitions, des caractéristiques de l'émergence et son utilisation. Nous exposerons quelques exemples des systèmes émergents.

3.1 Introduction

L'évolution rapide et constante des matériels informatiques, des logiciels et des réseaux de communication nécessite des approches nouvelles pour la conception de systèmes informatiques. Les futurs systèmes qui en résultent doivent permettre d'augmenter la prise en compte de la complexité, être plus robustes et plus autonomes. Il semble évident que pour répondre à ces futurs besoins, les techniques informatiques actuelles sont inadéquates. Nous considérons l'émergence comme une nouvelle voie pour explorer la conception de systèmes complexes. Ce phénomène très intéressant reste difficile à appréhender. Dans ce chapitre, nous présentons le phénomène d'émergence à partir d'un historique et d'un ensemble de définitions tirées de la littérature. Nous étudierons les caractéristiques, de son utilisation en informatique. Nous terminons notre chapitre par une représentation de quelques exemples de systèmes émergents.

3.2 Phénomène d'émergence

Le processus d'émergence est vu comme une boîte noire. Nous ne pouvons discerner que les entrées de plus bas niveaux et les sorties de plus hauts niveaux. Nous ne savons pas comment les entrées sont transformées et reliées aux sorties voir la Fig. 3.1.

L'émergence ne peut être réduite ni à la somme ni à la différence des forces co-opérantes.

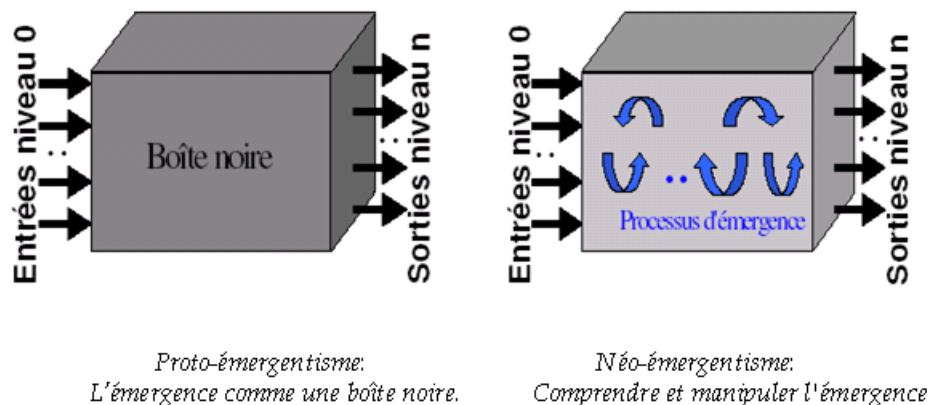


FIG. 3.1 – Travaux sur l'émergence, adapte de [Géor2003].

Cependant, la science s'est dotée de moyens permettant d'explorer cette boîte noire dans la deuxième moitié du *XXe* siècle. Nous pouvons appeler la recherche récente sur l'émergence le néo-émergentisme. Elle est liée à la théorie de la complexité actuelle et prend ses racines dans diverses approches comme la dynamique des systèmes en physique, en mathématiques et en informatique.

Le néo-émergentisme essaye de développer des outils, des méthodes et des constructions qui rendent le processus d'émergence moins opaque et par conséquent moins miraculeux [GMPG2003, Géor2003].

3.2.1 Définition de l'émergence

Dans cette sous section, nous donnons, à partir de la littérature [Vijv1997, Géor2003], les particularités de la notion d'émergence.

3.2.1.1 Auto-organisation

L'émergence fait référence à l'apparition durant le processus d'auto- organisation dans un système complexe de structures ou de schémas ("patterns") ou de propriétés nouvelles et cohérentes.

3.2.1.2 Irréductibilité

L'émergence en termes d'irréductibilité des propriétés associées à une théorie de haut niveau à des propriétés associées à des composants dans une théorie de plus bas niveau.

3.2.1.3 Nouveauté

D'une part, l'émergence présuppose qu'il y a apparition de nouveauté au niveau des propriétés, des structures, des formes ou des fonctions. D'autre part, elle implique qu'il est impossible de décrire, d'expliquer ou de prédire ces nouveaux phénomènes en termes physiques à partir des conditions de base définies aux niveaux inférieurs.

3.2.1.4 Interdépendance des niveaux

Des structures et/ou des fonctions apparaissent à un niveau macroscopique sans que la seule observation des propriétés des constituants permette de les prédire. Langton [Lang1990] définit l'émergence en termes de relation de feedback entre les niveaux dans un système dynamique. Les micro-dynamiques locales causent les macro dynamiques et les macro-dynamiques globales contraignent les locales.

L'émergence est le processus qui consiste à dériver des structures, des modèles et des propriétés nouvelles et cohérentes dans un système complexe. Les phénomènes émergents apparaissent suite aux interactions (non-linéaires et distribuées) entre les éléments du système.

3.3 Les propriétés d'un système émergent

Les propriétés d'interconnexion permettent d'identifier le phénomène comme émergent sont :

- L'observation d'un phénomène ostensible (*qui s'impose à l'observateur*) au niveau global ou macro,
- La nouveauté radicale du phénomène (*il n'est pas observé au niveau micro et n'est pas prévisible*),
- La cohérence et la corrélation du phénomène (il a une identité propre mais liée fortement aux parties qui le produisent),
- L'observation d'une dynamique particulière (*le phénomène n'est pas prédonné, il y a "auto-maintien" du phénomène*), voir [Géor2003].

Pour construire un système exhibant ces propriétés, il faut le doter d'un certain nombre de caractéristiques [Vijv1997, Ali1997].

3.3.1 La non-linéarité

La linéarité permet une traçabilité analytique des interactions locales jusqu'au phénomène global. Il n'y a donc pas d'apparition de nouveauté radicale et donc pas d'émergence possible. Il faut alors que les parties du système interagissent d'une façon non linéaire. En conséquence, une partie du système doit pouvoir être influencée par une autre partie même si elle ne lui est pas directement rattachée. Nous pouvons par exemple utiliser des boucles de feedback négatif ou positif qui sont non linéaires par nature.

3.3.2 Proche/Loin de l'équilibre

Au début, les théories cherchaient à comprendre comment des systèmes tendent vers un état final d'équilibre ou homéostasie. La théorie de la complexité s'intéresse plus aux conditions autour de l'équilibre qui entretiennent l'émergence. L'amplification des événements aléatoires est une des raisons clés pour lesquelles l'émergence fait apparaître des caractéristiques imprédictibles.

3.3.3 L'auto-organisation

Dans la théorie de la complexité nous nous référençons au comportement créatif et auto-généré qui produit les changements, ainsi qu'à la recherche d'adaptabilité d'un système complexe. Le système doit être capable de se modifier d'une façon ou d'une autre afin de changer de comportement en fonction de son environnement.

3.3.4 Les attracteurs

Dans la théorie classique des systèmes, le seul attracteur valide était celui qui amenait à un état final d'équilibre. Le fait d'utiliser un seul type d'attracteur simple avec comme finalité un état d'équilibre conduit à la prévisibilité du système. Cependant, nous constatons de nouveaux types d'attracteurs (le point fixe, le cycle limite et l'attracteur étrange) ce que nous retrouvons en théorie de la complexité. Ces attracteurs ne sont pas pré-donnés dans le sens de la "Gestalt". Ils ne dictent donc pas au système de l'état à atteindre mais lui donnent seulement les moyens de changer.

3.3.5 La dynamique

Cette métamorphose quantitative et qualitative du système si particulière est la conséquence de l'aspect de la non-linéarité et des nouveaux différents attracteurs. En effet, le tout doit être fait pour permettre au système d'exhiber un dynamisme suffisant pour en profiter et ainsi, tout en se maintenant dans des états proches/loins de l'équilibre, permettre l'auto-organisation qui conduit au phénomène émergent.

3.4 La résolution de problèmes par émergence

Les méthodes classiques de résolution de problèmes reposent sur la formalisation à priori du domaine du problème. Dans les domaines dynamiques et incertains cette formalisation devient difficile et demande des capacités adaptatives [Rama2006].

Dès 1995, l'émergence devait fournir une approche alternative pour la résolution de problèmes [Vijv1997]. La méthode classique de résolution de problèmes consiste à décomposer le problème en sous-problèmes et à donner les étapes de la résolution. Ensuite, le concepteur code le chemin que va parcourir le processus de résolution lors de son exécution. Dans l'approche émergente, le processus de résolution doit construire le chemin qui l'amènera à la solution.

3.4.1 Etudes des systèmes complexes par émergence

L'importance primordiale de la notion d'émergence apparaît lors de l'étude des phénomènes produits par les systèmes complexes [Géor2003]. Elle s'oppose au concept de réductionnisme mis en échec pour expliquer ces phénomènes. Van [Vijv1997] fait appel à la notion d'émergence lorsque la configuration et l'interaction des composants d'un système offre plus d'explications sur les dynamiques et les phénomènes observés, que les explications basées sur les parties seules. Les phénomènes observés sont alors expliqués par la notion d'émergence et sont caractérisés par le fait qu'ils ne sont ni prédictibles, ni déductibles, ni réductibles aux seules parties.

La distinction entre macro niveau (*niveau de l'émergent*) et micro niveau (*composants et interactions*). Nous admettons que l'on ne peut expliquer la dynamique du système uniquement par les comportements de ses parties de plus bas niveau.

Pour tenter d'expliquer les phénomènes émergents, on peut étudier, par exemple, leurs conditions d'apparition, leurs variations, les corrélations entre phénomènes émergents, leurs dynamiques intrinsèques, etc.

En résumé, les études sur l'émergence permettent de construire les outils nécessaires, les méthodes et les constructions (*systèmes artificiels, modèles*) pour mieux étudier le phénomène d'émergence et les systèmes dans lesquels il se produit.

3.4.2 Construction de systèmes complexes artificiels par émergence

L'émergence a une utilité très répandue dans le domaine d'informatique. Nous pouvons citer plusieurs champs d'application tels que :

- La Vie Artificielle,
- Les Automates Cellulaires,
- Les Algorithmes Génétiques,
- Les Réseaux de Neurones,
- Les Systèmes Multi-Agents,
- L'Optimisation par Colonies de Fourmis, ...

Cette association peut sembler prometteuse et peut aboutir rapidement à une utilisation répandue et concrète de l'émergence.

3.5 Les approches émergentes

Nous pouvons citer plusieurs approches émergentes comme : les automates cellulaires évolutionnaires et les systèmes adaptatifs à fonctionnalité émergente.

3.5.1 Les Automates Cellulaires Evolutionnaires (ACEv)

La résolution de problème par émergence consiste à découvrir les concepts des Automates Cellulaires Evolutionnaires.

3.5.1.1 ACEv : approche hybride

Afin d'utiliser les ACs pour la résolution de problèmes dans le domaine informatique [Neum1966], il nous faut une méthode de conception, qui permet de construire AC (règles locales de transition d'état), qui convient le plus à la tâche que l'on souhaite réaliser : la conception inversée [TBRW2005] ou le problème inverse.

Une manière de procéder serait de s'inspirer du principe d'évolution biologique des espèces. En utilisant les AGs pour concevoir AC [Mitc1996, Gang2001] qui convient. L'approche hybride ACEv du Santa Fe exploite la puissance du phénomène d'émergence dans les ACs d'une part [Gang2003, Wolf1984, KPre1984] et celle de l'évolution biologique (AGs) d'autre part [Thom2005, More2001]. En fait, ACEv est une approche hybride de conception, qui applique le principe des AGs pour identifier AC qui modélise au mieux le Système Complexe capable de résoudre de façon émergente un problème concret donné.

En résumé, les Automates Cellulaires Evolutionnaires (ACEv) sont donc un choix intéressant aux méthodes de conception classique 'Top-Down', pour la résolution de problèmes, particulièrement ceux du domaine informatique.

3.5.1.2 Principes de base des ACEv

Dans cette partie nous allons exposer comment les AGs peuvent être utilisés, pour faire évoluer une population des ACs [Mitc1996, More2001], afin qu'ils effectuent des tâches de haut niveau qui nécessitent une coordination globale.

ACEv a été mis en oeuvre, par les chercheurs de l'institut SantaFe aux USA. Il étudie principalement des ACs unidimensionnels à états binaires. Ce type des ACs est constitué d'une rangée de N machines cellulaires, chacune de ces cellules peut être dans l'un des deux états binaires 0 ou 1. Chaque cellule met à jour son état interne, en fonction des états courants des cellules d'un voisinage n de rayon r , selon une règle de transition ϕ (table de transition d'état).

Exemple :

Table de transition Φ :

Le Voisines n : **000 001 010 011 100 101 110 111**

État futur de la cellule : **0 0 0 1 0 1 1 1**

Mise à jour des états des cellules de l'automate :

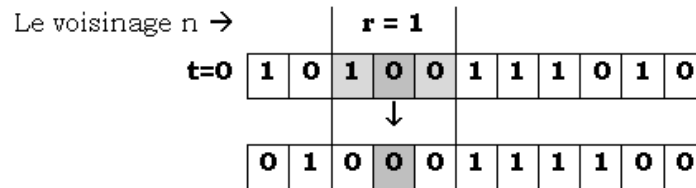


FIG. 3.2 – Illustration d'un AC unidimensionnel à états binaires, avec un Voisinage de $r = 1$ et $n = 11$,

Selon la Fig. 3.2, AC est initialisé par une configuration (configuration initiale : CI) d'états (valeurs), pour ses cellules. Cette configuration change à chaque transition discrète d'horloge t , selon la règle de transition Φ .

Pour résoudre une application donnée, ACEv s'appuie sur les AGs comme outils d'optimisation. AG démarre avec un ensemble de AC candidats, qu'on appelle population initiale. Chaque individu de cette population est représenté par un chromosome. Le chromosome est constitué de la concaténation des bits de sortie de la table de transition, qui représente la règle de changement d'état ϕ de AC. La taille du chromosome pour un voisinage r est de 2^{2r+1} [Mitc1996].

Exemple : pour $r = 2$, donc 5 cellules dans chaque voisinage, on a un chromosome de taille $2^5 = 32$ bits. Ce qui fait un espace de recherche de 2^{32} chromosomes possibles.

Pour un espace d'états aussi vaste la recherche exhaustive est exclue. La stratégie de recherche basée sur les AGs, utilise le principe d'évolution, pour retrouver le bon AC (*chromosome*). La population initiale de AG est évaluée, et chaque AC de cette population se voit attribuer un score (*fitness, pour affinité*), qui rend compte de son efficacité à résoudre la tâche demandée. Les individus les plus adaptés (*qui ont réalisé les meilleurs fitness*), sont sélectionnés pour construire la génération suivante, en combinant leur code génétique via des opérateurs génétiques de croisement et de mutation. Après plusieurs générations, nous obtenons une élite d'individus les plus adaptés. L'individu qui a réalisé le meilleur classement est considéré comme la solution à notre processus de recherche [Mitic1996, TBRW2005, Gras2005].

3.5.2 Les systèmes adaptatifs à fonctionnalité émergente : la théorie des SMAA

La théorie des SMAA proposée par [Géor2003, Ferb1995] est une théorie de l'auto-organisation basée sur la coopération dans laquelle :

- Le système est plongé dans un environnement,
- Le système réalise une fonction,
- Le système est composé d'agents en interaction,
- Chaque agent du système réalise une fonction partielle,
- L'organisation du système détermine le résultat du système.

3.5.3 La programmation émergente : le Calcul Emergent (CE)

L'idée proposée par Jean Pierre Georgé [Géor2002] est de réaliser un environnement permettant de concevoir des systèmes complexes adaptatifs afin de répondre aux problèmes de complexité croissante de la programmation, aux problèmes du dynamisme du système et de l'environnement, et à la contrainte de connaissance du problème et de la solution. Pour cela, il utilise comme base le concept de l'émergence et comme technique les SMA [Favi2004] auto-organiseurs utilisant la théorie SMAA [Vach2000], il propose de réaliser un langage complet de programmation dans lequel chaque instruction est un agent qui cherche à être coopératif avec les autres agents du système. Il dote ces agents de mécanismes d'auto-organisation, on obtient un système capable de s'adapter à la tâche requise par le programmeur, de s'auto-programmer.

3.6 Conclusion

Dans ce chapitre, nous avons présenté les principales réponses des interrogations scientifiques sur un concept très intéressant dans la résolution des problèmes complexes. Ce concept qui fait un échange de chemin de la résolution des problèmes de l'approche classique vers une nouvelle épistémologie c'est l'émergence, plutôt c'est l'approche émergente. Nous avons trouvé dans la littérature une sorte de réponse à l'incontournable question "c'est quoi le concept émergence?" donnée par exemple dans le travail [Géor2003]. Egalement, les propriétés d'un système émergent, la définition du phénomène émergent dans le cadre informatique, la résolution de problème par émergence ainsi que les approches émergentes. Nous avons cité ACEvs, les SMAs : la théorie des SMAA, et enfin les Calculs Emergents. Ces différentes informations recueillies nous permettent de bien comprendre le concept d'émergence.

Chapitre 4

SEGMENTATION D'IMAGE

Dans ce chapitre, nous présentons le processus de vision par ordinateur en général et la détection de contour en particulier. Nous essayons de modéliser la détection de contour qui représente un problème important dans le traitement d'images méthode non-conventionnelle.

4.1 Introduction

L'essor du traitement d'image (analyse d'image) n'a eu lieu que dans les années 1960 quand les ordinateurs commencèrent à pouvoir travailler sur des images, des méthodes et techniques de la vision par ordinateur. Cette discipline se trouve au carrefour des mathématiques appliquées et de l'informatique.

Le traitement est un terme précis pour couvrir les opérations. Ces dernières, à partir d'une image, génèrent une autre image. On parle aussi dans ce cas d'opérateur de traitement d'image. Pourtant, il y a bien une partie analogique et une partie numérique, mais on utilise peu le terme de traitement d'image analogique.

La compréhension du traitement d'image commence par la compréhension de ce qui est une image. Ensuite on trouve une étape d'acquisition de l'image. Dans beaucoup de cas c'est la qualité de cette étape qui va déterminer la faisabilité de la chaîne de traitement d'image à suivre. Ensuite on trouve en général ce que l'on nomme des prétraitements. Cela consiste souvent à corriger des petits défauts de l'image pour passer au traitement. Dans bien des cas ce sont des opérateurs de segmentation. Une bonne chaîne de traitement d'image se termine par un peu d'intelligence pour alors reconnaître les formes de l'image.

Dans le contexte de la vision artificielle, le traitement d'image se déplace après les étapes d'acquisition et de numérisation, assurant les transformations d'images et la partie de calcul permettant d'aller vers une interprétation des images, en faisant appel notamment à l'intelligence artificielle pour manipuler des connaissances, principalement sur les informations dont on dispose à propos de ce que représente les images traitées (connaissance du domaine).

4.2 Image à niveau de gris

Une image est représentée par une matrice de dimension "*nombre de lignes*" x "*nombre de colonnes*". Chaque pixel qui est élément de la matrice, représente l'intensité lumineuse comprise entre 0 et 255, soit 256 niveaux de gris. Le niveau de gris 0 correspond au *noir* tandis que 255 est représenté en *blanc*.

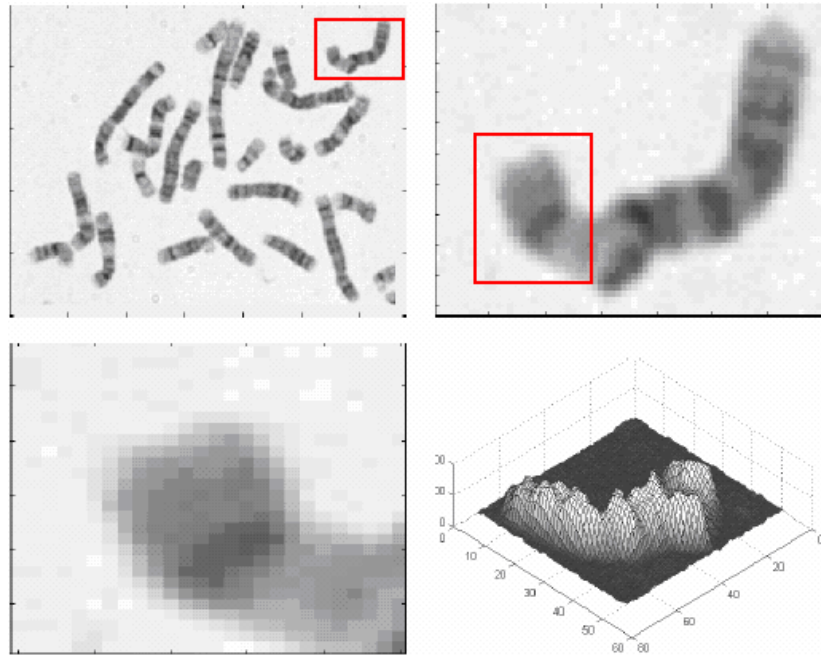


FIG. 4.1 – Représentation d’une image à niveau de gris du chromosome et des zooms d’une partie de cette image.

Nous constatons, en zoomant sur ces images l’effet de la discrétisation : des pixels carrés apparaissent, ce qui montre clairement la perte de l’information lors de l’acquisition.

Nous pouvons également visualiser les images comme une surface 3D, les axes x et y représentant la position spatiale des images tandis que des z représentent la luminance [Cath2003]. Donc, une image numérique est mathématiquement une application :

$$S = 1, \dots, M \times 1, \dots, N \longrightarrow 0, \dots, 255$$

$$s \longrightarrow Y(s)$$

12	15	33	3	75
0	11	12	0	1
123	176	0	0	1
111	121	0	1	1

TAB. 4.1 – Image en niveau de gris.

4.3 Image en couleur

Pour une image en couleur, nous mesurons la luminosité en un point s selon les trois canaux Rouge, Vert et Blue (**RVB**) [Jian2003]. Elle est alors représentée par un tableau de vecteurs de dimension 3 : autrement dit, une application de S dans $\{0, \dots, 255\}$.

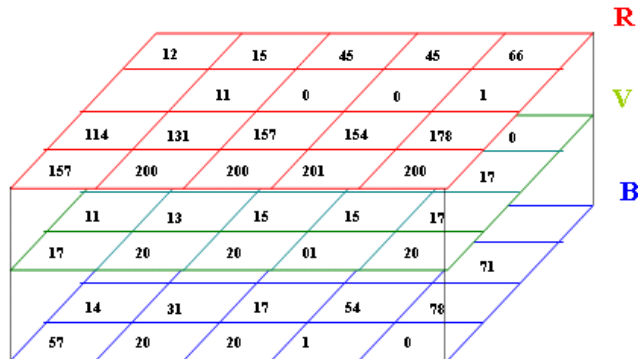


FIG. 4.2 – Image en niveau de couleur.

Nous citons ici les plus importantes qualités d'image numérique en couleur :

4.3.1 Images 24 bits (couleurs vraies)

Il s'agit d'une appellation trompeuse car le monde numérique (fini, limité) ne peut pas rendre compte intégralement de la réalité (infinie). Le codage de la couleur est réalisé sur trois octets, chaque octet représentant la valeur d'une composante couleur par un entier de 0 à 255. Ces trois valeurs codent généralement la couleur dans l'espace RVB. Le nombre de couleurs différentes pouvant être ainsi représenté est de 256 x 256 x 256 possibilités, soit près de 16 millions de couleurs.

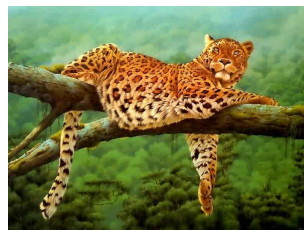


FIG. 4.3 – Image en couleur.

Comme la différence de nuance entre deux couleurs très proches mais différentes dans ce mode de représentation est quasiment imperceptible pour l'oeil humain. On considère commodément que ce système permet une restitution exacte des couleurs, c'est pourquoi on parle de "couleurs vraies".

Les images bitmap basées sur cette représentation peuvent rapidement occuper un espace de stockage considérable, chaque pixel nécessitant trois octets pour coder sa couleur.

R	V	B	Couleur
0	0	0	Noir
0	0	1	nuance de noir
255	0	0	Rouge
0	255	0	Vert
0	0	255	Bleu
128	128	128	Gris
255	255	255	Blanc

FIG. 4.4 – Les couleurs varient.

4.3.2 Images à palettes, images en 256 couleurs (8 bits)

Pour réduire la place occupée par l'information de couleur, on utilise une palette de couleurs " attachée " à l'image. On parle alors de couleurs indexées : la valeur associée à un pixel ne véhicule plus la couleur effective du pixel, mais renvoie à l'entrée correspondant à cette valeur dans une table (ou palette) de couleurs appelée 'look-up table' ou 'LUT' en anglais, dans laquelle on dispose de la représentation complète de la couleur considérée.

4.4 Les caractéristiques d'une image numérique

Plusieurs caractéristiques d'image numérique sont cités dans la littérature du domaine [Jack2001], nous citons ici les plus importantes :

4.4.1 La résolution

C'est le nombre de pixels par unité de longueur. La résolution est exprimée le plus souvent en ppp (point par pouce) ou en dpi (dots per inch).

- Un pouce = 2.54 cm.
- Une image de 100 ppp (ou 100 dpi) contient 10 000
- Points par pouce carré. (100 x 100 = 10 000).

La résolution définit la netteté d'une image et sa qualité d'affichage à l'écran. Plus la résolution est grande (c'est-à-dire plus il y a de pixels dans une longueur de 1 pouce), plus

voire image est précise dans les détails. C'est un critère important lorsque l'on utilise un scanner.

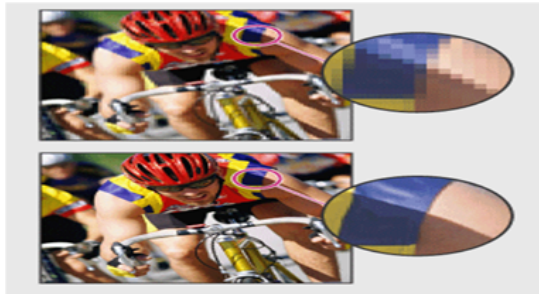


FIG. 4.5 – Différentes résolution.

4.4.2 Pixel écran-Pixel numérique

Il est important de bien dissocier le pixel d'un écran de télévision, qui a une taille physique, et le pixel numérique. Le pixel de l'écran de télévision est déterminé une bonne fois pour toutes lors de sa fabrication alors que le pixel digital a une taille physique uniquement lorsqu'on la lui donne. Un pixel digital est simplement une définition mathématique à l'intérieur de l'ordinateur que l'on caractérise en choisissant une taille.

4.4.3 La luminance

La moyenne ou luminance (Brillance) d'une image numérique en niveau de gris est définie comme la moyenne des pixels de l'image.

$$lum(F) = \frac{1}{m * n} \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} f(x,y) \quad (4.1)$$

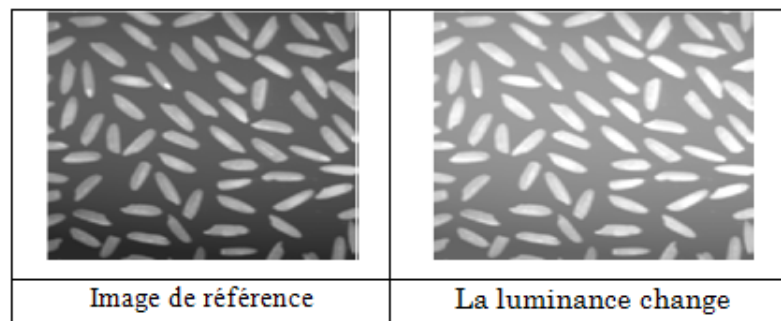


FIG. 4.6 – Echange de luminance.

4.4.4 Le contraste

Nous pouvons définir de plusieurs façons différentes le contraste d'une image numérique en niveau de gris, dans cette présentation nous nous limiterons à deux calculs de contraste, l'un plus adapté à des zones locales et le second à des images considérées dans leur globalité.

– Contraste local

Ce calcul est à effectuer sur un masque ou une partie de l'image.

$$Cont(V) = \frac{\max_{(x,y) \in v}(f(x,y)) - \min_{(x,y) \in v}(f(x,y))}{\max_{(x,y) \in v}(f(x,y)) + \min_{(x,y) \in v}(f(x,y))} \quad (4.2)$$

– Contraste global

Si le nombre de pixels d'une image est suffisant pour pouvoir utiliser de façon pertinente des moments statistiques d'ordre deux pour d'écrire la dispersion des valeurs des pixels autour de la moyenne de l'image (donc, autour de la luminance). Nous pouvons utiliser l'écart-type de l'image pour évaluer le contraste des images en niveaux de gris :

$$Cont(F) = \sqrt{\frac{1}{m * n} \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} (f(x,y) - lum(F))^2} \quad (4.3)$$

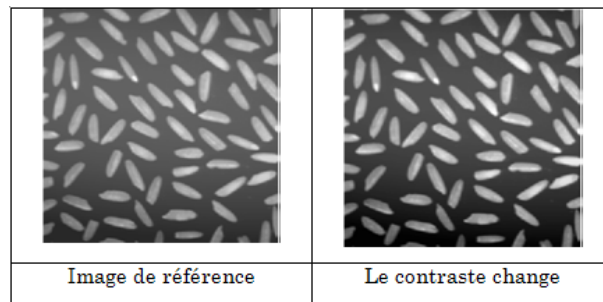


FIG. 4.7 – Echange de contraste.

4.4.5 Histogramme

Un histogramme est un graphique statistique permettant de représenter la distribution des intensités des pixels d'une image, c'est-à-dire le nombre de pixels pour chaque intensité lumineuse. Par convention un histogramme représente le niveau d'intensité en abscisse en allant du plus foncé (à gauche) au plus clair (à droite).

Ainsi, l'histogramme d'une image en 256 niveaux de gris sera représenté par un graphique possédant 256 valeurs en abscisses, et le nombre de pixels de l'image en ordonnées.

Dans la FIG. 4.8, l'histogramme fait apparaître que les tons gris clair sont beaucoup plus présents dans l'image que les tons foncés.

Pour les images en couleur plusieurs histogrammes sont nécessaires. Par exemple pour une image codée en RVB :

- Un histogramme représentant la distribution de la luminance,
- Trois histogrammes représentant respectivement la distribution des valeurs respectives des composantes rouges, bleues et vertes.
- Mémoire occupée par une image numérique.

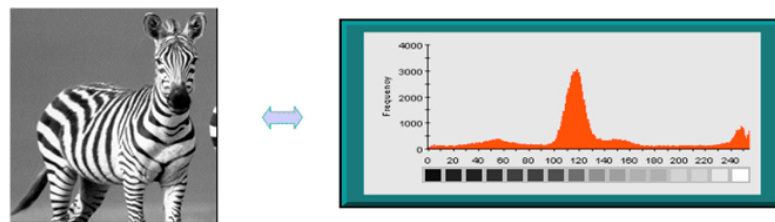


FIG. 4.8 – Histogramme d'une image.

4.5 Propriété d'une image numérique

L'image numérique est caractérisée par [Joli1997, Gill2001] :

- **Sa dimension** : la largeur par la longueur en centimètres ou en pixels.
- **Sa résolution** : ce paramètre définit la clarté ou la finesse atteinte par un moniteur ou une imprimante dans la production d'images et est exprimée en D.P.I. (Dots Per Inche, PPP ou Point Par Pouce en français). Nous utilisons aussi le mot résolution pour désigner le nombre total de pixels affichables horizontalement ou verticalement sur un moniteur ; Plus ce nombre est grand plus l'image dispose de détails fins.
- **La palette de couleurs utilisée** : c'est le nombre de couleurs disponibles pour reproduire l'image. Plus ce nombre est grand plus la reproduction est fidèle.
- **Son Poids informatique** : ce paramètre est appelé aussi " taille de l'image " ou " taille du fichier " qui est le nombre d'octets pour coder l'image. Plus il est grand et plus l'image prendra de l'espace sur le support de stockage plus les manipulations seront longues.

4.6 Le seuillage

Une image numérique, pour pouvoir être exploitée a généralement besoin d'être simplifiée, c'est le but du seuillage. Ce dernier consiste à transformer l'image codée sur 6, 8 ou 16bits, en une image binaire ou les pixels à 1 correspondent aux objets et les pixels à 0 au fond de l'image.

A la différence des différents traitements de l'image numérique, le seuillage est un passage obligatoire pour toute analyse morphologique ultérieure.

Le seuillage permet de sélectionner les parties de l'image qui intéressent l'opérateur, par exemple 2 types de grains (blancs et sombres) dans un mélange. On peut donc, par exemple, attribuer à tous les pixels de l'image numérique qui ont un niveau de gris compris entre deux valeurs i_1 et i_2 , choisies par l'opérateur, la valeur 1 ; à tous les autres pixels est attribuée la valeur 0.

Après seuillage, les parties de l'image sélectionnées seront traduites en noir et blanc. L'image, digitalisée par l'ordinateur (0et1), est appelée image binaire. Cette dernière, tout comme l'image numérique contient des informations superfétatoires, qu'il convient d'éviter, ou masquées qu'il faut révéler [Jack2001].

4.7 La segmentation d'images et détection de contours

Nous distinguons du traitement artificiel d'une scène visuelle [Cocq1995, Cath2003, Joli1997, Camp2000] trois étapes fondamentales :

- Acquisition : scène physique \rightarrow représentation numérique.
- Traitement (analyse) : Extraction de l'information pertinente par la segmentation \rightarrow description structurelle de l'image.
- Interprétation : description structurelle \rightarrow description sémantique.

Dans la section suivante nous expliquons bien l'étape de la segmentation d'une image numérique.

4.7.1 Présentation du problème

Une autre branche du traitement d'image s'occupe de l'analyse d'image ou analyse de scène. Dans ce domaine, l'entrée est toujours une image mais la sortie est une description

de l'image. La plupart des descriptions nécessitent une détection préalable des formes présentées dans l'image, étape appelée *segmentation* [Domi1994].

Autrement dit la segmentation d'images est un problème important dans le traitement numérique des images.

La segmentation est une opération qui a pour objectif la description de l'information contenue dans l'image en donnant une représentation plus condensée et facilement exploitable. Il est difficile de définir d'une manière absolue, une "bonne" segmentation. Cette dernière, souvent, n'est pas une fin en soi, sa qualité influe directement sur les résultats obtenus par les traitements situés en aval de l'étape de segmentation [Baill2003].

4.7.2 Définition

La segmentation est la séparation des divers éléments de l'image en régions connexes ayant mêmes propriétés (niveau de gris, intensité, couleur, texture) [Baill2003, Bure2001]. Etant un traitement de bas niveau [Domi1994], elle consiste à créer une partition de l'image A en sous-ensembles R_i , appelés régions tel que :

$$\forall R_i \neq \phi, \forall i, j; i \neq j, R_i \cap R_j = \phi, A = \cup_i R_i \quad (4.4)$$

L'image est un ensemble de points (pixels) ayant chacun un niveau encore de 3 composantes (*Intensité, Teinte et Saturation ...*). Nous ne disposons généralement que de ces informations de niveau de gris pour opérer la segmentation. Il existe deux approches principales de segmentation : approche frontière (*contour*) et approche région [Cocq1995].

Si l'homme sait naturellement séparer des objets dans une image c'est grâce à des connaissances de haut niveau (*compréhension des objets et de la scène*). Mettre au point des algorithmes de segmentation de haut niveau (*chaque région est un objet sémantique*) est encore un des thèmes de recherche les plus courants en traitement d'images.

Aujourd'hui, la segmentation est un pré-requis pour les traitements de plus haut niveau, tels que la reconnaissance de formes, le recalage ou la compression d'images... Elle touche plusieurs domaines d'application : imagerie satellitaire, aérienne, médicale.

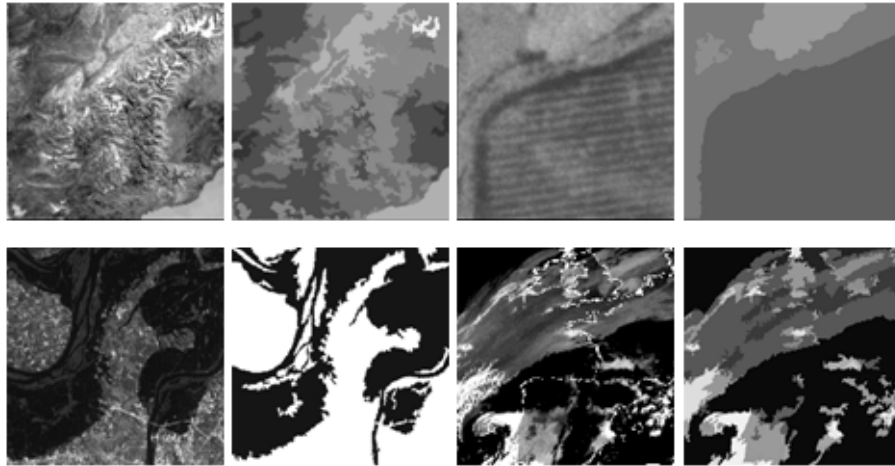


FIG. 4.9 – Segmentation en imagerie satellitaires et aériennes.

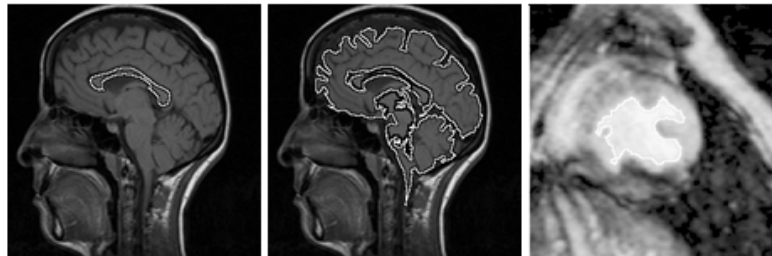


FIG. 4.10 – Segmentation en imagerie médicale.

4.7.3 Les types de segmentation d'image

La segmentation est une étape primordiale en traitement d'image. A ce jour, il existe de nombreuses méthodes de segmentation, que l'on peut regrouper en quatre classes principales :

1. Segmentation basée sur les régions (en anglais : region-based segmentation). On y trouve par exemple : la croissance de région (en anglais : region-growing), décomposition/fusion (en anglais : split and merge).
2. Segmentation basée sur les contours (en anglais : edge-based segmentation).
3. Segmentation basée sur classification ou le seuillage des pixels en fonction de leur intensité (en anglais : classification ou thresholding).
4. Segmentation basée sur la coopération entre les trois premières segmentations.

Le schéma suivant présente les différents types de segmentation des images :

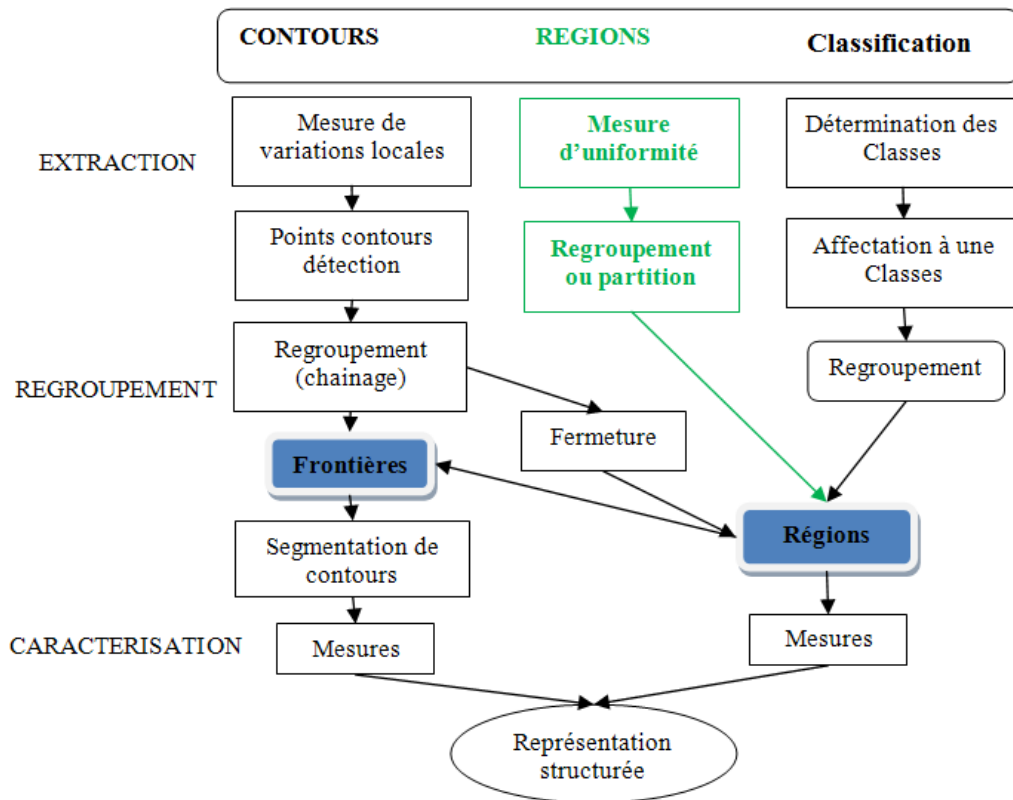


FIG. 4.11 – Type de segmentation d'images.

4.7.4 Les différentes approches de la segmentation

La segmentation d'images fait référence aux notions de différence et de similarité perçue par le système visuel humain. Ceci donne naissance à deux approches couramment qualifiées d'approche "*frontière*" et d'approche "*région*" [Jack2001, Cocq1995, Baill2003, TPav1982].

L'approche région s'attache à faire apparaître des régions homogènes selon un critère (*niveaux de gris ou texture*), alors que l'approche frontière tente de trouver des contours ou frontières de régions présentant une variation rapide du même critère.

Un algorithme de segmentation s'appuie donc sur :

1. La recherche de discontinuités afin de mettre en évidence les contours.
2. La recherche d'homogénéité locale pour définir les régions.
3. La coopération des deux approches.

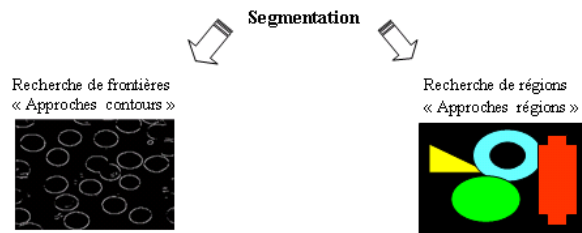


FIG. 4.12 – Les différentes approches de la segmentation.

4.7.4.1 Approche frontières (Contours)

Plusieurs définition d'un contour sont cités dans la littérature [Mong1995], la plus importante :

4.7.4.1.1 Définition d'un contour

Un contour est une zone de transition définie par son tracé qui permet de séparer deux plages non homogènes. La notion de frontière est associée à une variation d'intensité ou à une discontinuité entre les propriétés des deux ensembles connexes de points.

Attributs : largeur du contour, contraste, Contraintes de connexité et de géométrie.

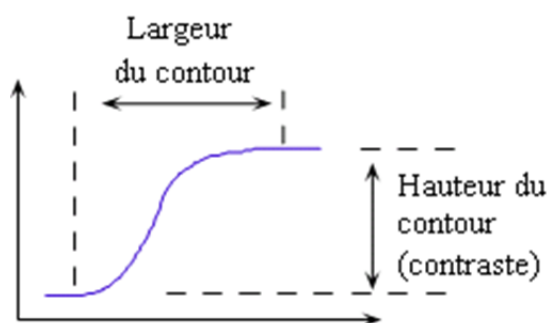


FIG. 4.13 – Exemple type contour.

Les approches frontières, sont parmi les méthodes les plus classiques en segmentation d'images. Ces méthodes supposent généralement un modèle a priori des discontinuités recherchées et opèrent de manière très localisée.

Les approches frontière peuvent aussi être classées en plusieurs catégories, à savoir : les méthodes dérivatives, surfaciques, morphologiques, markovienne et variationnelles. Les trois premières classes sont adaptées aux images qui présentent des régions uniformes au sens des niveaux de gris alors que les méthodes markovienne peuvent être utilisées pour la détection de frontières dans des images texturées ; elles fournissent des frontières de régions discontinues nécessitant ainsi une étape de post- traitement afin d'assurer la fermeture des contours. En revanche, les techniques variationnelles produisent des contours fermés. Elles prennent en compte l'information globale sur le contour, généralement issue d'un modèle a priori de contour.

4.7.4.1.2 But de la détection de contours

Le but de la détection de contours est de repérer les points d'une image numérique qui correspondent à un changement brutal de l'intensité lumineuse. Ces changements de propriétés de l'image traduisent en général des événements importants ou des changements dans les propriétés du monde. Ils incluent des discontinuités dans la profondeur, dans l'orientation d'une surface, dans les propriétés d'un matériau et dans l'éclairage d'une scène.

La détection de contour est un champ de la recherche qui appartient au traitement d'image et à la vision par ordinateur, particulièrement dans le domaine de l'extraction de caractéristiques. Dans une image en niveaux de gris, un contour est caractérisé par un changement brutal de la valeur. Le but de l'opération est de transformer cette image en une autre dans laquelle les contours apparaissent par convention en blanc sur fond noir [Neum1966].

4.7.4.1.3 Définition de l'approche frontières

La détection des contours peut être perçue différemment selon le but recherché. Ainsi, selon Pavlidis [Pav1992], elle consiste en une détection des discontinuités présentes dans une image. La détection des contours est alors considérée comme une des premières étapes de la vision par ordinateur et aucune information a priori sur les objets présents dans

l'image n'est utilisée. Toutefois, dans certaines applications, il est utile d'exploiter des informations sur la forme et l'emplacement approximatif de l'objet pour en détecter le contour.

Dans cette approche, on s'intéresse aux variations des niveaux de gris des pixels. Avec ce type de méthodes, on détermine un ensemble de contours candidats (des tracés susceptibles d'être des contours). Ensuite, on effectue un filtrage de ces contours pour ne garder que ceux qui semblent être les plus fiables.

La formation des régions va se faire en déterminant leurs frontières. Un point frontière ou contour est défini comme un point de l'image situé à la limite de deux zones, dont les propriétés sont fortement contrastées. De manière générale, les points contours seront les lieux de discontinuité de la luminance ou d'une autre fonction caractérisant l'image.

Inconvénients

L'approche contour n'aboutit pas directement à une segmentation, car les contours détectés ne sont pas toujours connexes. Il existe cependant des techniques permettant d'obtenir des contours fermés.

De nombreuses méthodes permettant l'extraction de contour existent dans la littérature. Elles reposent presque toutes sur des techniques de dérivation suivies de la recherche de maxima locaux ou de passage par zéro. Nous présentons deux principales approches :

4.7.4.1.3.1 Approche par gradient

Ce type de détecteur se base sur la première dérivée de l'image I en chacun de ces points dans les deux directions horizontale et verticale. Un point de contours aura une amplitude $A(i, j)$ et une direction $Dir(i, j)$.

$$\frac{\delta I(i, j)}{\delta x} = I_x(i, j); \frac{\delta I(i, j)}{\delta y} = I_y(i, j); A(i, j) = \sqrt{I_x(i, j)^2 + I_y(i, j)^2};$$

$$Dir(i, j) = \arctan(I_y(i, j)/I_x(i, j)) \quad (4.5)$$

La détermination des points contours est ramenée dans un premier temps à la recherche de filtre linéaire permettant d'estimer le gradient en chaque point.

De nombreux opérateurs sont ainsi apparus dans la littérature parmi lesquels nous pouvons citer les masques de *Sobel*, *Prewit*, *Robert*etc



FIG. 4.14 – Filtrage de Sobel.

La valeur du gradient est ainsi disponible en tout point de l'image permettant d'effectuer une recherche des maxima locaux. Ceux-ci correspondent aux passages par zéro de la dérivée seconde dans la direction du gradient ou encore aux points contours recherchés.

4.7.4.1.3.2 Approche par le laplacien

Ce type de détection de contour se base sur la dérivée seconde de l'image. Il est défini par :

$$\Delta^2 = \frac{\delta^2}{\delta_x^2} + \frac{\delta^2}{\delta_y^2} \quad (4.6)$$

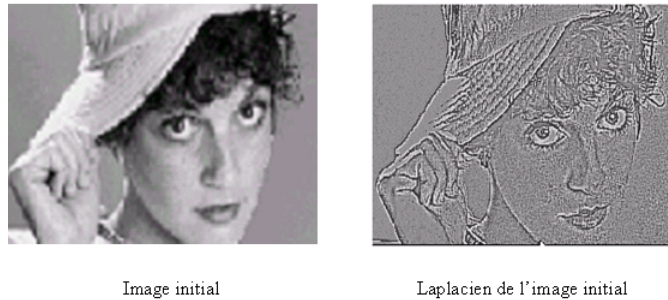


FIG. 4.15 – Filtrage Laplacien.

Contrairement au gradient, le laplacien permet d'obtenir des contours fermés et d'un pixel d'épaisseur, par contre il a l'inconvénient d'être plus sensible au bruit que le gradient pour cela le lissage et le laplacien sont regroupés en un seul filtre. Le laplacien est déterminé en chaque point de l'image par filtrage linéaire. Les points contours sont alors assimilés au passage par zéro du laplacien. A partir des résultats de Canny [Cany1986], les contours prennent une intensité d'image I a son entrée, et renvoie une image binaire de la même taille que I , avec 1 où la fonction trouve les contours et 0 ailleurs. La détection de contours supporte différentes méthodes :

- La méthode de Sobel trouve des contours en utilisant l'approximation de Sobel au dérivé. Elle renvoie des contours à points où le gradient de I est maximum.
- La méthode de Prewitt trouve des contours en utilisant l'approximation de Prewitt au dérivé. Elle renvoie des contours à points où le gradient de I est maximum.
- La méthode de Roberts trouve des contours en utilisant l'approximation de Roberts au dérivé. Elle renvoie des contours à points où le gradient de I est maximum.
- Le Laplacien de la méthode gaussienne trouve des contours en recherchant des passages à zéro après filtrage de I avec le Laplacien de filtre gaussien.
- La méthode de Canny trouve les contours en recherchant des maximum locaux du gradient de I . le gradient est calculé en utilisant le dérivé d'un filtre gaussien.

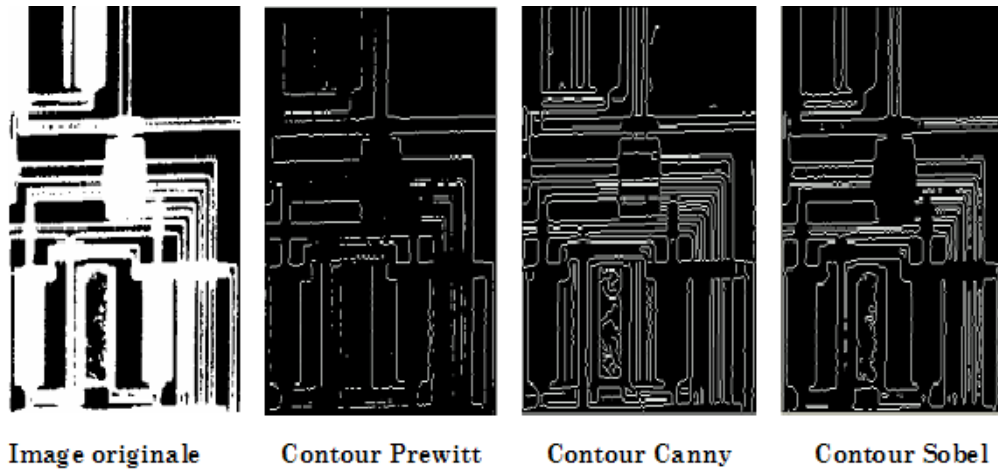


FIG. 4.16 – Les différentes méthodes de détection de contours.

A. Dérivations

Un contour est une frontière entre deux milieux différents (deux couleurs, deux niveaux de gris) [Jack2001]. Pour déterminer les contours, on utilise des méthodes de dérivation car ces contours correspondent en général aux extrema locaux du gradient ou aux zéros du Laplacien, de la fonction des niveaux de gris. Donc il existe deux approches : celle qui utilise la dérivée première de l'image (équation : 4.5) et celle qui utilise la dérivée seconde de l'image (équation : 4.6). Même si ces deux approches sont différentes, elles donnent pratiquement les mêmes résultats.

$$I_x(x, y) = \frac{\partial I(x, y)}{\partial x}, I_y(x, y) = \frac{\partial I(x, y)}{\partial y}, \Delta I(x, y) = \frac{\partial^2 I(x, y)}{\partial x^2} + \frac{\partial^2 I(x, y)}{\partial y^2} \quad (4.7)$$

B. Filtres dérivatifs de détection de contours

Ce sont des filtres, qui accentuent les contours. C'est-à-dire, qu'ils détectent les zones de forte courbure d'un signal, et ceci en calculant des dérivées du premier ou du second ordre.

- **Filtre dérivées premières** : le filtre le plus simple consiste à calculer les différences entre pixels voisins sur les horizontales puis sur les verticales. Chaque extremum correspond à un point d'un contour.
- **Filtre de prewitt** : le filtre de Prewitt introduit un flou. Chacune des deux matrices étant le produit du filtre dérivation dans la direction considérée par un filtre de flou rectangulaire selon l'autre direction.

- **Filtre de sobel** : la technique précédente est améliorée en remplaçant le filtre rectangulaire par un filtre triangulaire.
- **Filtre Canny** : c'est un filtre de Sobel précédent par un lissage Gausien et suivi par un seuillage. Ce filtre est conçu pour être optimal, au sens de trois critères.
- **Filtre de dérivées secondes** : celles ci se calculent simplement en différences finies et c'est maintenant un changement de signe qui correspond à un point du contour. On les utilise généralement à travers leur somme qui est le Laplacien.

4.7.4.1.4 Domaines d'application

Le traitement d'images possède l'aspect multidisciplinaire. On trouve ses applications dans des domaines très variés tels que les télécommunications (TV, vidéo, publicité ...), la médecine (radiographie, ultrasons ...), biologie, astronomie, géologie, l'industrie (robotique, sécurité), la météorologie, l'architecture, l'imprimerie, l'armement (application militaire). De nouvelles applications pratiques sont possibles aujourd'hui et touchent tous les domaines d'activités, tels que : métiers du spectacle, de la radio et de la création artistique.

4.7.4.2 Approche région

Nous citons ici la définition d'une région et ses importants attributs [Mong1995].

Définition d'une région

Une région est un ensemble connexe de points image (pixels) ayant des propriétés communes (*intensité, texture, ...*) qui les différencient des pixels des régions voisines.

Plusieurs attributs : aire, périmètre, centre de gravité, axes principaux, ...

Contrairement aux approches frontières qui recherchent les dissimilarités, les approches région recherchent plutôt la similarité. Ces approches fournissent une carte de régions fermées. Cependant la localisation des frontières reste généralement peu précise.

La segmentation d'images en région peut être comme la création d'une partition de

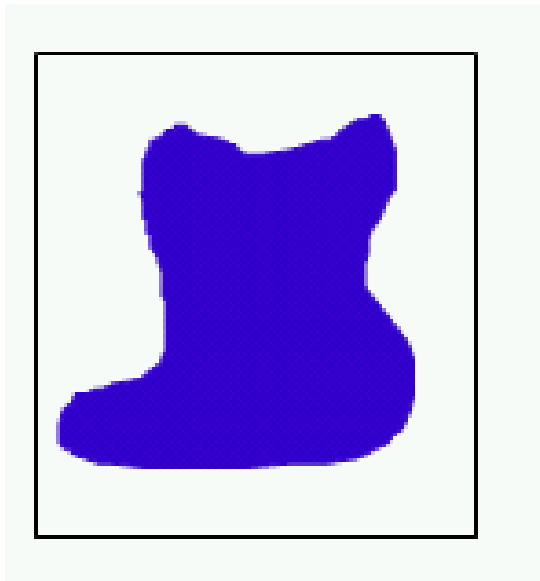


FIG. 4.17 – Exemple type région

l'image A en sous-ensembles R_i , appelés régions tel qu'aucune région ne soit vide et que l'intersection entre deux régions soit vide et l'ensemble des régions recouvre toute l'image. Une région est un ensemble de pixels connexes ayant des propriétés communes qui les différencient des pixels des régions voisines. Nous trouvons essentiellement trois types de méthodes [Cocq1995] :

Les méthodes appartenant à cette famille manipulent directement des régions. Soit elles partent d'une première partition de l'image, qui est ensuite modifiée en divisant ou regroupant des régions, et on parle alors de méthodes de type décomposition/fusion (ou split and merge en Anglais); soit elles partent de quelques régions, qui sont amenées à croître par incorporation de pixels jusqu'à ce que toute l'image soit couverte, et on parle alors de méthodes par croissance de régions. Des méthodes fondées sur la modélisation statistique conjointe de la régularité des régions et des niveaux de gris de chaque région existent.

4.7.4.2.1 Segmentation en régions, par décomposition/fusion

Ces méthodes ont pour principe, d'effectuer un découpage arbitraire de l'image, de réaliser des tests de cohésion des zones, puis de modifier ces zones.

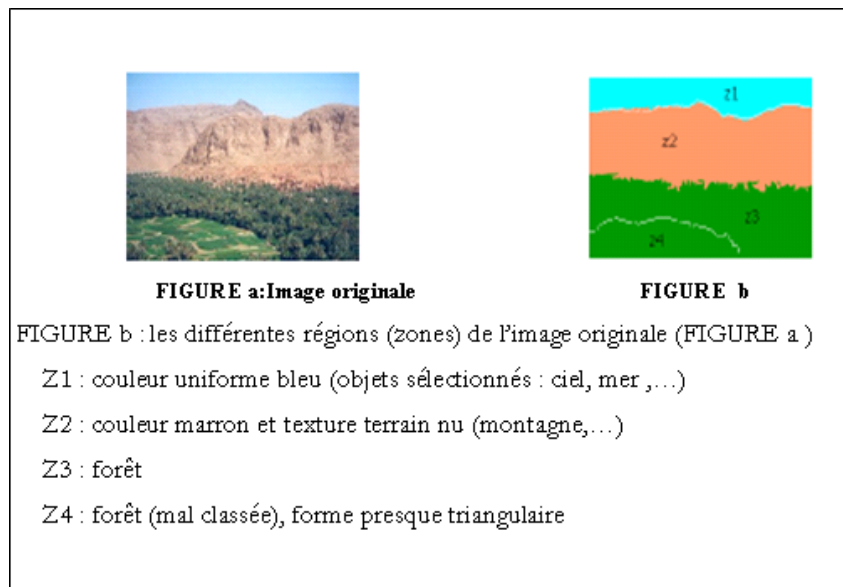


FIG. 4.18 – Les différentes régions de l'image.

- Décomposition

Cette méthode suppose au départ que tous les pixels appartiennent à la même région. Si la région n'est pas homogène (critère d'homogénéité), elle est divisée en plusieurs régions, sinon, le processus se termine. Chaque région nouvellement créée est potentiellement redivisée en plusieurs régions si elle n'est pas homogène. Le critère d'homogénéité peut être le même que précédemment (variance des niveaux de gris inférieure à un seuil). Lors de la division, on peut rajouter un critère d'arrêt sur la taille des régions (voir la FIG.4.19).

- Fusion

Cette méthode consiste à reprendre le résultat de la division et à essayer de fusionner des régions qui ont été malencontreusement séparées. Pour ce faire, on utilise un outil très précieux : le graphe d'adjacence des régions (voir la FIG.4.20).

4.7.4.2.2 Segmentation par croissance de régions

Elle consiste à regrouper itérativement, des ensembles de points connexes, en régions de plus en plus importantes, en utilisant des conditions dépendant des propriétés d'homogénéité [Hora1995]. On définit ici des critères de regroupement des pixels, permettant d'obtenir des régions homogènes.

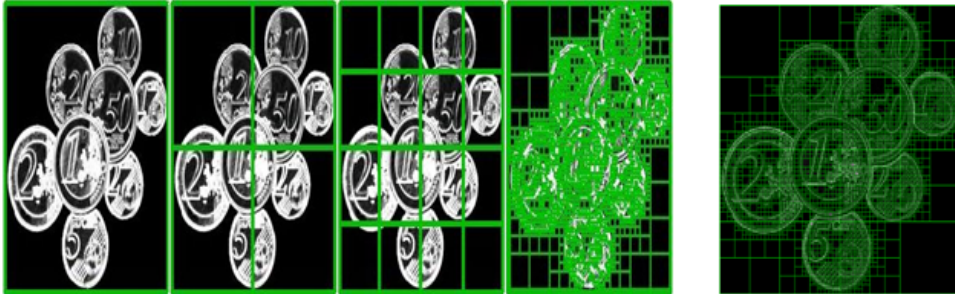


FIG. 4.19 – Décomposition de l'image.

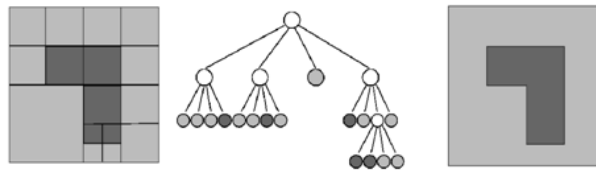


FIG. 4.20 – Division par Quad-Tree et regroupement feuilles similaires.



FIG. 4.21 – Résultat de segmentation par décomposition/fusion.

- Segmentation par agrégation de pixels

L'initialisation de cette méthode consiste à considérer chaque pixel comme une région. On va alors essayer de les regrouper entre elles avec un double critère de similarité des niveaux de gris et d'adjacence. Le critère de similarité peut par exemple être : la variance des niveaux de gris de la région R est inférieure à un seuil.

Le principe de l'agrégation de pixel est le suivant : on choisit un germe et on fait croître ce germe tant que des pixels de son voisinage vérifient le test d'homogénéité. Lorsqu'il n'y a plus de pixel candidat dans le voisinage, on choisit un nouveau germe et on itère le processus (voir la figure ci-dessous).

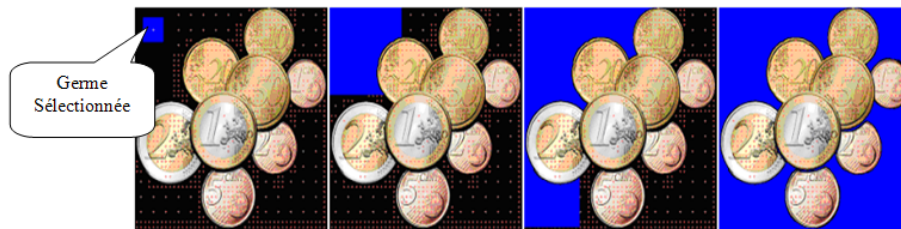


FIG. 4.22 – Segmentation par agrégation des pixels.



FIG. 4.23 – Résultat de segmentation par agrégation de pixels.

4.7.4.2.3 Détection des régions par fermeture de contours

On cherche d'abord les frontières entre les régions, plutôt que les zones homogènes en elles mêmes, en utilisant un algorithme de détection de contours. Cependant les contours obtenus sont rarement fermés. On doit donc, appliquer un algorithme de fermeture de contours.

4.7.4.2.4 Méthode de classification ou seuillage

Il s'agit de regrouper les différents éléments (*pixels*) en thèmes correspondants à la vérité du terrain. Le résultat est bien entendu une image segmentée. Dans ce domaine on procède, par l'attribution des pixels de l'image à des classes en exploitant leur similarité par opposition à l'approche frontière qui recherche les dissimilarités.

Les classes sont construites à partir des attributs (*en générale le niveau de gris*) des pixels. Chaque pixel est affecté à une classe unique. On procède, par l'attribution des pixels de l'image à des classes connues à priori (*c'est la classification supervisée*) ou à des classes inconnues (*classification non supervisée*).

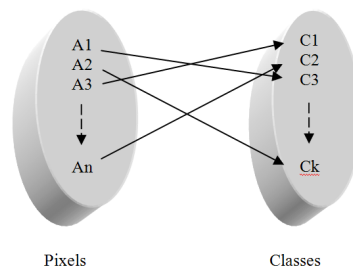


FIG. 4.24 – Classification de pixels.

La segmentation par les techniques de classification se heurte au problème du choix du nombre de classes (*approches non supervisées*) et au problème du choix des attributs, la connaissance à priori de l'appartenance de chaque échantillon de l'ensemble d'apprentissage à une classe donnée, ce qui revient à supposer une connaissance à priori sur l'image à segmenter (*approches supervisées*).

4.7.4.2.5 Méthodes structurelles

Les stratégies employées pour ces méthodes sont ascendantes suivant qu'elles exploitent des mécanismes de croissances ou de division de régions. Les méthodes du type division et fusion comportent, comme leur nom l'indique, deux étapes. Dans la première, l'image est divisée récursivement jusqu'à ce que toutes les régions vérifient le prédicat P (*critère*). Dans la deuxième étape, des régions adjacentes sont regroupées, tant que les régions résultantes vérifient P .

On optient ainsi une segmentation de l'image mais sans avoir de frontières précises entre les objets. En fait, un grand nombre de variantes sont possibles on fonction de la

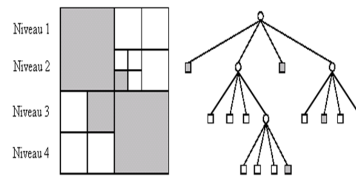


FIG. 4.25 – Méthode structurale.

taille minimale des zones que l'on utilise et du critère d'homogénéité.

4.7.4.2.6 Méthode stochastique

Dans ce type d'approches, les modèles sont souvent discrets et de nombreux travaux reposent sur la théorie des champs de Markov et l'estimation bayésienne. Les méthodes de segmentation par approche région basées sur les champs Markoviens s'appuient sur d'importants développements théoriques dans le domaine des champs Markoviens, de l'estimation de paramètres et de l'optimisation.

Notons que ce type d'approche est plus proche d'un algorithme de classification que de segmentation puisqu'à chaque pixel site considéré, on cherche à attacher une étiquette ce qui peut conduire à des pixels non connexes pouvant avoir la même étiquette.

4.7.4.3 Approche Coopérative

Concernant la segmentation, il est acquis depuis une dizaine d'années déjà qu'il faut faire coopérer les approches régions et les approches contours [Wolf1994] pour définir un bon critère de segmentation. Indépendamment des problèmes inverses à régulariser, de nombreux travaux ont été développés ces dernières années sur des modèles de contours. Ces modèles de contours sont développés pour la segmentation d'objet dans une image et n'étaient pas, jusqu'à récemment encore, envisagés comme des modèles de contours pour la segmentation par approche contour/région. Les travaux actuels tentent de combiner et d'unifier ces différents modèles.

4.7.5 Critère de choix de la technique de segmentation

Il n'y a pas de méthode unique de segmentation d'une image. Le choix d'une technique [Baill2003] est lié :

A la nature de l'image

- Eclairage non homogène, reflets.
- Présence de bruit, de zones texturées.
- Contours flous, en partie occultée.

Aux opérations situées en aval de la segmentation

- Localisation, mesure, calcul 3D.
- Reconnaissance des formes, interprétation.
- Diagnostic, contrôle de qualité.

Aux primitives à extraire

- Contours, segmentation de droite, angles, ...
- Région, formes.
- Textures.

Aux contraintes d'exploitation

- Complexité algorithmique, fonctionnement en temps réel.
- Taille de la mémoire disponible.

4.8 Conclusion

Dans ce chapitre nous avons présenté en particulier l'image de niveau de gris et celle en couleur. Ensuite nous avons développé le concept de segmentation ainsi que les techniques. Ce processus qui rend une image en entrée produit une autre image à la sortie, qui est le résultat d'une telle opération de transformation. Cette opération de transformation vise à extraire des informations spécifiques et satisfaire la vision humaine. Nous avons donné dans un aperçu sur le principe de fonctionnement des différentes méthodes de segmentation des images puis on s'intéresse à la méthode basée sur la détection de contour.

Le chapitre suivant sera consacré à la présentation d'une nouvelle approche basée sur les ACEv.

Chapitre 5

APPROCHE PROPOSÉE : GACE_v-DC

Dans ce chapitre, nous présentons notre nouvelle approche de résolution du problème de détection de contours par émergence. Cette nouvelle approche est basée sur les ACE_v pour la détection de contours dans des images en couleurs (GACE_v-DC).

5.1 Introduction

Le AC peut être interprété comme un ensemble de règles qui par un AC évolutionnaire (ACEv), nous pouvons trouver une ou plusieurs règles spécialisées pour un problème défini. Notre idée d'utiliser un paquet de règles dans la détection de contour et du filtrage sont dans le mérite de Rosin [Rosi2006]. En effet, elle est utilisée dans la restauration des images noir-blanc. D'ailleurs, Rosin [Rosi2006] a étudié ces meilleures règles en détails et a montré l'intérêt de chacun. Le résultat de son étude a prouvé qu'une règle simple peut enlever les Pixels isolées dans une image bruit noir-blanc.

Dans des travaux précédents [SLKa2008A, SLKa2009, SLKa2009A] nous avons déterminé un ensemble de règles (*paquet*) pour détecter des contours et des régions dans des images noires et blanches, en utilisant un AG. Nous avons trouvé les règles comme 0000000001, 1111111110 qui effectuent toujours la segmentation fine de contour [SLBM2007].

Dans cette thèse, nous utilisons un AG pour trouver un ensemble de règles pour extraire des contours efficaces dans une image donnée à niveau de gris ou en couleur [SLKa2008, SLKa2011]. En effet, un ACEv est appliqué afin de déterminer les meilleures règles locales du AC, en utilisant un AG sur une population de candidats de AC [Vlad2005].

Après cette introduction, la section deux présente la segmentation comme fonctionnalité émergente. L'approche de **GACEv-DC** est rapporté dans la section trois. Des conclusions sont tirées dans la dernière section.

5.2 La segmentation comme fonctionnalité émergente

Les ACs sont un modèle mathématique puissant, qui permet de simuler le comportement d'un système complexe [Wolf1984, Wolf1994]. Ils ont l'avantage d'être facile à comprendre et à implémenter. On doit mettre à profit certaines caractéristiques primordiales, dont la plus importante est la notion d'émergence [Vijv1997, Géor2003].

Cette propriété est la résultante directe de la complexité des interactions à l'intérieur du système. Cette interaction entre deux ou plusieurs comportements permettra l'émergence d'une nouvelle fonctionnalité par l'apparition d'un phénomène global qui n'a pas été explicitement programmé dans le comportement des ACs. Il est intéressant aussi de

savoir comment a priori déduire la règle souhaitable pour effectuer un tel calcul (émerger une capacité fonctionnelle).

Dans notre contexte, la fonctionnalité émergente correspond à la tâche de segmentation d'image, dans laquelle les règles locales de AC émergent (versus trouvés à la main), de l'évolution d'une population de AC candidats, par le biais d'une stratégie d'optimisation basée sur les AG [Mitc1996]. Un ensemble de règles locales (de AC) simples et efficaces pour la détection de contours sont découvertes par AG. Ces règles sont exécutées par un AC initialisé par les intensités des pixels de l'image à segmenter.

5.3 L'approche proposée : la Généralisation des Automates Cellulaires Evolutionnaires pour la Détection de Contours (GACEV-DC)

L'approche proposée [SLKa2008] tire profit des aptitudes calculatoires du AC, pour transformer les configurations initiales définies par un treillis numérique d'images comme des données d'entrée discrètes afin de trouver ses contours.

L'unité de AC est représentée par un rectangle de 9 cellules. En particulier le cas de l'image bi-couleur, le problème doit trouver les meilleurs patterns (configurations) ou des règles de AC pour la détection de contour parmi 2^{51} modèles ou règles possibles. Autrement, pour l'image en couleur avec plus de deux couleurs ; 16, 256, 16bits, 24bits, 32bits, alors le domaine de recherche est $2^{K^{8/5}}$ patterns ou règles possibles, où K est le nombre de couleur d'image et 5 est le nombre de transformations géométriques.

Dans cet énorme domaine large de recherche que nous analysons, nous recherchons le paquet optimum des règles qui donne la meilleure segmentation de contour. Ainsi manuellement, la méthode exhaustive de recherche consommera beaucoup de temps, d'effort et de ressources informatiques pour atteindre des résultats optimaux. Ainsi, nous devons travailler avec le problème inverse où nous avons trouvé l'ensemble de règles optimal qui donne la meilleure segmentation de contour en utilisant les possibilités d'évolution de recherche des algorithmes génétiques (voir la FIG.5.5).

La première observation importante est que ce domaine large des patterns de voisinage peut être réduit à un maximum de 8 couleurs classées, parce que chaque AC a un voisinage

de 8 cellules. Et cela engendre une absence de l'information (contour) quand, le maximum de 8 couleurs sont distinctes autour du pixel étudié qui est le centre de l'automate. Il est à noter que le domaine de recherche est de taille $2^{8^8/5}$.

Deuxièmement, pour s'assurer que chaque AC fonctionne indépendamment quel que soit la couleur indexe (classe), nous avons défini ce que nous appelons les patterns génériques. (voir au-dessous le principe de codage de chromosome).

$$\begin{array}{cccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & = & 9C_1 \\ 1 & 1 & 1 & 2 & 2 & 3 & 3 & 4 & 4 & = & 3C_12C_22C_32C_4 \end{array}$$

FIG. 5.1 – Générique pattern, Ci : classe i, i : numéro de classe.

Le AC peut être interprété comme un ensemble de règles où par un ACEv, nous pouvons trouver une ou plusieurs règles spécialisées pour un problème défini.

L'approche proposée appelée GACEv-DC [SLKa2008], prend une population des paquets ou des règles candidats et les évaluent pour la génération future, elle se base sur les mesures du rapport (fitness) avec quelques images segmentées bien connues.

La construction du paquet doit prendre la considération de quelques contraintes qui peuvent ralentir la convergence et peuvent falsifier le processus global.

1. *Redondance* : un paquet ne doit pas avoir des règles plus d'une fois.
2. *Contradiction* : un paquet ne doit pas avoir plus d'une règle par patch (deux patches avec différentes transitions).
3. *Similitude* : comme ces patterns colorés elles ont similaire $111223344 = 888115533$, parce qu'elles sont l'occurrence du pattern générique $C_1C_1C_1C_2C_2C_3C_3C_4C_4$.
4. *Inutile* : un paquet avec quelques règles qui n'ont pas le patch à opérer.

La structure de cette nouvelle méthode est composée de deux étapes :

5.3.1 Phase d'apprentissage

Elle a comme objectif d'extraire des modèles de règles. Ceci peut être fait par un ensemble des templates conçus manuellement ; ils ont une particularité de pouvoir figuré dans n'importe quelle image. Nous avons trois templates génériques (12 x 12 Pixel) :

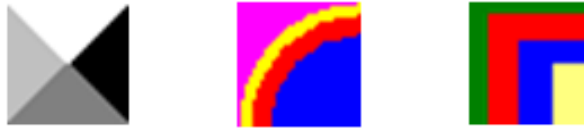


FIG. 5.2 – Patterns de base pour la phase d'apprentissage.

5.3.2 Phase de construction de paquet optimal

Elle a comme objectif de construire des règles plus stables que ceux extraits dans la phase d'apprentissage. Pour assurer la meilleure convergence nous devons renforcer le test par quelques traitements (toujours en utilisant des règles consacrées de AC) comme, réduction de couleur, filtrant avant et après le processus efficace de segmentation.

Ensuite, le paquet avec la meilleure fitness sera le paquet de base consacré à la segmentation de contour d'image.

Le code suivant décrit le processus de AG pour déterminer le meilleur paquet de règles de transition capable de réaliser la détection de contour des images en couleur.

Algorithme 1. Etapes de GACEv-DC pour la segmentation des images en couleur.

1. **Les données d'entrée** : image originale.

Image originale : images en niveau de gris ou en couleur. Classe d'image : images de forte homogénéité (ni texture, ni bruit). Représentation d'information :

- Image en niveau de gris : classifier les pixels de l'image en 5 classes et attribuer une valeur de [0-4] pour chaque classe.
- Image en couleur : classifier les pixels de l'image de chaque composant en 5 classes et attribuer une valeur de [0-4] pour chaque classe.

Unité d'Automate Cellulaire : matrice de 3×3 .

2. **Initialisation du AG** : construction au hasard des paquets de règle extraits à partir du modèle de voisinage figuré dans l'ensemble de l'image (voir la FIG.5.5).

- Méthode de détection de contour : pour chaque AC, le processus est d'abord de rechercher parmi les paquets courants ; la règle similaire selon son voisinage. En second lieu, il modifie le Pixel central selon la transition définie.
- Evaluation du résultat de détection de contour : nous calculons la distance entre le résultat de détection de contour et l'idéal considéré pour évaluer l'approche. Nous pouvons également évaluer l'erreur des Pixel mal-classés.

3. **Reproduction** : générer une nouvelle population en appliquant la sélection, le croisement et la mutation. Nous utilisons la détection de contour décrite ci-dessus dans le processus d'évaluation.
4. **Critère d'arrêt** : le processus réitère jusqu'à ce qu'il n'y ait aucune amélioration dans la fonction objective (seuil de similarité), ou pour un maximum fixe des itérations.
5. **Le résultat** : paquet optimal des règles.

5.3.2.1 Principe de base

Nous avons présentés ici le principe de base de l'approche GACEV-DC.

5.3.2.1.1 Format de règles

Nous représentons la règle de transition d'un AC par la concaténation des états des cellules du voisinage immédiat de la cellule à mettre à jour. Nous ajoutons l'état futur de la cellule après mise à jour [More2001]. Cette règle est transformée sous la forme d'un chromosome (linéaire) voir la FIG.5.3



FIG. 5.3 – Règle de AC, voisinage et état futur.

La règle est exécutée, lorsque sa partie voisinage coïncide avec un patch de même dimension de l'image. Alors, nous remplaçons le pixel central du patch par la valeur de l'état futur dans la règle. La correspondance entre la partie voisinage de la règle et un patch de l'image se fait modulo des opérateurs rotationnels (rotation à 90 deg, 180 deg, 270 deg, retournement "flip-flop" horizontal et vertical). Les règles sont donc symétriques illustrées par la FIG.5.4

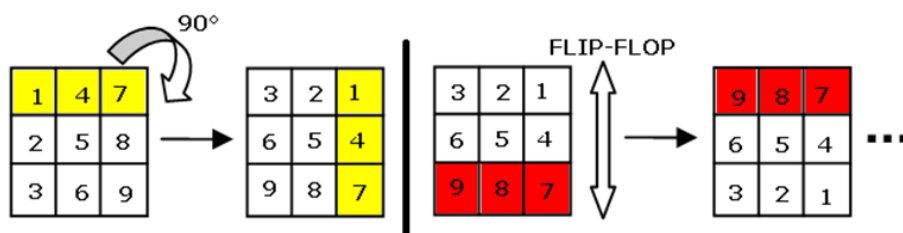


FIG. 5.4 – Equivalence rotationnelle des règles.

5.3.2.1.2 Initialisation de système

Dans cette étape, l'initialisation de système consiste à construire un modèle de voisinage.

5.3.2.1.2.1 Construction du modèle de voisinage

Chaque individu de la population est représenté par un chromosome. Le chromosome est simplement le vecteur de transition du modèle de voisinage (voir la FIG.5.5). Pour trouver toutes les combinaisons possibles de vecteur de transition selon la taille du modèle de voisinage, Kotecha [Kote2003] propose une règle qui définit la taille de la population.

Remarques

1. Pour le cas des images noires (0) et blanches (1) (le cas binaires), le nombre de combinaisons possibles pour construire l'espace de recherche sera diminué par rapport au cas général.
2. Pour rendre AC déterministe nous ajoutons la contrainte suivante : chaque règle du paquet doit être différente des autres (modulo les opérateurs de rotation)[SLKa2009A].

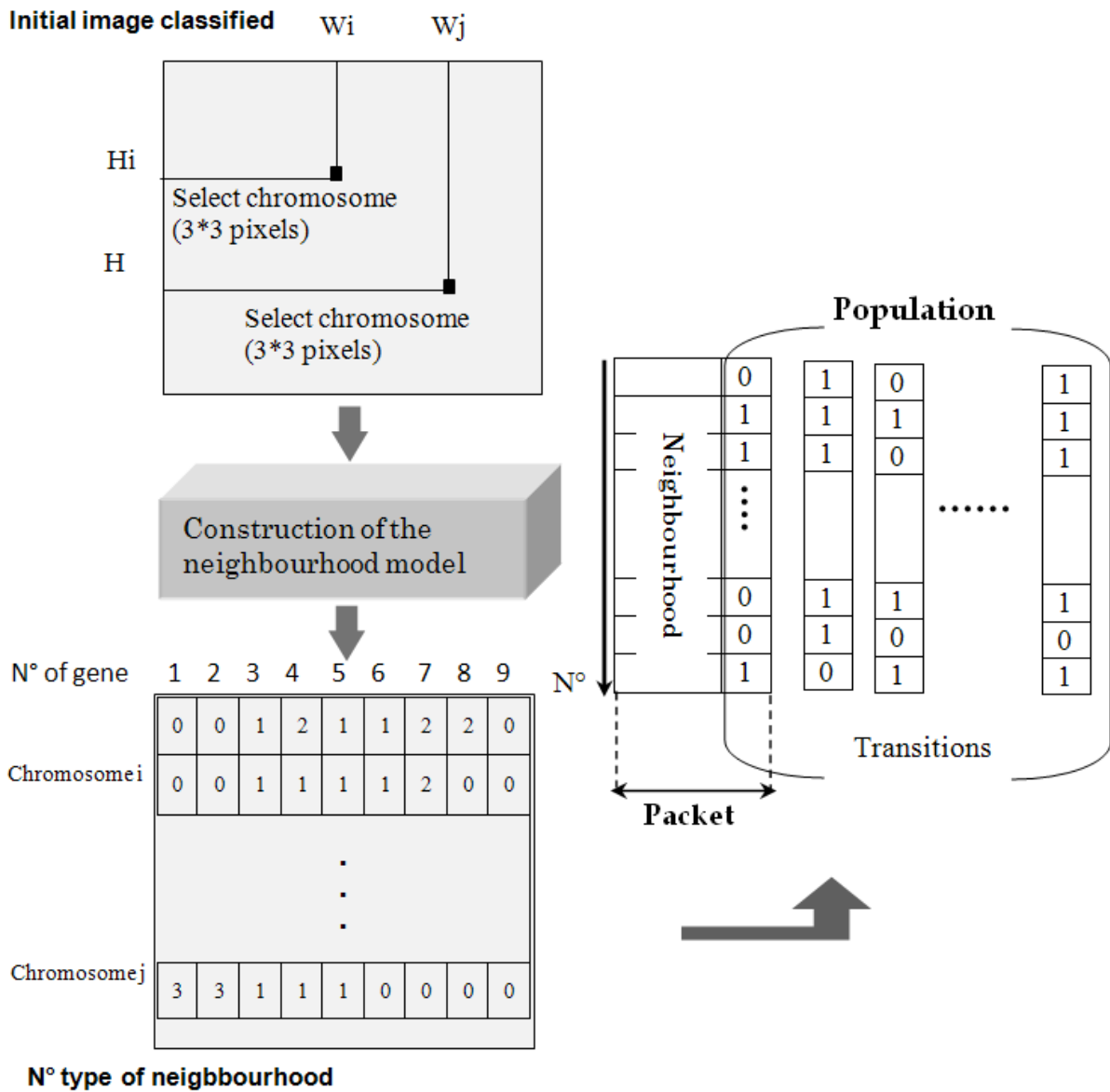


FIG. 5.5 – Construction du modèle de voisinage.

5.3.2.1.2.2 Population

Grâce à l'utilisation du concept d'AG, on a besoin d'une population initiale qui est composée d'ensemble de chromosomes. Chaque chromosome est initialisé aléatoirement par la valeur : 0 ou 1.

5.3.2.1.2.3 Construction de séquence d'exécution de règle

Chaque pixel dans l'image et ses voisins représentent un AC (matrice 3 * 3), cette matrice sera définie comme une séquence d'exécution de règles (voir la figure ci-dessous).

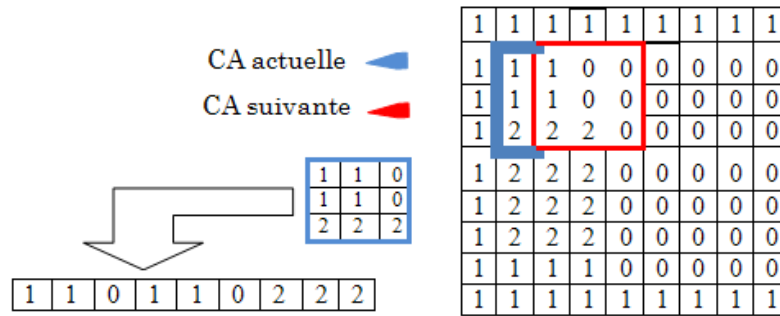


FIG. 5.6 – Construction de séquence d'exécution de règle.

L'exécution d'une règle c'est le changement d'état de la cellule centrale de AC par rapport à la règle correspondante.

5.3.2.1.3 La reproduction

Dans le processus d'optimisation l'étape de reproduction doit être assurée par ces deux opérateurs principaux.

5.3.2.1.3.1 L'opérateur de croisement

Pour donner la possibilité d'explorer une vaste surface de l'espace de recherche la représentation des chromosomes peut présenter deux types de croisement, un croisement horizontal et l'autre vertical. De par la forme du modèle de voisinage, nous utilisons l'opérateur de croisement horizontal. On prend deux vecteurs de transitions parents et on forme deux vecteurs de transitions fils en échangeant quelques états futurs de la cellule centrale.

Ceci permet de combiner les meilleures règles pour obtenir de meilleurs individus. Cette stratégie préserve une certaine forme de cohésion et de stabilité à l'opérateur de croisement.

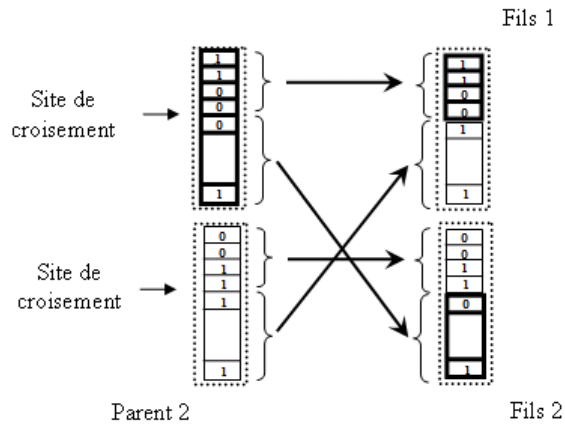


FIG. 5.7 – Opérateur de croisement horizontal.

5.3.2.1.3.2 L'opérateur de mutation

L'application d'une mutation à une transition, peut se faire en changeant l'état d'une cellule quelconque d'une transition quelconque (Fig.5.8). Ce changement (mutation) est réalisé en fonction d'une certaine probabilité. Bien sûr des précautions sont prises pour garder l'intégrité du paquet de règles (la redondance et la contradiction). Il doit toujours être valide et exécutable.

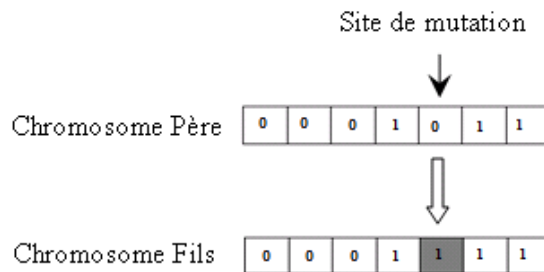


FIG. 5.8 – Opérateur de mutation.

5.3.2.1.4 La fonction Fitness et l'opérateur de sélection

La convergence de processus d'optimisation se base sur le bon choix de la fonction d'adaptation.

5.3.2.1.4.1 La fonction Fitness

L'évolution de l'évaluation de la qualité de segmentation par le processus évolutionnaire de AG sur AC qui représente l'image à segmenter est un processus interdisciplinaire. Ce processus peut utiliser l'image étalon comme un repère d'acceptation de la segmentation guidé par des critères bien définis, la précision du choix de ces critères (fonction Fitness) joue un rôle très important pour arriver à une meilleure segmentation. Cette fonction sera en fait construite à partir du critère que l'on désire optimiser. Pour ce fait, nous utilisons comme critère de fitness l'une de ces fonctions suivantes :

– **Méthode 1 [SLKa2008] :**

$$Err = nbrdepixels(image_{seg} \neq image_{ideal-seg}) \quad (5.1)$$

– **Méthode 2 :**

$$Err = Distance_{entreobservations}([image_{seg}; image_{ideal-seg}], "Metric") \quad (5.2)$$

$$Metric = 'Euclidean', 'Hamming' \quad (5.3)$$

– **Méthode 3 [Rosi2006] :**

FP = pixels blanches incorrectement classés.

FN = pixels noirs incorrectement classés.

L, H : Largeur et l'hauteur de l'image.

$$W = \frac{FP}{(L \times H)}$$

$$B = FN(L \times H)$$

$$Err = W + B \quad (5.4)$$

– **Méthode 4 :**

TP = pixels correctement classés.

Pourcentage de construction de détection de contours :

$$BDP_{contour} = \frac{TP}{(TP + FN)}$$

Pourcentage de construction de détection de font :

$$BDP_{fond} = \frac{FP}{(FP + FN)}$$

$$Err = BDP_{contour} \times BDP_{fond} \tag{5.5}$$

$$F = 1 - \frac{Err}{L \times H} \tag{5.6}$$

Autrement dit, la fonction Err présente le nombre de points trouvés inégaux dans les deux images, l'image segmentée (résultat) et l'image idéale segmentée (l'étalon de comparaison).

5.3.2.1.4.2 La fonction de Sélection

L'opérateur de sélection par tournoi donne plus de chance aux "bons" individus, les plus aptes de la population courante (génération courante). La probabilité de tirage est proportionnelle à la valeur de la fonction fitness de l'individu. Après tirage nous appliquons les opérateurs de croisement et de mutation aux individus élus afin de construire la nouvelle population.

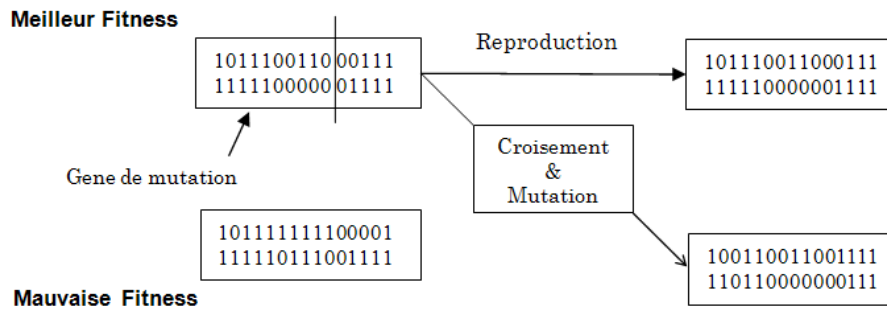


FIG. 5.9 – Sélection des chromosomes pour des opérations génétiques.

Les résultats obtenus par application de cette méthode seront abordés dans le chapitre suivant.

5.3.3 Phase d'adaptation

Elle a comme objectif d'extraire des modèles de règles. Ceci peut être fait par un ensemble des templates composées conçus manuellement ; ils ont une particularité de pouvoir figurer dans n'importe qu'elle image. pour assurer la meilleur converge nous devons renforcer le processus evolutionnaire par quelques images de synthèses et de test. Ensuite, le paquet avec la meilleur fitness sera le paquet de segmentation de contour d'image.

5.3.4 Architecture

Pour la réalisation du fonctionnement de segmentation par GACEv-DC dans des images en couleur, nous choisissons l'approche ascendante pour le développement du l'application. La tâche de segmentation est vue comme une interaction entre un ensemble des modules : "ACEv", "Adaptation Stage", "Building Optimal Packet Stage" composé par les sous modules principaux "Training Stage" et "Testing Stage", pour aboutir au module d'entrée/Sortie. Chaque module permet de générer un fonctionnement bien défini (voir la figure ci-dessous).

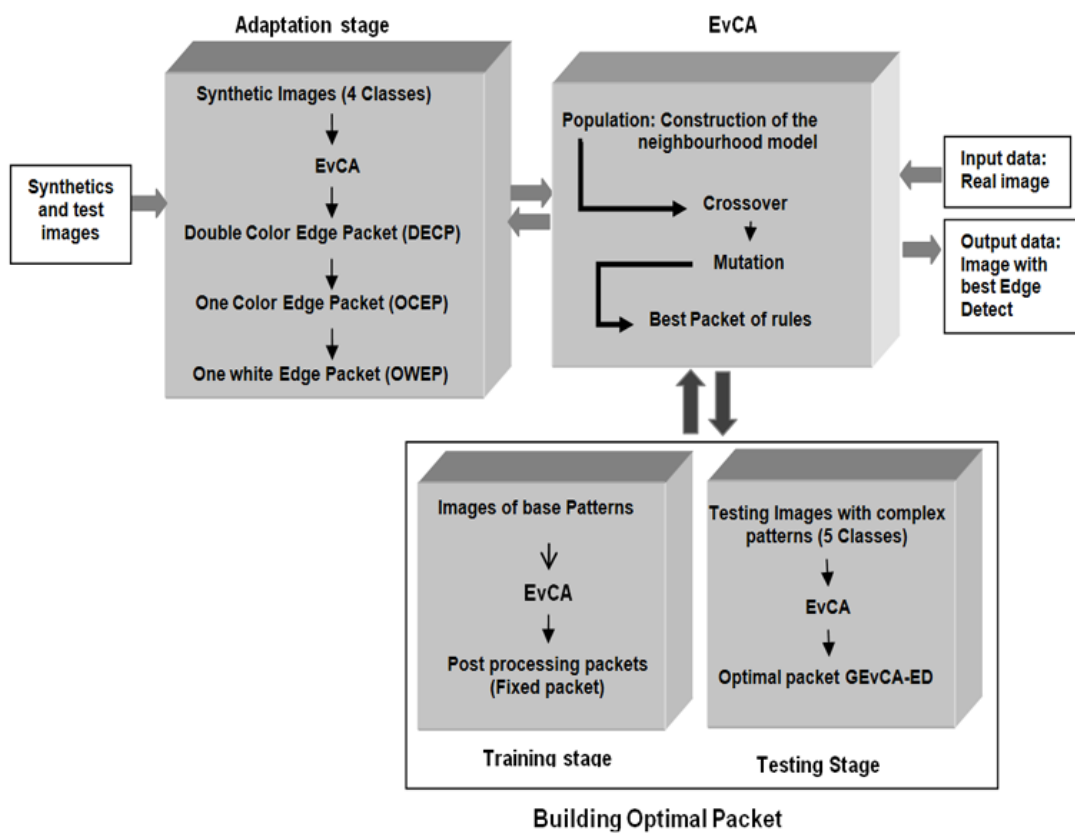


FIG. 5.10 – La Généralisation des Automates Cellulaires Evolutionnaires (GACEv-DC) pour la Détection de Contours dans des images en couleur.

5.4 Conclusion

Dans ce chapitre, notre modèle proposé de AC appliqué dans la détection de contour fournit des perspicacités sur la façon dont des processus évolutionnaires peuvent être utilisés pour découvrir les modèles locaux qui provoquent la détection optimale de contour d'une image donnée en couleur.

En effet, l'une des propriétés structurales locales a été identifiée par l'intermédiaire du AG et nous a permis d'analyser l'émergence évolutionnaire du calcul sophistiqué.

Chapitre 6

IMPLÉMENTATION ET RÉSULTATS EXPÉRIMENTAUX

Ce chapitre est consacré à la description en plus de l'approche proposée suivie par les résultats obtenus sur des images de synthèses et réelles. L'analyse de ces résultats expérimentaux sera également discutée.

6.1 Introduction

Les expériences suivantes sont exécutées en utilisant *MATLAB* (*Version7*) sur un *Pentium4*. L'unité centrale de traitement 1 : 70GHz avec 256MB. GACEv-DC utilise le paquet optimal GACEv-DC pour mieux extraire les contours dans tous les résultats expérimentaux exécutés sur toutes les classes des images (noir-blanc, en niveau de gris et en couleur). Le paquet optimal retenu est représenté par le voisinage et l'état futur de la cellule centrale. Les règles sont symétriques du fait qu'elles gardent le même résultat quelque soit la rotation qu'on applique au patch correspondant au voisinage.

6.2 Paquet de prétraitement

Nous avons un paquet optimal pour chaque image examinée, mais nous avons besoin d'un paquet optimal global pour toutes les images, ainsi nous devons assembler ces paquets sur un paquet et le filtrer de la redondance, de la contradiction et de la similitude.

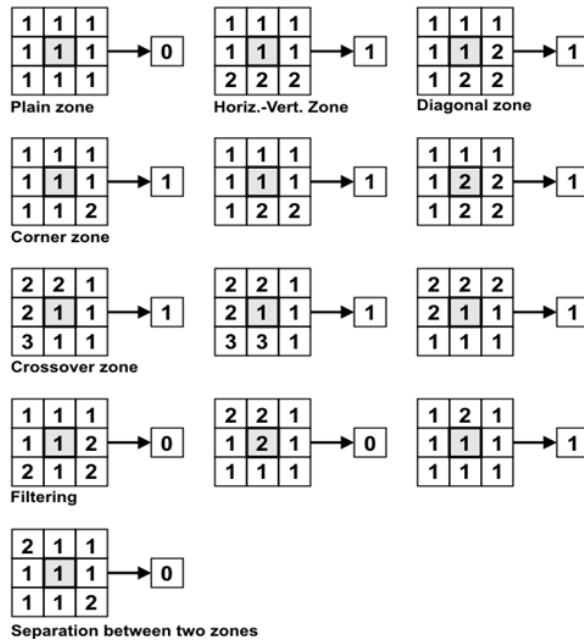


FIG. 6.1 – Paquet d'apprentissage, un meilleur paquet des règles de AC trouvé par AG. (1, 2, 3 : sont les index de couleur).

6.3 Paquet optimal GACEv-DC

Cette figure représente le paquet optimal fondé par le processus de GACEv-DC. Du côté gauche nous avons représenté seulement un sous-ensemble du paquet entier composé par 1057 règles. Nous donnons une vue tracée du paquet entier, où chaque rangée représente

une règle et la dernière colonne définit la partie de transition de la règle. Nous avons maintenu seulement les règles qui donnent des contours solides et où les patterns non figurés sont forment l'origine.

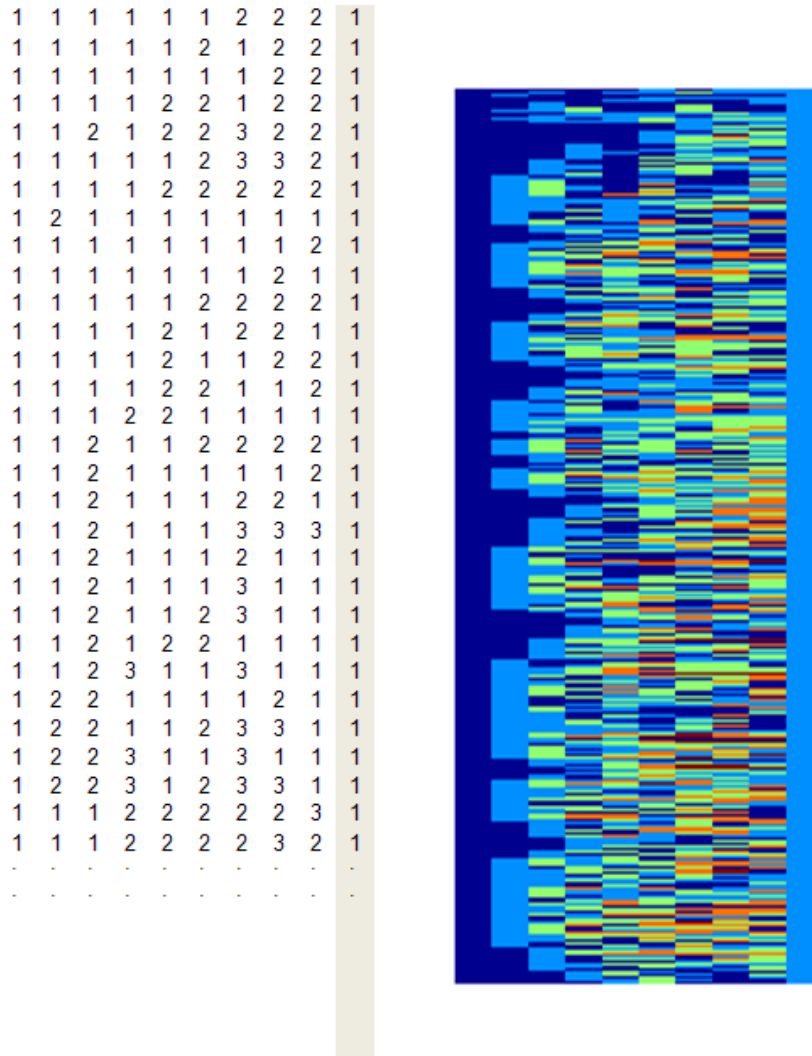


FIG. 6.2 – Paquet optimal GACEv-DC avec 1057 Règles.

6.4 Paquets d'adaptation

Le meilleur paquet retenu par ACEv pour des images en 4 Classes (en niveau de gris et en couleur) est représenté par le voisinage et l'état futur de la cellule centrale. Les règles sont symétriques du fait qu'elles gardent le même résultat quelque soit la rotation qu'on applique au patch correspondant au voisinage.

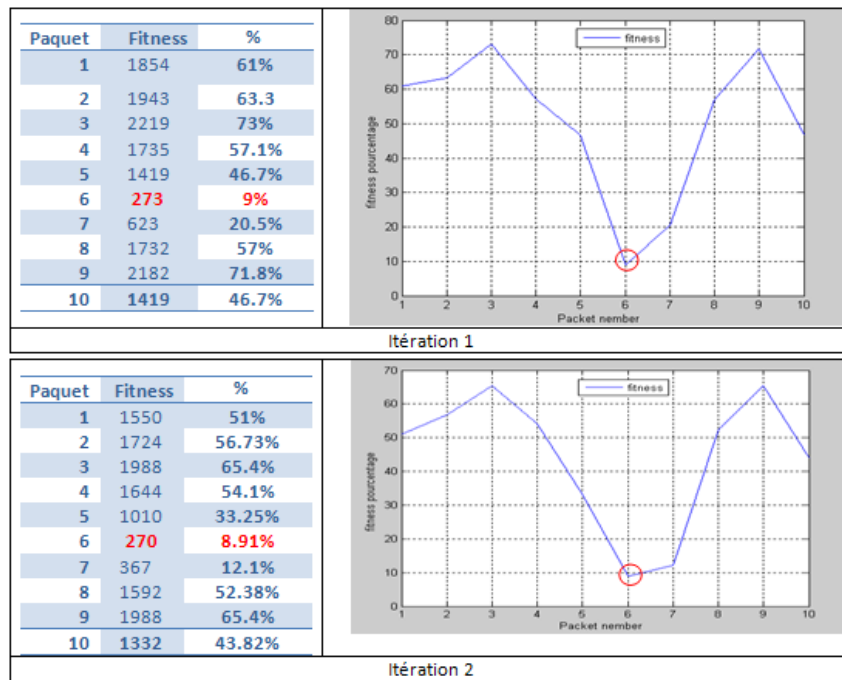
6.4.1 Paquet de Double Contour Coloré (PDCC)

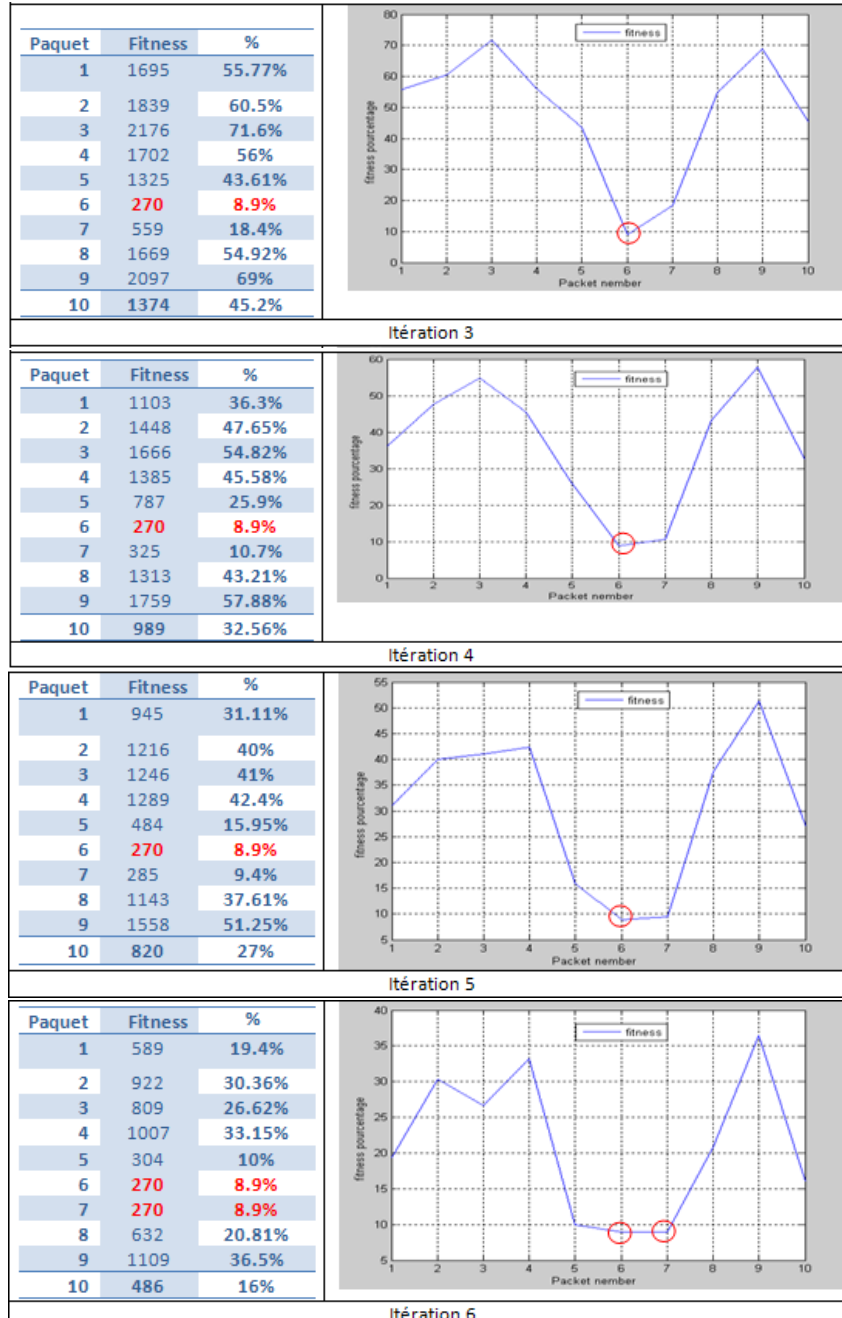
Dans ce contexte, On lance l'exécution de processus d'AG avec un nombre donnée d'itérations, où la taille de population était 50 paquets de règles, et chaque paquet contient 232 règles symétriques. En fin, on obtient 50 paquets résultats d'application ACEv avec des divers fitness. Nous choisissons pour la représentation les 10 paquets qui ont un seuil de similarité répétitifs.

Le meilleur Paquet est le paquet qui a le meilleur fitness (plus petit nombre de pixels erronés). On obtient, notre meilleur paquet dans la génération numéro 10 avec un fitness de 270 pixels erronées par rapport au nombre des pixels de l'image étalon de 32x95 pixels, compose entre les trois images d'apprentissage (voir la figure FIG.6.3). Ce paquet doit créer un contour doublé de même couleur de l'image originale (garde de sens d'information).



FIG. 6.3 – Image d'apprentissage collectif.





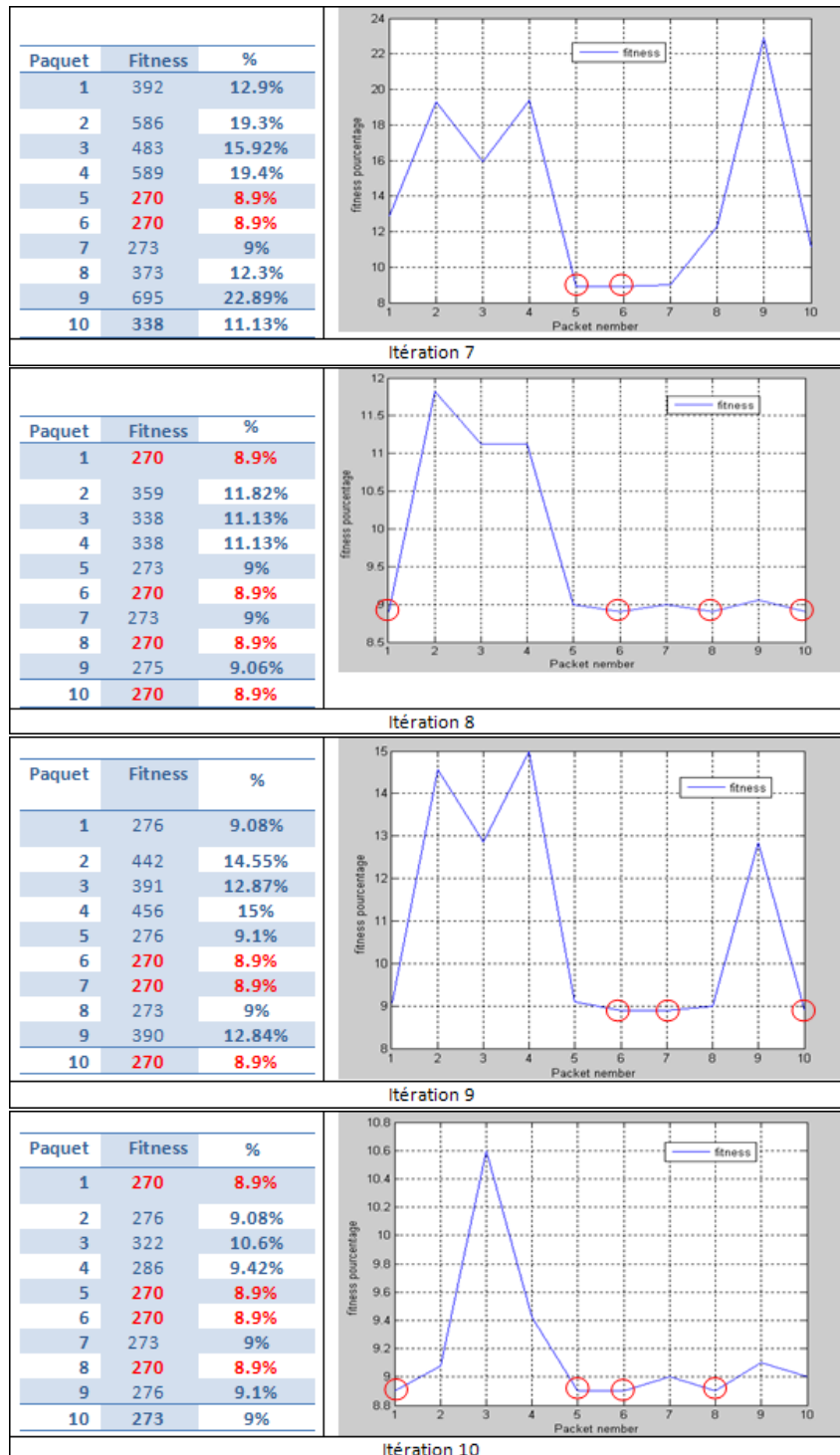


FIG. 6.4 – L'évolution de Fitness par générations pour des images de la FIG.6.3.

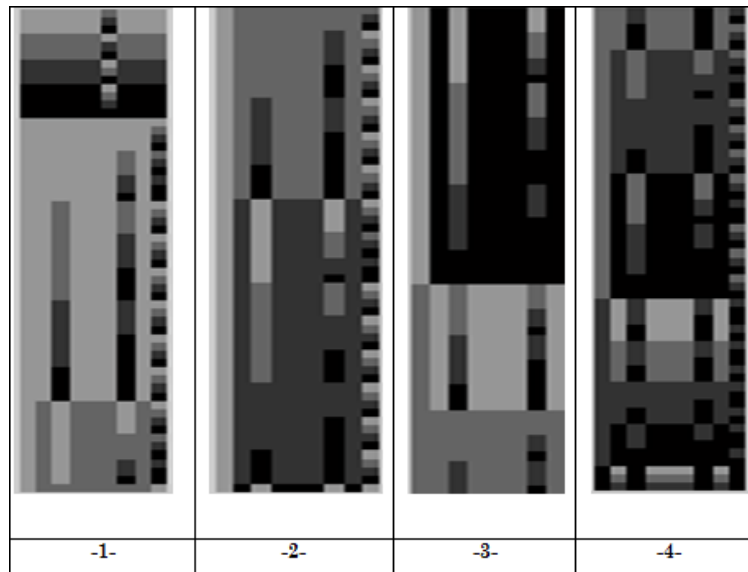


FIG. 6.5 – Paquet de Double Contour Coloré (PDCC).

6.4.2 Paquet d'un Seul Contour Coloré (PSCC)

L'effet de double contour détecté par le PDCC est le résultat de détection d'une part d'un contour extérieur et d'autre part d'un contour intérieur [SLBM2007]. Pour améliorer ces résultats, nous allons réappliquer le processus ACEv sur le PDCC qui sera étendu pour obtenir un deuxième paquet PSCC de taille 908 règles responsable de créer un seul contour de la même couleur de l'image originale dont nous introduisons un pool d'entraînements d'images de synthèses pour filtrer les règles spécialisées pour une double segmentation.

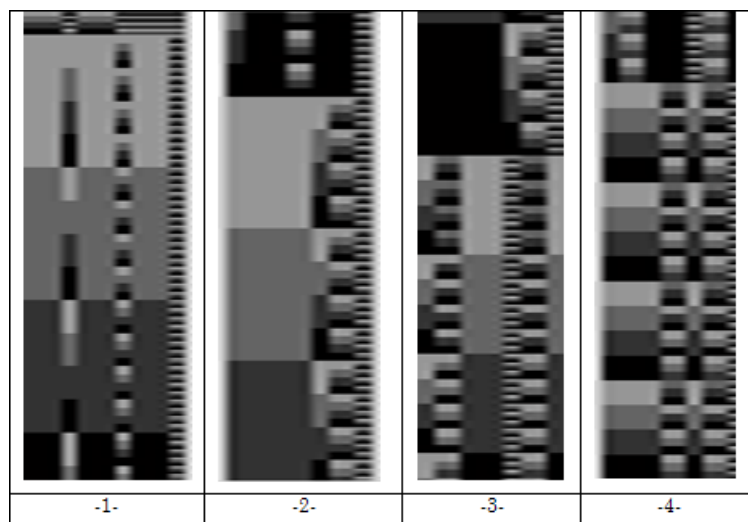


FIG. 6.6 – Paquet d'un Seul Contour Coloré (PSCC).

6.4.3 Paquet d'un Seul Contour Blanc (PSCB)

Un troisième paquet de taille 2209 règles est le PSCB qui est le résultat de l'application ACEv sur PSCC responsable pour la détection d'un seul contour blanc en se basant, d'une part sur un pool d'entraînement d'images, et d'autre part sur l'opérateur génétique mutation au niveau des gènes des vecteurs de transition pour unifier la valeur des bits de la cellule centrale de règles de transitions.

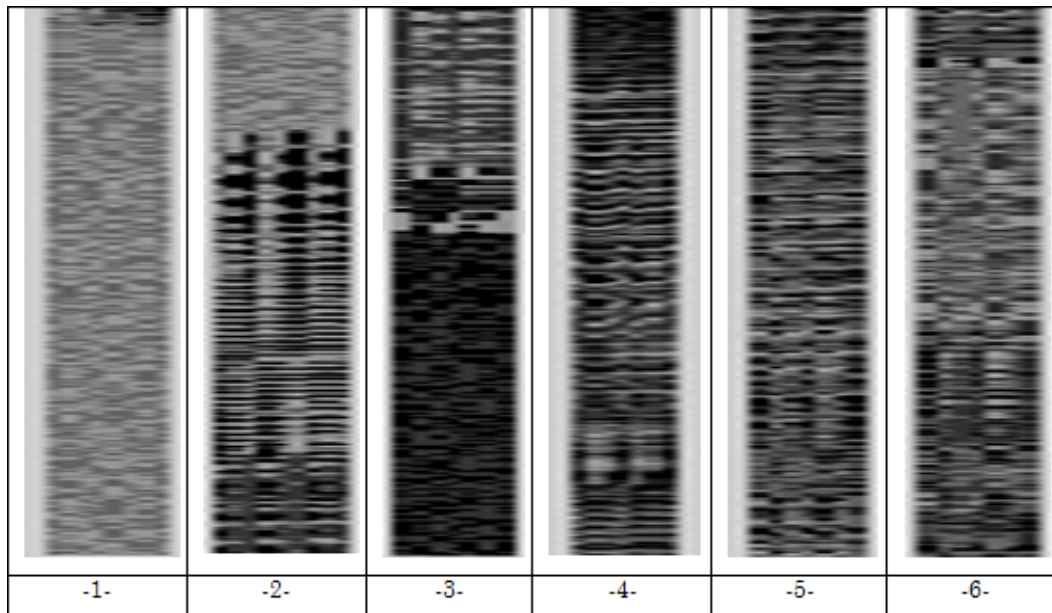


FIG. 6.7 – Paquet d'un Seul Contour Blanc (PSCB).

6.5 Présentation de l'interface de logiciel GACEv-DC

Notre logiciel contient une interface principale qui intègre deux sous interfaces, l'une pour les images en niveaux de gris et l'autre pour les images en couleurs (voir la figure ci-dessous).



FIG. 6.8 – Interface principale de logiciel GACEv-DC.

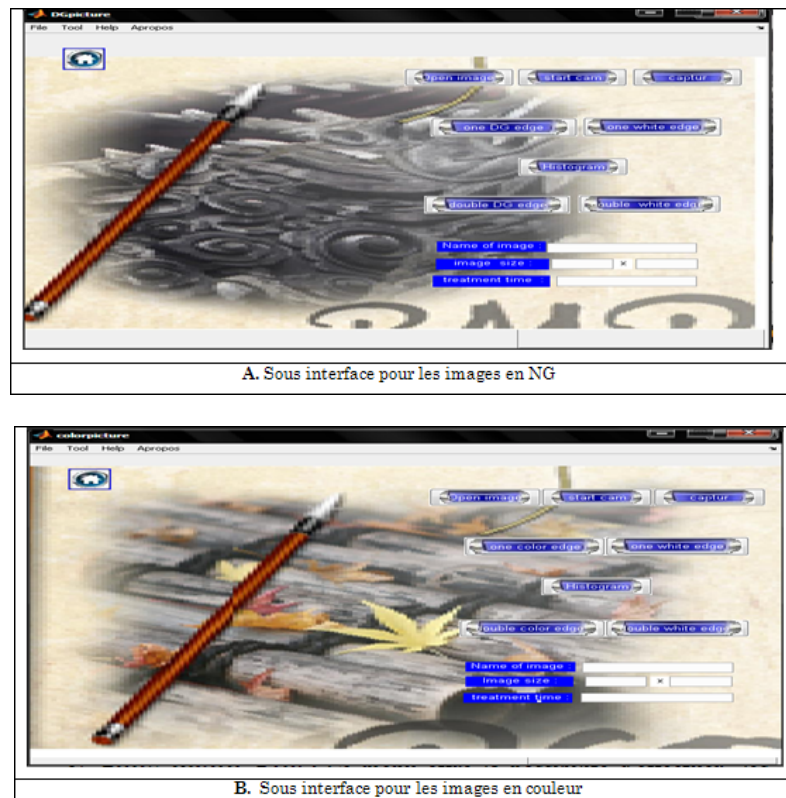


FIG. 6.9 – Présentation des sous interfaces de logiciel GACEv-DC.

6.6 Résultats expérimentaux

Nous présentons des résultats synthétiques et réels (voir les figures FIG. 6.10, 6.14, 6.15) du GACEv-DC comparé au meilleur contour de Canny.

En conséquence, pour une détection de contours de toutes classes d'images, un paquet de 1057 règles seulement est capable de donner de meilleurs résultats qui rapportent des gains substantiels. Au point de vue donnent en coût et de meilleures qualités de la détection de contours.

6.6.1 Résultats de phase d'apprentissage

Les quatre graphiques représentent la convergence du processus d'évolution du GACEv-DC contre chaque image dans l'ensemble d'apprentissage. Nous observons une convergence rapide dans le cas des images " cells " et " Vegetable-2 " et une convergence lente dans le cas des images " cameraman " et " Lina ". Pour tous les cas le rapport moyen de fitness est 92%, qui indique une très bonne de notre paquet optimal. Ainsi nous pouvons distinguer la haute qualité des images segmentées par GACEv-DC.

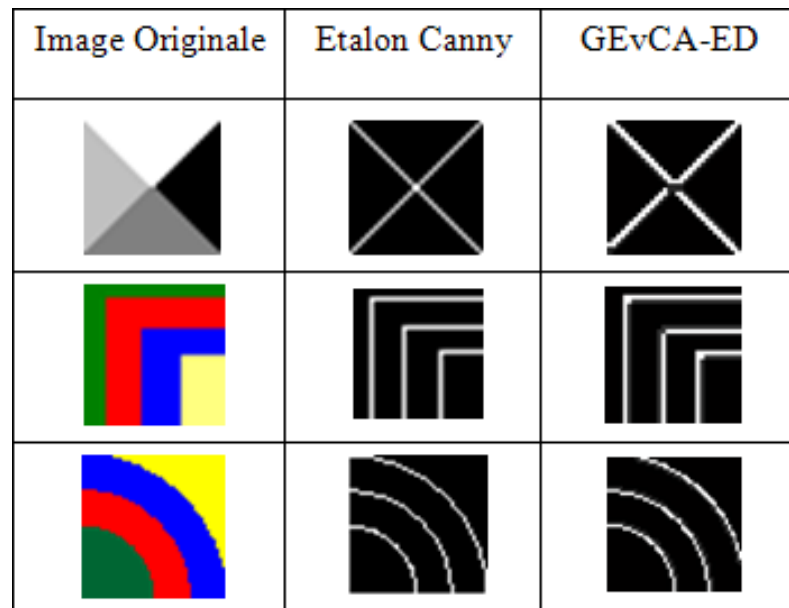


FIG. 6.10 – Détection de contour de la phase d'apprentissage de 1er ensemble d'images.

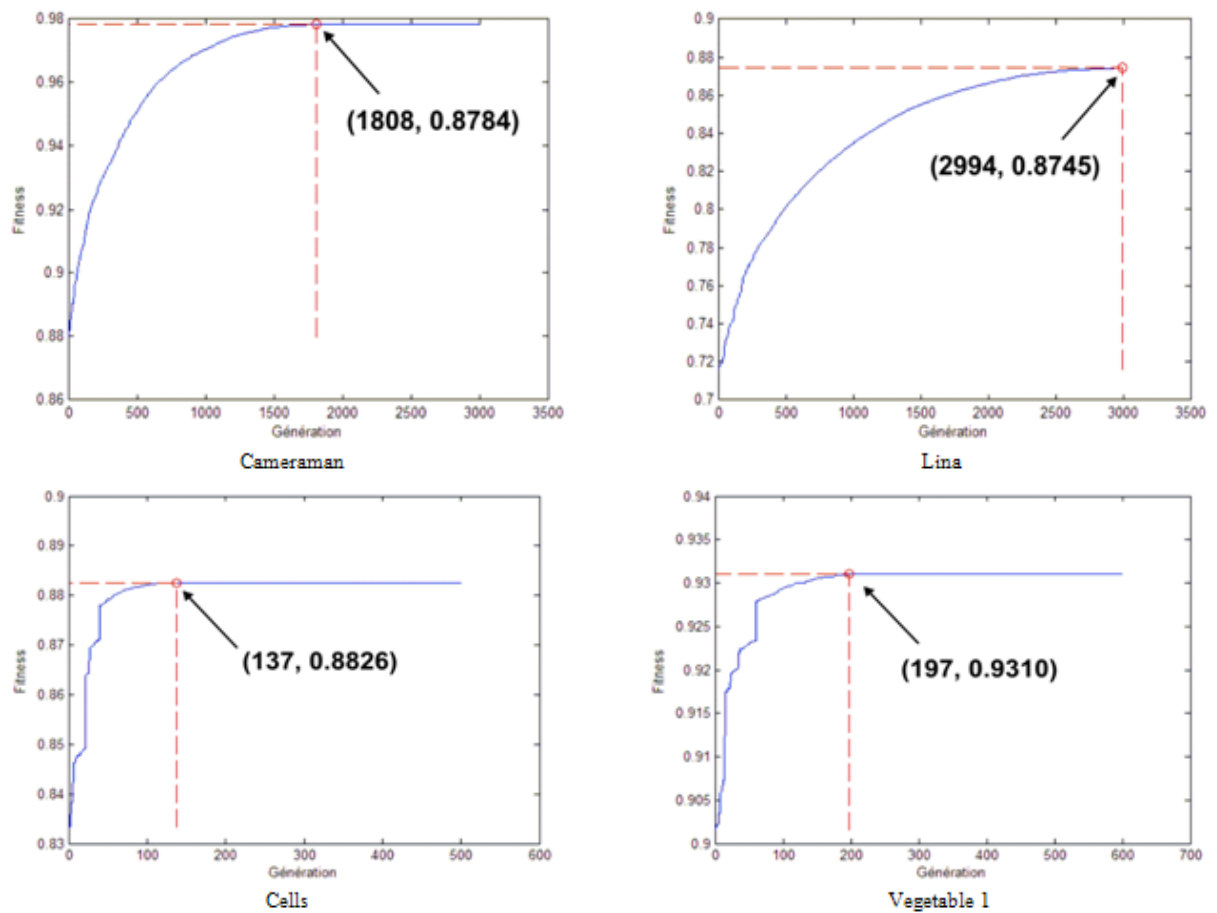


FIG. 6.11 – L'évolution de la fonction Fitness par générations pour les images de la figures 6.14, A, B, C, D.

IMAGE	FITNESS	TEMPS(SEC)	GENERATION	MAX GEN.
Cameraman	0.8784	5127	1808	3000
Lina	0.8745	136879	2994	3000
Vegetable 1	0.9310	9386	197	400
Cells	0.8826	6643	137	600

TAB. 6.1 – Les valeurs d'évaluation des expériences sont rapportées dans les figures 6.14, A, B, C, D.

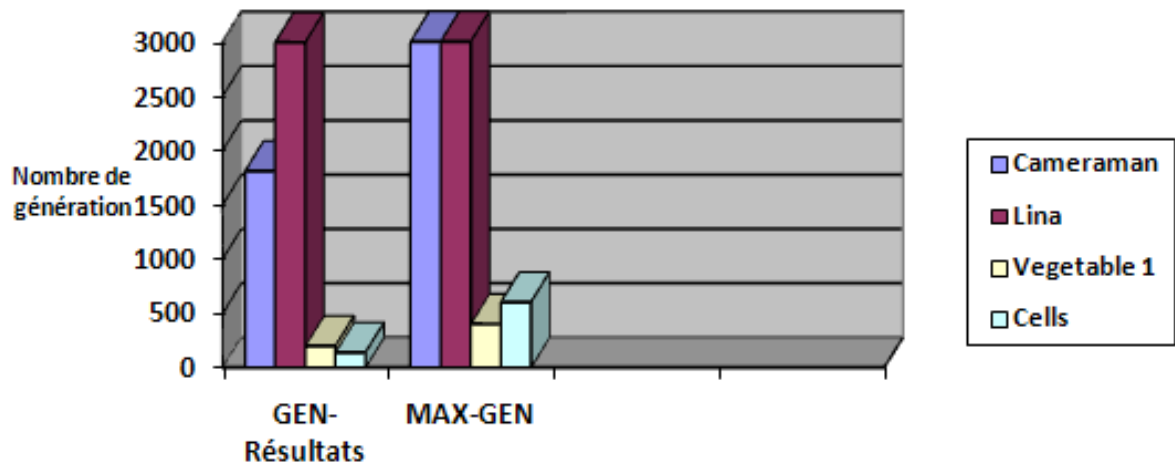


FIG. 6.12 – L'évolution de GEN-Résultats et MAX-GEN par générations pour des images de la Table 6.1 (GEN-Résultats : numéro de la génération de meilleur résultat, MAX-GEN : numéro de maximum de générations générées).

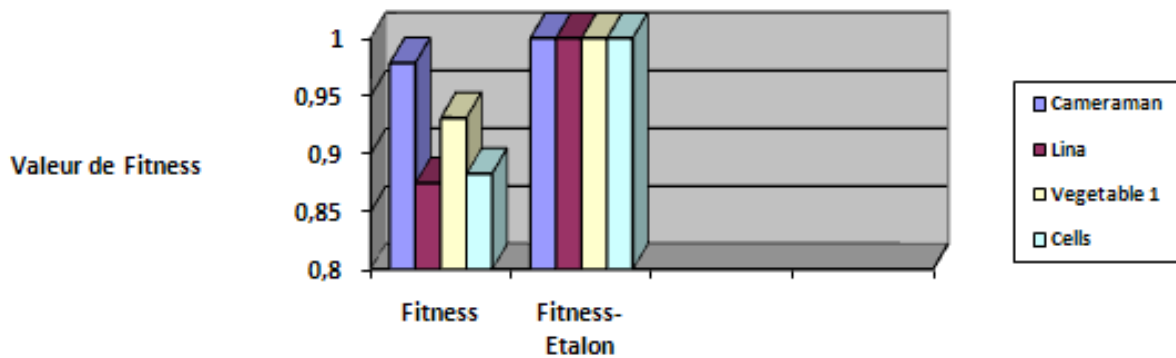


FIG. 6.13 – L'évolution de la fonction Fitness par Etalon pour des images de la Table 6.1.













Image Original	Image etalon	GEvCA-ED
		
A: Cameraman 96 × 96 pixels		
		
B: Lina 96 × 96 pixels		
		
C: Cells 96 × 96 pixels		
		
D: Vegetable 1 - 198 × 135 pixels		

FIG. 6.14 – Détection de contour de la phase d'apprentissage de 2ème ensemble d'images.

6.6.2 Résultats de phase de teste

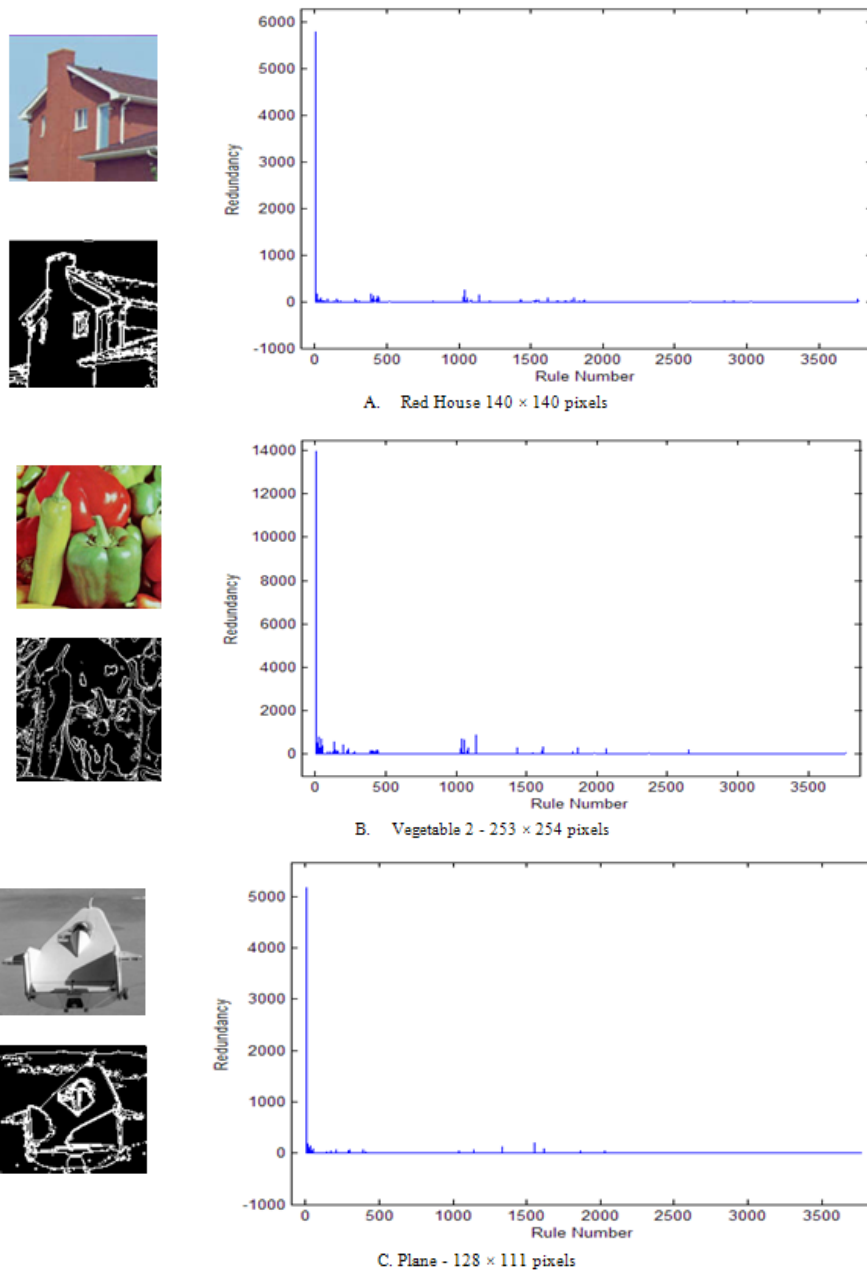


FIG. 6.15 – Détection de contour de la phase de teste de l'ensemble d'images.

Les images précédentes représentent les résultats de la segmentation par le paquet optimal trouvé par GACEv-DC. En plus, cela nous donne une représentation graphique de l'utilité (valeur de redondance) de chaque règle dans chaque contexte de segmentation d'image. Premièrement nous pouvons observer que les 10th premières règles du paquet optimal sont souvent utilisées dans toutes les procédures d'évaluation, deuxièmement l'image est semblable aux images d'apprentissage (comme vegetable-2) : les résultats de

segmentation sont les meilleurs.

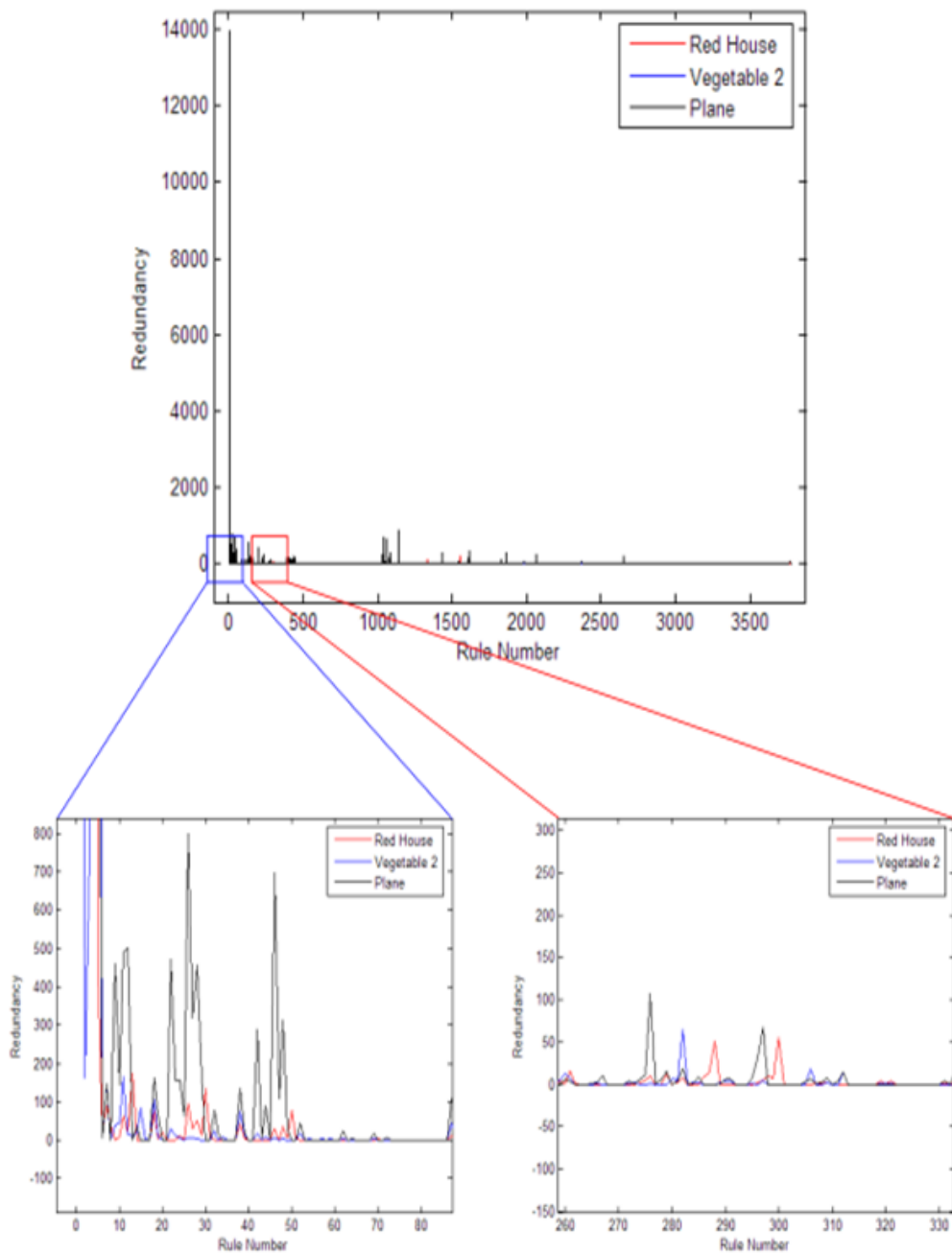


FIG. 6.16 – Evolution de la fonction Fitness par génération pour les images de la figures 6.15. A, B, C.

6.6.3 Discussion

La première image 'Apr2' présentée dans ces expériences montre quatre régions de forme rectangulaire. On peut voir que différents contours blancs des rectangles sont mieux extraits par PSCB que par le contour des PDCC , PSCC et de Canny.

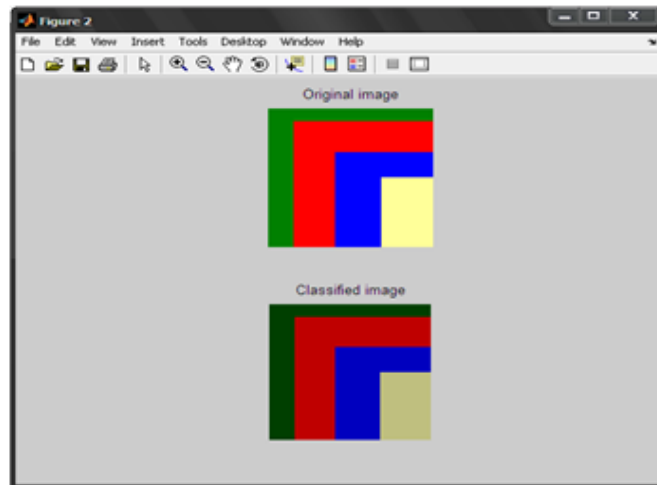


FIG. 6.17 – Résultats de classification de l'image 'Apr2'.

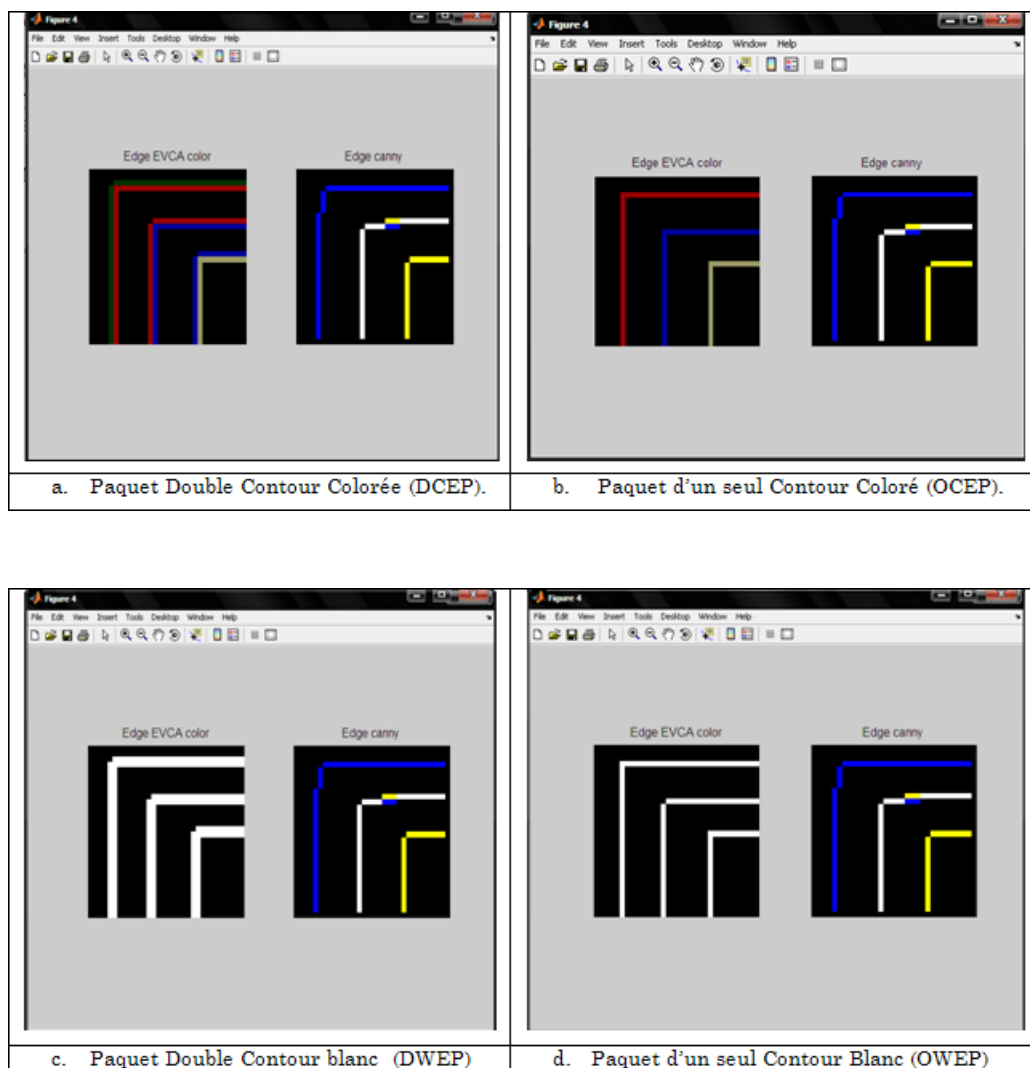


FIG. 6.18 – Les images étalon de 'Apr2'.

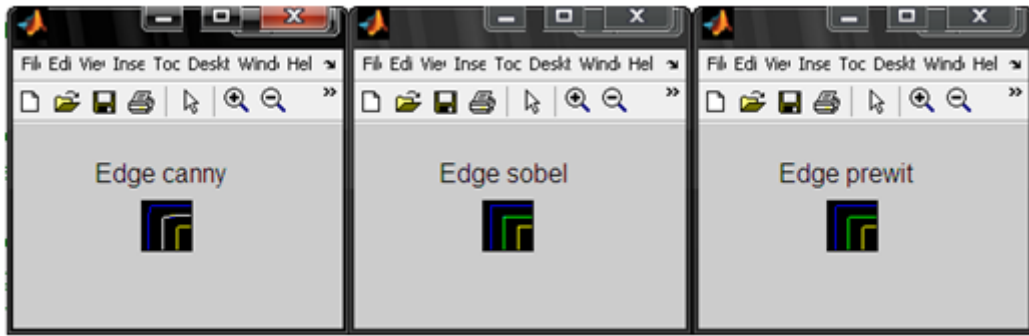


FIG. 6.19 – Les images étalon de 'Apr2'.

Les Figures FIG. 6.20, 6.21 présentent l'évaluation d'erreur d'ACEv-DC en utilisant les paquets d'adaptation PDCC, PSCC, PDCB et PSCB pour l'expérience 'Apr2'. A chaque composante de l'image, nous mettons la valeur d'évolution de la fonction fitness pour le résultat des paquets d'adaptation. Ceux-ci représentent graphiquement l'exposition claire qu'PSCB donne des résultats meilleurs de fitness sur les images en couleur.

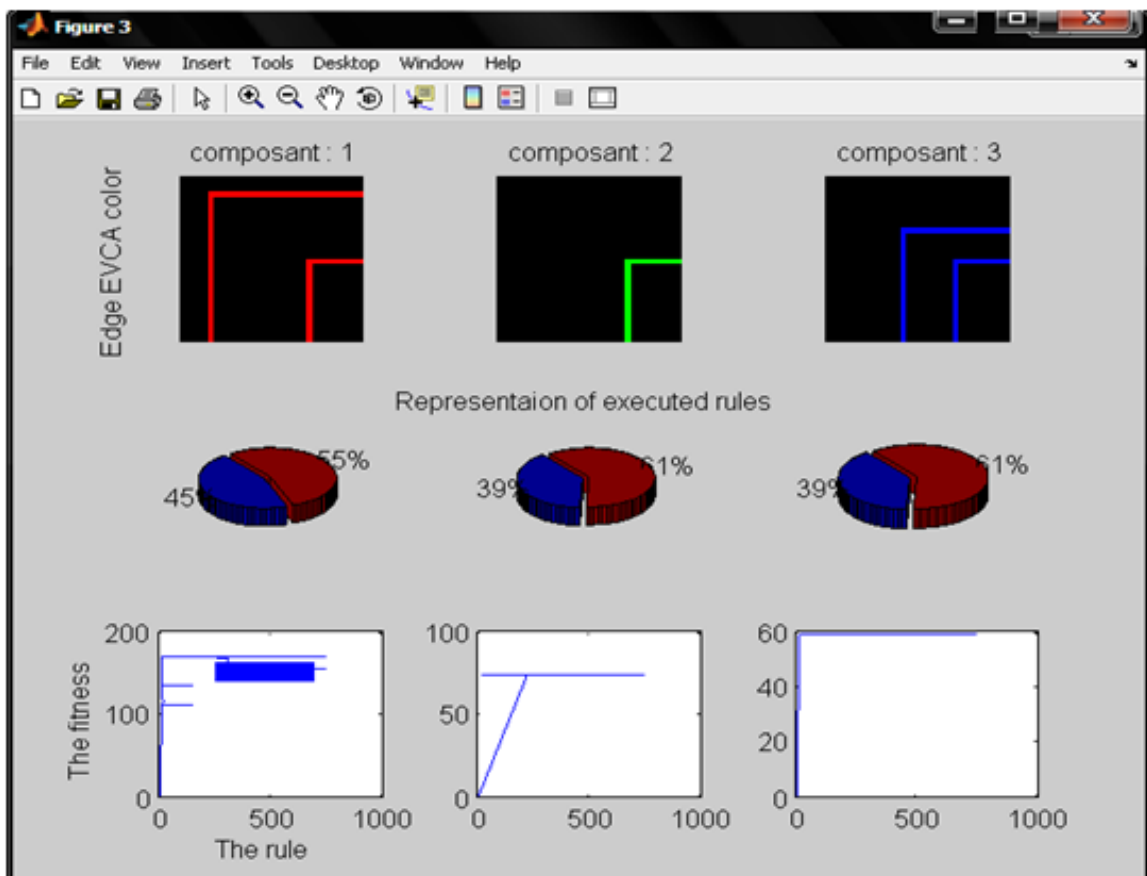


FIG. 6.20 – Diagrammes de détection de contour descriptifs des trois composants pour l'image 'Apr2'.

Dans la deuxième expérience enregistrée dans la figure FIG. 6.22, l'image observée re-

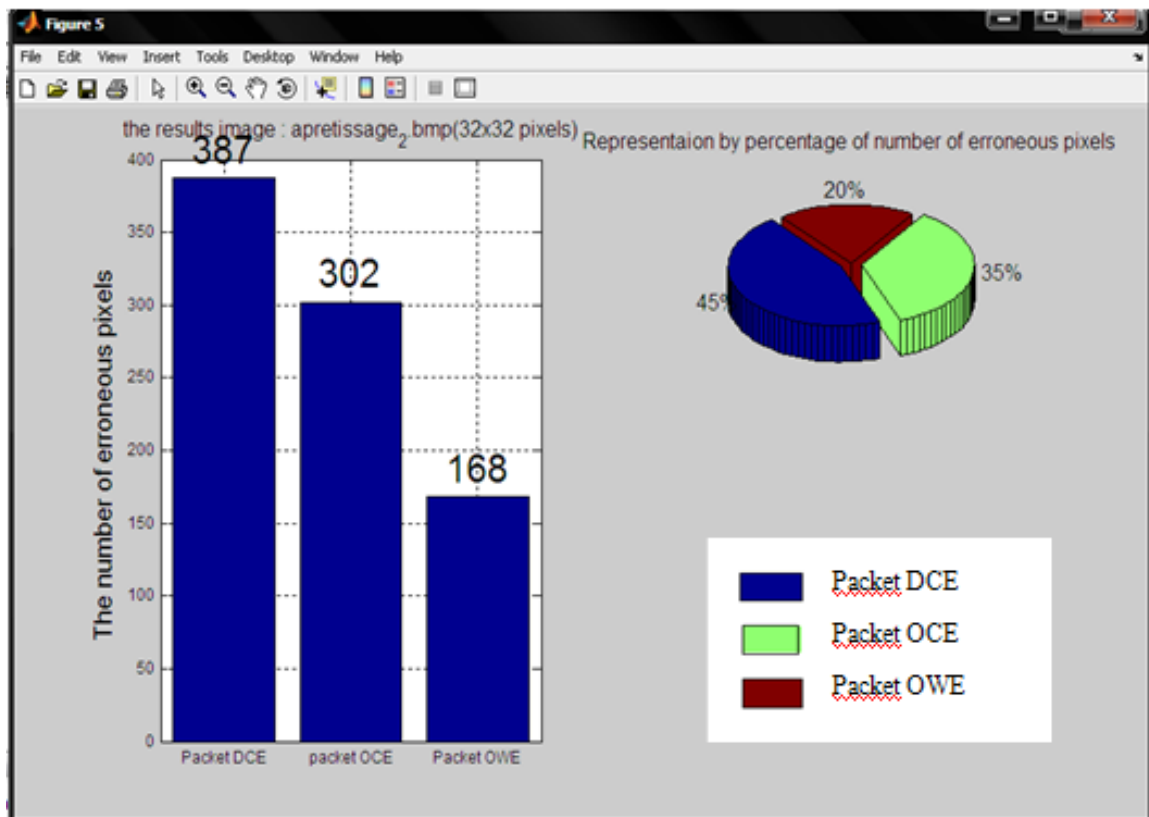


FIG. 6.21 – Histogramme de la fonction Fitness des trois paquets pour l'image 'Apr2'.

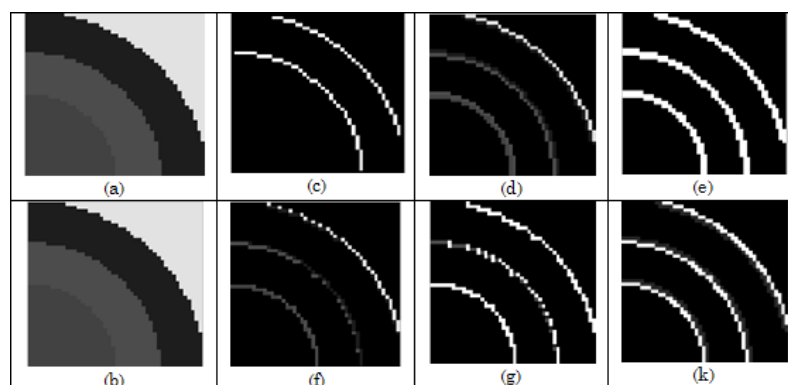


FIG. 6.22 – Détection de contour par les paquets d'adaptation a) 'Apr3' : image Originale en NG de 46*46 pixels, b) Image Classifier, c) Image Etalon Canny, d) PDCC, e) PDCB, f) PSCC, g) PSCB, k) paquet optimal GACEv-DC.

présente un ensemble des cercles avec une ombre de fonte ayant la forme d'escalier. D'une part, le résultat de PSCB est accepté comparablement aux autres résultats de PDCC, PDCB, PSCC et amélioré que par le contour de Canny. D'autre part, nous observons que le résultat de détection de contours par GACEv-DC utilisant le paquet optimal montre une continuité dans le contour détecté au niveau de l'extrémité des différents cercles avec une seule couleur en blanc.

Donc le résultat de segmentation par GACEv-DC utilisant le paquet optimal est meilleur et adéquat par rapport aux autres résultats des paquets précédents.

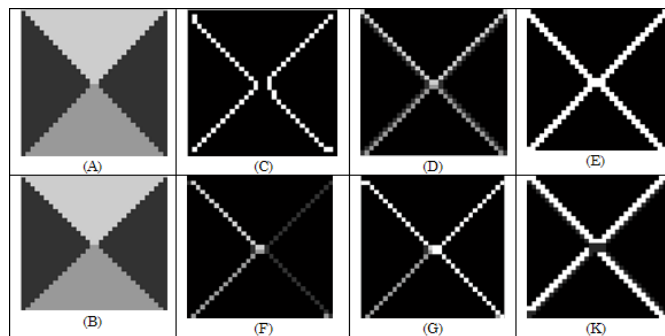


FIG. 6.23 – Détection de contour par les paquets d'adaptation a) 'Apr1' : image Originale en NG de 32*31 pixels, b) Image Classifier, c) Image Etalon Canny, d) PDCC, e) PSCB, f) PSCC, g) PSCB, k) paquet optimal GACEv-DC.

Dans cette figure FIG. 6.23, nous utilisons une image synthétique en niveau de gris contenant la forme des triangles. On peut voir que différents contours sont mieux extraits par GACEv-DC en utilisant le paquet optimal et PSCB, que le contour de Canny qui considère les deux régions en niveau de gris haut et bas comme une seule région dans la segmentation par contre le résultat de paquet optimal GACEv-DC et PSCB montre bien le contour entre ces deux régions.

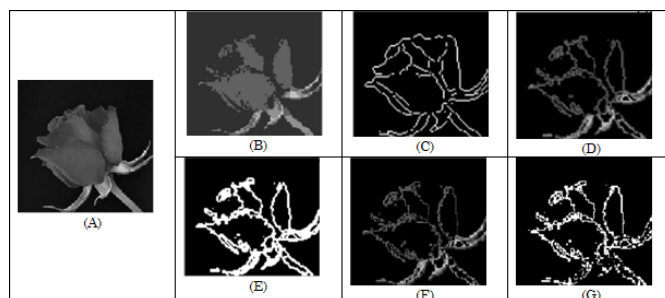


FIG. 6.24 – Détection de contour par les paquets d'adaptation a) 'Fleur' : image Originale en NG de 98*108 pixels, b) Image Classifier, c) Image Etalon Canny, d) PDCC, e) PDCB, f) PSCC, g) PSCB.

Le paquet optimal GACEv-DC est plus robuste dans cette tâche comparé aux autres paquets.

IMAGE		FITNESS	TEMPS (SEC)			
			Paquet DCE	Paquet DWE	Paquet OCE	Paquet OWE
Apr1	NG	200	26.82	127.07	94.85	26.82
	Couleur	105	04.41	09.40	02.65	06.96
Apr2	NG	134	01.24	04.13	08.16	23.94
	Couleur	117	54.39	58.99	80.12	149.23
Apr3	NG	194	54.39	58.99	80.12	149.23
	Couleur	140.66	04.16	32.39	03.64	19.06
Fleur	NG	1469	05.61	91.46	1915	09.65
	Couleur	194	21.79	91.46	01.82	55.3

TAB. 6.2 – Les valeurs d'évaluation des expériences sont contenues dans les figures 6.23, 6.18, 6.22 et 6.24

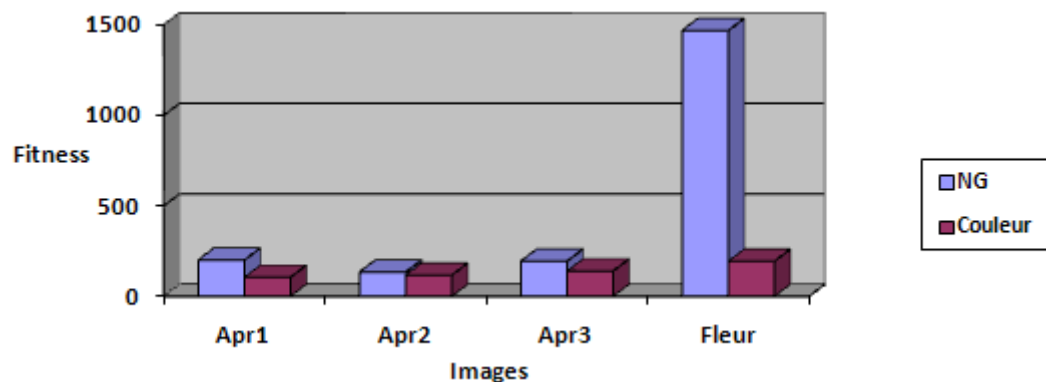


FIG. 6.25 – Evaluation comparative de la fonction Fitness de chaque image en niveau de gris et en couleur.

Pour plus d'illustration sur les résultats intermédiaire de détection de contour en couleur par PDCC, PSCC, PDCB et PSCB dans des images en couleur, la figure suivante présente les différents contours des régions en couleur des images de synthèse.

Image Originale en couleur			
Image Classifier			
Image Etalon Canny			
DCEP			
DWEP			
OCEP			
OWEP			

FIG. 6.26 – Résultats de détection de contour par les paquets d'adaptation dans des images en couleur.

6.7 Conclusion

Dans ce chapitre, nous avons démontré que la détection de contour peut être interprétée comme un phénomène évolutionnaire d'un ensemble de AC, contrôlées par les meilleurs paquets des règles appliquées sur une catégorie d'image.

En effet, l'image peut être modélée par un AC que nous pouvons mettre en application facilement. Ce meilleur paquet des règles trouvées par la GACEv-DC relié aux classe d'images.

Le phénomène d'émergence est clairement été évident dans le fait qu'une règle peut détecter des contours dans une classe donnée d'images. On peut résumer, que dans un problème donné chaque paquet a un degré puissant.

Chapitre 7

CONCLUSION GÉNÉRALE

Ce travail est consacré au domaine de la segmentation d'images basée sur les Automates Cellulaires et les Algorithmes Evolutionnaires. Dans ce travail, nous avons proposé une nouvelle approche évolutionnaire basée sur le concept d'émergence pour la détection de contours des images binaires. Particulièrement, nous avons utilisé les ACs comme un outil de modélisation puissant pour les images. Après avoir considéré l'image comme un AC, l'idée principale est de trouver les meilleures règles locales de transition qui détectent des points de contours dans une image. L'originalité de l'approche proposée repose sur une classification des règles du meilleur paquet selon leur puissance de détection de contour. Le résultat trouvé est dans la spécification des règles. En effet, un seul paquet que nous avons appelé le paquet optimal GACEv-DC est capable de détecter le contour avec un meilleur fitness par rapport aux autres paquets.

L'image peut être modélisée par un AC que nous pouvons mettre en application facilement.

Ainsi, nous pouvons dire que pour une tâche globale de calcul donnée, les AG évoluent dans un meilleur paquet des règles sachant que chacun a sa valeur de robustesse.

La GACEv-DC montrée le degré d'aspect de spécialisation de AC de détection de contour. Le phénomène d'émergence est clairement évident par le fait que quelques règles peuvent détecter des contours dans une classe d'image donnée.

Le processus de AG peut ne pas converger, car il est lié étroitement à l'image bordée par étalon ; premièrement parce que le AC utilise des règles globales générales, deuxièmement il y a l'influence de la fonction d'égalité utilisée comme mesure de fitness dans le processus de AG. Ainsi le choix de l'image contour étalon pour des images complexes

peut être non efficace pour une image contour "parfaite".

GACEv-DC donne des frontières toujours incluses et jette des régions minces. ACs fonctionne de manière synchrone avec les mêmes règles globales sans interférence ou l'échanger d'information. Ainsi ils produisent toujours les mêmes contours désirés.

Le plus important avantage du GACEv-DC est qu'il donne un paquet acceptable pour des images synthétiques et de détails très bas.

Les résultats numériques présents de l'approche proposée sont encourageantes et nous incitent à étendre notre approche aux images de faible homogénéité (avec texture et avec bruit).

Une première amélioration de cette approche est de la réalisation en matière d'architectures parallèles des ACs offre à l'approche GACEv-DC une robustesse au niveau du temps de calcul.

Dans une seconde perspective, nous projetons d'utiliser notre approche GACEv-DC en applications réelles comme par exemple l'imagerie médicale, la biométrie, la reconnaissance des formes et la robotique. En effet, notre approche GACEv-DC est extensible aux applications réelles comme par exemple la classification des données.

Bibliographie

- [Abb2006] Ali Abbassene, *Résolution de problèmes par émergence : application à la détection de contours*, mémoire de magister, L'Institut d'informatique de Mentouri Constantine, 2006.
- [Ali1997] S.M Ali, R.M Zimmer, *The question concerning emergence*, CASYS'97, Abstract Book, First International Conference on Computing Anticipatory Systems, CHAOS, 1997.
- [Amma2007] Ammar Lhlouhi, *Systèmes complexes Modèles et applications*, Cours de magister en informatique, Option : Systèmes d'informations avancés et systèmes distribué, Département d'informatique, Université de BISKRA, 2007.
- [Baill2003] Jean-Christophe Baillie, *Segmentation*, Support de Cours, ES322-traitement d'image et vision artificielle, ENSTA, 2003.
- [Bato2005] M. Batouche, *Algorithmes et complexité*, Cours de magister en informatique, Equipe Vision Infographie, Université Mentouri - Constantine, 2005.
- [Bato2005A] M. Batouche, *SYSTEMES COMPLEXES : Résolution de problèmes*, Cours de magister en informatique, Equipe Vision Infographie, Université Mentouri - Constantine, 2005.
- [Bato2005B] M. Batouche, *Vie Artificielle*, Cours de magister en informatique, Equipe Vision Infographie, Université Mentouri - Constantine, 2005.
- [BarY1997] Y. Bar-Yam, *Dynamics of Complex Systems*. Reading, MA : Addison- Wesley, 1997.
- [Bure2001] Gilles Burel, *Introduction au traitement d'images*, Simulation sous Matlab, 2001.
- [Camp2000] M.C. Campos, André, *Une architecture logicielle pour le développement de simulations visuelles et interactives individu-centrées : application à la simulation d'écosystèmes et à la simulation sur le web*, thèse de doctorat de l'Université du Blaisé Pascal-Clermont II, Septembre 2000.
- [Cany1986] J. Canny, A. *Computational Approach to Edge Detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 6, Nov. 1986.
- [Cath2003] Catherine Achard, *Introduction au traitement d'images*, support de Cours, 3 ects, code MSDI4213 code Scolar : MS213. 2003.

- [Cocq1995] Jean-pierre Cocquerez, Sylvie Philipp, *Analyse d'images : filtrage et segmentation*, Masson, Paris, 1995.
- [DCVZ2000] D. Castro, L. Von Zuben, *Artificial Immune Systems, A Survey of Applications*, Technical Report DCA-RT, Department of Computer Engineering and Industrial Automation, School of Electrical and Computer Engineering, State University of Campinas, Brazil, 2000.
- [Domi1994] Dominique Cavet, *Numérisation des images*, Support de Cours, Service de l'informatique éducative CNDP - Direction de l'ingénierie éducative, Octobre 1994.
- [Fate2001] N.Fatès, *Les Automates Cellulaires : Vers une Nouvelle Epistémologie ?*, DEA Histoire des Sciences. UFR 10, Paris I Sorbonne. Septembre 2001.
- [Fabr1998] Fabrice BELLET, *Une approche incrémentale à base de processus coopératifs et adaptatifs pour la segmentation des images en niveaux de gris*, Thèse de doctorat, L'Institut National Polytechnique de Grenoble, 1998.
- [Favi2004] Pierre-Alexandre Favier, *La notion d'intentionnalité dans la spécification de comportements d'agents autonomes situés en environnement virtuel*, Thèse de doctorat de l'Université de Bretagne Occidentale, 2004.
- [Ferb1995] J. Ferber, *Les Systèmes Multi-Agents, vers une intelligence collective*, Inter-Editions, 1995.
- [Gang2001] N. Ganguly, B.K. Sikdar, A. Deutsch, G. Canright, P.P. Chaudhuri, *A Survey on Cellular Automata*, Project BISON (IST-2001-38923), Deliverable - D03, 2001.
- [Gang2003] N. Ganguly, P. Maji and P.P. Chaudhuri, *Cellular Automata Machine for Pattern Recognition*. IEEE Trans. On System Machine Cybernetics Part-A, Juillet 2003.
- [Géor2004] Jean-Pierre Géorgé, *Résolution de problème par émergence, Etude d'un Environnement de Programmation Emergente*, Thèse de doctorat de l'Université du Paul Sabatier, Toulouse, juillet 2004.
- [Géor2003] Jean-Pierre Géorgé, *L'émergence*, Rapport interne IRIT n°2003-12-R, 2003.
- [Gues2003] Z. Guessoum, *Modèles et architectures d'agents et de systèmes multi-agents adaptatifs*, Mémoire d'habilitation à diriger des recherches, décembre 2003 à l'Université Pierre et Marie Curie, 2003.
- [Gill2001] Gilles Kagan, Thierry Buquet, *Numérisation d'images*, Support de Cours, Institut de Recherche et d'Histoire des textes (IRHT), CNRS, (délégation régionale de Paris A) 2001.
- [Gold1989] Goldberg, David, *Génétiques Algorithms in Search Optimisation and Machine Learning*, Addison-Wesley, 1989.
- [GeMa1997] M.Gell-Mann. *Le quark et le jaguar*, Editions Flammarion, 1997.

- [GeMa2003] M.Gell-Mann, S.Lloyd, *Effective Complexity*. Nonextensive Entropy - Interdisciplinary Applications, Oxford Univ Press, 2003.
- [Géor2002] Jean-Pierre Géorgé, *Une approche de la programmation émergente : le calcul émergent*, Colloque des Doctorants de l'EDIT (Ecole Doctorales Informatique et Télécommunications), Toulouse, 2002.
- [Gras2005] Uli Grasemann, Risto Miikkulainen, *Effective Image Compression using Evolved Wavelets*, Department of Computer Sciences The University of Texas at Austin, GECCO'05, Washington, DC, USA, 2005.
- [GMPG2003] Jean-Pierre Géorgé, Marie-Pierre Gleizes, Pierre Glize, *Conception de systèmes adaptatifs à fonctionnalité émergente : la théorie AMAS*, Revue RIA n°4/2003, 2003.
- [Heno2003] Sophie Henon, Génard Weisbuch, *Action Concertée Systèmes Complexes en SHS*, Rapport de Recherche, CNRS, PARIS, Mai 2003.
- [Hora1995] R. Horaud and O. Monga, *Vision par Ordinateur : outils fondamentaux*, edition Hermes, 1995.
- [Holl1975] J.H. Holland, *Adaptation in Natural and Artificial Systems*, Ann Arbor, University of Michigan Press, (Second Edition : Cambridge, MA : MIT press, 1992), 1975.
- [Horn2001] P. Horn, *Autonomic computing : IBM's Perspective on the State of Information Technology*, 2001.
- [Iani1995] Ianik Marcil, *Les implications épistémologiques des théories du chaos pour l'hypothèse des anticipations rationnelles*, thèse du l'Université du Québec à Montréal, 1995.
- [Jack2001] Jacky Desachy, *Analyse d'images*, Support de Cours, Institut d'informatique, Université des Antilles et de la Guyane Pointe a Pitre, Janvier 2001.
- [Jian2003] Jian-Feng Yao, *Analyse d'image par champs de markov*, Association des Professeurs de Mathématiques de l'Enseignement Public (APMEP), Paris Bulletin de l'APMEP. Num. 448. pp. 583-602, 2003.
- [Jimé2001] F. Jiménez Morales, J.P. Crutchfield, and M. Mitchell, *Evolving 2-d cellular automata to perform density classification*, Parallel Comp., 27 :571-585, 2001.
- [Joli1997] Jean-Michel Jolion, *Traitement d'images et vision artificielle*, Rapport de Projet TELESUN, Le laboratoire Reconnaissance de Formes et Vision, INSA, REV-INSA Lyon, 1997.
- [Kamp1991] G.Kampis, *Self-modifying systems in biology and cognitive science*, Pergamon Press, Oxford, England, 1991.
- [Kote2003] K. Kotecha, Nilesh Gambhava, *Solving Precedence Constraint Traveling Salesman Problem using Genetic Algorithm*, Proceedings of National Conference on Software Agents and Embedded System, July 2003, Coimbatore.

- [KPre1984] K. Preston and M.J.B Duff, *Modern Cellular Automata*, Theory and Applications, Plenum Press, London, 1984.
- [Lang1990] C.G. Langton, *Computation at the edge of chaos : Phase transition and emergent computation*, physica D 42, pp 12-37, 1990.
- [Lang1992] C.Langton. Preface. In C. Langton et al., editor, *Artificial Life II, volume X of SFI Studies in the Sciences of Complexity*, pages xiii-xviii. Addison-Wesley, New York., 1992.
- [Luca2000] C.Lucas, CALResCo Group, *Manchester U.K. "Quantifying Complexity Theory"*, in www.calresco.org, 2000.
- [Moig1999] Jean-Louis Le Moigne, *La modélisation des systèmes complexes*, Editions Dunod, 1999.
- [Marc1997] P. Marcenac, *Modélisation de Systèmes Complexes par agent*, Technique et science informatiques, Volume 16, N°8, 1997.
- [Maro2003] Marouane Abdellaoui et Abdelhaq El Jai, *Modélisation par Automates Cellulaires d'un Système Thermoélastique*, Annals of University of Craiova, Math. Comp. Sci. Ser. Volume 30, Pages 10-20, ISSN : 1223-6934, 2003.
- [Mela2006] Melanie Mitchell, *Complex systems : Network thinking, Artificial Intelligence*, Volume 170 Issue 18, Pub. Elsevier Science Publishers Ltd, December 2006.
- [Mitc1994] Mitchell M., Forrest S., *Genetic algorithms and artificial life*, Artificial Life, 1994.
- [Mitc1996] M. Mitchell, J.P. Crutchfield, R. Das, *Evolving Cellular Automata with Genetic Algorithms : A Review of Recent Work*, Proc of the first International Conference on Evolutionary Computation and Its Applications (EvCA'96), Moscow, Russia.
- [MiNe2001] M.Mitchell, M.Newman, *Complex systems theory and evolution*, Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, NM 87501, Avril 2001.
- [Mong1995] O. Monga, R. Haurod, *Cours de Vision par Ordinateur*, 2ème éd, Editions Hermès, 1995.
- [More2001] J.A. Moreno and M. Paletta, *Evolving cellular automata for noise reduction in images*, Universidad Central de Venezuela. Caracas, Venezuela. CAEPIA'01, 2001.
- [Neum1966] John von Neumann, *Theory of Self-Reproducing Automata*, Urbana : University of Illinois Press, 1966.
- [Pavl1992] T. Pavlidis, *Why progress in machine vision is so slow*, Pattern Recognition Letters, vol. 13, p. 221-225, 1992.
- [Peba1996] Per Bak, *how nature works ?*, Springer 1996.
- [Reyn1987] C-W.Reynolds. Flocks, herds, and schools : *A distributed behavioral model*. Computer Graphics : Proceedings of SIGGRAPH '87, 21(4) :25-34, July 1987, URL : [http //www.red3d.com/cwr/papers/1987/boids.html](http://www.red3d.com/cwr/papers/1987/boids.html)

- [Reyn2001] C. W. Reynolds, *Boids : Background and Update*, in : Boids (Flocks, Herds, and Schools : a Distributed Behavioral Model), URL : <http://www.red3d.com/cwr/boids>
- [Rous2003] Jean-Jacques Rousselle, *Les contours actifs, une méthode de segmentation : Application à l'imagerie médicale*, thèse de doctorat de l'université du François Rebelais de Tours, 2003-2003.
- [Rama2006] E. Ramat, *Introduction à la modélisation et à la simulation à événements discrets Modélisation et simulation multi-agents : applications aux Sciences de l'Homme et de la Société*, éd. Lavoisier, 448 pp, Septembre 2006.
- [Renn2002] Rennard Jean Philippe, *Vie artificielle*, Où la biologie rencontre l'informatique, Vuibert, 2002.
- [Roma2003] Romaric Charton, *Des Agents intelligents dans un environnement de communication multimédia : vers la conception de services adaptatifs*, thèse de doctorat de l'Université Henri Poincaré-Nancy 1, Décembre 2003.
- [Rosi2006] L. Paul Rosin, *Training Cellular Automata for Image Processing*, IEEE TRANSACTION ON IMAGE PROCESSING, VOL. 15, NO. 7, JULY 2006.
- [Russ1995] S. Russel, P.Norving, *Artificial Intelligence : a Modern Approach*, Prentice-Hall, 1995.
- [SLBM2007] S. SLATNIA, M.C. Batouche et K.E. Melkemi, *Evolutionary Cellular Automata Based-Approach for Edge Detection*, WILF 2007 SCIP special session, In F. Masulli, S. Mitra, and G. Pasi (Eds.) : Workshop on Fuzzy logic and Application, LNAI 4578, Revised Selected Papers pp. 404-411, 2007. Springer-Verlag Berlin Heidelberg 2007.
- [SLKa2008] S. SLATNIA, O. KAZAR, *Generalised Evolutionary Cellular Automata based approach for edge detection*, In International Review on Computers and Software (I.RE.CO.S.) Praise Worthy Prize 2008, ISSN 1828-6003, Vol. 3 N.4, July 2008, Manuscript received and revised June 2008, Accepted July 2008 pp 424-428.
- [SLKa2011] S. SLATNIA, O. KAZAR, *Evolutionary Cellular Automata for image segmentation and noise filtering using genetic algorithms*, to appear in Journal Applied Computer Science and Mathematics, Issues No. 10, March 2011, Editor in chief : Doru E Tiliute, ROMANIA, 2011.
- [SLKa2008A] S. SLATNIA, O. KAZAR, *Images Segmentation based contour using EvCA approach, Evolutionary Cellular Automata*. 18 th International Conference on Computer Theory Application, ICCTA 2008, 11-13 October 2008, Papiers Choisis Révisés pp123-128. Alexandria, Egypt.
- [SLKa2009] S. SLATNIA, O. Kazar, *Images segmentation based contour using EvCA approach, Evolutionary Cellular Automata*, Courrier du Savoir, On revue Périodique

- de l'Université Mohamed Khider Biskra, Algerie 2009, ISSN 1112-3338, Vol. N°09, Mars 2009, pp 113-118.
- [SLKa2009A] S. SLATNIA, O. KAZAR, *Evolutionary cellular automata based-approach for region detection*, 5ème symposium international, Image Multimédia Application Graphique et Environnement IMAGE, 03, 04-05 Novembre 2009, Papiers Choisis Révisés pp. 75-82, 2009. Biskra, Algérie.
- [Sloo1997] P-M-A. Sloom, *A.Schoneveld, J-F. de Ronde, J-A.Kaandorp*, Large scale simulation of complex systems, Conceptual framework, 1997.
- [TPav1982] T. Pavlidis, *Algorithms for Graphics and Image Processing*, Computer Science Press, 1982.
- [TBRW2005] Thomas Back, Breukelaar R., and Willmes L. *Inverse design of cellular automata by genetic algorithms : an unconventional programming paradigm*. UPP proceedings in the 'Hot Topics' subline of LNCS, 2005.
- [Thom2005] Thomas Back, Ron Breukelear, *Using a genetic algorithm to evolve behavior in multi dimensional cellular automata*. In proceedings UC 2005, LNCS 3699, pp. 1-10, 2005.
- [Ulam1974] S. Ulam, *Random Processes and Transformations*, in : S. Ulam, Sets, Numbers and Universes, Cambridge : MIT Press , pp. 326-337, 1974.
- [Vach2000] Jean-Philippe Vacher, *Un système adaptatif par agents avec utilisation des algorithmes génétiques multi-objectifs : Application à l'ordonnancement d'atelier de type job-shop $N * M$* , thèse de doctorat de l'Université du HAVRE, Avril 2000.
- [Vijv1997] G.Van De Vijver, *Emergence et explication*, Intellectica Emergence and explanation, 1997/2 n°25, ISSN n°0984-0028, pp 185-194, 1997.
- [Vlad2005] Vladimir Vezhnevets, Vadim Konouchine, *"GrowCut" - Interactive Multi-Label N-D Image Segmentation by Cellular Automata*, Conference Graphicon, 2005.
- [Mois2001] Valvassori Moise, *Une bibliothèque d'aide à la création de simulateurs de systèmes complexes par l'approche multi-agents*, Rapport de recherche, 2001.
- [Wolf1983] Stephen Wolfram, *Statistical Mechanics of Cellular Automata*, in Review of Modern Physics 55, P. 601-644, 1983.
- [Wolf1994] Stephen Wolfram, *Cellular Automata as simple self-organizing systems (1982)*, Addison-Wesley, 1994.
- [Wolf1984] Stephen Wolfram, *Universality and complexity in cellular automata*, Physica D, 10 :1-35, 1984.