

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

**UNIVERSITE MENTOURI DE CONSTANTINE
FACULTE DES SCIENCES DE L'INGENIEUR
DEPARTEMENT D'INFORMATIQUE**

N° d'ordre :.....
Série :.....

MEMOIRE

**Présenté pour l'obtention du diplôme de
Magister en Informatique**

**Calcul amorphe pour la classification des données
par GNG : application à la segmentation d'images
et à la bioinformatique**

Présenté par :

Chelloug Samia

Dirigé par :

Pr. Meshoul .S

Soutenu le : .. /.. /2006

Table des figures

Figure 1.1.a.: Réseau ordonné du quartz cristallisé.....	7
Figure 1.1.b.: Réseau désordonné du verre de slice	7
Figure 1.2.: Rayon de communication.....	8
Figure 1.3.: Principe de l’algorithme ‘Club’.....	9
Figure 1.4.a.: Simulation de réaction-diffusion sur une grille régulière.....	11
Figure 1.4.b.: Simulation de réaction-diffusion sur une grille irrégulière	11
Figure 1.5.a.: Média cup	13
Figure 1.5.b.: Badge actif.....	13
Figure 1.6.: Segmentation d’images par paintable computing	15
Figure 1.7.: Schéma d’un inverseur.....	17
Figure 1.8.: Fonction factoriel sous MitScheme	20
Figure 1.9.: Résultat de l’algorithme ‘club’.....	21
Figure 1.10.: Chargement des fichiers de simulation	22
Figure 1.11.: Génération d’une ligne avec GPL.....	23
Figure 1.12.: Déformation d’une organe.....	26
Figure 1.13.: Pliage d’un papier virtuel	27
Figure 1.14.: Génération des tubes par le langage de Weiss	29
Figure 1.15.a.: Décomposition d’un motif.....	30
Figure 1.15.b.: Réplication des particules.....	30
Figure 1.16.: Principe de la synthèse d’images	31
Figure 1.17.: Reconstruction d’images par calcul amorphe	32
Figure 1.18.a.: Cellules initiales.....	33
Figure 1.18.b.: Défaillance d’une cellule	33
Figure 1.18.c.: Auto réparation des cellules.....	33
Figure 1.19.: Principe de l’approche cognitive décentralisée d’ambiances.....	34
Figure 1.20.: Exemple d’un réseau de sonde.....	35
Figure 2.1.a.: Classification non-hiérarchique.....	40
Figure 2.1.b.: Classification hiérarchique	40
Figure 2.2.: Partitinement d’un espace par la méthode GAM.....	41
Figure 2.3.: Principe du K-means.....	43
Figure 2.4.: Modèle du neurone formel.....	45
Figure 2.6.: Schéma de SOM.....	47
Figure 2.7.: Résultat de GCS.....	49
Figure 2.8.: Evolution du GNG.....	52
Figure 2.9.: Réseau RBF	53
Figure 2.10.a.: Résultat du ENG.....	54
Figure 2.10.b.: GNG classique.....	54
Figure 2.11.a.: Exemple d’un environnement.....	55
Figure 2.11.b.: Planification par le GNG.....	55
Figure 2.12.: Images synthétisées par GNG	56

Figure 3.1.a. :	Image de cellules	65
Figure 3.1.b.:	Segmentation par seuillage	65
Figure 3.1.c.:	Segmentation par contour	66
Figure 3.1.d.:	Segmentation par région	66
Figure 3.2.:	Evolution du contour actif	68
Figure 3.3.:	Exemples de relations de voisinage (sites)	69
Figure 3.4.:	Segmentation par champ de Markov	69
Figure 3.5.:	Principe de la méthode de Melkemi	70
Figure 3.6.:	Etapas de segmentation par partitions de Voronoi généralisées	70
Figure 3.7.:	Résultats de segmentation par les partitions de Voronoi généralisées	70
Figure 3.8.:	Principe de la méthode Neut	72
Figure 3.9.:	Configuration d'un agent	73
Figure 3.10.:	Principe de la construction d'un pyramide par un SMA	74
Figure 3.11.:	Résultats de segmentation par l'approche des araignées	75
Figure 3.12.:	Principe de la méthode de Moussaoui	76
Figure 3.14. :	Fênetre de Netlogo	82
Figure 3.15.a.:	Initialisation de particules de calcul amorphe	83
Figure 3.15.b.:	Quelques connexions après dix itérations	83
Figure 3.15.c.:	Graphe final	83
Figure 3.15.c.:	Erreur d'apprentissage	83
Figure 3.16.:	Etapas de segmentation par région	84
Figure 3.17.:	Quelques images de test et leurs résultats de segmentation par contour.	84
Figure 4.1.:	Génome	93
Figure 4.2.a.:	Laboratoire sur puce	94
Figure 4.2.b.:	Cell-on-chip	94
Figure 4.3.:	Apport des puce à ADN	95
Figure 4.4.:	Fixation de sondes	97
Figure 4.4.b.:	Tête de fixation	97
Figure 4.5.:	Etapas de conception d'une puce à ADN	99
Figure 4.6.:	Matrice gène condition	99
Figure 4.7.a.:	Oubli de la préhybridation	100
Figure 4.7.b.:	Cristallisation des sels	100
Figure 4.8.a.:	Représentation des données	100
Figure 4.8.b.:	Résultat de normalisation	100
Figure 4.9.:	Fenêtre principale de Gene Cluster1	101
Figure 4.10.:	Résultat de la classification hiérarchique	102
Figure 4.11.:	Coorélation entre les gènes	103
Figure 4.12.:	Aperçu de la base PDR	108
Figure 4.13.:	Représentation des données des données scanner	108
Figure 4.14.:	Application de U-matrix	109
Figure 4.15.:	Visualisation des clusters après 1500 itérations	109

Sommaire

Introduction générale	1
Chapitre 1 :	
Contexte et état de l'art : Calcul amorphe	
1.1. Introduction	4
1.2. Systèmes complexes	4
1.3. C'est quoi amorphe ?	5
1.4. Caractéristiques d'un système de calcul amorphe	7
1.5. Métaphores du calcul amorphe	8
1.5.1. Hiérarchie du calcul amorphe	8
1.5.2. Propagation d'ondes	10
1.5.3. Mécanisme de réaction diffusion	11
1.6. Motivations du calcul amorphe	12
1.6.1. L'informatique pervasive	12
1.6.2. Wearable computing	13
1.6.3. Réseaux de sonde	14
1.6.3.1. Paintable computing	14
1.6.3.2. Pushing computing	15
1.6.3.3. Langage C@t	16
1.6.3.4. Limitations des réseaux de sonde	16
1.6.4. Calcul cellulaire	17
1.7. Systèmes multi agents réactifs et calcul amorphe	18
1.8. Médium de calcul amorphe	19
1.8.1. L'environnement HLSIM	19
1.8.1.1. MitScheme	20
1.9. Langages orientés calcul amorphe	22
1.9.1. Langage Growing Point Language 'GPL'	22
1.9.1.1. Exemple d'un programme GPL	23
1.9.1.2. Analyse d'un programme GPL	25
1.9.2. Langage Origami Shape Language 'OSL'	25
1.9.2.1. Avantages et inconvénients du langage OSL	27
1.9.3. Langage Relisient Serial Execution On amorphous Machines 'RSEAM'	27
1.9.4. Langage de Weiss	29
1.9.5. Intérêts de l'approche langage	29
1.10. Applications du calcul amorphe	30
1.10.1. Synthèse d'images	30
1.10.2. Reconstruction d'images	31
1.11. Approches basées calcul amorphe	32
1.12. Conclusion	35

Chapitre 2 :

Classification de données

2.1. Introduction	37
2.2. Processus de classification	37
2.3. Applications de la classification	38
2.4. Typologie des méthodes de classification	39
2.4.1. Classification locale versus globale	39
2.4.2. Classification statique versus dynamique	39
2.4.3. Classification floue versus exacte	39
2.4.4. Classification hiérarchique versus non-hiérarchique	40
2.4.5. Classification basée sur la densité	41
2.4.6. Classification basée sur un graphe	41
2.4.7. Classification basée sur un modèle	42
2.5. Méthode du K-means	42
2.5.1. K-means flou (FCM)	43
2.6. Approche évolutionnaire pour la classification	44
2.7. Réseaux de neurones	44
2.7.1. Apprentissage compétitif	45
2.7.2. Apprentissage incrémental	46
2.8. Cartes auto-organisatrices	46
2.8.1. Cartes auto-organisatrices probabilistes	48
2.9. Growing Cell Structure 'GCS'	48
2.10. Growing Neural Gas 'GNG'	50
2.10.1. Principe du GNG	50
2.10.2. Pseudo-code du GNG	50
2.10.3. Variantes du GNG	51
2.10.4. Applications du GNG	55
2.10.5. Avantages et inconvénients du GNG	58
2.11. Discussion	58
2.12. Analyse d'une méthode de classification	59
2.13. Conclusion	60

Chapitre 3 :

Calcul amorphe pour la segmentation d'images par GNG

3.1. Introduction	62
3.2. Segmentation d'images	62
3.2.1. Segmentation par seuillage	63
3.2.2. Segmentation par contour	64
3.2.3. Segmentation par région	65
3.2.4. Comparaison	66
3.3. Nouvelles méthodes de segmentation	67
3.3.1. Contours actifs	67
3.3.2. Champ de Markov	69

3.3.3. Méthodes basées sur les graphes	70
3.3.4. Systèmes multi agents	73
3.4. Segmentation par classification	75
3.5. Discussion	78
3.6. Segmentation d'images par calcul amorphe et le GNG	79
3.6.1. Pourquoi Netlogo ?	82
3.6.2. Résultats de segmentation	83
3.6.3. Calcul de la complexité	85
3.7. Conclusion	87

Chapitre 4 :

Calcul amorphe pour la classification des données issues des puces à ADN par GNG

4.1. Introduction	90
4.2. Terminologie en Biologie	90
4.3. Biopuces	93
4.3.1. Types des Biopuces	94
4.3.1.1. Laboratoires à ADN	94
4.3.1.2. Puces à cellule	94
4.3.1.3. Puces à ADN	94
4.4. Applications des puces à ADN	95
4.5. Conception d'une puce à ADN	96
4.5.1. Fixation des sondes	96
4.5.2. Marquage et hybridation	97
4.5.3. Acquisition des images	97
4.5.4. Traitement des données	99
4.5.5. Normalisation et filtrage	100
4.6. Analyse de données	101
4.7. Classification des données issues des puces à ADN	102
4.8. Classification des expressions de gènes par calcul amorphe et le GNG	105
4.8.1. Description de la base	107
4.8.2. Résultats de la classification	108
4.9. Conclusion	111
Conclusion générale	113
Références bibliographiques	115

Introduction générale

Construire des systèmes de calcul puissants, robustes et fiables, c'est tout l'art des systèmes distribués. Leur longue histoire est une suite de progrès donnant l'avantage tantôt à l'aspect hardware, tantôt aux méthodes de conception. Mais tous les systèmes distribués actuels sont très limités vis-à-vis des tâches complexes existantes. Toutefois, la question de la résolution d'une tâche complexe a fait, et fera encore, écouler beaucoup d'encre. Ce problème a divisé la communauté scientifique depuis plusieurs années. Cependant, les nombreuses potentialités offertes par la théorie d'émergence ont exacerbé la concurrence entre les concepteurs des systèmes distribués. Tout l'enjeu consiste à s'appuyer sur l'auto organisation. En présence de règles simples, les entités du système interagissent entre elles. Le résultat de l'interaction constitue un nouvel état caractérisé par des propriétés : connues ou inconnues au moment de la programmation. Les approches d'émergence sont maintenant couramment mises en œuvre par les systèmes multi agents réactifs. Ces systèmes aboutissent à la résolution d'un problème, souvent complexe, sur la base des interactions entre des entités de faible granularité et ayant une perception partielle de l'environnement.

D'autre part, la science informatique est en constante évolution grâce aux progrès récents des architectures matérielles. Ces progrès se sont accentués lorsqu'on a décidé de mettre en place des systèmes distribués radicalement différents de ceux que nous connaissons. Ces systèmes sont composés d'une masse importante d'éléments de calcul, distribués irrégulièrement, communiquant localement par diffusion et contenant le même programme. Logiquement, on peut se demander comment programmer ces éléments qui n'ont aucune identité ou information géométrique, qui doivent opérer en dépit d'une mémoire limitée et dont l'interconnexion peut se modifier au cours du temps. C'est là où peut intervenir le calcul amorphe. Ce paradigme s'intéresse à l'émergence d'un comportement cohérent à partir d'une multitude d'éléments désordonnés et dont l'abstraction est la clé du succès. Evidemment, le calcul amorphe permet de promouvoir une nouvelle vie informatique. Mais, il faut encore simuler nos programmes avant que ce rêve devienne une réalité. Dans ce contexte, l'organisation des particules de calcul amorphe selon une hiérarchie et la génération de motifs 2D sont deux idées déjà maîtrisées en matière du calcul amorphe.

Par ailleurs, les développements dans d'autres domaines nous obligent à manipuler, en quantité, des informations de plus en plus diverses. C'est pourquoi, la classification de données a envisagé de nouvelles solutions pour regrouper des objets similaires selon un critère de similarité. Si les méthodes non-hiérarchiques ont beaucoup servi à la classification de données, l'avantage des cartes auto organisatrices (SOMs) résulte de l'apprentissage incrémental et compétitif. Hormis, SOM qui fixe le nombre de neurones apprenant la topologie d'un espace d'entrée ainsi que la structure de la carte,

l'initialisation de deux neurones par l'algorithme Growing Neural Gas (GNG) suffit quant à elle pour la classification d'une masse de données en procédant par un cycle évaluation-insertion. Bien sûr, le GNG repose sur l'auto organisation.

En imagerie, la segmentation s'implante dans tout processus d'analyse d'images. Elle conduit à extraire les objets pertinents en s'intéressant aux contours et/ou régions. Les méthodes de segmentation sont nombreuses. Chacune propose des idées pour atteindre une bonne formulation, un temps d'exécution raisonnable et/ou une bonne qualité de segmentation. Pour exploiter les mécanismes de classification, la segmentation d'images s'est tournée vers la classification des pixels. L'objectif est d'accélérer le processus de segmentation.

En plus, il est dorénavant possible d'analyser simultanément des centaines, voir des milliers de gènes. En fait, les puces à ADN permettent de mener des diagnostics efficaces. Longtemps, l'étude des gènes permet d'effectuer une analyse à l'échelle moléculaire. Dès 1990, on a trouvé avec les puces à ADN, un moyen pour mesurer le transcriptome d'un échantillon de gènes. Mais désormais, le coût d'une puce à ADN est important, les expressions de gènes présentent quelques caractéristiques complexes et les données obtenues sont entachées de bruit. Cependant, le développement d'algorithmes d'analyse de données issues des puces à ADN est un thème d'actualité.

Si le calcul amorphe a une aussi grande importance, il peut être légitime de se demander sur son exploitation pour la résolution des problèmes complexes. Ce mémoire est une volonté pour envisager l'utilisation d'un système de calcul amorphe pour la classification. Il s'agit de la classification des particules de calcul amorphe pour la segmentation d'images ainsi que la classification des données issues des puces à ADN par des particules de calcul amorphe. Ce mémoire comporte quatre chapitres :

- ❖ Pour cerner le calcul amorphe, un état de l'art qui porte sur le calcul amorphe, ses motivations et les langages ainsi que les applications développées est présenté dans le chapitre 1.
- ❖ Le chapitre 2 est consacré à l'explication de la classification en mettant l'accent sur l'algorithme du GNG, ses particularités, ses variantes, ses applications, les avantages et les inconvénients d'une telle technique.
- ❖ La littérature sur la segmentation d'images est détaillée dans la partie 1 du chapitre 3. l'autre partie du même chapitre explicite la segmentation en adoptant le calcul amorphe et l'algorithme du GNG. Dans ce cas, les résultats obtenus sont démontrés et discutés.
- ❖ Dans la partie 1 du chapitre 4, il s'agit d'introduire les puces à ADN et les méthodes de classification qu'elles leur sont associées. Cependant, la partie 2 du chapitre 4 est une application de l'approche proposée pour la classification des données issues des puces à ADN par calcul amorphe.

Notons que ce travail a été réalisé en collaboration avec CMEP – PROGRAMME TASSILI sous le projet 05 MDU 642.

1.1. Introduction

L'évolution sans cesse croissante des besoins en puissance de calcul a fait naître plusieurs projets via les systèmes distribués. De même, les systèmes multi agents sont un outil promoteur dans le domaine de systèmes distribués qui s'écarte de la théorie analytique. En effet, l'ambition de systèmes multi agents est de montrer qu'un comportement intelligent est une caractéristique d'éléments distribués et interagissant entre eux. Malgré que les systèmes multi agents ont fait un grand nombre d'émules, les pionniers de l'informatique envisageaient toujours la conception d'un système plus puissant. Dans cette optique, le calcul amorphe a permis de prolonger les efforts précédents mais sa mise en œuvre pose un certain nombre de défis. Il s'agit d'émerger un comportement cohérent à partir de la coopération d'un très grand nombre d'éléments de calcul de faible capacité, positionnés irrégulièrement, communiquant localement par diffusion et dont l'abstraction de l'environnement est une heuristique importante. L'histoire du calcul amorphe a commencé vers les années 1997 avec des axes de recherche multiples. En fait, le challenge est l'auto organisation ou l'auto assemblage d'éléments de calcul pouvant être des processeurs, des robots miniaturisés ou des cellules vivantes. A l'aube du calcul amorphe, plusieurs processus biologiques, physiques et chimiques ont reconnu un regain d'intérêt. L'organisation d'un ensemble d'éléments en une hiérarchie, la simulation de la propagation d'ondes et celle du processus de réaction-diffusion sur une grille irrégulière ont été les premiers problèmes simulés par calcul amorphe. En étudiant en détail le calcul amorphe, on découvre plusieurs langages qui constituent un cadre rigoureux à son utilisation. A l'inverse du langage Growing Point Language (GPL), le langage Origami Shape Language (OSL) s'est orienté vers la génération d'un motif 2D par la programmation d'un papier virtuel de particules de calcul amorphe. Bien des années après, l'impact du langage Relisient Serial Execution On Amorphous Machines (RSEAM) est déjà considérable. En particulier, le traducteur associé offre la possibilité de la conversion d'un programme séquentiel en terme d'interactions entre des particules de calcul amorphe. En parallèle, plusieurs approches destinées à la robotique, la reconstruction d'images, la synthèse d'images et aux réseaux de sonde sont fondées sur l'utilisation du calcul amorphe en procurant les avantages d'un tel système. Mais avant de se pencher dans ce domaine, nous allons tout d'abord se placer dans un contexte général en expliquant le terme 'amorphe'. Le reste de ce chapitre est consacré à expliquer le quoi et le comment du calcul amorphe en présentant de manière chronologique une synthèse des travaux entrepris ainsi qu'une réflexion sur la différence entre calcul amorphe et systèmes multi agents réactifs.

1.2. Systèmes complexes

Beaucoup de systèmes ne peuvent être des îlots isolés au sein d'un environnement. Ainsi, la concentration actuelle tend à étudier ces systèmes dont le comportement dépasse les potentialités de leurs parties. On les qualifie de systèmes complexes. Par définition, un système complexe est tout système constitué d'entités indépendantes dits agents dont l'interconnexion joue un rôle important pour l'émergence d'une tâche. En

plus, la prédiction du fonctionnement d'un système complexe est assez difficile [1,2]. Les principales caractéristiques que l'on entend d'un système complexe sont [1, 3, 4]:

L'indéterminisme : en général, les systèmes complexes ont différents types de composants et diverses connexions.

La non-linéarité et le chaos : un système complexe fait preuve d'un entrelacement de rétroactions positives et négatives.

L'auto organisation : c'est le processus qui réduit l'entropie d'un système sans qu'il y ait une intervention extérieure.

D'ailleurs, la recherche en matière de systèmes complexes a été marquée par la résolution d'un problème par émergence. L'accent devrait être mis sur l'apparence de nouvelles caractéristiques globales au niveau macro par le biais d'interactions à un niveau micro [1,3]. Dans un souci de juger une approche basée sur l'émergence, deux voies sont possibles : contrôler un système afin d'aboutir à un objectif préalable ou un observateur externe est étonné du résultat non programmé auparavant. Si nous souhaitons concevoir des systèmes complexes, plusieurs théories ont été développées : les systèmes complexes adaptatifs, les automates cellulaires, les approches évolutionnaires, ... Cependant, les systèmes complexes actuels ont soulevé de nouveaux problèmes de conception : les entités du système sont placées irrégulièrement, n'ont aucune identité, l'interconnexion peut varier au cours du temps, ... En même temps, des recherches s'emploient pour exploiter ces systèmes innovants en adoptant un paradigme de calcul approprié. Dans ce sens, le calcul amorphe pourra répondre aux exigences de ces systèmes en s'intéressant à la conception des méthodes génériques.

1.3. C'est quoi amorphe ?

La littérature contient plusieurs définitions connexes, qui ont permis l'émergence de nouveaux paradigmes dans d'autres domaines scientifiques. Avec l'évolution technologique rapide des systèmes informatiques, l'utilisation du calcul amorphe, qui peut sembler quelque peu étrange, est devenue possible. Cependant, l'association de ces deux termes que nous allons expliciter n'est pas innocente. Il existait avant son essor 'la matière amorphe' qui est l'objet de recherche des équipes de physique du solide. Certes, l'atome est l'élément de base de chaque matière. Lorsque plusieurs atomes interagissent attractivement, ils commencent généralement à former des molécules qui serrent à constituer une matière. Habituellement, trois états de la matière sont à distinguer [5,6]:

Etat gazeux: c'est un état de faible densité, où les atomes (molécules) sont presque indépendants les uns des autres. Les distances entre les atomes sont très grandes et leurs interactions se font par l'intermédiaire de chocs.

Etat liquide: dans ce cas, les atomes, ions ou molécules sont en contact les uns avec les autres mais il n'existe pas d'ordre à grande ou longue distance.

Etat solide: les atomes (molécules) s'ordonnent en réseaux cristallins. A l'échelle microscopique, un milieu cristallin est caractérisé par la répétition périodique tri-dimensionnelle d'un motif toujours identique constitué d'atomes, ions ou molécules.

En revanche, les physiciens et chimistes ont déjà constaté que cette classification n'est pas suffisante pour désigner tous les objets qui existent dans la nature [7]. A court terme, plusieurs travaux ont ressorti deux grandes catégories de matériaux tenant compte de l'arrangement des particules.

- Les matériaux cristallins où il y a une structure bien ordonnée de molécules, atomes ou ions.
- Les matériaux amorphes caractérisés par un ordre à courte distance et un désordre à longue distance. Dans ce contexte, des exemples représentatifs ont été cités dans la référence [8]. En effet, les métaux, une partie des céramiques, les polymères et les minéraux sont tous des matériaux cristallins dont les atomes sont disposés régulièrement selon un réseau tridimensionnel. A l'opposé, les verres, un certain nombre de polymères et des éléments chimiques font parti de la classe des matériaux amorphes. Afin de mieux appréhender la notion de l'amorphe, le tableau suivant se focalise sur une comparaison entre l'état cristallin et amorphe.

Cristallin	Amorphe
<ul style="list-style-type: none"> •Arrangement selon une grille fixe et régulière. •Etat cristallin est stable. •Transformation cristalline vers amorphe est possible. •Cristallin devient liquide à une température spécifique. •Facile à étudier. 	<ul style="list-style-type: none"> •Arrangement selon une grille irrégulière. •Moins stable par rapport à l'état cristallin. •Transformation amorphe vers cristallin n'est pas toujours possible. •Le changement est graduel dans une zone entre le solide et le liquide. •Structure difficile à étudier et inaccessible par les rayons X et diffraction.

Tableau 1.1: Comparaison entre état cristallin et état amorphe [7,9].

Par ailleurs, une lecture de la thèse [9] permet de mettre en évidence différentes catégories de matériaux amorphes. Dans ce sens, le solide amorphe est muni d'une très grande fragilité aux chocs due au fait qu'il n'a pas d'ordre à grande distance. L'exemple typique est le verre. Sa transparence résulte de sa nature céramique. Il n'a pas d'électrons qui absorbent ou émettent de l'énergie lumineuse. A l'opposé, le silicium amorphe est caractérisé par une forte absorption de la lumière et peu de dégradation par les rayons X. Il a été découvert au début de l'année 1960 et il est fabriqué pour obtenir des cellules solaires avec excellente absorption du spectre solaire, des écrans plats et aussi pour des applications de xérogaphie qui incluent les photocopieuses et les imprimantes lasers. Enfin, on trouve les polymères amorphes qui, à haute température, ce sont des liquides viscoélastiques et si on les refroidit, ils deviennent des verres.

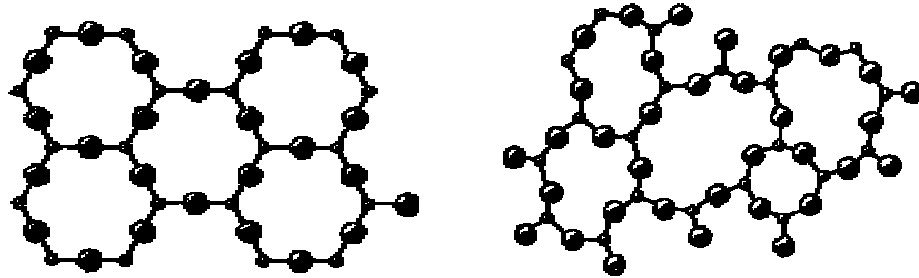


Figure 1.1.a.: Réseau ordonné du quartz cristallisé [10]. Figure 1.1.b.: Réseau désordonné du verre de silice [10].

1.4. Caractéristiques d'un système de calcul amorphe

L'émergence de nouvelles classes de l'informatique est due aux exigences qu'impose le monde industriel. Depuis longtemps, le développement de machines exploitant le parallélisme en fédéreront les algorithmes, techniques et outils mis en œuvre a pris une place importante. Aujourd'hui, nul n'ignore un des paradigmes baptisé 'calcul amorphe', qui est un projet de recherche qu'il faudra mener à maturité. Les premiers précepteurs de ce paradigme furent Gerry Sussman, Harold Abelson, Tom Knight, Jr. Au fil du calcul amorphe, l'identification des principes et langages pour observer, organiser et exploiter le comportement d'une multitude d'éléments de calcul constitue l'objectif central. Cependant, ces éléments respectent la conjonction de diverses contraintes [11,12] synthétisées comme suit :

- Le nombre de particules est important (de l'ordre de 10^6).
- Les entités sont identiques, asynchrones et interagissent localement par diffusion dans un rayon r qui est plus grand que la taille d'une particule et plus petit que le nombre des entités de calcul amorphe.
- Chaque entité possède une mémoire modeste et des capacités de calcul limitées.
- Les particules ne reçoivent aucune information géométrique ou coordonnées.
- L'horloge de chaque particule est initialisée à un temps aléatoire, mais la vitesse des horloges est identique.
- Les entités n'ont pas d'identités.
- Ne sont pas fiables et peuvent tomber en panne.
- L'association de capteurs et actionneurs aux particules est possible afin qu'elles puissent restituer des informations et réagir à l'environnement.
- En général, les particules sont immobiles.

Dans son principe, l'idée du calcul amorphe est séduisante car elle permet de capitaliser son effort à expliquer l'émergence dans les systèmes complexes par une combinaison d'outils très simples et dès fois traditionnels. Par ailleurs, le défi est de pouvoir répondre aux deux questions suivantes mentionnées par Radhika Nagpal dans son article [13] :

- Comment peut-on faire émerger un comportement complexe à partir de la coopération d'un très grand nombre de parties interconnectées de manière locale, irrégulière et dont l'interconnexion peut se modifier au cours du temps ?

- Comment peut-on traduire un objectif global en terme d'interactions locales entre plusieurs entités ?

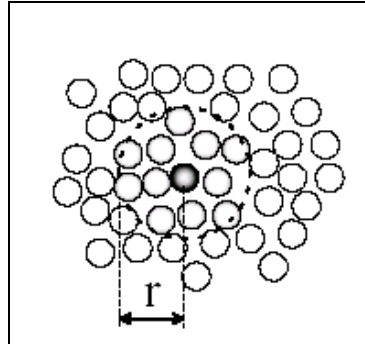


Figure 1.2: Le rayon de communication r [14].

1.5. Métaphores du calcul amorphe

Le calcul amorphe a été introduit pour éviter le recours à l'approche analytique ou encore pour émerger un comportement cohérent. Vu le nombre de contraintes que doivent respecter les particules associées, les premiers travaux du calcul amorphe se sont intéressés à l'expression de divers problèmes (chimiques, physiques, biologiques, ...) par un tel paradigme. Bien entendu, l'organisation de l'ensemble des particules désordonnées est une étape préliminaire pour résoudre un problème.

1.5.1. Hiérarchie de calcul amorphe

Les systèmes biologiques sont pourvus d'une grande capacité d'organisation. La finalité de ce processus est d'émerger des structures complexes en dépit d'entités de faible capacité, communiquant localement et opérants selon le même programme. De même, le résultat d'une hiérarchie de calcul amorphe est similaire à celui d'une hiérarchie classique. Néanmoins, le procédé de sa construction est différent. Lorsqu'on opte à organiser des particules de calcul amorphe par le biais d'une hiérarchie, plusieurs avantages sont offerts: la simplification de la programmation, la faisabilité d'analyse des algorithmes, l'augmentation de la robustesse et l'efficacité [15]. Dès le lancement du projet 'calcul amorphe', l'hiérarchie du calcul amorphe provoqua un grand intérêt à cause de son principe assez simple. On distingue deux algorithmes de base [16] :

L'algorithme club: approprié pour la construction d'une hiérarchie à un seul niveau. Après l'initialisation d'un ensemble de particules, l'algorithme 'club' se résume en la génération d'un nombre aléatoire r par chaque particule. Au cours de l'exécution, chaque particule décrémente r . Si ce dernier atteint zéro sans que la particule soit interrompue, elle se déclare chef et diffuse un message de recrutement aux autres particules. Dans le cas contraire, elle arrête le décomptage mais continue toujours à écouter les autres messages, lui permettant éventuellement d'appartenir à plusieurs groupes. Le problème posé est la possibilité du conflit qui nous oblige à réexécuter le même algorithme dans les régions de conflit. Cet algorithme est schématisé en figure

(1.3).

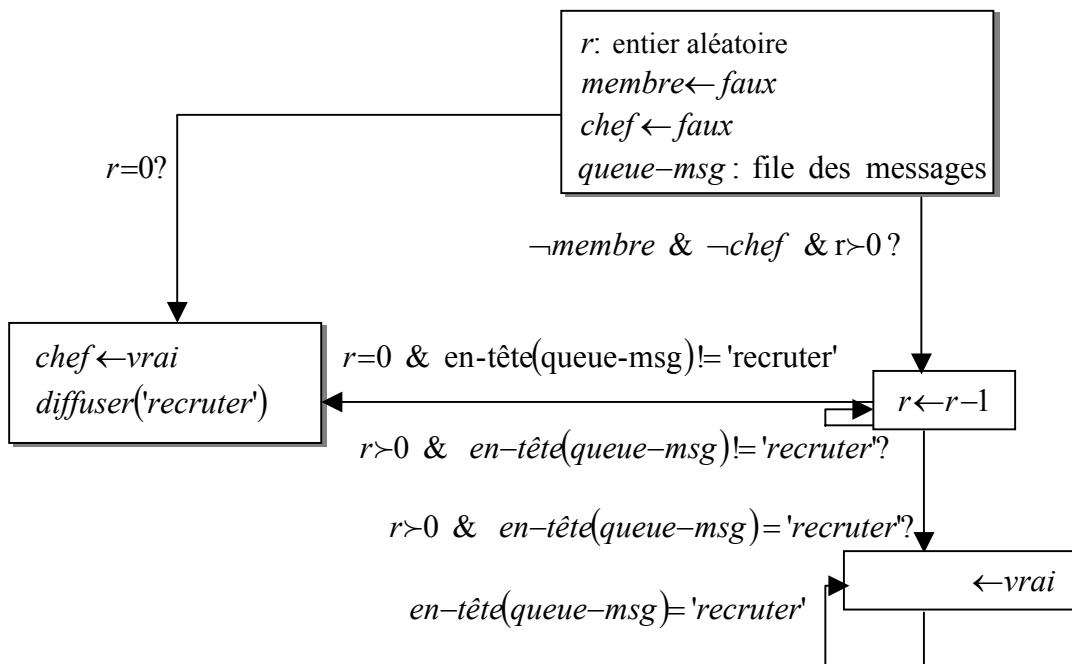


Figure 1.3.: Principe de l'algorithme 'Club'.

L'algorithme 'Tree-region' : cet algorithme inclut seulement d'autres paramètres pour la génération d'une hiérarchie contenant H niveaux. Après l'affectation d'une valeur à la variable H et la génération d'un nombre aléatoire r en fonction d'un temps t estimé, il s'agit d'appliquer le même algorithme précédent. En parallèle, chaque particule dont le nombre r atteint zéro se déclare racine et diffuse un message aux particules immédiates, qui doivent à leur tour recruter d'autres particules. Ainsi, une particule ne sera recrutée que si elle reçoit un message de recrutement de niveau inférieur.

R : Entier ; l'intervalle de la génération d'un nombre aléatoire.

r : Entier aléatoire.

t : Entier ; temps estimé pour créer une région.

H : Hauteur de la hiérarchie.

h' : Niveau actuel.

compétant : Booléen.

racine : Booléen.

$r = \text{Random}(R) \times t$

Proc () ;

Procédure Proc

Si $\neg \text{vide}(\text{queue-msg})$ **alors**

Si $\text{enfiler}(\text{msg-queue}) = \text{'recruter'}$ **alors**

Diffuser ('recruter', $h+1$)

$h' \leftarrow \text{msg.h}$

compétant \leftarrow faux

Attente ()

```

Sinon
  Si compétant=vrai alors
     $r \leftarrow r-1$ 
    Si  $r=0$  alors
      racine  $\leftarrow$  vrai
      Diffuser('recruter', 1)
      Attente ();
    Fin Si
  Sinon
    Proc();
  Fin Si
Fin Si
Fin Si

```

1.5.2. Propagation d'ondes

La propagation d'ondes a été introduite par les physiciens qui ont éprouvé le besoin de traiter des systèmes d'une grande complexité. Pour rappel, l'équation scalaire de propagation d'ondes a été dérivée à partir du principe de la conservation de la matière. Elle est de la forme [16]:

$$\frac{\partial^2 u}{\partial t^2} = C^2 \nabla^2 u \quad (1.1)$$

Où $u(x,y)$ est l'amplitude de l'onde au point (x,y) et C^2 la vitesse de propagation. En discrétisant la solution de l'équation (1.1) sur une grille régulière, les résultats obtenus sont très significatifs. Récemment, une nouvelle approche de résolution de l'équation (1.1) a été franchie par la mise au point du calcul amorphe. Le souci est de discrétiser la solution de divers problèmes physiques sur une grille irrégulière. C'est pourquoi Erick Rauch [17] a décrit un modèle; à travers duquel, plusieurs phénomènes ont renforcé le recours au calcul amorphe. Dans ce modèle, construit au tour de particules de calcul amorphe, des clusters sont définis afin de parvenir à une abstraction. En fait, l'échange du moment a été entrepris pour simuler la propagation d'ondes. Sachant que la quantité du moment échangé entre une particule i et une particule aléatoire j est fonction de la masse m et la vitesse v_i :

$$\Delta p_i = m \Delta v_i = m a_{ij} \Delta t = f(u_i - u_j) \Delta t = -\Delta p_j \quad (1.2)$$

La valeur de l'amplitude au point i est calculée indépendamment des autres particules :

$$\Delta q_i = \frac{p_i}{m} \Delta t \quad (1.3)$$

De cet exemple, nous constatons que l'organisation d'une tâche dans un système de calcul amorphe constitue un assouplissement par rapport aux approches analytiques s'intéressant à la résolution des équations différentielles.

1.5.3. Mécanisme de réaction-diffusion

L'utilisation du mécanisme de réaction-diffusion est un élément clé dans les travaux de recherche associés aux grands défis scientifiques, telles que la chimie, la biologie et la physique. Simuler ce phénomène passe nécessairement par la formulation des interactions par des équations de la forme [16] :

$$\frac{\partial u}{\partial t} = K \nabla^2 u + F(u) \quad (1.4)$$

Avec u un vecteur dénotant le changement de la concentration d'une certaine substance chimique en fonction du temps. Habituellement, la résolution de l'équation (1.4) tient compte d'une grille régulière. La principale raison est d'assurer le principe de la conservation de la matière. Le noyau du procédé de résolution consiste à faire l'approximation du terme $\nabla^2 u$ en fonction de la moyenne des valeurs voisines :

$$\nabla^2 u(i) = \sum_{i \in N_i} \frac{(u_{in}) - (u_i)}{N(i)} \quad (1.5)$$

Lorsque nous nous intéressons à un système de calcul amorphe, la taille du voisinage n'est pas constante. Là encore, l'abstraction se ressort comme une solution évidente. Il s'agit de discrétiser la résolution de l'équation (1.4) en considérant des paires de particules. En effet, la mise à jour de la valeur u du site i à l'étape $(n+1)$ est donnée par cette formule :

$$u_i^{n+1} = u_i^n + \Delta t \left(r_u (u_i^j - u_i^n) - u_i^n (u_i^n)^2 + f(1 - u_i^n) \right) \quad (1.6)$$

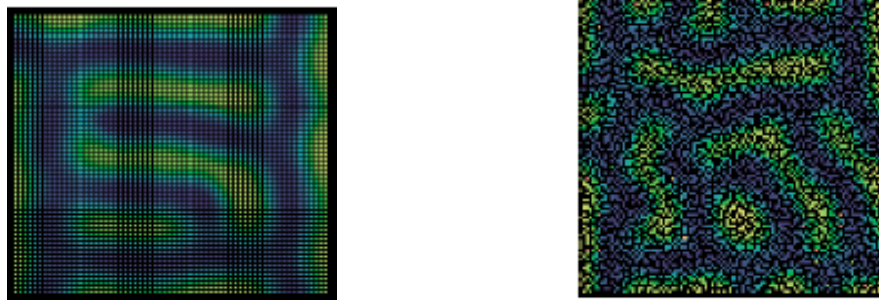


Figure 1.4.a.: Simulation de la réaction-diffusion sur une grille régulière. Figure 1.4.b. : Simulation de réaction-diffusion sur une grille irrégulière[16].

A partir des résultats expérimentaux, le calcul amorphe s'avère une approche prometteuse. Malgré qu'il est fondé sur le critère du désordre, sa convergence vers une solution a présenté un avantage par rapport aux systèmes connexes (automates cellulaires).

1.6. Motivations du calcul amorphe

Depuis l'an 2000, des systèmes liés à l'informatique omniprésente, au wearable computing et l'ingénierie cellulaire se sont succédés. L'objectif est d'expérimenter un comportement intelligent en fédéreront la notion du parallélisme. En fait, tous les avis convergent sur les avantages offerts par ces systèmes distribués[18] :

- Le coût de production de ces systèmes est réduit par rapport aux systèmes conventionnels.
- On peut mettre les parties constituant ces systèmes n'importe où.
- Ils peuvent traiter des environnements inaccessibles.
- Besoins de miniaturisation et de parallélisme.

Né de la constatation qu'on ne savait pas toujours résoudre des problèmes au sein de ces domaines, le calcul amorphe apparaît comme une unification et une solution intéressante pour programmer ces systèmes s'articulant sur la recherche de méthodes, outils et langages appropriés à des systèmes assez complexes. Etant une réponse efficace à la rude complexité de l'informatique omniprésente, au wearable computing et l'ingénierie cellulaire, on découvre du texte ci-après les caractéristiques de chacun de ces axes, les avancées qui y sont présentes et les limitations évoquées.

1.6.1 L'informatique omniprésente

On parle actuellement beaucoup de l'informatique et on en parlera encore plus sur l'informatique pervasive ces prochaines années, en donnant à ce domaine l'importance qu'il mérite. En effet, la promesse d'une telle technologie dépasse l'architecture client serveur. Elle harmonise plusieurs idées attrayantes dont le potentiel englobe la miniaturisation, la discrétisation et l'intégration des microprocesseurs dans les objets quotidiens. En plus, la prise en charge du contexte et l'interaction avec l'utilisateur semblent être à la puissance du calcul des microprocesseurs utilisés [19]. En 1998, Mark Weiser publia un article devenu célèbre, dans lequel il a introduit le concept d'informatique pervasive. De façon synthétique, l'informatique pervasive est une évolution marquante de l'informatique depuis l'an 2000. Elle consiste à permettre à chaque individu à être en contact permanent avec des milliers d'objets augmentés par des capacités de calcul : capteurs, actionneurs, ... D'un point de vue économique, ce bon en avant sera sans doute fructueux. Les progrès les plus visibles de l'informatique omniprésente sont liés au suivi et diagnostic à distance, le contrôle, l'automatisation, le réapprovisionnement automatique et l'optimisation du service client [19]. Avec les projets lancés en informatique omniprésente, les schémas d'utilisation des objets quotidiens ont été impérativement modifiés. Parmi ces projets :

Badge actif : C'est un projet qui a été cité dans plusieurs références. N.Davies [20] annonce qu'il a été développé en 1990 au sein du laboratoire Olivelly. Son utilisation est destinée à la détection des personnes dans les grandes entreprises pour faciliter le routage des appels téléphoniques.

Machine à laver: L'exigence d'avoir des environnements intelligents est certes la raison principale de l'introduction de l'informatique pervasive dans le secteur économique. Un autre exemple est celui d'une machine à laver qui lit les étiquettes des vêtements et récupère des informations concernant la taille, couleur, type et les instructions de lavage. Elle est très utile pour signaler à l'utilisateur des alertes en cas où il y a des vêtements qui ne doivent pas être lavés ensemble. En plus, elle peut récupérer des informations importantes sur les produits à laver [19].



Figure 1.5.a.: Mediacup [20].



Figure 1.5.b.: Badge actif [20].

Média cup: N.Davies [20] a présenté une autre réalité de systèmes pervasives concrétisée par une tasse équipée par un petit microcontrôleur, des capteurs et un système de communication sans fil. Elle est assez différente d'une tasse classique dans le sens où elle alerte son utilisateur dans le cas où la température interne serait très élevée.

Il est évident que l'informatique omniprésente est une innovation merveilleuse. Cependant, sa mise au point se heurte à plusieurs obstacles expliqués dans [19,20].

- Problème de reconfiguration dû au changement de l'environnement.
- Le fonctionnement en dépit des informations incomplètes.
- Gestion délicate à cause d'un très grand nombre de combinaisons matérielles et logicielles.
- Reconnaissance des tâches et intentions des utilisateurs n'est pas toujours une opération facile.

1.6.2. Wearable computing

Après l'apparition de l'informatique pervasive et ses débuts fracassants, le wearable computing est une déclinaison de cette technologie permettant une portabilité durant les opérations. Evidemment, le wearable computing est une solution motivante pour la réalité augmentée et lui partage les principaux attributs [21]: attirer l'attention de l'utilisateur, s'exécuter de manière continue et capter l'état actuel de l'utilisateur. Par ailleurs, le wearable computing fournit un accès constant aux informations et fait une médiation d'interaction avec l'environnement utilisateur. Par définition, les ordinateurs de cette génération sont une partie des vêtements de l'utilisateur. Ils ont peu de

ressources et une énergie limitée [22]. Depuis peu, les résultats de cet axe de recherche sont présents. Mithill est un projet au Media lab. La plate forme consiste en un ensemble de capteurs, un réseau et des composants de calcul. Les interfaces sont construites avec LinuxOS et la communication est sans fil. A cet égard, il est devenu possible de contrôler des paramètres vitaux associés à un patient (niveau de glucose, pression du sang, température,...). Ces informations seront ensuite transmises à une base dans laquelle les médecins stockent le fichier du patient et sont capables de soumettre des instructions aux patients. En cas d'un problème, une ambulance est envoyée à l'emplacement du malade qui est détecté par un système GPS. A l'inverse d'une consultation traditionnelle, le coût de l'hospitalisation est réduit et le taux de mort pour les maladies chroniques pourra se diminuer [23].

1.6.3 Réseaux de sonde

Le domaine des réseaux informatiques promet sans cesse de nouvelles et insoupçonnées possibilités nées des besoins d'organiser, restituer et traiter l'information pour créer un chemin entre plusieurs entités. A partir du moment où les capteurs ont été mis au point, une catégorie récente des réseaux s'ajoute. Le principe des réseaux de sonde est fondé sur le déploiement d'un grand nombre de capteurs de manière ad hoc. Plus particulièrement, ces réseaux sont connus dans des champs d'applications préalables comme la météorologie, le contrôle de qualité des produits et la restitution des données hospitalières. Le rôle des capteurs est de translater des quantités physiques en une représentation intéressante en terme de qualité, taille et coût en respectant quelques caractéristiques. Bien souvent, [24] les capteurs considérés ont une taille physique petite, consomment peu d'énergie et peuvent être conçus différemment pour capter simultanément la même information. Sur un plan comparatif, les réseaux de sonde présentent des attributs importants par rapport aux réseaux mobiles ad hoc (MANET). Le nombre de capteurs utilisés est d'autant plus grand que dans le cas des MANET, les capteurs peuvent tomber en panne, l'énergie disponible et les capacités de calcul sont limitées [25]. En dehors de la complexité de gestion associée aux capteurs, la robustesse et l'auto reconfiguration sont deux services offerts. De façon synthétique, A.Pilpret [26] a expliqué que l'expansion des réseaux de sonde est due à plusieurs considérations :

- Sont utiles pour des espaces où aucune infrastructure adéquate n'est disponible.
- Pas de supervision humaine.
- Capacité d'adaptation aux évènements.
- Optimisation de ressources.
- Capacité d'auto organisation par des méthodes de clustering.

1.6.3.1. Paintable computing

L'idée du paintable computing conçue par Buthera est étroitement liée aux réseaux de sonde. Grâce à la miniaturisation, plusieurs éléments de calcul peuvent être mélangés et positionnés sur une surface. L'exécution d'une tâche est confiée aux principes d'auto organisation. Le modèle de référence élaboré s'articule autour d'une IC (2mm x 2mm), RAM (50Kb), un bloque d'énergie, une horloge 50MHZ et des transmetteurs de communication [27]. Par défaut, les éléments utilisés n'ont aucune connaissance de

leurs positions et sont tous identiques. Ils opèrent localement sans se préoccuper de leurs interconnexions. En outre, l'architecture détaillée du paintable est donnée dans [28]. Elle bénéficie d'un appui important au sein des modèles d'auto assemblage. Dans cette architecture, tout le code est incorporé dans la particule. Il contient des fonctions pouvant être activées selon des conditions prédéfinies. Au cœur du paintable, la notion de fragments de processus (pfragments) est cruciale afin d'organiser le software en modules. En effet, les programmes résidents dans l'espace RAM qui contient une partie I/O et la page home. Cette dernière est l'emplacement où les programmes peuvent lire et écrire des données. Au moment de l'exécution, les pfragments peuvent migrer; chercher des particules et lire des données. En plus, le modèle de référence du paintable est riche par un ensemble de pfragments détaillés dans [28] et servant comme opérations de base à d'autres applications.

Module gradient: si l'objectif de chaque particule est d'estimer la distance entre une particule et un point de référence, ce fragment s'installe dans l'ensemble des particules à travers des portails d'entrées/sorties et se propage comme un virus en générant une copie de lui-même. Ce processus inclut la consultation de la page home, l'attribution d'une valeur initiale, la propagation et enfin l'incrément de la valeur et son adaptation.

Module de tessellation: permet de regrouper des particules en régions de Voronoi. Il contient à son tour un autre pfragment pour chercher les particules qui se retrouvent à des distances équitables.

Dans le cadre d'un certain nombre d'applications, Buthera a démontré l'utilité de son modèle pour plusieurs applications. Parmi ses résultats, la segmentation d'images est réalisée par un ensemble de capteurs qui peuvent capter l'intensité locale, encodent la valeur des pixels et inscrivent cette information dans la page home. Globalement, la segmentation est un processus de classification.

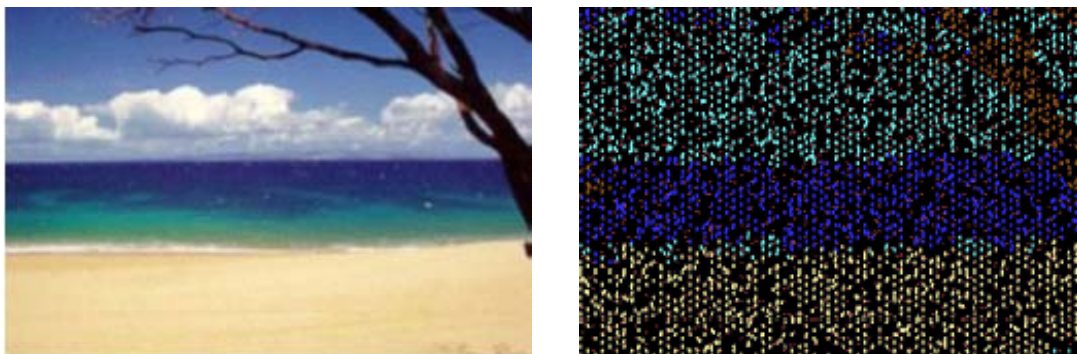


Figure1.6.: Segmentation d'images par paintable computing[28].

1.6.3.2. Pushing computing

Présenté comme étant une instance du paintable [29], le pushing computing est issu des réseaux de sonde dont le but est de diviser un réseau de sonde. Dans son apparence hardware, la plate forme contient environ 100 pushing chacun muni de deux pinces de longueur différentes qui soient positionnées sur un plan. De plus, il y a des modules

modifiables de prétraitement et de communications, des capteurs et des sources d'énergie [30]. La communication entre particules se fait à l'aide de l'infrarouge et le système d'exploitation utilisé et celui de Buthera [31]. Ce dernier est un modèle à base d'évènements. Il est divisé en une collection de composants logiciels dont la fonction est de déclarer les commandes utilisées ainsi que les évènements à signaler [30]. Cette technologie est concrétisée par un ensemble de prototypes tel que proto pushing où chaque pushing contient trois pinces (énergie, communication, ground). Il consiste en un LED (light emitter diode) et une résistance en série. Dans ce cas, il était question de démontrer les capacités de communications locaux entre particules essentiellement influées par la luminosité de chaque LED.

1.6.3.3. Langage C@t

Les réseaux de sonde sont connus pour être des systèmes plus répandus que les architectures distribuées et doivent être déployés à la demande afin de couvrir une grande palette de situations physiques. Leur fonctionnement accueillera un nombre extrêmement grand de parties dont les interactions sont très complexes et les ressources disponibles sont minimales. Face à cette situation, l'abstraction est une brique fondamentale pour répondre à la complexité pressante dans cet acheminement qui est en plein essor. Le principe du langage C@t qui est apparu pour la première fois dans la thèse [31] est de préserver la programmation séquentielle par hybridation de la programmation collective et déclarative. Tout simplement, un programme C@t est structuré comme suit :

Déclarations des unités Déclarations des clusters Spécifications Variables et fonctions
--

Sa syntaxe est similaire à celle du langage shceme et son compilateur a pour mission de traduire les programmes ainsi écrits en code C afin de bénéficier de quelques routines C. Voici quelques commandes requises quant à l'utilisation du langage C@t:

(declare-device device (attribut val) (attribut val)...)
(declare (cluster (device count) (device count)...))

La première commande définit une nouvelle unité appelée device possédant des attributs, chacun est muni d'une valeur initiale. La deuxième commande sert à déclarer un cluster composé d'un nombre 'count' d'unités.

1.6.3.2. Limitations des réseaux de sonde

Vu les contraintes des réseaux de sonde et leurs applications critiques, un assez grand nombre de paradigmes et langages est en cours de développement. L'objectif est la livraison des données à une destination en temps réel, la restitution des données par des capteurs, la production de la bonne réponse et l'envoi de messages dans des régions.

A cela s'ajoute le fonctionnement en dépit de : ressources (mémoire, temps CPU, énergie) minimales et même en l'occurrence d'erreurs [32]. Si certains langages existent, la mise en oeuvre des réseaux de sonde et leur réalisation opérationnelle pour résoudre une tâche spécifique demande encore des développements assez longs. A l'heure actuelle, le centre de recherche de Berkely est le plus intéressé à généraliser les méthodes existantes en présentant toute sorte d'abstraction pour l'exploitation industrielle des réseaux de sonde.

1.6.4. Calcul cellulaire

En Biologie moléculaire, la simulation des interactions des cellules est une préoccupation majeure. Concrètement, cette approche sert surtout pour la synthèse des produits pharmaceutiques et la fabrication moléculaire des biomatériaux. Au-delà des capacités limitées des cellules, celles-ci communiquent localement par diffusion et sont toutes gouvernées par le même programme ADN. D'ailleurs, les cellules biologiques représentent des entités de calcul amorphe. La recherche de méthodologies et langages appropriés à leur contrôle constitue un projet intitulé 'calcul cellulaire' qui est une instance du calcul amorphe s'intéressant aux cellules biologiques. Actuellement, la seule approche qui a été conçue et validée est celle de R.Weiss[33]. Sachant que la production de protéines est un processus important pour la survie des cellules, son explication passe par deux étapes : la transcription de la séquence ADN du gène en ARNm sous le contrôle d'un enzyme appelé ARN polymérase. A la suite, la translation utilise un enzyme complexe, ribosome et le ARNm produit pour obtenir une protéine [34]. Vu la complexité des mécanismes induits par les cellules, les biologistes ont formulé l'interaction entre les cellules selon des portes biologiques. L'exemple typique est celui d'inverseur biochimique [34]. De façon concise, si la protéine se joint au site répresseur du gène, le ARNm bloque la transcription et inversement. Revenons à la technique de R.Weiss, les étapes requises sont : l'établissement d'une librairie des circuits génétiques sous la forme de portes cellulaires, le choix de composants appropriés à une application, la simulation des interactions par des logiciels spécifiques, tel que Biospice et enfin la modification des composants jusqu'à aboutir au comportement voulu.

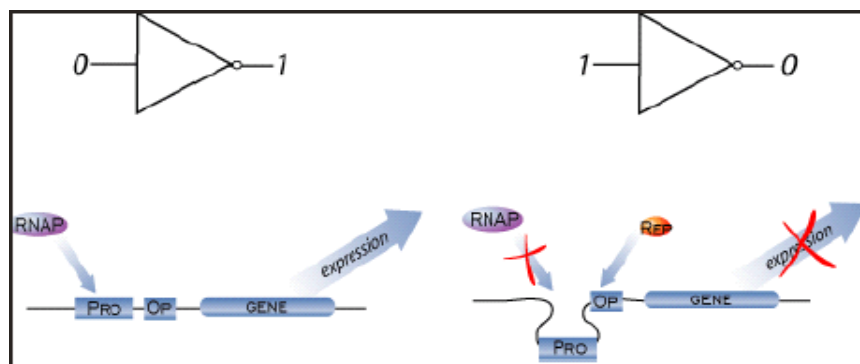


Figure 1.7. : Schéma d'un inverseur [33].

1.7. Systèmes multi agents réactifs et calcul amorphe

Comme certains domaines, l'informatique s'est confrontée à des problèmes complexes dont la résolution nécessite un degré d'intelligence. Dans un nouveau panorama de l'intelligence artificielle distribuée, les systèmes multi agents (SMA) sont constitués d'un certain nombre d'agents interagissant entre eux pour mener à bien une tâche considérable. A défaut de donner une définition précise de l'agent, la majorité d'auteurs ont dégagé les attributs de cette entité abstraite ou physique. La thèse [35] caractérise un agent par son rôle, spécialité, fonctionnalités, buts, croyances et ses différentes capacités décisionnelles, de raisonnement, de communication et d'apprentissage. En fonction de l'application envisagée, les études basées sur les SMA se proposent deux catégories d'agents: cognitifs et réactifs. C'est malgré qu'un SMA réactif soit composé d'une population massive d'agents où la notion d'émergence y est fortement inscrite, plusieurs indices servent à distinguer ce domaine du calcul amorphe :

SMA réactif	Calcul amorphe
<ul style="list-style-type: none"> • Représentation partielle de l'environnement. • Mémoire limitée. • En général, les agents réactifs sont mobiles pour acquérir de nouvelles connaissances et mettre à jour la représentation dont ils possèdent. • Problème d'identités et unicité du nom dans la phase de conception. Ce nom confère à l'agent des caractéristiques comportementales prédéfinies. • La taille d'un agent n'est pas important (il peut être petit ou grand) • Raisonnement réflexe permettant une exécution rapide des actions. • Basé sur des règles probabilistes [36]. • Hétérogènes. • Fonctionnement sur le postulat d'auto-organisation où les parties du système connaissent leurs orientations et ordre. 	<ul style="list-style-type: none"> • Aucune représentation de l'environnement. • Mémoire modeste. • Les particules de calcul amorphes utilisées sont statiques. L'organisation initiale est désordonnée impliquant un état non-stable. Au fur et à mesure, le système construit un état de stabilité. • Pas de problème d'identités. Les particules opèrent sans avoir des identités. • Les particules ont une taille très petite. Ce domaine est motivé par la nanotechnologie et le calcul cellulaire où des structures complexes émergent à l'échelle nanométrique. • Le changement d'état peut être moins rapide par rapport à un SMA réactif. • L'organisation de la tâche des particules par des langages est la préoccupation des chercheurs dans ce domaine [36]. • Identiques. • La problématique évoquée est l'auto-assemblage de plusieurs entités n'ayant pas une connaissance sur leurs orientations et configuration.

•Tire son inspiration de métaphores Biologiques (éco-systèmes) [36].	•Utilise des métaphores mathématiques, physiques, chimiques et biologiques [36].
--	--

Tableau 1.2. : Comparaison entre le réactif et l'amorphe.

Ces caractéristiques témoignent bien la nouveauté du calcul amorphe par rapport aux SMA réactifs. A cet égard, David Servat et Alexis Drogoul étaient les seuls qui ont fait l'air de penser à ces deux paradigmes dans leur article [36]. Ils ont proposé de combiner l'amorphe avec le réactif pour répondre à l'informatique pervasive. Cette nouvelle philosophie a été synthétisée par la question suivante :

« Comment peut-on avoir un comportement cohérent à partir d'un très grand nombre de parties hétérogènes, connectées de façon irrégulière, inconnue et variable dans le temps ? »

1.8. Médium de calcul amorphe

Lors du prototypage d'une architecture parallèle, le parallélisme peut être simulé sur un ordinateur classique ou obtenu directement sur un système réel. Pour le calcul amorphe, il est indispensable d'avoir plusieurs entités indépendantes munies de capteurs, actionneurs et interagissant localement dans un rayon R . Là encore deux approches existent. Du point de vue hardware, un seul prototype appelé 'Gunk On The Wall' est disponible. La thèse [16] décrit par le détail ce prototype; qui parmi ses caractéristiques figurent deux types de particules : PPP 'pinless processor particle' et EPP 'explicitly powered processor particle'. Dans le cas de EPP, la communication est sans fil et le couplage est électrostatique. Cependant, la section suivante, a pour but de montrer l'essence d'un environnement de simulation dédié au calcul amorphe.

1.7.1. Environnement HLSIM 'hight level simulator'

Au-delà des systèmes informatiques actuels, la mise en œuvre d'un système de calcul amorphe effectif constituera sans doute un élément critique. Désormais, la simulation de nos programmes est une idée systématique pour observer le résultat escompté, rédiger et traduire les objectifs initiaux. HLSIM a été développé en MIT. Sa philosophie est étroitement liée au calcul amorphe afin d'emprunter un calcul local avec un très grand nombre de processeurs dont l'arrangement est désordonné. Malgré sa puissance, HLSIM n'a pas été utilisé pour développer des projets dans d'autres instituts et laboratoires. Il y a aussi peu d'exemples dans sa bibliothèque. Pour résumer, HLSIM permet de simuler le fonctionnement d'un nombre important de processeurs qui ne possèdent qu'un rayon de communication et des unités d'entrées/sorties (capteurs/actionneurs). Il propose une notion très intéressante de threads qui peuvent exister à un niveau élevé ou bas. La synchronisation employée est une synchronisation par événements. Les événements doivent être transformés à un scheduler global pour les ordonner. Cela justifie la transformation d'un programme Gunk au style CPS

‘continuation past style’, où les futurs évènements peuvent être passés comme étant des arguments [37]. Enfin, HLSIM n’est pas directement exécutable. Il nécessite un autre langage fonctionnel développé en MIT qui s’appelle MITScheme. Ce dernier fait parti de la classe des langages déclaratifs fournissant une abstraction de la manière avec laquelle on calcul un résultat.

1.7.1.1. MitScheme

Mitscheme est actuellement à sa version 7.7. Il peut être installé sous Linux ou Windows NT. Il est bâti au-dessus du langage Scheme et présente des caractéristiques importantes. Ses variables sont définies statiquement avec leurs liens correspondants, les types sont associés aux valeurs et non aux variables, l’extension d’un objet (y compris les procédures ou fonctions) est illimitée. Enfin, MitScheme permet la récursivité et les arguments sont passés par valeur. Le tableau suivant synthétisé à partir de [37] illustre quelques concepts de MitScheme :

Concept	Explication
<ul style="list-style-type: none"> • #t, #f • () • #(define x 28) • (operator operand operand...) • #b number, #o number, #h number • Lambda fomula expression1 expression2 • Make-graphics-device. 	<ul style="list-style-type: none"> • Valeurs Booléennes. • Dénote une liste. • Définition d’une variable. • Spécification d’une expreesion. • Nombre binaire, digital ou hexadécimal. • S’évalue à un appel de procédure. • Permet la création d’un objet graphique.

Tableau 1.3.: Quelques concepts de MitScheme.

```

Edwin: *scheme*

;You are in an interaction window of the Edwin editor.
;Type C-h for help.  C-h m will describe some commands.
;Package: (user)

(define (factorial n)
  (if (< n 2)
      1
      (* n (factorial (- n 1)))))
;Value: factorial

(factorial 10)
;Value: 3628800

---*-Edwin: *scheme* (REPL: listen)---*Bot-----
  
```

Figure1.8. : Fonction factoriel sous MitScheme.

En plus, MitScheme nécessite une configuration de quelques variables d'environnement : `mitscheme-library_path` et `path`. Lors de son démarrage, la saisie des expressions d'un programme est faisable à l'aide d'un programme 'Read Eval Loop' qui visualise un prompt. L'exemple visualisé en figure (1.8) montre l'exécution d'une procédure de calcul de factoriel sous MitScheme. Un des principaux attraits de HLSIM est qu'il permet une communication par diffusion entre un ensemble de processeurs. La commande 'broadcast' permet à un programme de continuer son exécution. Toutefois, le message peut être n'importe quel objet `scheme`. D'après [16], si le message arrive à sa destination, il sera accompagné d'une primitive `message-event`. Sinon, un message de détection de collision est envoyé. En pratique, rien n'empêche d'associer un délai pour transmettre un message par le biais de la commande 'message-delay'. A la différence de plusieurs langages, la simulation est considérée comme étant un objet dans HLSIM. Elle est manipulée en suivant l'ordre de ces instructions qui ont été rapportées dans [38].

(*define sim (make-sim/1 1000 .05)*): Construit une simulation de 1000 processeurs avec un rayon de 0.05.
(*simulation.display! sim #t*): Affichage de la fenêtre graphique.
(*simulation.load sim 'init "foo"*): Chargement du programme `gunk` et `init`.
(*simulation.run sim*) : Lance l'exécution.
(*make-io-device initialise action*) : Associe des capteurs et actionneurs à un processeur.

La construction de clubs de processeurs est assez intéressante. Le code suivant implémente cette idée. Il associe aux processeurs des identités aléatoires dans un très grand intervalle. A la fin, les leaders changent leurs couleurs d'origine (du rouge en blanc) [15]:

```
(define (club-event-loop)
  (select
    (global-timeout 'done)
    ((and still-competing leader-timeout)
     (event.clear! leader-timeout)
     (color-me "red")
     (set! Still-competing? #F)
     (broadcast id) ; a LEADER message
     (club-event-loop))
    (primitive-message-event
     => (lambda (message)
         (event.clear! primitive-message-event)
         (if (eq? message 'collision)
             (set! collisions (+ collisions 1))
             (begin
              (set! Still-competing? #F)
              (set! Clubs-I-belong-to
               (cons message clubs-I-belong-to))
              (color-me (length clubs-I-belong-to))))
         (club-event-loop))))))
```



Figure 1.9.: Résultat de l'algorithme Club.

L'interprétation de ce fragment de code est assez simple. Chaque processeur choisit un temps leader-timeout qui est strictement inférieur au temps global-timeout nécessaire à exécuter tout le programme. La quatrième ligne indique qu'un processeur avec un temps Leader-timeout qui s'est écoulé sans être interrompue se déclare leader. Il change sa couleur en rouge et essaye de recruter d'autres processeurs. La ligne suivante explique que l'événement associé au thread de ce processeur passe à l'état non signalé et ne retournera plus de valeurs. Dans le cas contraire, le processeur est un membre de groupe. Sa couleur est fonction du nombre de groupes auxquels il appartient. Il passera son temps à écouter les autres messages. Les étapes de l'exécution sont visualisées dans les fenêtres suivantes :

```

Edwin: *scheme*
;You are in an interaction window of the Edwin editor.
;Type C-h for help. C-h m will describe some commands.
;Package: (user)

(load "c:/gunk/hlsim/stable/loader.scm")
;Loading "gunk\hlsim\stable\loader.scm"
;Loading "sim.com" -- done
;Loading "cps3-runtime.com" -- done
;Loading "simrun2.com" -- done
;Loading "comms3.com" -- done
;Loading "agenda-skip.com" -- done
;Loading "pnm.com" -- done
;Loading "devices.com" -- done
;Loading "neighbours.com" -- done
-- done
;Value: pseudo-uniform-2d-points

Edwin: *scheme*
(load "c:/gunk/hlsim/stable/cps3.scm")
;Loading "gunk\hlsim\stable\cps3.scm" -- done
;Unspecified return value

(cps "club5")
Syntax file: "club5.scm" "club5.bin"
Read...
  Time taken: .01 (process time); .01 (real time)
Syntax...
  Time taken: 0. (process time); 0. (real time)
Transform...
  Time taken: .01 (process time); .01 (real time)
Optimize...
  Time taken: 0. (process time); 0. (real time)
Generate SCode...
  Time taken: 0. (process time); 0. (real time)
;Dumping "club5.bin" -- done
;Loading "club5.tmp" -- done
;Dumping "club5.bin" -- done
;Unspecified return value

--**Edwin: *scheme* (REPL: listen)---Bot-----
--**Edwin: *scheme* (REPL: listen [level: 2])---

```

figure1.10. : Chargement des fichiers de simulation.

1.8. Langages orientés calcul amorphe

Si nous appréhendons le calcul amorphe, nous constatons que la plupart des travaux en MIT se situent plus particulièrement dans la catégorie langage. Partant d'une population de particules de calcul amorphe, le programmeur n'a qu'à écrire une spécification globale. C'est au compilateur de transformer la spécification initiale en un ensemble d'interactions locales. Malgré quelques limitations, les idées derrière ces langages sont simples en même temps merveilleuses.

1.8.1. Langage Growing Point Language 'GPL'

GPL est essentiellement un engouement pour le calcul amorphe. Ce n'est pas un hasard que l'on peut le classer comme étant une technique du contrôle des systèmes complexes. Son contexte peut-être résumé comme suit [39]: 'comment peut-on apprendre à contrôler un comportement global si l'on a seulement accès à des manipulations locales?'. Son objectif original est similaire au contrôle d'une collection de cellules biologiques interagissant entre elles et opérant selon le même code ADN. Plus particulièrement, GPL est désigné à spécifier la construction de topologies interconnectées dans un système de calcul amorphe. Le choix de la génération de motifs

interconnectés découle de l'issue de plusieurs points. En premier, on peut facilement vérifier si l'on a atteint le but spécifié. De plus, GPL permet un niveau de compréhension des propriétés macroscopiques dérivées de la manipulation des paramètres microscopiques. Par ailleurs, la nécessité de propager des informations de façon non locale est une partie délicate dans cette application.

Sur un plan pratique, l'élément de base de GPL est le growing point qui réside à chaque instant dans une seule position. Il se propage en transférant son activité d'une particule à une autre dans son voisinage, selon un tropisme. Dans son site actif, il peut déposer un matériel ou cacher de la phéromone dont la concentration joue le rôle de tropisme. Daniel Coore, le chercheur qui a conçu GPL, a bien défini les concepts relatifs au GPL.

- Une phéromone est décrite par un nom et des valeurs indiquant sa concentration.
- Les growing points possèdent des informations statiques (nom, paramètres, matériaux, tropisme) et une partie dynamique sur la taille actuelle et le dernier résultat.
- Une opération de filtrage est réalisée pour pouvoir propager des growing points.

1.8.1.1. Exemple D'un programme GPL

Afin de clarifier GPL, l'exemple suivant extrait de [15] est une spécification pour générer une ligne dont la construction requiert deux growing points installés dans ses extrémités. Le premier est statique. Il a pour mission d'attirer le second qui soit dynamique.

```
(define-growing-point (A-to-B-segment)
  (material A-material)
  (size 0)
  (tropism (ortho+ B-pheromone))
  (actions
    (when ((sensing? B-material)
          (terminate))
      (default
       (propagate))))))
```

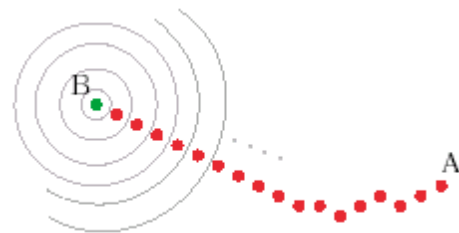


Figure 1.11.: Génération d'une ligne par GPL [17].

```
(define-growing-point (B-point)
  (material B-material)
  (size 0)
  (for-each-step
   (secrete 10 B-pheromone)))
(with-locations
 (A B)
 (at A (start-growing-point A-to-B-segment))
 (at B (start-growing-point B-point)))
```

Relativement à cet exemple, le premier growing point nommé (A-To-B-Segment) dépose dans son site actif un matériel (A-matériel) et se déplace selon le gradient de la quantité du phéromone (B-Phéromone). Par défaut, il se propage d'un site à un autre. S'il rencontre le point B il s'arrête. A l'opposé, le deuxième growing point (B-point) est statique. Dans ce sens, (B-point) n'a comme fonction que la génération d'une phéromone dont la quantité est symétrique autour de la source B.

Plus profondément, GPL est un langage qui modélise le parallélisme en maintenant un agenda des localisations qui exécutent actuellement des commandes. Si un growing point est activé dans un emplacement, il sera ajouté à l'agenda et demeurera jusqu'à l'achèvement de toutes ses commandes. Comme dans HLSIM, GPL repose sur le style CPS et la communication au bas niveau est inspirée des protocoles ALHOA et ISO. Les commandes GPL spécifiées seront ensuite traduites en primitives grâce à l'interpréteur Ecoli 'extensible calculus of local interactions' dans lequel chaque agent a une file de messages FIFO. Une liste de toutes les primitives utilisées est présentée dans [39]. Il est à signaler que lors de la translation, deux catégories de commandes sont distinguées. Celles qui dépendent de la coopération des particules voisines sont exécutées sur HLSIM (propagate et secrete). La deuxième catégorie est directement exécutable par l'interpréteur GPL qui assure que chaque commande sera terminée avant que la deuxième soit évaluée.

Notons aussi que la démarche de Ellie D'Hondt pour explorer GPL s'inscrit dans un contexte géométrique. Cette piste semble fondamentale puisqu'elle permet des calculs en graphisme, en systèmes d'informations géographiques et surtout en robotique. Parmi les problèmes traités, on y trouve la planification du mouvement d'un robot autonome. Le procédé de la planification doit prendre en compte le choix du plus court chemin optimisant le déplacement du robot. En premier lieu, elle a proposé un code HLSIM assez complexe dans lequel les processeurs détectent, par le biais de leurs capteurs, la présence d'obstacles. La source envoie un message à propager vers les autres processeurs. Un processeur qui reçoit un message s'oriente à renvoyer son identité, décrémenter le temps associé au message et enfin le propager. Dans le cas où le message atteint sa destination, chaque processeur diffuse un message avec la liste de ses parents, changeant sa couleur. Dans le prolongement de cette idée, Ellie D'Hondt a pu la réaliser en respectant les principes du langage GPL. En effet, le code ci-après [16] est très simple. Deux growing points sont initialisés. Le second est statique. Il diffuse une quantité de phéromone à chaque étape. Le rôle du premier est tout simplement de se propager jusqu'à y arriver à la destination. Dans son parcours, il se déplace en fonction du gradient de la concentration de la phéromone en évitant les sites contenant des obstacles.

```
(define-growing-point (robot-path)
  (material path-material)
  (size 0)
  (tropism (ortho+ goal-pheromone))
  (avoids obstacle-pheromone)
  (actions
    (when ((sensing? goal-material)
          (terminate))
      (default
        (propagate))))))

(define-growing-point (goal)
  (material goal-material)
  (size 0)
  (for-each-step
    (secrete 10 goal-pheromone)))
```

1.8.2. Analyse d'un programme GPL

Le langage GPL ouvre de nouvelles perspectives pour comprendre quant ou et comment des parties très simples émergent des structures complexes. On se référant à la thèse de Daniel Coore, nous constatons que GPL est accompagné d'un framework très puissant pour analyser théoriquement et expérimentalement les motifs obtenus. D'une part, cet outil permet d'organiser mathématiquement le fonctionnement des growing points. D'autre part, à partir de cette description mathématique, il devient possible de prédire le comportement d'un programme pour différentes conditions. Afin de mieux cerner cet outil, ces quelques définitions synthétisées de la référence [39] sont suffisantes pour dégager les principes importants.

Définition 1 : soit D un domaine, et d une métrique. Le voisinage d'un point p est :

$$N(p) = \{q \in D / d(p, q) \leq 1\} \quad (1.7)$$

Définition 2 : un growing point g est évalué dans p si p évalue le corps de g . Il se propage de p à q si $q \in N(p)$. g Restera immobile dans p , si p évalue le code de propagation mais aucun voisins ne satisfait les critères de propagation. La trajectoire de g est donnée par la définition récursive suivante:

$$p1(g, p) = \{q \in N(p) / g \text{ se propage de } p \text{ à } q\} \quad (1.8)$$

$$p(g, p) = \begin{cases} \{p\} \cup p1(g, p) / g \text{ est invoqué dans } p \\ \emptyset \quad \text{sinon} \end{cases} \quad (1.9)$$

Théorème : si g est invoqué dans p alors $p(g, p)$ est une composante connexe.

On dénombre également que ce framework est utile pour analyser les ressources du système. La taille des messages reçus, le nombre de phéromones et le nombre de matériaux sont des informations indispensables pour évaluer le temps de calcul requis.

1.8.2. Langage OSL 'Origami Shape Language'

Dés son apparition, les langages estampillés calcul amorphe étaient nombreux et représentaient un vrai décollage pour ce paradigme. Juste après la mise au point de GPL, Radhika Nagpal, une des pionniers du calcul amorphe, a annoncé dans sa thèse [40] intitulée 'Programming Self-Assembly: Constructing Global Shapes using Biologically-Inspired Local Interconnections and Origami Mathematics' le langage OSL. Sa réflexion repose sur l'explication de l'émergence de morphologies complexes par des interactions locales. Plus profondément, elle a suggéré l'idée du papier virtuelle composé d'agents pouvant s'auto assembler. Lors de son travail, elle s'est attachée à s'inspirer des mécanismes des cellules épithéliales ayant une capacité à se déformer et les mathématiques de Huzita permettant l'obtention de n'importe quelle forme par pliage en appliquant six axiomes.

Ainsi, le pliage du papier virtuel est simulé par le modèle des cellules épithéliales. Ce modèle a été conçu par Odell, Alberch et Burnside qui ont décrit la déformation d'une cellule à partir de l'ensemble de ses ressorts. La dynamique de chaque ressort est donnée par l'équation suivante :

$$m \frac{d^2 L}{dt^2} = -K(L - L_0) - \mu \frac{dL}{dt} \quad (1.10)$$

L : Longueur. K, μ : Constantes.

A partir de cette équation, le déplacement d'un point est donné par l'expression suivante :

$$m_i \frac{d^2 p_i}{dt^2} = -\sum_{j \in \text{neighbors}(i)} [h_{ij}(L_{ij} - L_{0ij}) + \mu_{ij} \frac{dL_{ij}}{dt}] e_{ij} \quad (1.11)$$

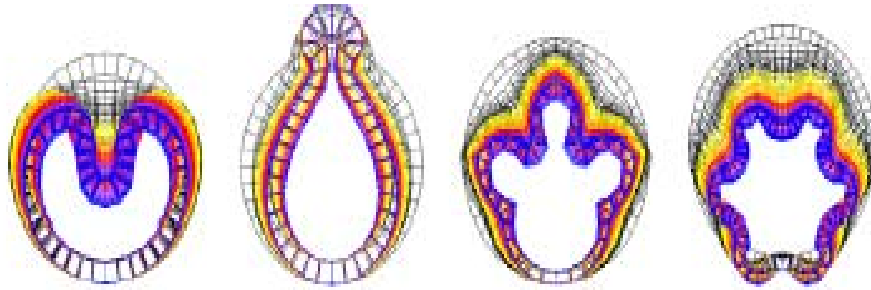


Figure1.12.: Déformation d'un organe [41].

Au niveau global, chaque cellule possède une surface apicale et basique. Elle peut plier virtuellement sa surface selon une orientation déterminée localement. Le programme des cellules est directement compilé à partir d'une description globale composée à l'aide des six axiomes suivants[41] :

1. Etant donné deux points p_1, p_2 , plier une ligne à travers p_1 et p_2 .
2. Etant donné deux points p_1, p_2 , plier p_1 sur p_2 .
3. Etant donné deux lignes l_1, l_2 , plier l_1 sur l_2 .
4. Etant donné un point p_1 et une ligne l_1 , plier l_1 sur elle-même à travers p_1 .
5. Etant donné deux points p_1, p_2 et une ligne l_1 , créer un pli qui place p_1 sur l_1 à travers p_2 .
6. Etant donné deux points p_1, p_2 et deux lignes l_1, l_2 , créer un pli qui place p_1 sur l_1 et p_2 sur l_2 .

Au niveau implémentation, ces axiomes seront traduits en un ensemble de concepts très simples. Par exemple, l'axiome 1 se traduit par la création d'un gradient g_1 par p_1 . Si g_1 arrive à p_2 , ce dernier crée un gradient g_2 et attendent tous deux un moment pour s'assurer que le gradient a traversé le pli. Enfin p_1 envoi un gradient g_{end} pour marquer la terminaison.

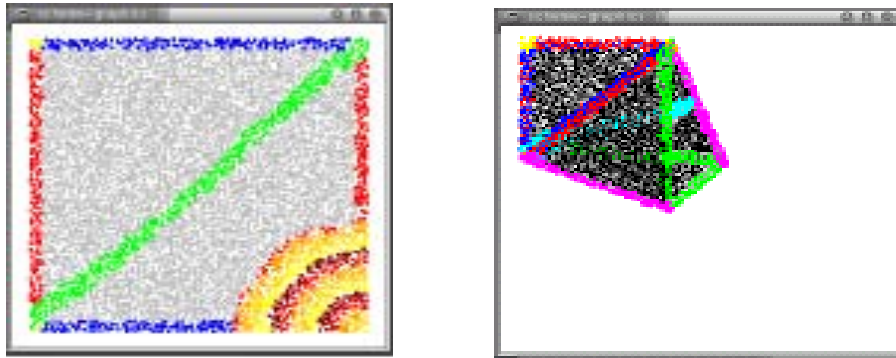


Figure1.13 : Pliage d'un papier virtuel[40].

La figure [14] considère une surface d'agents homogènes à l'exception des agents de bords qui possèdent des états initiaux. Afin d'atteindre la forme finale, la spécification considère quatre coins et quatre lignes. Les axiomes génèrent de nouvelles lignes à partir des points et lignes existants. Les lignes sont utiles pour délimiter des régions.

1.8.2. Avantages et inconvénients de OSL

Pour déterminer que OSL est un langage avantageux, Radhika Nagpal s'est motivée par le fait que la robustesse joue un rôle crucial lors de l'utilisation de son langage. Les justifications principales sont [40] :

- Le comportement d'un point ou ligne est le comportement moyen de quelques cellules.
- Les axiomes ne dépendent pas d'une estimation exacte de la distance.
- De nouveaux centre de contrôle sont ajoutés de façon constante afin d'éviter le contrôle hiérarchique ou centralisé.
- Une grande population de cellules.

Cependant, quelques critiques sont associées au langage OSL :

- L'élaboration du motif peut échouer s'il n'y a pas de cellules adéquates.
- Si plusieurs défaillances dans plusieurs points, on n'aura pas la structure souhaitée

Ce petit tour d'horizon permet de tracer une question qui constitue à la fois la faiblesse et la force de OSL :

« Quel est l'ensemble des axiomes appropriés pour émerger une structure connue à l'avance ? »

1.8.3. Langage Relisient Serial Execution on Amorphous Machines 'RSEAM'

RSEAM est un langage qui préconise la transformation d'un programme linéaire en un code série [18]. A l'inverse de l'approche traditionnelle, le code est dynamique. Il

peut s'auto répliquer. Afin d'assurer une simplicité sémantique des programmes, RSEAM est développé au-dessus d'un autre langage RGLL qui doit identifier la topologie induite et la relation au monde extérieur. A première vue, RGLL est semblable aux langages fonctionnels tels que Lisp et Scheme. Cette similarité est clairement justifiée par les besoins d'abstraction. RGLL supporte peu de types : integer, boolean, et list. Le typage est statique et la syntaxe des expressions est préfixée.

<ul style="list-style-type: none"> •Defconst constant-name constant value •Seq body... •Set biding-name exp •If, and, case, or, for, random, trandom-range, list, mem ? list 	<ul style="list-style-type: none"> •Affecte la valeur au contenu de la variable. •Exécute toutes les expressions dans le corps s'il n'est pas vide. •Localise le lien biding-name dans l'environnement actuel et met à jour sa valeur à exp. •Le sens classique.
--	--

Tableau 1.4 : Quelques concepts de RGLL

Ce mode de pensée englobe la notion de capsule. En fait, tout message diffusé ou reçu est encapsulé pour le rendre indivisible. Par analogie à la notion de classe, chaque capsule a un nom, une liste des noms des capsules parents et l'environnement contenant tous les liens ainsi que leurs valeurs correspondantes. Il y a aussi un ensemble de méthodes qui ne se sont rien qu'une association de noms et liste d'attributs. Le code d'une capsule peut avoir différents types: code exécuté si la capsule arrive à un nœud, code périodique, code exécuté si la capsule rencontre une autre instance d'elle-même ou un code de réparation qui sera diffusé aux autres capsules. Avec ce mécanisme, l'exécution du programme RGLL est réduite à la définition, appel et diffusion de capsules à l'aide des primitives suivantes :

Call method-name arguments: effectue l'appelle d'une méthode.

Broadcast capsule: c'est la commande appropriée pour diffuser une capsule vers un autre nœud.

Aussi, le langage RSEAM est conçu avec une sémantique de base appropriée au calcul amorphe. On regroupe des nœuds en cellules pour fournir un degré de résilience contre la défaillance d'un nœud. Sur la base de RGLL, des capsules mobiles passent leurs résultats aux autres. Il est à noter que la formation de cellules est obtenue à travers la construction d'une hiérarchie de calcul amorphe que l'on a présenté au début de ce chapitre. RSEAM s'appuie sur les fonctions suivantes:

RPC : pour localiser un code, débiter l'exécution, vérifier la complétude et retourner la réponse.

checkpoints : permettent de stocker un calcul et le récupérer en cas de défaillance.

code repositories : qui contiennent toutes les capsules présentes dans un programme.

De manière très précise, l'idée de RSEAM peut être résumée comme suit : 'un code linéaire est transféré en une série horizontale de capsules interagissant entre elles'

1.8.4. Langage de Weiss

Etend donné une multitude d'entités de calcul amorphe, la génération d'un motif par le langage de Weiss est très simple. De part son principe, ce langage permet la spécification d'un ensemble de règles indépendantes exécutées de façon asynchrone par les éléments du calcul amorphe. Par définition, l'état de chaque particule est représenté par des marqueurs. A chaque marqueur est associé un temps mesurant la durée de sa valeur. Lors de la réception d'un message et si la conjonction des marqueurs est satisfaite, la règle correspondante est activée [42]. Le résultat d'exécution est la diffusion d'un message indiquant la distance de diffusion ou la mise à jour des marqueurs. Par exemple, si le motif spécifié consiste en l'altération des tubes bleus et rouges, le programme correspondant permet l'initialisation de quelques marqueurs pour la propagation des messages make-seg-red et make-seg-blue.

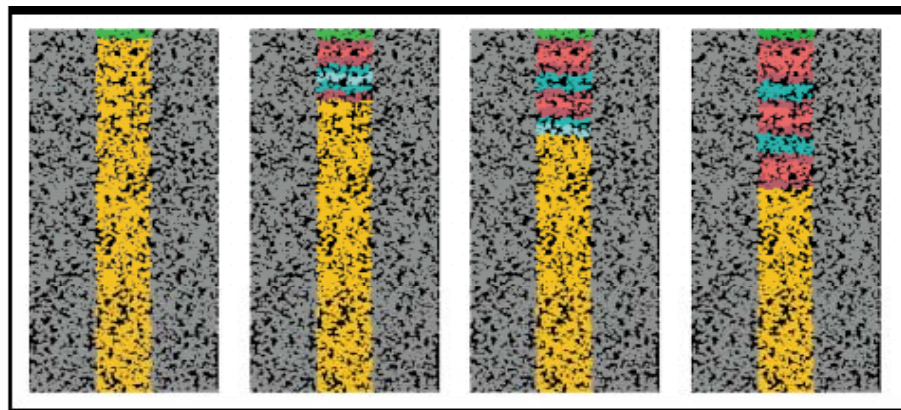


Figure 1.14. : Génération de tubes par le langage de Weiss[10].

D'un point de vue application, ce langage n'a pas été testé que pour le cas de l'exemple précédent. Néanmoins, nous constatons clairement que la mise en œuvre de motifs compliqués nécessite plus de marqueurs.

1.8.4. Intérêt de l'approche langage

A la suite de cette synthèse, il est clair que l'approche langage est indispensable dans le cas des systèmes complexes ayant un très grand nombre de parties interconnectées de manière inconnue. Pour assurer son fonctionnement, l'abstraction de l'environnement et même la topologie est issue des principes de base du calcul amorphe. Dans ce contexte, il est sain d'utiliser des langages, écrire nos programmes à un niveau élevé d'abstraction et c'est au traducteur (compilateur) de s'occuper des détails d'implémentations. Radhika Nagpla a survolé les avantages de ce style de programmation ainsi que l'apport du passage d'un objectif global à des interactions très simples [13] :

- On peut tirer profit des techniques traditionnelles pour générer un motif et se contenter de modèles biologiques pour assurer la robustesse à une échelle locale.
- On peut raisonner sur la classe de structures pouvant être obtenues.

- L'analyse d'un système complexe devient faisable.

1.9. Applications du calcul amorphe

Les recherches concernant le calcul amorphe datent d'une dizaine d'années. Cette période est relativement courte pour avoir permis de nombreux développements méthodologiques. Le but de cette section est de montrer deux applications du calcul amorphe, à savoir la synthèse et la reconstruction d'images 2D. Il convient alors d'écrire une spécification globale en terme de règles très simples. En traduisant cette spécification par le biais de primitives de diffusion, les résultats obtenus démontrent une robustesse et permettent d'acquérir une information approfondie sur le calcul amorphe.

1.9.1. Synthèse d'images

La synthèse d'images 2D vise la création des images dites de synthèse à partir d'un certain nombre de paramètres. C'est ainsi que A.Kondacs, un des membres de MIT, a publié un langage dédié à la synthèse d'images 2D à partir d'entités de calcul amorphe. Selon son article [43], l'objectif est de compiler une forme prédéterminée en un programme qui contrôle la réplication de cellules afin de générer approximativement la forme 2D souhaitée.

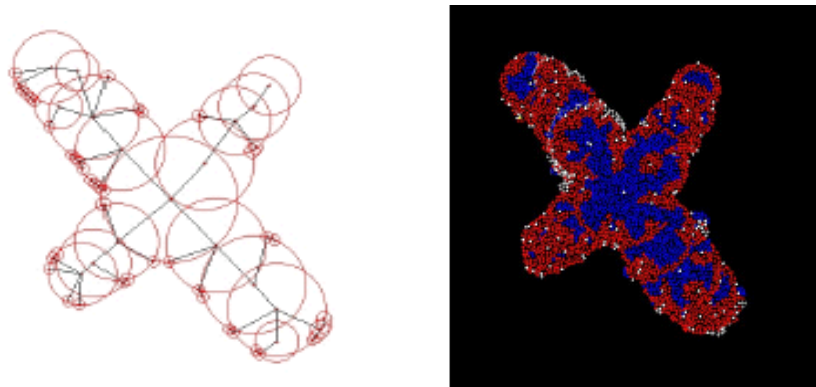


Figure 1.15.a.: Décomposition d'un motif. Figure 1.15.b.: Réplication des particules[43].

Son langage procède en deux phases :

- L'image est décomposée en un ensemble de disques se recouvrant entre eux tels que les disques adjacents soient reliés par des points de référence.
- Chaque cellule exécute deux primitives de base : localisation des points de référence, réplication.

Au niveau le plus bas, chaque cellule peut exécuter cinq actions de base : changer l'état interne, émettre un gradient, se reproduire, mourir, diffuser un message aux particules voisines. A un niveau élevé, une cellule peut activer ou désactiver une autre cellule et mesurer la distance par triangulation. Si elle est point référence, elle diffuse un gradient

aux autres cellules afin qu'elles puissent avoir des coordonnées. De plus, la procédure de compilation qui a comme entrée une carte de pixels peut être schématisée comme suit :

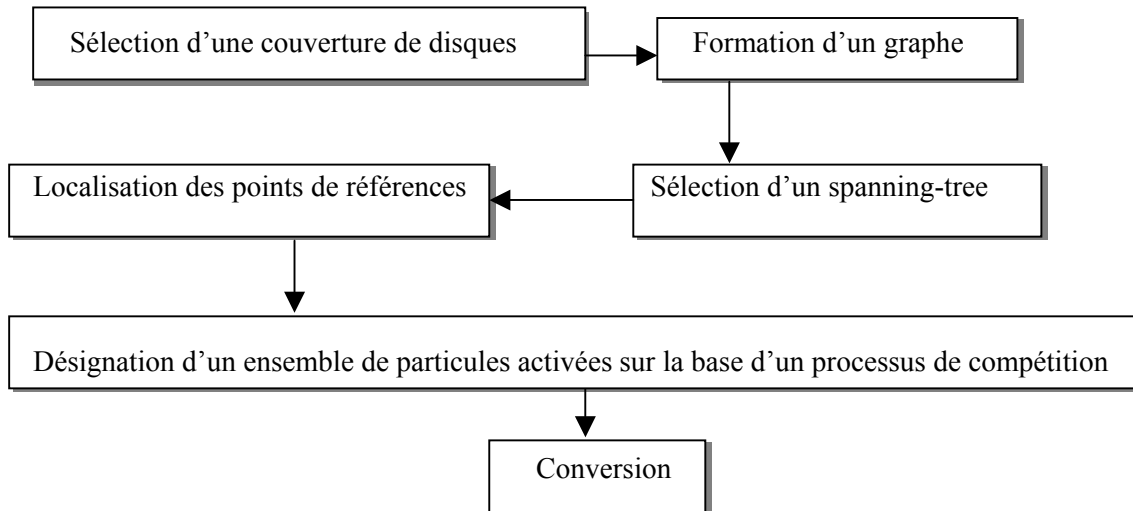


Figure1.16. : Principe de la synthèse d'images par calcul amorphe.

1.9.2. Reconstruction d'images

A ce stade, on peut facilement constater que la génération de motifs 2D est l'application la plus dominante dans le contexte du calcul amorphe. Ayant analysé les langages orientés calcul amorphe, C.Frost [44] a présenté une approche décentralisée pour la reconstruction d'images 2D de façon amorphe. Son point de départ constitue une critique aux approches proposées représentant l'image dans un ordinateur classique. Le principe de base est la construction d'une carte contenant la description d'une image inconnue 2D :

- Injecter une grande population de cellules(particules).
- Calculer l'intensité et diffuser un message aux autres cellules pour les informer si la particule se trouve sur l'image ou pas.
Ecouter aux messages diffusés par les autres et repérer l'emplacement de chaque message.
- Si chaque particule reçoit un message de toutes les autres particules un observateur (éventuellement, l'une de cellules) est interrogé sur l'emplacement des autres cellules.

Bien que ses motivations soient très intéressantes, cette approche repose sur les suppositions suivantes: environnement statique, capteurs à lumière, mémoire suffisante pour stocker tous les messages, communications locales et des identités uniques. Dans cette méthode particulière, C.Frost a utilisé un environnement de simulation Cell-Sim développé à l'université de Virginia. Il s'agit d'un langage qui s'exécute sous Java. Cell-Sim permet la distribution d'une collection de cellules chacune exécutant son programme localement et de manière asynchrone. Son travail a abouti à trois méthodes. L'une d'elles affecte des coordonnées relatives aux cellules recherchées par largeur

d'abord. Une cellule de référence est marquée par (0,0). Elle attribut progressivement des coordonnées aux autres cellules sur la base de la distance calculée pas à pas. L'utilisation de couleurs permet de différencier les cellules qui se trouvent sur l'image. Le résultat a été analysé par des méthodes statiques telle que l'analyse par composantes principales.

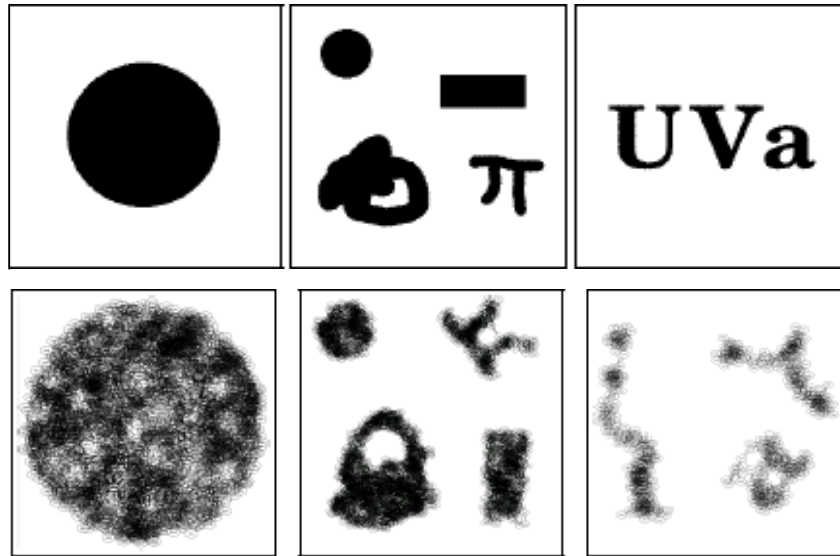


Figure 1.17. : Reconstruction d'images 2D par calcul amorphe [44].

1.10. Approches basées calcul amorphe

En abordant un nouveau paradigme comme le calcul amorphe, il est légitime de présenter quelques techniques liées à ce domaine et les positionner par rapport au calcul amorphe. Ainsi, nous constatons que chacune des techniques distribuées présentées ci-dessous respecte seulement quelques contraintes du calcul amorphe.

❖ Bien adapté aux systèmes tolérants aux pannes, le modèle proposé par S.George, D.Evans, et S.Marchette [45] présente presque les mêmes aspects du calcul amorphe. Son principe a été tiré des cellules biologiques pouvant s'auto réparer. En revanche, les cellules communiquent localement et peuvent s'adapter en exécutant différentes fonctions. Ce modèle qui a été appliqué au cas des réseaux de sonde. Il permet la spécification d'une configuration initiale de cellules. Chaque cellule peut se diviser, produire une protéine ou diffuser une substance chimique. En l'absence de diffusion, les cellules concernées se régénèrent. Au final, les cellules sont conçues comme un automate qui, à partir d'un état peuvent transiter vers un autre état selon une syntaxe : Transitions, Diffuses, State, ... Ce modèle supporte toutes les contraintes du calcul amorphe sauf introduction de la direction des cellules. Par exemple, les figures suivantes schématisent la génération d'une sphère à partir d'une cellule initiale.



Figure 1.18.a : Cellules initiales[47]. Figure 1.18.b. : Défaillance d'une cellule [47]. Figure 1.18.c : Auto répartition des cellules[47].

❖ Les concepteurs peuvent trouver Co-fields un outil efficace pour coordonner le mouvement d'un ensemble d'agents. Signalons tout d'abord que les agents considérés doivent communiquer par diffusion locale de champs. Co-fields est original dans le sens où la valeur et la forme du champ calculé guideront seule le mouvement d'un agent. Les équations décrivant le déplacement d'un agent i sont [46]:

$$\frac{dx_i}{dt} = -V \frac{\partial CF(x,y)}{\partial x} \quad (1.12)$$

$$\frac{dy_i}{dt} = -V \frac{\partial CF(x,y)}{\partial y} \quad (1.13)$$

$CF(x,y)$: champ de coordination, V : la vitesse. Le principe est de calculer un champ de coordination CF dont le minimum ou le maximum guidera le mouvement de chaque agent.

❖ Quant aux autres approches basées calcul amorphe, la technique proposée par K.Stoy et R.Nagpal [47] est orientée pour l'auto configuration des modules d'un robot. Plus précisément, la taille de la configuration s'ajuste en temps réel avec le nombre des modules disponibles. L'approche combine deux axes : la configuration globale est représentée par un plan pour construire un motif à base de briques et des règles locales permettant aux modules de recruter les modules voisins. Chaque module possède six connecteurs : est, ouest, nord, sud, haut, bas. Il peut capter les autres modules dans son voisinage. Dotée d'idées simples et très claires, cette méthode choisit un module site aléatoirement. Si les voisins se retrouvent dans l'emplacement souhaité, des coordonnées seront affectées à eux. Dans le cas contraire, le module site attire ses voisins à l'aide d'une substance chimique. Vis à vis du calcul amorphe, l'orientation de chaque module est une information indispensable. D'un point de vue avantage, le nombre de messages et le temps de configuration sont deux paramètres importants favorisant le recours à cette technique.

❖ Digital Hormone Model publié par W-M.Shen, P-Will, A.Galastyan et C-M.Chuong en 2004, est une approche bioinspiring étroitement liée au calcul amorphe. L'objectif est de contrôler les tâches d'un ensemble de robots. D'après [48], ce modèle mathématique considère des robots comme étant des cellules pouvant communiquer et collaborer via

des hormones. Dans cette optique, le modèle établi est un réseau dynamique de robots possédant des connecteurs à l'environnement physique, un ensemble de récepteurs probabilistes désignant les fonctions de réaction et un ensemble d'équation de réaction-diffusion des hormones. La sélection d'une action B_i est faite sur la base d'une fonction probabiliste P qui dépend elle-même de l'information du connecteur C , l'information du capteur S , la valeur des variables V_i et l'hormone reçue H tel que $P(B/C,S,V,H)$. L'équation de réaction-diffusion d'une hormone avec concentration C est :

$$\frac{\partial C}{\partial t} = \left(a \left(\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} \right) + R - bC \right) \quad (1.14)$$

Le premier terme de cette équation dénote la diffusion tandis qu'est le terme de réaction et b est le taux de dissipation. Le principe à retenir consiste à sélectionner une action par P , l'exécuter, propager une hormone et enfin simuler la diffusion. Par rapport au calcul amorphe, ce modèle respecte bien la contrainte de communication locale par diffusion entre un ensemble de robots n'ayant pas des identités globales.

❖ L'omniprésence des ordinateurs est, sans doute, appelée à devenir une infrastructure du calcul de demain. Mais reste encore le problème de l'interaction avec un tel système. G. Bour, G. Huzler et B. Gortais [49] ont développé une approche cognitive et décentralisée s'intéressant à l'interaction avec des systèmes ubiquistes (omniprésents). Dans le cadre de cette méthode, un ensemble d'entités mobiles peuvent s'adapter en produisant des ambiances visuelles et sonores qui considèrent un acteur (humain ou objet) et le système comme partenaire où l'expression de l'un aide l'autre et inversement. Le fonctionnement de ces entités pouvant changer leurs couleurs est schématisé comme suit :

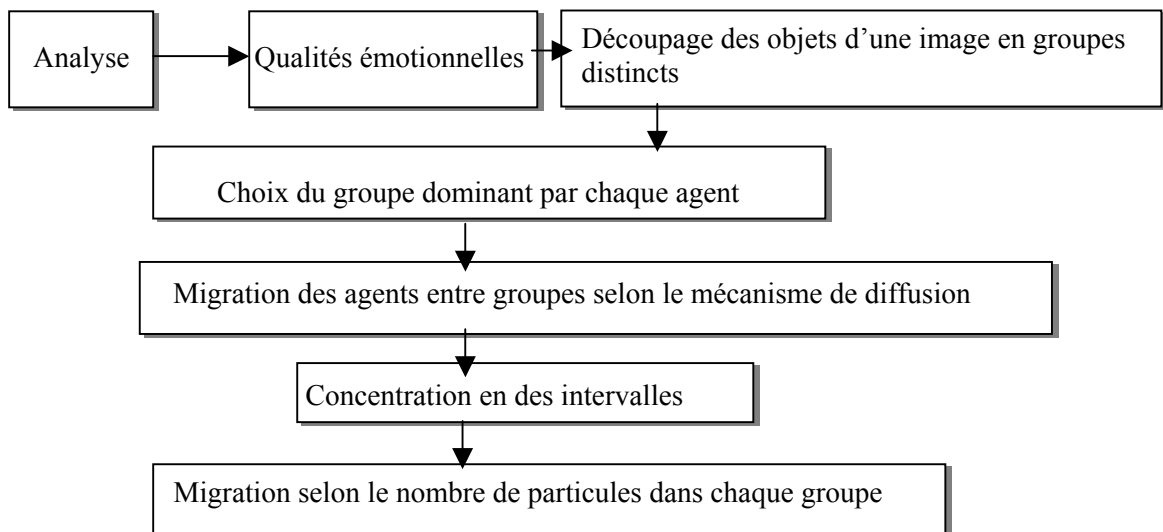


Figure 1.19. : Principe de la méthode.

La particularité du modèle conçue réside dans ses potentialités en terme de performance. Il converge rapidement vers une solution globale.

❖ De nombreuses approches concernant les réseaux de sonde posent le problème du routage. Pour ce type de problème, une approche de routage basée calcul amorphe a été publiée par L. Gang, J. Liu et X. Jin [50]. Il s'agit de la simulation d'un environnement inaccessible (telle que la planète de Mars) contenant des entités de calcul. Ces entités doivent router des paquets de données selon l'information locale de stimulus en maximisant le rapport énergie coût.

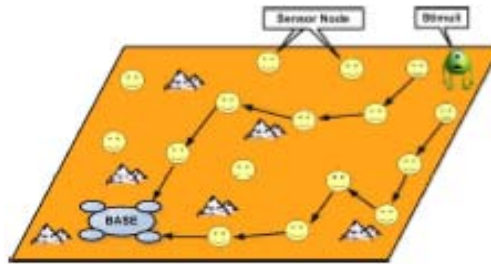


Figure 1.20 : Exemple d'un réseau de sonde[49].

A la différence des schémas classiques stochastiques ou ceux basés sur le coût, un nœud N_i choisit un nœud N_j selon cette métrique :

$$M_i = \max_{j \in FT_i} \frac{E_j}{C_j^\alpha} \quad (1.15)$$

En plus, les expériences ainsi réalisées prennent en compte le temps associé à chaque entité. Elles ont démontré un temps de calcul nettement inférieur par rapport aux techniques traditionnelles de routage.

1.11. Conclusion

Entre la robustesse, le parallélisme et la translation d'un objectif global en terme d'interactions locales, le calcul amorphe est destiné aux systèmes complexes où la notion d'abstraction est indispensable. Son objectif est d'autant plus fort que les automates cellulaires. En utilisant seulement une grille irrégulière, Les particules de calcul amorphe sont capables d'émerger un résultat intéressant. A partir de cette synthèse, il est clair que le calcul amorphe repose sur des contraintes générales déjà présentes en informatique omniprésente, en wearable computing et en ingénierie cellulaire. Ces domaines sont très asymptotiquement équivalents dans le sens qu'ils cherchent à rendre le calcul encore plus intelligent et discret par un nombre massivement parallèle d'entités. Allant du langage GPL au langage RSEAM, le principe est différent mais l'objectif est le même. Il s'agit de traduire des règles de haut niveau en terme de primitives biologiques très simples.

Confrontés aux problèmes d'organisation d'une tâche globale au sein d'un environnement de calcul amorphe, les pionniers du calcul amorphe ont mis au point un seul algorithme s'intéressant à la construction d'une hiérarchie de calcul amorphe. Cet algorithme constitue le cœur de l'idée qui sera présentée en chapitre 3 et 4.

2.1. Introduction

Dans pratiquement tous les domaines, l'analyse de données prend une place essentielle. Cet axe de recherche s'est développé grâce aux travaux théoriques et expérimentaux de la statistique, mathématiques et l'informatique. A l'ère de ce processus, la classification est la technique adoptée pour regrouper des objets similaires. En utilisant un critère de similarité, il devient possible d'exploiter des bases volumineuses par la construction de groupes d'objets. Le Data mining et la segmentation d'images sont deux champs d'application de la classification. Le premier se charge de l'extraction de connaissances à partir des groupes obtenus, tandis que le second partitionne une image en classes de pixels similaires. Récemment, une accumulation de méthodes de classification est disponible chacune promettant des avantages et des inconvénients et pouvant répondre à des situations particulières. En plus des méthodes hiérarchiques et non-hiérarchiques qui sont les plus connues, une gamme de réseaux connexionnistes apprenant la topologie de l'espace d'entrée est sans doute d'un intérêt important. Fonctionnant selon un apprentissage incrémental, l'algorithme du Growing Neural Gas (GNG) permet la construction d'un réseau reflétant la structure induite par les données présentées. En effet, le GNG est une méthode très simple procédant itérativement par auto organisation. Cette technique est l'objet de notre travail mais son explication doit passer par l'état de l'art des techniques de classification.

2.2. Processus de classification

On désigne par classification le processus d'assignation des objets à leurs classes correspondantes. Selon l'hypothèse de départ, le procédé de classification prend un acheminement différent. En présence d'attributs caractérisant les classes, l'objectif de la classification est la dérivation d'une connaissance de haut niveau. Si aucun attribut n'est disponible, la finalité consiste à maximiser la qualité des groupes obtenus [51]. Partant d'un ensemble P de points $\{p_1, p_2, \dots, p_n\}$, la thématique de la classification consiste à regrouper les données semblables telle que la similarité inter-groupe soit minimisée et la similarité intra-groupe soit maximisée [52]. Dans le domaine de la classification, deux points méritent une attention particulière :

- Sélection d'une mesure de similarité : les méthodes de classification recourent fortement à des distances pouvant être calculées entre les objets à agréger ou les objets et les clusters déjà constitués [52]. Ainsi, différents critères de similarité ont été définis. Soient C_i, C_j, C_k des clusters de P , p, p' deux points de P , n_i le nombre de points dans C_i et m_i la moyenne de C_i :

$$S(C_i, C_j) = \min_{p \in C_i, p' \in C_j} \{dis(p, p')\} \quad (2.1)$$

$$S(C_i, C_j) = \max_{p \in C_i, p' \in C_j} \{dis(p, p')\} \quad (2.2)$$

$$S(C_i, C_j) = \frac{1}{n_i n_j} \sum_{p \in C_i} \sum_{p' \in C_j} \{dis(p, p')\} \quad (2.3)$$

$$S(C_i, C_j) = dis(m_i, m_j) \quad (2.4)$$

En raffinant encore les définitions des critères de similarité, les distances suivantes ont été proposées dans la littérature [53] :

Distance Euclidienne :

$$D = \sqrt{\sum_{k=1}^m (x_{ik} - x_{jk})^2} \quad (2.5)$$

Distance de Hamming :

$$D = \sum_{k=1}^m \|x_{ik} - x_{jk}\| \quad (2.6)$$

Distance de Minkowski :

$$D = \left[\sum_{k=1}^m (x_{ik} - x_{jk})^p \right]^{\frac{1}{p}} \quad (2.7)$$

- La classification des données en établissant un schéma visualisant les groupes de données.

2.3. Applications de la classification

Entre le Data mining et la segmentation d'images, la classification est une voie médiane pour le traitement de données discrètes ou continues. En abordant le Data mining, nous devons être conscients, que l'aboutissement à un modèle final doit respecter un processus long qui doit, à son tour, bénéficier des résultats offerts par d'autres techniques. Passer par toutes les étapes du Data mining, c'est vraisemblablement appliquer une phase de prétraitement puis une autre de traitement des données mises en jeux. Pour pallier cela, clarifier le problème visé afin de le formuler est l'étape initiale dans toute méthodologie du Data mining. Puis vient la transformation des données, l'inventaire des données utiles, l'observation des corrélations et le contrôle de la cohérence. Dans une deuxième phase, on est amené à chercher un modèle et à évaluer les résultats obtenus [54]. A côté de l'analyse des associations et la régression, la classification peut être pratiquée, quant à elle dès le démarrage d'un projet de Data mining. Généralement, comprendre les similitudes entre les données permet d'élargir l'imagination sur leur utilisation. Par conséquent, on pourra construire une abstraction utile devant un nombre gigantesque de données. D'autre part, la segmentation d'images, qui s'apparente à l'assignation des pixels aux régions correspondantes, est axée sur l'adaptation des algorithmes conventionnels de classification. Ainsi, on soulignera

également les nombreux travaux en biologie, médecine et en intelligence artificielle qui ont conservé l'avantage des techniques de classification pour la segmentation d'images.

2.4. Typologie des méthodes de classification

Au-delà des méthodes de classification, anciennes ou récentes; proposées dans la littérature, le choix d'une méthode est parfois problématique. Cela est relativement lié à la nature des données, l'objectif de la classification et les préférences utilisateur [55]. Exploitant judicieusement une masse de données, toutes les approches de classification se situent certainement parmi les prometteuses et peuvent appartenir aux catégories suivantes:

2.4.1. Classification locale versus globale

Si le voisinage local est le seul facteur affectant la décision d'appartenance d'un point à un groupe, la classification est dite locale, sinon elle est considérée globale. A titre de comparaison, les membres d'un groupe restent invariables lors d'un processus de classification local tant qu'un changement apparaît en dehors du voisinage du point manipulé. En plus, les méthodes de classification locales sont surtout retenues dans le cas d'une classification incrémentale [52]. Les méthodes de classification locales sont de plus en plus nombreuses. L'utilisation de ressources limitées est la raison principale de leur introduction. Elle est controversée par un temps d'exécution important.

2.4.2. Classification statique versus dynamique

Un des objectifs poursuivis par les méthodes de classification basées sur un modèle dynamique est la normalisation de la mesure de similarité en proportion du voisinage [52]. En fait, la classification dynamique est étroitement liée à la classification locale. Elle permet de prendre en compte le phénomène du mouvement de l'ensemble d'unités effectuant la classification. Ce type de classification est un exemple de techniques où le calcul du voisinage et la définition d'une mesure de similarité s'articulent parfaitement pour partitionner l'ensemble de données en groupes. Pour comprendre cette catégorie, on pourra se rapporter à la section du Growing Neural Gas issu du principe de la classification dynamique.

2.4.3. Classification floue versus exacte

A l'inverse de la classification exacte qui affecte chaque point à un seul cluster, la classification floue affecte un point à plusieurs clusters selon un degré d'appartenance. Cette technique nécessite la spécification d'une partition floue de N objets en K partitions. Pour cela, on décrit les membres $N \times K$ d'une matrice U , tel que $u_{ij} \in [0,1]$ [56]. Pour réaliser une classification floue, le critère de similarité prend en compte la distance et le degré d'appartenance. Plus explicitement, la classification floue produit des groupes qui ne sont pas mutuellement exclusives. Le passage d'une classification floue à une classification exacte peut être mené selon le degré d'appartenance

maximum. Intuitivement, les méthodes floues sont très utiles lorsque les limites d'un groupe ne sont pas bien définies [52].

2.4.4. Classification hiérarchique versus non-hiérarchique

Les méthodes de classification hiérarchiques peuvent être pratiquées en procédant en la division de l'ensemble P par plusieurs étapes [57]. La diversité des approches de classification hiérarchiques se décompose à son tour en deux catégories :

Bottom-Up : s'appuie sur la création de n groupes, chacun contenant au moins un objet. L'objectif est la fusion des groupes qui soient similaires d'après l'un des critères cités dans la section précédente. Ce processus est répété jusqu'à un point fixe.

Top-Down : suppose l'existence d'un seul cluster contenant tous les objets. Cette technique divise l'ensemble initial jusqu'à la construction d'un nombre préalable de clusters. Ce type de classification doit désigner le cluster à diviser et en même temps spécifier la démarche de sa division.

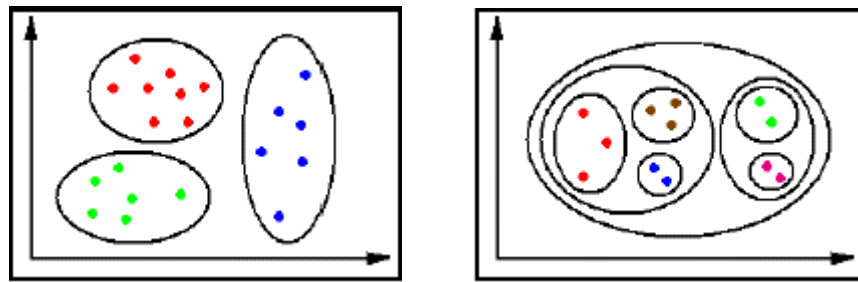


Figure 2.1.a. : Classification non-hiérarchique [56]. Figure 2.1.b. : Classification hiérarchique [56].

En revanche, si l'on évoque les méthodes automatiques de classification hiérarchiques, la méthode Balanced Iterative Reducing And Clustering Using Hierarchies (Birch) est sans doute la plus connue. Cette méthode construit récursivement un arbre. Elle distingue deux composantes : des attributs de classification CF et des attributs de l'arbre de classification CF-Tree. Chaque CF est défini comme un triplet (n, \bar{L}, ss) où n est le nombre de points, \bar{L} la somme linéaire des n points et ss la somme au carré des n points. A chaque insertion d'une entrée, Birch consulte récursivement l'arbre de classification en sélectionnant la feuille adéquate. Si la feuille correspondante peut absorber cette entrée, la donnée sera ajoutée sinon une nouvelle feuille est créée. Si la taille de la feuille est supérieure à un taux prédéfini, le nœud sera partitionné [52].

Cependant, l'autre piste ouverte est attribuée à la classification non-hiérarchique. Etant donné n objets et un nombre K , $K \ll n$, la classification non-hiérarchique construit K partitions en une seule étape. L'intérêt de cette démarche est d'avoir des groupes, chacun contenant au moins un objet où chaque objet doit appartenir à un seul groupe [57]. En pratique, l'obstacle majeur pour les méthodes hiérarchiques découle de la difficulté de l'interprétation du dendrogramme associé [52]. D'autre part, le nombre initial

de partitions est un paramètre important quant à l'utilisation des méthodes non-hiérarchiques.

2.4.5. Méthodes basées sur la densité

En présence d'objets complexes, les approches basées sur la densité doivent être ajoutées à la panoplie des outils de classification [58]. Ces méthodes procèdent par la définition d'une forme arbitraire (cercle, grille). Il faut ensuite calculer la densité des régions définies. Dans bien des cas, une région peut s'exprimer comme un bruit selon une faible densité ou encore comme un cluster en s'exprimant selon une grande densité. Nous donnons ci-dessous, à titre d'exemple, un aperçu de la méthode GAM [52]:

- On suppose que l'espace des points est représenté dans une grille.
- Initialiser une valeur minimum et maximum pour le rayon des cercles.
- Tester chaque région (cercle).
- Toute région possédant une grande densité est considérée comme un cluster.
- Augmenter le rayon.
- Tester les régions calculées.

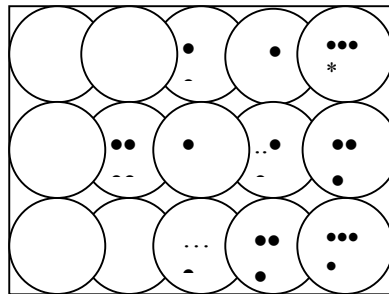


Figure 2.2 : Partitionnement d'un espace par la méthode GAM.

Cette méthode donne généralement un bon résultat. Cependant, le temps de calcul est important. La raison principale réside dans la possibilité de tester le même point plusieurs fois au sein d'une seule itération.

2.4.6. Méthodes basées sur un graphe

A l'instar de la classification basée sur un graphe, le mécanisme sous-jacent construit un graphe de proximité modélisant les interactions spatiales entre les points représentant les observations. Cette technique propose de regrouper des observations par l'étude des connexions en calculant un critère de similarité. D'un point de vue avantage, la classification basée sur un graphe se donne pour but d'être une méthode efficace en dépit du bruit. Cependant, elle doit être appliquée soigneusement car elle risque de supprimer des nœuds ou des connexions qui déconnectent le graphe. En matière de la classification basée sur un graphe, SMTIN est une conséquence de la triangulation de Delaunay et le principe de la classification. D'après une lecture de cette approche, on apprécie que le graphe de proximité est celui de Delaunay. Le processus de classification distingue les connexions intra groupe et celles inter groupe. Dans ce contexte, le principe consiste à

éliminer toute connexion dont le poids dépasse un certain seuil. Certes, le seuil utilisé constitue l'un des inconvénients de cette méthode dont la mise en œuvre est très facile [52].

2.4.7. Classification basée sur un modèle

La particularité de la classification basée sur un modèle est la construction d'un modèle statistique qui reflète approximativement la distribution des données. Pour résoudre le problème du choix de modèle, quelques indices ont été présentés dans la littérature: l'objectif du cluster à analyser, les propriétés de la base de données, ... [58]. A partir du modèle établi, l'objectif est la dérivation de critères servant à la classification. Leur rôle est d'optimiser l'assignation d'un objet à un cluster. Notons que le modèle Gaussien et les chaînes de Markov cachées sont les deux modèles les plus utilisés. Il faut cependant mentionner que le coût d'établissement d'un modèle limite énormément le recours à cette méthode.

2.5. Méthode du K-Means

L'algorithme du K-Means fut l'une des approches non-hiérarchiques les plus populaires [56]. Il nécessite comme entrée un ensemble d'observations et un ensemble de centres. Il affecte chaque observation au centre le plus proche. Soit un ensemble $A = \{z_1, z_2, \dots, z_n\}$ d'observations, l'objectif est de résumer A par $w = \{w_c, c=1 \dots k\}$. Chaque vecteur w_c est appelé vecteur prototype. Bien entendu, la fonction qui effectue la transformation de A à w est notée χ . Elle tente de diminuer la distance entre les observations et les vecteurs codes. En effet, la finalité du K-means est de minimiser la fonctionnelle suivante [59]:

$$I(w, \chi) = \sum_C \sum_{z_i \in P \cap C} \|z_i - w_c\|^2 \quad (2.8)$$

Formellement, la méthode du K-means a été décrite par les étapes suivantes :

1. Phase d'initialisation, $t=0$.
Choisir les k vecteurs codes initiaux (en général, de manière aléatoire).
2. Phase itérative : à l'itération t , l'ensemble des vecteurs prototypes w^{t-1} de l'étape précédente est connu.
Mise à jour de la fonction χ :

$$\chi(z) = \operatorname{argmin}_i \|z - w_i\|^2 \quad (2.9)$$

Calcul des vecteurs prototypes:

$$w_c = \frac{\sum_{z_i \in P_c \cap A} z_i}{n_c} \quad (2.10)$$

n_c : Nombre d'éléments du cluster c .

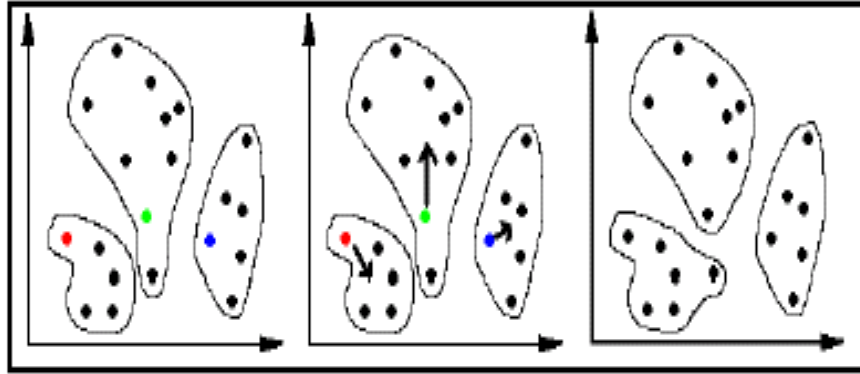


Figure 2.3 .Principe du K-means [56].

En ce qui concerne les performances de cette méthode, peu de paramètres doivent être spécifiés. Malheureusement, le nombre k de clusters doit être fixé dès le départ et demeure fixe tout au long de la classification [60]. Par analogie au K-means, la méthode ISODATA (iterative self-organizing data analysis technique) est une méthode de classification automatique affectant chaque observation d'entrée au centre le plus proche. A la différence du k-means, ISODATA exige la spécification de six paramètres utilisateur. En plus, elle fait incorporer une étape supplémentaire. A chaque itération, les clusters jugés larges sont partitionnés et les clusters dont les centres sont proches les uns des autres doivent être fusionnés. Avec cette étape, on élimine l'un des inconvénients du K-means. Plus précisément, une grande valeur de k ne posera aucun problème pour ISODATA pouvant optimiser le nombre de centres par fusion.

2.5.1. K-means flou (FCM)

L'algorithme FCM couvre beaucoup plus le domaine d'imagerie médicale. C'est une méthode de classification floue. Il autorise l'appartenance d'un objet à plusieurs classes selon un certain degré. La fonctionnelle à minimiser par FCM est décrite par l'expression suivante [61]:

$$I(w, U, \chi) = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m d^2(w_j, z_i) \quad (2.10)$$

Non seulement, les vecteurs prototypes sont recalculés mais aussi les degrés d'appartenance. A chaque étape de l'algorithme FCM, on évalue :

$$u_{ij} = \sum_{k=1}^C \left(\frac{d_2(z_j, w_i)}{d_2(z_j, w_k)} \right)^{2(m-1)} \quad (2.11)$$

$$w_i = \frac{\sum_{k=1}^N u_{ik}^m z_k}{\sum_{k=1}^N u_{ik}^m} \quad (2.12)$$

m est une valeur expérimentale. Le critère d'arrêt peut être étudié en prenant en compte la norme de la matrice U ou la stabilité des centres.

2.6. Approches évolutionnaires

Les nouveaux travaux issus de l'intelligence artificielle s'articulent au tour de deux approches : la vie artificielle et les systèmes multi agents. Dans la veine de la vie artificielle, les approches évolutionnaires ont été originalement conçues pour optimiser une solution dans un espace de recherche. En effet, on a même abouti à la classification de données par les approches évolutionnaires. Il s'agit d'optimiser la localisation des centres des clusters. L'exploitation de cet axe de recherche passe par les étapes suivantes [56]:

- Choisir une population aléatoire de solutions chacune correspondant à une partition k .
- **Répéter**
Attribuer à chaque solution une fitness qui est inversement proportionnelle à l'erreur.
Utiliser les opérateurs de sélection, combinaison et de mutation et évaluer la nouvelle population.
- **Jusqu'à** convergence

2.7. Réseaux de neurones

La simulation d'un phénomène complexe par apprentissage est l'autre bout de la vie artificielle. C'est ainsi que se sont développés les réseaux de neurones dont le fonctionnement est inspiré du cerveau humain. Historiquement, le premier modèle simpliste réalisé est celui du neurone formel [62]. Le principe de cet outil est la génération d'une réponse selon une fonction de seuillage. En particulier, chaque neurone est considéré comme une unité de traitement produisant une réponse y en fonction d'un vecteur d'entrée x et les poids w des connexions :

$$y = \sum_{i=1}^n w_i x_i \quad (2.13)$$

A la fin, la réponse du neurone est donnée par la fonction z :

$$z = \text{bin}(y - \mu) \quad (2.14)$$

bin est une réponse binaire et μ est le résultat de seuillage.

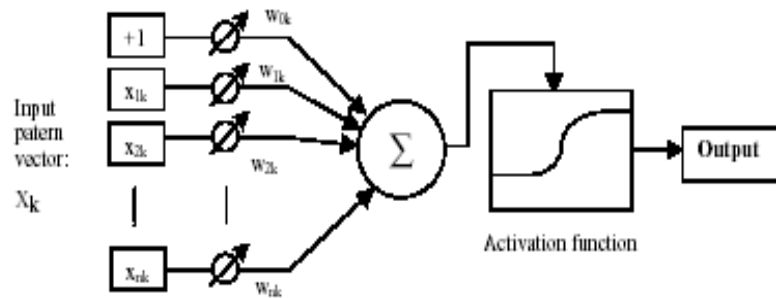


Figure 2.4. : Modèle du neurone formel [62].

Du coup, d'autres réseaux de neurones sont apparus. Ces réseaux opèrent selon une architecture et une stratégie d'apprentissage. Aujourd'hui, les réseaux de neurones sont en pleine exploitation. Les avantages de ce principe est la capacité de la généralisation des données et un grand parallélisme du à l'architecture du réseau de neurone [62]. Habituellement, les réseaux de neurones sont spécialisés dans la reconnaissance de formes et leur utilisation n'est possible qu'après une longue phase d'apprentissage. En revanche, de nouvelles catégories des réseaux de neurones ont prouvé leurs capacités à regrouper des données de façon non explicite.

2.7.1. Apprentissage compétitif

Aujourd'hui, nous sommes en présence d'une multitude de techniques d'apprentissage, chacune se décline différemment. On trouve d'ailleurs l'apprentissage supervisé et non supervisé qui se distinguent par la connaissance préalable de la réponse souhaitée à une entrée. Par ailleurs, l'apprentissage compétitif a permis de formuler le problème de classification d'un ensemble $\{x_1, x_2, \dots, x_n\}$ de n vecteurs d'entrée. L'idée qui a été empruntée consiste à développer un réseau de neurones constitué de k unités en compétition représentant k clusters. Ces unités reçoivent les poids w à partir d'une couche d'entrée. Le processus de compétition exécuté de façon récursive est décrit par les trois étapes suivantes [63] :

- Association d'un vecteur w à chaque neurone.
- Sélection d'un vecteur d'entrée x .

Calculer $x \cdot w_i, \forall i=1..k$

Sélection de l'unité b tel que $x \cdot w_b \geq x \cdot w_i, \forall i=1..k$

- Mise à jour des w .

2.7.2. Apprentissage Incrémental

Le développement de méthodes d'apprentissage pour la classification de données a reconnu une révolution majeure au cours de ces dernières années, avec en particulier l'avènement des méthodes d'apprentissage incrémental [64]. Lors des techniques d'apprentissage traditionnelles, les deux étapes d'apprentissage et d'utilisation sont à distinguer. On s'est très vite rendu compte que telles méthodes étaient contraignantes si l'on veut ajouter de nouvelles informations. Pour cette raison, on s'est progressivement dirigé vers l'apprentissage incrémental qui consiste à apprendre des informations de manière progressive sur des intervalles de temps plus ou moins éloignés au fur et à mesure qu'elles seraient présentées. L'avancé significative qui a été apportée par l'apprentissage incrémental est l'intégration de nouvelles informations sans reconsidérer l'ensemble de ce qui a été appris. Ainsi, dans des environnements temps réel, l'apprentissage incrémental a démontré son efficacité pour répondre à des problématiques spécifiques en évitant la désignation de toutes les situations possibles que doit affronter le système. En outre, de nombreuses études ont montré que l'apprentissage incrémental s'applique dans plusieurs domaines et correspond aux objectifs suivants [65]:

- Minimisation de l'erreur.
- Maximisation de l'entropie en cas où chaque vecteur prototype d'entrée a la même chance d'être le gagnant pour un signal ζ .
- Transformation d'attributs en préservant la relation de similarité.

2.8. Cartes auto-organisatrices (SOM)

Agissant à deux niveaux (couche d'entrée, couche de neurones), les cartes auto-organisatrices forment des réseaux de neurone très sophistiqués. D'un point de vue structure, une carte auto-organisatrice φ est constituée de l'interconnexion de neurones positionnés régulièrement. Ainsi, la distance δ entre deux neurones quelconques c, r est le plus court chemin entre c et r . Chaque neurone est associé d'une zone d'influence modélisée par une fonction. Généralement, les fonctions les plus utilisées sont des fonctions paramétrées T [59] :

$$K^T(\delta) = K(\delta/T) \quad (2.15)$$

Plus explicitement, les fonctions paramétrées K qui ont été définies sont les suivantes :

$$K(\delta) = \begin{cases} 1 & \text{Si } \delta < 1 \\ 0 & \text{Sinon} \end{cases} \quad K^T(\delta) = \begin{cases} 1 & \text{Si } \delta < T \\ 0 & \text{Sinon} \end{cases} \quad (2.16.a)$$

$$K^T(\delta) = \exp(-|\delta|) \qquad K^T(\delta) = \exp\left(\frac{-|\delta|}{T}\right) \qquad (2.16.b)$$

$$K(\delta) = \exp(-\delta^2) \qquad K^T(\delta) = \exp\left(\frac{-\delta^2}{T^2}\right) \qquad (2.16.c)$$

La fonction coût minimisée par SOM est :

$$J_{\text{som}}(\chi, w) = \sum_{x \in A} \sum_{c \in C} K^T(\delta_{c, \chi(x)}) \|x - w_c\|^2 \qquad (2.17)$$

Notamment, la minimisation de l'équation (2.18) a donné lieu à plusieurs versions de SOM. Vis-à-vis du processus de classification, SOM offre les avantages suivants :

- Apprentissage implicite de la topologie d'un espace d'entrée.
- Réduction de la dimension.
- Visualisation de l'information.

Pratiquement, le procédé de base de SOM est schématisé en figure (2.6).

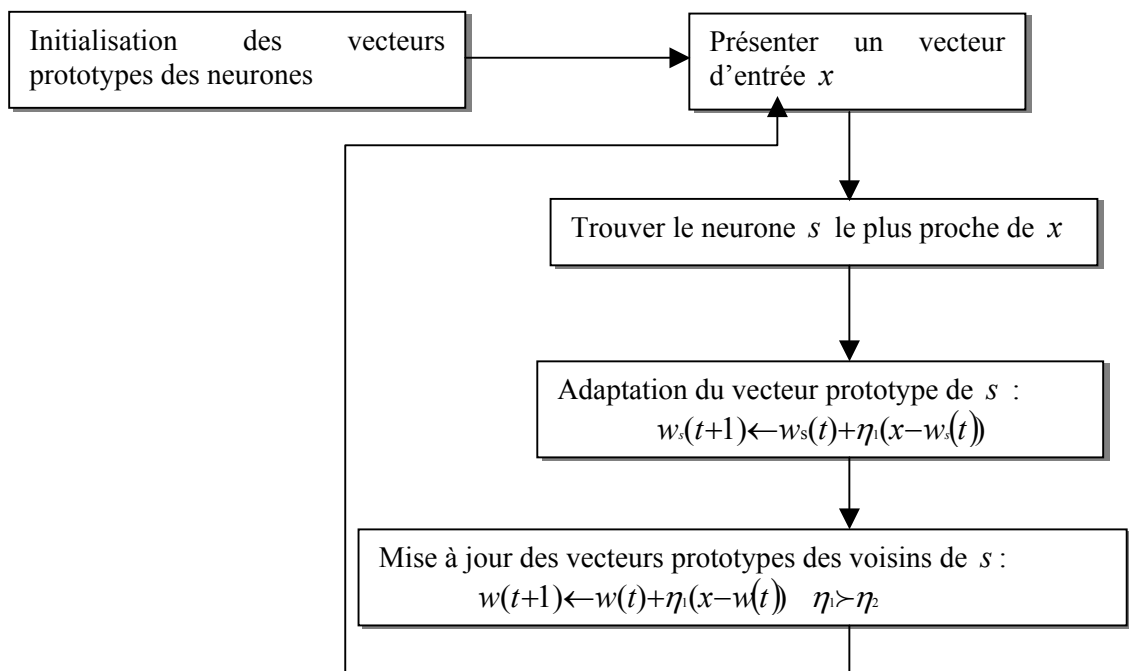


Figure 2.5. : Principe de SOM [63].

Le critère d'arrêt peut être :

- Un temps ou un nombre de neurones maximum.
- Stabilité de la carte.

2.8.1. Cartes auto-organisatrices probabilistes (PRsOM)

En terme de fonctionnalité, la version probabiliste des cartes auto-organisatrices associe à chaque neurone c de la carte une fonction de densité normale f_c qui a pour expression [58]:

$$f_c(x, w, \sigma) = \frac{1}{(2\pi)^{n/2} \sigma_c^n} \exp\left(-\frac{\|x - w_c\|^2}{2\sigma_c^2}\right) \quad (2.18)$$

Par analogie au K-means, deux étapes sont requises :

- Affecter chaque observation au neurone le plus probable selon la fonction de densité P_c :

$$\chi(x) = \operatorname{argmax}_{P_c}(x) \quad (2.19)$$

- Calculer w^t, σ^t en respectant les formules suivantes :

$$w_r^t = \frac{\sum_{i=1}^N x_i k(\delta(r, \chi^{t-1}(x_i))) \frac{f(x_i, w_r^{t-1}, \sigma_r^{t-1})}{P_{\chi}^{t-1}(x_i)}}{\sum_{i=1}^N k(\delta(r, \chi^{t-1}(x_i))) \frac{f(x_i, w_r^{t-1}, \sigma_r^{t-1})}{P_{\chi}^{t-1}(x_i)}} \quad (2.20)$$

$$(\sigma_r^t)^2 = \frac{\sum_{i=1}^N \|w_r^{t-1} - x_i\|^2 k(\delta(r, \chi^{t-1}(x_i))) \frac{f(x_i, w_r^{t-1}, \sigma_r^{t-1})}{P_{\chi}^{t-1}(x_i)}}{n \sum_{i=1}^N k(\delta(r, \chi^{t-1}(x_i))) \frac{f(x_i, w_r^{t-1}, \sigma_r^{t-1})}{P_{\chi}^{t-1}(x_i)}} \quad (2.21)$$

L'implication des cartes auto organisatrices est l'un des objectifs poursuivis lors d'analyse de données satellitaires, ... On cherche à translater l'ensemble d'observations en un ensemble de faible dimension en préservant la topologie initiale [59]. Cependant, la structure de la carte et le nombre de neurones nécessaires sont deux questions auxquelles il faut répondre.

2.9. Growing cell structure (GCS)

Cette méthode à la frontière des cartes auto organisatrices et l'apprentissage incrémental, a été développée par Bernd Fritzke (1991) [66]. Elle permet aux utilisateurs d'avoir une première brique pour la translation d'un espace de grande dimension en un Espace de faible dimension en préservant les relations topologiques. Pour cela, la procédure proposée opère sur une structure prédéfinie et paramétrée K , telle que :

Si $K=1$ le graphe est une ligne.

Si $K=2$, l'élément de base est un triangle.

Si $K=3$, l'élément de base est un tétraèdre.

Côtés variables, le GCS attribut à chaque nœud un vecteur prototype w et une erreur. Ainsi, les règles employées dans le processus d'apprentissage ont été formulées comme suit :

1. Générer un signal d'entrée ζ .
2. Déterminer l'unité s la plus proche de ζ .
3. Mise à jour de l'erreur de l'unité s :

$$\Delta E = \|w_s - \zeta\| \quad (2.22)$$

4. Déplacement de s et ses voisins topologiques :

$$\Delta w_s = e_s (\zeta - w) \quad (2.23)$$

$$\Delta w_i = e_r (\zeta - w_i) \quad (2.24)$$

5. Après chaque λ itérations, déterminer le nœud r possédant l'erreur maximum. r doit déterminer le nœud q , parmi ses voisins, possédant l'erreur maximum.

$$w_r \leftarrow 0.5(w_q + w_r) \quad (2.25)$$

6. Mise à jour de l'erreur par rapport à la moyenne des voisins.

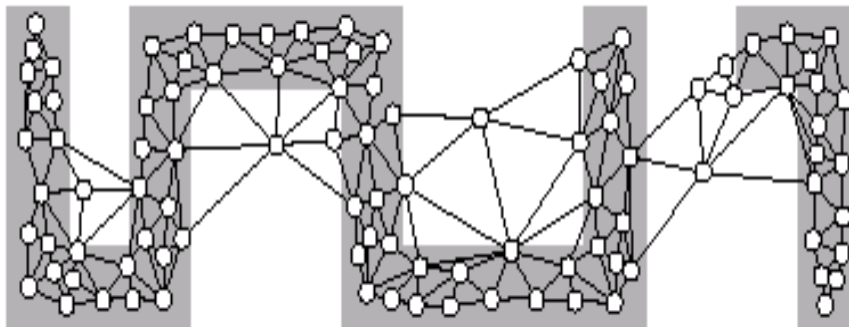


Figure 2.6. : Résultat du GCS [66].

Ce petit tour d'horizon de la méthode GCS montre bien qu'elle n'est pas une méthode de classification... Elle est toutefois une amélioration des cartes auto-organisatrices et constitue la base de l'algorithme suivant.

2.10. Growing Neural Gas (GNG)

Lors de l'utilisation d'une carte auto organisatrice se trouvent deux contraintes déjà mentionnées : la structure de la carte et le nombre de neurones. Ces constatations ont été prises en charge par l'algorithme GCS. Cependant, GCS n'a pas réussi à faire la classification de données. Partant de là, Bernd Fritzke a proposé l'algorithme du GNG s'adaptant aux données par apprentissage incrémental et compétitif. Pour utiliser cette méthode, nous prenons comme condition l'initialisation de deux nœuds reliés par une connexion. Avec un tel algorithme, l'apprentissage d'un espace de données donnera lieu à la création d'un graphe $G=[V(t),E(t)]$. $V(t)$ Est l'ensemble des sommets à l'instant t et $E(t)$ l'ensemble des arrêtes [67]. Le regroupement des données similaires consiste à adapter le nœud le plus proche du stimulus courant, supprimer les connexions impliquant une faible vraisemblance et minimiser l'erreur par l'ajout d'un nœud (dans la zone de grande erreur).

2.10.1. Principe de base du GNG

Le GNG se présente sous la forme du pseudo-code décrit ci-dessous manipulant les paramètres et variables suivantes [68]:

w_k : Vecteur prototype associé à chaque nœud k et représentant sa position dans l'espace.

erreur : Accumulation locale de l'erreur.

e_w, e_n : Deux paramètres différents associés au gagnant et ses voisins.

age : Variable mesurant la vraisemblance entre les nœuds correspondants tel qu'une connexion âgée signifie une vraisemblance faible de la relation topologique et vice versa.

λ : Période d'ajout d'un nouveau neurone.

age_{max} : Paramètre pour faire disparaître une connexion.

α, β : Serrent à contrôler l'oubli des signaux d'erreur.

Informellement, l'idée du GNG peut être résumée par l'exécution récursive des étapes suivantes :

- Création de deux neurones initiaux.
- Génération d'un stimulus d'entrée.
- Adaptation de l'erreur du gagnant.
- Mise à jour du vecteur prototype du gagnant et ses voisins.
- Mise à jour de la variable âge.
- Ajout d'un nouveau neurone après chaque λ itération.

2.10.2. Pseudo-code du GNG

Le GNG a été formulé par Jim Holstrom [68] comme suit :

1. Création aléatoire de deux nœuds connectés par un lien dont l'âge vaut zéro.
2. Génération d'un vecteur x selon une distribution préalable.
3. Localisation des nœuds s et t les plus proches de x tel que $\|w_s - x\|^2$, $\|w_t - x\|^2$ sont les deux plus petites valeurs.
4. La modification de l'erreur du nœud s est donnée par:

$$erreur_s \leftarrow erreur_s + \|w_s - x\|^2 \quad (2.26)$$

5. Le neurone s et ses voisins topologiques se déplacent vers x en proportion de e_w, e_n

$$w_s \leftarrow w_s + e_w(x - w_s) \quad (2.27)$$

$$w_n \leftarrow w_n + e_n(x - w_n), \forall n \in \text{voisinage}(s) \quad (2.28)$$

6. Incrémenter l'âge des connexions liant s à un autre nœud.
7. Si s et t sont connectés alors l'âge de cette connexion doit être remis à zéro. Sinon, création d'une connexion entre s et t .
Supprimer toutes les connexions dont $l'age > age_{max}$
8. Après chaque λ itération, trouver le neurone u avec l'erreur max. Ainsi, u doit localiser le neurone v parmi ses voisins possédant l'erreur max.

Insertion d'un nouveau neurone r entre u et v :

$$w_r \leftarrow \frac{w_u + w_v}{2} \quad (2.29)$$

9. Mise à jour de l'erreur de tous les nœuds :

D'un point de vue interprétation, l'objectif de la mise à jour de l'erreur du gagnant est de favoriser le neurone s aux prochains stimuli. De plus, lors de la mise à jour du vecteur prototype du gagnant, ce dernier sera translaté avec un mouvement linéaire. Ainsi, l'insertion d'un nouveau nœud contribuera à la minimisation de l'erreur.

2.10.3. Variantes du GNG

A titre d'exemples, nous citons quelques variantes du GNG. Certaines ont été conçues pour améliorer les performances d'apprentissage. D'autres ont été développées pour répondre à quelques limitations du GNG.

❖ Bernd Fritzke a proposé une autre forme du GNG bien adaptée aux distributions non stationnaires de données [68]. Ce modèle connu sous le nom de GNG avec utilité (GNG-U) ajoute une autre variable locale (utilité) mesurant la variation de l'erreur. En fait, une grande utilité exprime que l'augmentation de l'erreur est importante alors qu'une faible utilité désigne un changement négligeable de l'erreur. En admettant cette terminologie, il

convient alors de se demander sur l'initialisation de la variable utilité. Malheureusement, cette étape reste purement expérimentale puisqu'elle n'a pas été précisée par Bernd Fritzke. La fonctionnalité de base du GNG-U est semblable à celle du GNG avec introduction des étapes suivantes :

Après mise à jour de l'erreur, l'adaptation de l'utilité doit respecter l'équation suivante:

$$u_s \leftarrow u_s + (\text{erreur}_i - \text{erreur}_s) \quad (2.30)$$

Suppression des nœuds dont l'utilité soit faible :

$$\frac{\text{erreur}_i}{u_i} > K \quad (\text{erreur}_j; \text{erreur}_{\max}) \quad (2.31)$$

Lors de l'insertion d'un nouveau nœud r entre u et v , l'assignation d'une utilité à r est donnée par l'expression:

$$u_r \leftarrow \frac{(u_u + u_v)}{2} \quad (2.32)$$

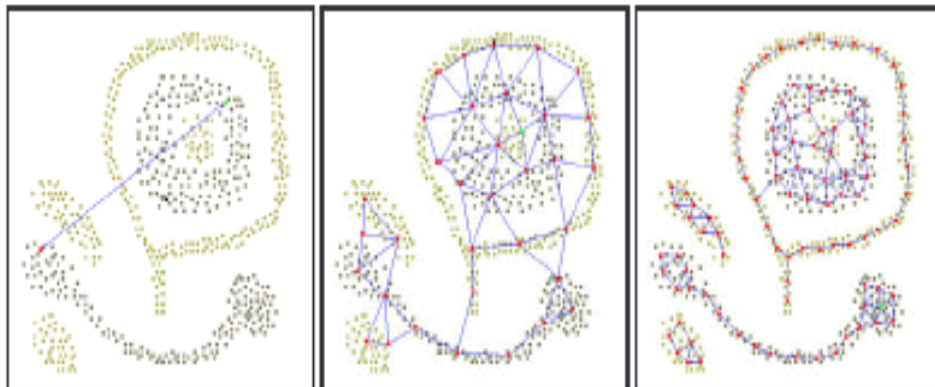


Figure 2.7. : Evolution du GNG [68].

❖ Les réseaux de neurones interviennent dans la classification et la reconnaissance de formes. L'une des catégories des réseaux de neurones multicouches est connue sous l'appellation réseau RBF 'Radial Basis Function'. Ainsi, on considère qu'un réseau RBF est constitué de trois couches : une couche d'entrée, cachée et de sortie. Chaque neurone de la couche cachée est muni d'une variable σ_j décrivant un champ à l'emplacement j . Toutefois, l'activation d'un nœud dépend de sa distance par rapport à l'entrée actuelle. Afin de générer la couche cachée, les réseaux RBF ont tiré profit des avantages du GNG supervisé 'SGNG' [65]. Dans ce contexte, le pseudo-code du GNG a été adapté comme suit [68]:

L'adaptation de l'erreur a été modifiée selon l'équation suivante:

$$erreur_s \leftarrow erreur_s + \|d - y\|^2 \quad (2.33)$$

Tel que \bar{d} est le résultat souhaité et \bar{y} est la réponse obtenue.

Après la mise à jour des vecteurs prototypes, le neurone j qui a été translaté doit adapter sa variable σ_j :

$$\sigma_j = mean\|w_n - w_k\| \quad \forall n \in \text{voisinage}(j) \quad (2.34)$$

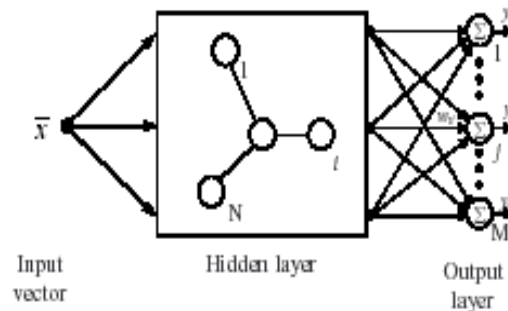


Figure 2.8. : Réseau RBF [68].

❖ Récemment, L.Lella et I.Licat [69] ont publié un article intitulé ‘Evolutionary Neural Gas’ (ENG). Comme nous expliquons les auteurs, l’objectif de l’introduction du ENG est la construction d’un réseau intelligent basé non seulement sur l’apprentissage mais aussi sur l’évolution. L’idée de cette méthode comprend la définition des nœuds du graphe comme étant une population dont l’évolution est régit par la création et l’élimination des connexions. En plus des paramètres manipulés par le GNG, ENG associe à chaque nœud une variable d représentant la distance par rapport à un stimulus d’entrée. Initialement, d doit contenir une valeur maximum. Si la distance par rapport à un stimulus est inférieure à d , la distance ainsi calculée deviendra la nouvelle valeur de d . Le principe de base du EGN est le suivant :

1. Sélection des centres les plus proches d’une entrée.
2. Mise à jour des vecteurs prototypes des centres ainsi calculés :

$$w(t+1) = w(t) + \alpha(x - w(t)) \quad (2.35)$$

$$w_i(t+1) = w_i(t) + \beta(x - w_i(t)) \quad \forall i \in \text{voisinage} \quad (2.36)$$

3. Evolution de la population :

Si l’unité i peut rencontrer une unité j gagnante selon une probabilité pr alors :

$$w_i \leftarrow \frac{w_i + w_j}{2} \quad (2.37)$$

$$w_i \leftarrow \frac{w_i + w_j}{2} \quad (2.38)$$

Sinon exécution de la règle d'insertion du GNG classique
Fin Si

❖ Dans le but de démontrer la stabilité du réseau lors d'un apprentissage incrémental, la technique décrite dans [70] distingue deux types de nœuds : jeune et vieux. Sans toutefois rentrer dans le détail de cette technique, l'une des différences vis-à-vis du GNG est le test établi par l'unité gagnante à la suite de la présentation d'un signal ζ :

$$distance(\zeta, w) \leq \sigma \quad (2.39)$$

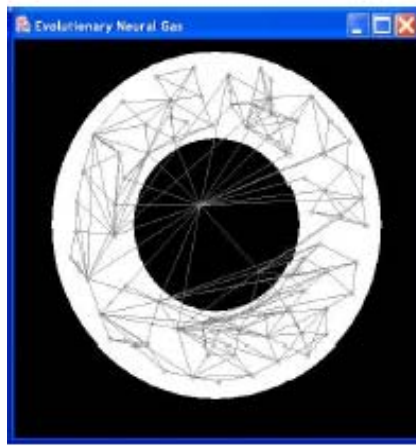


Figure 2.9.a. : Résultat du ENG [69].



Figure 2.9.b. GNG classique [69].

Si cette condition n'est pas vérifiée, l'unité gagnante ajoute un nouveau nœud jeune et affecte à son vecteur prototype w la valeur de ζ . Dans le cas contraire, on poursuit les mêmes étapes du GNG (modification des vecteurs prototypes, ...). Il est aussi à constater que si l'âge dépasse un certain seuil, l'unité correspondante passe de l'état jeune à l'état vieux.

❖ Afin de synthétiser l'information produite par le GNG, une méthode de classification hiérarchique basée sur le GNG a été conçue par I-P.Ivrissimtzis [71]. Sachant que le GNG standard s'exécute à partir d'un graphe minimal de deux nœuds, Tree Growing Neural Gas (TGNG) initialise un pointeur à ce graphe. A chaque fois que le graphe établi par le GNG se divise, les pointeurs du TGNG sont mis à jour. De même, la fusion de deux sous graphes permet la création d'un nouveau nœud dans le TGNG. Ce nœud représente le fils des sous graphes générés.

❖ Si la plupart des versions du GNG ont été développées suite aux inconvénients des cartes auto organisatrices, D.Deng et N.Kababov [72] ont suggéré et validé les performances en terme de temps d'une nouvelle technique 'Evolving Self organizing-map' ESOM. Le résultat de cette technique constitue une hybridation des cartes auto

organisatrices et du GNG. Ne définissant pas une carte préalable, ESOM diffère du GNG au moment de l'insertion d'un nouveau nœud qui n'est pas régularisée à chaque λ itération. Cependant, l'insertion a lieu chaque fois que tous les nœuds possèdent une distance supérieure à un seuil prédéfini. Le principe de cette méthode est le suivant :

- Présenter un vecteur x
- Si $\|w_i - x\| \geq \xi \quad \forall i$ alors

Insérer un nouveau nœud $w_{n+1} = x$

Sinon

Adaptation du vecteur prototype du gagnant et ses voisins

$$\Delta w = \gamma e^{-d^2(w,x)/2\delta^2} (x - w) \quad (2.40)$$

Fin Si

- Mise à jour des connexions.

2.10.4. Applications du GNG

Les domaines d'application du GNG sont globalement variés. Il peut s'agir de la robotique ou l'imagerie. Plus particulièrement, le GNG est une méthode de classification qui a été utilisée avec succès.

❖ D.Vasquez, T.Fraichard, O.Aycard et V.Laugier [73] ont proposé une méthode de prédiction des mouvements d'un ensemble de robots décidant de leurs trajectoires sur la base de la planification. Pour cela, l'approche mise en œuvre, pouvant établir un plan en temps réel, se conforme à deux étapes : la modélisation du système et la prédiction pour estimer l'état futur. En revanche, la modélisation du système est réalisée à l'aide des chaînes de Markov associées à un ensemble d'observations $O = \{O_1, O_2, \dots, O_n\}$. Chaque observation O_t représente la position à l'instant t . Elle est accompagnée d'une variable binaire pour modéliser la fin de la trajectoire. Dans ce cadre, deux types du GNG ont été définis afin d'estimer les paramètres du modèle (position, nombre d'états).



Figure 2.10.a : Exemple d'un environnement [73]. Figure 2.10.b : Planification par le GNG [73].

❖ La démarche développée par M.A Montes de Oca, L.Garrido et J.Aguirre [74] consiste en l'hybridation des colonies de fourmis et le GNG. L'objectif est la classification d'un ensemble d'objets pouvant être déplacés par des fourmis. L'information obtenue par l'une des fourmis sert à l'entraînement de son GNG. La règle

de classification autorise le déplacement d'un objet considéré dissimilaire par rapport aux objets de son voisinage. Le dépôt d'un objet tient compte de la dissimilarité entre les objets. Plus précisément, lors de la classification d'un objet, la fourmi applique le GNG en s'orientant vers le nœud le plus proche du stimulus. L'inconvénient majeur de cette méthode est le recours à plusieurs paramètres.

❖ Pour tenter de synthétiser des images rapidement, I-P.Ivrissimtzis, W-K.Jeong, H-P.Seidel [75] ont entrepris de réaliser cette opération par le GNG. Leur approche a démontré des résultats satisfaisants en dépit des courbes convexes. Grâce à l'algorithme proposé, deux types de nœuds sont à distinguer : actifs et non actifs. Cela est intrinsèquement lié à un certain paramètre prédéfini. Le principe est le suivant :

1. Echantillonnage de l'espace d'entrée.
2. Sélection d'un point.
3. Identification du nœud le plus proche du point considéré. Modifier sa position ainsi que la position de ses voisins.
4. Après un nombre préalable d'itérations, les nœuds actifs se divisent selon un processus spécifique. Les nœuds inactifs sont supprimés.



Figure 2.11. : Images synthétisées par le GNG [75].

❖ Se définissant comme étant le processus de réduction du nombre de couleurs dans une image, la quantification est l'un des processus les plus répondu pour la compression d'images. Les méthodes de quantification peuvent être fixes ou adaptatives. Globalement, le principe des méthodes de quantification consiste à appliquer deux étapes : génération d'une carte de couleurs représentatives puis la transformation de chaque couleur en fonction de la carte obtenue. Le principe des méthodes adaptatives se résume par le partitionnement de l'image en clusters. Le centre de chaque cluster contribuera à la construction de la carte. D.N-Kasabov [76] a motivé le recours à l'algorithme ESOM pour la quantification par le temps d'exécution et la préservation de la topologie. Si I dénote une image couleur, chaque vecteur d'entrée est de la forme $I_i=(R_i,G_i,B_i)$. En adaptions l'algorithme ESOM, le processus de quantification est vu comme une fonction :

$$Q: I_i \rightarrow C_m \quad (2.41)$$

Tel que $C=\{C_j/ j=1\dots 256\}$ est la carte obtenue.

Par rapport aux autres méthodes, la qualité de quantification est si importante. Elle a été calculée en fonction de l'erreur et la variance de quantification :

$$\zeta = \left[\frac{1}{N} \sum_{i=1}^N d(I_i, C_m)^2 \right]^{\frac{1}{2}} \quad (2.42)$$

$$\delta = \left[\frac{\left(\sum_i \|I_i - C_m\| - \zeta \right)^2}{N} \right]^{\frac{1}{2}} \quad (2.43)$$

❖ A la suite du développement de l'algorithme du GNG, plusieurs méthodes exploitant l'efficacité du GNG ont été élaborées. G.Westermann [77] a proposé et validé une approche qui se fonde sur la construction d'un réseau de neurone afin d'apprendre le passé des verbes en Anglais. En effet, cette technique est issue des inconvénients des modèles anciens fonctionnant sur l'hypothèse d'une architecture fixe. Typiquement, le réseau construit est une version du SGNG et possède évidemment certaines qualités. Pour cela, il produit une réponse par rapport à l'identité du verbe et non pas par rapport à sa structure en distinguons par exemple le passé du verbe 'hear' et 'fear' par 'heard' et 'feared'. L'apprentissage par le SGNG initialise deux neurones cachés, déplace le gagnant et ses voisins, propage l'activation de l'unité gagnante pour une entrée et enfin insère un nouveau neurone. Chaque unité est considérée comme un champ dans le sens où elle sera active pour quelques unités. En terme de performance, à partir d'un corpus de 24 802 dont 57,2% sont des verbes réguliers et 42,8% sont des verbes irréguliers, les ressources utilisées sont d'autant plus minimales que celles comparés à d'autres méthodes.

❖ L'enjeu de la méthode décrite dans [78] est l'apprentissage du comportement réactif des robots mobiles par l'utilisation des réseaux RBF intégrant un apprentissage supervisé et non supervisé. Le travail réalisé sur ce thème est motivé par l'apprentissage des comportements réactifs non connus au moment de la programmation. Il s'agit d'obtenir un comportement d'un robot en passant par deux étapes : une étape de prétraitement de l'information capteur par un apprentissage non supervisé et une autre d'exécution du comportement par un apprentissage supervisé. De façon synthétique, l'apprentissage des informations capteurs est vu comme une fonction d'approximation $y=F(x)$ sur la base d'un ensemble d'exemples d'apprentissage $(x_i, y_i, i=1..m)^T$

$$x_i = (x_{i1}, x_{i2}, \dots, x_{im})^T \quad (2.44)$$

$$y_i = (y_{i1}, y_{i2}, \dots, y_{im})^T \quad (2.45)$$

La classification des exemples (x_i, y_i) a été menée par un réseau RBF dont la couche cachée est créée par le GNG. L'avantage du GNG est la possibilité d'adapter sa structure afin d'apprendre de nouvelles tâches.

2.10.5. Avantages et inconvénients du GNG

Evidemment, le GNG tire ses avantages du principe de l'apprentissage non supervisé, incrémental et compétitif. En effet, le GNG possède les qualités suivantes:

- Méthode simple contenant globalement une étape d'adaptation, de mise à jour de l'âge et d'insertion d'un nœud.
- Il a démontré sa capacité à considérer une masse importante de données.
- Une méthode visuelle pouvant tirer profit des techniques d'analyse appliquées aux cas des cartes auto organisatrices.
- Par analogie aux méthodes basées sur un graphe, l'âge des connexions permet d'étudier facilement la similarité entre les nœuds impliqués.

Mais comme toute technique d'apprentissage, le GNG contient les limitations suivantes:

- Nombre considérable de paramètres influant directement sur le résultat.
- Une méthode qui n'a pas été testée pour différents types de données.
- Le nœud favorisé aux prochains stimuli peut rendre le mécanisme de compétition non équitable entre des nœuds importants.
- Repose sur des formules très restrictives pour le mouvement des nœuds et l'insertion. Cela pose dès fois des questions de la conservation de la topologie de l'espace d'entrée.

2.11. Discussion

Après avoir expliqué la classification de données, l'objectif de ce travail est désormais l'exploitation du calcul amorphe présenté dans le chapitre 1. Comme le succès d'une méthode basée calcul amorphe nécessite quelques conditions, cette partie a détaillé le GNG qui n'est qu'une instance des cartes auto organisatrices. A ce point, on pourra poser la question qui porte sur les motivations du GNG dans un contexte du calcul amorphe. Mais voilà le tableau 1 qui résume des aspects théoriques du GNG lui permettant d'effectuer approximativement une classification dans un environnement de calcul amorphe. En plus, Le tableau 1 dresse une comparaison entre le GNG et SOM.

Cependant, l'originalité du calcul amorphe lui fait un nouveau paradigme de calcul et l'application du GNG se heurte notamment aux constatations suivantes:

- Les nœuds du GNG sont dynamiques.
- Communication directe entre les nœuds.
- Voisinage constitué de l'ensemble des nœuds connectés par des connexions au nœud considéré.

SOM	GNG
<ul style="list-style-type: none"> • Méthode d'apprentissage locale. • Constitue un système parallèle. • Fixe la structure de la carte. • La notion d'auto organisation y est fortement inscrite. • Voisinage variable. • Fixe le nombre de clusters. • Dépend d'un ensemble de neurones de faible capacité. 	<ul style="list-style-type: none"> • Adaptation dans un voisinage. • Les nœuds opèrent en parallèle. • Aucune structure préalable de la carte. • S'appuie sur l'auto organisation du graphe généré. • Voisinage fixe. • Aucun choix préalable du nombre des clusters. En fait, le GNG bénéficie des avantages des techniques hiérarchiques et non-hiérarchiques. • Approprié pour des unités de faible capacité en présentant à chaque instant un vecteur d'entrée.

Tableau 2.1: Comparaison entre SOM et le GNG.

2.12. Analyse de la méthode de classification

Sans aller jusqu'aux extrêmes de toutes les approches de classification proposées dans la littérature, il est clair que chacune d'elles est accompagnée de quelques inconvénients. Cependant l'évaluation d'une bonne technique de classification doit prendre en compte les critères suivants [79] :

- La taille de données manipulées doit être logiquement importante et les performances doivent décroître linéairement en proportion du volume des données.
- Les objets à regrouper sont de différents types (numériques, booléens, catégories, ...).
- Paramètres d'entrée minimaux.
- Robustesse en dépit du bruit.
- Indépendance de l'ordre des données et leur dimension.
- La possibilité de découvrir différents clusters ayant différentes formes.

Malgré l'existence d'une taxonomie de méthodes de classification, l'émergence de nouveaux systèmes informatiques imposent la mise en œuvre d'une technique appropriée. Sa validation est faisable grâce aux critères mathématiques suivants :

- Soit $\{v_1, v_2, \dots, v_n\}$ l'ensemble des clusters produits, $\{u_1, u_2, \dots, u_n\}$ les classes réelles, n_{ij} le nombre d'objets appartenant à u_i et v_j , (u_i, v_i) le nombre d'objets de la classe u_i et v_i respectivement. Le critère R s'évaluant entre 0 et 1 est décrit par cette formule :

$$R = \frac{\sum_{i,j} \binom{n_j}{2} - [\sum_i \binom{n_i}{2} \sum_j \binom{n_j}{2}] / \binom{n}{2}}{\frac{1}{2} [\sum_i \binom{n_i}{2} + \sum_j \binom{n_j}{2}] - [\sum_i \binom{n_i}{2} \sum_j \binom{n_j}{2}] / \binom{n}{2}} \quad (2.46)$$

- Etend donné une matrice (n,m) où n est la taille de la population et m le nombre de mesures effectuées. Si $R(x,e)$ est la valeur de x par rapport à la mesure e et $\mu_{C_i}(e)$ est la valeur moyenne correspond au cluster C_i par rapport à la mesure e , la mesure de qualité du cluster C_i est défini par le critère suivant :

$$F(e,k) = \sqrt{\frac{1}{2} \sum_{i=1}^k \sum_{x \in C_i} (R_i(x,e) - \mu_{C_i}(e))^2} \quad (2.47)$$

La procédure est répétée pour toutes les mesures et plus F est petit plus le cluster est de bonne qualité.

3.13. Conclusion

La classification apparaît fréquemment lors d'analyse de données. Ce chapitre a montré que plusieurs choix possibles existent quant à la classification de données. Nous avons pu constater que toute méthode de classification, ancienne ou nouvelle, combine quelques critères de : localité, hiérarchisation, ... Ainsi, Toutes les méthodes partagent de multiples avantages : domaines d'application variés, abstraction de données, ... Si l'on s'interroge sur la technique appropriée, divers indices sont disponibles. Nous pouvons cité la nature de données et les préférences utilisateur. Historiquement, les méthodes non-hiérarchiques sont reconnues par leur capacité à partitionner l'espace d'entrée en une seule étape. L'exemple typique de méthodes non-hiérarchiques est celui du K-means. Suite à l'apparition des réseaux de neurones, de nombreuses typologies ont vu le jour. Les cartes auto organisatrices sont fondées sur l'auto organisation. Elles peuvent apprendre la topologie d'un espace d'entrée en fixant aussi bien la structure de la carte que le nombre de neurones. A la fin d'apprentissage, les vecteurs d'entrés similaires se trouvent affecter à des neurones adjacents. Néanmoins, quelques limitations de SOM ont été soulevées. Par rapport à la typologie des méthodes de classification, le GNG est une méthode locale, exacte, dynamique et basée sur un graphe. En effet, le GNG est une méthode bénéficiant des avantages des techniques d'apprentissage incrémental en évitant le choix préalable de la structure de la carte et le nombre des neurones. Pourtant le GNG a donné naissance à une autre catégorie de réseaux de neurones (RBF), notre objectif a porté sur l'explication approfondie du GNG et les motivations de base seront éclairées dans le chapitre 3.

3.1. Introduction

Dans l'univers médical, biologique, aérien et géographique, l'exigence du traitement d'images est une contrainte réelle. En permettant l'extraction de régions et/ou contours, la segmentation d'images est l'un des processus de bas niveau. Auparavant, les premières alternatives de segmentation se sont consacrées à l'application d'une détection locale pour extraire les bords. Il faut dire aussi qu'on a largement bénéficié des techniques de division/fusion afin de mener une segmentation par région. Toutefois, ces dernières années, les avancés en matière des outils électroniques et des méthodes informatiques ont permis l'émergence d'une panoplie de méthodes de segmentation. Quant aux nouvelles approches de segmentation par contour, les contours actifs constituent l'une des méthodes tirant profit des avantages des modèles déformables et pouvant converger vers une solution. En parallèle, les champs de Markov, ont été adoptés pour des images médicales entachées d'imperfection. Néanmoins, les diagrammes de Voronoi apparaissent comme des structures géométriques importantes dont l'exploitation en segmentation d'images a donné de bons résultats. Actuellement, la nouvelle tendance consiste à aborder le problème de segmentation par le mécanisme d'émergence en s'appuyant sur les systèmes multi agents. Leur puissance réside dans la résolution d'un problème de façon distribuée et intelligente. Par ailleurs, la segmentation peut être évoquée par le mécanisme de classification des pixels. A l'instar des cartes auto organisatrices, qui procurent les avantages d'un apprentissage incrémental, l'attribution des pixels aux classes correspondantes est liée aux paramètres pixels et au critère de similarité défini. L'objectif de la première partie de ce chapitre est de dresser un état de l'art concernant les techniques de segmentation d'images. La seconde partie est destinée à l'introduction d'une approche de segmentation basée calcul amorphe.

3.2. Segmentation d'images

Technique exercée durant tout processus d'analyse d'images, la segmentation est une opération relativement ancienne qui est, jusqu'à présent, fonction des images manipulées. Informellement, la segmentation se définit comme étant un traitement qui distingue les objets d'intérêts afin de parvenir à les isoler du fond ou des autres objets. En d'autres termes, la segmentation est la stratégie adoptée par un système informatique cherchant à comprendre une image (ou éventuellement une séquence d'images) et de construire une information plus riche que l'ensemble des pixels isolés. A la fin de la segmentation, chaque pixel se voit attribuer une région et chaque région correspond à une partie sémantique de l'image. Les discontinuités entre les régions sont les contours des objets. Réciproquement, un ensemble de régions R_1, R_2, \dots, R_n est une segmentation d'une image R [80] si :

$$\bigcup_{i=1}^n R_i = R, \quad \forall i=1 \dots n \quad (3.1.a)$$

$$R_i \cap R_k = \Phi, \quad \forall i \neq k \quad (3.1.b)$$

$$R_i \text{ est connexe, } \forall i=1 \dots n \quad (3.1.c)$$

$$\text{Il y a un predicat } P \quad P(R_i) = \text{vrai}, \forall i=1..n \quad (3.1.d)$$

$$P(R_i \cup R_k) = \text{faux} \quad \forall i \neq k \text{ et } R_i \text{ et } R_k \text{ sont deux régions adjacentes} \quad (3.1.e)$$

En effet, la segmentation d'images n'est pas une simple opération. Lors de son achèvement, toute région doit être homogène (couleur, moyenne, écart type) et la fusion de deux régions n'est pas possible. Cependant, les nombreux travaux réalisés dans ce domaine ont donné lieu à plusieurs méthodes, aux fortunes diverses, en fonction de la philosophie, paradigme et le système utilisé.

3.2.1. Méthodes de segmentation

La complexité est une propriété accompagnant toute méthode de segmentation. Le nombre de pixels et l'espace de représentation des valeurs attribuées aux pixels sont des problèmes de fond durant la segmentation d'images. Face à cette situation, il était essentiel de mettre en œuvre une stratégie de segmentation d'images. De même, toutes les techniques de segmentation sont structurées selon les stratégies suivantes :

3.2.1.1. Segmentation par seuillage

Ce type de méthodes requiert d'abord la transformation d'une image à un histogramme. Le résultat final modélise la fréquence d'apparition des niveaux de gris (couleur). Si l'on arrive à déterminer des seuils correctement, la segmentation pourra avoir lieu. Il s'agit d'affecter chacun des pixels à sa classe correspondante. Evidemment, le problème réside dans la définition des seuils appropriés [81]. Pour cela, les deux techniques suivantes ont été testées. La première ne tient pas compte des relations spatiales, tandis que la deuxième, qui est moins performante, prend en charge les relations spatiales.

Détection des vallées : son principe se résume par la détection des $(m-1)$ minima à partir de l'histogramme H . Si $m, m+1$ sont les valeurs moyennes des deux classes, s_i est un seuil si :

$$H(s_i) = \text{Min}[H(k)]_{k=]m, m+1[} \quad (3.2)$$

Maximisation de l'entropie : grâce à cette méthode, le choix d'un seuil est renforcé par la quantité d'information dans chaque classe. En disposant d'un nombre préalable de classes, l'entropie totale est :

$$E(C) = \sum_i E(C_i) \quad (3.3)$$

Tel que :

$$E(C_i) = - \sum_j (p_j \log_2 p_j) \quad (3.4)$$

p_j est la probabilité à posteriori du niveau de gris j . Elle est obtenue par :

$$P_j = \frac{H(j)}{\text{taille de l'image}} \quad (3.5)$$

Reconnues par une qualité de segmentation intrinsèquement liée aux seuils fixés, les méthodes de seuillage ne sont pas pleinement utilisées.

3.2.1.2. Segmentation par contour

Typiquement, les contours sont les lignes frontières des objets [82]. Du fait que l'image est considérée comme une fonction, l'analyse de la dérivée première et seconde est un outil mathématique très puissant pour extraire les contours. Dans cette optique, la différentielle de la fonction image est :

$$dz = \frac{\partial f(x,y)}{\partial x} \cdot dx + \frac{\partial f(x,y)}{\partial y} \cdot dy \quad (3.6)$$

D'où, l'amplitude du gradient et sa direction sont calculés, respectivement, par les formules suivantes :

$$|\nabla f(x,y)| = \sqrt{\left(\frac{\partial f(x,y)}{\partial x}\right)^2 + \left(\frac{\partial f(x,y)}{\partial y}\right)^2} \quad (3.7)$$

$$\phi = \arctang \frac{\frac{\partial f(x,y)}{\partial y}}{\frac{\partial f(x,y)}{\partial x}} \quad (3.8)$$

Plus concrètement, l'approximation de la première dérivée peut être calculée selon les formules (3.9) et (3.10):

$$\frac{\partial f(x,y)}{\partial x} = f(x+1,y) - f(x,y) \quad (3.9)$$

$$\frac{\partial f(x,y)}{\partial y} = f(x,y+1) - f(x,y) \quad (3.10)$$

Segmenter une image par contour revient à appliquer les deux masques de convolution suivants:

$$G_x = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad G_y = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad (3.11)$$

D'un autre point de vue, un contour est caractérisé par le passage à zéro de la deuxième dérivée. Ainsi, la dérivée seconde est:

$$\nabla^2 f = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} \quad (3.12)$$

L'approximation de l'équation (3.12) est la suivante :

$$\nabla^2 f = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4.f(x,y) \quad (3.13)$$

Les masques de convolution destinés au calcul de la seconde dérivé ont été formulés comme suit :

$$L_x = \begin{bmatrix} 1 & 0 & -1 \\ C & 0 & -C \\ 1 & 0 & -1 \end{bmatrix} \quad L_y = \begin{bmatrix} -1 & -C & -1 \\ 0 & 0 & 0 \\ 0 & C & 1 \end{bmatrix} \quad (3.14)$$

Si $C=1$, les masques calculés sont appelés les masques de Sobel.

3.2.2.3. Segmentation par région

Alors que la segmentation par seuillage fonctionne sur la base de l'histogramme, la segmentation par région utilise quelques paramètres (variance, niveau de gris, moyenne, texture) pour identifier les régions similaires [83]. A propos de la segmentation par région, la technique la plus répandue est celle du Split & Merge. Son principe consiste à diviser une image en régions. A chaque étape, les régions non homogènes sont divisées. Dans une deuxième phase, les régions adjacentes satisfaisant ensemble des critères de similarité sont fusionnées. D'autre part, la technique des pyramides des régions quantifie une image par une arborescence de graphes en regroupant les régions voisines déclarées similaires. Tout d'abord, la base de cette pyramide est constituée des pixels isolés et plus on gravit les étages, plus le nombre de régions diminue par fusion. Enfin, à partir du graphe d'adjacence représentant les voisins susceptibles de se fusionner, un processus de décimation des nœuds, produit un nouveau graphe. Le critère d'arrêt dépend de la stabilisation des attributs des nœuds définis.

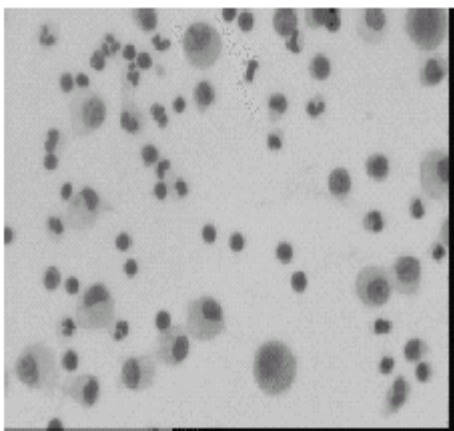


Figure 3.1.a. : Image de cellules [84].

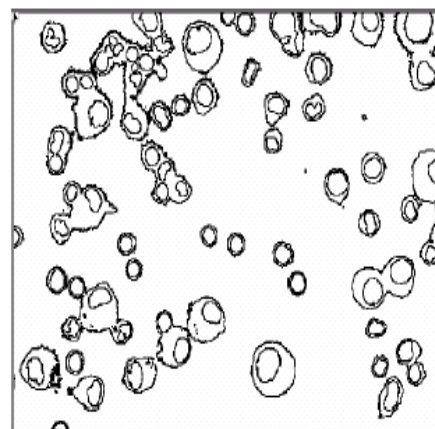


Figure 3.1.b. : Segmentation par seuillage [84].

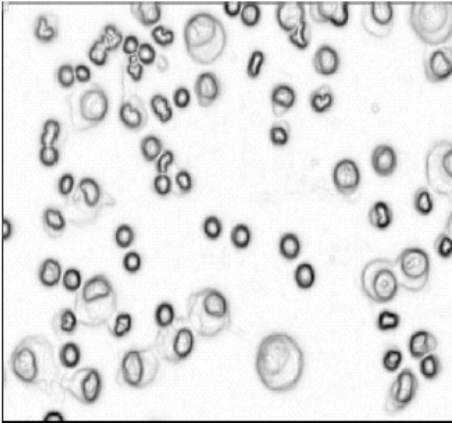


Figure 3.1.c: Segmentation par contour [84].

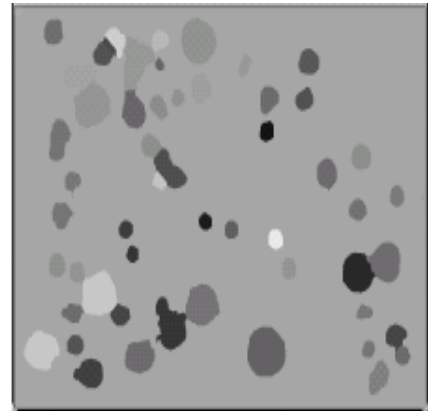


Figure 3.1.d: Segmentation par région [84].

3.2.1.4. Comparaison

Nombreux sont les articles qui proposent des méthodes de segmentation, mais les contraintes en temps de calcul et la qualité de segmentation souhaitable par un utilisateur freine leur application réelle. Les techniques conçues portent différentes appellations, alors que le principe derrière est une segmentation par région et/ou contour. Il faut savoir que la segmentation par seuillage n'a pas été bien formulée. Son exploitation s'est souvent limitée à une pré segmentation. Historiquement, de nouvelles idées en matière de la segmentation par région et/ou contour ont été diffusées. En effet, l'application d'une méthode particulière dépend des préférences utilisateur, le type d'images et les contraintes machine. Dans ce contexte, les caractéristiques générales d'une segmentation par région ou contour sont énumérées dans le tableau suivant :

Segmentation par contour	Segmentation par région
<ul style="list-style-type: none"> • Les fortes variations du gradient ne sont pas appropriées pour détecter les bords des objets présents dans tout type d'images (en particulier les images texturées). • A cause du bruit, les contours obtenus peuvent être ouverts. • Nécessite le stockage de l'image entière. • Malgré que l'extraction des points de contraste puisse être menée en une seule étape, le problème réside dans la formation des lignes contours. • Un simple test suffit quant à lui pour la détermination des points de contraste. • chaque pixel connaît sa configuration dans l'espace. 	<ul style="list-style-type: none"> • L'homogénéité (variance) n'est pas un critère adéquat pour tout type d'images. On s'interroge souvent sur le critère de similarité le plus approprié. • Peut séparer des régions homogènes à cause du bruit. • Nécessite le stockage de l'image entière. • Malgré que le processus soit itératif, quelques étapes peuvent s'exécuter en parallèle. • Nécessite des structures de données importantes. • Chaque pixel connaît sa configuration dans l'espace.

Tableau 3.1. : Comparaison entre segmentation par région et contour.

3.3. Nouvelles méthodes de segmentation

Vu les contraintes machine, le temps de calcul et l'exigence du stockage de l'image entière, on a tenté de proposer de nouvelles idées attrayantes. Inspirés des modèles déformables, les contours actifs remportent un succès grâce à leur convergence dès l'initialisation d'une courbe. A côté de ce modèle, les champs de Markov basés sur une information à priori peuvent adapter l'information de chaque site à la suite de l'interaction avec les sites voisins. Notamment, les diagrammes de Voronoi peuvent séduire les concepteurs par leur simplicité. Aujourd'hui, les travaux suggérés sont liés aux systèmes multi agents. L'objectif est de bénéficier de la puissance de tels systèmes pour la segmentation d'images.

3.3.1. Contours actifs

Les premiers essais de segmentation d'images par contour étaient effectués par l'application des filtres. Cette démarche semble simpliste mais souffre surtout du manque d'un cadre méthodologique. Cette limitation a provoqué de nouvelles versions pour segmenter des images. Par définition, la méthode des contours actifs consiste à faire évoluer un contour fermé initial vers une position d'équilibre représentant les bords d'un objet à détecter [85]. Cette technique issue des modèles déformables, vise l'extraction d'un contour par la mise en évidence d'une courbe fermée au voisinage du contour. La forme de cette courbe ainsi que sa position peuvent être modifiées au cours du processus de déformation. Cela est relativement dû à [86]:

- Energie propre à la courbe assimilée à l'énergie mécanique de tension.
- Energie potentielle imposée par l'image.
- Energie externe traduisant les contraintes spécifiques du problème.

Mathématiquement, un contour actif est une courbe fermée paramétrée C :

$$C:[a,b] \rightarrow \mathbb{R}^2$$

$$S \rightarrow V(s) = \begin{bmatrix} x(S) \\ y(S) \end{bmatrix} \quad (3.15)$$

S est l'abscisse curviligne. a, b sont les extrémités de C .

Pour se déplacer dans le domaine de l'image, la courbe C doit minimiser la fonctionnelle suivante :

$$E(C) = E_{interne(C)} + E_{externe(C)} + E_{image} \quad (3.16)$$

$$E_{interne} = \frac{1}{2} \int_b^a \alpha(S) \left| \frac{\partial C}{\partial S} \right|^2 + \beta(S) \left| \frac{\partial^2 C}{\partial S^2} \right|^2 dS \quad (3.17)$$

$\left| \frac{\partial C}{\partial S} \right|^2$ est un terme qui agit sur la longueur de la courbe en l'empêchant d'être trop élastique et $\left| \frac{\partial^2 C}{\partial S^2} \right|^2$ est un terme qui agit sur la courbure du contour actif. De même, l'énergie de l'image a été définie par l'équation suivante :

$$E_{image} = -\lambda(S) \|\nabla I(V(S))\|^2 \quad (3.18)$$

Le minimum d'énergie est atteint lorsque l'équation d'Euler-Langrange est satisfaite:

$$(-w_1 V') + (w_2 V'') + \nabla P(V) = 0 \quad (3.19)$$

Tel que :

$$P = -\lambda(s) \|\nabla I(V(s))\|^2 \quad (3.20)$$

Pour parvenir à résoudre cette équation, deux solutions sont envisageables: différences finies en remplaçant les dérivées partielles par des différences divisées ou les éléments finis en remplaçant la courbe par des segments élémentaires. Les éléments mécaniques seront calculés sur ces segments. En notons par n le nombre de points contours, A la matrice des dérivées partielles tout au long de C , la formulation finale de la solution de l'équation (3.19) est :

$$C_n = (1 - \gamma A) [C_{n-1} + \gamma E_{externe}(C_{n-1})] \quad (3.21)$$

Malgré la convergence du système décrit par l'équation (3.19), cette méthode fait partie des méthodes semi-interactives extrêmement liées à l'initialisation de la courbe définie qui doit être proche du contour réel. A cela s'ajoute l'inconvénient de la difficulté de détecter les contours de plusieurs objets.

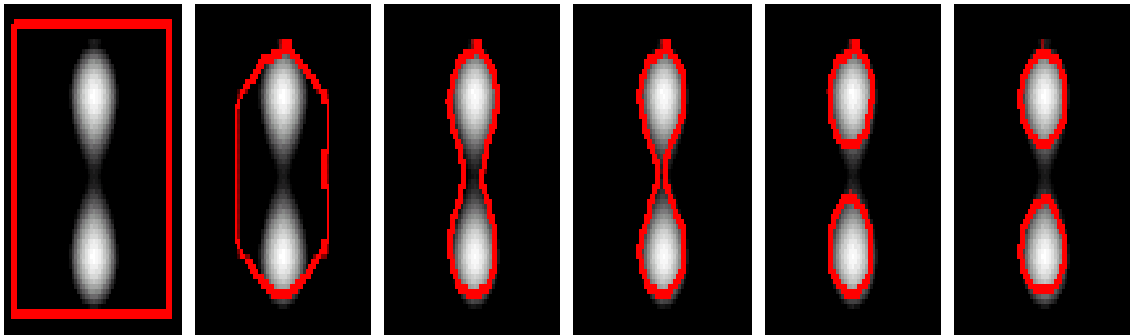


Figure 3.2. : Evolution d'un contour actif [86].

3.3.2. Champ de Markov

Depuis longtemps, la statistique est un outil d'aide à la décision permettant la construction de nouvelles connaissances à partir des faits observés. Pour estimer les paramètres d'un modèle établi en fonction d'un ensemble de variables explicatives, plusieurs alternatives ont été développées. L'une des approches consiste à construire un modèle à priori portant sur le processus à estimer et un modèle ayant conduit à l'observation faite. Ces modèles sont ensuite combinés dans le cadre Markovien [87]. Sans aucun doute, ce formalisme est d'un rôle majeur pour des images IRM bruitées. Globalement, les champs de Markov permettent l'assignation d'un ensemble de sites, munis de relations de voisinage, aux classes correspondantes.

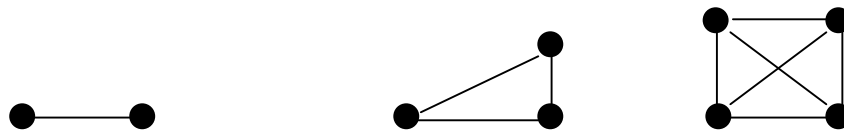


Figure 3.3.: Exemples de relations de voisinage (cliques).

Si V dénote l'ensemble des sites spatialement organisés indexés $\{1, \dots, N\}$ où chaque site s'exprime en deux types de données: l'un observé (intensité) et l'autre caché codant la classe du site, l'objectif est la modification de l'image d'intensité Y et celle des étiquettes Z par des champs aléatoires. Pour cela, une loi à priori de l'ensemble de configurations (étiquettes) $P(Z)$ et une autre d'attache des données $P(Y/Z)$ sont définies. L'attribution des pixels aux classes adéquates peut être menée par la maximisation de la probabilité $P(Z/Y)$ pouvant être calculée par la loi de Bayes :

$$P(Z/Y) = \frac{P(Z) \cdot P(Y/Z)}{P(Y)} = \frac{P(Z) \prod_{i=1, \dots, N} P(Y_i/Z_i)}{P(Y)} \quad (3.22)$$

Au niveau d'images IRM, les champs de Markov ont donné des résultats très satisfaisants en modélisant aussi bien les observations sur les valeurs d'intensité que le bruit. La prise en charge de l'information bruit et l'intensité était l'objet du mémoire [87].

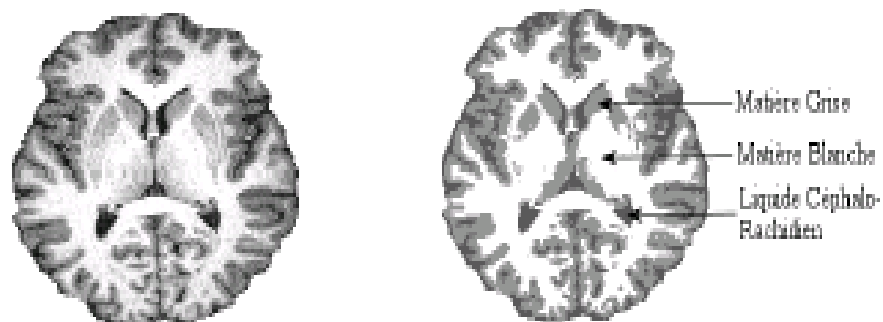


Figure 3.4. : Segmentation par champ de Markov [86].

3.3.3. Méthodes basées sur un graphe

Il est également possible d'envisager la segmentation d'images à partir d'un graphe modélisant des proximités entre les nœuds. Cela permet de partitionner un espace de faible dimension par rapport à l'espace des pixels.

❖ Si la plupart des tentatives de segmentation se sont intéressées à la manipulation des propriétés de l'image, les diagrammes de Voronoi ont bouleversé les techniques de segmentation en proposant la formulation de la segmentation par des structures géométriques importantes. Souvent, on se sert des diagrammes de Voronoi pour partitionner un espace de façon équitable entre un ensemble de points. Le résultat est un graphe attribuant à chaque point l'ensemble des points les plus proches de lui. Dans le cadre de la segmentation d'images, Melkemi a suggéré et validé sa méthode synthétisée comme suit [88]:

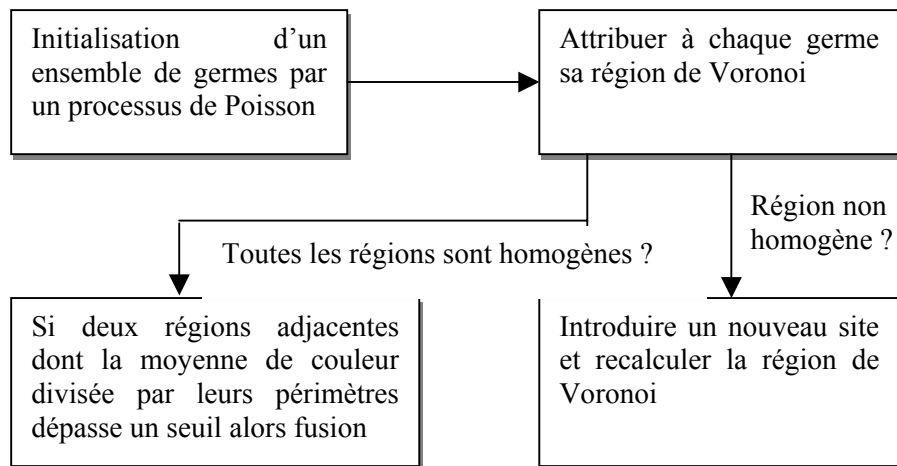


Figure 3.5. : Principe de la méthode de Melkemi.

❖ Par partition de Voronoi généralisée, nous entendons un ensemble de sites et une métrique de distance. Récemment, P.Andrés Arbaléaz et L-D.Cohen [89] ont démontré qu'il est possible de modéliser la segmentation d'images par partitions de Voronoi généralisées. Les étapes de cette méthode sont schématisées en figure (3.6).

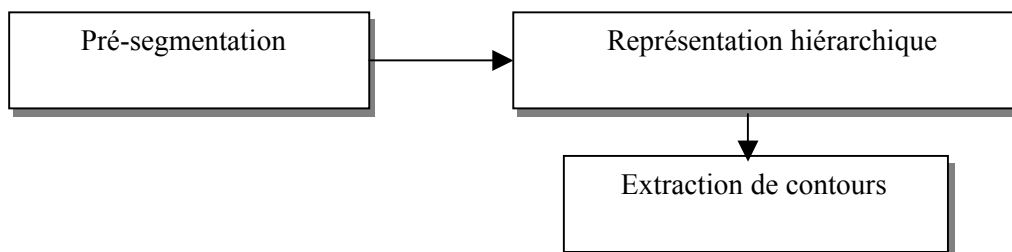


Figure 3.6. : Etapes d'une segmentation par régions de Voronoi généralisées.

En ce qui concerne la première étape, l'ensemble des composantes extrémales de luminosité constitue les points sites. Chaque couleur est assignée à sa région de Voronoi. Pour atteindre le deuxième objectif, une mesure de contraste basée sur la couleur est appliquée :

$$d^c(R_1, R_2) = \frac{\sum \delta^2(p_1, p_2)}{\text{longueur}(\partial R_1 \cup \partial R_2)} \quad (3.23)$$

δ^2 est la distance dans l'espace de couleur. p_1 est un point dans R_1 et p_2 est un point dans R_2 .

Finalement, l'apport de cette technique est nettement visible lors de la troisième étape. Une simple opération de seuillage fournit l'ensemble des bords. P.Andrés Arbaléaz et L-D.Cohen ont mis en évidence le concept de saillance d'un point. C'est le plus grand rayon r , tel que ce point appartient à une frontière de la partition de Voronoi. Quelques résultats expérimentaux sont visualisés en figure (3.7).



Figure 3.7. : Résultat de segmentation par régions de Voronoi généralisées [89].

❖ De nouvelles formes de segmentation apparaissent sans cesse et formalisent ce problème par des outils intéressants. C'est vrai qu'il est facile de comprendre le problème de segmentation mais c'est la qualité du résultat obtenu qui fait la différence entre les techniques proposées. Devenue aujourd'hui une référence, la méthode de J. Shi et J. Malik [90] développée en 2000 donne plus d'importance à la structure globale de l'image et ramène le problème de segmentation à la recherche des valeurs propres d'un produit matriciel. En effet, elle considère une image comme un graphe non orienté et sa segmentation comme un processus de partitionnement dans un graphe $G = (V, E)$. Implicitement, V est l'ensemble des pixels et E est l'ensemble d'arrêtes pondérées reliant chaque pair et mesurant la similarité (différence d'intensité ou du niveau de gris) entre les nœuds correspondants. L'objectif de cette idée est de partitionner le graphe dans les endroits de faible poids.

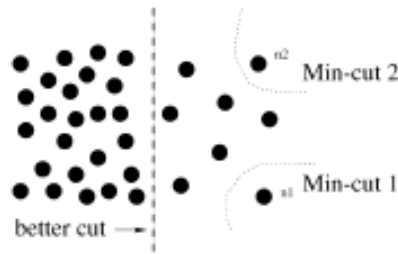


Figure 3.8. : Principe de la méthode Ncut[90].

Pour calculer la similarité entre deux partitions A, B , la formule (3.24) se base sur les poids des arrêtes reliant les éléments de A et B :

$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v) \quad (3.24)$$

Tel que $w(u, v)$ est le poids de la connexion (u, v) .

Il s'agit de minimiser cette valeur pour partitionner le graphe. Habituellement, on prend en compte les relations entre la partition A et les autres partitions :

$$N_{cut}(A, B) = \frac{cut(A, B)}{ass(A, V)} + \frac{cut(A, B)}{ass(A, V)} \quad (3.25)$$

Où $ass(u, v)$ est la somme des poids entre A et les autres partitions. La solution pour minimiser cette valeur est sa mise sous forme matricielle. Soient les notations suivantes :

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \quad \text{Tel que} \quad \begin{cases} x_i = 1 & \text{si } i \in A \\ x_i = -1 & \text{si } i \in B \end{cases} \quad (3.26)$$

D une matrice diagonale où chaque élément est la somme des w par rapport à la colonne j . La solution qui a été décrite et démontrée est la suivante :

$$\min_{Ncut} = \min_y y^T \frac{(D-W)y}{Y^T.Dy} \quad (3.27)$$

Cela revient à chercher les valeurs propres de la matrice $D^{\frac{1}{2}}(D-w)D^{\frac{1}{2}}$.

3.3.4. Systèmes multi agents

Si pour résoudre un problème complexe, les méthodes anciennes avaient recours à des mécanismes compliqués, l'apparition des systèmes multi agents s'est appuyée sur des règles simples pour appréhender un problème. L'idée derrière consiste en l'utilisation des principes d'auto organisation, auto reproduction, d'apprentissage et d'adaptation lors d'un comportement. Les travaux modernes sur les SMAs ont prouvé leur puissance dans le domaine d'imagerie et en particulier pour la segmentation d'images. En ce sens, l'environnement associé aux agents est l'image elle-même. Durant sa vie, chaque agent se retrouve rattaché à un pixel. L'objectif est l'exploration de l'image pour réunir sous un même label les pixels possédant des caractéristiques similaires.

❖ Dans son approche, Liu [91] initialise des agents, autonomes, distribués aléatoirement et communiquant localement. Suivant le type de l'image à segmenter, un certain nombre de classes est défini. Par exemple, dans le cas d'image IRM d'un cerveau, les classes à distinguer sont la matière blanche, grise, le contour et la tumeur. Si un agent constate que son voisinage est homogène selon les mesures suivantes:

$$\mu = \frac{1}{N} \sum I(K,i) \quad (3.28)$$

$$\sigma = \sqrt{\frac{1}{N} \sum (I - \mu)^2} \quad (3.29)$$

il fait étiqueter le pixel correspondant, reproduit un fils et devient inactif. De même, le fils considéré doit explorer l'environnement en suivant la direction de son père. Dans le cas contraire, l'agent doit se déplacer pour en chercher une zone homogène.

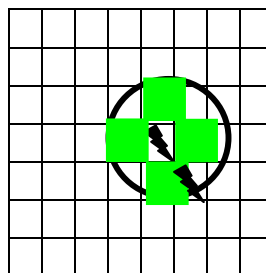


Figure 3.9. Comportement d'un agent dont la région est homogène.

❖ Le rôle de la plate forme mise en œuvre par H.Settache [92] est si important. Elle peut être considérée comme l'intégration des outils de segmentation proposés séparément pour la segmentation région et contour en permettant un parallélisme entre des agents réactifs. Non seulement, elle associe aux différents agents (région, contour) des règles simples, mais elle repose aussi sur les pyramides irrégulières pour effectuer une segmentation spatiale. L'utilisation de cette plate forme passe par trois étapes. La pré-segmentation région par Quadtree est l'étape initiale qui attribue à chaque agent région un identificateur, la somme des niveaux de gris, la surface de sa région, la moyenne, l'écart type et les relations de voisinage. Pour extraire les contours, l'application du filtre de Sobel suivi d'un seuillage est une solution qui a servis d'avantage au processus de coopération. Ainsi, à chaque contour est affecté un agent contour caractérisé par son identificateur, la largeur du contour et un Booléen indiquant si le contour est fermé ou ouvert. D'ailleurs, le point important de tout système multi agent est l'explication des règles d'interaction qui se substituent à une coopération entre les agents contours et régions. Les six règles, exécutées récursivement, sont les suivantes:

- Coopération région-région pour la fusion des régions : il doit exister un désir mutuel de fusion entre les agents régions correspondants.
- Coopération contour-région: pour la fusion des régions il ne doit pas y exister un contour entre les deux régions.
- Coopération contour-contour pour le raccordement des contours.
- Coopération région-contour : une région traversée par plusieurs contours peut les inciter à raccorder.
- Coopération région-contour : pour le prolongement des contours à condition que les régions voisines ne s'y opposent pas.

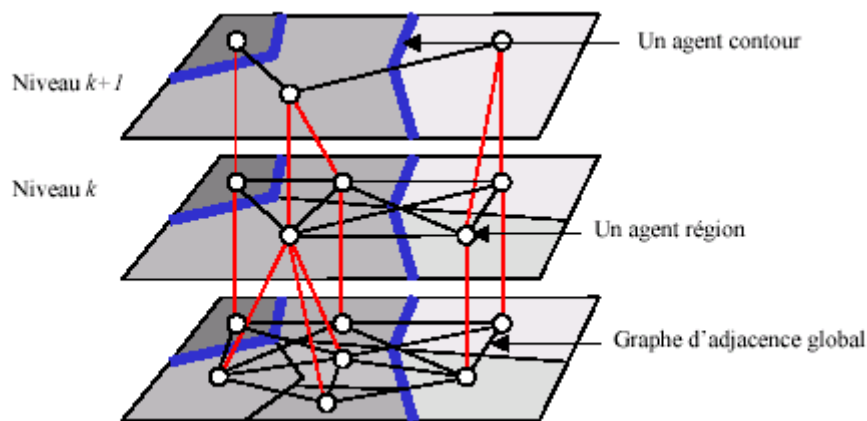


Figure 3.10.: Principe de la construction d'une pyramide par un SMA [92].

❖ Sachant que les systèmes multi agents tirent leur fonctionnement de métaphores biologiques et sociales, C.Bourjot et V.Chevrier [93] ont conçu une nouvelle méthode basée sur des agents réactifs. L'approche ainsi développée représente une déviation par rapport aux anciennes méthodes utilisant des mécanismes inspirés du comportement des fourmis. En revanche, elle s'inspire des interactions des araignées. Ces insectes vivent

ensemble et partagent la même toile. Globalement, leur comportement collectif consiste à construire la toile en respectant des règles architecturales, la prise de décision lors du transport d'une proie, et la synchronisation lors de la prédation. Vis-à-vis d'un système multi agent, l'environnement est une grille carrée, où chaque case correspond à un piquet pouvant être relié par des fils. Pour explorer l'espace, chaque agent (araignée) fixe un fil dans une case aléatoire et se comporte selon les règles suivantes associées elles-mêmes d'une pondération:

- Allez à un piquet adjacent.
- Allez à un piquet en suivant le fil que l'araignée a tissé.
- Allez à un piquet suivant le fil des congénères.

A partir de cette description simpliste du comportement des araignées, la formulation du problème de segmentation a bénéficié d'un nombre réduit de règles. En plus, chaque piquet correspond à un ou plusieurs pixels et possède une hauteur représentant le niveau du gris. Intuitivement, le nombre initial d'araignées, ayant une perception locale, est un paramètre important. Plus profondément, chaque araignée se déplace en fonction de la hauteur du piquet actuel.

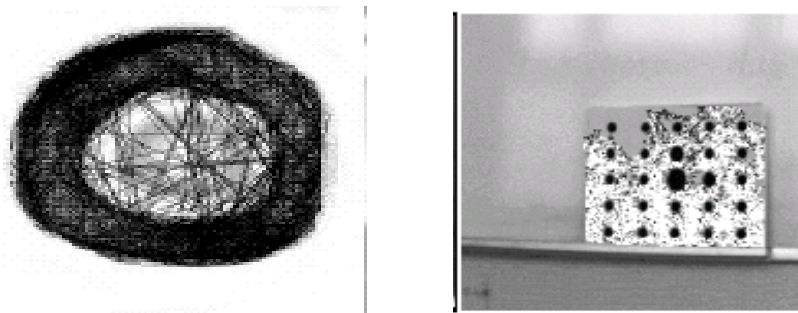


Figure 3.11. : Résultats de segmentation par l'approche des araignées [93].

3.4. Segmentation par classification

Face aux problèmes posés par les méthodes de segmentation, il est apparu urgent d'extraire les objets pertinents d'une image de façon rapide. Pour parvenir à cette fin, il semble intuitif de tirer profit du K-means. Plus récemment, plusieurs articles ont présenté des méthodes sophistiquées s'inspirant des cartes auto-organisatrices.

❖ Typiquement, la méthode du K-means a connu un succès en dépit de son application à différents types de données (biologiques, numériques) et même dans le cadre de l'imagerie. En dotant du principe de la classification non hiérarchique, chaque pixel est attribué au centre le plus proche. Par analogie à l'algorithme présenté dans le chapitre précédent, le principe de la classification des pixels par K-means consiste en l'application des étapes suivantes [94]:

- Initialisation : chaque pixel est affecté à une partition aléatoire.

- A chaque itération, les centres des différents groupes sont recalculés et chaque pixel est affecté de nouveau à un groupe en fonction du centre le plus proche.
- Arrêt si convergence.

❖ Un compromis entre un apprentissage supervisé et non supervisé est un outil presque idéal pour la classification de données. A.Moussaoui, K.Benmahammed, N.Ferahta et C.Chen [95] ont développé une méthode originale faisant appel à l'algorithme FCM. L'intégration d'un terme impliquant un apprentissage supervisé constitue le noyau de cette méthode. Ayant prouvé une bonne qualité de segmentation d'images IRM, cette technique est schématisée en figure (3.12). A partir des modes de l'histogramme et l'estimation de la fonction de densité de probabilité, le nombre de cluster est déterminé. Il est aussi optimisé.

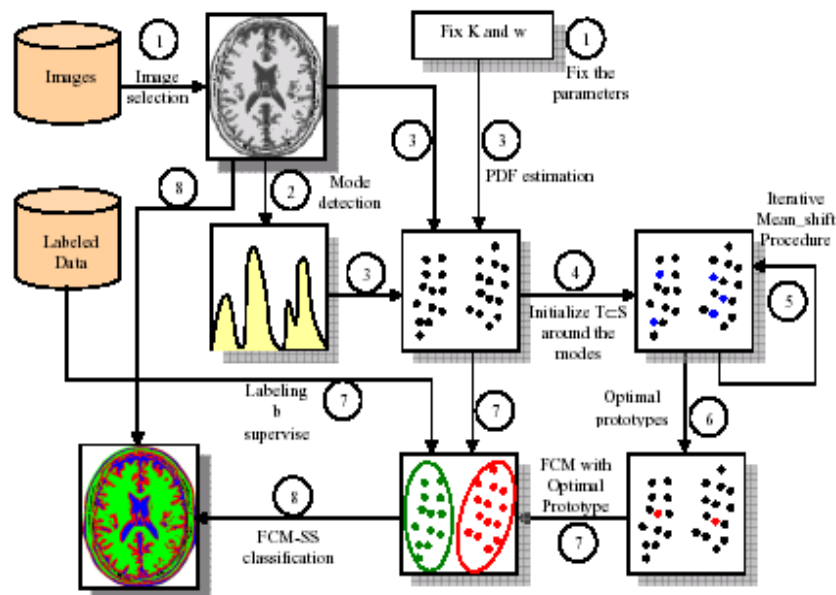


Figure 3.12. : Principe de la méthode de Moussaoui [95]

❖ Certaines techniques de segmentation se prennent pour des images d'un type particulier et ne s'attèlent pas à la tâche de segmentation requise pour d'autres situations. A partir de cette motivation, la méthode proposée par C. Charrier et O. Lezoray [96] est issue des techniques hybrides de segmentation. Elle fonctionne par classification des pixels en fusionnant les résultats initiaux de segmentation. Ainsi, chacune des cartes initiales est associée d'une mesure de confiance représentant la qualité de segmentation. Cette démarche a été validée pour des images couleur en fusionnant trois cartes. Chacune est le résultat de segmentation dans un plan de couleur. Dans l'étape intersection des images segmentées, un pixel localisé à (x,y) peut être classé de la même façon ou différemment :

Si sa classe est déterminée sans ambiguïté **alors**
Créer une nouvelle classe

Sinon

Utiliser la technique de Dempster-Shaffer pour attribuer ce pixel à l'élément le plus probable

Fin Si

❖ Si la classification floue vient de concurrencer les méthodes exactes de classification, N.Bouloudani, P.Lambert et D.Coquin [97] ont présenté une approche améliorant le résultat de segmentation du FCM. L'idée consiste à ajouter une classification hiérarchique. Avec cette technique, la fusion de deux classes est liée à leur compacité. L'intérêt est qu'un seul paramètre α est largement suffisant pour pratiquer cette méthode. En exigeant un nombre préalable de classes, la première étape consiste à procéder par une classification FCM. Puis vient le processus d'agglomération pour restreindre le résultat à un nombre limité de grandes régions. Pour cela, les deux concepts suivants ont été proposés :

L'isolation I_{ij} entre deux classes C_i, C_j est :

$$I_{ij} = \left(\sum_{n=1 \dots N} \min(U_{in}, U_{jn}) / \sum_{n=1 \dots N} \max(U_{in}, U_{jn}) \right) \quad (3.30)$$

Plus I_{ij} est proche de 0, plus les classes sont isolées. La compacité d'une classe K composée, elle-même de l'union de p classes est :

$$CP_k = \frac{1}{p} \sum_{i=1 \dots N} \min(I_{ij}) \quad (3.31)$$

Si les classes sont isolées, CP_k est proche de 0. La règle de fusion est la suivante :

Si $I_{ij} \leq \alpha \cdot CP_{ij}$, alors

Ne pas fusionner les classes i, j

Sinon fusion des classes i, j

❖ Dans la logique de la méthode du K-means, les centres des classes sont initialisés aléatoirement. Mais, ils sont recalculés à chaque itération. S.Benelmadjat [98] a repris dans son rapport une méthode de classification spatio-calorimétrique qu'elle a intégré avec le système EDISON(edge detection and image segmentationOn). Ce dernier propose une interface pour le traitement d'images. A l'opposé du K-means, cette classification n'affecte aucun pixel à sa classe qu'après avoir identifier les centres adéquats. En effet, la classification spatio-calorimétrique caractérise une région comme étant un sous-ensemble de pixels fortement connexes et ayant des couleurs homogènes. A cet égard, le degré de compacité quantifie simultanément l'arrangement spatial des pixels et la dispersion des points couleur. Il est défini par :

$$D(S) = DC(S) \times DH(S) \quad (3.32)$$

Tel que, le degré de connexité d'un sous espace S est donné par :

$$DC(S) = \frac{\sum \gamma(P)}{car\{P \in S\}} \quad (3.33)$$

γ Est le degré de chaque pixel et DH est le degré d'homogénéité qui dépend de l'écart ou la mesure de dispersion. Pour désigner les centres des espaces, un domaine candidat ayant un degré de compacité élevé est choisit et le domaine le plus éloigné de cet espace est éliminé.

3.5. Discussion

Le cadre le plus en mesure de juger les méthodes de segmentation est probablement la comparaison dressée en fonction des critères critiques. les approches multi agents et la classification par cartes auto organisatrices sont effectivement deux modèles qui coïncident avec Le principe d'auto organisation qui ont su se justifier par des résultats étonnants. En terme d'un système multi agents, segmenter une image revient à définir une population d'agents ayant une perception, souvent locale, et des compétences de calcul (couleur, homogénéité,...). Les règles de base sont, en général, des déplacements (en laissant une trace) ou une production d'un nouveau congénère. Par ailleurs, la classification par une carte auto organisatrice, représentant un espace de faible dimension par rapport à l'espace des pixels, préserve la topologie de l'image présentée. Plus particulièrement, le tableau suivant illustre quelques éléments de fond, qui démontrent l'utilité de ces deux approches.

Cartes auto organisatrices	SMAs
<ul style="list-style-type: none"> •L'auto organisation est une caractéristique de base lors de leurs utilisations. •Règles simples d'adaptation •Nombre fixe d'unités de calcul. •Les unités sont similaires. 	<ul style="list-style-type: none"> •Peuvent émerger une organisation reflétant les régions et/ou les contours d'une image. •Règles plus simples que celles des cartes auto-organisatrices. Elles sont souvent d'inspiration biologique ou écologique. •Peuvent nécessiter ou non le nombre de classes. •L'organisation peut être décomposée en plusieurs agents ayant des compétences différentes dont la coopération est la clé du succès.

Tableau 3.2.: Comparaison entre SOM et SMA.

3.6. Segmentation par calcul amorphe

Sachant que les performances d'une application sont à l'image du système informatique utilisé, divers systèmes distribués ont été spécifiquement conçus pour offrir une puissance de stockage ou de calcul. Hormis les systèmes distribués existants, un système de calcul amorphe possède la qualité d'abstraction de l'environnement et une fiabilité accrue. Evidemment, ces caractéristiques sont en adéquation avec la problématique de classification d'une masse importante de données. Mais compte tenu des contraintes d'un système de calcul amorphe, le choix et l'adaptation de l'une des méthodes de classification proposées dans la littérature impose quelque d'implications. En d'autres termes, la capacité mémoire des particules de calcul amorphe, leur distribution irrégulière et la contrainte d'interaction locale par diffusion sont des facteurs qui laissent prévoir les constatations suivantes:

- La méthode de classification proposée dans le contexte du calcul amorphe doit être incrémentale en évitant le stockage de toutes les données.
- Le processus de classification doit être local.
- La technique adoptée ne doit pas imposer une structure préalable pour la classification de données.
- Les particules de calcul amorphe sont statiques.

Partant de ces hypothèses, nous avons pris le choix de l'application de l'algorithme du GNG dans le cadre de la classification de données. En fait, le GNG ne suppose aucune organisation préalable, opère par apprentissage incrémental, associe à chaque nœud ses voisins topologiques et s'appuie sur des règles d'apprentissage simples en manipulant l'erreur, la distance et l'âge des connexions. De son côté, le GNG reste une approche classique et son adaptation dans un contexte de calcul amorphe doit prendre en charge les points suivants [99]:

- Comment peut-on organiser la tâche de localisation de la particule possédant la distance minimum ou l'erreur maximum ?
- La création des connexions entre les particules de calcul amorphe exige l'information concernant leurs coordonnées.
- La sélection d'une particule satisfaisant l'équation (2.27), (2.28) ou (2.29) n'est pas toujours possible.

Avec un système de calcul amorphe, ces questions sont procurées comme suit :

- La construction d'une hiérarchie de calcul amorphe est une solution très simple et en même temps d'un intérêt important pour organiser une telle tâche.
- Quant il s'agit d'attribuer des coordonnées aux particules de calcul amorphe, la méthode de C.Frost peut contribuer à assigner des coordonnées relatives aux particules de calcul amorphe.
- L'adaptation des équations (2.27), (2.28) et (2.29) a été menée comme suit :

$$w_s \leftarrow w_s + e_v(x - w_s) \pm \delta \quad (3.34)$$

$$w_n \leftarrow w_n + e_n(x - w_n) \pm \delta \quad (3.35)$$

$$w_r \leftarrow \frac{w_u + w_r}{2} \pm \delta \quad (3.36)$$

δ a été expérimentalement fixé entre 0 et 1. L'objectif de l'équation (3.38) et (3.39) est d'autoriser un mouvement non linéaire.

Pour cela, on propose le pseudo-code suivant qui est une réécriture du GNG [99] classique en terme d'interactions entre particules de calcul amorphe. Ainsi, chaque particule i exécute son programme de façon asynchrone :

Début

Cycle : Nombre d'itérations
etat_i : Variable booléenne. Vaut vrai si la particule est la racine de la hiérarchie.
queue-msg_i : Une file pour stoker les messages reçus.
s_i, t_i, u₁, u₂, v_i : Variables réelles.
active_i : Booléen. Vaut vrai si la particule est incorporée dans le graphe généré.
x : Vecteur d'entrée.
erreur_i : L'accumulation d'erreur.

Répéter

Diffuser un vecteur x par l'une des particules de calcul amorphe

Diffuser ('La distance', $\|w_i - x\|^2$)

Si *etat_i*=vrai alors

$s_i \leftarrow \min(\text{queue-msg}_i)$

Diffuser ('Distance min', s_i)

Fin Si

Si *active_i*=vrai & $\|w_i - x\|^2 = s_i$ alors

Appliquer l'équation (2.26)

Choisir une autre particule non active h selon l'équation (3.34)

active_h \leftarrow vrai

active_i \leftarrow faux

Pour chaque $n \in \text{voisinage}(i)$ **faire**

Choisir une autre particule non active h' suivant l'équation (3.35)

active_{h'} \leftarrow vrai

active_n \leftarrow faux

Fin Pour

Incrémenter l'âge de tous les liens dans le voisinage

Fin Si

Si cycle mod $\lambda = 0$ alors

Diffuser ('L'erreur', *erreur_i*)

```

Si (etati=vrai) alors
   $u_{i1} \leftarrow \max(\text{queue-msg}_i)$ 
  Diffuser ('L'erreur max1',  $u_{i1}$ )
Fin Si

Si activei=vrai & erreuri= $u_{i1}$  alors
  Pour chaque  $n \in \text{voisinage}(i)$  faire
    Diffuser ('L'erreur', erreuri)
  Fin Pour
   $u_{i2} \leftarrow \max(\text{queue-msg}_i)$ 
  Diffuser ('L'erreur max2',  $u_{i2}$ )
Fin Si

Si activei=vrai & erreuri= $u_{i2}$  alors
  Choisir une autre particule  $r$  selon l'équation (3.36)
Fin Si
Fin Si
Cycle  $\leftarrow$  Cycle +1
Jusqu'à un critère d'arrêt

```

A ce stade, la segmentation d'images par calcul amorphe a été conduite par l'algorithme décrit ci-dessus. Elle considère un très grand nombre de particules de calcul amorphe dont la classification reflète la topologie de l'image impliquée. Cependant, les paramètres fixés sont mentionnés dans le tableau suivant:

e_w	e_n	α	β	a_{\max}	m	λ	δ
0.25	0.23	0.75	0.6	20	4	2	[0,0.8]

Tableau 3.3. : Les paramètres considérés.

Il est à noter que l'incorporation d'une nouvelle particule de calcul amorphe est réalisée après chaque deux itérations. Le souci est d'augmenter la probabilité de trouver une particule satisfaisant l'équation (3.36).

3.6.1. Pourquoi Netlogo ?

Considéré comme étant le successeur de Startlogo, cet environnement propose un éventail de modèles pour la simulation des systèmes complexes (biologiques, physiques, chimiques). Chaque modèle est composé de quelques règles simples permettant l'émergence d'un comportement global. Cela rend Netlogo un instrument particulier. En plus, Netlogo distingue trois catégories d'agents :

- Turtles : peuvent être statiques ou dynamiques.
- Patches : sont des agents statiques.

Observer : un agent particulier qui joue le rôle d'observateur.

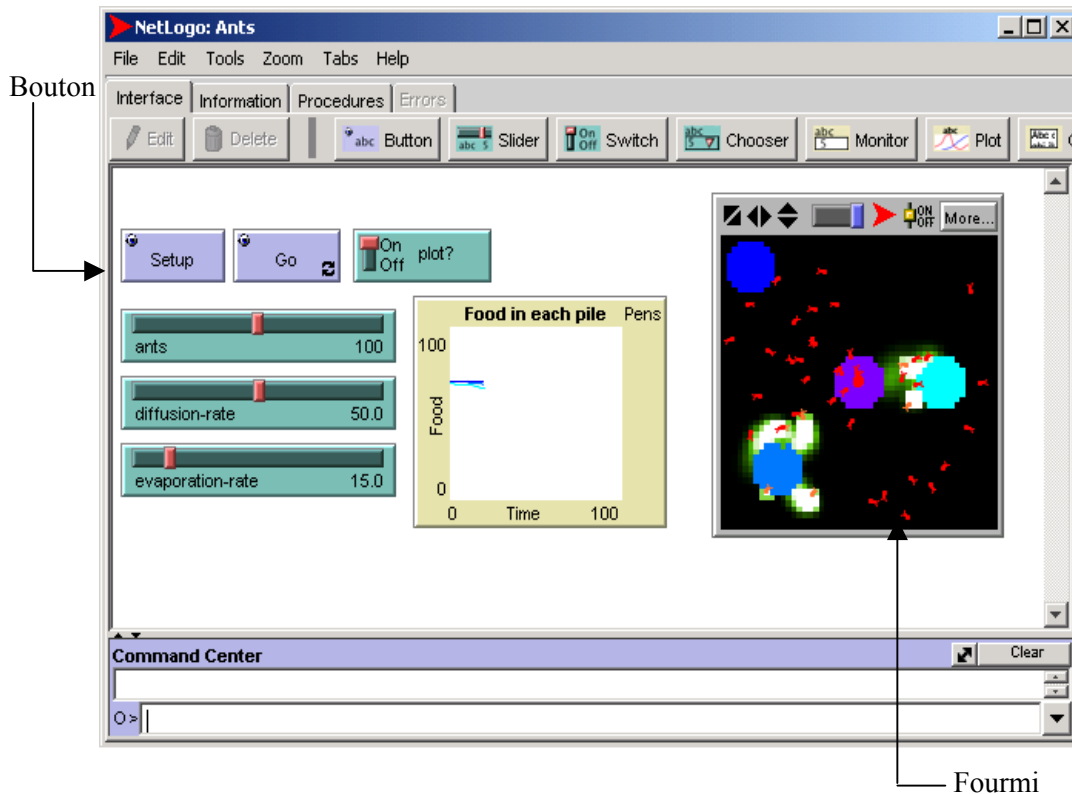


Figure 3.13. : Fenêtre de Netlogo.

Côté programmation, Netlogo offre diverses primitives pour commander aussi bien les fourmis que les patches. Plus spécifiquement, il permet le chargement d'images PGM (portable grey map).

Par ailleurs, plusieurs difficultés peuvent sembler affecter la mise en œuvre d'une approche basée calcul amorphe. Parmi ces difficultés figure le choix d'un environnement de simulation approprié. A cet égard, nous avons trouvé plusieurs éléments de réponse apportés par la plate forme Netlogo :

- Permet la définition d'une multitude d'agents.
- Permet l'exécution asynchrone des commandes.
- Permet des interactions locales.
- Repose sur des règles simples pour l'émergence d'un comportement.
- On peut disposer d'agents positionnés irrégulièrement.

Signalons la nécessité de masquer quelques variables associées aux agents. L'objectif est de respecter les contraintes du calcul amorphe.

3.6.2. Résultats de la segmentation par calcul amorphe

Cette approche a été expérimentée pour la segmentation d'images par contour et par région. Initialement, chaque particule calcule l'intensité correspondant à sa position. Son

vecteur prototype w contient ses coordonnées relatives. Dans le contexte de la segmentation par contour, une particule contribuera à la génération d'un stimulus d'entrée x si l'une des particules dans son voisinage ait une valeur différente. Par exemple, l'initialisation de deux particules connectées par un lien en vert, donnera lieu à quelques connexions après dix itérations.

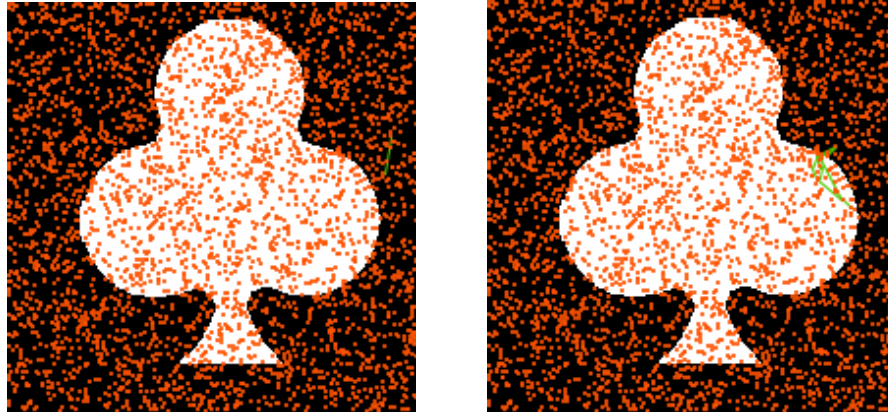


Figure 3.14.a : Distribution initiale des particules de calcul amorphe. Figure 3.14.b : Quelques connexions en vert après dix itération [99].

Après le processus d'adaptation du graphe, le réseau final obtenu est visualisé en figure (3.15.c).

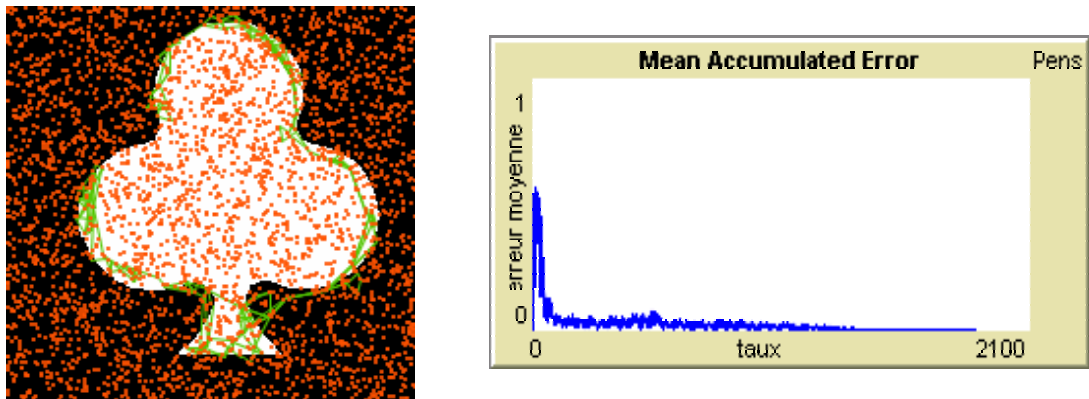


Figure 3.14.c : Le graphe obtenu après 2000 itérations. Figure 3.14.d : Le graphe de l'erreur.

Comme toute technique d'apprentissage, il était question de vérifier la minimisation de l'erreur d'apprentissage durant le processus de classification. Evidemment, la figure (3.14.d) illustre le décroissement de l'erreur qui a atteint 0.0025 après 2000 itérations. Pour mettre au point une segmentation par région, l'image de synthèse démontrée en figure (3.15.a) a été utilisée pour tester l'approche ainsi proposée. Comme le visualise la figure (3.15.d), deux grands clusters sont obtenus, chacun correspondant à un carré.

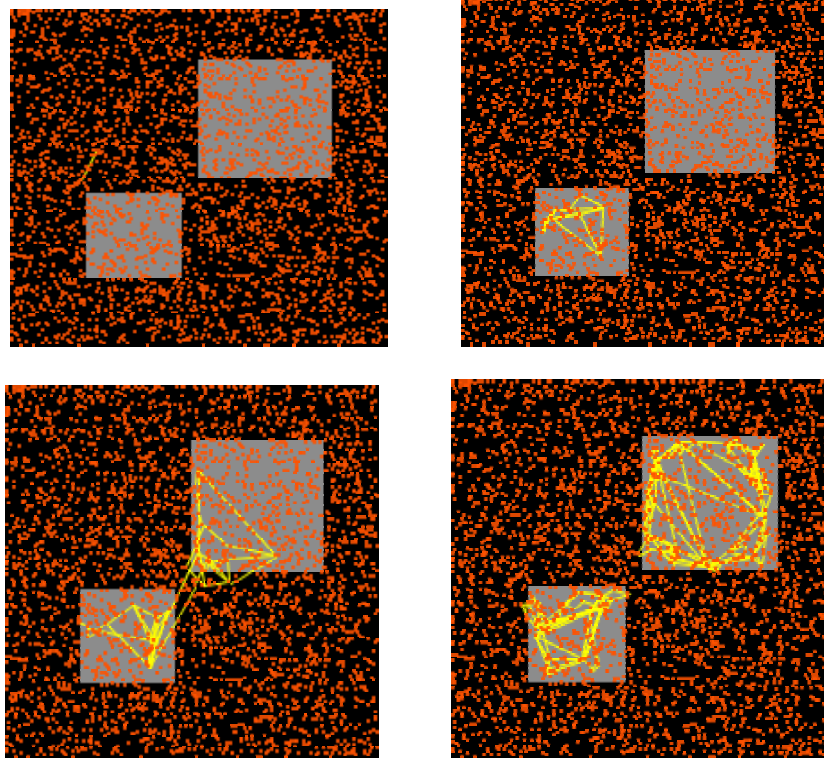


Figure 3.15. :Etapes de segmentation par région

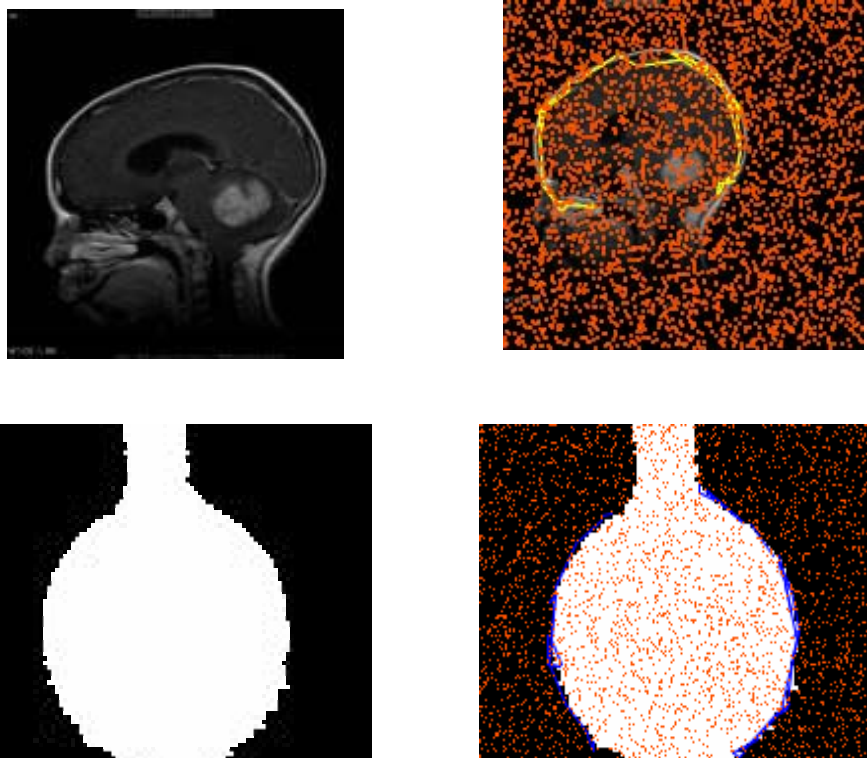


Figure 3.16. : Quelques images de test et leurs résultats de segmentation par contour.

3.6.3. Calcul de la complexité

L'une des méthodes les plus intuitives pour comparer deux algorithmes consiste à calculer la complexité.

Cas du GNG classique :

Soit N le nombre de neurones, initialisé à 2. Au fur et à mesure, l'ajout de nouveaux neurones permet d'apprendre la topologie de l'espace d'entrée.

K : Nombre de voisins topologiques d'un neurone.

M : Nombre d'itérations.

Le cas extrême $\lambda=1$

N opérations de test pour localiser le neurone ayant la distance minimum par rapport à un stimulus.

1 opération d'adaptation de l'erreur.

1 opération d'adaptation du poids.

K opérations d'adaptation du poids.

1 opération de création d'un neurone.

2 opérations de mise à jour de l'erreur.

N opérations de mise à jour de l'erreur.

Au total :

$(2K+2N+5)$ opérations.

Par rapport à M itérations, le nombre d'opérations coûteuses est :

$M (2K+2N+5)$.

Puisque $K \leq N$, on estime que la complexité est d'ordre :

$$O(MN)$$

Cas du GNG proposé :

N : Nombre de particules de calcul amorphe.

N_i : Nombre de leaders de niveau i .

K : Nombre de niveaux dans la hiérarchie de calcul amorphe.

R : Rayon de communication.

Le cas extrême (évaluation-insertion)

$$\text{Diffuser } \left(\|w_i - x\|^2 \right)$$

Si $etat_i=vrai$ **alors**

$s_i \leftarrow \min(queue-msg_i)$

Diffuser (s_i)

Fin Si

$N + \sum_{i=1}^{k-1} N_i$ opérations de diffusion

N opérations de test

$1 + \sum_{i=1}^{k-1} N_i$ opérations de diffusion

1 affectation

Au total :

$2 + 2 \sum_{i=1}^{k-1} N_i + 2N$

Si $active_i=vrai$ & $\|w_i-x\|^2=s_i$ **alors**

Appliquer l'équation (2.26)

Choisir une autre particule non active h selon l'équation (3.34)

$active_h \leftarrow vrai$

$active_i \leftarrow faux$

Pour chaque $n \in voisinage(i)$ **faire**

Choisir une autre particule non active h' suivant l'équation (3.35)

$active_{h'} \leftarrow vrai$

$active_i \leftarrow faux$

Fin Pour

Incrémenter l'age de tous les liens dans le voisinage

Fin Si

N tests

R diffusions

1 diffusion

R tests

1 affectation

Au total :

$N + (2R+2) \times R$

Si $cycle \bmod \lambda = 0$ **alors**

Diffuser ($erreur_i$)

Si ($etat_i=vrai$) **alors**

$u_i \leftarrow \max(queue-msg_i)$

Diffuser (u_i)

Fin Si

$N + \sum_{i=1}^{k-1} N_i$ diffusions

N tests

1 affectation

$1 + \sum_{i=1}^{k-1} N_i$ diffusions

Au total :

$2N + 2 + 2 \sum_{i=1}^{k-1} N_i$ opérations

Si $active_i = vrai$ & $erreur_i = u_i$ **alors**

Choisir une autre particule r selon l'équation (3.36)

Fin Si

Fin Si

N test

R diffusions

1 diffusion

N opérations de mise à jour

Au total :

$2N + R + 1$ opérations

D'où la complexité totale est :

$$N + (2R + 2) \times R + 2 + 2 \sum_{i=1}^{k-1} N_i + 2N + 2N + 2 + 2 \sum_{i=1}^{k-1} N_i + 2N + R + 1$$

Par rapport à M itérations, la complexité est d'ordre

$$O(MN)$$

Elle est approximativement égale à la complexité dans le cas du GNG classique.

3.7. Conclusion

Elles sont le cœur de tout système d'analyse d'images. Les méthodes de segmentation d'images constituent une solution ancienne pour déterminer les objets pertinents. Une multitude de méthodes de segmentation a été décrite algorithmiquement par le biais d'outils mathématiques simples en proposant une segmentation par contour, région ou coopération contour/région. Ce chapitre a montré la grande utilité des nouvelles

méthodes de segmentation. Nous constatons à travers les techniques présentées succinctement, la qualité de segmentation qui reste restreinte à un type particulier d'images. Néanmoins, les SMAs ouvrent une nouvelle voie de recherche par l'émergence des régions ou contours d'une image. Ces systèmes font l'objet de plusieurs techniques marquées par l'utilisation de règles simples. Si l'on s'intéresse à la segmentation d'images comme étant un processus de classification, l'algorithme du K-means et les cartes auto organisatrices ont montré leurs capacités à attribuer chaque pixel à sa classe correspondante. Même en se basant sur des estimateurs flous, la classification par l'une des méthodes non hiérarchiques bénéficie d'un temps d'exécution raisonnable. Dans la partie 1 de ce chapitre, nous avons montré la relation entre la segmentation et la classification. Dans la deuxième partie, nous avons conçu une contribution de segmentation d'images par calcul amorphe. Malgré que les particules de calcul amorphe respectent une liste exhaustive de contraintes, leur classification a été facilement réalisée en adaptant l'algorithme du GNG. Cette méthode laisse entrevoir des résultats satisfaisants en cas d'un système de calcul amorphe. Par rapport aux méthodes classiques de segmentation, la technique proposée a porté sur la segmentation d'un espace de faible dimension vis-à-vis de l'espace des pixels.

4.1. Introduction

La nanotechnologie est devenue l'une des branches les plus actives. Son impact sur la démarche d'investigation en biologie moléculaire est déjà considérable par l'avènement des biopuces. L'étude des biopuces est passionnante par la richesse des solutions développées tels que les laboratoires sur puce, les puces à cellule et les puces à ADN. Lors de la conception des puces à ADN, plusieurs ARN peuvent être analysés simultanément. Exploitant judicieusement cette technologie requiert d'abord la construction de la puce. Pour cela, des milliers de sondes sont déposées sur une lame en verre. La deuxième étape recommande l'extraction de l'ARN de deux types cellulaires : inconnu et de témoin. Partant de là, les deux types cellulaires sont marqués par deux fluoréscents puis hybridés. Enfin, l'analyse des images scanner et le prétraitement des valeurs d'intensité obtenues jouent un rôle prépondérant pour l'identification du niveau d'expression des gènes. Notamment, les puces à ADN peuvent être employées différemment selon la perspective dans laquelle on se place : détection d'une maladie, prédiction de la fonction d'un nouveau gène... Malgré que le recours aux puces à ADN soit une décision lourde de conséquences, l'apport technologique est très bénéfique. Cependant, la manipulation efficace d'un tableau gigantesque (gène, condition) impose souvent la classification en regroupant les profils d'expression qui soient similaires par rapport à un critère de similarité. Actuellement, la collaboration des informaticiens, électroniciens et biologistes a conduit à l'analyse de données issues des puces à ADN en s'intéressant à une classification supervisée ou non supervisée. Le rôle des méthodes de classification exposées dans le chapitre 2 (k-means, SOM, la classification hiérarchiques, ...) est si important qu'elles ont abouti à des résultats encourageants dans le cadre des puces à ADN. Néanmoins, il est devenu nécessaire d'envisager de nouvelles solutions pour analyser des expressions de gènes complexes. Connaissons la puissance d'un système de calcul amorphe, l'idée développée consiste à regrouper les gènes similaires par un système de calcul amorphe en appliquant l'algorithme du GNG. La première partie de ce chapitre est une explication des termes biologiques. On trouve aussi, les étapes de conception d'une puce à ADN et les méthodes de classification déjà utilisées afin de rapprocher la compréhension de la méthode proposée. La seconde partie contient une description d'une nouvelle approche de classification par calcul amorphe en démontrant et discutant les résultats obtenus.

4.2. Terminologie

La bioinformatique est une science encore jeune faisant intervenir des problèmes biologiques dont la modélisation, résolution et simulation sont confiées à des méthodes informatiques et des outils mathématiques. De nos jours, les biopuces forment une partie importante de la bioinformatique. Mais avant d'entamer ce domaine, nous allons exposer les quelques définitions de base attachées à la bioinformatique et représentant le jargon du biologiste et informaticien.

Cellule : c'est l'élément de base de tout organisme vivant [100]. A l'échelle microscopique, une cellule est constituée du cytoplasme, un noyau et une membrane.

L'interaction entre les cellules par le biais de réactions biochimiques sert au fonctionnement de l'organisme. Le nombre de cellules varie d'un organisme à un autre. Il peut varier de un (bactérie et levures) jusqu'à plus d'un milliard pour les organismes complexes [100].

Protéine : tout organisme vivant repose sur un éventail de protéines, glucides, lipides et acides nucléiques. L'explication des protéines à part entière mérite un chapitre mais il est plus intéressant de mentionner les rôles remplis par les protéines. Essentiellement, les protéines de structure sont assimilées à des briques cellulaires chargées du transport des autres molécules [101]. L'exemple célèbre est celui d'homoglobine transportant l'Oxygène. L'autre catégorie de protéines est représentée par les enzymes qui ont pour mission l'accélération des réactions chimiques. Ainsi, la glucose 6 phosphates est l'outil biochimique qui initie la dégradation du glucose. Enfin, la défense de l'organisme contre les virus est assignée aux protéines d'immunité (anticorps).

ADN (Acide désoxyribonucléique) : figure parmi les éléments fondamentaux pour la production d'une ou plusieurs protéines. Il se présente dans le noyau de chaque cellule [102]. D'un point de vue structure chimique, l'ADN est composé de deux brins (hélices) de nucléotides, composés eux-mêmes du sucre (ribosome), phosphate et l'une de quatre bases. Les deux polymères sont enchaînés par des liens d'hydrogène [101]. Plus explicitement, les bases constituant l'ADN appartiennent à l'alphabet suivant:

A : Adénine
G : Guanine
C : Cytosine
T : Thymin

A cause de quelques propriétés chimiques, T est le pair de A et C est le pair de G.

Gène : de façon très concise, un gène est un fragment ADN contenant une information génétique [101]. Plus particulièrement, chaque gène comporte non seulement la séquence en nucléotides qui sera transformée en acides aminés, mais aussi les séquences permettant de régulariser cette fabrication. Les spécificités des gènes résident dans les points suivants [100]:

- Leur localisation sur une molécule ADN est difficile. La raison principale est strictement liée à leurs disséminations le long d'ADN.
- Ont une taille variable.
- Peuvent être morcelés en nombreux fragments informationnels (exons), séparés par des morceaux de séquences inconnues (introns).

Chromosome : concrètement, les gènes sont localisés dans des chromosomes, construits au tour d'ADN et protéines [103].

ARN (Acide ribonucléique): chargé du transfert et traitement de l'information dans la cellule [104]. L'ARN n'a suscité un intérêt que durant ces dernières années. Il est constitué d'un seul brin contenant presque les mêmes bases que l'ADN. On distingue : A, G, C et l'Uracil (U). Impliqué dans plusieurs problèmes, telle que la résistance aux

Chapitre 4 : Calcul amorphe pour la classification des données issues des puces à ADN par GNG

antibiotiques, le ARN possède une structure structurale et enzymatique complexe [105]. Il peut jouer différents rôles selon son type : ARNm (messager), ARNr (ribosomique) ou l'ARNt (transfert).

Production d'une protéine : malgré que ce processus est assez compliqué, il a été synthétisé par les deux étapes suivantes [103]:

- Transcription : en substituant à chaque base ADN son complément pour la production d'un ARNm.

ADN AGATACAGGGACCATCCATCCAG
 ARNm UCUAUGUCCUGGUAGGUAGGUC

- Translation du ARNm jusqu'au cytoplasme en produisant une protéine. La protéine est obtenue par découpage successif de la chaîne ADNc (ADN complémentaire).

UCU UCG UCC UCA	Serine	GAU GAC	Asparagine acid
CCU CCC CCA CCG	Proline	GAA GAG	Glutamic acid
ACU ACC ACA ACG	Thréonine	UGU UGC	Cysteine
GCU GCC GCA GCG	Thyrosine	UGA	Stop
UAA UAG	Stop	UGG	Tryphosphar
CCA CAG	Glutamine	UUU UUC	Phenylalanine
AAU AAC	Asparagine	UUA UUG CUU CUA CUG	Lencine
AAA AAG	Lysine	AUG	Méthionine
CGU CGC CGA CGG	Arginine	AGA AGC	Arginine
AGU AGC	Serine	GGU GGC GGA GGG	Glycine

Tableau 4.1. : Liste des protéines [101].

Transcriptome: c'est l'ensemble des ARNm transcrits, c'est à dire des milliers de transcrits différents dans une cellule à un instant donné. Il peut varier selon [102]:

- Type cellulaire (cellule sanguine, peau, neurone, ...).
- Différentes phases de la vie d'une cellule.
- L'environnement et les conditions du stress.
- Une maladie éventuelle.

Niveau d'expression : c'est une mesure discrète ou continue qui fournit une description quantitative de l'expression d'un gène en mesurant la quantité des molécules intermédiaires produites (ARNm et ARNt) [103].

Génome : c'est l'ensemble de l'information génétique d'un organisme se présentant dans chacune des cellules.

La figure suivante tente de synthétiser tous les concepts évoqués au-dessus.

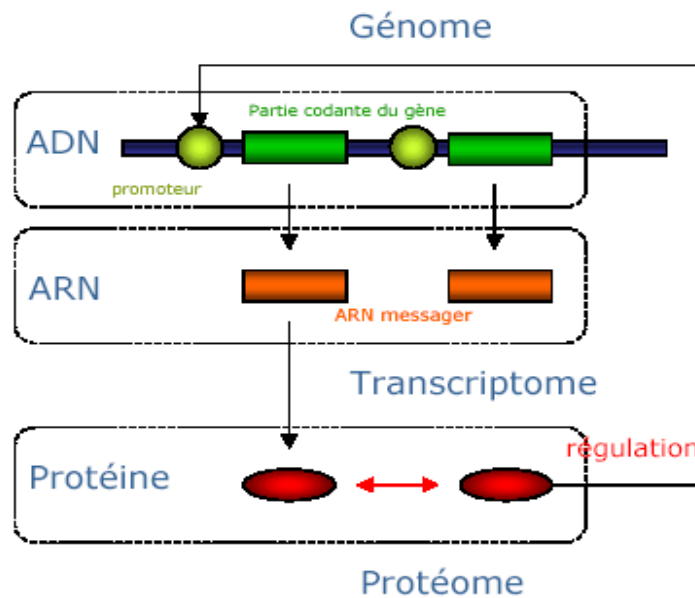


Figure 4.1 : Génome [106].

4.3. Biopuces

Outre les investigations dans l'industrie biologique, l'émergence des biopuces est la tendance adoptée sur laquelle il faut compter. Construites au tour d'une surface en cm^2 , les biopuces sont des microsystèmes permettant l'analyse d'un nombre considérable de séquences contenues dans un échantillon ADN. Elles offrent l'avantage d'automatiser, miniaturiser et paralléliser différentes étapes lors d'analyse biologique. Conséquemment, le gain en temps est d'autant plus important que les expériences réalisées auparavant qui nécessitaient des journées [107, 108]. Bien entendu, le terme 'biopuce' date au début des années 90 et désigne tels systèmes s'appuyant sur l'appariement des bases de deux séquences ADN. Cela est motivé par la capacité des brins extraits de la double hélice ADN à reformer spontanément cette double hélice dès qu'ils se retrouvent face au brin complémentaire [108]. Comme tout appareil électronique, chaque biopuce possède ses propres caractéristiques influant directement sur son prix. Elle est accompagnée de logiciels pour son fonctionnement. En général, les biopuces peuvent être impliquées dans divers champs d'application. En particulier, les biopuces permettent l'hybridation des brins synthétiques avec ceux d'un patient pour s'assurer que le malade est porteur de l'affection recherchée [108].

4.3.1. Types des biopuces

La diversité des biopuces mises en œuvre est strictement liée aux problématiques biologiques abordées ainsi qu'au prix de construction. Essentiellement, les laboratoires sur puce, les puces à ADN et les puces à cellules sont les biopuces les plus commercialisées.

4.3.1.1. Laboratoires sur puce

Ces biopuces se fondent sur l'ensemble des techniques allant de la préparation préalable à l'utilisation d'une puce à ADN. Elles intègrent : un système de dosage, de mélange d'échantillons liquides, de séparation, de purification des molécules d'intérêt et de transfert des mélanges vers un compartiment à température. Cependant, plusieurs outils sont associés aux laboratoires sur puce. Typiquement, la PCR (réaction en chaîne de polymérisation) permet de photocopier l'ADN en plusieurs copies. Elle amplifie la présence de l'ADN dans une solution [108].

4.3.1.2. Les puces à cellule 'cell-on-chip':

Pour éviter aux biologistes de dépenser des coûts importants lors de la fabrication d'un médicament, les puces à cellule sont actuellement commercialisées pour analyser des cellules vivantes. Ces biopuces sont capables de déterminer de façon précoce les niveaux de toxicités et d'efficacité d'un médicament tout au long de son cycle de développement [108]. L'une des applications les plus intéressantes consiste à faire un tri cellulaire. Avec une puce à cellule composée d'un réseau de microélectrodes, il est dorénavant possible de manipuler chaque cellule individuellement. Cela permet d'approfondir l'étude d'un organisme complexe.

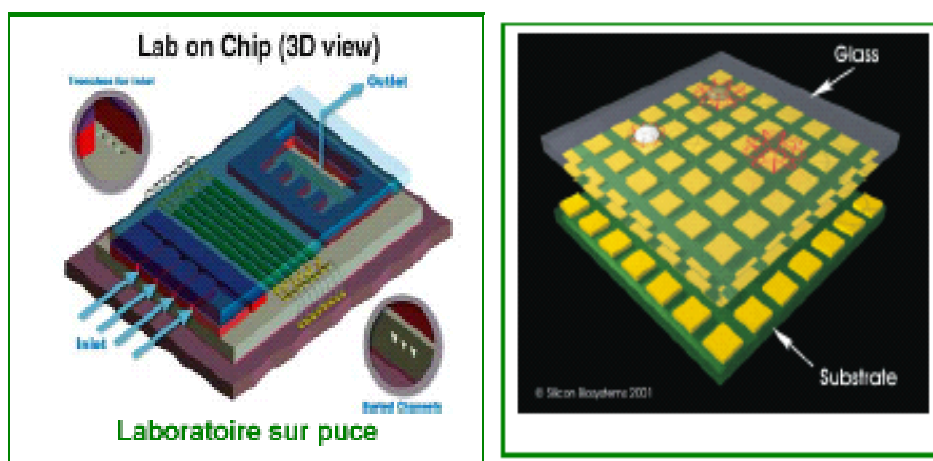


Figure 4.2.a. : Laboratoire sur puce [108]. Figure 4.2.b. : Principe d'une puce à cellule [108].

4.3.1.3. Puces à ADN

Moins performantes que les laboratoires sur puce, les puces à ADN sont constituées d'un support de petite taille en plastique ou en verre, sur lequel sont synthétisées ou greffées

des milliers de sondes (brins synthétiques). Les sondes s'hybrident avec l'acide nucléique à élucider, qui est préalablement purifié, amplifié et marqué [108]. Les signaux d'hybridation sont détectés, puis analysés par traitement informatique. A partir du moment où l'on décide utiliser une puce à ADN, deux types de support, dont le choix est conditionné par des propriétés physico-chimiques, sont disponibles : neutre et actif. Plus profondément, le support neutre (verre) assure une fixation optimale des sondes. En plus de la fixation des sondes, l'acquisition du signal est l'une des caractéristiques du support (silicium, électrodes de carbone et les fibres optiques) [109].

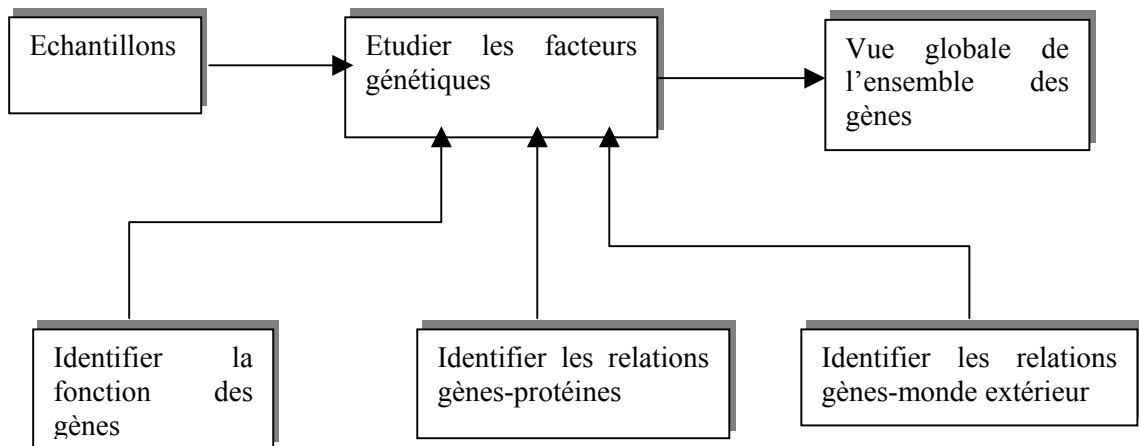


Figure 4.3 : Apport des puces à ADN.

4.4. Applications des puces à ADN

Une puce à ADN est unique en son genre. Pour deux raisons essentielles. L'une, ses différentes fonctionnalités. L'autre est liée à sa rapidité d'analyse d'informations biologiques provenant de nombreux échantillons. Elle peut remplir les rôles suivants [110]:

- Séquençage par hybridation. Le gène étudié est considéré comme un ensemble de séquences chevauchantes dont la détermination simultanée puis l'assemblage permettent de reconstituer la séquence.
- Détermination des distances génétiques qui existent entre Bactéries. Par exemple, une puce dédiée à *Escherichia Coli* a été utilisée pour obtenir des informations sur les gènes d'une espèce (tsé-tsé) très proche phylogénétiquement d'*Escherichia Coli*.
- Etudier la cartographie et les modifications du génome en déterminant les sites de fixation d'une protéine sur l'ADN.
- Mesurer directement les concentrations en protéines et leur conformation spatiale. Pour cela, des anticorps spécifiques sont fixés pour piéger les protéines.

A l'origine, la plus grande partie de recherches sur les puces à ADN est menée par le CEA qui s'intéresse beaucoup plus aux affections par le virus HIV et l'étude des cellules musculaires [111]. Cependant, les puces à ADN peuvent être impliquées dans [110]:

- Le secteur agro-alimentaire : pour le contrôle de qualité.
- Le domaine de l'environnement : détection des agents infectieux dans l'eau ou l'eau.

Signalons aussi que l'application industrielle des puces à ADN doit prendre en charge les facteurs suivants :

- Coût important.
- Nécessité des plusieurs outils : fluorescence, radioactivité,...

4.5. Conception d'une puce à ADN

En langage biologique, l'abondance des espèces ARNm contenues dans un échantillon biologique correspond à l'expression transcriptionnelle des gènes. Pour aller plus loin, les puces à ADN sont un outil, par excellence, apte à mesurer le niveau d'expression des gènes inconnus par rapport à un témoin de référence. Selon un signal fluorescent, on peut estimer le taux d'expression différentiel du gène manipulé [110].

4.5.1. Fabrication et fixation des sondes

Par définition, les sondes sont des séquences doubles brins caractérisant les gènes dont on cherche à déterminer la fonction [100]. Ainsi, pour fixer des sondes sur une puce à ADN, deux voies sont possibles :

- Fixation des brins ADN déjà synthétisés.
- Synthèse in situ.

Respectivement à la première méthode, des micro gouttelettes des solutions ADN obtenues par amplification PCR sont déposées par un robot selon une matrice d'emplacement définis et traitée par revêtement chimique pour fixer l'ADN [112]. A la différence de cette approche conçue pour la première fois par les chercheurs de l'institut de biologie moléculaire de Moscou en 1989 et s'inscrivant dans un cadre mécanique, l'adressage électrochimique a été développé en France par Cis BioInternational en partenariat avec CEA de Grenoble. Il consiste à activer spécifiquement un microélectrode afin d'adresser une séquence définie d'oligonucléotides [109]. En revanche, la technique in situ développée par les Hollandais d'Affymetrix est restreinte à quelques sociétés spécialisées mais bénéficie d'une certaine baisse de prix et un contrôle de qualité accrue [112]. En matière d'utilisation, la technique in situ consiste d'abord à déposer des couches successives des quatre nucléotides sur un support de verre. Ensuite, un masque, dont la configuration varie selon chaque couche, est employé. Il assure une succession correcte des bases de telle sorte à corriger les éventuelles erreurs de synthèse. Avant d'entamer la prochaine étape, on dénature l'ADN pour qu'il se trouve sous la forme d'un seul brin [100].

4.5.2. Marquage et hybridation

Quant aux expériences menées par une puce à ADN, deux approches ont été testées: la comparaison d'une condition expérimentale (stress, pathologie ou un état de différenciation cellulaire) avec une condition de référence, ou la mesure du transcriptome d'un génome. Par rapport au premier cas d'usage, toutes les molécules ARNm des deux types cellulaires sont extraites puis marquées par deux types de fluorophores. Les fluorophores classiquement utilisés sont la Cyanine 5 (Cy5) et la Cyanine 3 (Cy3) qui émettent respectivement dans le rouge et dans le vert [109]. Si l'on s'intéresse à mesurer le transcriptome, les ARN extraites doivent être transcrites inversement. Le marquage portera sur ces deux types de séquences. Au cours de l'étape d'hybridation, les ARN cibles sont mises en présence des sondes. Après la lecture des lames, la couleur de la fluorescence de chacun des spots traduit la plus ou moins grande abondance des transcrits du gène correspondant dans la condition expérimentale ou dans la condition de référence. Expliquons que l'hybridation fait référence à la formation et stabilité des hybridés [113]. Elle est liée aux sondes (structure et localisation) ainsi qu'à la température d'hybridation. Celle-ci doit, en général, être comprise entre 5 et 10°C.

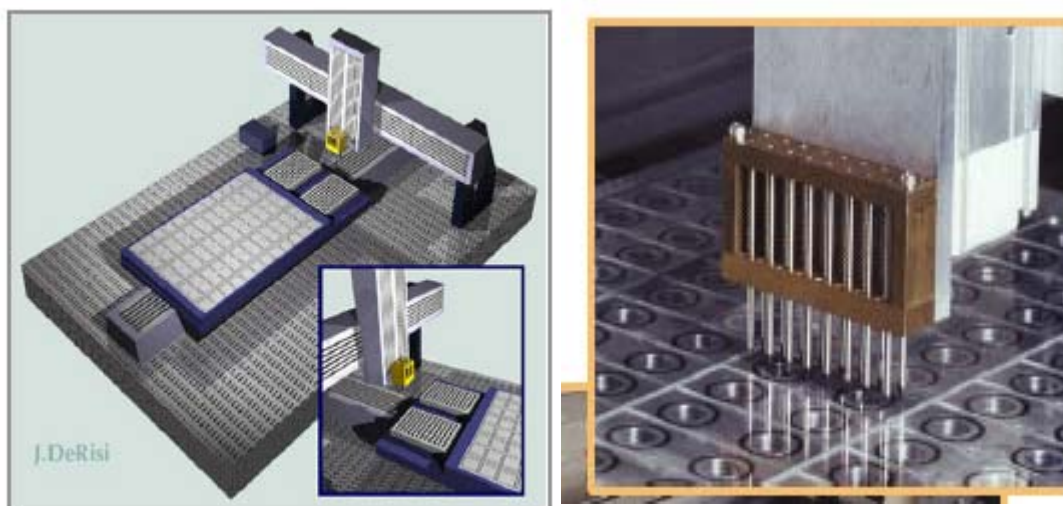


Figure 4.4.a. : Fixation des sondes [114]. Figure 4.4.b. : Tête de fixation [114].

4.5.3. Acquisition des images

Cette étape est menée par la lecture des puces sur un scanner. En ce sens, le scanner est muni de deux lasers qui permettent l'acquisition simultanée des signaux émis par les fluorophores Cy3 et Cy5. Le résultat de la lecture est modélisé par deux images en niveau de gris chacune correspondant à un fluorophore. En général, les images obtenues par chaque canal sont au format Tiff 16 bits en 65535 niveaux de gris, telle que la gamme d'intensité détectée est comprise entre 0 (noir) et 65535(banc) [115]. En général, les images scanner sont entachées de bruit. Par conséquent, le traitement d'images scanner est une étape primordiale pour améliorer la qualité des images obtenues afin d'extraire des valeurs significatives pour les gènes.

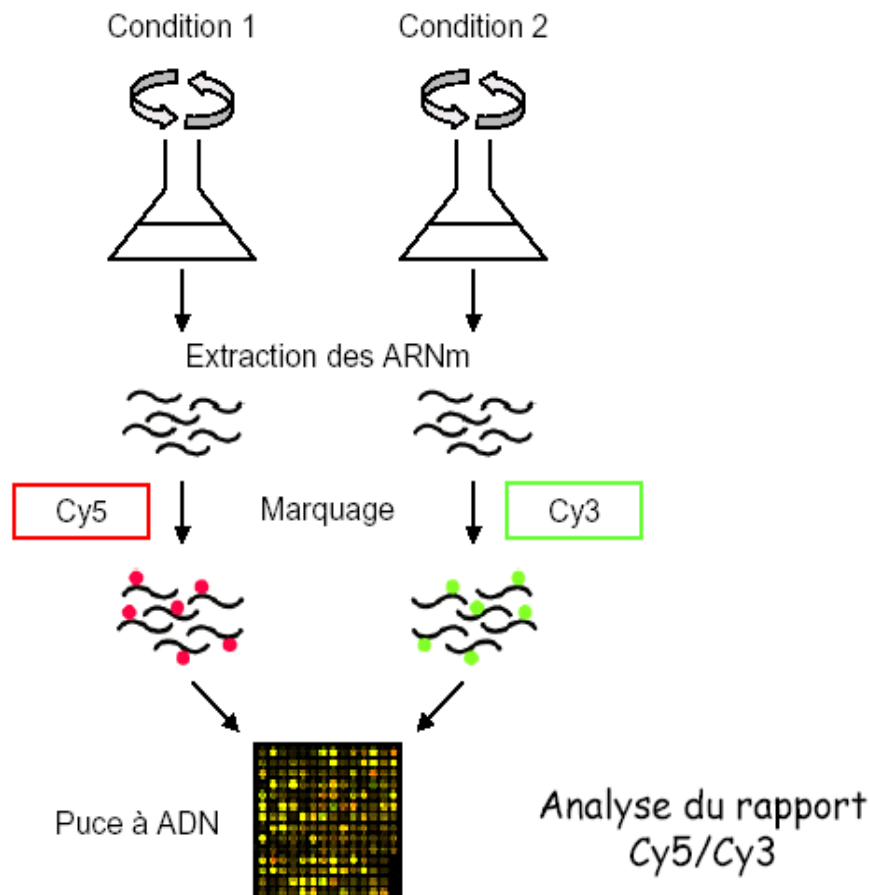


Figure 4.5. : Etapes de conception d'une puce à ADN [116]

La segmentation d'images régit le fonctionnement d'un grand nombre de systèmes. Le fondement théorique de ce processus a été expliqué dans le chapitre 3. Afin de discerner pour chaque spot les pixels bruits, il importe de segmenter les images scanner. Notamment, les concepteurs des puces à ADN se sont efforcés dans l'application d'une technique de segmentation appropriée. L'expérience la plus quotidienne consiste en l'utilisation d'une segmentation par cercle fixe [100]. Technique adaptée pour des spots homogènes, la segmentation par cercle fixe identifie les pixels à l'intérieur du cercle comme étant un signal. A la différence des pixels signales, les pixels bruits se retrouvent à l'extérieur du cercle. Constatons l'inconvénient de sensibilités aux irrégularités de la taille et forme de spots, les concepteurs des puces à ADN ont mis au point des méthodes de segmentation par cercle adaptable dont le diamètre est estimé par une méthode automatique. Pour réaliser une segmentation plus simple, le seuillage à partir de l'histogramme est une approche intuitive. Il convient de fixer un seuil permettant de fixer la classification des pixels. Cependant, un compromis entre la qualité de segmentation et le temps d'exécution est un facteur important quant à l'application pratique d'une technique de segmentation.

Une fois les images scanner ont été analysées, les étapes qui suivent seront exécutées par un système informatique intégrant des logiciels spécifiques afin de calculer le transcriptome du génome testé.

4.5.4. Traitement des données brutes

A la fin de l'étape d'analyse d'images, chaque sonde est caractérisée par deux mesures d'intensité de fluorescence verte et rouge ainsi qu'une mesure du bruit local. Les signaux rouges et verts ne peuvent pas être interprétés séparément [109]. Les puces à ADN permettent de mesurer une variation d'expression d'un gène entre deux conditions expérimentales : il s'agit d'une mesure relative. Schématiquement, pour chaque sonde, le ratio de la fluorescence de la condition expérimentale sur la fluorescence de témoins est calculé. Si le rapport est inférieur à 1, le gène correspondant à la sonde déposée est réprimé, dans le cas contraire, il est induit. Réellement, c'est le logarithme (en base 2) du ratio qui est étudié. Le résultat final est représenté par une matrice (gène, condition), où la case (i, j) dénote la valeur prise l'expression i dans la condition j [110].

La valeur prise par le gène Gène5 dans la condition Cond1

	Cond1	Cond2	Cond3	Cond4	Cond5
Gène1	0.08	0.44	0.24	0.32	0.88
Gène2	0.66	0.95	0.67	0.55	0.95
Gène3	0.22	0.11	0.38	0.71	0.51
Gène4	0.12	0.10	0.05	0.61	0.27
Gène5	0.13	0.99	0.35	0.24	0.25

Figure 4.6. : Résultat d'une puce à ADN.

Jusqu'ici, nous avons présenté les étapes que doivent suivre les biologistes et chimistes. Au cours de leur mission, les questions suivantes sont à expliciter [110, 115]

- Quel est le meilleur échantillon à utiliser pour réduire la variance biologique ?
- Quel est le nombre de répétitions de mesures pour augmenter la précision ?
- Comment éviter les problèmes de marquage de la cible ?
- Comment mener une bonne hybridation de sorte que la solution ADN ne tend pas à se concentrer sur les bords des spots en séchant ?

Pareil pour l'informaticien qui se retrouve confronté à des expressions de gènes complexes. Il est bien connu que l'analyse de données issues des puces à ADN doit prendre en charge les aspects suivants :

- Dissymétrie du tableau (gène, condition). Expliquons que le nombre de conditions expérimentales est bien souvent très faible par rapport au nombre de gènes.
- Les gènes ne sont pas indépendants. La distribution des niveaux d'expression est dissymétrique.
- Les valeurs obtenues peuvent être bruitées. Cela est relativement lié à la poussière ou la précipitation de sels.
- Présence de valeurs manquantes.



Figure 4.7.a.: Oubli de la préhybridation [115].

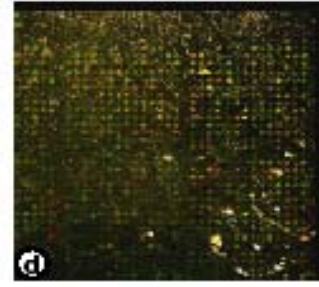


Figure 4.7.b.: Cristallisation de sels [115].

4.5.5. Normalisation et filtrage

L'objectif de cette phase est d'ajuster l'intensité globale des images acquises sur chacun des canaux, de manière à corriger les biais techniques. Plus précisément, les biais techniques tendent à faire déséquilibrer le signal de l'un des canaux par rapport à l'autre. L'hypothèse de la normalisation est que la majorité des gènes ont un comportement invariant entre deux conditions expérimentales, ce qui se traduit par $\log_2(r)=0$. La normalisation vise à ramener la moyenne de cette grandeur à 0 sur chaque lame. En général, l'immense nombre des critères de normalisation s'explique par les résultats obtenus et les besoins utilisateur [110].

- La normalisation globale est basée sur l'hypothèse que les intensités rouges R et vertes V soient globalement liées par un facteur constant. En effet, la normalisation par moyenne suit le processus suivant:

$$\begin{cases} R=KG \\ \text{Log}_2 \frac{R'}{G'} = \text{Log}_2 \left(\frac{R}{KG} \right) = \text{Log}_2 \left(\frac{R}{G} \right) - \text{Log}_2 K \end{cases} \quad (4.1)$$

K est la moyenne des Log ratio pour l'ensemble des gènes.

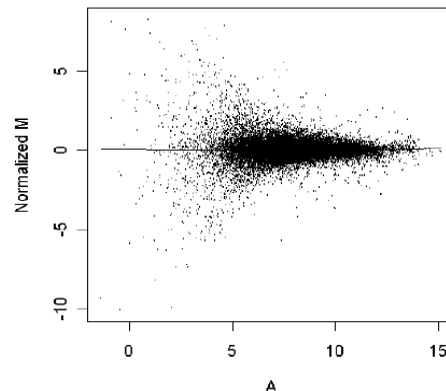
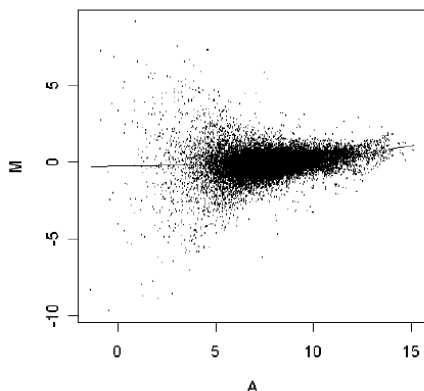


Figure 4.8.a. : Représentation des données [116]. Figure 4.9.b. : Résultat de normalisation [116].

- La normalisation linéaire est de la forme :

$$\text{Log}_2 R = a + b(\text{Log}_2 G) \quad (4.2)$$

$$\begin{cases} \text{Log}_2 R' = \text{Log}_2 R - a \\ \text{Log}_2 G' = b(\text{Log}_2 G) \end{cases} \Rightarrow \begin{cases} R' = \frac{R}{a} \\ G' = G^b \end{cases} \quad (4.3)$$

Par contre, le filtrage a été développé pour éliminer les sondes pour lesquelles une des mesures d'intensité est inférieure à un seuil (arbitraire ou choisi a priori).

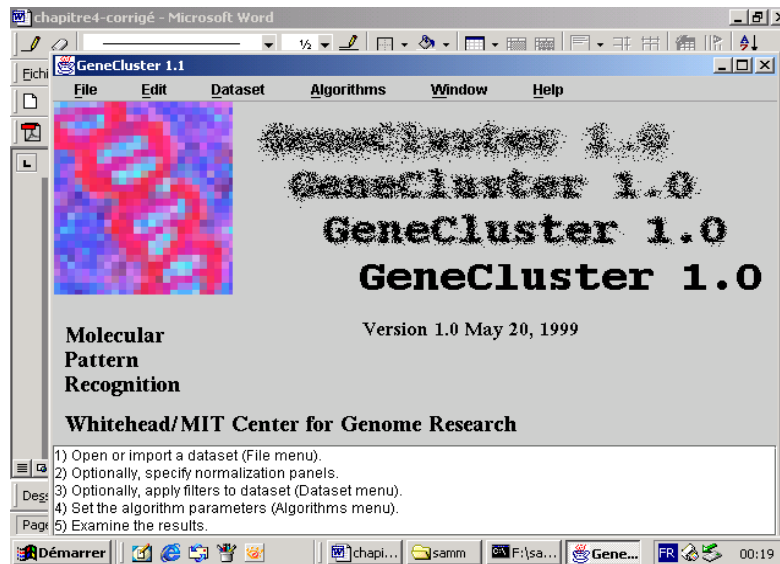


Figure 4.9.: Fenêtre initiale de GeneCluster1.

4.6. Analyse de données

Pour qu'elle soit rentable, une puce à ADN doit être associée d'une technique d'analyse de données. De façon classique, on peut s'attacher à la prédiction d'un ou plusieurs phénotypes connus en termes des profils d'expression de certains gènes. Logiquement, la conception d'une méthode de prédiction est basée sur un apprentissage supervisé. Bien qu'une diversité de méthodes de prédiction a été utilisée aux cas des puces à ADN, il semble cependant difficile de sélectionner les caractéristiques d'apprentissage. Pour tenter de cerner les limitations des méthodes d'apprentissage supervisé, les méthodes de classification citées dans le chapitre 2 ont été retenues pour l'identification de phénotypes au départ inconnus à partir des profils d'expression. Dans ce contexte, on cherche à générer des hypothèses sur les gènes regroupés ensemble. Dès l'apparition des puces à ADN, les concepteurs ont privilégié les techniques hiérarchiques. Ces méthodes semblent faciles à utiliser. En plus, elles offrent une

représentation visuelle. Par exemple, la figure (4.10) illustre le résultat de l'application d'une méthode hiérarchique. Par ailleurs, l'algorithme SOM et le K-means ont prouvé leurs capacités à regrouper des expressions de gènes. Si l'on s'intéresse à l'apprentissage non supervisé, plusieurs logiciels téléchargeables implémentant le K-means et SOM fournissent une solution rapide et performante. Les biologistes et médecins, les trouvent un moyen souple et efficace pour analyser des milliers de gènes. En initialisant les neurones à des poids aléatoires, l'apprentissage par SOM permet de partitionner l'ensemble de gènes. Les gènes ayant une expression élevée se retrouvent affecter à des neurones adjacents. Parmi les outils mis en œuvre, on y trouve GeneCluster. Il est à sa version 2. C'est un outil qui permet la classification des expressions de gènes par l'algorithme SOM et le K-means. En plus, il permet le filtrage de données. Pour cela, GeneCluster accepte des bases de données d'extension (res, gct). Un simple clic sur la commande (SOM, K-means) à partir du menu algorithms donne à l'utilisateur la possibilité d'adaptation des paramètres (voisinage, taille de la carte) pour pouvoir visualiser les corrélatons entre les différents gènes impliqués.

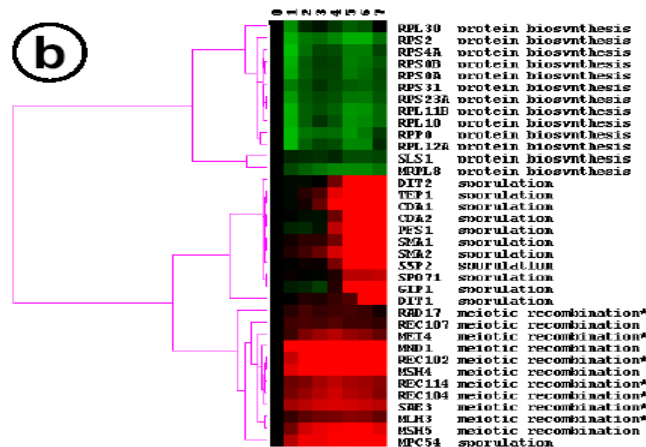


Figure 4.10. : Résultat de la classification hiérarchique [57]

Si l'on se rapporte au chapitre précédent, on apprécie que le K-means et SOM souffrent de quelques limitations. Par conséquent, plusieurs directions s'ouvrent actuellement pour la conception d'une méthode de classification qui se conforme aux puces à ADN.

4.7. Classification des données issues des puces à ADN

Plusieurs auteurs ont récemment tenté de proposer des méthodes de classification qui ont vu le jour dans le domaine des puces à ADN. Parmi lesquelles, on peut citer :

- ❖ La méthode de classification qui a été décrite par B.Chen, P.C.Tai, R.Harrison et Y.Pan [115] a été expérimentée au cas des données issues des puces à ADN. Elle fait appel à la classification hiérarchique et l'algorithme du K-means. Le principe de cette solution est la classification hiérarchique d'un certain pourcentage de données biologiques. Enfin, la moyenne de chaque cluster servira comme centre initial pour

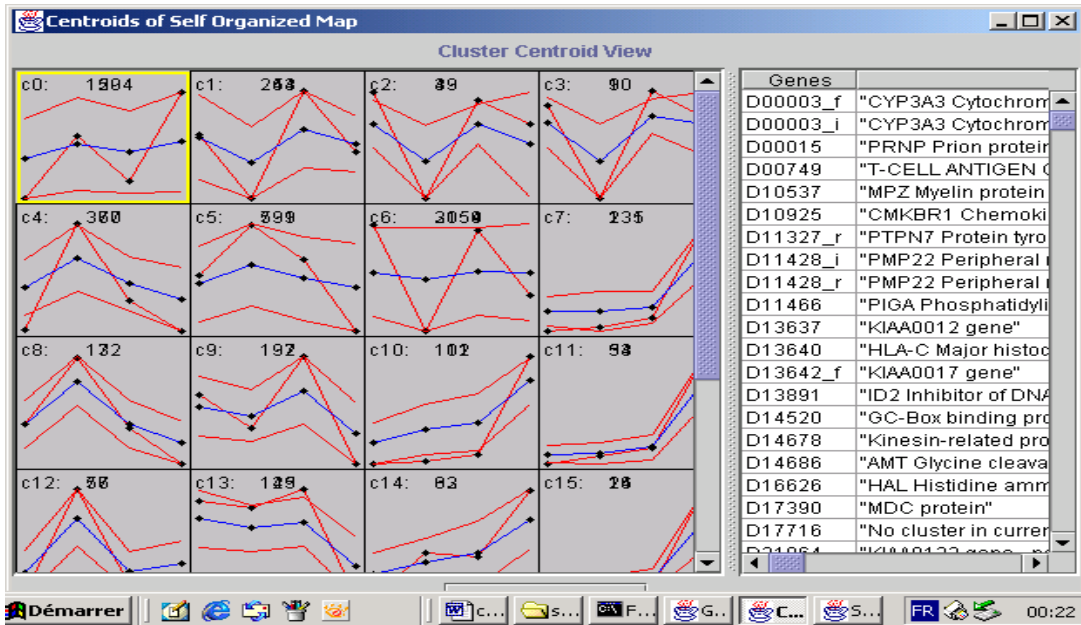


Figure 4.11. : Corrélations entre gènes

L'algorithme du K-means chargé de partitionner rapidement le reste des données aux centres correspondants. En plus, cette technique met l'accent sur le critère d'assignation d'un gène à un cluster lors de l'utilisation du K-means. Plus précisément, tout gène dont la distance est inférieure à un certain seuil se trouve affecter à une classe particulière 'minor-group'. Pour des besoins spécifiques, le choix de cette méthode est encouragé par la rapidité du K-means ayant comme entrée un ensemble de centres générés de façon méthodique. Regroupant les gènes similaires, en évitant la connaissance préalable des centres de clusters, cette méthode est synthétisée comme suit [117]:

1. Répéter

Trouver deux objets (gènes) similaires.

Calculer les nouvelles valeurs comme étant la moyenne des anciennes valeurs.

Jusqu'à un certain pourcentage.

2. Appliquer le K-means.

❖ J.Kasturi, R.Acharya et M.Ramanatan [118] furent parmi plusieurs auteurs qui ont exploité la théorie de l'information dans le cadre de la classification de données. Le problème important lié au critère de similarité est remplacé par l'application de la mesure d'entropie. Précisons que le terme entropie signifie le degré du désordre. Si $P(x)$ et $Q(x)$ sont deux masses de probabilité, l'entropie de Kullback-Leibler est :

$$P(P||Q)=\sum_{x \in X} P(x) \log_2 \frac{P(x)}{Q(x)} \quad (4.4)$$

Malgré que la classification des données issues des puces à ADN a été empruntée par SOM, la nouveauté de cette méthode est l'application de l'entropie de Kullback-Leibler comme critère de similarité. L'algorithme SOM tel qu'il a été adapté pour donner des résultats extrêmement meilleurs que ceux du SOM classique est le suivant :

1. Initialiser K centres de clusters C_1, \dots, C_n .
2. **Répéter** jusqu'à convergence
Sélectionner un gène g normalisé

Calculer $D(G\|C_i) \quad i=1 \dots N$

Identifier le centre K le plus proche de g

Mise à jour des poids par la règle d'apprentissage classique tel que :

$$\eta 1(n+1) = \frac{\eta 1(n)}{1+0.0005n^{0.01}} \quad (4.5)$$

$$\eta 2(n+1) = \frac{\eta 2(n)}{1+0.005\eta^{0.05}} \quad (4.6)$$

❖ Le cancer est une maladie riche en communications scientifiques, mais dont les solutions médicales sont assez lointaines. S'intéressant à des données des gènes représentant cette maladie, la démarche développée par H.Li, K.Zhang et T.Jiang [119] a perforé de très bon résultat. Elle intègre un processus de minimisation de l'entropie au sein de chaque cluster. Le critère de minimisation de l'entropie porte sur la formule suivante:

$$j = \sum_{j=1}^m P_j H(x/C_j) \quad (4.7)$$

Sachant que la technique conçue accepte comme entrée m clusters et n objets, la sortie générée est un ensemble de m clusters minimisant l'entropie. Pour atteindre cet objectif, la méthode ainsi conçue peut être vue comme un processus d'affectation de chaque gène au cluster adéquat. L'idée derrière est la suivante :

Pour chaque gène x **faire**

Si le cluster C_j contenant la majorité des voisins de C_i est différent de C_i **alors**

$$h \leftarrow \sum_j (H(C_y) - H(C_x)) \quad (4.8)$$

y est l'ensemble des voisins. H , H' sont les entropies avant et après affectation de x à j .

Si $h < 0$ **alors**

affecter x à C_j

Fin Si

Fin Si

Fin Pour

❖ Ayant une fondation statistique et offrant une représentation graphique pour faciliter l'interprétation des résultats, les réseaux Bayésiens ont été adoptés comme formalisme pour la classification des données issues des puces à ADN [120]. A court terme, un réseau Bayésien est un ensemble de nœuds (variables) et des liens orientés représentant les dépendances. Pour chaque variable x possédant plusieurs parents y_1, \dots, y_n , on définit une probabilité conditionnelle $P(x/y_1 \dots y_n)$. Pour sélectionner une structure du réseau Bayésien, plusieurs modèles sont mis en jeu. Lors de l'application des réseaux Bayésiens, la question fondamentale est l'apprentissage des valeurs de probabilité assignées aux gènes afin d'attribuer chacun d'eux à sa classe correspondante.

❖ En alternative de SOM, J.Herrero A.Valencia et J.Dopazo [121] ont proposé une technique de classification pour l'identification des gènes corrélés. Elle se base sur l'algorithme Self-organizing tree (SOTA) en s'orientant vers une classification hiérarchique obtenue par le processus d'adaptation des cartes auto-organisatrices. Plus particulièrement, la dynamique du réseau est analogue à celle du GCS. L'adoption de cette méthode est surtout motivée par le temps d'exécution qui est proportionnellement linéaire au nombre de gènes. La stratégie développée est la suivante :

1. Initialiser deux cellules reliées par un nœud. Leurs vecteurs prototypes sont initialisés à la moyenne de chaque colonne.
2. Adaptation des poids (vecteurs prototypes) selon les valeurs du gène présenté.
3. Effectuer un test sur les cellules hétérogènes pour générer deux cellules descendantes. Une cellule est considérée hétérogène selon sa ressource R calculée comme suit :

$$R_i = \frac{\sum_{k=1}^K dP_k C_i}{K} \quad (4.9)$$

d : Coefficient de corrélation de Pearson.

P : L'expression d'un gène associé à la cellule i .

C_i : Vecteur prototype de i .

4.8. Classification des expressions de gènes par calcul amorphe et le GNG

L'essor des puces à ADN, permettant l'analyse quantitative des gènes, a accompagné le besoin du traitement parallèle de tout un génome. L'intégration d'une technique de classification adéquate est un facteur clé dans la découverte des gènes similaires (régulés par le même processus). On notera toutefois le recours à l'algorithme du K-means pour

la classification des données issues des puces à ADN. La rapidité du partitionnement des gènes aux classes correspondantes est un atout majeur pour décider rapidement et au mieux aux objectifs attendus lors de la conception de la puce à ADN. Si le K-means a été appliqué avec succès pour le cas de données biologiques, l'inconvénient de l'estimation des centres des classes est un point qui a été mentionné dans le chapitre 2. Compte tenu des avantages offerts par le GNG ne comportant pas l'initialisation du nombre de clusters, ni l'information concernant la distribution initiale, nous avons préconisé cet outil pour la classification des données issues des puces à ADN. Au-delà des avantages du GNG, l'investigation de ses potentiels pour retrouver l'organisation des gènes est motivée par :

1. Bien adapté pour des données numériques.
2. C'est un outil pouvant visualiser implicitement les relations entre un nombre considérable de gènes.
3. L'apprentissage incrémental permet de reconstituer de nouvelles valeurs pouvant masquer le bruit inhérent.

D'autre part, les acquis du calcul amorphe sont bien intégrés dans la génération de motifs 2D. Néanmoins, la puissance escomptée par un système de calcul amorphe reste inexploitée pour la prise en charge de plusieurs difficultés. C'est dans ce contexte que l'on a imaginé la classification de données issues des puces à ADN par une machine de calcul amorphe composée de processeurs (particules) miniaturisés et opérant en dépit de l'abstraction de l'environnement. En effet, l'utilisation d'un système de calcul amorphe pour la classification de données issues des puces à ADN présente les aspects suivants:

1. Puissance de calcul liée directement au nombre de particules de calcul amorphe dont le fonctionnement est asynchrone.
2. Le prix de construction d'un tel système est nettement inférieur au prix induit par les autres systèmes. Ce facteur rend faisable l'intégration d'une puce à ADN, dont le prix est considérable, avec un système de calcul amorphe.

Partant de ces principes, nous avons adapté l'algorithme présenté dans la partie 2 du chapitre 3 dont la finalité consiste à adapter les connexions liant les particules de calcul amorphe afin d'épuiser les contours d'une image. Dans le cas de la classification des données issues des puces à ADN, l'objectif se résume en l'adaptation des vecteurs prototypes associés aux particules de calcul amorphe pour apprendre la topologie de l'espace des gènes. Sachant que les problèmes évoqués lors de l'utilisation du GNG par particules de calcul amorphe ont été expliqués au niveau du chapitre 3, l'algorithme tel qu'il a été modifié est le suivant :

Répéter

Générer un vecteur x

Diffuser ('La distance', $\|w_i - x\|^2$)

Si *etat=vrai* **alors**

$s \leftarrow \min(queue - msg_i)$

Diffuser ('Distance min', s_i)

Fin Si

Si $active_i = vrai$ & $\|w_i - x\|^2 = s_i$ **alors**

Appliquer l'équation (2.26)

S'adapter selon la formule (2.27)

Choisir une autre particule non active h selon l'équation (3.34)

$active_h \leftarrow vrai$

$active_i \leftarrow faux$

Pour chaque $n \in voisinage(i)$ **faire**

S'adapter selon l'équation (2.28)

Choisir une autre particule non active h' suivant l'équation (3.35)

$active_{h'} \leftarrow vrai$

$active_n \leftarrow faux$

Fin Pour

Incrémenter l'age de tous les liens dans le voisinage

Fin Si

Si Cycle mod $\lambda = 0$ **alors**

Diffuser ('L'erreur', $erreur_i$)

Si ($etat_i = vrai$) **alors**

$u_n \leftarrow \max(queue - msg_i)$

Diffuser ('L'erreur max1', u_{n1})

Fin Si

Si $active_i = vrai$ & $erreur_i = u_{n1}$ **alors**

Pour chaque $n \in voisinage(i)$ **faire**

Diffuser ('L'erreur', $erreur_i$)

Fin Pour

$u_{n2} \leftarrow \max(queue - msg_i)$

Diffuser ('L'erreur max2', u_{n2})

Fin Si

Si $active_i = vrai$ & $erreur_i = u_{n2}$ **alors**

Choisir une autre particule r selon l'équation (3.36)

Fin si

Fin Si

Cycle \leftarrow Cycle +1

Jusqu'à un critère d'arrêt

Il est important de préciser que le vecteur d'entrée x contient à chaque itération les valeurs d'un gène. Cependant, les vecteurs prototypes w ont été initialisés à des valeurs aléatoires.

4.8.1. Description de la base

On a appliqué l’algorithme ci-dessus aux données issues de la base PDR. Cette dernière se présente sous la forme d’un fichier Excel. La matrice correspondante est de la forme (gène, condition). Elle a une taille de (6244, 11). Chaque gène possède un identifiant. La précision de la mesure d’une expression peut atteindre 15 chiffres après la virgule. La figure (4.12) illustre un aperçu de la base PDR.

B1	GENE										
A	B	C	D	E	F	G	H	I	J	K	
1	ORF	GENE	pdr1-3(1)	pdr1-3(2)	2h	4h	7h(1)	7h(2)	10h(1)	10h(2)	14h
2	YAL001C	TPC3	0.0285691521	0.8318772411	0.1375035237	0.535331732	0.13606154957603				0.0840
3	YAL002W	VPS8	0.2868811477	0.3785116232	0.18903382439002		0.34007544159762				0.0426
4	YAL003W	EFB1	0.0703893278	0.7990873060	0.089267338097087		0.27500704749987				0.104
5	YAL004W				0.358453970	0.4150374997	0.35845397091248				0.3219
6	YAL005C	SSA1	0.5160151470	0.4005379295	0.4223230006	0.168122758	0.01435529297707		0.6690267655	0.1375035237	0.3103
7	YAL007C	ERP2	0.1243281350	0.1890338243	0.028569152196771		0.074000581443777				0.1375
8	YAL008W	FUN14	0.0285691521	0.8153754289	0.25153876699596		0.18442457113743				0.0389
9	YAL009W	SPO7	0.0840642647	0.2016338611	0.78387519464715						0.0703
10	YAL010C	MDM10	0.251538766	0.4750848829	0.32192809488736						0.304
11	YAL011W				0.1890338243	0.1375035237	0.22650852980868		0.01435529297707		0.3771
12	YAL012W	CYS3	0.1505596765	0.0703893278	0.1890338243			0.029146345659517			0.1634
13	YAL013W	DEP1	0.1110313123	0.1634987322	0.83650126771712						0.286
14	YAL014C				0.4005379295	0.2986583155	0.15055967657538				0.0426
15	YAL015C	NTG1	0.5260688116	0.089267338	0.074000581443777						0.304
16	YAL016W	TPD3	0.321928094	0.3673710656	0.35845397091248			0.200912693926			0.2630
17	YAL017W	FUN31	0.1243281350	0.3219280948	0.94341647163363						0.104
18	YAL018C				0.2868811477	0.23446525363702	1.3219280948874				0.6690
19	YAL019W	FUN30	0.321928094	0.3561438102	0.184424571	0.1505596765	0.23878685958712		0.0143552929	0.2387868595	0.058
20	YAL020C	ATS1	0.6959938131	0.120294233	0.014499569695115		0.13750352374994		0.8399595874	0.4541758931	0.514
21	YAL021C	CCR4			0.1634987322	0.0143552929	0.0976107966	0.1763227726	0.0565835283	0.043943347	0.184
22	YAL022C	FUN26	0.152003093	0.321928094	0.7048719644	0.0565835283	0.20163386116965		0.54596836910529		0.3103
23	YAL023C	PMT2	0.1505596765	0.2750070474	0.4854268271	0.184424571	0.41142624572647		0.4005379295	0.9855004303	0.120
24	YAL024C	LTE1			0.26303440583379		0.36737106564853		1.0426443374085		0.304

Figure 4.12 : Aperçu de la base PDR

L’élaboration de la matrice (gène, condition) a été obtenue grâce aux valeurs initiales des deux canaux. Ainsi, la distribution des valeurs des gènes en fonction des deux échantillons de départ est visualisée dans la figure (4.13).

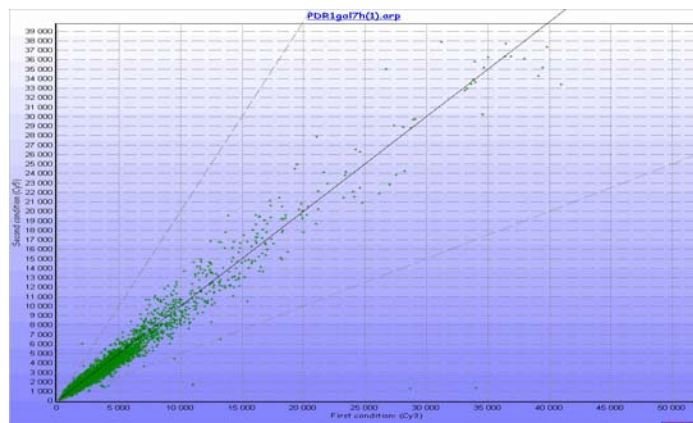


Figure 4.13. : Représentation des données scanner.

8.3.2. Résultats de la classification

Après avoir appliqué l’algorithme en admettant les paramètres illustrés en figure 4.3, nous avons analysé les clusters obtenus par le biais de la méthode U-Matrix ‘Unified

Matrix'. Souvent associée aux cartes auto organisatrices, le principe de U-Matrix est très simple. Soit NN l'ensemble des particules connectées à une particule n , la valeur $u(n)$ est la suivante [122]:

$$u(n) = \sum_{m \in NN(n)} d(w_n - w_m) \quad (4.10)$$

d : Distance.

w_n, w_m : Les vecteurs prototypes associés à n et m respectivement.

e_w	e_n	α	β	a_{max}	m	λ	δ	Particules
0.25	0.23	0.75	0.6	20	4	2	[0,0.8]	3000

Tableau 4.2. : Les paramètres considérés.

D'après les figures (4.14), nous constatons qu'aucun cluster n'est visible au cours des itérations initiales [122].

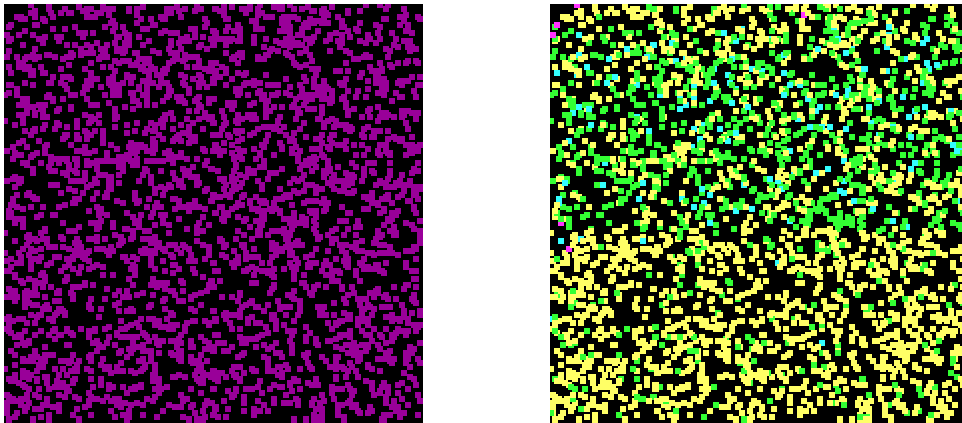


Figure 4.14 : Application de U-matrix après une dizaine d'itérations [122].

A la fin de l'apprentissage, la méthode a convergé et trois clusters de particules de calcul amorphe ont été obtenus. Ces particules de calcul amorphe reflètent implicitement la classification des expressions de gènes.

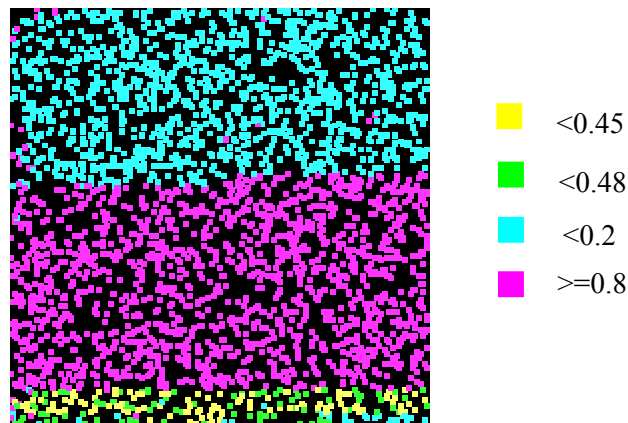


Figure 4.15. : Visualisation des clusters après 1500 itérations [122].

Chapitre 4 : Calcul amorphe pour la classification des données issues des puces à ADN par GNG

Le tableau suivant synthétise l'information de chaque cluster ainsi que quelques statistiques désignant que la moitié des expressions ont une expression élevée[122]. Ce sont des gènes actifs. Seulement 4.73% des expressions ont un niveau d'expression très faible. Ce sont des gènes inactifs.

Color of the cluster	Gene expressions	Number of entities	%
Cyan	YPR198W YAL010C YAL038W YAL004W YAL044C YAL043C YPR196W YPR193C YAL003W YAL002W YAL035W	1263	42.1
Magenta	YAL045C YAL039C YAL009W YAL047C YAL036C YAL011W YAL042W YAL048C YAL001C YPR194C YPR195C YPR199C YPR197C YAL049C YAL007C YAL037W YAL008W YAL043C-A YAL005C YAL040C YAL012W	1533	51.1
Yellow	YAL046C	114	4.73

Tableau 4.3. : Clusters des gènes.

A la fin, il est question de calculer la qualité de la structure obtenue. Pour cela, on s'est inspiré de la méthode AQE (average quantisation error), souvent associée aux cartes auto-organisatrices. L'erreur AQE est définie par :

$$\frac{1}{1+AQE} \quad (4.11)$$

Tel que :

$$AQE + \frac{1}{N} \sum \|x - w_c\| \quad (4.12)$$

x : Vecteur de donnée.

w_c : Vecteur de la particule gagnante.

Après 1500 itérations, la valeur obtenue est 0,9722. Elle signifie une bonne classification.

A partir de la section (4.7), nous remarquons aisément que la classification des données issues est largement dominée par SOM et le K-means. Nous insistons aussi sur le fait que le GNG et le calcul amorphe n'ont pas été utilisés pour regrouper des expressions de gènes. En effet, c'est pourquoi on a pas pu positionner les résultats obtenus par rapport aux méthodes existantes.

Au terme de cette étude consacrée au calcul amorphe et la classification par le GNG, il est temps de dresser le bilan de la contribution proposée. Par rapport au GNG classique, cette méthode nécessite un nombre réduit d'itérations. Les aspects les plus importants sont mentionnés dans le tableau suivant.

La méthode proposée	GNG classique
<ul style="list-style-type: none"> • Pas de synchronisation entre les particules de calcul amorphe. Une fois une particule passe le calcul à une autre, elle peut contribuer à l'exécution d'une nouvelle tâche. • Malgré que la distribution initiale des particules soit irrégulière, l'hierarchie de calcul amorphe permettra de dépasser facilement cette limitation. • Offre un certain degré de fiabilité grâce à la construction d'une hiérarchie de calcul amorphe. 	<ul style="list-style-type: none"> • Les neurones doivent se synchroniser pour établir le graphe final. • Cet algorithme ne nécessite pas la structure de la carte comme entrée. Cependant, il impose la création des neurones dans des emplacements spécifiques. • Non fiable. Une fois un neurone tombe en panne, le graphe établi peut ne pas conserver la topologie.

Tableau 4.4.: Apport de la méthode proposée.

4.9. Conclusion

La biologie se dote enfin des puces à ADN largement diffusées en France. Avec les puces à ADN, l'analyse des milliers d'expression de gènes s'en trouve simplifiée. Les

puces à ADN ne sont pas à proprement parler révolutionnaires. Ce sont la conséquence de l'intégration des techniques de miniaturisation électroniques, des outils de traitement de signal et des algorithmes informatiques. L'impact de cette innovation se mesure davantage dans son usage. Les puces à ADN se sont imposées pour la découverte des gènes corrélés et la détermination de la structure des protéines. Si la décision porte sur l'utilisation d'une puce à ADN pour l'exploitation des données biologiques, ce choix est facile à expliquer. Bien entendu, l'accélération de l'analyse des expressions de gènes et la capacité du traitement d'un nombre important de données sont les qualités offertes par un tel système. Au cœur des puces à ADN, l'extraction des échantillons à analyser, la fixation de sondes, l'hybridation, l'analyse des images scanner et enfin la préparation de données brutes constituent un prétraitement pour l'analyse quantitative. Une fois la matrice (gène, expérience) est établie, la classification est l'une des techniques classiques d'analyse de données visant la recherche d'objets similaires. A côté de SOM et la classification hiérarchique, l'algorithme du K-means a été très tôt introduit pour analyser des données issues des puces à ADN. Il connaît un engouement extraordinaire et son immense succès s'explique par un temps d'exécution raisonnable quant à la prise de décision. Compte tenu des inconvénients associés aux techniques de classification, le regroupement des profils de gènes reste une préoccupation majeure.

A l'inverse des ordinateurs classiques, la puissance d'un système de calcul amorphe est justifiée par le nombre de processeurs et leur fonctionnement asynchrone. Avec un tel système, le coût est nettement réduit. A mesure que le choix d'une puce à ADN reste problématique, son intégration avec un système de calcul amorphe peut contribuer à la classification des expressions de gènes. C'est dans ce contexte que s'inscrit l'idée proposée qui confère à un ensemble de particules de calcul amorphe la tâche de la classification des données issues des puces à ADN. En effet, quand on s'adresse à un système de calcul amorphe, le défi de proposer une technique adéquate reste un fait d'actualité. Vis-à-vis des contraintes imposées dans un tel système, nous avons adapté l'algorithme du GNG procédant par apprentissage incrémental et n'exigeant pas une structure préalable des neurones. D'après les résultats obtenus en respectant les caractéristiques d'un système de calcul amorphe, la convergence de l'approche proposée a été vérifiée en distinguant quelques clusters de gènes.

Conclusion générale

Nous avons présenté le calcul amorphe, dont la recherche doit continuer pour sensibiliser les concepteurs des systèmes distribués de son apport en offrant un cadre méthodologique quant à la programmation de nouveaux systèmes distribués robustes et dépassants le problème de coût. A partir de l'état de l'art présenté, nous constatons aisément que les chercheurs du groupe MIT ont pu formuler un paradigme s'inspirant de mécanismes naturels qui sera d'un intérêt évident pour d'autres domaines connexes (nanorobotique, réseaux de sondes, applications industrielles pharmaceutiques, ...). Ainsi, le calcul amorphe s'inscrit dans le cadre d'un projet plus vaste que les systèmes multi agents réactifs. Depuis 1997, plusieurs langages et environnements tels que GPL, OSL, RSEAM et HLSIM ont permis d'enrichir la littérature concernant le calcul amorphe. Ils permettent à un novice de manipuler concrètement cet axe de recherche. Il s'agit d'écrire une spécification globale indépendamment des particules de calcul amorphe. Les compilateurs développés s'en chargent eux-même de la génération des interactions entre les particules de calcul amorphe en s'appuyant sur des primitives de base. En parallèle, d'autres métaphores et approches basées calcul amorphe sont à exploiter.

Notre contribution a porté sur deux facettes : théorique et expérimentale. Nous avons essentiellement situé le calcul amorphe par rapport aux autres paradigmes. En plus, nous avons manipulé l'environnement HLSIM et Netlogo. A cela, s'ajoute l'exploitation de l'hierarchie de calcul amorphe et l'algorithme du GNG pour la classification de données. Dans ce cadre, deux applications ont été conçues et développées : la segmentation d'images et la classification des données issues des puces à ADN. Le noyau de notre réflexion consiste en la traduction du processus de classification en terme d'interactions locales entre des particules de calcul amorphe organisées selon une hiérarchie et communiquant par diffusion. A travers les simulations effectuées, nous nous sommes convaincus que le schéma d'utilisation du calcul amorphe doit modifier impérativement la conception de nos programmes. A partir des justifications présentées, on s'est orienté vers l'adaptation d'une approche classique (GNG) dans un contexte de calcul amorphe.

La segmentation d'images, l'une des priorités lors d'analyse d'images, doit distinguer les objets pertinents dans une image. Elle doit obéir à un formalisme facilitant sa réalisation. Adoptons le GNG et le calcul amorphe, l'apprentissage des contours ou régions a bénéficié d'un nombre réduit d'itérations. En effet, la contribution proposée exerce son fonctionnement grâce aux interactions asynchrones entre les entités du calcul amorphe. Malgré que la classification a porté sur les particules de calcul amorphe, les contours et les régions ont été globalement identifiés. Fruit de la convergence des travaux chimiques, biologiques et informatiques, les puces à ADN ont permis l'analyse simultanée d'un échantillon de gènes. Sous l'impulsion de quelques techniques de classification, le regroupement des gènes similaires s'articule autour de SOM et le K-means. Face aux inconvénients accompagnant ces deux algorithmes, nous avons exploré;

pour la première fois, l'algorithme du GNG pour la classification des données issues des puces à ADN. En utilisant le calcul amorphe, la qualité de classification est si importante. Il convient aussi de signaler la capacité des particules de calcul amorphe à apprendre des valeurs complexes des expressions de gènes afin d'émerger la classification des expressions. Notons que l'interprétation des groupes de gènes obtenus nécessite la collaboration d'un biologiste. Vis-à-vis des contraintes d'un système de calcul amorphe, les résultats obtenus pour le cas de la segmentation d'images et les puces à ADN sont encourageants. En fait, on peut se satisfaire d'une grille irrégulière et des interactions asynchrones pour la classification de données. Mais bien sûr, l'idéal consiste à définir une approche purement amorphe.

Une poursuite envisageable de ce travail constituera :

- ❖ Implémenter l'approche proposée sous l'environnement HLSIM pour garantir tous les aspects du calcul amorphe.
- ❖ Optimiser la méthode proposée.
- ❖ Penser à la résolution d'autres problèmes complexes par calcul amorphe.

Références bibliographiques

[1] S.Meshoul, "Optimisation par les systèmes complexes pour le recalage et la mise en correspondance en analyse d'images", thèse de Doctorat d'état, Université Mentouri (Constantine), Juin 2004.

[2] P-A.Favier, "La notion d'intentionnalité dans la spécification de comportements d'agents autonomes situés en environnement virtuel", thèse de Doctorat, Université de Bretagne Occidentale, Décembre 2004.

[3] C.Gershenson, F.Heyligen, 'How can we think the complex?', to be published in: Kurt A Richardson (ed.): Managing the complex: Philosophy, Theory, and Practice, (Institute of the study of the coherence and Emergence/Information Age Publishing), 2004.

pespmc1.vub.ac.be/Papers/ThinkingComplex.pdf

[4] H.Parunak, S.Brueckner, 'Engineering swarming systems', Forthcoming in F.Bergenti, M-P.Gleizes, F.Zambonelli, eds., Methodologies and software engineering for agent systems. Fluwer, 2004

<http://www.erim.org/~vparunak/MSEASO3.pdf>

[5] A.Authier, "Cristallographie Géométrique", Techniques de l'ingénieur, Volume AFP2, pp A1305-1/A1305-27, Janvier 2005.

[6] W.H.Braggs, W.L braggs, " Solids/Crystal structures and Solutions"

crystals.llnl.gov/pdf/crys-solids.pdf

[7] J.Marc di Meglo, "La Matière Molle", Techniques de l'ingénieur, Volume AFP2, pp A1195-1/1195-8, Janvier 2005.

[8] E.Adam, P.Biuffandeau, A.Darraspen, R.Fix, S.Jouve, O.Welker, "Méthodes d'Analyse de Composants Amorphes", école des Mines, Saint-Etienne

www.emse/soms6/fichespdf/amorph.pdf

[9] P.Chabloz, "Les Couches épaisses en Silicium Amorphe : application comme détecteur de rayon X", thèse de Doctorat en sciences, Ecole polytechnique de Lausanne, 1996.

[10] état cristallin, état amorphe

etab.ac-orleans-tours-fr/lyc-dessignes-blocs/cristallo/etats.html

[11] H. Abelson, D. Allen, D. Coore, C. Hanson, G. Homsy, T. Knight, Jr., R. Nagpal, E. Rauch, G. Sussman, R. Weiss, 'Amorphous Computing', *AI MEMO 1665*, MIT, August 1999.

[12] J.Beal, 'A Robust Amorphous Hierarchy From Persistent Nodes', *AI Memo 2003-012*, MIT, April 2003.

[13] R.Nagpal, M.Mamei, 'Engineering Amorphous Computing Systems', Invited chapter in Methodologies and software engineering for agent systems, editors Bergenti, Gleizes, Zambonelli, Kluwer Academic Publishing, (in press) 2003.

polaris.ing.unimo.it/MOON/papers/pdf/amo2004.pdf

[14] P.Haddow, P-V.Mortel, 'From here to there: Futute robust ehw technologies for large digital designs', in D.Keymeulen, A.Stoica, J.Lohn & R.S.Zebulum, editors, Proc. *The Third NASA/DoD Workshop on Evolvable Hardware*, EH 2001, pp 232-239, IEEE Computer Society, 2001.

[15] D.Coore, Radhika Nagpal, R.Weiss, 'Paradigms For Structure an Amorphous Computer', *AI Memo 1614*, MIT, June 1997.

[16] E. D'Hondt, "Exploring The Amorphous Computing Paradigm", Master's thesis in computer science, Vrije University (Brussel), August 2000.

[17] E.Rauch, 'Discrete amorphous phenomena'

<http://www.swiss.ai.mit.edu/projects/amorphous/>, Master thesis, 1999

[18] A.Sutherland, "Towards RSEAM: Serial Execution on Amorphous Machines", Master's thesis, MIT, June 2003.

[19] Z.Nehme, 'L'informatique Pervasive', Techniques de l'ingénieur, Vol HB1, pp RE 12-1/ RE 12-6, Novembre 2003.

[20] N.Davies, H. Gellerso, *in Proceedings of the IEEE Pervasive Computing*, pp 26-35, 2001.

[21] T.Starner, 'The Challenge of Wearable Computing: part1', *in Proceedings of the IEEE MICRO*, pp 44-22, 2001.

[22] Kevin M.Baird, "Evaluating the Effectiveness of Augmented Reality and Wearable Computing for a Manufacturing Assembly Task", Thesis, Virginia Polytechnique Institute and State University, June 1999.

[22] K.Diatchka, "wearable Computing and Personal Health Monitoring Systems"

ece399/f03/explo/diatchka.pdf

- [24] J.Hill, “A Software Architecture Supporting Networked Sensors”, master’s of science, University of California, Berkely, 2000.
- [25] R.Krishnan, “Efficient Self-organization of Large Wireless Sensor Networks, Dissertation, Boston University, 2004.
- [26] A.Pilpré, “Self-Properties of Multi Sensing Entities in Smart Environments”, Master’s Thesis, MIT, June 2005.
- [27] W.Buthera, “Programming a Paintable Computer”, thesis, Media Lab, MIT, 2002.
- [28] W.Buthera, “Paintable Computing”, thesis’s proposal, Media lab, MIT, 2002.
- [29] M.Joseph, “Localization and Sensing Applications in the Pushing Computing Network”, Master’s of Engineering, MIT, February 2005.
- [30] J.Lifton, “Pushing Computing”, Thesis, MIT, 2002.
- [31] P.Seetharamakrishnan, “C@t: A Language for Programming Massively Distributed embedded Systems”, Master’s thesis, MIT, 2002.
- [32] J. Stankov, T. Abdelazhar, C. Lu, L. Sha, J. Hon, ‘Real Time Communication and Coordination In Embedded Sensor Networks’, *proceedings of the IEEE*, vol 91, NO7, pp 1002-1024, July 2003.
- [33] R.Weiss, “Cellular Computation and Communications using Engineered Genetic Regularity Networks”, Thesis, MIT, September 2001.
- [34] V.Sharma, ‘Is it possible to build computers from Living Cells?’, *Biotech Journal*, Vol 2, pp 53-60, 2004.
- [35] E.Tranvouez, “IAD et ordonnancement: une approche coopérative de réordonnement par système multi agents”, Thèse de Doctorat en sciences, Université d’Aix Marseille 3(France), May 2001.
- [36] D.Servat, A.Drogoul, ‘Combining Amorphous Computing and Reactive Agent: A paradigm for Pervasive Intelligence?’, *in Proceedings of AAMAS 2002 Conference*, pp 441-448, Bologna(Italy), 2002.
- [37] C.Hanson, “Mit Scheme Reference manual”, Edition 1.94, July 2001.
- [38] A.Stephen, “High Level Simulator”, Internal publication, MIT, 1997
- [39] D.Coore, “ Botanical Computing, a Development Approach to Generating Interconnect Topologies on an Amorphous Computer”, Thesis, MIT, February 1999.
- [40] R.Nagpal, “Programming Self-Assembly: Constructing Global Shapes using Biologically-Inspired Local Interconnections and Origami Mathematics”, Thesis, MIT, 2001.

- [41] R.Nagpal, 'Programmable Self-Assembly Using Biologically-Inspired Multi agent Control', *in Proceedings of the AAMAS 2002 Conference*, pp.418-425, Bologna (Italy), 2002.
- [42] H.Abelson, 'Amorphous Computing', *Communications of the ACM*, Vol 43, N° 5, pp 74-82, May 2004.
- [43] A. Kondacs, 'Biologically Inspired, Self-Assembly Of Two-Dimensional Shapes Using Global To Local Compilation', *International Joint Conference On Artificial Intelligence (IJCAI)*, 2003.
- [44] C.Frost, "Amorphous Shape Mapping", Thesis, Virginia University, May 2004.
- [45] S.George, D.Evans, S.Marchette, 'A Biologically Programming Model for Self Healing', *in Proceedings of SSRS'03*, October 2003.
- [46] M.Mamei, F.Zambonelli, L.Leonardi, 'Co fields: an approach for motion coordination', Co fields, *Pervasive computing*, Vol 3, N° 2, pp 52-61, June 2004.
- [47] K. Stoy, R. Nagpal, 'Self-Repair through Scale Independent Self-Reconfiguration', *in Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai (Japan), October 2004.
- [48] W-M.Shen, P-Will, A.Galastyan, C-M.Chuong, 'Hormone-Inspired Self-Organization and Distributed Control of Robotic Swarms', *in Proceedings of the Autonomous Robots*, Vol 17, pp 93-105, Netherlands, May 2004.
- [49] G. Bour, G. Huzler, B. Gortais, 'Ambient Cognitive Environments and the Distributed Synthesis of Visual Ambiences', *to be published in Engineering Self-Organizing Applications ESOA'04 Conference*, Lecture Notes in Computer Science, Springer Verlag, Berlin, 2004.
- [50] L. Gang, J. Liu, X. Jin, 'Agent-Based, Energy Efficient Routing In Sensor Networks', *in Proceedings of the AAMAS'04*, Hong Kong (Japan), 2004.
- [51] M.Schubert, "Advanced Data Mining techniques for compound objects", Dissertation, Munchen University, November 2004.
- [52] Clustering 'chapter2'
- www.newcastle.edu.au/services/library/adt/uploads/approved/ast-NNCU2005309.183528/restricted/02chaps1-4.pdf
- [53] Y.Yang, "Data clustering for autonomic application replication", Master's thesis, Vrije University, August 2000
- [54] T.Guerram, "Une approche Datamining basée agents", Mémoire de Magister, Université Mentouri, Juillet 2003.

- [55] C-S. Mooler Levet, “Modelling and Clustering of gene expression time-series”, Thesis, Manchester University, 2005.
- [56] A-K.Jain, M-N.Murty, P-J.Flynn, ‘Data clustering : A review’, *ACM computing surveys*, Vol 31, N° 3, pp 265-323, September 1999.
- [57] J.Tsung-Hsein Tsai, “Bioinformatics data mining and database interface design/coding”, Thesis, University of Queensland, October 2001.
- [58] M.Schubert, “Advanced Data Mining techniques for compound objects”, Dissertation, Munchen University, November 2004.
- [59] Cartes auto-organisatrices
www-rdf.efrei.fr/LivreRNchap07.pdf
- [60] T.Lio, “Image segmentation and region merging“, Thesis, Monash University, November 2003.
- [61] A.Moussaoui, “Segmentation et partitionnement de l’encephate humaine en IRM 3D: étude de modèles et applications”, thèse de Doctorat, Université Ferhat Abbas(Sétif), 2005.
- [62] D-K.Van Laerhoven, “On-Line Adaptative Context Awareness Startend Vanuit Low-Level Sensoren“, Thesis, Vrije University, May 1999.
- [63] J. Neumann, “Holistic Processing Of Hierarchical Structures In Connectionist Networks”, Phd thesis, University of Edinburgh, 2001.
- [64] H.Talbi, “Apprentissage par réseaux de neurones chaotiques“, Mémoire de magister, université Mentouri(Constantine), 2003.
- [65] M.Strickert, “Self-Organizing Neural Networks for Sequence Processing“, Thesis, University of Osnabrûch (Germany), June 2004.
- [66] B.Fritzke, ‘Unsupervised ontogenic networks’, *handbook of neural computation*, **C2.4:1-16**, 1997.
- [67] B.Fritzke, ‘A Growing Neural Gas Network Learns Topologies’, publié dans *Advances in Neural Information Processing Systems 7*, MIT Press, Cambridge MA, pp 625-632, 1995.
- [68] J. Holmstrom, “Growing Neural Gas experiments with GNG, GNG with utility and supervised GNG”, Uppsala Master’s thesis in computer science, Uppsala University, Sweden, 2002.
- [69] L.Lella, I.Licat, ‘Evolutionary Neural Gas: A scale-free-self organizing Neural Net without Topological Constraints’, Short communication, pp 43-53, 2004.
- [70] Y.Prudent, A.Ennaji, ‘A new learning algorithm for incremental self organizing map’, *ESANN’2005*, Proceedings ‘European Symposium on Artificial Neural Networks’, Bruges(Begium), ISBN 2-930307-05-6, April 2005.

- [71] I.P.Ivrissimtzis, W-K.Jeong, H-P.Seidel, ‘Tree-GNG-Hierarchical topological clustering’, *in Proceedings of the ESANN2005 Conference*, Belgium, April 2005.
- [72] D.Deng, N.Kababov, ‘Evolving self-organizing maps for on line learning, data analysis and modelling’, Discussion paper series, ISSN 1172-6024, University of Otago, March 2002.
- [73] D.Vasquez, T.Fraichard, O.Aycard, V.Laugier, ‘Intentional Motion ON-Line Learning and prediction’, *in Proceedings of the International conference on field and service robotics*, Prt Douglas, July 2005.
- [74] M.A Montes de Oca, L.Garrido, J.Aguirre, ‘An hybridization of an ant based clustering algorithm with growing neural networks for classification tasks’, *in Proceedings of the SAC’05*, Santa fe , USA, March 2005.
- [75] I.P.Ivrissimtzis, W-K.Jeong, H-P.Seidel, ‘Neural Meshes : statistical learning methods in surface reconstruction’, Germany Research report, *MPI-I-2003-4-007*, April 2003.
- [76] D.N-Kasabov, ‘Evolving Localized Learning For On-Line Colour Image Quantization’, The information science discussion paper series, ISSN 1172-6024, November 2000.
- [77] G.Westermann, ‘A constructive neural network learns the past tense of English verbs’, in *proceedings of the Gala 97 conference on language acquisition*, pp 393-398, Edinburgh, 1997.
- [78] J.Li, T.Duckett, ‘Growing RBF networks for learning reactive behaviors in mobile robotics’, *to appear in International journal of vehicle autonomous systems (IJVAS)*, 2004.
- [79] J.Nilsson, ‘methods for classification of gene expressions’, Master’s thesis, Lund University, 2002.
- [80] D.Larlus-Larrando, ‘La segmentation d’images’, rapport de Master, INRIA, 2004.
- [81] Segmentation d'images : principes Détection de vallées
glotin.univ-tln.fr/M1_OP11/ Segmentation_images_principes.pdf
- [82] D.MERAD, ‘Reconnaissance 2D/2D et 2D/3D d’objets à partir de leurs squelettes’, thèse de Doctorat, Université d’Evry-Val d’Essonne(France), Décembre 2004.
- [83] J.Baillie, “Segmentation“, Module D9 , ES322- Traitement d’images et vision artificielle, ENSTA, 2003.
- [84] L.Olivier, “La réalisation d’une étude sur le tri informatique en cytologie“, Mémoire de DEA, Université de Caen, 95.
- [85] A.COUTANT, “La méthode des contours actifs en traitement des images“, Mémoire pour l’examen probatoire en Calcul Scientifique, février 2005.
- [86] J.Montagnat, “Modèles déformables pour la segmentation et la modélisation d’images médicales 3D et 4D“, Thèse de Doctorat, Université Nice Sophia-Antipolis, Décembre 1999.

- [87] B.Scherrer, “Segmentation Markovienne Distribuée et Coopérative des Tissus et Structures présents dans des IRM Cérébrales“, Mémoire de Master, Université Joseph Fourier de Grenoble, Juin 2005.
- [88] P.Bertolino, “Contribution des pyramides irrégulières en segmentation d’images multiresolution“, Thèse de Doctorat, Institut national polytechnique de Grenoble, 1995.
- [89] P.Andrés Arbaléaz, L-D.Cohen, ‘Segmentation d’images couleur par partitions de Voronoi’
www.ceremade.dauphine.fr/~cohen/mypapers/PabloTS05.pdf
- [90] J.Shi, J.Malik, ‘Normalized Cuts and Image Segmentation’, *IEEE transactions on pattern analysis and machine intelligence*, VOL. 22, NO. 8, pp 888-905, August 2000.
- [91] T.Zouagui, “Approche fonctionnelle générique des méthodes de segmentation d’images “, Thèse de Doctorat, Institut nationale des sciences appliquées de Lyon, Août 2004.
- [92] H.Settache, “Une plate forme multi agents pour la segmentation d’images : application dans le domaine des IRM cérébrales 2D“, Rapport de stage de DEA, Université de Caen, Septembre 2002.
- [93] C.Bourjot, V.Chevrier, ‘De la simulation de la construction collective à la détection des régions dans un image à niveau de gris : l’inspiration des araignées sociales ‘,*In Proceedings of FFIADSM 2001 Conference*, Montreal,(Canada), 2001.
- [94] Chapter 16 Segmentation Using Clustering Methods
descai.urg-es/mia/complementario/t1/book3draft/clustering.pdf
- [95] A.Moussaoui, K.Benmahammed, N.Ferahta, C.Chen, ‘A New MR Brain Image Segmentation Using An Optimal Semi-Supervised Fuzzy C-Means and Pdf Estimation’, *Electronic letters On Computer Vision and Image Analysis*, 5(4) :1-11,2005.
- [96] C. Charrier , O. Lezoray, ‘Segmentation d’images couleur par la théorie de l’évidence’, *in Proceedings of COREASA’05 Conference*, Lyon (France), 2005.
- [97] N.Bouloudani, P.Lambert, D.Coquin, ‘Segmentation automatique des images couleurs à base d’indicateurs de performance’, *in Proceedings of the CORESA’03 Conference*, Lyon, Janvier 2003.
- [98] S.Benelmadjat , “Implémentation d’une méthode de segmentation couleur“, Rapport de projet, 2005.
- [99] S.Chelloug, S.Meshoul, M.Batouche, ‘Clustering using the growing neural gas algorithm within the amorphous computing paradigm’, *in Proceedings of the International Workshop on Text Image and Speech Recognition TISR’05 Conference*, pp 38-45, Annaba, December 2005.
- [100] P.Soularue, X.Gidrol, ‘Puces à ADN’, Techniques de l’ingénieur, Volume P5, pp RE6-1/RE6-10, Juin 2002.

- [101] I.Quinkal, ‘Quelques termes-clefs de Biologie Moléculaire et leur définition’, Rapport, INRIA, Septembre 2003.
- [102] A.Clare, “Machine learning and data mining for yeast functional genomics“, Thesis, University of Wales, February 2003.
- [103] J.Blachly, “Mathematic’s emergent role in analysis of gene expression“, Thesis, University of Arkansas, 2003.
- [104] A.Layeb, “Approche quantique évolutionnaire pour l’alignement multiple de séquences en bioinformatique“, Mémoire de Magister, Université Mentouri (Constantine), Novembre 2005.
- [105] “ARN cible et agent thérapeutique“, Biofutur, N°255, Mai 2005.
- [106] “La bioinformatique à L’INRIA“, Rapport de recherche, Septembre 2003.
- [107] C.Abou, C.Leger-Waltz, “Les biopuces“, Dossier CEA, 2002.
- [108] “La révolution des biopuces“, CEA, Octobre 2002.
- [109] “Bioinformatique et Post-génome“, N°252, Février 2005.
- [110] N.Reymond, “Bioinformatique des puces à AND et application à l’analyse du transcriptome de Buchnera aphidicola”, thèse de Doctorat, Institut national des sciences appliquées de Lyon (France), Décembre 2004.
- [111] J.Mary, “Axe bioinformatique recherche d’attributs pertinents pour l’apprentissage de régularités en génomique“, Stage de DEA, Février 2004.
- [112] Hybridation notes
www.iro.umontreal.ca/csuros/IFT6100/materiel/hybridation-notes.pdf
- [113] L.Wehenkel, « Application de l’apprentissage automatique à l’extraction et la gestion de connaissances des DNA arrays », Mémoire, Université de Liège, 2003.
- [114] E-F.Nuwaysir, ‘Toxicogenomics in the trenches’, Continuing education course, PM-16, National Institute of Environmental health sciences, March 2000.
- [115] W.Lin, “les puces à ADN sur lames de verre : principes et méthodes de confection, l’application expérimentale et d’analyse de données’, extraites de thèse de doctorat : ‘Applications de la technologie des puces à ADN à l’étude de la différenciation méiotique et des mécanismes de recombinaison chez la levure Saccharomces cerevisie“, CNRS, Avril 2004.
- [116] T.Pasanen, J.Saarela, I.Saarikko, T.Toivanen, M.Tolvanen, M.Vihinen, G.Wong, ‘DNA Microarray Data Analysis’, Part 1, Editors Jarno Tuimala And M. Minna Laine CSC, ISBN 952-9821-89-1, 2003.

- [117] B.Chen, P.C.Tai, R.Harrion, Y.Pan, ‘Novel hybrid hierarchical-K-means clustering method (H-K-means) for microarray analysis’, *in Proceedings of the 2005 IEEE Computational Systems Bioinformatics Conference (CSB2005)*, California, August 2005.
- [118] J.Kasturi, R.Acharya, M.Ramanatan, ‘An information theoretic approach for analysing temporal patterns of gene expression’, *in the Proceedings of the Bioinformatics Conference*, Vol .19 n°4 2003, pp 449-459 October 2002.
- [119] H.Li, K.Zhang, T.Jiang, ‘Minimum entropy clustering and applications to gene expression analysis’, *in Proceedings of the 2004 IEEE Computational Systems Bioinformatics Conference (CSB2005)*, California, August 2005.
- [120] E.V-L-V. Themaat, “On the use of learning Bayesian Networks to analyze gene expression data : classification and gene network reconstruction“, Master’s thesis, University of Amsterdam, June 2005.
- [121] J.Herrero A.Valencia, J.Dopazo, ‘A dynamically growing self-organizing tree for hierarchical clustering gene expression profiles’, *in the Proceedings of the Bioinformatics*, Vol 17 , N° 2, pp 126-136 , 2001.
- [122] S.Chelloug, S.Meshoul, M.Batouche, ‘Clustering microarray data within the amorphous computing paradigm and the growing neural gas’, *in Proceedings of the IEA-AIE’06 Conference*, LNCS Series, France (to appear).

Résumé

Ce mémoire traite le problème de la classification sous l'angle du calcul amorphe. En présence d'une multitude d'éléments de calcul, positionnés irrégulièrement, communiquant localement par diffusion et n'ayant aucune information concernant leur environnement, le calcul amorphe vise l'émergence d'un comportement global cohérent. Ce paradigme apporte une solution quant à la conception d'un système distribué fiable en s'appuyant sur les principes d'auto organisation ou d'auto assemblage et en évitant l'assignation d'identités aux entités de calcul amorphe. Afin de simuler la classification des particules de calcul amorphe pour la segmentation d'images, l'algorithme du Growing Neural Gas (GNG) a été adapté. Le GNG possède quatre facettes: un modèle d'auto organisation, facile à utiliser, procède par classification locale et n'exige aucune structure préalable. L'utilisation conjointe du calcul amorphe et le GNG nécessite alors la construction d'une hiérarchie de calcul amorphe. Cette réflexion nous a ainsi amené à démontrer la classification de données issues des puces à ADN par calcul amorphe. Une puce à ADN correspond à un système miniaturisé permettant l'analyse simultanée d'un échantillon de gènes. Cependant, les gènes présentent quelques caractéristiques complexes et les données obtenues sont entachées de bruit. Par conséquent, la recherche d'une méthode de classification servira à un diagnostic efficace. A travers les résultats obtenus par la plate forme Netlogo, nous avons répondu aux contraintes d'un système de calcul amorphe. Par rapport à l'algorithme du GNG classique, cette approche a bénéficié d'un nombre réduit d'itérations pour concrétiser le résultat escompté. D'une part, la convergence de la méthode mise en œuvre a abouti à l'établissement d'un graphe regroupant les particules de calcul amorphe représentant un contour ou une région homogène. D'autre part, la qualité de partitionnement d'un échantillon de gènes est si importante. La motivation principale est liée au fonctionnement asynchrone des particules de calcul amorphe.

Mots clés : calcul amorphe, auto organisation, hiérarchie du calcul amorphe, GNG, classification, puce à ADN, Netlogo.