

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Med Khider de Biskra
Faculté des Sciences exactes et des sciences de la nature et de la vie

Département d'Informatique

N° d'ordre :.....

Série :.....



Mémoire

En vue de l'obtention du diplôme de Magister en informatique

Option : Synthèse d'image et vie artificielle

L'Ontogenèse pour la création de créatures artificielles.

Présenté par : Mlle. Nedjma DJEZZAR

Dirigé par : Pr. NourEddine Djedi

Mémoire soutenu le : .../.../2010

Composition de jury :

Président :	Pr BOUFAIDA Mahmoud,	Professeur,	Président	Université de Constantine
Rapporteur :	Pr DJEDI NourEddine	Professeur,	Rapporteur	Université de Biskra
Examineur :	Dr BABAHENINI Med Chaouki	MCF A	Examineur	Université de Biskra
Examineur :	Dr CHERIF Foudil	MCF A	Examineur	Université de Biskra
Examineur :	Dr MOUSSAOUI A/Ouahab	MCF A	Examineur	Université de Sétif

Remerciements.

En premier lieu, Je tiens à exprimer tous mes remerciements à mon encadreur Mr NourEddine Djedi, Professeur à l'université de Biskra de m'avoir fait l'honneur de diriger ce travail et dont l'aide précieuse m'a été indispensable sur le plan scientifique. Je suis très touchée par sa compréhension, sa sympathie, sa disponibilité, sa conscience scientifique, et sa rigueur dans le travail, qui font de lui un encadreur exemplaire, c'est le moins qu'on puisse dire. Qu'il soit ici assuré de toute mon estime et de mon profond respect.

Je tiens également à exprimer toute ma reconnaissance et ma gratitude à mon co-encadreur Mr Yves Duthen, Professeur à l'université de Toulouse 1, de m'avoir proposé un sujet si passionnant, pour son aide scientifique inestimable, ainsi pour m'avoir accueilli chaleureusement dans son équipe de recherche. Je suis également très sensible à l'honneur qu'il me fait en codirigeant mon mémoire. Et Je le remercie pour ses suggestions, ses critiques et ses encouragements qui ont permis l'amélioration de ce manuscrit. Qu'il soit ici assuré de mon très grand respect.

Je remercie très sincèrement l'ensemble des membres jury qui me font le grand honneur d'avoir accepté de juger mon travail.

Un remerciement très particulier à Mr Mahmoud Boufaïda, Professeur à l'université de Constantine, pour l'honneur qu'il me fait en acceptant la présidence de ce jury malgré ses nombreuses responsabilités. Je tiens aussi à le remercier pour les précieuses remarques qui m'a donné au cours de mes deux dernières années de mon cycle d'ingénieur qui sont et qui resteront toujours d'une grande utilité pour ma carrière scientifique. Qu'il soit ici assuré de toute mon estime et de ma profonde reconnaissance.

J'adresse mes très vives remerciements à Mr Babahenini Med Chaouki, maître de conférences à l'université de Biskra, à Mr Cherif Foudil, maître de conférences à l'université de Biskra, et à Mr Moussaoui A/Ouahab, maître de conférences l'université de Sétif pour avoir eu l'amabilité de participer à mon jury de soutenance.

Ma plus grande reconnaissance à Sylvain Cussat-Blanc pour ses précieux conseils et ses encouragements qui m'ont permis de progresser. L'aide qui m'a procurée ainsi que nos nombreuses discussions ont contribué énormément à l'amélioration de mon travail.

Je remercie vivement tous les membres du laboratoire LESIA.

J'exprime toute ma reconnaissance également à Mr Jean-Pierre Jessel pour m'avoir accueilli chaleureusement au sein de l'équipe VORTEX ainsi à tous les membres de l'IRIT-UT1 en particulier Mr Hervé Luga, aussi aux thésards : Nicolas, Jonathane, Paul, Tristram, David et Vincent qui m'ont permis de travailler dans un environnement si sympathique au cours de mes deux stages effectués au sein de l'équipe VORTEX de l'IRIT. Je tiens également à remercier tous ceux qui n'ont pas leur nom dans ce paragraphe et qui pensent devoir être remerciés dans ce mémoire.

Mes remerciements vont également à Joseph El Gemayel, Deneche Abdelhakim et Stéfan Bornhofen, qui ont répandu à toute aide demandée en phase de conception ou de réalisation de mon projet.

J'adresse un grand merci à toute ma famille, en particulier mon père, ma sœur et tata, les plus chers au monde qui m'ont permis de mener à bien mes études.

Enfin, je remercie toutes les autres personnes qui m'ont aidé de près ou de loin et que je n'ai pas citées ici.

Le travail réalisé dans ce mémoire a été effectué dans le cadre de l'accord de coopération bilatérale Algéro-Français TASSILI-CMEP N° 08MDU746, géré par l'Egide.

*Au meilleur père du monde,
mon père.*

Résumé.

Dans le but de synthétiser différentes créatures artificielles pour peupler des mondes virtuels ou les reproduire sous forme de robots, nous proposons un modèle hybride ontogénétique et phylogénétique pour générer des créatures complètes possédant un métabolisme, une morphologie et un comportement. Les modèles issus de l'ontogenèse artificielle simulent le développement d'organismes multicellulaires en partant d'une cellule unique. Notre travail a pour but de générer, dans un premier temps, des organismes simples servant à définir par la suite des créatures complètes. Ainsi, nous proposons dans ce mémoire, un modèle bio-inspiré unifiant les approches divisant l'ensemble des systèmes d'ontogenèses artificielles : grammaticales et celles basées sur la chimie cellulaire. Pour la partie grammaticale, nous proposons une variante des L-systèmes paramétriques (APL-système) afin de simuler d'une manière adaptative la morphogenèse des organismes en fonction de l'état interne des cellules les composent. Les organismes développés possèdent un métabolisme utilisant des substrats de l'environnement pour évoluer, croître et agir. De plus, ils sont capables de s'auto-réparer. Ainsi, si certaines cellules de l'organisme sont détruites après ou même durant la phase de développement, le modèle est capable de régénérer la structure endommagée. Les résultats obtenus par simulation sont soumis à une comparaison critique avec les modèles de références présentés dans l'état de l'art. A notre connaissance, notre modèle représente la première utilisation de L-systèmes pour orienter la morphogenèse d'une créature multicellulaire artificielle.

Mots clés : ontogenèse artificielle, Phylogenèse artificielle, métabolisme, lignée cellulaire, chimie artificielle, APL-systèmes.

Abstract.

With an aim of synthesizing various artificial creatures to populate virtual worlds or to reproduce them in the form of robots, we propose an ontogenetic and phylogenetic hybrid model to generate complete creatures having a metabolism, a morphology and a behaviour. The models resulting from artificial ontogenesis simulate the development of multicellular organisms on the basis of a single cell. The purpose of our work is to generate, initially, of the simple organisms being used to define complete creatures thereafter. Thus, we propose in this memory, a bio-inspired model unifying the approaches dividing the whole of the systems of artificial ontogenesis: grammatical and those based on cellular chemistry. For the grammatical part, we propose an alternative of the parametric L-systems (APL-system) in order to simulate in an adaptive way morphogenesis of the organisms according to the internal state of cells compose them. The developed organisms have a using metabolism of the substrates of the environment to evolve/move, grow and act. Moreover, they are able car-to be repaired. Thus, so certain cells of the organisms are destroyed afterwards or even during the phase of development, the model is able to regenerate the damaged structure. The results obtained by simulation are subjected to a critical comparison with the models of references presented in the state of the art. To our knowledge, our model represents the first use of the L-systems to direct the morphogenesis of an artificial multicellular creature.

ملخص.

في سبيل تشكيل مخلوقات اصطناعية قصد تجميعها على صورة عوالم افتراضية مختلفة أو من أجل إنتاج أشكال من الروبوتات، نقترح نمودجا هجينا في سبيل استعماله لإيجاد مخلوقات اصطناعية لديها كل مقومات النمو، النشوء والتطور الكامل مع التمثيل الغذائي، والتشكل والسلوك. تسمح النماذج النابعة من تطور الخلية البيولوجي الاصطناعي بتمثيل كائنات حية متعددة الخلايا، وهذا انطلاقا من خلية واحدة. يهدف عملنا الى توليد ، كخطوة أولى، كائنات بسيطة و التي ستستخدم بدورها لإيجاد مجموعة كاملة من المخلوقات.

نقترح في هذه الأطروحة، نمودجا ملهم بيولوجيا قصد توحيد النظامين المعمول بهما الشكلي القائم على الخلية، من جهة والكيميائي، من جهة أخرى.

بالنسبة للجزء المتعلق بالشكل، نقترح بديلا لنظام L-system و هو النظام المسمى APL-system وذلك للمحاكاة بطريقة تكيفية لنظام morphogenesis الخاص بالنظم وفقا لحالة الداخلية للخلايا التي تشكلها.

و تمتلك الكائنات المنشأة نظاما داخليا يسمح لها باستخدام الركائز البيئية قصد التطور، النمو والعمل. و هي علاوة على ذلك، قادرة على القيام بإصلاح آلي لنفسها. و هكذا، إذا تم تدمير خلايا معينة لأي سبب كان خلال عملية الإنشاء لهذه الكائنات أو حتى خلال مرحلة من مراحل نموها تتم عملية إعادة إنشاء البنية المتعرضة للعطب.

وتخضع النتائج التي تحصلنا عليها لمحاكاة مقارنة حاسمة مع النماذج الموجودة في إطار الدراسات المنجزة. و على حد علمنا، يمثل النمودج الذي اقترحناه أول تطبيق لتقنية توجيه النمو (morphogenesis) على مخلوقات اصطناعية متعددة الخلايا.

Tables des matières

Introduction générale	1
1.1 Contexte générale.	1
1.2 Problématique.....	2
1.3 Contribution.....	3
1.4 Structure du mémoire.....	4
Chapitre 2 : La vie artificielle	6
2.1 Introduction.....	6
2.1.1 Origines et objectifs.....	6
2.1.2 Définition.....	9
2.1.3 Emergence et auto-organisation.....	10
2.2 Morphogenèse.	10
2.2.1 Automates cellulaires.....	10
2.2.2 Fractales.....	12
2.2.3 L-systèmes.....	12
2.3 Evolution et adaptation.....	17
2.3.1 Algorithmes évolutionnistes.....	17
2.3.2 Boucle générationnelle.....	19
2.3.3 Synthèse.....	23
2.4 Apprentissage.....	24
2.4.1 Réseaux de neurones.....	24
2.4.2 Systèmes de classeurs.....	27
2.5 Créatures Artificielles.	30
2.4.3 Epigenèse.....	30
2.4.4 Phylogénèse.....	32
2.4.5 Ontogenèse.....	39
2.6 Conclusion.....	40
Chapitre 3 : L'ontogenèse artificielle.....	41
3.1 Développement.....	41
3.1.1 Rôle du développement dans l'orientation de l'évolution.....	42
3.1.2 Principaux mécanismes.....	43
3.2 Ontogenèse artificielle.	44
3.2.1 Définition et objectif.....	45
3.2.2 Avantages et Inconvénients.....	46
3.3 Modèles de développement artificiel.....	47
3.3.1 Système de réaction-diffusion.....	47

3.3.2	Biomorphs.....	48
3.3.3	Réseaux booléens aléatoires.....	49
3.3.4	Embryologie artificielle.....	51
3.3.5	Morphogenèse de réseaux de neurones.....	55
3.3.6	Combinaison de mécanismes multiples pour la morphogenèse.....	56
3.3.7	Expression différentielle de gènes.....	57
3.3.8	Réseaux artificiels de régulation.....	59
3.3.9	METAMorph.....	65
3.3.10	Problèmes du drapeau.....	66
3.3.11	Cell2organ.....	67
3.4	Synthèse.....	69
3.4.1	Encodage directe versus indirecte versus implicite.....	69
3.4.2	Approches grammaticales versus basées sur la chimie cellulaire.....	70
3.5	Conclusion.....	72
Chapitre 4 : Un modèle bio-inspiré pour la synthèse de créatures artificielles.....		74
4.1	Description globale.....	75
4.1.1	Objectifs du modèle.....	75
4.1.2	Démarche.....	77
4.1.3	Vue d'ensemble.....	78
4.2	Environnement embryonnaire.....	79
4.2.1	Diffusion.....	80
4.2.2	Cellules.....	81
4.3	Métabolisme.....	81
4.3.1	Chimie artificielle.....	82
4.3.2	Chimie artificielle proposée.....	82
4.4	Les interactions cellulaires et environnementales.....	84
4.4.1	Capteurs membranaires.....	85
4.4.2	Système de sélection d'action.....	86
4.4.3	Actions cellulaires.....	88
4.5	Morphogenèse.....	88
4.5.1	Développement artificiel dans les L-systèmes.....	89
4.5.2	L-systèmes comme modèle de croissance.....	89
4.5.3	Modèle de croissance proposé.....	91
4.5.4	Apport de l'extension paramétrique.....	92
4.5.5	APL-système: Variante de L-système paramétrique.....	92
4.6	Evolution.....	98
4.6.1	Génotype.....	98
4.6.2	Démarche évolutionniste.....	100
4.6.3	Opérateurs génétiques du chromosome APL-système.....	104
4.6.4	Opérateurs génétiques du chromosome système de sélection d'actions.....	106
4.7	Implémentation du modèle.....	107
4.8	Conclusion.....	108

Chapitre 5 : Expériences et résultats.....	109
5.1 Développement d'organismes simples.	109
5.1.1 Expérimentation 1 : Génération de formes.....	110
5.1.2 Expérimentation 2 : développement du métabolisme.....	112
5.2 Expérimentation3 : développement dans des environnements asymétriques...118	
5.3 Propriétés d'auto-réparation.....	119
5.3.1 Expérimentation4 : blessures Apres développement.....	119
5.3.2 Expérimentation 5 : blessures en cours de développement.....	120
5.4 Vers la construction de drapeaux.....	121
5.5 Limites du modèle.	122
5.5.1 Complexité.....	122
5.5.2 Paramétrage.....	122
5.6 Etude comparative.	123
5.6.1 Comparaison sur l'environnement.....	124
5.6.2 Comparaison sur les mécanismes et propriétés simulés.....	124
5.6.3 Comparaison sur le contrôle du développement.....	126
5.6.4 Comparaison sur les organismes développés.	127
5.7 Inspiration biologique : discussion.	129
5.8 Bilan	129
5.9 Conclusion.....	130
Conclusion générale.....	131
6.1 Perspectives	133
Annexe A.....	134
Annexe B.....	139
Bibliographie.....	141

Liste des figures

Figure 2.1 : Le canard de Vaucansan.....	7
Figure 2.2 : quelques structures de jeu de la vie initialisées avec une configuration donnée. (a) Le recouvreur du plan, la figure initiale recouvre petit à petit le plan d'une configuration stable. (b) Le store vénitien, un vaisseau de période 2.....	11
Figure 2.3 : quelques structures du jeu de la vie qui apparaissent spontanément.	12
Figure 2.4-fractale : Flocon de Koch.....	12
Figure 2.5- Interprétation graphique du L-système.....	13
Figure 2.6-structure d'arbre générée par un DOL-système parenthésé.....	14
Figure 2.7-L-système stochastique [PL90].....	15
Figure 2.8-Communication entre plante et environnement dans les L-systèmes ouverts.....	16
Figure 2.9-Place des algorithmes évolutionnistes comme méthodes d'optimisation.	18
Figure 2.10- La boucle générationnelle de l'évolution.	19
Figure 2.11 : Neurone biologique.....	25
Figure 2.12 : Exemple d'un réseau de neurone : le perceptron multi-couches.....	26
Figure 2.13 : Interactions entre l'environnement et le système de classeurs.....	27
Figure 2.14: schéma de fonctionnement d'un système de classeurs.....	28
Figure 2.15- Simuler le comportement des créatures à l'aide de réseau de neurones. (a) Poisson artificielles de Demetri Terzopoulos. (b) co-évolution des deux créatures Cliff et Miller : comportement de fuite (jaune) et de poursuite. (c) comportement d'anticipation de David Panzoli.	31
Figure 2.16-Simuler le comportement des créatures à base de système de classeur (a) stratégie d'équipe dans un jeu de football de Cédric Sanza. (b) Stratégie d'équipe dans le cadre de jeu de basketball d'Olivier Heguy. (c)32	
Figure 2.17-Graphe de développement.....	33
Figure 2.18-Créatures de Karl Sims. (a) Des créatures nageant. (b) Deux créatures en lutte pour s'emparer d'un cube.....	34
Figure 2.19-Créatures de Hornby.(a) Dans le simulateur. (b) Réalisation sous forme physique (robot).	34
Figure 2.20-Framsticks. (a) créature terrestre. (b) Créature aquatique.....	35
Figure 2.21-Créatures de Lassabe. (a) La montée d'escalier. (b) le déplacement avec un skateboard.....	35
Figure 2.22-Panel de créatures artificielles. (a)créatures de Miconi, (b) créatures de Chaumont, (c, d) créatures de Lipson dans le simulateur, sous forme phasique (d), (e) Créatures de Ray, (f) créatures de Taylor.	36
Figure 2.23 : (a) Darrwin Pond, version antérieur de GenePool. (b) Life Drop.	38
Figure 2.24: Darwin's Park. (a) jardin virtuels. (b) Compétition sur le terrain virtuel.	39
Figure 2.25-Organisme développés grâce au modèle de développement de Doursat [Dou09]. (a) la salamandre.(b)l organisme à 8 pattes.....	39
Figure 3.1 : Le développement (l'ontogenèse) et sa place dans l'orientation de l'évolution.	42
Figure 3.2-Photos d'un Embryon d'humain de 6-7 semaines.	44
Figure 3.3- (a)Structure peu de poisson [KA95]. (b)Structure de coquille [Mei95].....	48
Figure 3.4- Exemple de structures développées par réaction diffusion	48
Figure 3.5: Exemple de Biomorphs.	49
Figure 3.6: Exemple d'un réseau booléen aléatoire. a) configuration du réseau b) la table de règles de transition ; c) le diagramme de l'espace d'états.....	50
Figure 3.7 : Exemple de trios types d' RBA .N = 32. Carré noire = état "1", carré blanc = état '0'. États initiaux au dessus et le temps se coule de haut en bas. (a) régulier (K = 1) ; (b) critique (K = 2) ; (c) Chaotique (K = 5).	50
Figure 3.8-Les 14 états de voisinage pour des cellules 2D.....	51
Figure 3.9-Format d'un chromosome simplifié.	52
Figure 3.10-Forme triangulaire et rectangulaire évoluée. Les cellules vides représentent la forme désirée.	52

Figure 3.11-L-forme désirée et ses régions composantes.	53
Figure 3.12-le chromosome de deux gènes utilisé pour la génération de la forme en L.	54
Figure 3.13-L-forme produite par un chromosome de deux gènes. (a) après expression du gène1. (b) après expression du gène 2. (La forme cible est esquissée par les cellules vides).	54
Figure 3.14: L-forme produite par un chromosome de deux gènes qui a évolué en utilisant la technique de formation (les cellules vides représentent la forme cible).	55
Figure 3.15-formation d'un organisme multicellulaire (cellules en bleu et molécules en rouge).	56
Figure 3.16:-Exemple d'un segment du génome d'Eigenberger.	57
Figure 3.17-En haut groupes de cellules résultant d'une croissance aléatoire. En bas : formes obtenus après l'ajout d'un morphogène pour moduler la croissance.	58
Figure 3.18-Exemples de formes tridimensionnelles évoluées.	59
Figure 3.19-L'expression du gène et la régulation dans le génome artificiel. La taille du gène N est égale à 6.	60
Figure 3.20 -Exemples de développement temporel des concentrations de protéines. (a) un changement de concentration oscillatoire ; (b) un changement de concentration lent et régulier ; (c) Transition de changement de concentration entre deux protéines.	62
Figure 3.21: La structure du génome et les détails d'un gène régulateur.	63
Figure 3.22-(a)L e carré de deux couleurs. (b) : Le carré de trios couleurs.	64
Figure 3.23-Drapeau Français de Chavoya. (a) Le drapeau de (27 ×9). (b) L e drapeau de (21 × 7).	64
Figure 3.24-METAMorph, réseau de régulation génétique cellulaire.	65
Figure 3.25- Développement d'un embryon en forme de cigare. (a)Début du développement. (b) Les cellules se développent sur les des deux extrémités. (c) Le cigare enfin réalisé, se composant de 684 cellules.	66
Figure 3.26- Drapeau Français de Miller. (a) Croissance à partir d'une seule cellule blanche. (b) Drapeau français capable de se reproduire après un endommagement aléatoire [Mil04].	67
Figure 3.27: Créatures développés. (a) un organe capable de transférer un substrat d'un point à l'autre. (b) Organisme autoalimenté : coopération de 4 différents organes. (c) une étoile de mer artificielle. (c) méduse artificielle.	68
Figure 3.28-Intégration des 3 niveaux de simulation (de gauche à droite : le niveau hydrodynamique, le niveau chimique et le niveau physique).	69
Figure 4.1 - Contribution.	77
Figure 4.2 - Le modèle de développement cellulaire : vue d'ensemble.	79
Figure 4.3 - La grille torique bidimensionnelle.	80
Figure 4.4 - La diffusion de molécules.	80
Figure 4.5 - L'état interne d'une cellule artificielle.	81
Figure 4.6- La cellule artificielle dans son environnement. Elle contient des capteurs (cercles) pour mesurer la densité de substrats (pentagones réguliers) valable dans son voisinage.	85
Figure 4.7- Positionnement des capteurs de la cellule : voisinage de Von Neumann.	86
Figure 4.8 - L'ensemble de systèmes de sélection d'action de chaque créature définis sur un alphabet ternaire {0,1,#}.	86
Figure 4.9 - Panel de systèmes réalisé par L-système. (a, c, d) Développement de plantes virtuelles [BL08], [MP9 respectivement]. (b) Modélisation de la rétine [KTV99]. (e, f) Conception évolutionniste de créatures artificielles dans le simulateur, sous forme de robots.	90
Figure 4.10 - Représentation des règles dans l'APL-système.	93
Figure 4.11- Exemple de développement d'une créature en utilisant APL-système. Les flèches rouges en haut de chaque cellule déterminent la direction courante. Les formules entre parenthèses se trouvant à proximité de chaque cellule déterminent l'état interne de celles-ci.	97
Figure 4.12 - Comparaison entre deux organismes développés par deux L-systèmes différents. (a) : Organisme produit par APL-système. (b) : Organisme produit par L-systèmes paramétriques classique équivalent.	98
Figure 4.13 - Le génome d'une créature artificielle est composé de deux chromosomes : le sélecteur d'action et le APL-système.	99
Figure 4.14 - La Boucle générationnelle de l'évolution illustrée par une approche ontogénétique.	102

Figure 4.15- mutation dans le chromosome système de sélection d'actions. (a) La mutation porte sur une condition. (b) La mutation porte sur une action. (c) La mutation porte sur la priorité. La condition est définie sur un vecteur de 8 valeurs et l'action est définie sur un vecteur de 4.....106

Figure 4.16 - Croisement dans le chromosome système de sélection d'actions.....107

Figure 5.1- morphologies émergées de l'évolution du chromosome L-systèmes.....111

Figure 5.2- Croissance de d'un organisme (écrevisse artificielle) se composant de 146 cellules. (a) Début de la simulation. (b) L'organisme se développent en utilisant les ressource de l'environnement. (c) fin du développement.115

Figure 5.3- Développement d'un organisme (chauve-souris artificielle) se composant de 561 cellules. (a) Début de la simulation. (b) La chauve-souris se développent en utilisant les ressources de l'environnement. (c) la chauve-souris s'est développée.....117

Figure 5.4 -(a) organisme multicellulaire créé dans un environnement équilibré. (b): organisme multicellulaire artificiel créé dans un environnement moins équilibré (insuffisance du substrat bleu).118

Figure 5.5- Blessures en fin du développement. (1) L'organisme après lui avoir infligé une blessure au centre. (2-5) L'organisme se régénère.120

Figure 5.6-Blessures infligées au cours du développement. (1) la créature se développent. (2) Réalisation d'un trou centrale au niveau des cellules nouvellement créés. (3, 4, 5, 6) La créature se régénère. (7) la réalisation simultanée de deux blessures : la première sur la branche latérale de la créature, la deuxième au centre bas de la créature. (8, 9, 10) La créature se régénère.120

Figure 5.7-Morphologies en forme de drapeau.121

Liste des tableaux

<i>Tableau 2.1-Terminologie des algorithmes évolutionnistes.....</i>	<i>19</i>
<i>Tableau 2.2-Principales différences entre les différents types d'algorithmes évolutionnistes.</i>	<i>23</i>
<i>Tableau 3.1: Synthèse des travaux sur le développement artificiel.</i>	<i>71</i>
<i>Tableau 4.1- Relation entre le mécanisme du développement simulé et l'outil informatique utilisé pour le faire.</i>	<i>79</i>
<i>Tableau 4.2- Alphabet du L-système.</i>	<i>91</i>
<i>Tableau 4.3- Chromosome L-système : Opérateurs génétique de croisement.</i>	<i>105</i>
<i>Tableau 4.4- Chromosome L-système : Opérateurs génétique de mutation.....</i>	<i>106</i>
<i>Tableau 5.1-Liste des actions possibles des cellules pour développer leurs métabolismes.....</i>	<i>114</i>
<i>Tableau 5.2-Système de sélection d'action produit par le génome du meilleur organisme obtenu grâce à l'algorithme génétique.</i>	<i>116</i>
<i>Tableau 5.3-Comparaison sur les propriétés des environnements des modèles de développement.</i>	<i>124</i>
<i>Tableau 5.4-Comparaison sur les mécanismes ou propriétés simulés.</i>	<i>125</i>
<i>Tableau 5.5- Comparaison sur les mécanismes impliqués dans le contrôle du développement.....</i>	<i>127</i>
<i>Tableau 5.6-Etude comparative : comparaison sur les organismes développés.....</i>	<i>128</i>

Chapitre 1

Introduction générale

“The important thing in science is not so much to obtain new facts as to discover new ways of thinking about them.” William Lawrence Bragg.

1.1 Contexte générale.

S’inspirer du vivant est de nos jours capital pour le développement de la technologie du futur et l’informatique du demain. Alliant informatique et biologie et naissant du croisement de ces deux disciplines, la synthèse de créatures artificielles est depuis peu un domaine de recherche qui présente un intérêt de plus en plus grandissant, à la fois pour son potentiel en ingénierie pour la conception de futurs nano et biosystèmes, d’une part et pour la modélisation et la simulation pour la compréhension des mécanismes du vivant, d’autre part. Deux grands types de travaux sont menés pour créer ces “être” artificiels peuplant actuellement les mondes virtuels.

La *phylogenèse artificielle* permet de générer des créatures en utilisant un mécanisme évolutionniste pour réaliser l’évolution conjointe de la morphologie et du comportement. Ces créatures sont plongées dans des environnements 3D qui simulent les phénomènes physiques naturels (forces physiques) et sont confrontées à des problèmes à résoudre (se déplacer, nager, faire la planche à roulette, etc.) qui constituent dans la plus part des cas leurs fonction d’évaluation. De nos jours, les travaux les plus aboutis dans ce domaine, restent ceux de Karl Sims [Sim94] qui a été le premier à générer des créatures artificielles adaptées à leur environnement et ce en assemblant des blocs assujettis à des contrôleurs construits par évolution.

La deuxième approche, *L’ontogenèse artificielle* consiste à *développer* la créature – ou tout au moins un organe de la créature - à partir d’une seule cellule. Cette approche rapprochera donc les créatures artificielles de celles existantes dans la nature. Elle est donc la voie si on veut obtenir des créatures avec la capacité de s’auto-réparer ou des robots modulaires. Par

ailleurs, elle permet de vérifier les hypothèses formulées par les biologistes quant à l'étude des systèmes vivants.

1.2 Problématique.

Les modèles issus de l'ontogenèse artificielle se divisent également en deux approches : les approches grammaticales et celles basées sur la chimie cellulaire. Cette classification proposée par Stanley et Miikkulainen [SM03] est basée sur les mécanismes utilisés pour contrôler le *développement*. En d'autres termes, les mécanismes biologiques simulés ou pris en compte lors de la construction de la structure développée.

En effet, dans les approches grammaticales, le *développement* est contrôlé par un mécanisme de lignée cellulaire (*cell lineage*), simulé par un ensemble de règles de production appliquées récursivement pour transformer un état initial (cellule) en un phénotype (organisme multicellulaire).

Ces approches produisent de très bons résultats en ce qui concerne la morphogénèse de réseaux de neurones [Kit90, BKHS93, Gru95], la conception évolutionniste de créatures artificielle ou réelles (robots) [Sim94a, HP02, PHLF03], etc. Néanmoins, elles ne semblent pas bien adaptées à des processus suggérant l'autonomie des éléments du système évolué. Pour cela, de nombreux chercheurs ont tenté de développer d'autres méthodes plus proches de la réalité biologique en simulant, via les approches basées sur la chimie cellulaire, des aspects de bas niveau de l'ontogenèse naturelle. En effet, Dans ce second type d'approche, le développement artificiel est basé sur des dynamiques plus proches que celles vues dans le développement naturel telles que : la diffusion, la réaction/diffusion, l'expression et la régulation de gènes, le métabolisme, les réseaux de régulation génétiques, la différenciation, la division cellulaire, etc.

Notons que les modèles de développement basés sur ces mécanismes et notamment, les réseaux de régulation artificielle prennent compte des phénomènes plus élémentaires que ceux proposés par l'approche grammaticale et permettent ainsi de manière générale de définir des représentations génétiques plus expressives (la taille du génotype est moins liée à celle du phénotype). Cependant, comme le note Dellaert et Beer [DB96], les opérateurs génétiques sont plus difficiles à définir et la perte de performances en terme de convergence n'est pas compensée par le gain potentiel d'expression. Par exemple, dans le cadre du développement des réseaux de neurones, jusqu'à ce jour, les approches grammaticales sont plus performantes.

Ainsi, un couplage de ces deux approches sera intéressant à étudier afin de pallier le problème de convergence et bénéficier des avantages de chaque approche, en simulant dans un seul modèle de développement cellulaire les éléments majeurs de chacune des deux approches.

Par ailleurs, en tant qu'approches grammaticales, les L-systèmes sont considérés comme un formalisme très approprié pour décrire beaucoup de processus de croissance dans les organismes, en particulier s'ils comportent des structures auto-similaires et hiérarchiques comme les morphologies des créatures de Karl Sims. Beaucoup d'études ont exploré avec succès l'évolution de ces systèmes [KTV99, HP01b, HP02, PHLF03], notamment dans le contexte du développement de plantes virtuelles [MP96, Nos03, BL08]. Explorer le potentiel de ces systèmes dans un modèle de développement cellulaire semble également être intéressant à étudier.

1.3 Contribution.

Dans l'optique de synthétiser des créatures entières à partir d'une cellule unique possédant un métabolisme, une morphologie et un comportement, nous voulons proposer un modèle hybride ontogénétique et phylogénétique. Afin de synthétiser, d'abord des organismes multicellulaires servant à décrire par la suite des créatures complètes, le but de ce mémoire est de proposer un nouveau modèle de développement cellulaire unifiant les deux approches divisant l'ensemble des modèles d'ontogenèse artificielle : les approches grammaticales et celles basées sur la chimie cellulaire. Les organismes développés par ce modèle devront être composés de plusieurs centaines ou milliers de cellules. De plus, ils devront posséder une morphologie générée par mécanisme de lignée basé sur les L-systèmes, un métabolisme et des capacités d'auto-réparation. En faisant l'hypothèse que les blocs de Karl Sims peuvent être remplacés par des regroupements de cellules formant des sortes d'organes, ces organismes pourront à long terme posséder un module comportemental de haut niveau leur permettant d'accomplir quelques fonctions de locomotion dans un environnement où la physique est simulée.

Pour simuler le développement de nos organismes multicellulaires, nous simulons leur métabolisme, leur morphogenèse et leur évolution. Pour ce faire, notre modèle contient un environnement avec une chimie artificielle simplifiée [DZB01, RCNA03, Hut07] et des cellules qui accomplissent différentes actions. Pour simuler la morphogenèse des cellules évoluant dans cet environnement, nous proposons une variante des L-systèmes paramétriques que nous baptisons *APL-système*.

Nous montrons que notre modèle est capable de produire diverses formes de créatures, utilisant des substrats de l'environnement pour croître et agir. De plus, notre modèle peut véhiculer des capacités d'auto-réparation de la structure de la créature. Si certaines cellules de l'organisme sont détruites après ou même au cours du développement, le modèle est capable de régénérer les parties détruites.

1.4 Structure du mémoire

Après ce chapitre liminaire consacré à la description de la problématique et de la contribution, le deuxième chapitre pose les repères nécessaires à une compréhension du contexte générale dans lequel s'inscrit ce mémoire. Après une introduction générale au domaine de la vie artificielle, nous présenterons trois concepts fondamentaux sur lesquels ce champ de recherche est basé : la morphogenèse, l'évolution, l'adaptation et l'apprentissage.

Enfin, ce chapitre dresse un état de l'art sur la synthèse de créatures artificielles.

Afin de comprendre puis d'établir un modèle de développement cellulaire qui peut être simulé sur un ordinateur, nous présentons brièvement dans le troisième chapitre quelques pré-requis biologiques sur le développement naturel, sa place dans l'orientation de l'évolution ainsi que ces principaux mécanismes. Ensuite, nous définissons l'ontogenèse artificielle et présentons ses objectifs ainsi que ses avantages et ses inconvénients comme sous-discipline de calcul évolutionniste et comme une approche permettant de synthétiser des créatures artificielles. Après, nous passons en revues les modèles pionniers d'ontogenèse artificielle qui ont marqué l'évolution du domaine, ainsi que les principaux développements récents. Le chapitre est conclu par une synthèse de ces travaux basée sur deux classifications reconnues par la littérature qui sont : l'encodage directe/indirecte/implicite et l'approche grammaticale/basée sur la chimie cellulaire.

Le modèle proposé est détaillé dans le quatrième chapitre. La première section est consacrée à une description générale de ce modèle en présentant ses objectifs, la démarche à suivre pour y répondre, puis une vue d'ensemble de sa structure. Après, nous détaillons la partie ontogénétique de notre modèle en commençant par l'environnement embryonnaire dans lequel nos organismes multicellulaires se développent, puis les différents mécanismes utilisés pour simuler leur développement à savoir : le métabolisme, les interactions cellulaires et environnementales, la morphogénèse et enfin l'évolution. Le chapitre est clôturé avec les principes d'implémentation de notre modèle.

Nous présentons dans le cinquième chapitre les expériences évolutionnistes conduites avec notre modèle. Nous montrons ensuite ses limites. Puis, afin de mieux positionner notre modèle par rapport aux modèles de références présentés dans l'état de l'art, nous proposons une étude comparative basée sur les principales propriétés des modèles de développement, à savoir: les propriétés des environnements, les mécanismes simulés par le modèle, les mécanismes impliqués dans le contrôle du développement et les propriétés des organismes obtenus. Nous discutons ensuite l'inspiration biologique dans ce modèle. Le chapitre est conclu avec un bilan remettant en cause les notre modèle et sa capacité à répondre aux objectifs envisagés.

Le sixième et dernier chapitre est réservé à la conclusion générale du travail. Il replace les résultats obtenus dans le contexte de la problématique de ce mémoire, discute les

perspectives de recherches futures, ainsi que l'application de notre modèle en bio et nano-systèmes.

Chapitre 2

La vie artificielle

"L'ordre ne crée pas la vie.", Antoine de Saint-Exupéry

La discipline qui est de nos jours le fer de lance pour les approches d'inspiration biologique est sans conteste, la vie artificielle. Cette discipline constitue le cadre dans lequel s'inscrit ce mémoire. Pour cela, nous l'introduisons dans ce chapitre en présentant ses origines des mythes jusqu' à sa naissance comme une approche scientifique. Nous la définirons puis nous présenterons l'ensemble des techniques qui sont d'inspiration purement informatique (L-systèmes, automates cellulaire, etc.) et/ou biologique (algorithmes génétique, réseau de neurone, etc.), et qui sont maintenant attribués à la vie artificielle. Nous regrouperons, ces techniques en trois classes à savoir : les mécanismes de morphogenèse, d'évolution et d'apprentissage, et nous montrons en fin de chapitre comment cette discipline utilise ces mécanismes pour simuler le vivant : synthétiser des créatures artificielles.

2.1 Introduction.

La vie artificielle est une discipline de recherche qui étudie la vie par la simulation et la synthèse. Les scientifiques de ce domaine essayent, en effet, de capturer les prospérités fondamentales du vivant pour la conception de techniques artificielles et en particulier informatiques, ou bien pour créer la vie non pas telle qu'on la connaît mais telle qu'elle pourrait être. Nous nous inscrivons pleinement dans cette discipline, c'est pourquoi nous présentons maintenant quels en sont les origines et les objectifs.

2.1.1 Origines et objectifs.

Très tôt dans l'histoire, l'homme a voulu "recréer" l'œuvre de la nature en produisant des machines semblables à l'image qu'il se faisait de la vie. On pourrait faire remonter les premières manifestations aux mythes.

Vie artificielle dans les mythes, racines historiques.

Dans le *Lieh Tzu* (cité par [Ren02]), est racontée l'histoire d'un artificier talentueux, Yen Shih, qui aurait fabriqué avec du cuir, du bois, de la colle et de la peinture de couleurs variées, une créature qui pouvait marcher, chanter, bouger sa tête de haut en bas comme une vraie personne. Cet automate était si convaincant qu'il suscita la colère du roi Mu quand ce dernier le vit faire un clin d'œil à l'une de ses concubines. Et les exemples littéraires sous-jacents ne manquent pas. A titre d'exemple, dans le mythe raconté par le poète Ovide [Ovi92], Pygmalion, un jeune roi de Chypre, sculpta dans l'ivoire un corps de femme d'une beauté surréaliste. Cette création est nommée "Galatée", et sa vie lui est insufflée par Aphrodite, la déesse de l'amour et de la beauté. Nous citons également, le récit de l'alchimiste Paracelse dans lequel, il nous transmet une recette de l'homoncule (littéralement "petit homme"). Enfin, dans le célèbre roman *Frankenstein*, Mary Shelley [She97] raconta l'histoire d'un médecin (Frankenstein), qui assembla des morceaux de cadavres pour former un être monstrueux.

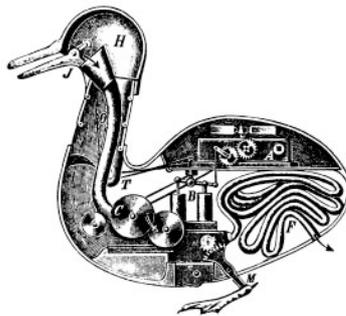


Figure 2.1- Le canard de Vaucanson.

Première copies réelles du vivant, automates mécaniques.

Les tentatives pour mimer la nature ne furent pas le fait seul de personnage de mythes, et après de nombreux progrès techniques, en particulier dans le domaine de l'horlogerie, beaucoup d'automates mécaniques ont été fabriqués en tant que copies du vivant. Jacques de Vaucanson, à 7 ans, construisit sa première horloge. Puis il courait après un rêve fou : créer un homme artificiel, un androïde. Il y laissa sa modeste fortune. Criblé de dettes, il construisit ensuite de petits automates qu'il exhibait dans les foires. Il assembla successivement entre 1738 et 1739 : le Flûtiste, le célèbre Canard, qui pouvait picorer du grain, boire et même digérer, et un joueur de tambourin et de flageolet (genre de flûte à bec). Dans la même époque, Pierre Jacquet-Droz construisait aussi des automates stupéfiants, dont les plus célèbres : un écrivain, un dessinateur et une joueuse de tympanon (piano simplifié).

Avènement de l'ordinateur.

La révolution scientifique du XXème siècle a accéléré le développement de la vie artificielle en apportant de nombreux outils. Les travaux d'Alan Turing sur la logique formelle l'amènent à poser l'ordinateur comme un outil de simulation du réel. En effet, Turing démontra en 1936 qu'une machine universelle (disposant d'une mémoire infinie) peut calculer n'importe quelle fonction récursivement calculable (c.à.d. exprimable à l'aide de règles formelles, donc sous forme d'un algorithme) si cette machine est pourvue du programme adéquat [Tur36]. Ce travail fut l'origine de nos ordinateurs, qui sont en fait des machines universelles finies (mémoire limitée). La première tentative de simulation d'une propriété du vivant : l'intelligence, revint aussi à Turing qui avait proposé un test fameux permettant de déterminer si un ordinateur « pense » [Tur50].

Naissance de la vie artificielle contemporaine.

Les premiers travaux cherchant à étudier le vivant à travers des modèles informatiques sont ceux de Jon Von Neumann [Von66]. Celui-ci a inventé un automate, baptisé “kinématon”, capable de mimer un des comportements essentiel du vivant, l'autoréplication. Hormis le fait remarquable que le développement de ces modèles auto-répliquant précède la découverte de l'ADN (notons les similitudes qui existent entre ce modèle et les découvertes biologiques dans ce sens), Von Neumann base son approche sur un constructeur capable d'universalité. En effet, son automate permet, en plus de se générer, de générer des variantes de lui-même. Cette approche a ouvert la voie à d'autres travaux sur des automates capables d'autoréplication.

Par la suite, des travaux dans de nombreux domaines scientifiques ont été effectués en utilisant des concepts qui sont aujourd'hui liés à la vie artificielle. A titre d'exemple, nous citons les travaux sur les automates cellulaires, en particulier le jeu de la vie (section 1.2.4) de John Conway [Gar70] qui se présente en un système artificiel simple qui évolue en un système complexe localement organisé ; les travaux sur l'évolution artificielle comme les algorithmes génétiques [Hol75] et les stratégies d'évolution [Rec73, Sch75] (section 2.3.1). En fin, nous citons, les travaux sur les L-systèmes [Lin68] (section 1.2.6) et les réseaux de neurones [MP43] (section 2.4.1) qui ont eu également, par la suite, un impact important dans le développement du domaine.

Tous ces travaux entraînent, en 1987, au workshop organisé par Langton au Los Alamos National Laboratory, l'apparition officielle de la discipline nommée Vie Artificielle. Sa “création” est alors motivée par l'incompréhension des processus complexes qui modèlent la vie.

2.1.2 Définition.

La première définition de cette discipline est donnée par son fondateur, Christopher Gale Langton [Lan89] :

“La vie artificielle est l’étude des systèmes construits de mains d’hommes qui exhibent des comportements caractéristiques des systèmes naturels vivants. Elle vient en complément des sciences biologiques traditionnelles qui analysent les organismes vivants, en tentant de synthétiser des comportements semblables au vivant au sein d’ordinateurs et d’autres substrats artificiels. En étendant les fondements empiriques sur lesquels la biologie est basée au-delà de la vie à base de carbone qui a évolué sur Terre, la vie artificielle peut contribuer à la biologie théorique en positionnant la vie telle que nous la connaissons au sein d’un espace plus large : la vie telle qu’elle pourrait être”.

En d’autres termes, le but de la vie artificielle est d’étudier les principes généraux de la vie de façon empirique par l’intermédiaire de trois approches [Bed92] :

- les systèmes auto-organisés, la génération spontanée de l’ordre, de l’émergence et de la coopération;
- la métabolisation, l’autoreproduction et l’autopoïèse¹ ;
- l’apprentissage, l’adaptation et l’évolution.

La vie artificielle peut être également définie comme un ensemble de propriétés dont voici la liste [FB89]:

1. L’être humain a contribué au processus d’apparition de tout système de vie artificielle.
2. Un système de vie artificielle est autonome.
3. Un système de vie artificielle est en interaction avec son environnement.
4. Il y a émergence de comportements dans un système de vie artificielle. Les trois propriétés suivantes ne sont pas indispensables mais restent néanmoins très présentes.
5. Un système de vie artificielle peut se reproduire lui-même.
6. Un système de vie artificielle possède une capacité d’adaptation.
7. Un système de vie artificielle n’est pas une unité. l’opposé de la vie, un système de vie artificielle peut être réparti en plusieurs endroits : par exemple, un robot et un ordinateur peuvent effectuer les calculs reliés par ondes. Même à l’intérieur d’un ordinateur, rien ne garantit que les octets de ce système soient tous regroupés.

¹ Le terme autopoïèse vient du grec auto (soi-même), et poièsis (production, création). Il définit la propriété d’un système à se produire lui-même (et à se maintenir et se définir lui-même).

2.1.3 Emergence et auto-organisation.

Les recherches dans la vie artificielle s'appuient sur deux concepts centraux : l'émergence et l'auto-organisation. On identifie l'émergence lorsqu'on peut déduire les propriétés globales d'un système à partir des interactions locales entre ses composants. Ces interactions locales peuvent suivre des règles simples qui font émerger le comportement global du système, mais ne peuvent être prévues en analysant chaque composant à part. En d'autres termes, un système émergent possède certaines propriétés qui ne sont pas dues aux propriétés de ses constituants mais à leur organisation. Certains systèmes développent une forme d'autonomie ; ils s'organisent de l'intérieur : on parle alors d'auto-organisation. Selon cette vision, les organismes vivants sont considérés comme tel (vivants), non pas en raison des propriétés de leurs constituants, mais de leur auto-organisation.

2.2 Morphogenèse.

Les premiers travaux sur la génération (genèse ou création) de formes/structures dans des systèmes informatiques trouvent leurs origines dans les travaux de Turing avec son article publié en 1952, intitulé les bases chimiques de la morphogenèse (*chemical basis of morphogenesis*) [Tur52]. Ce travail ainsi que d'autres sur la morphogenèse seront présentés en section 3.3. Nous présentons dans cette section les principaux mécanismes, utilisés dans la vie artificielle, comme générateurs de la morphogenèse.

2.2.1 Automates cellulaires.

L'origine des automates cellulaires remonte aux années soixante et à Stanislav [Ula62]. Il s'agit d'un système dynamique discret qui modélise des objets ayant une évolution parallèle synchrone. C'est un système discret parce que le temps avance par pas et l'espace est représenté par une grille régulière de cellules. Chacune possède un état choisi parmi un ensemble fini et qui peut évoluer au cours du temps. L'état d'une cellule au temps $t+1$ est déterminé selon un ensemble de règles de transition et dépend de son état au temps t et d'un nombre fini de cellules appelé son voisinage. À chaque nouvelle unité de temps, les mêmes règles sont appliquées simultanément à toutes les cellules de la grille. Les états de toutes les cellules constituent l'état global de l'automate. Formellement, un automate cellulaire peut être défini par un 4-uplets (N, V, Σ, ϕ) où :

- N , la dimension de la grille.
- V , le voisinage pris en compte.
- Σ , l'ensemble d'états de la cellule.
- ϕ , les règles de transition locales.

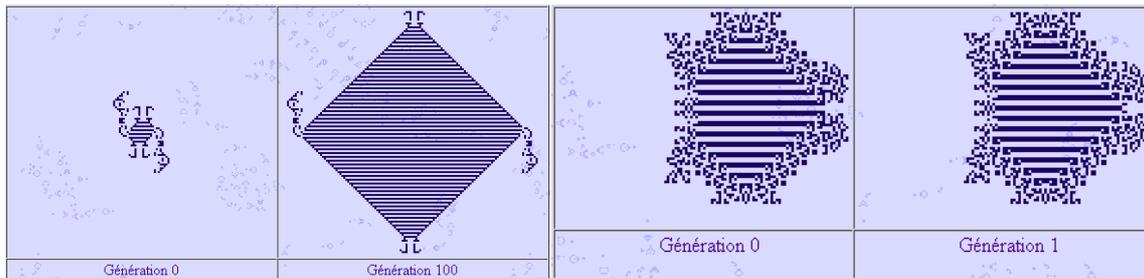


Figure 2.2- quelques structures de jeu de la vie initialisées avec une configuration donnée. (a) Le recouvreur du plan, la figure initiale recouvre petit à petit le plan d’une configuration stable. (b) Le store vénitien, un vaisseau de période 2.

Malgré la simplicité de leur formalisation, les automates cellulaires peuvent faire émerger des comportements très complexes et chaotiques. Outre leur intérêt théorique comme modèle mathématique, leur complexité, induit par des règles locales élémentaires, permet l’étude de l’auto-organisation de phénomènes naturels [Wol83].

Jeu de la vie.

Il est difficile de parler des automates cellulaires sans évoquer le fameux Jeu de la Vie. Inventé par John Conway à la fin des années 60, et popularisé par Martin Gardner [Gar70], il constitue actuellement le plus connu de tous les automates cellulaires. Ceci grâce à ses résultats surprenants, induits par des règles très simples, où des cellules naissent, vivent et disparaissent dans un espace bidimensionnel toroïdale.

Dans cet automate, chaque cellule possède deux états possibles 0 et 1, ou bien morte et vivant d’où le nom de l’automate. Le voisinage est défini comme l’ensemble des cellules adjacentes, diagonales comprises. Toute cellule possède donc huit voisines. Concernant les transitions d’état, deux règles élémentaires sont utilisées :

- Naissance : si une cellule morte est entourée de trois cellules vivantes, elle naît.
- Survie : une cellule vivante reste vivante à condition d’être entourée de deux ou trois cellules vivantes, autrement, elle meurt.

A partir de ces deux règles, une multitude de structures émergentes peut être trouvée. De manière générale, une configuration initiale aléatoire se stabilise par la formation de motifs qui peuvent être classés en trois catégories :

- Les blocks ou structures stables, c.à.d. des groupes de cellules qui n’évoluent plus (telles que les figures 2.3 A-C),
- Les oscillateurs ou structures périodiques, qui reviennent à leur forme originelle après un certain nombre de pas du temps (par exemple le “clignotant” de la figure 2.3 D),

- les vaisseaux spatiaux, sont des oscillateurs qui se déplacent au cours de leur cycle (comme le “planeur” ou “glisseur” de la figure 2.3 E).

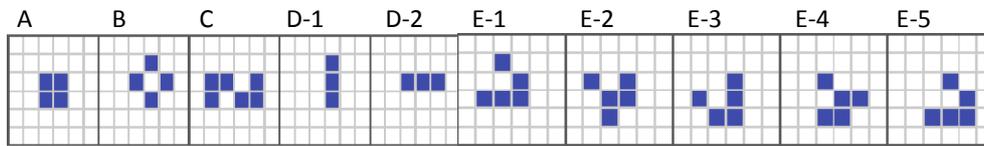


Figure 2.3- quelques structures du jeu de la vie qui apparaissent spontanément.

A part ces formes principales, il existe de nombreuses autres structures plus complexes. Nous citons, à titre d'exemple, les “canons à planeurs” qui émettent un planeur à un rythme régulier ou la “locomotive à vapeur” qui se déplace dans l’automate en laissant une traînée de cellules vivantes sur son chemin. Il existe également d’autres structures, qui n’apparaissent pas spontanément comme “le recouvreur de plan” ou “le store vénitien” (figure 2.2). Ces structures peuvent sembler avoir peu d’intérêt scientifique. Cependant, le jeu de la vie est bien plus qu’un jeu. Il s’agit d’un modèle déterministe aux processus émergents nombreux et variés.

2.2.2 Fractales.

Les fractales font partie des approches génératrices de la morphogenèse. Une fractale² est un objet mathématique de structure complexe dont la création ou la forme met en jeu des règles utilisant le fractionnement. La génération de fractales emploie des algorithmes récursifs qui permettent à une ou plusieurs fonctions de s’appeler elles-mêmes.

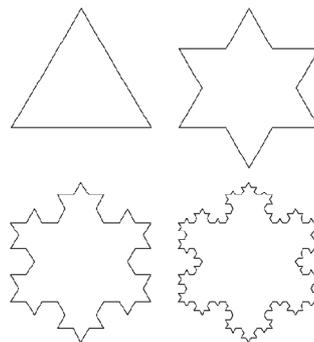


Figure 2.4-fractale : Flocon de Koch.

2.2.3 L-systèmes.

Une des conséquences de la ressemblance entre les approches génératrices, telles que les fractales et les systèmes vivants tels que les fougères, a été l’émergence de disciplines tentant de reproduire, via des mécanismes récursifs, certaines propriétés morphogénétiques des systèmes complexes biologiques. L’un des travaux les plus marquants est celui fourni par

² Le terme fractal vient du mot latin *fractus* qui signifie brisé.

Lindenmeyer en 1968 sur les systèmes de réécriture, appelés L-system, capables de générer des structures de plantes ou des colonies de bactéries [1968].

Ce formalisme est basé sur les principes de la théorie des langages, plus précisément sur les concepts de la grammaire formelle de Chomsky [Cho57]. Un L-système élémentaire est un L-système Déterministe et hors contexte (*Deterministic context-free L-systems*) appelé DOL-système. A l'image des grammaires formelles, un DOL-système est défini par le triplet (V, ω, P) où :

- V , un alphabet c'est à dire un ensemble finit de symboles non terminaux et terminaux (qui peuvent ou non être remplacés par d'autres symboles de l'alphabet respectivement),
- ω , un axiome qui désigne le mot initial,
- P , un ensemble de règles de production définissant les transformations, décrites par "symbole prédécesseur \rightarrow symboles successeurs".

Un L-système s'exécute en appliquant itérativement et parallèlement les règles de P sur ω . Dans l'exemple suivant :

- $V = \{F, +, -\}$,
- $\omega = F$ et
- $P: F \rightarrow F + F - F + F$

Les premières itérations génèrent :

Génération 1 : F

Génération 2 : $F + F - F + F + F + F - F + F - F + F - F + F + F + F - F + F$

Afin d'obtenir des formes géométriques, chaque symbole possède une interprétation graphique en terme de tracé, d'orientation et de déplacement. Cette interprétation s'appelle "la géométrie de la tortue" [Pru86], en imaginant une tortue virtuelle qui se déplace dans un plan et laisse une trace derrière elle. En mettant la spécification suivante :

- F : tracer un segment et avancer d'une unité,
- $+$: tourner à gauche de l'angle $\delta = 60^\circ$,
- $-$: tourner à droite de l'angle $\delta = 60^\circ$.

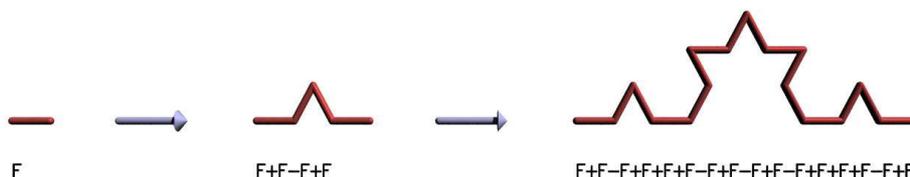


Figure 2.5- Interprétation graphique du L-système.

L'interprétation graphique de ce L-système correspond à la création d'une forme fractale, le célèbre flocon de Koch proposée par Von Koch en 1905 [Von05] et illustrée sur la figure 2.4. Le processus de génération est montré par la figure 2.5. Afin de générer des formes hiérarchiques (arborescentes), une pile d'état et deux caractères spéciaux sont utilisés : le caractère "[" permet d'empiler l'état courant de la tortue (sa position et son orientation) et le caractère "]" permet de l'épiler. De tels L-systèmes sont nommés L-systèmes parenthésés (*bracketed L-systems*). L'ajout de parenthèse au L-système précédent donne par exemple :

- $V = \{F, +, -, [,]\}$,
- $\omega = F$ et
- $P : F \rightarrow F[+F]F[-F]$

Ce qui permet de générer, après un certain nombre d'itérations, une structure d'arbre (figure 2.6).

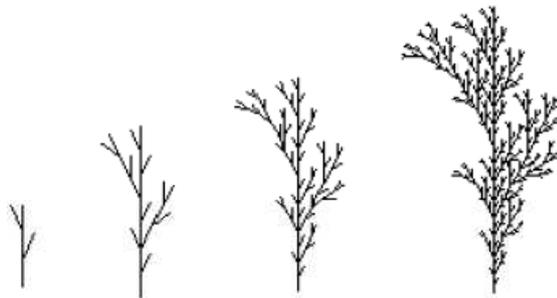


Figure 2.6-Structure d'arbre générée par un DOL-système parenthésé.

A partir du formalisme de base, plusieurs extensions ont été proposées, nous en présentons à présent les plus connus et les plus utilisées.

L-systèmes stochastiques.

Les L-systèmes stochastiques associent, à chaque règle de production, une probabilité de déclenchement. En effet, le formalisme de base est étendu au quadruplet (V, ω, P, π) , où $\pi \rightarrow (0,1)$, appelé fonction de distribution de probabilité, fait correspondre à chaque règle de production une probabilité. Notons que, pour tout symbole $a \in V$, la somme des probabilités des règles ayant comme prédécesseur a est égale à 1. De cette manière, différentes règles de production peuvent être appliquées à plusieurs occurrences du même symbole prédécesseur dans un même pas de dérivation donné.

Ainsi, un L-système stochastiques est capable de produire une grande variété de structures différentes à partir du même jeu de règles. Prenant l'exemple suivant :

- $V = \{F, +, -, [,]\}$,
- $\omega = F$
- $P : \left\{ F \xrightarrow{0.33} F[+F]F[-F], F \xrightarrow{0.33} F[+F]F, F \xrightarrow{0.34} F[+F]F \right\}$

Les trois productions ont une probabilité d'approximativement 1/3. La figure 2.7 montre des exemples de structures générées. Notons que ces structures ressemblent à des spécimens de la même espèce.

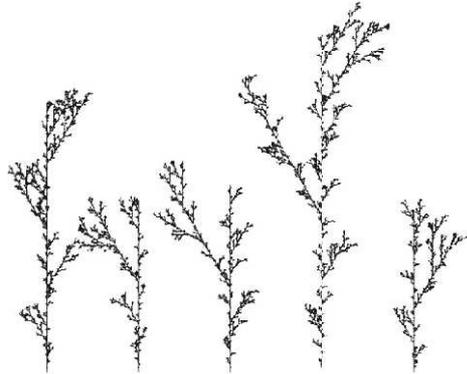


Figure 2.7-L-système stochastique [PL90].

L-systèmes contextuels.

Dans les L-systèmes à grammaire contextuelle (*context sensitive L-systems*), les règles de production dépendent, non seulement du caractère à remplacer, mais aussi de ses voisins (son contexte). Cette extension permet de modéliser l'interaction entre les constituants d'un organisme naturel. Dans le cas d'une plante par exemple, elle permet le transport de ressources entre ses parties adjacentes.

Un 2L-système possède des règles qui considèrent le contexte des deux côtés. Un symbole produit des symboles successeurs si et seulement si il est précédé d'un certain contexte gauche et suivi d'un certain contexte droit. Pour la détermination du contexte, les commandes de rotation de la tortue sont généralement ignorées. Les productions d'un 1L-système ne considèrent le contexte que d'un côté uniquement. Plus généralement, un (k, l) -L-système tient compte de k symboles vers la gauche et l symboles vers la droite. Les règles d'un L-système contextuel sont spécifiées par la notation :

$$\text{contexte gauche} < \text{prédécesseur} > \text{contexte droit} \rightarrow \text{successeurs}$$

Les productions contextuelles peuvent créer des conflits de priorité lors du remplacement d'un symbole. De manière générale, si deux productions sont applicables, la règle vérifiant plus de contexte est sélectionnée.

L-systèmes paramétriques.

Dès les débuts des L-systèmes, il a été envisagé d'associer des paramétriques numériques aux symboles. Un L-system paramétrique [Lin74] opère en effet sur des mots paramétriques, qui sont des chaînes de modules. Un module est constitué de symboles (formant son nom) et de paramètres formels associés. Formellement un L-système paramétrique est défini par le quadruplet (V, Σ, ω, P) où :

- V , l'alphabet du L-système,
- Σ , un ensemble de paramètres formels,
- $\omega \in (V \times \mathcal{R}^*)^+$, un mot paramétrique non vide, axiome et,
- $P \subset (V \times \Sigma^*) \times C(\Sigma) \times (V \times \xi(\Sigma))^*$, un ensemble finit de règles de reproduction. $(V \times \Sigma^*)$ est l'ensemble des prédécesseurs de la production, $C(\Sigma)$ l'ensemble des conditions, et $(V \times \xi(\Sigma))^*$ est l'ensemble successeurs de la production.

En outre, deux symboles : et \rightarrow sont utilisés pour séparer les trois composants d'une reproduction : le prédécesseur, la condition et le successeur. Par exemple, une production avec $A(n)$ comme prédécesseur, $n > 6$ comme condition et $B(n + 1)$ B comme successeur s'écrit :

$$A(n): n > 6 \rightarrow B(n + 1)$$

Une production satisfait un module dans un mot paramétré si et seulement si les conditions suivantes sont vérifiées :

- la lettre du module correspond au prédécesseur de la production ;
- le nombre de paramètres du module est équivalent à celui des paramètres formels du prédécesseur de la production ;
- la condition appliquée aux paramètres formels de la production est évaluée à vrai.

Les L-systèmes ouverts.

Les L-systèmes ouverts [MP96] tiennent compte des interactions de l'organisme naturel avec le milieu dans lequel il habite. En effet, ils utilisent un symbole réservé pour la communication bilatérale. La figure 2.8 représente l'interaction entre une plante et son environnement.

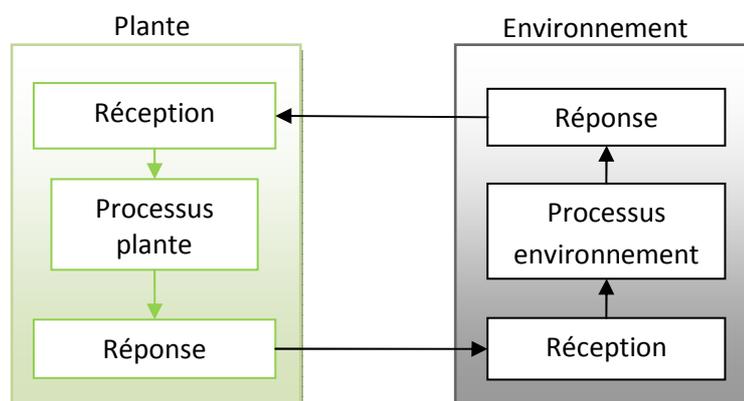


Figure 2.8-Communication entre plante et environnement dans les L-systèmes ouverts.

En bref, des paramètres associés à une occurrence du symbole de communication peuvent être placés par l'environnement et être transférés à la plante, ou placés par la plante et transférés à

l'environnement. Dans ce cas, L'environnement n'est plus représenté par une fonction simple, mais devient un processus actif qui peut réagir à l'information de la plante. Ainsi, les plantes sont modélisées en tant que systèmes cybernétiques ouverts, envoyant et recevant l'information de l'environnement.

2.3 Evolution et adaptation.

Dans *L'Origine des espèces*, Darwin [Dar59] affirme que l'apparition d'espèces distinctes se fait par le biais de la *sélection naturelle* de *variations individuelles*. Cette sélection naturelle est fondée sur la lutte pour la vie (*struggle for existence*), due à une population tendant naturellement à s'étendre mais disposant d'un espace et de ressources finis. Il en résulte que les individus les plus adaptés tendent à survivre plus longtemps et à se reproduire plus aisément. Le terme "adapté" se réfère à l'environnement, que l'on peut définir comme étant l'ensemble des conditions externes à un individu, ce qui inclut les autres individus. Le point clef est l'apparition, par hasard, de variations individuelles. C'est ce hasard qui permet d'expliquer les phénomènes d'évolution et d'adaptation sans avoir recourt ni à une création, ni à une modification directe de l'hérédité par le milieu.

Cette théorie de l'évolution a, cependant, du mal à s'imposer dans la communauté scientifique car elle ne fournit pas une explication sur l'origine des variations et la façon dont elles sont transmises aux générations. L'explication viendra plus tard avec la redécouverte des travaux de Mendel sur les "lois de l'hérédité" [Men07], puis de la structure de l'ADN [CW53] et, plus généralement de l'ensemble de la génétique moderne. Cette théorie synthétique de l'évolution a été baptisée *néo-darwinisme* et correspond aux bases de l'approche en vigueur dans la communauté scientifique.

Simuler l'évolution sur un ordinateur afin de créer des machines plus « intelligentes » anima Turing dès 1948 [Mou05]. Mais il faut attendre le milieu des années 1960 pour que se soient développés les premiers algorithmes appliquant les concepts du néo-darwinisme à l'informatique, baptisés algorithmes évolutionnistes³ (*evolutionary computation*).

2.3.1 Algorithmes évolutionnistes.

Les algorithmes évolutionnistes sont des méthodes d'optimisation stochastiques inspirées du paradigme de l'évolution darwinienne. Ces algorithmes ne s'appuient pas sur la connaissance des moyens de résolution d'un problème donné mais sur la définition même de ce problème. Du point de vue de la théorie de l'optimisation, ces algorithmes constituent des méta-heuristiques d'optimisation, au même titre que la descente du gradient [AH58], le recuit simulé [KGV83] et la recherche Tabou [GL97] (figure 2.9). Cependant, les algorithmes évolutionnistes se distinguent de ces techniques parce qu'ils

³ Le terme préféré dans les publications scientifiques est algorithmes évolutionnaire, ce terme est cependant incorrecte en français, pour cela nous utilisons algorithme évolutionniste, terme répertorié dans la bibliothèque TermSciences.

sont stochastiques pour la première ou parce qu'ils manipulent un ensemble de solutions plutôt qu'une solution individuelle pour la seconde et la troisième.

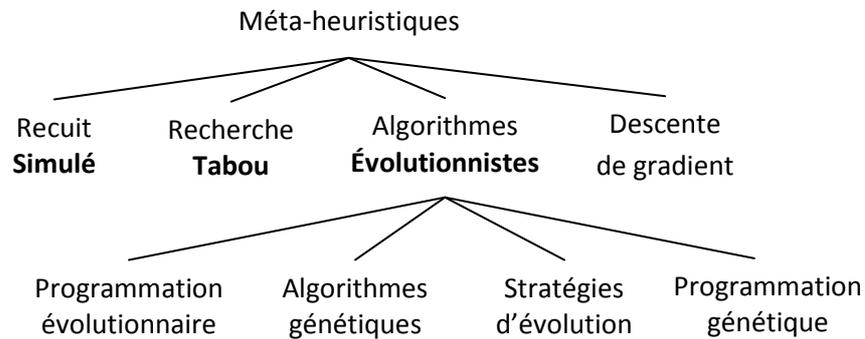


Figure 2.9-Place des algorithmes évolutionnistes comme méthodes d'optimisation.

Historiquement, les recherches ont menés à quatre grandes familles d'algorithmes évolutionnistes. Leurs principales différences viennent de la représentation utilisée pour l'information génétique :

- *La programmation évolutionniste* développée par Lawrence Fogel [FOW66], s'inspire de l'évolution organique au niveau des espèces, elle fait évoluer des automates à états finis.
- *Les algorithmes génétiques*, qui sont les plus connus et utilisés dans le calcul évolutionniste, développés par John Holland [Hol75] et publiés par David Goldberg [Gol89], codent l'information génétique sous la forme d'une chaîne binaire de taille fixe.
- *Les stratégies d'évolution*, développées par Ingo Rechenberg et Hans-Paul Schwefel [rec73, sch75] parallèlement avec les algorithmes génétiques, font évoluer des génomes de valeurs réelles se composant de deux types de gènes : des gènes discret représentant les caractéristiques de la solution et des gènes stratégiques propres au processus évolutionniste (concerne l'opérateur de mutation).
- *La programmation génétique*, conçue par John Koza [Koz92], consiste à faire évoluer des programmes représentés sous forme d'arbres. Les nœuds des arbres représentent des fonctions primitives du langage utilisé et les feuilles sont des symboles terminaux comme les variables ou les constantes.

Deux nouvelles familles ont émergé plus récemment, les algorithmes à estimation de distribution (EDA, voir par exemple [LL02]) et les algorithmes multi-objectifs (voir [Deb01]) [Mou05].

Tous les algorithmes évolutionnistes utilisent la même terminologie inspirée du langage de la génétique. Le tableau 2.1 indique les correspondances entre la terminologie biologique et artificielle utilisée dans les mécanismes d'évolution.

Terminologie biologique	Terminologie artificielle
Chromosome/ADN	Solution/ individu/ chromosome
Génome	Patrimoine génétique de la solution
Gène	Partie de la solution codée
Allèles	Allèles/valeurs possibles de la solution codée
Pool	Pool/ population
Génotype	Ensemble de gènes
phénotype	Solution /résultat

Tableau 2.1-Terminologie des algorithmes évolutionnistes.

2.3.2 Boucle générationnelle

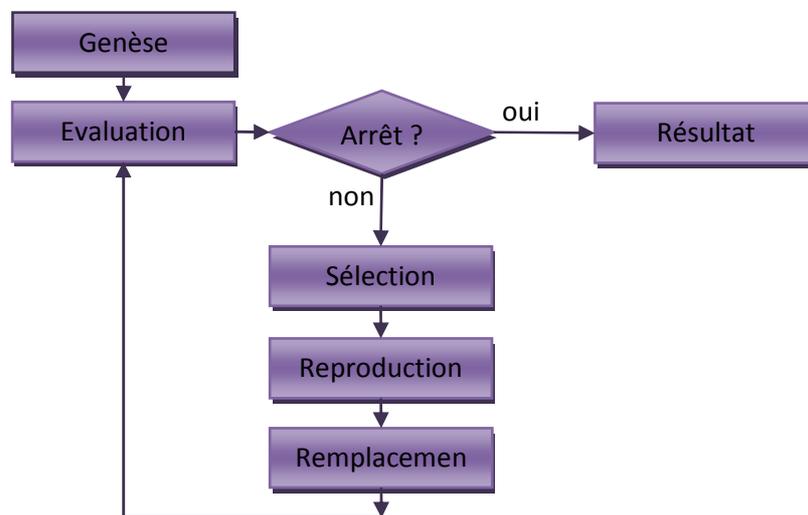


Figure 2.10- La boucle générationnelle de l'évolution.

Tous les algorithmes évolutionnistes reposent sur le même principe de base : la boucle générationnelle de l'évolution (figure 2.10). Tout d'abord, une population de solutions est générée aléatoirement. Chaque solution représente un génotype qui s'exprime en un phénotype. Après évaluation selon une fonction d'adaptation qui permet de distinguer les phénotypes les plus performants, la population va subir le processus de sélection (les individus les plus performants ont une probabilité plus importante de léguer à leur descendant leurs génotypes). Une fois les individus sélectionnés ont reproduits (par croisement et mutation), le remplacement va décider de la survie des solutions existantes et de la façon de choisir les individus nouvellement créés pour faire partie de la nouvelle population. La boucle est répétée jusqu'à atteindre un critère de terminaison (solution d'une qualité suffisante, temps limité,...).

Genèse.

La construction d'une population initiale représente le point de départ d'un algorithme évolutionniste. Elle est, la plupart du temps, réalisée aléatoirement mais en respectant certaines contraintes évidentes liées au problème. Dans le cas de la recherche de solution à une équation mathématique, on ne va, par exemple, pas générer d'individus de valeur négative si on cherche une solution dans \mathcal{M}^+ . L'optimisation de cette initialisation va en fait permettre d'augmenter la rapidité de convergence de l'algorithme en lui fournissant une population initiale plus homogène. Cette présélection risque néanmoins de supprimer des informations importantes dans le patrimoine génétique global de la population, empêchant ainsi l'algorithme d'atteindre le maximum global de sa fonction d'adaptation.

Il est également nécessaire de définir la taille de la population par génération. Elle ne doit pas être trop élevée pour des raisons de temps de calcul, ni trop faible pour que l'algorithme soit capable de trouver une solution convenable. Il est également possible de faire varier la taille de la population en fonction de l'état de l'algorithme, en vue d'optimiser dynamiquement le compromis entre complexité et performance.

Evaluation.

L'évaluation permet de quantifier la capacité de chaque individu à survivre. Cette quantité est appelée "degré d'adaptation" (fitness). Elle dépend de toute évidence du problème à résoudre. Son rôle est donc déterminant car elle va diriger l'évolution et "régler" la pression de l'environnement. Dans la majorité des cas, il s'agit d'une valeur réelle. Cette valeur est associée à une configuration du génotype et représente l'adaptation du phénotype qu'il représente.

Terminaison.

Après évaluation de la population, il est à décider de la continuité ou de l'arrêt de la boucle générationnelle. Pour cela, il existe différentes conditions d'arrêt, parmi les quelles nous pouvons citer :

- Arrêt après un certain temps limité,
- Arrêt après un certain nombre d'itérations,
- Arrêt lorsque la fitness d'un individu a atteint un certain seuil fixé.
- Arrêt de la convergence après un certain nombre d'itérations, c'est-à-dire la fitness de la meilleure solution n'augmente plus.

Dans la pratique, le critère d'arrêt de la boucle générationnelle dépend du problème à résoudre et il est à définir soigneusement par le créateur de l'algorithme.

Sélection.

La sélection décrit la manière dont les candidats sont choisis pour la reproduction. Par analogie avec la sélection naturelle, les solutions les plus performantes sont favorisées dans la transmission de leurs gènes à la génération suivante. Il existe de nombreuses techniques de sélection. Elles présentent différentes approches pour traiter le dilemme entre l'exploration de l'espace de recherche par diversité génétique et l'exploitation des meilleurs individus par pression sélective. Une pression sélective trop forte provoque la convergence prématurée dans un ou des optima locaux puisqu'elle tend à concentrer la recherche sur les meilleurs individus. En revanche, une pression sélective trop faible mène à une marche aléatoire des solutions et rend l'algorithme inefficace. Les méthodes principales sont :

- *La sélection à la roulette* [Gol89], la probabilité de sélection d'un individu est proportionnelle à sa note,
- *La sélection par rang* [Whi89], on ne choisit que les meilleurs individus,
- *La sélection par tournoi* [GD91], un tournoi consiste à prendre aléatoirement un sous-ensemble de n individus et à en déterminer le meilleur. Le gagnant est retenu pour la création de la génération suivante.

Comme ce sera le cas pour les autres opérateurs, il existe très peu de bases théoriques pour guider le choix de la méthode de sélection. Seuls des tests concernant une application spécifique permettent de déterminer l'approche la plus efficace.

Reproduction.

La reproduction se fait par le biais du croisement et de la mutation.

- *Le croisement* : permet d'améliorer les performances de la population en manipulant la structure des génotypes des individus. Les caractéristiques des enfants devraient être le fruit d'un mélange de celles des parents, cependant ce mélange peut se révéler bénéfique comme négatif du point de vue de la fitness.
- *La mutation* : apporte aux processus évolutionnistes la possibilité d'explorer l'ensemble de l'espace de recherche par diversité génétique. Quelque soit le codage utilisé, le principe est le même : ajouter de petites modifications au génotype afin d'explorer de nouvelles solutions potentielles au problème traité.

Les deux opérateurs, croisement et mutation, dépendent du codage utilisé. Dans le cas des algorithmes génétiques classiques, les solutions potentielles sont représentées par des chaînes binaires. L'opérateur principal est alors le croisement auquel s'ajoute la mutation. Les opérateurs les plus utilisés sont: *le croisement à un point* [Hol62], *le croisement à deux points* [SP91], *le croisement uniforme* [Sys89], *le croisement multi-parent* [EVK95], *la mutation bit flip* [Gol89].

Dans le cas des structures arborescentes de la programmation génétique, le croisement dit "standard" consiste à déterminer aléatoirement un nœud sur chacun des deux parents, puis

à intervertir les sous-arbres enracinés au niveau de ces nœuds [Koz92]. Au fil du temps, une variété de croisements complémentaires a été développée [AK96]. Lors d'une mutation, le sous-arbre enraciné au niveau du nœud affecté est typiquement remplacé par un arbre créé de façon aléatoire.

À l'inverse de ces deux premiers types d'algorithmes évolutionnistes, les stratégies d'évolution et la programmation évolutionniste utilisent essentiellement l'opérateur de mutation. Dans la dernière, il n'y a pas de différence entre le phénotype et le génotype : comme l'algorithme était destiné à être utilisé principalement sur des automates d'états finis, le génotype représente directement cet automate. Lors de la reproduction, il n'y a pas de croisement et les mutations se font directement sur l'automate en ajoutant ou en supprimant des états ou des liens.

Enfin, dans les stratégies d'évolution, le génome est un vecteur de nombres réels. Les enfants seront créés après avoir subi des mutations qui correspondent à un bruit gaussien sur les valeurs réelles. Le taux de mutation⁴ peut lui-même être codé dans le génome et subir donc l'évolution. De plus, il existe quatre croisements différents : *le croisement local ou global* [sch95], *le croisement discret*, *le croisement intermédiaire (arithmétique)*, *Le croisement intermédiaire BLX- ϕ (Blend Crossover)* [ES93].

En général, les deux opérateurs de reproduction sont appliqués avec des probabilités à définir soigneusement. Si elles sont trop faibles, l'algorithme est lent et peu efficace. En revanche, dans le cas de probabilités trop élevées, ces opérateurs risquent de détruire trop de matériel génétique, ce qui se traduit par une recherche aléatoire sans progrès.

Remplacement.

Une fois la reproduction effectuée, une nouvelle population est mise en place. Ici, il s'agit de décider de la survie des solutions existantes et de la façon dont les nouveaux individus, qui feront partie de la population, sont choisis. D'une manière générale, il existe trois méthodes de remplacement :

- *Remplacement générationnel* : la nouvelle population est entièrement formée à chaque génération. Certains individus enfants peuvent être une copie exacte des parents qui n'ont pas été perturbés par les opérateurs génétiques de croisement et de mutation. Dans les versions dites "élitistes", le ou les meilleurs individus de chaque génération sont recopiés dans la population de la génération suivante, afin d'écartier la possibilité de perdre les solutions d'une pertinence déjà acquise.
- *Remplacement stationnaire (steady-state)* : L'idée principale est qu'une grande partie de la population puisse survivre à la prochaine génération. Ainsi, seulement un ou quelques individus sont remplacés [Rya00]. Les individus à retirer de la population sont typiquement les moins performants ou bien les plus âgés.

⁴ Le taux de mutation peut être une seule valeur ou un vecteur de réel.

- *Remplacement déterministe* : dans les stratégies d'évolution, il existe deux schémas de remplacement, le remplacement déterministe (μ, λ) et $(\mu + \lambda)$. A chaque génération les λ enfants sont générés à partir des parents sélectionnés par tirage uniforme (pas de sélection au sens darwinien). Dans le schéma (μ, λ) , les meilleurs μ enfants en termes de fitness deviennent les parents de la génération suivante. Dans le schéma $(\mu + \lambda)$, les meilleurs des $\mu + \lambda$ parents plus enfants constituent les parents de la génération suivante. Cette deuxième approche est donc élitiste car les solutions les plus performantes d'une génération ne sont pas perdues. Ce dernier schéma ressemble à celui utilisé en programmation évolutionnaire, quoique développé complètement indépendamment. Cependant le nombre d'enfants et celui des parents est identique : $(\mu + \mu)$.

2.3.3 Synthèse.

Les principales différences entre les quatre familles des algorithmes évolutionnistes viennent de la représentation choisie pour le code génétique des individus et les différents opérateurs utilisés (sélection, reproduction, remplacement). Comme nous l'avons déjà vu, les opérateurs de reproduction dépendent du codage utilisé. Les algorithmes génétiques et la programmation génétique se basent essentiellement sur l'opérateur de croisement alors que la programmation évolutionnaires et les stratégies d'évolution utilisent essentiellement la mutation.

La sélection et le remplacement sont indépendants de la représentation choisie. De plus, ils sont liés puisqu'ils permettent de créer la nouvelle génération des parents à partir des enfants (en deux temps : sélection puis remplacement). Un de ces deux temps est toujours stochastique, l'autre est souvent déterministe.

Cependant, en fonction des algorithmes, le déterminisme ne porte pas sur le même opérateur. En effet, dans les algorithmes génétiques et la programmation génétique, c'est l'opérateur de remplacement qui est souvent déterministe tandis que l'opérateur de sélection est toujours stochastique. Pour les stratégies d'évolution et la programmation évolutionnaire c'est l'inverse.

Un opérateur déterministe permet, tout simplement, d'obtenir un ensemble d'individus de taille donnée à partir d'un ensemble plus grand, en prenant directement les meilleurs individus.

	Algorithmes Génétiques	Programmation évolutionnaire	Stratégies d'évolution	Programmation génétique
auteurs	J. Holland	L. Fogel	I. Rechenberg, H. Schwefel	J. Kosa
codage	chaîne de bits	automates à états finis	vecteurs de réel	programmes
sélection	roulette, tournois, rang, ...	déterministe	déterministe	Roulette, tournois, rang, ...
croisement	à un point, à deux points, uniforme multi-parent	pas de croisement	Local/global, Discret/intermédiaire ou BLX- ϕ	standard
mutation	bit flip	modifications directes	bruit Gaussien	Remplacement d'un sous-arbre
remplacement	générationnel, stationnaire	schéma ($\mu + \mu$)	schéma(μ, λ), ($\mu + \lambda$)	générationnel, stationnaire

Tableau 2.2: Principales différences entre les différents types d'algorithmes évolutionnistes.

Actuellement, la distinction entre ces quatre familles est de plus en plus floue, le génotype étant souvent constitué d'un mélange de structures complexes (arbres, graphes, listes de paramètres, etc.). La différence entre ces quatre approches est donc essentiellement d'ordre historique. Ces familles sont aujourd'hui considérées comme des instances particulières des *algorithmes évolutionnistes* [de 02].

2.4 Apprentissage.

Dans cette section nous présentons les mécanismes d'apprentissage utilisés dans la vie artificielle. Tout système d'apprentissage fonctionne de la même façon, il *“agit sur une structure de données définissant le comportement d'un système en modifiant cette structure de manière à adapter le comportement à un environnement. L'adaptation se fait généralement en fonction d'un «feedback» de la part de cet environnement vers le système d'apprentissage pour moduler la modification de la structure de données ”* [Lug09]. Les systèmes d'apprentissage les plus connus et utilisés sont les réseaux de neurones et les systèmes de classeurs.

2.4.1 Réseaux de neurones.

Les réseaux de neurones (*Artificial Neural Network*) s'appuient, comme la plupart des mécanismes précédents, sur une analogie d'un phénomène naturel : le fonctionnement des cellules nerveuses. Ici, on va simuler le fonctionnement des cellules présentes dans le système nerveux des êtres vivants (humains ou non). Pour cela, on assemble ces cellules de façon à former des réseaux qui pourront, après une phase d'apprentissage, être utilisés

comme mécanisme de décision dans des systèmes dynamiques intégrés dans des entités possédant des capteurs et des effecteurs.

Neurone biologique et neurone artificiel.

Le système nerveux de la plupart des espèces naturelles est constitué d'un réseau de cellules spécialisées appelées neurones. Un neurone est constitué de :

- Dendrites, servent à capter les signaux envoyés au neurone ;
- Le corps cellulaire, contient les molécules essentielles à la vie de la cellule ;
- axone, un axone peut être relié aux dendrites d'un autre neurone ou à des effecteurs tels que les muscles ;
- Synapses, c'est une jonction entre deux neurones, généralement entre l'axone d'un neurone et une dendrite d'un autre.

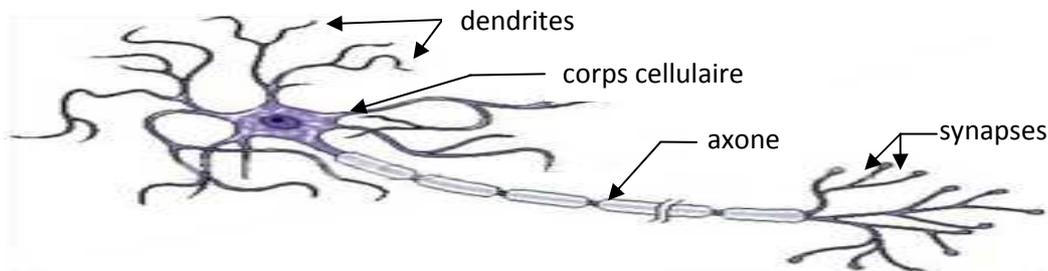


Figure 2.11- Neurone biologique.

Le neurone reçoit des informations, venant d'autres neurones, grâce à ses dendrites. Il va ensuite y avoir sommation, au niveau du corps cellulaire, de toutes ces informations et via un potentiel d'action (un signal électrique). Le résultat va transiter le long de l'axone jusqu'aux terminaisons synaptiques. A cet endroit, lors de l'arrivée du signal, des neurotransmetteurs (médiateurs chimiques) vont se libérer dans la fente synaptique.

Le signal électrique ne pouvant pas passer la synapse, les neurotransmetteurs permettent donc le passage des informations, d'un neurone à un autre. Les neurotransmetteurs excitent ou inhibent le neurone suivant et peuvent ainsi générer ou interdire la propagation d'un nouvel influx nerveux.

Un neurone artificiel [MP43] est une abstraction mathématique d'un neurone biologique. Il possède généralement plusieurs entrées (E_1, E_2, \dots, E_n) et une sortie (S) qui correspondent respectivement aux dendrites et à l'axone.

Les actions excitatrices et inhibitrices des synapses sont représentées, par des coefficients numériques (w_1, w_2, \dots, w_n); des poids synaptiques associés aux entrées. Si le poids est positif, on parle de synapse excitatrice alors que s'il est négatif, on parle de synapse inhibitrice. Les poids synaptiques sont ajustés dans une phase d'apprentissage. Dans sa version la plus simple, un neurone formel calcule la somme pondérée des entrées reçues, puis applique à

cette valeur une *fonction d'activation* (ou *fonction de transfert*). Le résultat obtenu est la sortie du neurone.

Réseau de neurones.

Un réseau de neurone est un ensemble de neurones formels interconnectés, associés en couches et fonctionnant en parallèle. L'information donnée au réseau se propage couche par couche, de la couche d'entrée à la couche de sortie, en passant soit par aucune couche, par une couche ou par plusieurs couches intermédiaires ("*hidden layout*").

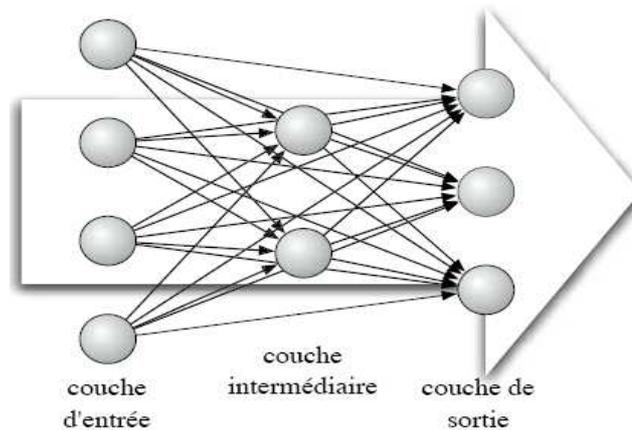


Figure 2.12- Exemple d'un réseau de neurone : le perceptron multi-couches.

Malgré la simplicité de fonctionnement d'un neurone c'est l'interconnexion d'un certain nombre d'entre eux qui rend le comportement de l'ensemble très complexe. Si les interconnexions vont de la couche d'entrée vers la couche de sortie sans retour en arrière, le réseau est appelé perceptron ("*feedforward*"). Si, par contre, les interconnexions peuvent boucler à l'intérieur du réseau, on l'appelle réseau *récurrent* ("*feedback*"). Si, par ailleurs, chaque neurone est relié à tous les autres neurones, on dit du réseau qu'il est "*complet*".

Apprentissage.

L'apprentissage consiste à mettre à jour les poids des connexions au sein du réseau, afin qu'il réussisse la tâche qui lui est demandée de réaliser. Plusieurs méthodes d'apprentissage existent :

- *La rétro-propagation* (mode supervisé). Dans cet apprentissage, l'environnement fournit au réseau des couples entrées/sorties qui vont former un jeu d'entraînement. le réseau va mettre à jour ses poids en utilisant la différence entre le résultat qu'il a calculé, en fonction des entrées fournies, et la réponse attendue en sortie (la réponse donnée). Ainsi, le réseau va se modifier jusqu'à ce qu'il trouve la bonne sortie. On l'appelle le mode supervisé car l'environnement doit fournir la sortie correcte pour chaque jeu d'entrées, jouant ainsi le rôle du superviseur. On emploie généralement le terme de feedback instructif.

- L'apprentissage par renforcement. Cet apprentissage diffère légèrement du précédent. En effet, au lieu de fournir la sortie correcte à une entrée donnée, l'environnement fournit au réseau une valeur rendant compte son adaptation à l'environnement. Le réseau apprend en essayant de maximiser cette valeur. Ici on parle de feedback évaluatif.
- L'apprentissage non supervisé ou auto-organisationnel. Cet apprentissage ne repose pas sur la notion de feedback. Il est basé sur des probabilités.

2.4.2 Systèmes de classeurs.

Les systèmes de classeurs, introduits par Holland en 1976 [Hol76], sont des systèmes capables d'interagir avec l'environnement dans lequel ils évoluent et qui sont dotés de capacités d'apprentissage. Un système de classeurs gère une base de règles appelées classeurs. Ces règles sont de la forme : condition-action et peuvent être pondérées par des attributs de qualité [BSD06]. Le système possède un cycle de fonctionnement au cours duquel il perçoit son environnement, en déduit les règles applicables, puis exécute une action issue de ces règles et peut percevoir une rétribution de l'environnement (figure 2.13). C'est ce feedback (rétribution de l'environnement) qui est utilisé par la suite pour modifier les règles ou leurs attributs de qualité. Le système de classeur apprend, par expérimentation, l'association de conditions et d'actions maximisant la récolte de rétributions. A la fin de l'apprentissage, la base de règles contient la(les) solution(s) au problème posé.

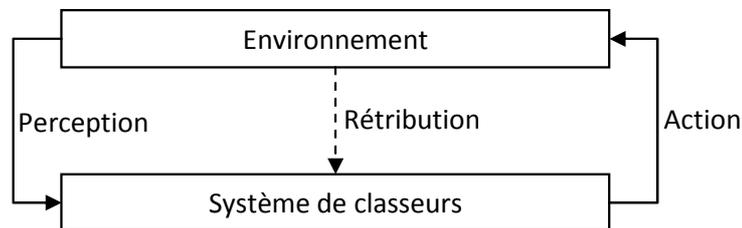


Figure 2.13- Interactions entre l'environnement et le système de classeurs.

Fonctionnement.

Le fonctionnement d'un système de classeurs est présenté dans la figure 2.14 et s'appuie sur une structure globale d'un 7-uplet : $(I_e, [P], [M], [A], \text{Comparaison}, \text{Sélection}, I_s)$ où :

- I_e est l'interface d'entrée. Elle fait correspondre à toute perception de l'environnement un code (binaire ou autre en fonction du formalisme de représentation) ;
- $[P]$, appelé population, est l'ensemble des classeurs du système ;
- $[M] \subseteq [P]$ est l'ensemble des classeurs dont la partie condition s'apparie avec les informations perçues de l'environnement pour un cycle de sélection. Il est appelé Match-set ;

- $[A] \subseteq [M]$ est l'ensemble des classeurs représentant l'action sélectionnée. Il est appelé Action-set ;
- Comparaison est le mécanisme permettant de passer de [P] à [M]. Il s'agit généralement d'une règle d'appariement entre la condition et l'information provenant de le ;
- Sélection est le mécanisme permettant de passer de [M] à [A]. Il détermine, en fonction de critères spécifiques aux différentes versions de systèmes de classeurs, l'action choisie.
- Is est l'interface de sortie faisant correspondre à un code donné l'activation d'Action.

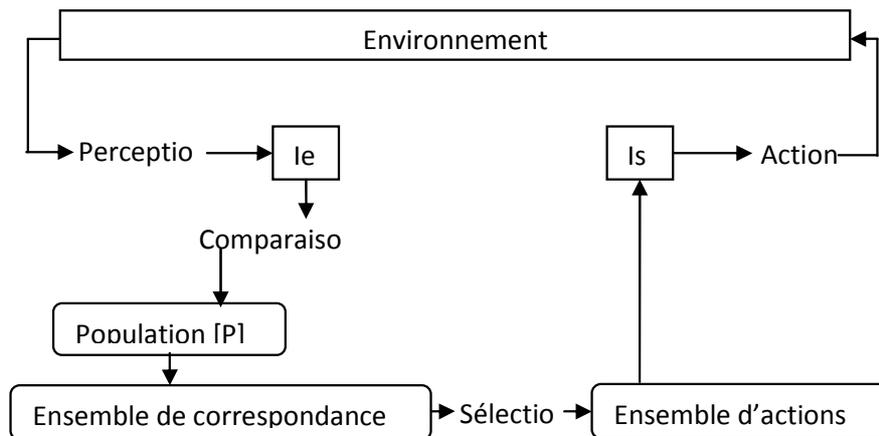


Figure 2.14- Schéma de fonctionnement d'un système de classeurs.

Apprentissage.

Dans les systèmes de classeurs, l'apprentissage se fait par mise à jour de la qualité des règles, grâce à des algorithmes d'apprentissage par renforcement (par exemple : Qlearning [Wat89], Bucket Brigade [Hol86]) et par Génération de règles à l'aide d'algorithmes évolutionnistes et d'opérateurs de recouvrement (*covering*). La dynamique des systèmes de classeurs apprenants est alors basée sur le cycle : Perception / Comparaison / Evolution (*recouvrement*) / Sélection / Action / Evaluation / Evolution (algorithme évolutionniste). Avant de passer en revue les principales familles des systèmes de classeurs, nous expliquons d'abord les mécanismes de sélection d'évaluation et d'évolution.

1/ Mécanisme de Sélection : La sélection est guidée par la qualité des règles qui sont regroupées selon leur partie A. Pour choisir l'action à exécuter, il existe deux stratégies:

- dans l'exploitation, la sélection est déterministe et basée sur la meilleur qualité, les algorithmes génétiques sont inhibés ;
- dans l'exploration, la sélection est soumise à une probabilité permettant de sélectionner aléatoirement ou de sélectionner le classeur ayant la meilleure qualité. Souvent, c'est un mécanisme de sélection à la roulette.

2/ Mécanisme d'Evaluation : Ce mécanisme aussi appelé, mécanisme de rétribution (ou mécanisme de renforcement), consiste à distribuer les rétributions aux règles qui ont contribué à les obtenir. Il augmente la qualité des règles déclenchées précédemment à l'obtention de la rétribution et diminue les autres. Sa définition influe sur la longueur pertinente d'une chaîne d'actions: nombre d'enchaînements de règles considérées nécessaires pour atteindre un but.

3/ Mécanisme d'Evolution : Ce mécanisme, également appelé mécanisme de génération, permet de créer ou de supprimer des classeurs afin d'éviter l'explosion combinatoire de règles tout en préservant celles permettant la récolte de rétributions. Pour cela, il existe deux mécanismes: le *recouvrement* et les *algorithmes génétiques*.

- Le *recouvrement* ou l'opérateur de couverture est invoqué dans chaque situation où l'ensemble [M] est vide ou dans le cas où il contient des classeurs dont la *qualité* est considérée insuffisante. Cet opérateur intègre dans la population un nouveau classeur activable dans la situation courante, dont l'action est choisie au hasard et dont la *qualité* est égale à la *qualité* moyenne de la population. Chaque test de la partie Condition peut prendre la valeur # avec une probabilité P(#).
- Les *algorithmes génétiques* [Hol75, Gol89] sont utilisés pour générer de nouvelles règles à partir de celles existantes. Le choix des règles servant de base pour les opérations génétiques est généralement effectué par une sélection à la roulette basé sur la fitness. Les algorithmes génétiques peuvent intervenir sur les règles se situant dans [P] ou dans [A] selon les versions de système de classeurs [Wil94, Wil95]. La fréquence d'utilisation des algorithmes génétiques par rapport au cycle d'un système de classeurs est un facteur important pouvant influencer sensiblement les performances d'apprentissage.

Approche Pittsburgh versus Michigan.

Les systèmes de classeurs peuvent être classés en deux catégories: les approches de type Pittsburgh et les approches de type Michigan. Dans les systèmes proposés par Holland et qui constituent, le type Michigan [HR78], l'algorithme génétique est combiné avec un mécanisme d'apprentissage par renforcement et s'applique à l'intérieur d'un seul système de classeurs où chaque règle représente un individu.

Dans l'approche de type Pittsburgh [Smi80], le seul mécanisme adaptatif mis en œuvre est un AG qui s'applique à une population de systèmes de classeurs pour choisir parmi cette population le système de classeurs le plus adapté à un problème donné. L'approche de type Michigan s'est rapidement imposée comme le cadre standard des systèmes de classeurs. Cependant, on assiste, à présent, à un renouveau des approches de type Pittsburgh [LG02, BGB07, LSS05, LS08].

La vie artificielle utilise des mécanismes, qui sont dans la plupart des cas inspirés du vivant, tels que les mécanismes de génération, d'évolution et d'apprentissage présentés précédemment, pour simuler le vivant : synthétiser des créatures artificielles.

2.5 Créatures Artificielles.

Le développement des créatures artificielles a commencé d'abord en synthèse d'image vers les années 80-90. Les chercheurs ont travaillé sur la simulation des comportements des créatures connues sous le nom d'acteurs ou agent virtuels. Partant d'une forme prédéfinie, l'évolution porte uniquement sur le comportement. Après, on a commencé à faire évoluer conjointement la forme et le comportement des créatures pour qu'elles puissent s'adapter à l'environnement. Le premier à avoir travaillé sur la coévolution et même le premier à avoir utilisé le terme créatures artificielle est Karl Sims. Depuis une dizaine d'années, un certain nombre de travaux tendent à produire des créatures basées sur une unité de base bien plus petite : la cellule artificielle.

Pour classer les travaux sur la synthèse de créatures artificielles, nous utilisons les trois modèles qu'a utilisés Daniel Mange pour décrire toute entité vivante.

*Tous les êtres vivants peuvent être décrits par trois grands modèles. Selon le modèle **Phylogénétique**, ils sont capable d'évoluer, leur matériel génétique peut se transformer pour donner naissance à des descendants différents ; selon le modèle **ontogénétique**, les êtres vivants ont la capacité de croître à partir d'une cellule mère, par une série de divisions et de différenciations cellulaires ; le troisième est le modèle **épi-génétique**, selon lequel ils sont capables d'apprendre grâce à des systèmes particuliers comme le système nerveux ou le système immunitaire.*

Daniel Mange [Man06]

Les travaux tendant à simuler le comportement des créatures, utilisent généralement des mécanismes d'apprentissage comme les réseaux de neurones et les systèmes de classeurs. Ces travaux peuvent être classés dans la catégorie des systèmes épi-génétiques. Les systèmes phylogénétiques peuvent regrouper les modèles utilisant un mécanisme évolutionniste de type algorithme génétique pour faire la coévolution de la forme et du comportement de la créature. Finalement, les systèmes tendant à développer la créature à partir d'une cellule initiale peuvent être regroupés dans la catégorie des systèmes ontogénétiques.

2.4.3 Epigenèse.

Dans cette catégorie, on classe les travaux qui utilisent des méthodes évolutionnistes couplés à des systèmes d'apprentissage pour faire émerger des comportements des créatures (fuite, poursuite, stratégies d'équipe...) sans décomposer précisément le problème

à résoudre. Dans ce cas là, la morphologie est fixée et seule une modélisation générique du contrôleur ainsi qu'une fonction d'évaluation sont nécessaires. La pression de l'évolution génétique permettra l'émergence d'un contrôleur parfaitement adapté au problème posé. L'apprentissage se fait à l'aide des réseaux de neurones ou des systèmes de classeurs.

Poissons artificiels de Terzopoulos.

Dimitri Terzopoulos est un des premiers à utiliser cette technique dans l'animation comportementale [TT94]. Il développe des poissons dont le squelette est dessiné par la main de l'homme et dont le comportement définit par des réseaux de neurones évolués à l'aide d'un algorithme génétique. Il obtient grâce à cela une nage des poissons très réaliste dans un simulateur physique (figure 2.15(a)).

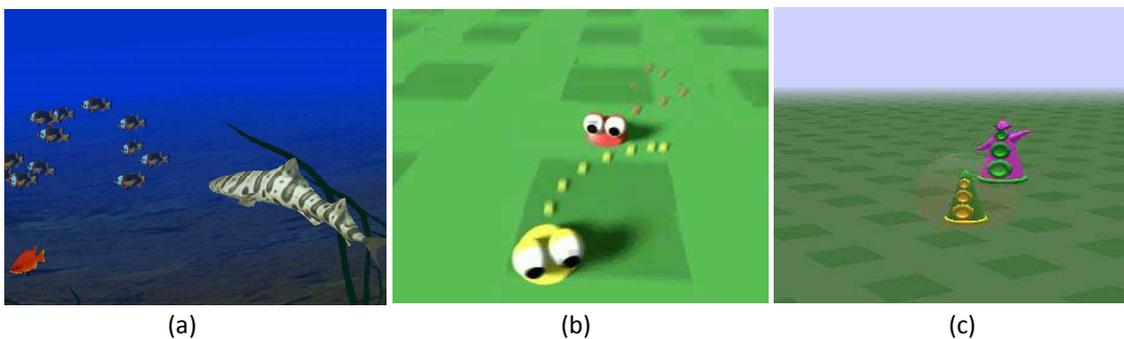


Figure 2.15- *Simuler le comportement des créatures à l'aide de réseau de neurones. (a) Poisson artificielles de Demetri Terzopoulos. (b) co-évolution des deux créatures Cliff et Miller : comportement de fuite (jaune) et de poursuite. (c) comportement d'anticipation de David Panzoli.*

Comportement de fuite et de poursuite de Cliff et Miller.

Dave Cliff et Geoffrey Miller [CM96] utilisent l'évolution pour obtenir des comportements simples de fuite et de poursuite (figure 2.15(b)). Les deux créatures sont contrôlées par des réseaux de neurones récurrents évolués par algorithme génétique. Un point important de ce travail, réside dans la coévolution, ce qui signifie que l'évolution ne concerne pas chaque créature dans son environnement mais les deux, l'une par rapport à l'autre.

Stratégies d'équipe de Cédric Sanza et Olivier Heguy.

Cédric Sanza [San01] a utilisé des systèmes de classeurs pour produire des agents virtuels capables d'élaborer des stratégies d'équipe dans le cadre de jeux de football (figure 2.16(a)).

Comportements anticipatifs de Rasmus.

Rasmus [Ram07] utilise un système de classeurs pour reproduire puis étudier des comportements anticipatifs dans une simulation de chien de berger.

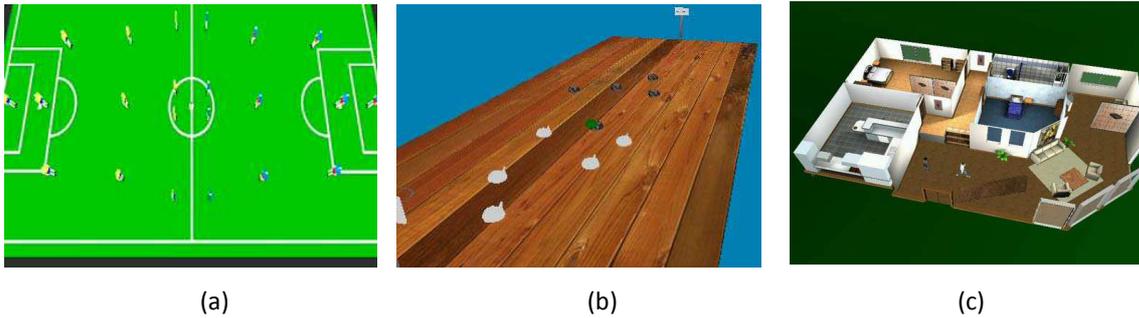


Figure 2.16-Simuler le comportement des créatures à base de système de classeur (a) stratégie d'équipe dans un jeu de football de Cédric Sanza. (b) Stratégie d'équipe dans le cadre de jeu de basketball d'Olivier Heguy. (c) Comportement réaliste des humanoïdes de Stéphane Sanchez.

Autres travaux.

Olivier Heguy [Heg03] a également utilisé des systèmes de classeurs pour faire évoluer des comportements ou des stratégies d'équipe mais dans un jeu de basket-ball virtuel (figure 2.16(b)). David Panzoli quant à lui, a utilisé des réseaux de neurones pour faire évoluer des stratégies d'anticipation. En n'exprimant également que le but global du système Stéphane Sanchez [SBLD04] a conçu l'architecture ViBes (*Virtual Behaviors*) basée sur les agents hybrides, dans la quelle des humanoïdes virtuels doivent être capables de se comporter de manière réaliste. En outre, ils doivent apprendre à manipuler et interagir avec les objets qui les entourent (figure 2.16(c)).

2.4.4 Phylogenèse.

La phylogenèse artificielle consiste à générer des créatures artificielles en réalisant l'évolution conjointe de la morphologie et du comportement. La morphologie est générée en utilisant des composants déjà prêts comme des blocs, des bâtonnets, des cylindres, comme organes de créatures. Le comportement d'une entité est défini en créant des liens entre ses capteurs et ses effecteurs. Ces derniers contrôlent les jointures entre les différentes parties de la créature et permettent ainsi de la faire bouger en modifiant les angles, vitesses, etc. appliqués à ces jointures. Dans ce type d'approche, on utilise généralement une fonction d'évaluation (fitness) pour diriger l'évolution. L'environnement simulé est tridimensionnel et intègre un simulateur physique. Les créatures sont plongées dans cet environnement et confrontées à des problèmes à résoudre qui constituent dans la plupart des cas leurs fonctions d'évaluation : par exemple si la tâche consiste à nager, la fonction d'évaluation serait la capacité de nager mesurée par exemple par la distance parcourue en nageant.

Créatures de Karl Sims.

Les créatures de Karl Sims [Sim94a; Sim94b; Sim94c] restent parmi les premiers travaux les plus évolués et les plus convaincants du domaine. Ce dernier a montré que l'évolution

artificielle est apte de produire de manière complète et réaliste des créatures artificielles, chose difficile à réaliser en réalité virtuelle. Il propose, en effet, d'utiliser les algorithmes génétiques afin de faire évoluer la morphologie et le comportement des créatures artificielles. Pour générer la morphologie, il utilise un graphe orienté récursif (figure 2.17) permettant de générer une grande variété de formes à partir d'une représentation très compacte. Pour simuler le contrôleur, on utilise un réseau de neurones.

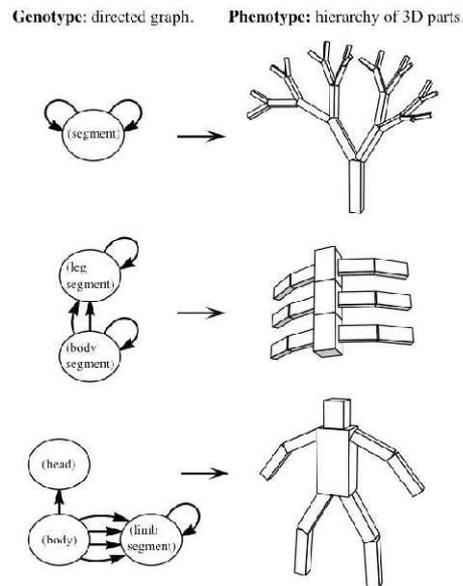


Figure 2.17-Graphe de développement.

Le génotype est un graphe orienté contenant des informations définissant la morphologie et le contrôleur de la créature. Ce graphe se compose de nœuds et de connexions dont :

- Les nœuds contiennent la description des blocs : leur taille, le type de joint qui les relie à leur parent.
- Chaque nœud peut être connecté sur lui-même ou à d'autres nœuds ;
- Les connexions entre les blocs indiquent le placement du bloc relativement à leur bloc parent, la symétrie, l'échelle et l'orientation ;
- Chaque bloc possède un réseau de neurones ;
- les réseaux de neurones sont reliés entre eux par un réseau de neurones qui supervise le comportement de la créature ;
- Les neurones sont des fonctions mathématiques telles que des fonctions sinusoïdales.

Le réseau de neurones prend en entrée les valeurs des capteurs qui mesurent différents aspects du monde tel que le contact entre deux surfaces, l'angle entre deux blocs. Ensuite, des effecteurs agissent sur le monde virtuel en actionnant les muscles au niveau des joints.

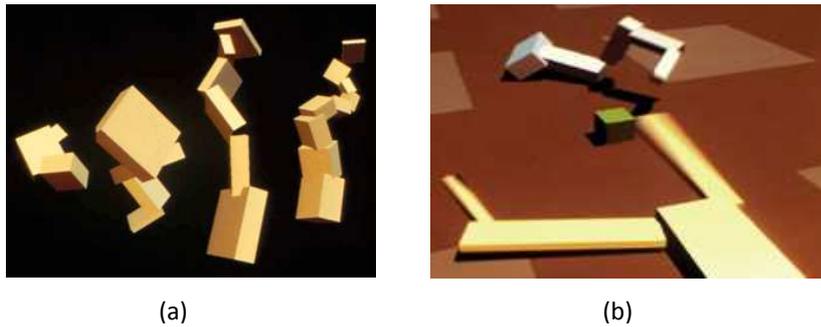


Figure 2.18-Créatures de Karl Sims. (a) Des créatures nageant. (b) Deux créatures en lute pour s'emparer d'un cube.

Durant chaque simulation, une créature doit réaliser une tâche, qui constitue sa fonction d'évaluation. Dans la première partie de ses travaux, Sims obtient une grande variété de créatures pouvant se déplacer, nager, suivre une source lumineuse. Dans la seconde partie de ces travaux, il met directement les créatures en confrontation pour essayer de s'approprier un cube. Ces résultats sont très intéressants, et après beaucoup de chercheurs ont tenté de les reproduire.

Créatures de Gregory Hornby.

Beaucoup de chercheurs ont tenté de reproduire les résultats de Sims avec différentes approches pour la représentation de la morphologie et du contrôleur. Hornby [HP01, HP02, PHLF03] proposa l'utilisation de L-systèmes pour la génération de la morphologie. Le contrôleur utilise des réseaux de neurones. Les créatures développées sont d'une grande variété et possèdent un grand nombre de segments. De plus, elles ont été réalisées sous forme de physique (robots).

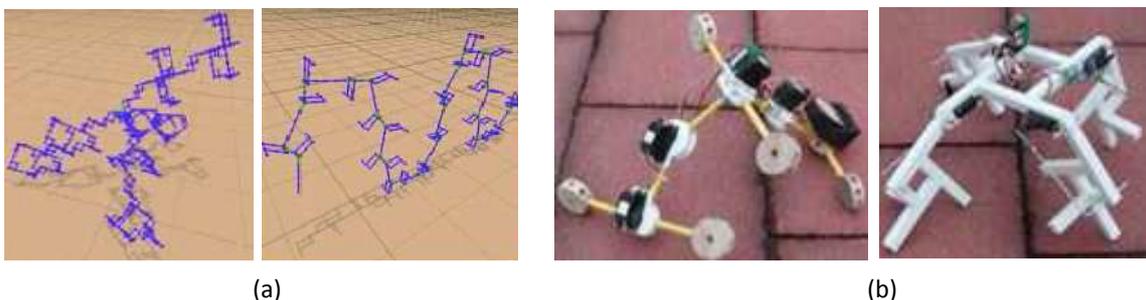


Figure 2.19-Créatures de Hornby.(a) Dans le simulateur. (b) Réalisation sous forme physique (robot).

Framsticks.

Framsticks est un projet réalisé par Komosinski [Kom00, Kom03, Kom05, AK05], dans lequel les créatures sont assimilables à celles de Karl Sims. Elles évoluent cependant dans un environnement plus riche. Cet environnement contient des sphères énergétiques que les

créatures doivent s'approprier pour survivre. De plus, un nombre plus grand de capteurs est spécifié (des capteurs pour le sens tactile, des capteurs pour le sens de l'équilibre,..., etc.). En outre, l'évolution est dirigée soit par une fonction d'évaluation soit par la sélection naturelle. La morphologie des créatures est composée de bâtonnets (*sticks*), chacun peut être spécialisé dans une fonction : assimilation, ingestion, ... Les mouvements sont accomplis par un réseau de neurones qui agit sur les jointures (muscles en rouge sur la figure 2.20).



Figure 2.20-Framsticks. (a) créature terrestre. (b) Créature aquatique.

Créatures de Nicolas Lassabe.

Lassabe [LLD07a, LLD07b, LLD07c] propose l'utilisation d'un nouveau type de contrôleur basé sur un système de classeurs. La morphologie de ses créatures est générée en utilisant un graphe orienté comme dans le modèle original de Karl Sims.

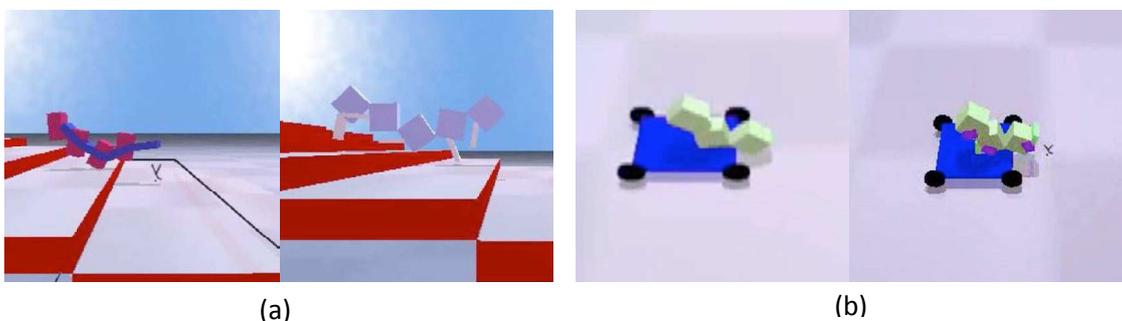


Figure 2.21-Créatures de Lassabe. (a) La montée d'escalier. (b) le déplacement avec un skateboard.

Afin de valider son approche, il propose d'utiliser des environnements plus complexes. Pour cela, les créatures ont été confrontées à des situations mettant en jeu des objets statiques : escalier, murs, branchées. La figure 2.21 (a) présente des créatures essayant de monter un escalier. Ensuite, une autre série d'expérimentations a été menée dans le but d'étudier l'interaction entre les créatures et un objet mobile. La figure 2.21 (b) présente un ensemble de créatures se déplaçant avec un skateboard.

Autres approches.

Beaucoup de chercheurs ont reproduit les créatures de Karl Sims, citons Miconi [MC05, MC06a, MC06b, Mic07] qui a fait évoluer des créatures très semblables mais sans obtenir

des résultats meilleurs, il a cependant amélioré les algorithmes de coévolution. Nicolas Chaumont [CEA07, CEA08] a proposé en plus, d'évaluer les créatures sur leur capacité de catapulter un block. Afin d'améliorer les performances, Tim Taylor et Colm Massey [TM00, Tay00] ont proposé l'utilisation de cylindres à la place des parallélépipèdes rectangles. Ray [Ray00] quant à lui, a utilisé un algorithme évolutionniste interactif, où l'utilisateur sélectionne les créatures selon, par exemple, un but esthétique. Ses créatures colorées évoluent dans un environnement liquide. Les autres travaux qui ont utilisés la sélection par utilisateur sont ceux de Karl Sims avec Genetic Image et Galapagos dans lequel il a fait évoluer des plantes virtuelles [Sim91].

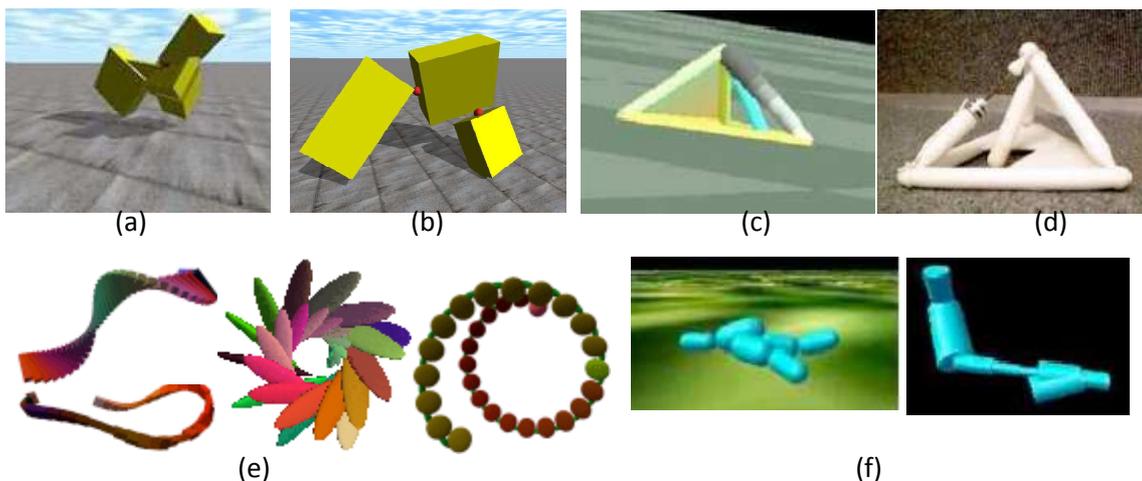


Figure 2.22-Panel de créatures artificielles. (a)créatures de Miconi, (b) créatures de Chaumont, (c, d) créatures de Lipson dans le simulateur, sous forme physique (d), (e) Créatures de Ray, (f) créatures de Taylor.

Parmi les créatures finalisés du domaine sont on trouve ceux de Hold Lipson [LP00], réalisées dans le cadre du projet Golem. Ses créatures sont composées essentiellement d'éléments dits 1D, tubes et de pistons. Le but est de réaliser des robots avec le moins d'interventions humaines. Pour cela, ils utilisent aussi l'évolution afin de trouver la morphologie et le contrôleur des robots qui sont ensuite manufacturés par une imprimante tridimensionnelle. L'homme n'intervient sur le processus de fabrication qu'à la fin où il assemble les moteurs aux structures imprimées.

Ecosystèmes.

Un écosystème artificiel permet de faire co-évoluer un ensemble de créatures (plantes virtuelles comprises) en même temps dans un même environnement. L'évolution, ici, n'est pas guidée par une fonction d'évaluation (fitness), c.à.d. la sélection est naturelle. Les individus se disputent donc les ressources de l'environnement : A titre d'exemple, l'espace de la mémoire, comme dans Tierra [Ray91a; Ray91b]) ; la nourriture, comme dans Gene Pool [Ven96, Ven98a, Ven98b, Ven05] et Darwin's Park [BL07a, BL07b, BL08a, BL08b, BL08c]), etc. Ainsi, un individu meurt quand il n'a plus d'énergie ou d'espace pour survivre. Les individus les plus adaptés peuvent cependant survivre et faire passer leurs gènes à leurs descendants.

L'intérêt d'un tel système est d'essayer de comprendre et d'analyser l'évolution d'un écosystème naturel. En outre, L'avantage est que l'évolution est laissée libre et peut aboutir à des résultats intéressants.

- **Tierra.** L'écosystème de Thomas Ray développés dans le cadre du projet Tierra [Ray91a; Ray91b], permet la création d'organismes numériques. Ces organismes numériques sont des séquences d'instructions machines, qui sont mis en compétition pour l'espace mémoire et le processeur de l'ordinateur, ressources dont ils se servent pour s'auto-répliquer. Les instructions que constitue chaque organisme sont vues comme des molécules qui ont la capacité de se déplacer dans l'espace (la mémoire), à se transformer ou à s'associer à d'autres « molécules », on parle alors de chimie artificielle (section 4.3.1) [Ren02]. La réplication se fait en utilisant une procédure empruntée à la biologie moléculaire, le "chaînage par la forme". La protéine A ne "sait" pas où se trouve la protéine B, son partenaire dans la reproduction, mais présente une forme dont la surface est complémentaire de celle de la protéine B. Quand la diffusion rapproche les deux protéines, leurs formes s'aimantent et les protéines entrent en interaction. De façon similaire, pour qu'un organisme numérique puisse se reproduire dans Tierra, il doit être capable de trouver son complément le plus proche dans la mémoire de l'ordinateur. Un organisme codé 1001, par exemple, peut se répliquer quand il trouve 0110. [Hay01]. Dans tous les lancements de Tierra, Ray utilise toujours une séquence de 80 instructions appelée ancêtre. Des mutations sont introduites ensuite dans les descendants à un rythme régulier. Quand la mémoire est saturée, les organismes les plus âgés ou ceux qui commettent le plus d'erreurs lors de l'exécution des instructions sont tués. Le résultat le plus intéressant selon Gell Mann [Gel95] est l'apparition, après une longue période d'évolution, d'une version plus comprimée du programme initiale (l'ancêtre), se composant seulement de 36 instructions au lieu de 80 et contenant un algorithme plus complexe.
- **Gene Pool.** Gene Pool [Ven96, Ven98a, Ven98b, Ven05] est un écosystème aquatique bidimensionnel peuplé de créatures appelées *swimbots*. Ces créatures ont une morphologie et un comportement. La morphologie des créatures est composée de rectangles aux angles arrondis qui sont reliés par des articulations. Chaque créature possède un état qui est décrit par un diagramme d'état comportant quatre possibilités et deux actions: recherche d'un partenaire ayant envie de se reproduire, se diriger vers le partenaire, rechercher de la nourriture, se diriger vers la nourriture, se reproduire et manger. La recherche de partenaires se fait à l'aide de certains critères par exemple : la taille (petit, grand), la couleur (similaire, opposée), la réactivité (actif, mou). Les créatures sont sexuées et la sélection est naturelle. Ainsi, les individus qui ne sont pas adaptés pour se nourrir ou se reproduire finissent par disparaître par manque d'énergie. L'utilisateur peut alors observer en temps réel l'évolution des individus grâce à une interface présentant une grande panoplie

d'options : la possibilité de déplacer les créatures, de les reproduire, de les supprimer, de modifier leurs énergies,... etc. Il est aussi possible de modifier leur environnement ou de leur ajouter de la nourriture et de constater par la suite les effets.

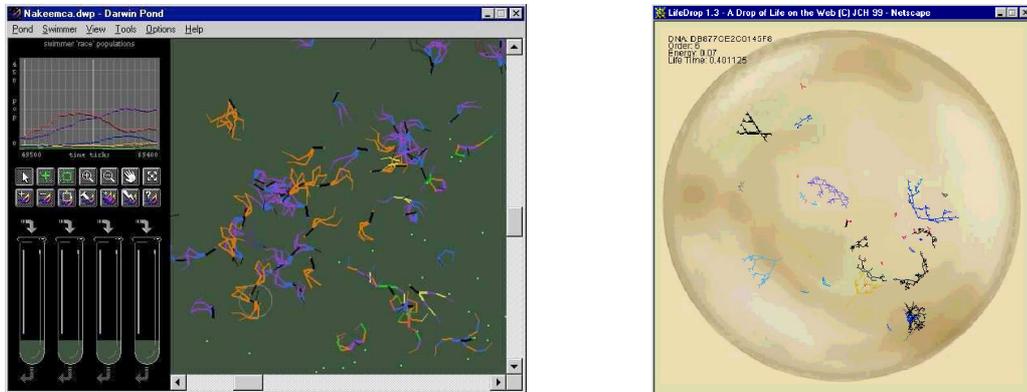


Figure 2.23- (a) Darwin Pond, version antérieure de GenePool. (b) Life Drop.

- **Life Drop.** Life Drop [MLH 02] est un écosystème simulant une goutte d'eau artificielle, habitée par de petites créatures. Ces dernières sont inspirées des Biomorphs (section 3.3.2). L'environnement simule les forces physiques et prend en compte des paramètres comme le Ph et la fluidité de l'eau. Chaque créature est un agent autonome décrit par un modèle en couches inspiré de l'architecture de Brooks [Bro91]. Le modèle implémenté comporte cinq couches : Génotype : composé de 32 bits codant toutes les caractéristiques de la créature; Métabolisme : contrôle les cycles essentiels comme le cycle de vie, de croissance, etc. ; Dynamiques : contrôle "l'incarnation" de la créature simulant l'impact "des conditions physiques" comme la fluidité de l'eau ; Comportement réactif : contrôle des comportements réactifs de base tels que l'évitement d'obstacle, la fuite d'un danger, etc. ; Comportement cognitif : contrôle des comportements tels que le choix d'un partenaire. En plus des paramètres génétiques, chaque créature obtient un ensemble additionnel de paramètres après son développement. Ceux-ci incluent -entre autres- ses états comme "vivante", "affamée", "fécondée", "stressée"...etc. Le modèle physique qui permet aux créatures de se déplacer dans l'environnement s'inspire de celui de [Rey99]. Les créatures se reproduisent entre elles suivant leur affinité génétique et leur niveau de stress.
- **Darwin's Park.** Darwin's Park [BL07a, BL07b, BL08a, BL08b, BL08c] est une plateforme de simulation d'écosystèmes de plantes virtuelles. Chaque plante possède une morphologie et une physiologie. Les morphologies aériennes et souterraines sont chacune définies par un L-système. Quant à la physiologie, elle est définie par un modèle transport-résistance [Tho98] qui décrit les processus d'assimilation et d'allocation de ressources par un système d'équations différentielles.



Figure 2.24: Darwin's Park. (a) jardin virtuels. (b) Compétition sur le terrain virtuel.

- **Autres.** Il existe d'autres travaux sur les écosystèmes artificiels, parmi eux nous citons PolyWorld [Yae94, YS08] habité par une variété d'organismes qui s'appuient sur des réseaux de neurones. Dans le projet Cyber Rodent [DU05, EUD05], une population de robots doit maintenir un niveau d'énergie suffisant pour survivre dans un environnement où existent des batteries de recharge. Pour que son génome se propage, un robot doit trouver une source d'énergie mais aussi des partenaires pour la reproduction et effectuer un échange de matériel génétique.

2.4.5 Ontogenèse.

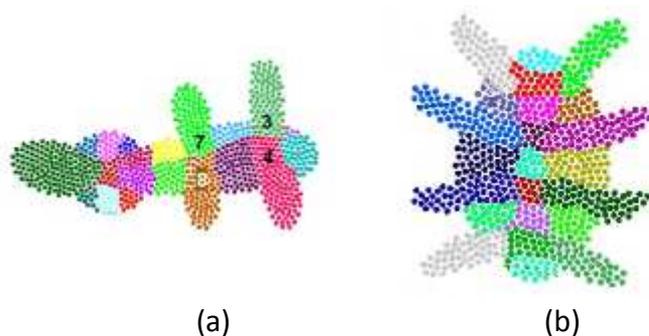


Figure 2.25-Organisme développés grâce au modèle de développement de Doursat [Dou09]. (a) la salamandre.(b) organisme à 8 pattes

Au lieu de générer directement la morphologie et/ou le contrôleur d'une créature artificielle, l'ontogenèse est une voie de recherche qui permet de s'inspirer une fois de plus du vivant afin de *développer* des créatures à partir d'une cellule ou d'un petit nombre de cellules initiales. Toute cette thématique de recherches ainsi que les principaux travaux reliés feront l'objet du chapitre suivant.

2.6 Conclusion.

La vie artificielle s'inspire du vivant pour élaborer des techniques similaires aux mécanismes existant dans la nature comme les mécanismes d'adaptation, d'évolution et de morphogenèse. Ces techniques sont utilisées, entre autres, pour simuler le vivant. Synthétiser des créatures artificielles en vue de les reproduire sous forme de robots ou de peupler des mondes virtuels. Nous avons distingué dans ce chapitre trois approches pour la synthèse de créatures artificielles.

L'épigenèse artificielle selon notre vision, consiste à faire émerger le comportement (fuite poursuite, anticipation stratégie d'équipe) d'une créature. Pour ce faire, on utilise un algorithme génétique pour faire évoluer le contrôleur de la créature, basé généralement sur des méthodes d'apprentissage comme les réseaux de neurones ou les systèmes de classeurs.

La phylogenèse consiste à générer des créatures, en faisant l'évolution conjointe de leur comportement et de leur morphologie. Nous avons vu que, dans ce type d'approche, tous les contrôleurs, hormis ceux de Nicolas Lassabe sont basés sur des réseaux de neurones, pour la morphologie à part le travail de Hornby où il propose l'utilisation des L-systèmes. La plupart des travaux utilisent les graphes de développement (*graphical*) avec des constituants plus ou moins différents (tubes, bâtonnets, parallélépipèdes rectangles). Dans ce type d'approche, on distingue également, les écosystèmes artificiels. A l'image des écosystèmes naturels, les écosystèmes artificiels permettent de faire co-évoluer un ensemble de créatures (plantes virtuelles comprise) dans un même environnement et qui peuvent, dans les meilleurs des cas, interagir en eux. L'avantage ici est que la sélection est naturelle et peut aboutir à des résultats intéressants.

Enfin, au lieu de générer directement la morphologie et/ou le contrôleur d'une créature artificielle, l'ontogenèse s'inspire, quant à elle, beaucoup plus du vivant pour développer des créatures à partir d'un petit ensemble de cellules initiales. Cette approche constitue, à notre avis, la voie de recherche la plus prometteuse car elle s'inspire de la nature et l'un des buts de la réalisation des créatures artificielles est d'essayer de reproduire le vivant. L'ontogenèse rapprochera donc les créatures artificielles de celles déjà existantes dans la nature. Elle est aussi la voie si on veut avoir des créatures artificielles, voir des robots, avec la capacité de s'auto-réparer. En outre, et hormis le domaine de la génération de créatures artificielles, on assiste de nos jours, à une tendance et à un renouveau de différents systèmes informatiques pour incorporer des inspirations biologiques. A titre d'exemple, nous citons les systèmes multi-agents. Nous nous inscrivons pleinement dans cette approche c'est pourquoi nous y reviendrons en détails dans le chapitre qui suit.

Chapitre 3

L'ontogenèse artificielle.

*" Les vérités sont choses à faire et non à découvrir, ce sont des constructions et non des trésors. ",
Paul Valéry*

Le développement est comme la plupart des phénomènes biologiques, un processus très compliqué. Afin de comprendre puis d'établir, par la suite, un modèle de ce processus qui peut être simulé sur un ordinateur (un modèle de développement artificiel), dans ce chapitre nous expliquerons le développement naturel, son rôle fondamental dans l'orientation de l'évolution ainsi que ses principaux mécanismes. Ensuite, avant de passer en revue les principaux modèles de développement artificiels, nous définirons l'ontogenèse artificielle, dont la compréhension nécessite une bonne connaissance du développement. Enfin, nous présenterons ses avantages ainsi que ses inconvénients comme une sous discipline du calcul évolutionniste et comme une approche permettant de synthétiser des créatures artificielles.

3.1 Développement.

Chaque organisme vivant, que nous observons aujourd'hui, est le produit des deux processus biologiques les plus remarquables : le développement et l'évolution. Chaque organisme multicellulaire adulte était un jour une seule cellule œuf. Une séquence complexe d'interactions génétiques, cellulaires et environnementales a eu comme résultat le développement de cette cellule en un organisme adulte. Les ancêtres de cet organisme étaient de simples amibes. A travers des millions d'années d'évolution, les descendants de ces amibes ont diversifié pour produire une large gamme de formes phénotypiques. Une question consistante dans la biologie évolutionniste réside dans la manière dont les organismes vivants ont pris leurs formes courantes ? La réponse conventionnelle, fournie par le paradigme Néo-Darwinien (section 2.3), souligne le rôle de la sélection naturelle et de la mutation dans l'orientation de l'évolution.

Cependant, pendant les deux dernières décennies, les chercheurs dans le domaine de la biologie de développement évolutionniste ont argué le fait que la sélection naturelle est incomplète comme explication de l'histoire de l'évolution [lew06]. Ils suggèrent que les changements phénotypiques graduels sont le résultat des processus micro-évolutionnistes. Une technique telle que la mutation est également, une explication insuffisante de l'évolution des différences significatives entre les espèces humaines et de leurs structures corporelles. En effet, ils prétendent que la compréhension de l'histoire de l'évolution requiert une appréciation du rôle joué par le développement.

3.1.1 Rôle du développement dans l'orientation de l'évolution.

Le développement consiste en une séquence d'événements cellulaires (une ontogenèse) guidée par des interactions génétiques, épi-génétiques et environnementales. Le système de développement génétique qui gouverne ce processus est sujet à des mutations qui peuvent modifier l'ontogenèse, ayant pour résultat la production de nouveaux phénotypes. L'évolution se produit quand ces nouvelles variantes sont favorisées par la sélection naturelle et survivent pour passer à travers leurs gènes aux futures générations [Gea06].

En effet, Le développement est important pour diriger l'évolution parce qu'il sert de médiateur entre le niveau génotypique, auquel la variation héréditaire se produit, et le niveau phénotypique, sur lequel la sélection naturelle agit. Le schéma de la figure 3.1 montre la place du développement dans l'orientation de l'évolution.

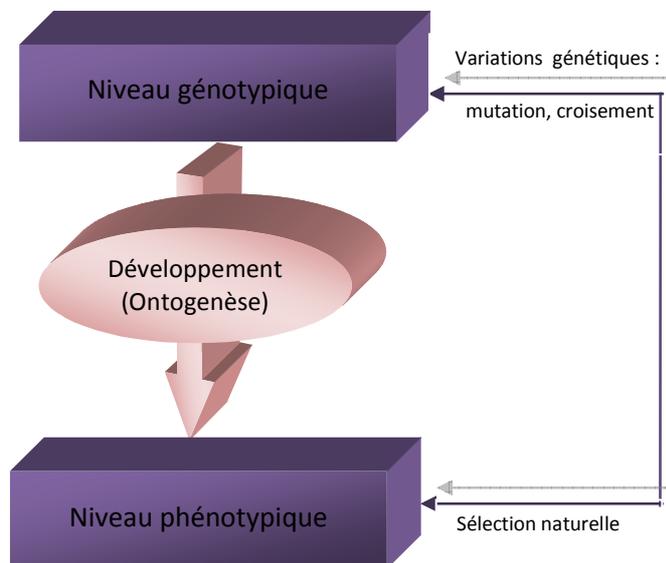


Figure 3.1 : Le développement (l'ontogenèse) et sa place dans l'orientation de l'évolution.

Du point de vue du traitement automatique de l'information, le développement est fondamentalement un processus computationnel [Gea06]: une forme d'information, **le génome de l'organisme**, est transformé en une autre, **l'organisme développé lui-même**. Le

développement est également, comme la plupart des phénomènes biologiques, un processus très compliqué. Afin d'établir ou comprendre un modèle de ce processus qui peut être simulé sur un ordinateur, nous présentons dans la section qui suit quelques pré-requis biologiques.

3.1.2 Principaux mécanismes.

Le développement d'un organisme à partir d'une cellule souche implique trois mécanismes: la croissance, la différenciation et la morphogenèse [Lwe06]. Ces processus ne sont pas isolés ou séquentiels, mais se produisent en parallèle durant le développement. Présentons maintenant ces trois mécanismes.

La croissance.

Pendant le développement, la croissance d'un organisme se produit par l'intermédiaire de la division cellulaire, et l'augmentation de la taille des cellules individuelles. Il y a deux modes primaires de division cellulaire. Au commencement du développement, des séries de divisions se produisent à l'issue desquelles le nombre de cellules augmente, mais il y a peu d'augmentation dans la masse globale des cellules. Ensuite, la prolifération des cellules peut se produire et durant cette prolifération la division cellulaire est accompagnée d'une augmentation de la masse des cellules.

Quand une cellule se divise, les deux cellules nouvellement créées ne sont pas nécessairement identiques. Le contenu de la cellule mère peut être divisé asymétriquement entre les deux cellules filles, ce qui aboutit à une différenciation de la taille et/ou du comportement de chacune. Trois modes possibles de division peuvent être distingués:

- *La prolifération* : les deux cellules filles ont un comportement identique à la cellule mère ($A \rightarrow A A$);
- *Le stem-cell* : l'une des cellules fille possède le même comportement que la cellule mère, la seconde se comportant différemment ($A \rightarrow A B$);
- *La diversification* : les deux cellules fille se comportent différemment de la cellule mère et différemment l'une de l'autre ($A \rightarrow B C$).

La différenciation.

Pendant que les cellules se divisent, elles subissent des changements dans leurs propriétés physiques et chimiques qui résultent, éventuellement, de leur différenciation en des types bien spécialisés. La cellule souche possède la capacité de produire des cellules qui différencieront au niveau de tous les types qui peuvent se trouver dans l'organisme. Pendant que le développement s'opère, les nouvelles cellules, créées par division, ont un pouvoir de différenciation de plus en plus restreint. À un certain point dans son développement, le destin de la cellule (*cell fate*) sera déterminé; après ce point il n'est plus possible que la cellule se différencie dans n'importe quel autre type.

La morphogenèse.

L'organisation des cellules en des structures fonctionnelles repose sur des facteurs chimiques et mécaniques. Des gradients chimiques sont établis de façon à ce qu'ils déterminent les axes à travers lesquels la structure corporelle de l'organisme est organisée. L'environnement embryonnaire peut être responsable de ces gradients chimiques. Ils peuvent, également, être établis par la signalisation inductive entre les cellules [Lwe06]. Ces gradients fournissent les informations aux cellules qui leur permettent de se différencier dans le type correspondant à leur location physique.

Les embryons subissent, également, une variété de comportements mécaniques qui ont comme conséquence le changement de l'arrangement physique des cellules, y compris la migration de cellules.

3.2 Ontogenèse artificielle.

En nature, l'ontogenèse¹ (embryogenèse) est définie par les interactions entre les gènes, leurs effets phénotypiques et l'environnement dans lequel l'embryon se développe [BEN99]. Elle est définie également, comme la trajectoire des événements du développement qui transforme un œuf en un embryon [Gear06].



Figure 3.2-Photos d'un Embryon d'humain de 6-7 semaines².

¹L'embryogénèse englobe l'ensemble des processus qui définissent la formation d'un organisme pluricellulaire, végétal ou animal, de la cellule œuf, issue de la rencontre des gamètes parentaux, à un être vivant autonome.

² Source : <http://www.intellego.fr/soutien-scolaire--/aide-scolaire-svt/extraordinaires-photos-d-embryon-et-de-foetus-humain-/16068>

3.2.1 Définition et objectif.

L'ontogenèse artificielle a été définie par Stanley et Miikkulainen [SM03], comme une sous discipline du calcul évolutionniste dans laquelle les phénotypes subissent une phase de développement. La première utilisation du terme embryogenèse artificielle dans les systèmes évolutionnistes trouve son origine dans les travaux d'Hugo de Garis [de 92] (section 3.3.4) qui a développé des "embryons artificiels" en tant que formes bidimensionnelles de cellules artificielles en utilisant le paradigme d'automate cellulaire. Par la suite, les chercheurs ont employé plusieurs noms pour les systèmes évolutionnistes artificiels qui utilisent une phase développementale, y compris "l'encodage cellulaire" [Gru94], "la morphogenèse" [Jak95], "l'embryogenèse computationnelle" [BK99], "l'ontogenèse artificielle" [BP01], "le développement computationnel" [KB03b] et "le développement artificiel" [CD07b].

Bentley et Kumar [BK99] ont défini l'embryogenèse comme le processus de croissance qui définit comment le génotype peut être tracé en un phénotype et ont précisé que le terme correct est embryogénie³ par opposition à l'embryologie, car l'embryogénie est le processus embryologique du développement lui-même, alors que l'embryologie est la science ou l'étude du processus du développement. Dans ce manuscrit, nous adoptons indifféremment les deux termes embryogenèse ou ontogenèse artificielle pour nous référer à la classe entière de tels systèmes.

Les mêmes chercheurs ont défini le domaine du développement computationnel, comme l'étude des modèles artificiels du développement cellulaire de l'embryon, dans le but de comprendre comment des structures et des formes complexes peuvent se développer à partir d'un petit groupe de cellules souches [KB03a]. Et la manière la plus naturelle d'étudier le développement est par la simulation de quelques modèles abstraits des dynamiques du processus de développement cellulaire, souvent dans le contexte d'un processus évolutionniste [STK05].

Le but de l'embryogenèse artificielle est de faire évoluer des systèmes qui rivalisent avec le niveau de complexité vu dans les organismes vivants. Pour cela, elle constitue l'une des voies les plus prometteuses dans la vie artificielle car elle s'inspire directement du vivant. Dans le domaine de la génération de créatures artificielles en vue de peupler des mondes virtuels ou les construire sous forme de robots, elle représente une approche qui permet de synthétiser des créatures artificielles - ou tout au moins des organes de créatures - à partir d'une cellule ou d'un petit nombre de cellules œufs. Par opposition à l'approche phylogénétique qui utilise des blocs ou des bâtonnets déjà prêts comme organes des créatures ou sous-parties de leurs corps ayant des fonctions spécifiques.

³ Le terme embryogénie regroupe l'ensemble des transformations qui interviennent au sein d'embryons animaux ou végétaux

3.2.2 Avantages et Inconvénients.

Comme sous discipline du calcul évolutionniste et comme une approche permettant de générer des créatures artificielles, l'ontogenèse artificielle possède des avantages et des inconvénients :

Avantages.

L'ontogenèse artificielle véhicule les avantages suivants :

- *réduction de l'espace de recherche.* L'ontogenèse permet de représenter les génotypes d'une manière très compacte. Cette compression (souvent récursive, hiérarchique ou multifonctionnelle) a comme conséquence de représenter les génotypes avec moins de paramètres que les phénotypes correspondants, causant une réduction de la dimensionnalité de l'espace de recherche, et fournissant, par conséquent, un espace de recherche plus réduit pour les algorithmes évolutionnistes [BEN99].
- *meilleure énumération de l'espace de recherche.* L'ontogenèse artificielle permet à deux espaces très différemment organisés de coexister, c'est à dire un espace de recherche conçu pour être facilement recherché (espace génotypique) peut permettre aux algorithmes évolutionnistes de localiser les solutions correspondantes dans un espace de solution très dur à rechercher (espace phénotypique) [BEN99].
- *solutions plus complexes.* En utilisant des "instructions de croissance" à l'intérieur des génotypes pour définir comment des phénotypes devraient être générés, un génotype peut définir des phénotypes très complexes [BEN99].
- *répétition.* Des systèmes d'ontogenèse artificielle correctement conçus où les mêmes gènes sont utilisés plusieurs fois dans le processus de développement des phénotypes, peuvent améliorer la capacité d'évolution de produire des solutions avec des "structures répétées" telles que la symétrie, la segmentation, et les sous-programmes. Une telle "réutilisation de gènes" permet aussi une représentation très compacte de phénotypes très complexes [SM06].
- *Adaptation.* Il est possible de développer des phénotypes à partir des génotypes de manière adaptative, permettant à des contraintes d'être satisfaites, à l'amélioration de conditions variables ou à la correction des défauts de fonctionnements dans les conceptions [BEN99].
- *auto-organisation et émergence.* Les systèmes d'embryogenèse artificielle peuvent tirer profit de possibilités d'auto-organisation par exemple la forme finale d'un phénotype ou le type d'un élément phénotypique, une cellule par exemple, peuvent être des propriétés émergentes de la dynamique du système ou de l'interaction intercellulaire et non pas pré-spécifiés.

- *distribution*. Chaque élément phénotypique (par exemple une cellule) embarque généralement la même représentation génotypique (même génome). De ce point de vue, un système d'embryogenèse artificielle peut être implémenté sur des architectures parallèles, ce qui correspond parfaitement à la nature parallèle du processus de développement biologique vu dans la nature.
- *Modularité et autoréparation*. Pour la génération de créatures artificielles, l'ontogenèse est certainement la voie à suivre si on veut obtenir des créatures, voire des robots, avec la capacité de s'auto-réparer. Par ailleurs, elle est très utile dans la confection de robots qui tendent à être de plus en plus modulaires.

Inconvénients.

L'embryogenèse artificielle souffre également de quelques inconvénients [BEN99] :

- *La difficulté de conception*. Tous les modèles d'ontogenèse artificielle exigent une conception très soignée et, jusqu'à présent, seuls les chercheurs qui sont capables d'exercer cet art difficile ont pu présenter des résultats réussis [BEN99].
- *La difficulté de l'évolution*. Si le soin n'est pas pris, L'embryogenèse artificielle peut introduire des problèmes pour les algorithmes évolutionnistes. L'explosion et l'absence de solutions sont possibles, ayant pour résultat le besoin d'opérateurs génétiques soigneusement conçus [BEN99].
- *Halting problem*. L'arrêt du processus développementale est l'un des problèmes récurrents qui doit également être déterminé soigneusement par le concepteur du système.

3.3 Modèles de développement artificiel.

Dans cette section nous passerons en revue les principaux modèles de développement artificiel.

3.3.1 Système de réaction-diffusion.

Alan Turing [Tur52] a été le premier à modéliser le développement biologique mais à un niveau chimique avec son modèle de *réaction-diffusion*. Bien que le modèle n'était pas censé englober tout le processus de développement, il a réussi à produire des formes très semblables à celles vues sur la peau des organismes naturels, tels que les structures colorées vues sur les plumes d'oiseaux ou les pelages d'animaux (figure 3.3).

En effet, Turing a montré comment un milieu, initialement homogène, pourrait développer des formes structurées résultant d'une instabilité de l'équilibre homogène, déclenchée par de petites perturbations aléatoires.

Dans son modèle, des substances appelées morphogènes ou des générateurs de forme, réagissent ensemble et se diffusent dans un milieu, qui pourrait être un tissu. Un ensemble d'équations différentielles qui décrivent comment les concentrations de ces substances changent à travers le temps ont été utilisées. L'une des substances est auto-catalytique et tend à synthétiser plusieurs copies d'elle-même.

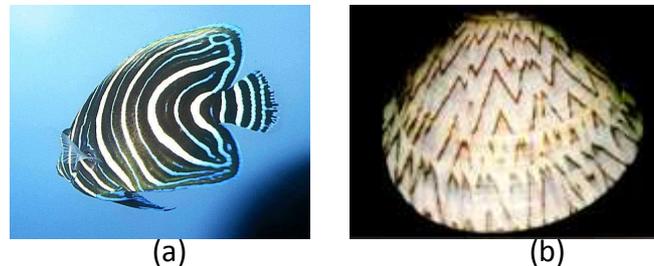


Figure 3.3- (a)Structure peu de poisson [KA95]. (b)Structure de coquille [Mei95].

En même temps une autre substance favorise la synthèse de la première, mais cette dernière empêche la synthèse de l'autre. Un élément clef du modèle est que les deux substances ont chacune un coefficient de diffusion différent et par conséquent l'une d'elles se diffuse beaucoup plus rapidement que l'autre. Le système peut être ajusté avec les paramètres appropriés de telle sorte qu'à un certain point, la perturbation la plus légère dans l'équilibre peut être amplifiée et être propagée par le milieu produisant des structures imprévisibles.

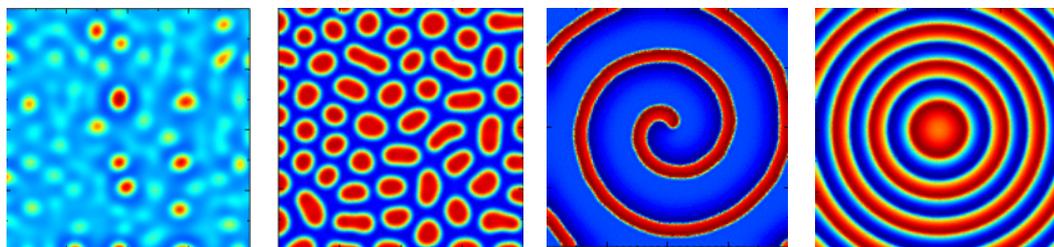


Figure 3.4- Exemple de structures développées par réaction diffusion⁴.

Les structures produites par le modèle sont très semblables aux modèles rencontrés dans la nature, donnant crédibilité au modèle.

3.3.2 Biomorphs.

Les biomorphs sont des créatures numériques bidimensionnelles. Elles ont été initialement créées par Richard Dawkins dans son fameux livre "l'horloger aveugle" [Daw86]. Il les appela "biomorphs" en reprenant le terme utilisé par Desmond Doris pour décrire les formes animales de ses peintures surréalistes. Dans sa version originelle, un biomorph représente le phénotype d'un génome qui se constitue d'un ensemble de neuf gènes dont chacun est un

⁴ Source : http://en.wikipedia.org/wiki/Reaction%E2%80%93diffusion_system

nombre entier qui peut prendre une valeur dans l'intervalle $[-9,9]$. Les huit premiers gènes codent la direction et la longueur d'un embranchement tracé dans un plan bidimensionnel, alors que le neuvième code la profondeur de la récursion.

Le développement des biomorphs est basé sur un algorithme récursif ; "l'horloger aveugle" dont le principe est très simple : à chaque itération un nouvel embranchement est généré. L'objectif de Dawkins était de générer des formes d'arbres en partant d'un tronc, mais à sa grande surprise l'utilisation de son algorithme avait démontré qu'il pouvait générer plusieurs catégories de formes d'inspiration biologiques (insectes, chauves-souris) ou non (tels que les avions, candélabres). La figure 3.5 illustre un panel de structures possibles.

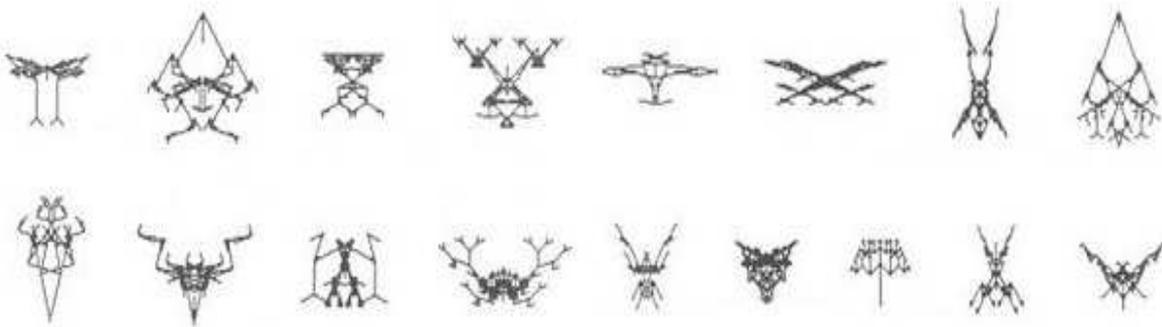


Figure 3.5: Exemple de Biomorphs.

L'auteur a proposé, ensuite, un algorithme évolutionniste "interactif" où l'œil de l'utilisateur joue le rôle de la sélection naturelle, en se basant sur deux règles simples : la micro-mutation et la sélection cumulative. Au commencement, l'exécution de l'algorithme présente à l'utilisateur un biomorph parent et ses dix-huit biomorphs fils résultant de l'addition ou la soustraction d'une unité (micro-mutation) sur chacun des neuf gènes dans le génome. Ensuite, l'utilisateur doit décider quelle forme il veut faire évoluer, une araignée, un pin... Après, à chaque étape de l'algorithme, il choisit le biomorph qui ressemble le mieux à la forme qu'il désire obtenir (choix cumulatif).

3.3.3 Réseaux booléens aléatoires.

Les réseaux booléens aléatoires (*Random boolean Network*) ont été proposés par Kauffman comme un modèle aléatoire de réseau de régulation génétique [Kau69, Kau71, Kau93]. Un réseau booléen aléatoire (RBA) se compose d'un ensemble de N nœuds et de K connexions. Chaque nœud représente un gène qui peut être actif ou non. Les connexions sont produites aléatoirement, mais une fois déterminées, elles demeurent fixes pendant toute la simulation. La dynamique du réseau est déterminée par sa topologie ainsi que la table de règles de transition produites aléatoirement au niveau de chaque nœud. Les états de chaque nœud dépendront de tous les états des nœuds auxquels il est relié, et ils sont mis à jour de façon synchrone.

Les RBAs peuvent être vus comme une généralisation des automates cellulaires booléens mais à la différence de ceux-ci, l'état d'un nœud ne dépend pas nécessairement des états des nœuds voisins [Ger04]. La figure 3.6 (a) représente un exemple d'un RBA avec $N=K=3$.

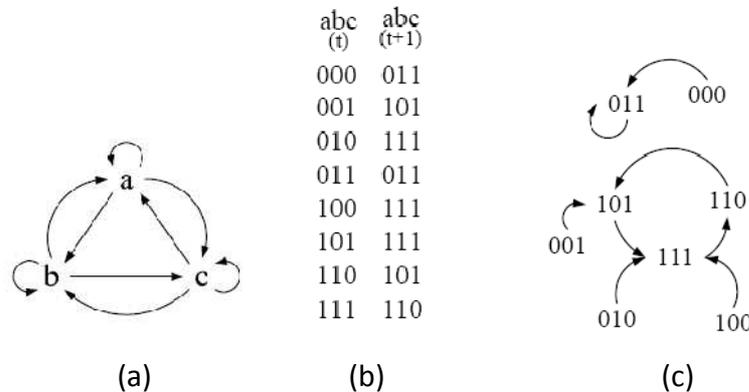


Figure 3.6: Exemple d'un réseau booléen aléatoire. a) configuration du réseau b) la table de règles de transition ; c) le diagramme de l'espace d'états.

Il est possible de distinguer trois comportements ou structure qui émergent des dynamiques du réseau à savoir : régulier, chaotique et critique. Afin d'identifier chacun de ces comportements, des graphes sont produits. L'axe des ordonnées du graphe représente le temps tan disque l'axe des abscisses représente les nœuds dans l'état "1" esquissés par un petit carré noire. La figure 3.7 montre le graphe qui correspond à chaque type d' RBA. Des les premières études, il a été démontré que le paramètre K a une grande influence sur le type d'RBA produit. En effet, les réseaux avec $K \leq 2$ correspondent au type régulier, alors que le type chaotique peut se trouver dans les réseaux avec $K \geq 3$.

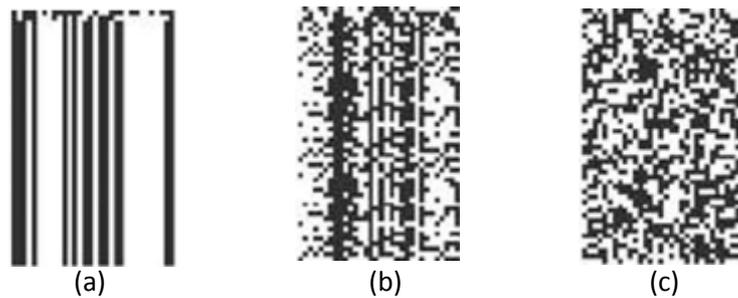


Figure 3.7 : Exemple de trios types d' RBA . $N = 32$. Carré noire = état "1", carré blanc = état '0'. États initiaux au dessus et le temps se coule de haut en bas. (a) régulier ($K = 1$) ; (b) critique ($K = 2$) ; (c) Chaotique ($K = 5$).

Les chercheurs considèrent le comportement critique comme le plus intéressant car le type régulier est trop statique pour faire émerger des observations utiles applicables aux systèmes dynamiques, et le type chaotique est trop aléatoire pour permettre d'étudier n'importe quelle sorte de propriété reproductible. Dans [Heu98], le type critique a été

nommé "le bord du chaos" qui dans les RBAs signifie que des états qui changent dynamiquement sont en phase de transition entre régulier et chaotique.

Kauffman a également découvert que ces phases transitoires ont été liées à des aspects divers de la stabilité d'un réseau. Il a entrepris des expériences où il a manuellement perturbé ou endommagé des nœuds dans l'RBA afin d'étudier les effets de ces altérations. Il a trouvé que dans le régime régulier, l'endommagement d'un nœud n'affecte généralement pas les autres nœuds. Dans le régime chaotique un petit changement peut avoir de larges conséquences dans la dynamique du réseau. Finalement dans le "le bord du chaos" des changements peuvent disséminer, mais n'affectent pas nécessairement le réseau en entier.

3.3.4 Embryologie artificielle.

Hugo de Garis est l'un des premiers chercheurs ayant travaillé sur l'embryogenèse artificielle et même le premier à avoir utilisé ce terme [de 92, de 99]. Il a développé des "embryons" artificiels en tant que formes bidimensionnelles, convexes ou concaves, constituées par un ensemble de cellules. La reproduction de celles-ci est contrôlée par un automate cellulaire qu'il a fait évoluer par un algorithme génétique. Pour cela, ce chercheur a employé ce qu'il a nommé les "chromosomes différenciables", pour coder l'ensemble de règles de reproduction (condition C /action A) de l'automate cellulaire.

Dans ce modèle, les cellules peuvent se reproduire, si et seulement si, il y a au moins une cellule adjacente vide. En effet, en supposant que seulement les cellules de bord peuvent se reproduire et qu'aucune cellule isolée n'est produite, les cellules dans une grille bidimensionnelle peuvent être dans l'un des quatorze états possibles de la figure 3.8.

Si un état est codé par un bit, alors seulement quatorze bits sont nécessaires pour coder tous les états, avec une position prédéfinie pour chaque état dans la chaîne binaire. S'il y a un "1" dans le bit correspondant à un état, alors toutes les cellules de bord dans cet état peuvent se diviser pour produire une cellule fille, tandis qu'un "0" signifie que les cellules dans cet état ne sont pas autorisées à se reproduire.

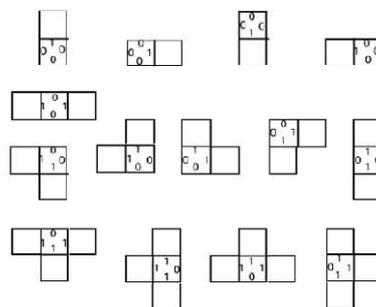


Figure 3.8-Les 14 états de voisinage pour des cellules 2D.

Quand on permet aux cellules de bord de se reproduire, la direction suivant laquelle la cellule fille doit être placée doit être déterminée. Pour les états où il y a uniquement une seule cellule adjacente vide, ce n'est pas nécessaire de coder la direction de reproduction. Cependant, quand il y a deux ou trois cellules adjacentes vides, un choix doit être fait. Pour

des états avec deux cellules adjacentes vides, seulement un bit est nécessaire pour déterminer la direction de reproduction, avec une direction prédéfinie pour chaque valeur de bit et chaque état.

Pour des états avec trois cellules vides, deux bits sont suffisants pour coder la direction de la reproduction, avec une direction de reproduction prédéfinie pour chaque combinaison de deux bits. Par conséquent, puisqu'il y a 6 états avec 2 cellules adjacentes vides, et 4 états avec 3 cellules adjacentes vides (voir la figure 3.8) il est nécessaire d'employer $6 \times 1 + 4 \times 2 = 14$ bits pour indiquer la direction de reproduction pour tous les états.

Ainsi, la combinaison de deux chaînes de 14 bits peut déterminer pour chaque cycle de reproduction quelles cellules sont autorisées à se diviser (première chaîne) et dans quelle position elles placeront leurs cellules filles (deuxième chaîne). Afin de faire évoluer les règles de reproduction pour plus d'un cycle avec une règle de reproduction probablement différente pour chaque cycle, il est nécessaire d'introduire dans le chromosome différenciable le nombre d'itérations désirées (nombre d'instances des deux chaînes de 14-bit). De cette manière, le nombre d'itérations nécessaires pour créer une forme prédéfinie peut également évoluer. La figure 3.6 représente la structure d'un tel chromosome.

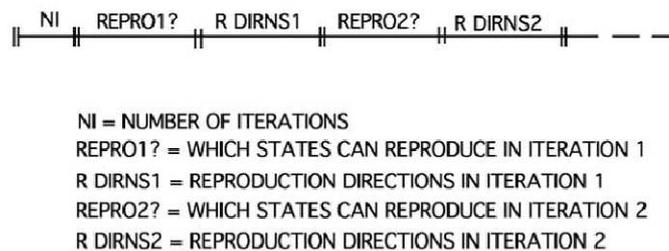


Figure 3.9-Format d'un chromosome simplifié.

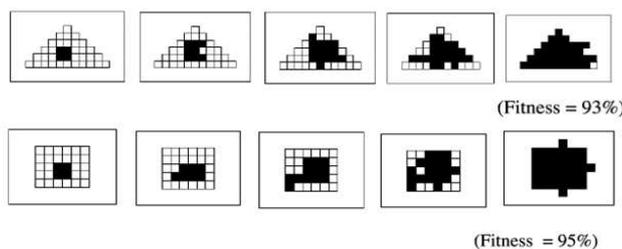


Figure 3.10-Forme triangulaire et rectangulaire évoluée. Les cellules vides représentent la forme désirée.

En utilisant les chromosomes ainsi définis, de Garis a pu développer plusieurs formes convexes.

La fitness utilisée est définie comme suit : $\frac{ins*1/2out}{des}$ où *ins* est le nombre de cellules remplies qui sont à l'intérieur de la forme désirée, *out* est le nombre de cellules remplies à l'extérieur de la forme désirée, et *des* est le nombre total de toutes les cellules à l'intérieur de la forme désirée. Ainsi, une valeur de fitness de 1 représente une parfaite correspondance entre la forme désirée (forme cible) et la forme développée.

Après, de Garis a conduit plusieurs séries d'expérimentations pour essayer de développer des formes concaves. Cela n'a cependant pas pu être produit. En effet le chercheur a conclu que l'évolution d'un embryon artificiel implique un déploiement de formes séquentielles et synchronisées. Par exemple, après que le corps principal soit développé, suivi de la tête et les membres, d'autres formes plus détaillées, comme celles correspondant aux doigts et aux orteils, peuvent apparaître.

Pour valider son idée, de Garis a essayé la génération d'une forme en L (L-forme) qui est une forme concave. La figure 3.11 montre la forme cible et sa décomposition dans deux régions. L'idée fondamentale est d'employer deux gènes pour produire la forme en L, où le premier gène produirait la région A, puis il serait désactivé pour laisser les cellules de la région R exprimer le deuxième gène afin de produire la région B.

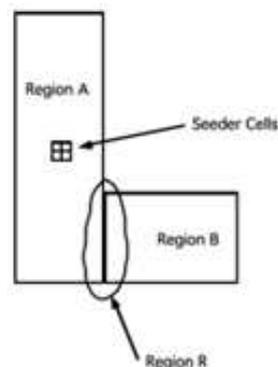


Figure 3.11-L-forme désirée et ses régions composantes.

Pour ce faire, il est nécessaire que les cellules puissent déterminer leurs positions dans la grille. Pour cela l'auteur s'est proposé d'employer un gradient de concentration d'un produit chimique pour fournir aux cellules l'information positionnelle. Chaque cellule possède une quantité de ce produit chimique, qui sera transmise à sa cellule fille. Ainsi un gradient de concentration est formé avec un pic à la position initiale des cellules.

Afin de déterminer la direction de plus haute concentration du produit chimique pour chaque cellule, 8 bits de stockage sont utilisés. À chaque pas de temps, la concentration du produit chimique au niveau de chaque cellule est mesurée. De ce fait, on obtient huit valeurs de concentration pour chaque cellule (parce que chaque cellule possède 8 cellules voisines). Le bit correspondant à la direction avec la plus haute concentration est alors mis à 1, et tous les autres bits à 0.

De cette manière, une cellule peut déterminer son gradient de concentration et obtenir une évaluation de sa position relative. Par exemple, les cellules dans le sud-est de la région A auraient leurs gradients le plus élevés vers la direction du nord-ouest et auraient, donc, leur bit du nord-ouest égale à 1. D'autre part, il est nécessaire que les cellules sachent lequel des deux gènes A et B est actif. La solution mise en œuvre, dans ce cas, consiste à exploiter un compteur de génération interne.

Sous ces prétentions, le chromosome de deux gènes montré dans la figure 3.12 a été proposé. Le *champ X* code le *compteur d'itération*. Quand le compteur d'itération atteint $X + Y$, le deuxième gène est désactivé. Le *champ NEWS DIRNS* possède 8 bits et détermine les cellules qui peuvent se reproduire après X itérations. Néanmoins, les cellules qui ont leur bit de direction égal à 1 quand le même bit de direction dans ce champ est également à 1 peuvent se dupliquer. Par exemple, si le gradient le plus élevé d'une cellule est *NW*, et le bit correspondant au *NW* dans le champ des *NEWS DIRNS* est à 1, alors on permet à la cellule de se reproduire après X itérations. Enfin, les *champs REPRO?/DIRN PAIRS* sont similaires aux *REPRO? Et R DIRNS* présentés dans la figure 3.9. La figure 3.13 montre les résultats obtenus en utilisant cette approche et en essayant de produire la forme en *L*.

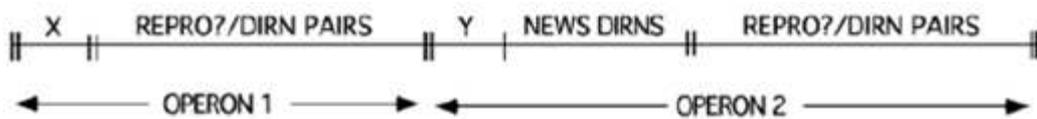


Figure 3.12-le chromosome de deux gènes utilisé pour la génération de la forme en *L*.



Figure 3.13-L-forme produite par un chromosome de deux gènes. (a) après expression du gène 1. (b) après expression du gène 2. (La forme cible est esquissée par les cellules vides).

Afin d'obtenir de meilleurs résultats, l'auteur a utilisé une technique qu'il a nommée "formation", qui consiste à diviser le processus évolutionniste en phases avec des buts intermédiaires. Dans ce cas, le deuxième gène est désactivé et l'évolution a été, initialement, conduite avec l'objectif de produire des chromosomes qui pourraient produire la région *A* de la forme en *L* (voir la figure 3.11). Les chromosomes résultants représentent la population initiale de l'AG, mais cette fois avec les deux gènes entièrement fonctionnels. Des résultats de ces expériences sont montrés à la figure 3.14.

Malgré l'utilisation de chromosomes qui ont été déjà conditionnés pour commencer par la région correcte, la technique de formation n'a pas produit de meilleurs résultats. Cet auteur a essayé également, de produire d'autres formes concaves, telle qu'un homme de neige, une tortue avec une combinaison de forme de cercle, mais il a encore encouru dans des limitations dans les valeurs maximales de fitness qu'il a pu obtenir.

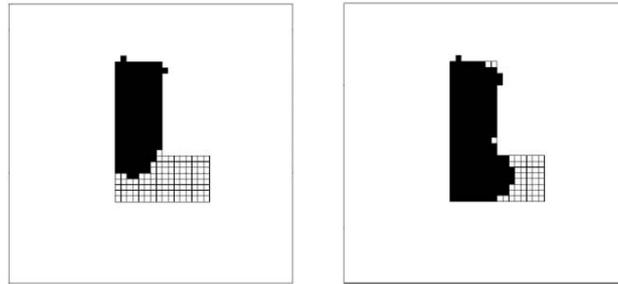


Figure 3.14: *L-forme produite par un chromosome de deux gènes qui a évolué en utilisant la technique de formation (les cellules vides représentent la forme cible).*

Bien que l'approche employée par de Garis ait prouvé le potentiel de l'application des algorithmes évolutionnistes dans la génération de formes (*Pattern formation*), ses résultats ont eu un succès limité. Cependant, il a été l'un des premiers chercheurs qui a employé le concept de l'activation séquentielle des gènes pour la production des structures multicellulaires.

3.3.5 Morphogenèse de réseaux de neurones.

Le développement de la structure des réseaux de neurones est l'un des challenges majeurs pour l'embryogenèse artificielle. Les premiers travaux dans ce domaine reviennent à Kitano [Kit90], qui a utilisé les algorithmes génétiques pour faire évoluer les règles de reproduction des L-systèmes afin de générer la morphologie du réseau, au détriment d'un encodage direct de sa topologie. Par la suite, beaucoup de scientifiques ont développé des modèles similaires, en explorant le potentiel des L-systèmes. Parmi eux, nous citons les travaux de Boers [BKHS93].

Par la suite, d'autres approches ont été proposées. Grueau [Gru93] a développé un modèle nommé encodage cellulaire basé sur les transformations d'un arbre [Gru93]. L'encodage cellulaire permet de contrôler la division des cellules qui interviennent dans le réseau de neurones.

Kitano [Kit95] a proposé un autre modèle de développement basé sur la simulation du métabolisme. L'idée de base était de coder les réactions métaboliques dans le génome de la cellule au lieu de coder les règles de réécritures du L-système. La production de certains produits chimiques par les règles réactions conduit à la synthèse de la molécule d'ADN. Quand la concentration de cette molécule atteint un certain seuil, les cellules peuvent se diviser et produire de nouvelles cellules. Les résultats de ce modèle ont montré qu'il était capable de faire évoluer une large gamme de réseaux neuronaux, et exhiber un phénomène analogue à la différenciation cellulaire.

3.3.6 Combinaison de mécanismes multiples pour la morphogénèse.

Fleischer et Barr ont utilisé une approche qui combine des facteurs chimiques, électriques, forces mécaniques et de mécanismes de contrôle de la lignée cellulaire (*cell lineage*) pour présenter une plateforme de simulation de l'étude de la formation de structures multicellulaires [FB94, Fle96].

Les cellules sont représentées, dans le modèle, comme des entités discrètes avec une forme (cercle ou sphère) et un vecteur d'état représentant la quantité de protéines à l'intérieur de la cellule. Les cellules peuvent exhiber plusieurs comportements qui incluent : la migration cellulaire, la capacité de se développer dans la taille, la libération ou l'absorption des produits chimiques environnementaux, le changement de la quantité de protéines particulières dans la membrane, la différenciation, la division cellulaire et l'apoptose.

Toutes les cellules simulées interagissent l'une avec l'autre ainsi qu'avec l'environnement. Celui-ci, intègre des barrières physiques ainsi qu'un certain nombre de produits chimiques qui peuvent se diffuser. Pour simuler le fonctionnement interne d'une cellule, un système dynamique est employé simulant l'expression de gènes, la régulation génétique, le métabolisme cellulaire, la cinétique chimique et le transport cellulaire interne, etc.

Le modèle du génome artificiel utilisé est constitué de l'ensemble des équations d'état de la cellule. Ces équations sont des équations différentielles qui déterminent comment l'état de la cellule -sa constitution en protéines- change en fonction de son environnement local ainsi que de son état courant. Dans les cellules biologiques chaque gène est responsable de la synthèse d'une protéine spécifique. Dans ce modèle, un gène c'est à dire une équation différentielle conditionnelle détermine le changement d'une variable liée à la quantité d'une protéine.

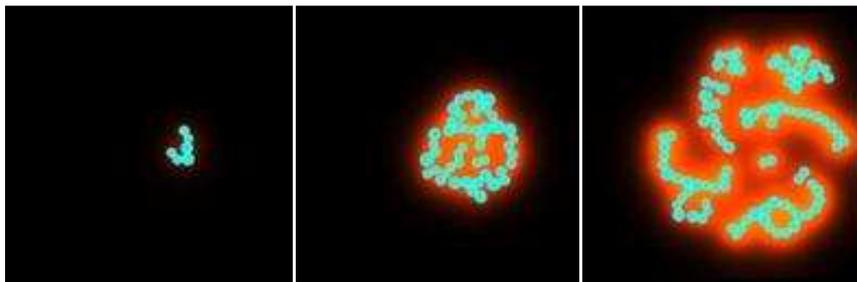


Figure 3.15-formation d'un organisme multicellulaire (cellules en bleu et molécules en rouge).

Les simulations commencent par une cellule (ou un petit groupe de cellules) qui se divisent et se différencient pour produire des organismes multicellulaires. Pour chaque simulation, un ensemble d'équations d'état et de conditions initiales différent peut être spécifié. Ceci fournit à l'utilisateur un substrat dans lequel il peut commencer à poser des questions sur la façon dont les changements aux équations régissant la dynamique des cellules affectent la morphogénèse et le comportement de l'organisme entier. Les résultats de ce modèle

incluent: des organismes avec des branches, des réseaux de neurones, des cellules dans des modèles de réaction diffusion...etc.

Le but initial des auteurs était le développement de réseaux de neurones. Cependant par la suite, leur objectif a été décalé vers l'étude des mécanismes multiples impliqués dans la morphogenèse.

3.3.7 Expression différentielle de gènes.

Eggenberger est l'un des premiers chercheurs qui a utilisé la notion de sites régulateurs pour simuler la morphogenèse d'organismes en 3D, en se basant sur l'expression différentielle des gènes [Egg97]. Dans Son modèle appelé AES pour Système Evolutionniste Artificiel, ce chercheur a proposé un génome structuré qui contient deux types d'éléments: des gènes structurels qui codent pour des substances spécifiques et des unités régulatrices qui agissent en qualité d'inhibiteurs ou renforçateurs des gènes structurels (figure 3.16).

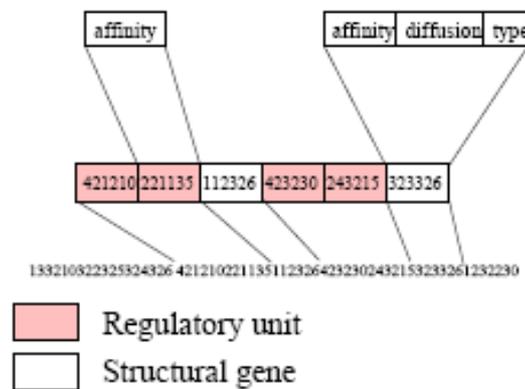


Figure 3.16:-Exemple d'un segment du génome d'Eggenberger.

Chaque gène, dans le génome, consiste en un nombre fixe d'entiers. Chaque cellule contient une liste de facteurs de transcriptions (FT) qui influencent son génome. Le FT est également représenté par une chaîne d'entiers qui peut être comparée avec les unités régulatrices afin de calculer l'affinité dans la base appropriée. Le total des différences entre les entiers du FT et l'unité régulatrice détermine le degré d'affinité. Ensuite, le produit de l'affinité et de la concentration du FT à chaque unité régulatrice est calculé puis attribué à une fonction sigmoïdale où un seuil fixe détermine si le gène correspondant est activé (signe positif) ou inhibé (signe négatif). Les équations correspondantes sont:

$$r_j = \sum_{i=1}^n aff_i * conc_i, \quad a_k = \frac{1}{1+e^{-\sum_{j=1}^n r_j}}, \quad g_k = \begin{cases} -1.0 & \text{if } a_k < 0.2 \\ 1.0 & \text{if } a_k > 0.8 \\ 0.0 & \text{autrement} \end{cases},$$

Où aff_i est l'affinité du FT_i avec l'unité régulatrice j , $conc_i$ est la concentration du FT_i , r_i est l'activité de l'unité régulatrice j , a_k est la somme totale des activités de toutes les unités régulatrices d'un gène K . Enfin, g_k est l'activité inhibition ou renforcement du gène K .

Un gène structurel actif peut exécuter un certain nombre de fonctions qui dépendent de son type (les trois premiers entiers d'un gène structurel figure 3.16). Il peut produire un FT , une molécule d'adhésion de cellule qui peut connecter deux cellules, un récepteur qui régularise la communication ente les cellules, ou exécuter une action cellulaire comme la division ou l'ameitose.

Pour simuler la différenciation cellulaire, Eggenberger a implémenté trois façons possibles pour échanger de l'information entre les cellules : la première avec des substances intracellulaires qui régularisent l'activité de leurs propres gènes, la deuxième avec des substances qui peuvent pénétrer la membrane de la cellule et atteindre les cellules voisines, la troisième avec des récepteurs spécifiques sur la surface de la cellule qui peuvent être stimulés par des substances appropriées. Les substances qui peuvent être diffusées dans les cellules voisines sont appelées morphogènes et permettent de produire des cellules avec l'information positionnelle. Ces morphogènes sont implémentés comme une sorte de FT . Le génome lit la concentration de ces morphogènes et les gènes sont activés dans une direction spécifique qui dépend de la concentration.

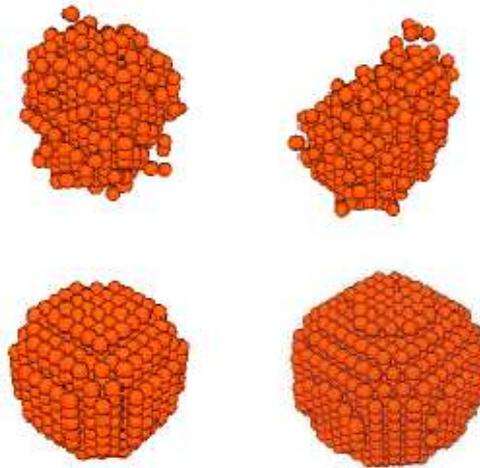


Figure 3.17-En haut groupes de cellules résultant d'une croissance aléatoire. En bas : formes obtenus après l'ajout d'un morphogène pour moduler la croissance.

Dans la première série d'expériences, chaque cellule cherche une place vide dans son voisinage de Margolus, puis en choisit une aléatoirement pour mettre sa cellule fille. S'il n'y a aucun voisin vide autour de la cellule, la division cellulaire est inhibée (croissance aléatoire : partie haute de la figure 3.17). Puis, un morphogène qui active la division cellulaire est positionné au milieu de la grille pour moduler le développement des cellules. Si l'influence de la concentration du morphogène et de l'affinité sont au dessus d'un certain seuil, la cellule stoppe la division (partie basse de la figure 3.17).

Dans une autre série d'expériences, une fitness qui récompense le bilatéralisme a été utilisée. Avant de permettre à la cellule initiale de se reproduire, des sources de morphogènes ont été placées dans l'environnement. Chacune positionnée suivant l'un des trois axes à distances variables de l'origine. Les trois morphogènes sont différents, de telle sorte qu'ils puissent régulariser la croissance indépendamment les uns des autres. La fonction de fitness dépend du nombre total de cellules et de leur position qui doit respecter l'un des axes. La figure 3.18 montre les formes 3D résultantes.

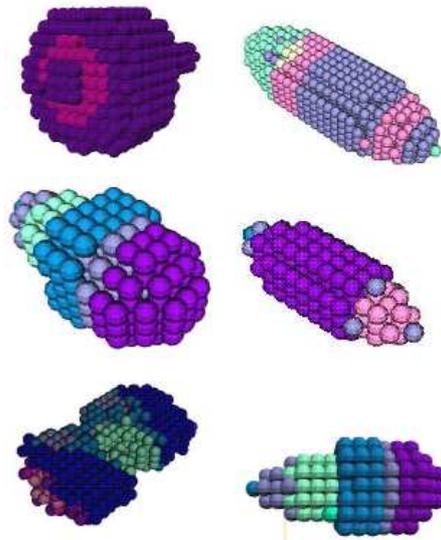


Figure 3.18-Exemples de formes tridimensionnelles évoluées.

Ces formes évoluées par l'AES peuvent être vues comme plans de robots tridimensionnels qui peuvent être utilisés pour produire des robots réels, D'autre part, il a été démontré que des réseaux neuronaux pour des robots réels peuvent être développés en utilisant le même modèle ce qui permet l'investigation de la coévolution de la morphologie et de son contrôle neuronal pour des organismes multicellulaire tridimensionnels [Egg97].

3.3.8 Réseaux artificiels de régulation.

Les réseaux artificiels de régulation (RARs) sont apparus dans le but d'imiter les réseaux génétiques rencontrés dans la nature. Torsten Reil a été le premier chercheur à proposer un génome artificiel avec des propriétés biologiques [Rei99]. Depuis plusieurs modèles ont été proposés. Dans cette section nous présentons les principaux modèles.

Le modèle de Torsten Reil

Dans son modèle, Reil a utilisé un génome défini comme une chaîne de chiffres générée aléatoirement. Les gènes, sont identifiés par une séquence qui les précède appelée promoteur. La séquence choisie comme promoteur est 0101(figure 3.19) ; similaire aux

boites TATA trouvées dans les génomes biologiques. Les N chiffres qui suivent le promoteur constituent le gène.

Après qu'un gène ait été identifié, il est traduit en une séquence régulatrice en utilisant une transformation simple. Chaque chiffre dans la séquence est incrémenté, avec l'opération modulo correspondante. Ces séquences se nomment "cis - éléments", comme leurs contreparties biologiques, et chacune d'entre elles contrôle les gènes qui la succède immédiatement.

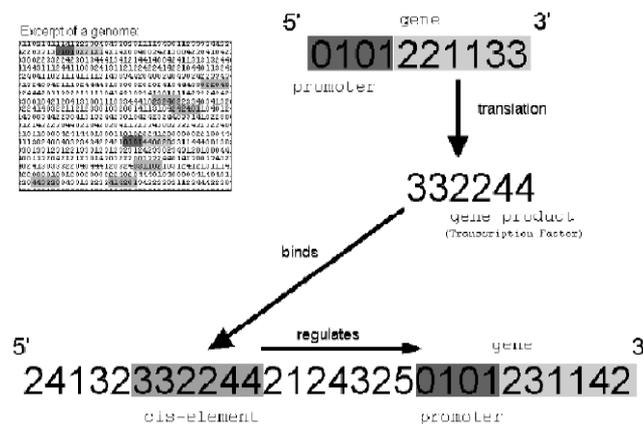


Figure 3.19-L'expression du gène et la régulation dans le génome artificiel. La taille du gène N est égale à 6.

Une séquence régulatrice peut se comporter comme renforçatrice pour activer un gène, ou comme inhibitrice pour bloquer son activité. Son rôle est défini par la valeur de son dernier chiffre. La structure résultante de l'expression de gènes est visualisée dans ce que l'auteur a appelé un *graphe d'expression*; un graphe 2D avec le temps sur l'axe des abscisses et un point noir pour chaque gène actif sur l'axe des ordonnées. Comme dans les RBAs et d'autres systèmes dynamiques, trois types de comportements ont été identifiés: *régulier*, *chaotique* et *complexe*.

Le modèle de Banzhaf

Après les travaux de Reil, un avancement très important dans le design du génome artificiel a été proposé par Banzhaf [Ban03]. Son génome consiste en une chaîne de nombres entiers de 32 bits générés aléatoirement. Similairement à d'autres modèles, un "promoteur" a été utilisé pour signaler le commencement du gène dans le génome. La séquence choisie pour le promoteur est "XYZ01010101", où "XYZ" représente une combinaison de 3 octets arbitrairement choisis. Ainsi, dans un génome généré aléatoirement, un promoteur peut apparaître avec la probabilité de $2^{-8} \% \approx 0.0039 = 0.39\%$. Les gènes qui suivent le promoteur sont définis comme une chaîne de cinq séquences de 32 bits, pour un total de 160 bits par gène. Il précède la séquence du promoteur, deux sites spéciaux un renforçateur et un inhibiteur avec une longueur de 32 bits également pour chacun.

Les cinq séquences de 32 bits juste après le promoteur sont traduites en protéine en utilisant la règle de la majorité: le premier bit dans la protéine traduite correspond au bit qui est dans la majorité en première position des cinq régions de codage de protéine, et ainsi de suite jusqu'à la fin de la séquence de 32 bits.

Après qu'une protéine ait été produite, elle est alors comparée bit par bit avec toutes les séquences d'inhibition et de renforcement de gènes dans le génome. Cette comparaison est accomplie avec l'opération XOR.

Banzhaf a confirmé, après des simulations multiples, que des génomes aléatoirement produits avec des tailles diverses contiennent le nombre souhaité de gènes. Par exemple, un génome de 100.000 bits contenait 409 gènes, qui est conforme à la règle de 0,39%.

Le degré de correspondance entre les protéines et les sites renforçateurs ou inhibiteurs définit le degré d'activation ou d'inhibition des gènes, respectivement. L'influence d'une protéine est exponentielle avec le nombre de bits de correspondance. La force de l'excitation en ou de l'inhibition in d'un gène i avec $i = 1 \dots, n$ est défini comme :

$$en_i = \frac{1}{n} \sum_{j=1}^n c_j e^{\beta(u_{ij}^+ - u_{max}^+)} \quad in_i = \frac{1}{n} \sum_{j=1}^n c_j e^{\beta(u_{ij}^- - u_{max}^-)}$$

où n est le nombre total de gènes régulateurs, c_j est la concentration de la protéine j , β est une constante, u_{ij}^+ et u_{ij}^- sont les nombres de correspondance entre la protéine j et les sites renforçateurs ou inhibiteurs du gène i respectivement, u_{max}^+ et u_{max}^- sont les correspondances maximales et minimales réalisables entre une protéine et un site renforçateur ou inhibiteur, respectivement. Une fois que en et in sont obtenues pour tous les gènes régulateurs, le changement de concentration correspondant pour la protéine i dans un pas de temps est calculé par l'équation:

$$\frac{dc_i}{dt} = \delta(en_i - in_i)c_i$$

où δ est une constante qui régularise le degré de changement de concentration d'une protéine. Ainsi, les concentrations de protéines sont mises à jour.

En utilisant des génomes générés aléatoirement sans aucune évolution impliquée, Banzhaf a découvert plusieurs types de dynamiques de concentration de protéine dans le temps. La figure 3.14 présente quelques exemples.

Après avoir observé la dynamique des protéines de plusieurs génomes qui n'avaient éprouvé aucune évolution, Banzhaf a décidé d'utiliser la programmation génétique (PG) afin d'essayer de conduire la dynamique de l'expression des gènes vers quelques comportements spécifiques. Pour cela, il a commencé par faire évoluer le génome pour obtenir une concentration cible d'une protéine particulière. Il a rapidement découvert, ensuite, qu'en général le processus évolutionniste converge rapidement vers l'état cible.

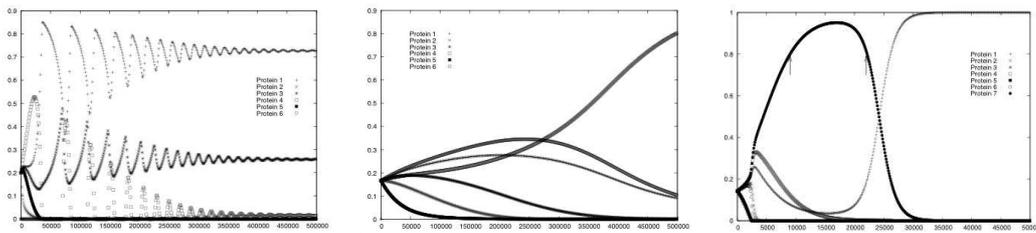


Figure 3.20 -Exemples de développement temporel des concentrations de protéines. (a) un changement de concentration oscillatoire ; (b) un changement de concentration lent et régulier ; (c) Transition de changement de concentration entre deux protéines.

Dans une recherche pour trouver d'autres applications de ce modèle, Banzhaf et ses collègues [KB04] ont fait évoluer des génomes où les concentrations en protéine ont été employées pour moduler les rendus de fonctions telles que : les fonctions sinusoïdales, exponentielles et sigmoïdales [KLB04]. Une chaîne de 62-bit, aléatoirement sélectionnée dans le génome, a été choisie pour fonctionner comme un renforceur et un inhibiteur, qui ont permis d'interagir librement avec les protéines exprimées. Cependant, au lieu de contrôler l'expression de gène, ces sites ont été utilisés pour calculer les solutions d'une fonction dans l'intervalle $[-1, 1]$. Ces auteurs ont constaté qu'il était faisable de faire évoluer des génomes pour générer des séries temporelles pour l'optimisation de fonctions. Ils ont aussi utilisé l'évolution du modèle du génome artificiel pour produire des réseaux de neurones.

Le modèle de Nehaniv

Le groupe de recherche de Nehaniv a travaillé sur l'évolution d'un RAR afin de simuler un modèle d'horloge biologique [KNSQ06, KNS07]. Dans ce modèle, le génome consiste en un nombre de gènes où l'expression de chacun est régularisée par un certain nombre de sites régulateurs complexes. Un algorithme évolutionniste a été employé pour faire évoluer le RAR autant que système de contrôle actif qui répond à des stimuli environnementaux périodiques de différents types par les comportements périodiques appropriés. Bien que le modèle d'horloge périodique ne considère que l'évolution d'un génome d'une seule cellule, les résultats de ce modèle peuvent être utilisés pour synchroniser la reproduction de cellules dans un modèle de développement artificiel.

Le modèle de Chavoya et Duthen

Afin de tester les fonctionnalités des RARs évolués par un AG, Chavoya et Duthen [CD07c] ont développé un modèle de croissance cellulaire basé sur le paradigme d'Automates Cellulaires. La croissance cellulaire est accomplie à travers l'expression de gènes structurels qui sont eux mêmes contrôlés par un RAR.

Le génome utilisé est constitué d'une chaîne binaire commençant par une série de gènes régulateurs, suivie d'un certain nombre de gènes structurels (figure 3.20). Le réseau de

régulation génétique implémenté est basé sur le modèle de RAR proposé par Banzhaf [Ban03].

D'autre part, les gènes structurels codent la forme particulière qui va être développée par les cellules produites. Ces gènes structurels ont été obtenus en utilisant la méthodologie présentée dans [CD06a]. Brièvement, un automate cellulaire a été choisi comme un modèle de croissance cellulaire et s'est vu évoluer par un AG afin de produire des formes bidimensionnelles prédéfinies.

Chaque gène régulateur dans le RAR se compose d'un site inhibiteur, d'un site renforçateur et des séries de régions de codage de protéine régulière (figure 3.21). Chaque gène structurel correspond à un gène régulateur. Par exemple, le gène structurel numéro "1" est associé au gène régulateur numéro "1". L'expression structurelle de gène est visualisée dans le modèle de croissance cellulaire comme une couleur distincte externe de la cellule.

La règle de translation des régions de codage de protéines régulatrices en une protéine régulatrice ainsi que l'opération de comparaison qui permet de déterminer le degré de correspondance, les formules de calcul de la force d'inhibition, la force d'excitation et le changement de concentration d'une protéine sont les mêmes présentés par Banzhaf.

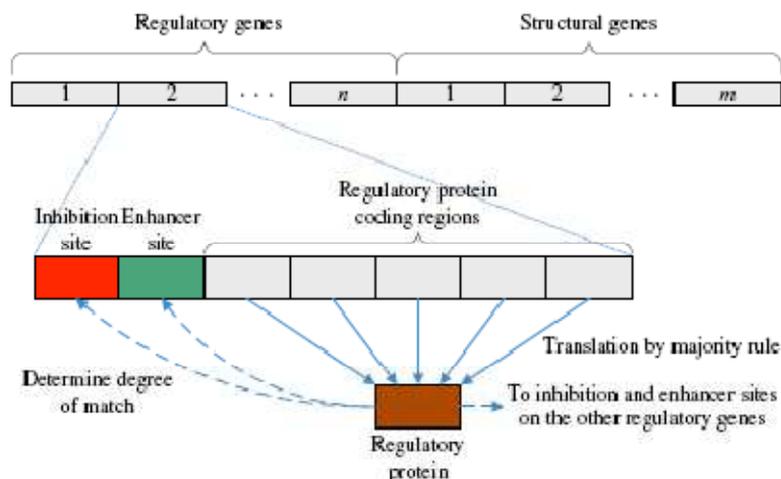


Figure 3.21: La structure du génome et les détails d'un gène régulateur.

Un gène structurel est actif si et seulement si la concentration de la protéine régulatrice traduite par le gène régulateur correspondant était au-dessus d'un certain seuil. La valeur choisie pour le seuil était 0.5, puisque la somme de toutes les concentrations en protéine est toujours égale à 1.0, et il ne peut y avoir, à la fois qu'une seule protéine avec une concentration au-dessus de 0,5. Ainsi, un seul gène structurel peut être exprimé à chaque pas de temps. Si un gène structurel est actif, alors la table de règle de reproduction de l'automate cellulaire codée dans ce gène est utilisée pour contrôler la reproduction des cellules.

Plusieurs séries d'expériences ont été reportées. En premier lieu, le modèle de croissance cellulaire a été utilisé pour produire des formes simples. La figure 3.22 (a) montre un carré

21×21 de deux couleurs établi par l'expression de deux gènes structurels codés pour un carré d'une couleur différente.

Dans le but de créer des structures plus complexes, un carré de trois couleurs exprimant trois gènes structurels où chacun d'entre eux code pour un carré, a été développé. Dans ce cas, il a été difficile de synchroniser l'expression des trois gènes régulateurs dans un ordre spécifique. Pour cela et afin d'augmenter la probabilité pour que l'AG puisse trouver le RAR approprié, au lieu d'utiliser une série de trois gènes structurels, un tandem de deux séries de trois gènes structurels a été employé, pour un total de six gènes structurels. De ce fait, pour créer par exemple le carré d'intérieur, le RAR pourrait exprimer le gène structurel numéro "1" ou le gène numéro "4" et de même pour les autres carrés. En utilisant un RAR avec cette configuration des gènes structurels, le carré 25×25 avec trois couleurs montré dans la figure 3.22 (b) a été obtenu.

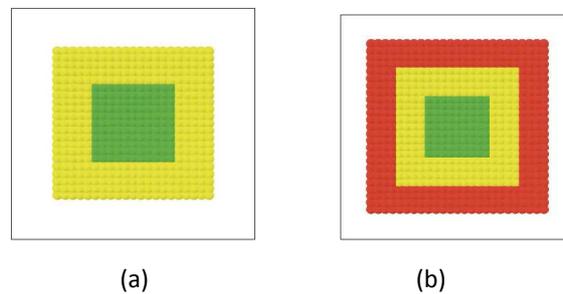


Figure 3.22-(a) Le carré de deux couleurs. (b) : Le carré de trios couleurs.

Enfin, afin d'explorer le résultat de la combinaison de quelques gènes structurels, le modèle a été appliqué au problème de la génération du drapeau français. Un drapeau de 21 × 7 (figure 3.23 (b)) a été développé par l'expression des trois gènes. Après que le carré blanc soit formé, le carré bleu gauche est développé, suivi du carré rouge droit. Pour illustrer un ordre différent d'activation du gène, un second drapeau de 27 × 9 (figure 3.23 (a)) a été développé. Notons que dans ce cas, le carré rouge est formé avant le carré bleu.

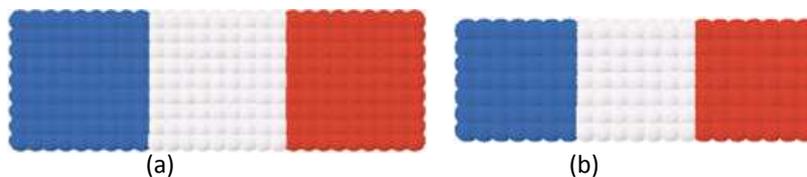


Figure 3.23-Drapeau Français de Chavoya. (a) Le drapeau de (27 × 9). (b) Le drapeau de (21 × 7).

Les expériences présentées par ces auteurs montrent que l'utilisation d'un RAR évolué par un algorithme génétique donne des résultats reproductibles pour créer des structures cellulaires bidimensionnelles prédéfinies à partir d'une seule cellule initiale. En particulier, en utilisant ce modèle de RAR, il est possible de synchroniser jusqu'à trois gènes, bien que quelques problèmes aient été rencontrés en essayant de synchroniser l'activation de plus de trois gènes dans un ordre précis.

3.3.9 METAMorph.

METAMorph pour *Model for Experimentation and Teaching in Artificial Morphogenesis* est une plateforme qui permet de concevoir des génomes artificiels et de visualiser le processus de développement morphologique correspondant [STK05].

Ce modèle s'inscrit également dans le cadre des travaux sur les réseaux de régulation artificielle mais sans l'utilisation d'un algorithme évolutionniste. Un organisme multicellulaire artificiel est développé à partir d'une cellule unique. Le réseau de régulation modélise l'interaction entre les gènes, les protéines, et l'environnement cellulaire (figure 3.24). Une protéine est définie par : un nom unique, un type (interne ou externe) et deux constantes: affaiblissement (le niveau de la protéine diminue à chaque pas de temps) et diffusion (les protéines à un site de diffusion se déplacent vers d'autres sites, dans la même cellule ou dans les cellules adjacentes). Les protéines internes ne peuvent se diffuser qu'à l'intérieur de la cellule, tandis que les protéines externes traversent les membranes de cellules et peuvent par conséquent être utilisées pour la signalisation intercellulaire.

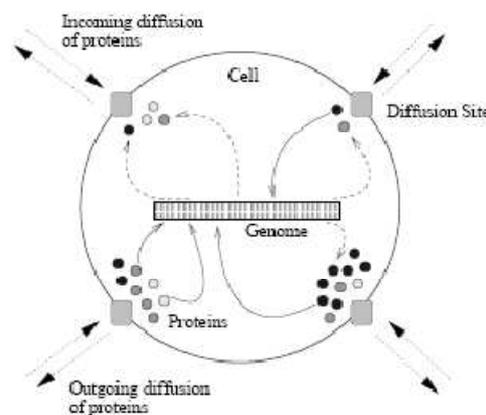


Figure 3.24-METAMorph, réseau de régulation génétique cellulaire.

Toutes les cellules possèdent le même génome composé d'un certain nombre de gènes. Chaque gène produit exactement une protéine, et la même protéine peut être produite par plusieurs gènes. Les cellules sont représentées par des sphères et chacune occupe une position unique dans une grille iso-spatiale tridimensionnelle. Chaque cellule peut effectuer un certain nombre d'actions. Chacune ne peut être déclenchée que si la concentration moyenne d'une protéine spécifique dans la cellule dépasse un certain seuil. Les actions possibles sont :

La division cellulaire : s'il y a une place vide valable, la cellule fille est placée dans la direction de l'axe mitotique.

Le mouvement de l'axe mitotique : l'orientation de la cellule est réalisée par la définition d'un "axe mitotique" se dirigeant vers l'une des 12 positions adjacentes possibles. L'orientation de cet axe peut être modifiée quand le niveau de quelques protéines spécifiques atteint un certain seuil.

L'apoptose : la cellule est enlevée de sa position laissant un espace vide.

La différenciation : le type de la cellule est visualisé dans le modèle par une couleur externe distincte. Le type d'une cellule n'est pas lié à sa fonction.

En utilisant cette plateforme de simulation, un embryon en forme de cigare d'à peu près 700 cellules a été développé. Pour cela, une ligne d'approximativement 10 cellules qui forment l'axe central du cigare a été créée, (figure 3.25(a)). Toutes ces cellules émettent une protéine externe (figure 3.25(b)). Les cellules se développeront alors partout où la concentration de cette protéine de signalisation est au-dessus d'un certain niveau, en produisant la forme désirée (figure 3.25(c)).

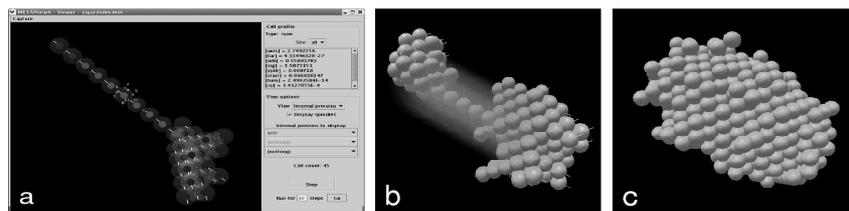


Figure 3.25- Développement d'un embryon en forme de cigare. (a) Début du développement. (b) Les cellules se développent sur les des deux extrémités. (c) Le cigare enfin réalisé, se composant de 684 cellules.

METAMorph est un outil interactif potentiel pour l'étude et la compréhension de processus de croissance cellulaire. Cependant, bien que les auteurs proposent quelques techniques d'aide à la conception, l'inconvénient principal de cette plateforme demeure dans le fait que le modèle de développement cellulaire à concevoir passe par un processus d'essai-erreur qui est limité par la capacité du concepteur à fournir les valeurs appropriées des paramètres. De l'avis des auteurs, ce processus d'essai-erreur implique un coût de calcul considérable, puisque les temps de simulation sont d'habitude élevés dus à la nature parallèle du processus morphogénétique.

Par conséquent, METAMorph peut tirer des bénéfices considérables en introduisant dans la plateforme un algorithme évolutionniste de telle sorte que le système puisse trouver par lui-même une conception appropriée pour le modèle de croissance cellulaire désiré.

3.3.10 Problèmes du drapeau.

Les problèmes de génération de drapeaux (*flags problems*) sont très récurrents dans les systèmes d'embryogénèse artificielle. Il s'agit de développer un modèle de développement dont le phénotype doit correspondre à une forme 2D donnée (drapeau). La première utilisation de ce terme trouve son origine, dans les travaux de Wolpert [Wol68], qui a cherché à formuler le problème de régulation et de développement de structure cellulaires. Ce dernier a utilisé un drapeau français (French flag problem). Par la suite plusieurs chercheurs ont travaillé sur de tels modèles.

Récemment, Devert [DBS07] a développé un modèle de développement continu dans lequel il valide son approche sur le problème de drapeau. Contrairement aux modèles de la

Morphogenèse des réseaux de neurones (section 3.3.5) qui consistent à développer des réseaux de neurone, ce modèle utilise un réseau de neurone pour contrôler le développement des organismes. Les cellules, dans ce modèle, sont fixées sur une grille bidimensionnelle. Chacune peut inter-changer les produits chimiques avec les cellules qui se trouvent dans son voisinage. La quantité de chaque produit chimique que la cellule produit ainsi que la valeur de sa différenciation (une couleur) sont contrôlés par un réseau de neurone, qui prend comme entrées les produits chimiques produits par les cellules voisines dans le pas de temps précédent. Les résultats du modèle ont montré qu'il est capable d'exhiber des caractéristiques d'autoréparation (autoreproduction).

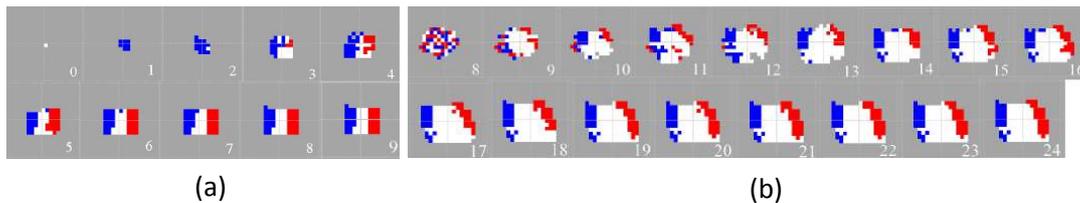


Figure 3.26- Drapeau Français de Miller. (a) Croissance à partir d'une seule cellule blanche. (b) Drapeau français capable de se reproduire après un endommagement aléatoire [Mil04].

Il existe d'autres travaux sur l'autoréparation et l'autorégulation. Parmi lesquels, nous citons le travail pionnier de Julian Miller [Mil04]. Ce chercheur a développé un drapeau Français de 12 x 9 capable de se reproduire à 95%. Pour contrôler le développement, un circuit logique booléen a été utilisé [MB03] et optimisé par la programmation génétique cartésienne [MT02].

Les travaux cités précédemment, comme ceux de Garis qui a généré des formes 2D convexe et concaves, s'inscrivent aussi dans ce cadre. Les travaux de Chavoya également qui a travaillé sur des carrés (de deux et de trois couleurs) ainsi sur la génération de drapeaux Français.

3.3.11 Cell2organ.

Cell2organ [CLD08, CLD09a, CLA09b, CLD09c] est un modèle de développement cellulaire qui permet de créer des organes ainsi que des créatures multicellulaires avec une morphologie définie par l'utilisateur. L'environnement est implémenté sur la base d'une grille torique bidimensionnelle. Il contient différents substrats et des cellules possédant chacune

- un ensemble de capteurs,
 - un système de sélection d'action,
 - une représentation du réseau de régulation génétique et
 - est capable d'exécuter un certain nombre d'actions, qui sont : la transformation de substrat, l'absorption ou le rejet de substrats, la duplication, la survie et l'apoptose.
- La liste des actions valables pour chaque cellule est définie dans son code génétique.

Dans ce modèle, La duplication est une action spécifique dans laquelle la cellule peut être spécialisée pour optimiser un groupe d'actions au détriment d'autres. Cette spécialisation

est réalisée par un réseau de régulation génétique. Chaque action possède un coefficient d'efficacité qui correspond au niveau d'optimisation de l'action : plus ce coefficient est élevé, moins est l'énergie nécessaire à l'exécution de cette action. De plus, si ce coefficient est nul, l'action n'est plus valable pour la cellule fille. En outre, La somme de tous les coefficients d'efficacité doit rester constante durant la simulation. En d'autres termes, si une action est optimisée pour augmenter son coefficient d'efficacité durant la duplication, une autre (ou plusieurs) doit être optimisée pour la diminuer. La cellule est spécialisée en faisant varier les coefficients d'efficacité durant la duplication. Les règles de ces variations sont données par le réseau de régulation artificiel. Ce dernier est simulé par un réseau dont :

- les nœuds représentent les actions avec leurs coefficients d'efficacité ;
- les arcs sont étiquetés par des poids et représentent les quantités du coefficient d'efficacité qui vont être transférées pendant la duplication.

Afin de contrôler le développement de leurs créatures, les auteurs ont utilisé un algorithme génétique dont le génome se compose de trois chromosomes : la liste des actions valables, le système de sélection d'actions ainsi que le réseau de régulation génétique qui permet la spécification durant la duplication.

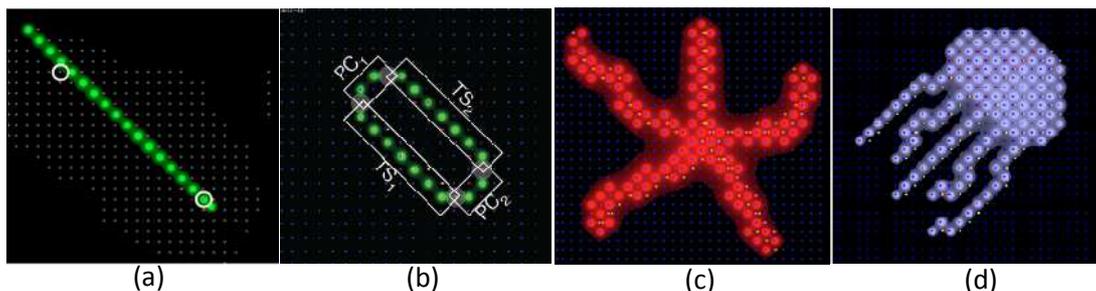


Figure 3.27: Créatures développées. (a) un organe capable de transférer un substrat d'un point à l'autre. (b) Organisme autoalimenté : coopération de 4 différents organes. (c) une étoile de mer artificielle. (d) méduse artificielle.

Ce modèle est capable de produire différents organes⁵ : le moissonneur, capable de dégrader un substrat particulier de l'environnement. Un autre organe capable de déplacer un substrat d'un point à un autre de l'environnement (figure 3.28 (a)). Suite au développement de ces deux organes, les auteurs ont voulu tester les capacités d'assemblage de leur modèle. En partant du système de transfert et en développant deux nouveaux organes, ils ont pu produire un organisme plus complexe en assemblant quatre organes différents et en obtenant un organisme autoalimenté utilisant au mieux les ressources de l'environnement (figure 3.28(b)).

Ce modèle est capable aussi de produire des créatures avec une morphologie décrite par utilisateur : étoile de mère (figure 3.28(c)), une méduse (figure 3.28(d)). De plus ces créatures sont capables de faire émerger la capacité d'autoréparation.

⁵ Un organe, par définition, est un ensemble de cellules spécialisées qui possède une fonction spécifique

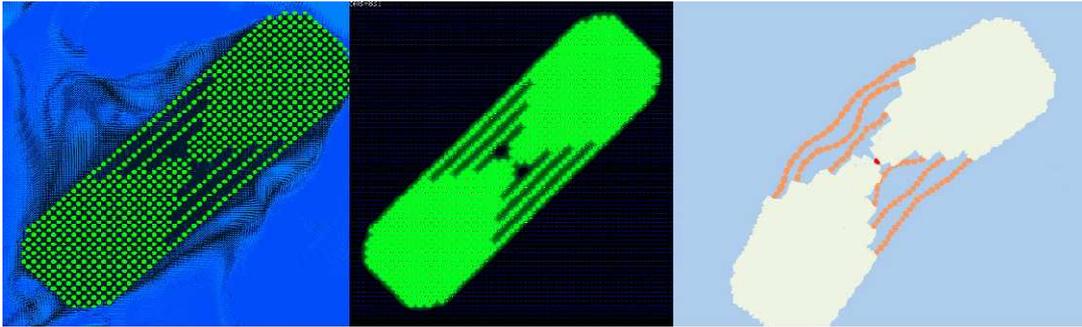


Figure 3.28-Intégration des 3 niveaux de simulation (de gauche à droite : le niveau hydrodynamique, le niveau chimique et le niveau physique).

Le but initial des auteurs était de générer des créatures artificielles à partir d'une cellule unique possédant des capacités locomotives. Pour cela, enfin de leur travaux ils ont présenté une étude préliminaire de la création de deux simulateurs supplémentaires autour de ce modèle : un simulateur physique permettant de plonger leurs créatures dans un monde aux lois newtoniennes et un simulateur hydrodynamique permettant la simulation des flux de matière de l'environnement (figure3.29). Les premiers résultats qu'ils ont présenté sont très encourageants et montrent la complexité de futures créatures qu'on puisse avoir en intégrant les trois simulateurs.

3.4 Synthèse.

Nous synthétisons, dans cette section, les principaux travaux sur le développement artificiel, qui ont fait objet de la section précédente. Pour ce faire, nous allons d'abord les classier. Dans la littérature, on peut distinguer deux types de classification des systèmes d'embryogenèse artificielle : encodage directe\indirecte\implicite (pour plus de détails voir [BK99]), approche grammaticales\approches basées sur la chimie cellulaire (pour plus de détail voir [SM03]). Historiquement, la seconde classification reflète la distinction classique en intelligence artificielle entre les approches de bas niveau (*top down*) et celles de haut niveau (*buttom up*). Pour d'autres raisons que nous expliquerons par la suite, c'est cette classification que nous utiliserons pour distinguer les différents travaux sur l'ontogenèse que nous avons présentée dans ce chapitre.

3.4.1 Encodage directe versus indirecte versus implicite.

Cette classification fait intervenir la représentation ou le codage choisi pour l'algorithme évolutionniste. En effet, si la représentation de la structure évoluée est utilisée telle quelle, on parle de codage direct (*direct mapping*). Si la représentation est utilisée en tant que moyen pour d'autres dynamiques pour générer la structure, on parle dans ce cas du codage indirect (*indirect mapping*). Plus précisément, dans le codage direct, il n'y a pas de développement. Le génotype est la représentation directe du phénotype. A notre avis, les

systemes utilisant ce type de codage ne font pas partie des systemes d'embryogenèse artificielle. Dans le codage indirect, le développement est un programme explicite - généralement basé sur une grammaire - et le phénotype représente le résultat de l'exécution du génotype. Dans cette classification, on distingue également le codage implicite (*implicit Embryogeny*). Dans ce dernier, la représentation utilisée est plus compacte que dans le codage indirect et le développement est basé sur des dynamiques plus proches de celles vues dans le développement naturel : le métabolisme, la diffusion chimiques, l'expression des gènes, actions cellulaire, etc.

3.4.2 Approches grammaticales versus basées sur la chimie cellulaire.

Cette classification fait intervenir les mécanismes utilisés pour contrôler le développement, c'est à dire les mécanismes biologiques qui ont été simulés, pour distinguer deux approches. Les approches dites grammaticales se basent sur des mécanismes de lignée cellulaire. Elles utilisent des règles de réécritures appliquées récursivement pour transformer un état initial en un phénotype. Ce type d'approche correspond au codage indirect de la première classification. La première utilisation des grammaires pour simuler le développement d'organismes multicellulaires trouve son origine dans les systemes de Lindenmayer [Lin68] qui avait, au départ, travaillé sur la croissance des bactéries, des champignons et des algues. Après, l'informatique a permis d'exploiter ces systemes pour simuler graphiquement des modèles de développement de plantes très complexes [Pru86, Pru87].

Parmi les travaux qui appartiennent à ce type d'approche, nous trouvons les Biomorphs de Dawkins qui illustrent la façon dont des structures complexes, comme l'œil humain, peuvent être créées par des règles très simples telles que la micromutation et la sélection cumulative. Nous citons également les travaux de Kitano, Boers et Grueau sur la morphogenèse de réseaux de neurones.

Les approches basées sur la chimie cellulaire se basent sur les aspects de bas niveau de l'embryogenèse naturelle. En effet, dans ce type d'approche, le développement est décrit par la simulation de mécanismes tels que : la diffusion, la réaction/diffusion, l'expression et la régulation de gènes, le métabolisme, les réseaux de régulation génétiques, la différenciation, la division cellulaire... etc. Les premiers travaux, dans ce type d'approche, trouvent leurs origines dans le modèle de réaction diffusion de Turing dans lequel des substances appelées morphogènes réagissent entre elles pour produire des structures très complexes.

Les réseaux booléens aléatoires proposés par Kauffman représentent l'une des premières approches qui a utilisé des réseaux booléens pour modéliser les réseaux de régulation génétique. La topologie du réseau est produite aléatoirement et consiste en un ensemble de nœuds connectés. Chaque nœud représente un gène qui peut être actif ou non en fonction des règles qui y sont codées et qui sont également produites aléatoirement.

travaux	Approche		Contrôle du développement							Génotype			évolution		phénotype			environnement		Mécanismes simulés				
	grammaticale	chimique	RBA	L-système	AC	RAR	Equations différentielle	Réseau de neurones	Expression + régulation de gènes	autres	L-système	RAR	Réseau de neurone	AG	PG	graphes	Réseau de neurones	drapeau	autres	2D	3D	diffusion	métabolisme	Action s cellulaires
[Tur52]		X					X												X	X		X		
[Daw96]	X			X																X				
[Kau69, 71, 93]			X													X				X				
[De, 92, 99]		X			X								X					X		X				
[Kit90]	X			X						X			X				X			X				
[BKHS93]	X			X						X			X				X			X				
[Gru93, 94,95]	X			X													X							X
[Kit95]		X								X													X	X
[FB94, Fle96]	X	X		X		X		X									X		X	X	X	X		X
[Egg97]		X						X									X		X		X	X		X
[Rei99]		X				X									X					X				
[Ban03]		X				X					X			X	X					X				
[KLB04, KB04]		X				X											X							
[Mil04]		X							X									X		X				X
[STK05]		X				X													X		X			X
[KNSQ06, KNS07]		X				X					X								X					
[Cha07]		X			X	X							X					X		X				
[DBS07]		X					X											X		X				
[CLD08, CLD09]		X							X				X						X	X		X	X	X

Tableau 3.1: Synthèse des travaux sur le développement artificiel.

Hugo de Garis, quant à lui a été l'un des premiers à employer le concept d'activation séquentielle de gènes pour simuler le développement d'embryons artificiels de forme concave et convexe. Son travail était basé sur une approche intégratrice des modèles de diffusion chimique, des automates cellulaires, et des algorithmes génétiques.

Fleischer et Barr ont également utilisé des modèles de diffusion chimique afin de développer une plateforme de simulation pour l'étude de la formation d'organismes multicellulaires. Cette plateforme inclut des facteurs électriques, mécaniques (tels que l'adhésion) et des mécanismes de contrôle de la lignée cellulaire. Ainsi, ce modèle se situe à mi-chemin entre les deux types d'approches.

D'un autre côté, Eggenberger a utilisé une approche évolutionniste pour simuler la morphogenèse d'organismes en 3D basée sur la diffusion chimique et l'expression différentielle des gènes.

Les réseaux artificiels de régulation (RARs) sont apparus dans le but d'imiter les réseaux génétiques rencontrés dans la nature. Nous avons présenté plusieurs travaux sur ce sujet : les travaux de Banzhaf Nehaniv, Chavoya, etc. Ces travaux ont été faits dans la lignée des travaux de Torson Reil qui a été le premier chercheur à avoir proposé un génome artificiel avec des propriétés biologiques.

En fin, Cell2Organ est un autre modèle de développement cellulaire qui simule la spécialisation cellulaire en cellules plus adaptées à leur environnement, les réactions chimiques ainsi que la diffusion de substrats dans l'environnement.

Dans le tableau 3.1, nous récapitulons tous les travaux que nous avons présentés dans ce chapitre, en définissant pour chaque modèle : l'approche utilisée, les mécanismes qui ont été employés pour contrôler le développement (automate cellulaire (AC), réseaux de neurones, RAR, L-système...etc.), le génotype utilisé, le phénotype développé, l'espace environnementale, ainsi que l'algorithme évolutionniste choisi s'il a y eu une évolution artificielle.

3.5 Conclusion.

Nous avons présenté dans ce chapitre, les modèles pionniers d'ontogenèse artificielle qui ont marqué l'évolution du domaine, ainsi que les principaux développements récents. Après une brève introduction sur le développement biologiques et sa place dans l'évolution, nous avons décrit également ses mécanismes de base pour mieux comprendre les modèles artificiels qui s'y inspire. Ensuite, nous avons défini l'ontogenèse artificielle, et présenté ses caractéristiques et objectifs. Enfin avant de conclure le chapitre, dans la dernière section, nous avons présenté une synthèse des différents travaux sur le développement artificiel en nous basant sur deux classifications reconnues par la littérature qui sont : l'encodage directe/indirecte/ implicite et l'approche grammaticale/ basée sur la chimie cellulaire. Nous

reviendrons d'une manière plus détaillée sur cette deuxième classification dans le chapitre suivant, pour expliquer le modèle de développement cellulaire proposé dans ce mémoire et qui est fondé sur la combinaison de ces deux approches.

CHAPITRE 4

Un modèle bio-inspiré pour la synthèse de créatures artificielles

"Créer n'est pas un jeu quelque peu frivole. Le créateur s'est engagé dans une aventure effrayante qui est d'assumer soi-même jusqu'au bout les périls risqués par ses créatures.", Jean Genet.

Les deux chapitres précédents ont abordé différentes approches et mécanismes concernant la synthèse de créatures ou organismes artificiels. L'objectif de ce chapitre est de présenter un nouveau modèle de développement cellulaire destiné à répondre aux questions de la problématique définie dans ce mémoire. Ce modèle se situe à mi-chemin entre les approches grammaticales et celles basées sur la chimie cellulaire. D'une part, il simule les éléments majeurs d'un modèle de chimie cellulaire à savoir : les réactions chimiques d'une façon simplifiée, la diffusion de substrats dans l'environnement, les interactions cellulaires et environnementales ainsi que quelques actions cellulaires comme la division ou l'apoptose. D'autre part, la morphogenèse est fondée sur une approche grammaticale utilisant des règles de production codées dans un APL-système ; variante de L-système paramétrique que nous proposons pour contrôler d'une manière adaptative la croissance de nos organismes multicellulaires en fonction de l'état interne des cellules les composants.

Dans la première section, nous décrivons notre modèle d'une manière générale en définissant ses objectifs et présentons la démarche à suivre pour y répondre ainsi que sa structure générale. Après, nous détaillons la partie ontogénétique de notre modèle en commençant par l'environnement embryonnaire dans lequel nos organismes multicellulaires se développent. Nous décrivons ensuite, les différents mécanismes que nous proposons pour contrôler le développement à savoir :

- le métabolisme basé sur un modèle simplifié d'une chimie artificielle,

- les interactions cellulaires et environnementales modélisées par un système de sélection d’actions,
- et la morphogenèse basée sur notre APL-système.

Ensuite, pour guider le développement, nous utilisons un algorithme génétique que nous décrivons avant la fin du chapitre en présentant le génotype, les opérateurs génétiques utilisés ainsi que la boucle générationnelle de l’évolution illustrée par une approche ontogénétique. Enfin, nous présentons les principes d’implémentation de notre modèle. Ce travail est présenté également dans [DDCD09].

4.1 Description globale.

4.1.1 Objectifs du modèle.

Nous présentons dans cette section les objectifs de notre modèle.

Développement de créatures complètes.

L’objectif de notre travail s’inscrit dans le cadre de la génération de créatures artificielles. Dans l’état de l’art, nous avons vu qu’il existe plusieurs modèles permettant de simuler, à des niveaux d’abstraction très différents, ces créatures peuplant actuellement les mondes virtuels.

D’une part, les modèles issus de la phylogénèse artificielle permettent d’assembler des blocs et des contrôleurs construits par évolution pour produire de grosses créatures capables de marcher, nager ou faire de la planche à roulette.

Les modèles issus de l’ontogénèse artificielle permettent, quant à eux, de simuler le développement d’un organisme multicellulaire en partant d’une cellule unique. Cette approche permet, d’un côté, de vérifier les hypothèses formulées par les biologistes quant à l’étude des systèmes vivants. D’un autre côté, elle permet de rapprocher les créatures artificielles de celles existantes dans la nature. Elle est également utile, si on veut avoir des créatures, voir même des robots avec la capacité de s’auto-réparer.

Dans le but de produire des créatures à partir d’une seule cellule initiale et qui pourront accomplir quelques fonctions de locomotion telle que la capacité de se déplacer nous voulons créer un pont entre les deux approches.

Couplage approche grammaticale / basée sur la chimie cellulaire.

Nous avons présenté également les principaux travaux sur le développement artificiel. Nous avons pu remarquer que ces travaux peuvent être classifiés en deux catégories : les approches grammaticales et chimiques.

Concernant les mécanismes utilisés pour contrôler le développement (mécanismes biologiques simulés ou pris en compte par le modèle), dans les approches grammaticales le développement est contrôlé par un mécanisme de lignée cellulaire basé sur un ensemble de règles de réécritures appliquées récursivement pour transformer un état initial (cellule) en un phénotype (organisme multicellulaire).

Ces approches produisent de très bons résultats en ce qui concerne la morphogénèse de réseaux de neurones [Kit90, BKHS93, Gru95], la conception évolutionniste de créatures artificielle ou réelles (robots) [Sim94a, HP02, PHLF03], etc. Néanmoins, elles ne semblent pas bien adaptées à des processus suggérant l'autonomie des éléments du système évolué. Pour cela, de nombreux chercheurs ont tenté de développer d'autres méthodes plus proches de la réalité biologique en simulant, via les approches basées sur la chimie cellulaire, des aspects de bas niveau de l'embryogénèse naturelle. En effet, dans ce second type d'approche, le développement artificiel est basé sur des dynamiques plus proches que celle vus dans le développement naturel telles que : la diffusion, la réaction/diffusion, l'expression et la régulation de gènes, le métabolisme, les réseaux de régulation génétiques, la différenciation, la division cellulaire, etc.

Notons que les modèles de développement basés sur ces mécanismes et notamment, les réseaux de régulation artificiels prennent en compte des phénomènes plus élémentaires que ceux proposés par l'approche grammaticale et permettent ainsi de manière générale de définir des codages génétiques plus expressifs (la taille de la représentation est moins liée à celle du phénotype). Cependant, comme le note Dellaert et Beer [DB96], les opérateurs génétiques sont plus difficiles à définir et la perte de performance en terme de convergence n'est pas compensée par le gain potentiel d'expression. Par exemple, dans le cadre du développement des réseaux de neurones, jusqu'à ce jour, les approches grammaticales sont plus performantes.

Donc, comme récapitulation, nous intégrons l'approche grammaticale au second type d'approche pour les raisons suivantes :

- L'intégration du mécanisme de lignée cellulaire aux mécanismes que nous simulons via l'utilisation de l'approche de chimie cellulaire.
- Les approches grammaticales sont jusqu'à ce jour plus performantes que celles basées sur la chimie cellulaire.
- En outre, les approches grammaticales et surtout les L-systèmes ont donné de bons résultats comme modèles de développement. Pour cela nous voulons également explorer le potentiel de ces systèmes.

Intégration du métabolisme.

Comme nous l'avons vu également dans l'état de l'art, il n'existe que très peu de modèles intégrant un métabolisme. Celui-ci est pourtant primordial dans le maintien de l'intégrité de

tout l'organisme en permettant à chaque cellule de survivre dans son environnement. Nous voulons intégrer ce mécanisme dans notre modèle de développement cellulaire à cette fin.

Combinaison de plusieurs mécanismes pour le développement.

Fleischer et Barr ont montré dès 1996 qu'il est difficile de maintenir la taille et la forme d'un organisme multicellulaire sans la combinaison de multiples mécanismes lors de la formation de l'organisme. Pour cela, nous prenons en compte les mécanismes suivants :

- Un mécanisme simple de diffusion chimique,
- un modèle très simplifié d'une chimie artificielle,
- quelques actions cellulaires comme la division et l'apoptose,
- l'interaction des cellules avec l'environnement dans lequel elles se développent,
- un mécanisme de lignée cellulaire modélisé par un L-système paramétrique.

4.1.2 Démarche.

Dans l'optique de générer des créatures complètes à partir d'une seule cellule initiale possédant une morphologie, un métabolisme et un comportement, nous utilisons **une approche hybride ontogénétique et phylogénétique**.

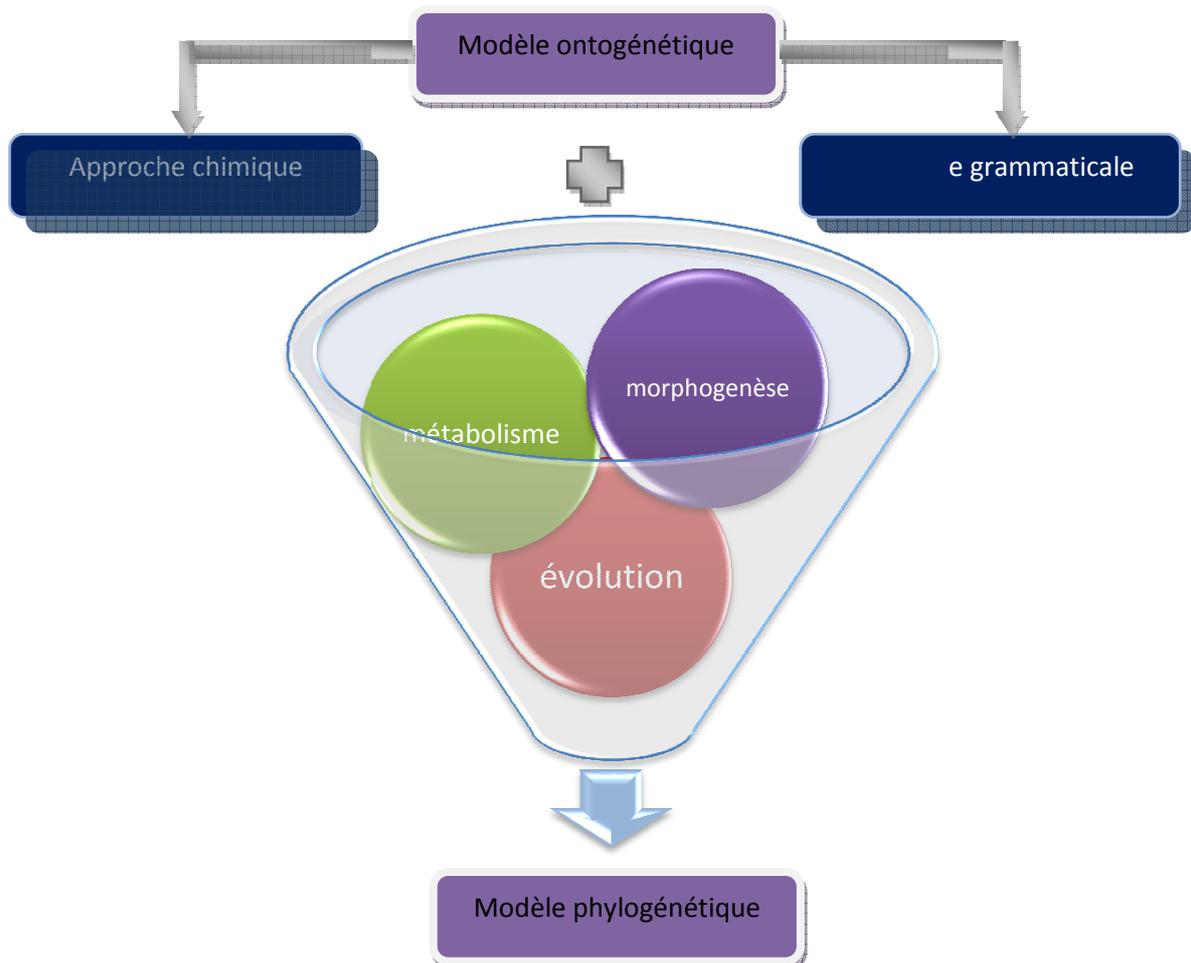


Figure 4.1 - Contribution.

Pour ce faire et afin de synthétiser en premier lieu des organismes multicellulaires artificiels (partie ontogénétique) qui pourront par la suite être utilisés pour développer des créatures complètes (partie phylogénétique), nous proposons **un modèle de développement cellulaire fondé sur l'unification des approches grammaticales et celles basées sur la chimie cellulaire**. Nous utilisons ces deux approches afin de simuler dans un seul modèle de développement les mécanismes suivants :

- La diffusion de substrat, les réactions chimiques, le métabolisme, la division l'apoptose, etc. (approche chimique),
- Mécanisme de lignée cellulaire (approche grammaticale).

Pour simuler le développement de nos créatures multicellulaires artificielles, nous simulons leurs **métabolismes**, leurs **morphogénèses** et leur **évolution**.

Une fois un organisme multicellulaire créé avec ce modèle, un mécanisme évolutionniste tel que celui de Nicolas Lassabe dans sa thèse [Las08] pourra être mis en place pour faire émerger des créatures avec des capacités de locomotion. La figure 4.1 illustre l'ensemble de notre contribution.

4.1.3 Vue d'ensemble.

Pour simuler le *développement* de nos organismes multicellulaires, nous simulons leurs métabolismes leur morphogénèse et leur évolution. Pour cela Notre modèle contient un environnement avec une **chimie artificielle simplifiée** [DZB01, RCNA03, Hut07] et des cellules qui possèdent un état interne et qui peuvent interagir entre elles ainsi qu'avec l'environnement via un **système de sélection d'actions**. Pour simuler la morphogénèse des cellules évoluant dans cet environnement, nous proposons une **variante des L-systèmes paramétriques**.

Enfin, pour simuler l'évolution afin de trouver l'organisme multicellulaire le plus adapté à un problème spécifique, nous utilisons un algorithme génétique qui agit d'un côté sur un **génotype** se composant de l'ensemble de règles de croissance, les règles d'interactions et les règles métaboliques. D'un autre côté, l'algorithme génétique agit par un opérateur de sélection sur la structure phénotypique (**phénotype**) qui représente l'organisme développé.

La figure 4.2 schématise le processus de notre modèle de développement. Le tableau 4.1 fait la relation entre chaque mécanisme du développement que nous voulons simuler et le mécanisme ou l'outil informatique que nous utilisons pour le faire.

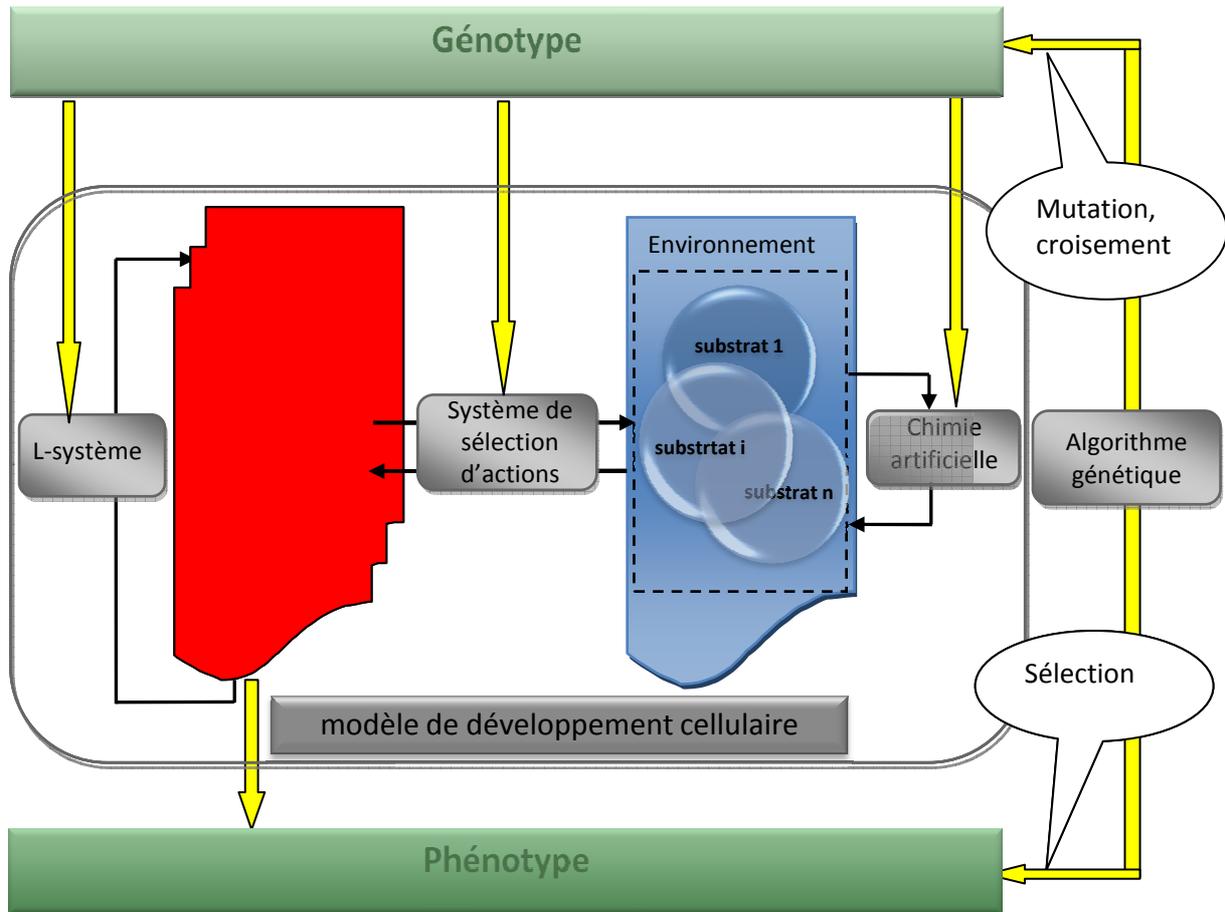


Figure 4.2 - Le modèle de développement cellulaire : vue d'ensemble.

Mécanisme simulé	Outil informatique
Métabolisme	Chimie artificielle
Interactions cellulaires et environnementales	Système de sélection d'action
Morphogenèse	Variante de L-système paramétrique
Evolution	Algorithme génétique

Tableau 4.1- Relation entre le mécanisme du développement simulé et l'outil informatique utilisé pour le faire.

4.2 Environnement embryonnaire.

Nos créatures multicellulaires artificielles évoluent dans un environnement embryonnaire représenté par une grille torique bidimensionnelle similaire à celle présentée dans [CLD08, CLD09].

D'un côté, Nous avons choisi un espace bidimensionnelle pour réduire la complexité de la simulation et donc le temps de calcul, tout en gardant suffisamment de liberté dans le modèle.

D'un autre côté, une topologie toroïdale permet, du point de vue des simulations informatiques, d'imaginer un espace infini. En outre, et à l'image des embryons naturels, elle permet de simuler un espace fermé et continu dans lequel les substances chimiques peuvent circuler continuellement.

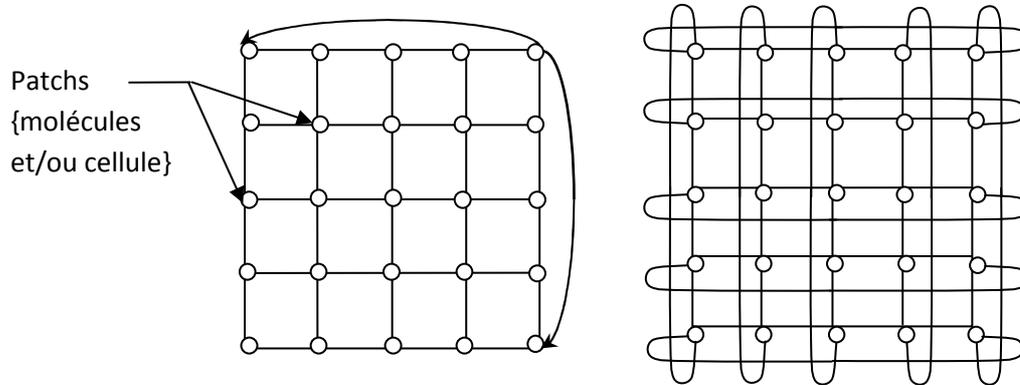


Figure 4.3 - La grille torique bidimensionnelle.

Une grille torique est obtenue en connectant les bords opposés d'une grille simple (voir figure 4.3). Dans notre modèle, une grille $[n][m]$ est une matrice de patch où chaque patch peut contenir une ou plusieurs molécules et/ ou une seule cellule.

L'environnement peut contenir différentes molécules qui se diffusent dans la grille et des cellules qui accomplissent différentes tâches.

4.2.1 Diffusion.

Le but de la diffusion est d'équilibrer les quantités de molécules dans l'environnement en minimisant les variations des quantités de molécules entre deux points adjacents de la grille.

La répartition initiale de ces molécules se fait d'une manière non uniforme et chaque molécule peut se diffuser vers les huit points voisins dans la grille. Cette diffusion se fait en deux étapes comme illustré sur la figure 4.4 :

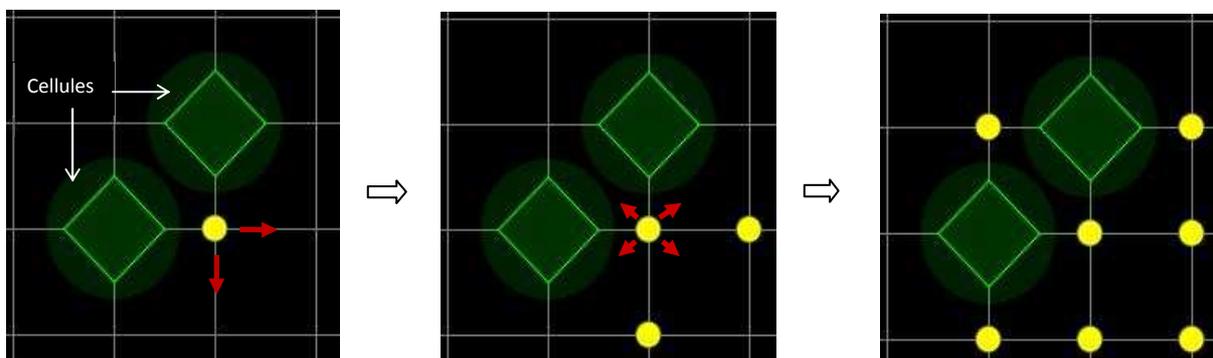


Figure 4.4 - La diffusion de molécules.

- Premièrement, la molécule se diffuse vers les quatre points cardinaux.
- Ensuite, si la quantité de molécules est suffisante, la molécule se diffuse sur les diagonales.

Si la quantité de molécules n'est pas suffisante pour se diffuser de manière égale, les dernières diffusions de l'étape de diffusion sont choisies aléatoirement. De plus, si un point voisin du point de diffusion contient plus de molécules, la diffusion ne se fera pas dans cette direction étant donné que le but de celle-ci est de minimiser les variations de molécules entre les points voisins de l'environnement. Enfin, aucune diffusion ne se fait à travers une cellule, que ce soit de l'environnement vers l'intérieur de la cellule ou le contraire. Seule une action déclenchée par la cellule peut permettre un échange de molécules avec l'environnement.

4.2.2 Cellules.

Les cellules évoluent dans cet environnement et plus précisément dans la grille de diffusion de celui-ci. Chacune de ces cellules possède :

- une forme (losange, cercle...),
- un état interne qui représente sa constitution en molécules cytoplasmiques ainsi que son niveau d'énergie. Cet état interne est codé à l'aide d'un vecteur de valeurs réelles (voir figure 4.5),
- un certain nombre de capteurs positionnés sur sa membrane (décrit dans la section 4.4.1),
- Un système de sélection d'actions permettant à la cellule de choisir la meilleure action à effectuer à chaque instant de la simulation (décrit dans la section 4.4.2),
- une liste de capacités, autrement dit, une liste d'actions, (décrite dans la section 4.4.3)

Etat interne									
43	101	0	8	95	220	43	76	11	1024
Molécule ₁	Molécule ₂			...				Molécule _n	Energie vitale

Figure 4.5 - L'état interne d'une cellule artificielle.

4.3 Métabolisme.

Afin de simuler le métabolisme de nos créatures multicellulaires artificielles, notre modèle intègre une chimie artificielle simplifiée. Nous définissons dans cette section la chimie artificielle, nous donnons quelques motivations, puis nous présentons la chimie artificielle proposée.

4.3.1 Chimie artificielle.

La chimie est la science de la nature qui étudie la composition de la matière et ses modifications. La chimie artificielle (*artificial chemistry*) quant à elle, est un système créé par l'homme et qui est similaire au système chimique réel [DZB01].

En faisant abstraction sur les processus moléculaires naturels, la chimie artificielle est une sous discipline de la vie artificielle qui tente d'investiguer les dynamiques de ces systèmes complexes.

Formellement, une chimie artificielle est définie par le triplet (S, R, A) [DZB01] où :

- $S = \{s_1, \dots, s_i, \dots, s_n\}$: la population, représentant l'ensemble de toutes les molécules valides qui peuvent apparaître dans le modèle.
- R : représente l'ensemble des règles de réaction (règles de collision) décrivant les interactions entre les molécules $s_i \in S$.
- A : le réacteur, est un algorithme qui décrit les dynamiques de la population en déterminant comment l'ensemble R des règles est appliqué à une collection de molécules.

Une chimie artificielle est un milieu très convenable dans lequel l'évolution artificielle peut être explorée [Hut07]. Par ailleurs, différents travaux existent sur ce sujet et permettent, pour la plupart d'entre eux, une bonne simulation des mécanismes biologiques [DZB01, RCNA03, Hut07]. Dans ces travaux, la chimie artificielle est fortement développée et permet une bonne simulation des mécanismes cellulaires. A titre d'exemple, Ono [OI99] a développé un modèle qui permet une simulation réaliste de la division cellulaire et la formation de la membrane. Cependant, la complexité d'un tel modèle empêche le développement d'un nombre important de cellules.

4.3.2 Chimie artificielle proposée.

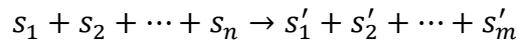
Molécules.

Dans notre modèle, nous définissons un système chimique simple qui se compose d'un nombre fini de molécules que nous appelons également *substrats*. Ces substrats se diffusent dans la grille et possèdent un nom unique (A, B, C, \dots) ainsi que quelques propriétés comme la vitesse de diffusion, la couleur et le type (cytoplasmique ou environnemental). Cet ensemble de substrats correspond à l'ensemble S de la chimie artificielle décrite par Dietrich.

Réactions.

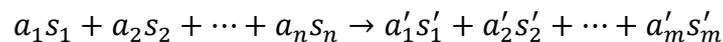
Les substrats peuvent interagir entre eux via un ensemble fini de règles d'interaction. Chaque règle d'interaction correspond à la définition formelle des réactions de l'ensemble R

du triplet présenté ci-dessus. Elles sont souvent présentées par des formules de la forme suivante :



Dans cette formule, s_i et s'_j ($i \in 1 \dots n, j \in 1 \dots m$) sont des substrats de l'ensemble S . Les s_i dans la partie gauche de l'équation correspondent aux réactifs. Les s'_i à droite de l'équation correspondent aux produits résultant de l'interaction des réactifs. Une réaction ne peut être déclenchée que si les réactifs sont présents en même temps et au même endroit.

Comme dans la nature, la quantité de chaque molécule qui sera consommée doit être prise en compte. On obtient alors la formule pour une réaction :



Les coefficients $a_i \in N$ et $a'_j \in N$ ($i \in 1 \dots n, j \in 1 \dots m$) correspondent à la quantité des molécules intervenants dans la réaction.

Algorithme A:

Molécules :

$$S = \{\alpha_1, \alpha_2, \dots\};$$

Réactions :

$$R = \{r_1, r_2, \dots\}$$

Début

Tant que (\neg terminé())

$n :=$ nombre – aléatoire ($[S]$)

$s_1 :=$ molécule – aléatoire (S);

$s_2 :=$ molécule – aléatoire (S);

$s_n :=$ molécule – aléatoire (S);

si ($\exists (r = a_1s_1 + a_2s_2 + \dots + a_ns_n \rightarrow a'_1s'_1 + a'_2s'_2 + \dots + a'_ms'_m (\delta_{energy})) \in R$)

et peut – être – déclenchée($r = \text{vrai}$) **alors**

exécuter r ;

fin tant que

fin

Dynamiques des réactions.

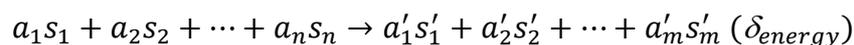
Pour simuler la dynamique d'une population de molécules, c'est à dire l'algorithme A, nous avons employé une approche de collision moléculaire stochastique. Cet algorithme tire

aléatoirement un sous-ensemble de substrats présents dans la population S et vérifie si une règle $r \in \mathcal{R}$ peut être appliquée. Dans le cas positif, les substrats de la partie gauche de la règle r (réactifs) seront remplacés par les substrats donnés par la partie droite de cette règle (produits). Dans le cas contraire, aucune règle ne peut être appliquée et le processus est réitéré.

Notons, que ces règles de réaction ne peuvent être déclenchées qu'à l'intérieur des cellules. C'est à dire seule la cellule est capable d'effectuer la transformation chimique. Cela permet de réduire la complexité en limitant la réaction au seul environnement intracellulaire.

En effet, la cellule ne pourra déclencher la réaction chimique que si et seulement si tous les composés sont réunis à l'intérieur de sa membrane. La réaction à exécuter est déterminée par l'algorithme A. Ces réactions définissent la quantité d'énergie qu'elles produisent ou consomment. Cette énergie correspond à l'énergie vitale que récupérera ou dépensera la cellule lors du déclenchement de la réaction chimique.

On obtient finalement la formulation suivante d'une réaction chimique :



Avec $\delta_{energy} \in \mathcal{R}$, la quantité d'énergie produite (si sa valeur est positive) ou consommée (si sa valeur est négative) par la réaction.

Par exemple la transformation $A + 2B \rightarrow C(-62)$ produit une unité d'un substrat C à l'aide d'une unité du substrat A et deux unités du substrat B. Cette transformation consomme 62 unités de l'énergie vitale.

Les substrats produits peuvent être rejetés par la cellule dans l'environnement et ainsi une autre cellule peut les absorber pour les transformer en d'autres substrats. Cependant, il y a des substrats qui ne sont que cytoplasmiques, c.à.d. qui ne peuvent pas être présents dans l'environnement et par conséquent la cellule ne peut ni les absorber ni les rejeter dans l'environnement, mais doit par contre les produire en utilisant d'autres substrats. Les substrats cytoplasmiques peuvent être utilisés par la cellule pour exécuter des actions spécifiques comme la division cellulaire, les autres servent à la production des premiers ainsi que pour l'interaction intercellulaire indirecte.

4.4 Les interactions cellulaires et environnementales.

Afin que les cellules puissent interagir avec l'environnement ainsi qu'avec d'autres cellules, nous avons imaginé un système de sélection d'actions leur permettant de choisir la meilleure action à effectuer à chaque moment de la simulation. Pour cela, les cellules sont munies de capteurs membranaires et peuvent accomplir différentes actions cellulaires.

4.4.1 Capteurs membranueux.

Les capteurs, représentés par des cercles (figure 4.7), sont positionnés sur la membrane de chaque cellule. Ils permettent de mesurer la quantité de substrat disponible dans son voisinage de Von Neumann (voir figure 4.6). Ainsi, pour chaque substrat environnemental possible (pentagone régulier), la cellule possède un capteur associé. Seul le capteur correspondant à un substrat donné peut mesurer sa densité.

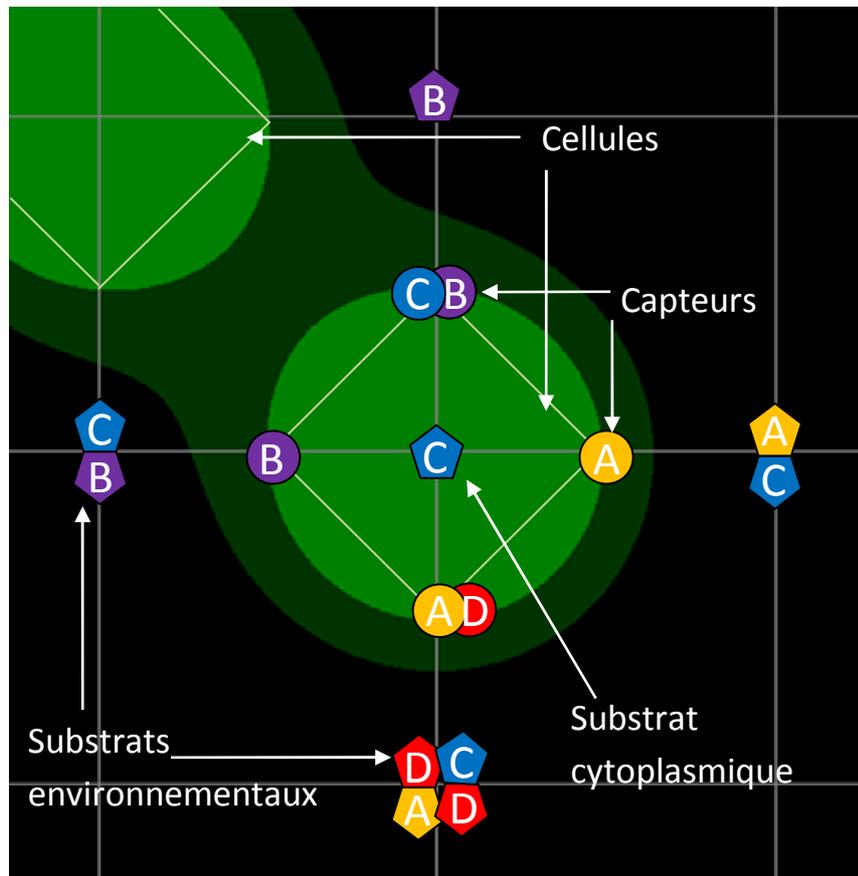


Figure 4.6- La cellule artificielle dans son environnement. Elle contient des capteurs (cercles) pour mesurer la densité de substrats (pentagones réguliers) valable dans son voisinage.

Par exemple, dans la figure 4.7 la cellule possède dans le coin bas un capteur pour le substrat *A* et un pour le substrat *D*. Les résultats des mesures de substrats sont :

- une unité de substrat *A* ;
- et deux unités de substrat *D* car deux unités de substrat *D* sont présents en bas de la cellule.

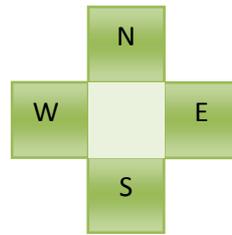


Figure 4.7- Positionnement des capteurs de la cellule : voisinage de Von Neumann.

4.4.2 Système de sélection d'action.

Les cellules contiennent également, un système de sélection d'actions qui leur permet de sélectionner la meilleure action à exécuter à chaque moment de la simulation.

L'ensemble des systèmes de sélection d'action de chaque créature est basé sur un système de classeurs de type Pittsburgh [Smi80]. Comme nous l'avons introduit en section 2.4.2, ce système de classeurs se différencie des systèmes de classeurs de type Michigan (XCS, ZCS, ACS...etc.) par le fait qu'il utilise pour son évolution ainsi que pour son évaluation un algorithme génétique. Historiquement, l'approche de type Michigan s'est rapidement imposée comme le cadre standard des systèmes de classeurs. Cependant, on assiste à l'heure actuelle à un renouveau des approches de type Pittsburgh [LG02, BGB07, LSS05, LS08].

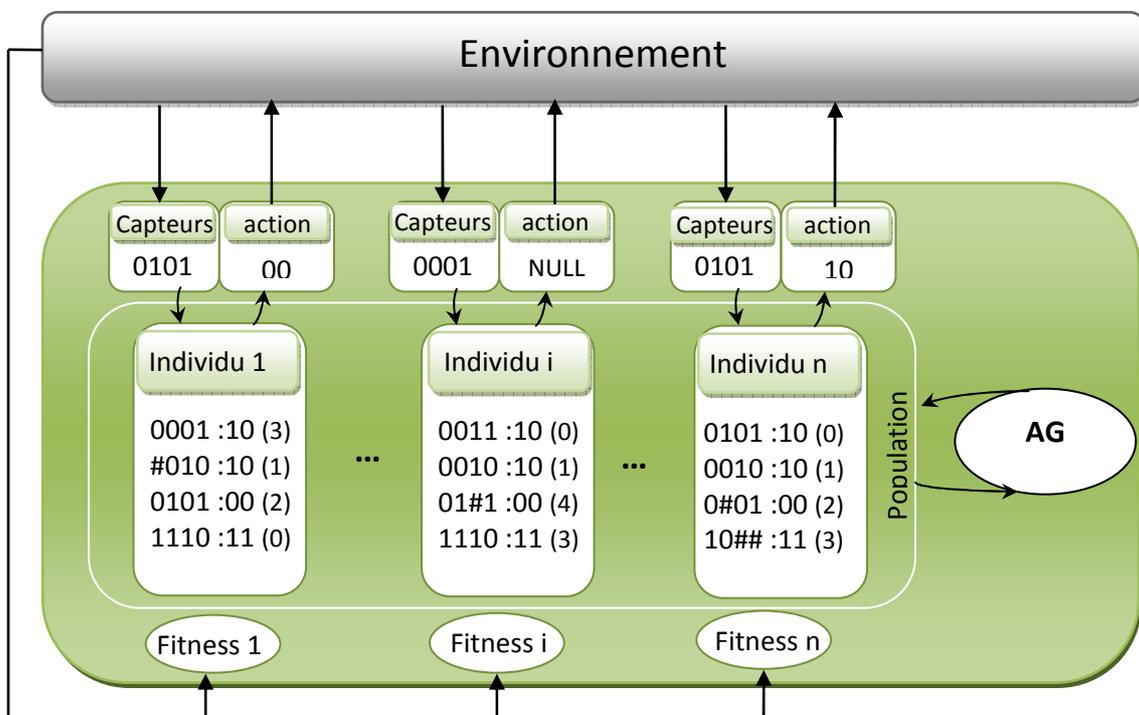


Figure 4.8 - L'ensemble de systèmes de sélection d'action de chaque créature définis sur un alphabet ternaire {0,1,#}.

Inspiré de ce dernier type de système de classeurs, notre système de sélection d'actions n'utilise donc pas un apprentissage par renforcement. En effet, le seul mécanisme adaptatif que nous avons mis en œuvre est un algorithme génétique différé qui s'applique à une population où chaque individu est un système de classeurs tentant de résoudre le problème posé. La figure 4.8 représente la structure de ce système.

Le système de sélection d'action de chaque créature fonctionne de la façon suivante : il utilise les données des capteurs de la cellule et les apparie avec la partie condition de chaque classeur pour produire l'ensemble des classeurs déclenchables. Une fois fait, il sélectionne la meilleure action à exécuter à partir de cet ensemble en se basant sur la plus grande priorité.

Classeurs.

Dans notre système de sélection d'actions, chaque classeur ou règle dispose de trois parties : condition, action et priorité :

- La condition sert à capter le signal provenant de l'environnement ou de l'état interne de la cellule. Elle constitue l'ensemble des capteurs de la cellule : les capteurs membranaires et les capteurs d'état interne.
- La partie action détermine l'action qui doit être exécutée par la cellule si la condition correspondante se trouve vérifiée.
- La priorité permet de choisir une seule action dans le cas où plusieurs classeurs sont actifs en même temps.

Cette priorité est représentée par un nombre entier. Plus ce nombre est grand, plus l'action aura de chance d'être sélectionnée. Étant donné que ce nombre est généré aléatoirement pour chaque classeur dans la phase d'initialisation, si deux actions peuvent être déclenchées en même temps, et possèdent par hasard la même priorité, la première dans la liste sera choisie.

Un exemple d'un système de sélection d'actions pourrait être :

(Capteur A = 3) and (capteur B = 1)

and (0 < CapteurC ≤ 5) → (action1) (52)

(CapteurB = 8) → (action2) (67)

→ (Action3) (15)

Dans cet exemple, l'action 1 ne peut être déclenchée que si la valeur du capteur associé au substrat A est égale à 3 unités, la valeur du capteur associé au substrat B est égale à une

unité et la valeur du capteur associé au substrat C et est supérieure strictement à 0 et inférieure ou égale à 5. L'action 2 sera déclenchée si la valeur du capteur associé au substrat B est égale à deux unités. Enfin, l'action 3 pourra toujours être déclenchée car elle n'a pas de conditions. Si plus d'une action est déclenchable à un instant t de la simulation, la priorité des règles ordonnera les actions dans l'ordre :

$$\text{Action2} > \text{Action1} > \text{action3}.$$

4.4.3 Actions cellulaires.

Les actions cellulaires, que nous simulons dans notre modèle, sont les suivantes :

L'absorption ou le rejet d'un substrat : Ces actions peuvent déclencher (respectivement, être déclenchées par) une action de transformation de substrats. Elles consomment de l'énergie vitale.

La transformation des substrats : cette action permet de déclencher une collision stochastique de deux substrats à l'intérieur de la cellule. Cette collision catalysera la formation d'un nouveau substrat via une règle de réaction. Ainsi, les substrats qui sont dans la partie gauche de la règle de réaction sont détruits pour donner naissance à un troisième substrat en produisant ou en consommant de l'énergie.

La survie : cette action permet à la cellule d'attendre un signal de l'environnement ou de son état interne. Elle consomme de l'énergie vitale.

L'apoptose : cette action permet à la cellule de se suicider si elle n'a pas assez d'énergie vitale pour survivre. Quand une cellule meurt, toute sa constitution en substrats revient à l'environnement.

4.5 Morphogenèse.

La croissance d'une créature multicellulaire artificielle se fait, dans notre modèle, par l'intermédiaire de la division cellulaire. Cette dernière et son orientation sont spécifiées par un modèle de croissance basé sur des règles de production codées dans un DOL-système (Determinist Context free L-System) [PL90].

Ces systèmes introduits en section 2.2.3, ont été introduits par Lindenmayer en 1968 [Lin68] comme un langage formel qui permettait de décrire le processus de développement et de prolifération d'organismes multicellulaires.

The development of an organism may [...] be considered as the execution of a 'developmental program' present in the fertilized egg. The cellularity of higher

organisms and their common DNA components force us to consider developing organisms as dynamic collections of appropriately programmed finite automata. A central task of developmental biology is to discover the underlying algorithm for the course of development.

Aristid Lindenmayer and Grzegorz Rozenberg [LR75]

À cette époque, il travaillait sur les levures, les champignons, les algues et les bactéries. Après c'est l'informatique qui a permis d'exploiter graphiquement ce type de systèmes. L'interprétation géométrique des L-systèmes a été initialement proposée par Szilard et Quinton [SQ79]. Après, Prusinkiewicz [Pru86] a proposé l'interprétation de tortue et a présenté des exemples de modèles de plantes produites pour l'infographie. Ensuite, il a étendu son modèle pour inclure les accolades (*bracketed L-system*) puis les modèles tridimensionnels [Pru87].

4.5.1 Développement artificiel dans les L-systèmes.

Le développement dans les L-systèmes est simulé par un processus de réécriture: une règle de réécriture ou de production est appliquée à une lettre, ayant pour résultat le remplacement de cette lettre par une nouvelle lettre si l'état du module doit être changé, ou par un groupe de lettre si le module se divise.

Les mécanismes de lignée sont modélisés en considérant seulement l'état courant d'un module au moment de la sélection de la règle à appliquer. L'application des règles se fait en parallèle au lieu qu'elle le soit séquentiellement. Cette différence reflète la motivation biologique des L-systèmes qui est de décrire la morphogenèse comme un processus émergeant du fonctionnement parallèle des unités qui constituent un organisme. D'un autre côté, beaucoup de divisions cellulaires peuvent apparaître en même temps dans l'organisme.

Ainsi, les L-systèmes permettent de contrôler le développement par **mécanismes de lignée cellulaire** et cela en passant, d'une manière discrète, l'information génétique du module parent (cellule mère) au module fils (cellule fille).

4.5.2 L-systèmes comme modèle de croissance.

Nous avons choisi les L-systèmes comme modèle de croissance par le fait que ces derniers constituent une représentation génétique très adéquate pour les systèmes qui ont pour but de simuler l'évolution morphologique naturelle et ceci pour les raisons suivantes:

- Ils permettent une nécessaire et très convenable distinction entre le génotype et le phénotype ;
- Ils fournissent un processus (appelé morphogenèse) bien défini pour générer le second à partir du premier ;

- La récursivité et l'autosimilarité, qui sont des concepts intrinsèques aux L-systèmes, sont des ingrédients nécessaires pour la morphogenèse naturelle ;
- Les L-systèmes sont très appropriés pour décrire la lignée cellulaire ainsi que le contrôle génétique de la division cellulaire [FB94].

En outre, les L-systèmes ont été appliqués avec succès dans plusieurs domaines : la modélisation des plantes et de fleurs virtuelles [MP96, Nos03, BL08], la modélisation de la rétine [KTV99], la prédiction de structures de protéines [EOK05], la conception évolutionniste des réseaux de neurones [Kit90, Gru93, BKHS93] et également dans la génération des morphologies et des contrôleurs de créatures artificielles ou réelles (robots) [HP01b, HP02, PHLF03]. Pour cela, et pour les raisons évoquées ci-dessus, nous voulons également explorer le potentiel des L-systèmes en les utilisant pour générer et guider la morphologie de nos organisme multicellulaires.

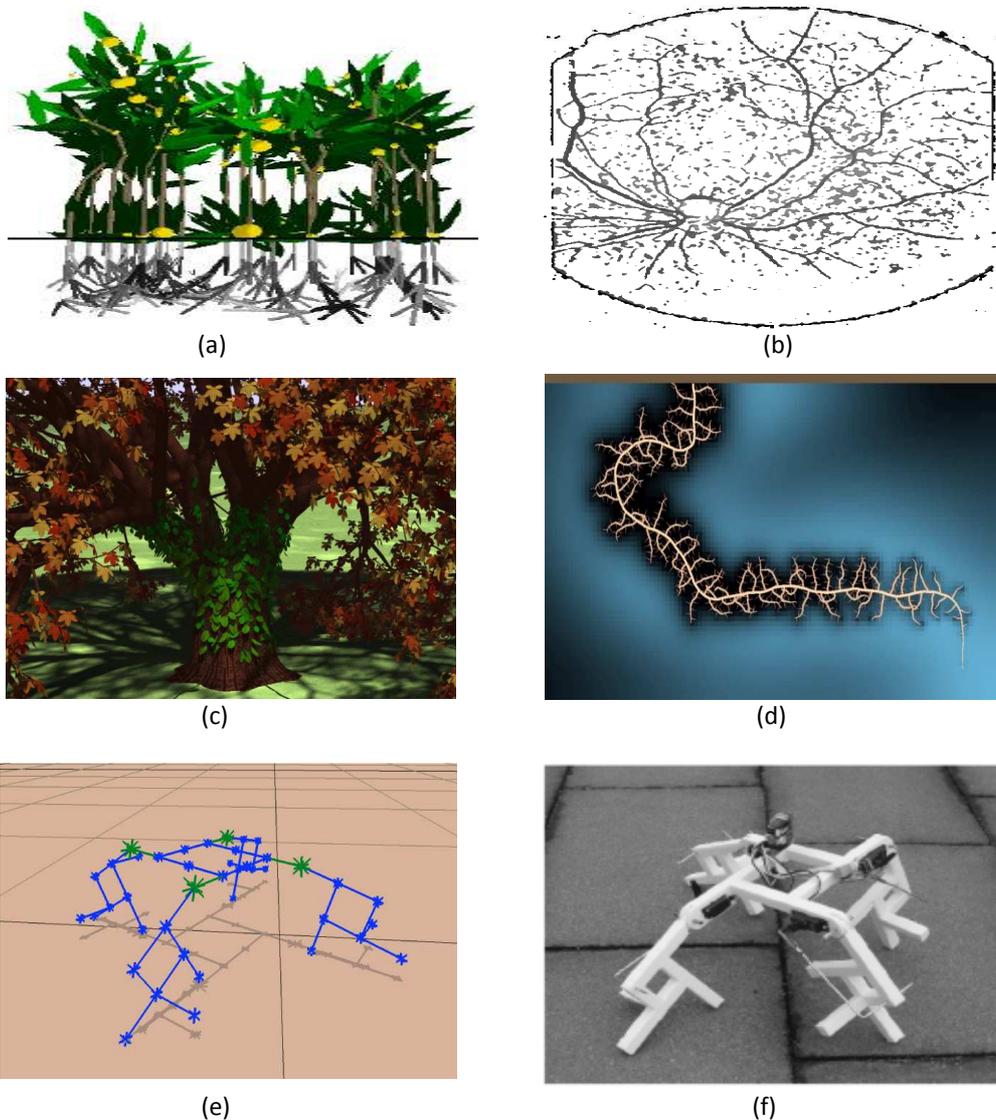


Figure 4.9 - Panel de systèmes réalisé par L-système. (a, c, d) Développement de plantes virtuelles [BL08], [MP9 respectivement]. (b) Modélisation de la rétine [KTV99]. (e, f) Conception évolutionniste de créatures artificielles dans le simulateur, sous forme de robots.

4.5.3 Modèle de croissance proposé.

Dans la version originale des L-systèmes, un organisme multicellulaire est créé par une grammaire formelle. Chaque cellule est représentée par un symbole. A chaque génération, les cellules se divisent et le symbole correspondant est remplacé par un ou plusieurs autres symboles en utilisant les règles de réécritures.

Dans notre modèle, le L-système est utilisé pour orienter le développement de nos organismes multicellulaires. En effet, au lieu d'utiliser un symbole pour représenter une cellule, nous utilisons un symbole pour représenter l'une des actions cellulaires qui est la division. En outre, le L-système que nous utilisons représente une variante de L-systèmes paramétriques que nous proposons pour simuler le développement des organismes multicellulaires en fonction des cellules les composant. Nous présentons, dans ce qui suit l'alphabet que nous utilisons. Nous présentons, par la suite, l'action de division cellulaire puis l'apport de l'extension paramétrique pour notre modèle de croissance. Enfin, nous décrivons la variante de L-systèmes paramétriques que nous proposons ainsi que ses possibles utilisations dans d'autres systèmes.

Alphabet du L-système.

L'alphabet que nous utilisons est décrit dans le tableau ci-dessous, où les caractères majuscules constituent les symboles non-terminaux et dénotent les prédécesseurs des règles tandis que les caractères minuscules représentent les symboles terminaux : "d" est un symbole permettant à la cellule concernée d'exécuter l'action de division cellulaire, les symboles "+" et "-" représentent des rotations bidimensionnelles utilisées pour déterminer la direction suivant laquelle la cellule mère pourra positionner sa cellule fille. Enfin, les symboles "[", "]" permettent d'empiler et de dépiler l'état courant.

Symboles	Fonctions
A ...Z	Non terminaux représentant les prédécesseurs de règles.
d	Terminal déterminant une action de division.
+, -	Terminaux représentant une rotation bidimensionnelle.
[,]	Terminaux indiquant une ramification.

Tableau 4.2- Alphabet du L-système.

Division cellulaire.

Quand une cellule se divise, elle crée une nouvelle cellule fille dans la direction spécifiée par le L-système contrôlant la morphogénèse de la créature. L'action de division ne peut être exécutée par la cellule que si les conditions suivantes sont vérifiées :

- La cellule doit avoir assez d'énergie vitale pour se diviser ;

- Les quantités de certains substrats nécessaires pour la création d’une nouvelle cellule doivent dépasser un certain seuil ;
- L’orientation de la division correspond à une position non occupée par une autre cellule.

Durant la division, les substrats cytoplasmiques ainsi que l’énergie vitale sont partagés équitablement entre la cellule mère et sa fille.

4.5.4 Apport de l’extension paramétrique.

Les L-systèmes paramétriques introduits en section 2.2.3 prolongent le concept original des L-systèmes en associant des paramètres numériques aux symboles. La définition du formalisme est donc étendue au quadruplet $G = (V, \Sigma, \omega, P)$ où Σ représente l’ensemble des paramètres formels. Ceci permet d’une part, une quantification facile des attributs géométriques du modèle, et offre la possibilité de transmettre plus d’information par symbole et de pouvoir spécifier davantage de conditions pour le déclenchement d’une règle.

L’extension paramétrique fournit également des moyens simples pour l’expression des processus continus à l’intérieur des organismes, tels que : la diffusion des hormones et la distribution des concentrations résultantes [Han92], le transfert des ressources dans une plante du sol à la partie souterraine d’une plante virtuelle [Bor08], etc.

Nous utilisons l’extension paramétrique pour simuler le développement des organismes multicellulaires en fonction des cellules le composant.

4.5.5 APL-système: Variante de L-système paramétrique.

Pour modéliser la croissance de nos organismes multicellulaires artificiels nous proposons une variante des L-systèmes paramétriques que nous baptisons *APL-systèmes* (pour *Alternative Parametric L-system*). Cette extension aux L-systèmes basiques permet de modéliser le développement d’organismes multicellulaires en fonction de l’état interne de chaque cellule. Cependant, à la différence des L-systèmes paramétriques classiques, dans notre variante, toutes les règles de production du L-système peuvent être déclenchées. Cela signifie que les prédécesseurs des règles n’ont pas de pré-conditions. Cependant, les symboles terminaux qui composent les chaînes successeurs des règles possèdent des pré-conditions. Le schéma de la figure 4.10 illustre la différence entre une règle paramétrique classique et la nouvelle règle paramétrique définie dans notre APL-système.

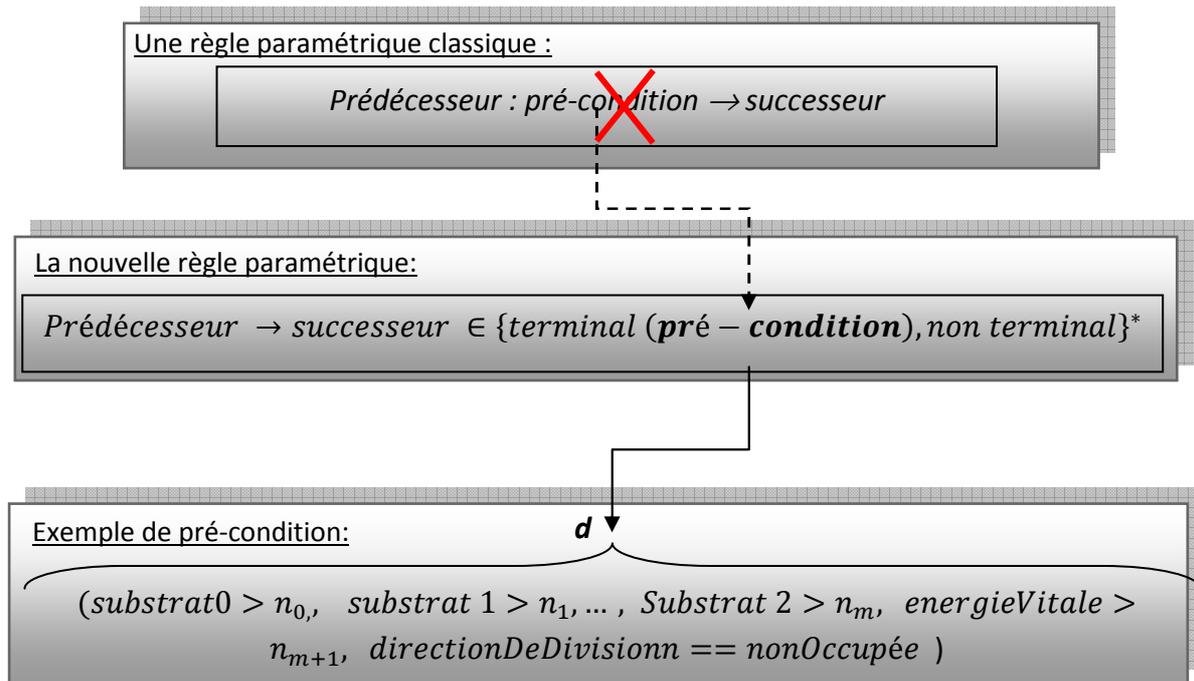


Figure 4.10 - Représentation des règles dans l'APL-système.

Définition.

Nous définissons formellement un APL-système par le quadruplet (V, Σ, ω, P) où :

- $V = V1 \cup V2$, l'alphabet du L-système, avec $V1$ et l'ensemble des non terminaux et $V2$ est l'ensemble des terminaux ;
- Σ , un ensemble de paramètres formels ;
- $\omega \in V$ un axiome ;
- $P \subset V \times (V1 \times (V2 \times C(\Sigma)))^*$, un ensemble fini de règles de reproduction. V est l'ensemble des prédécesseurs de la production, et $V1 \times (V2 \times C(\Sigma))$ est l'ensemble successeurs de la production où $C(\Sigma)$ est l'ensemble des conditions.

Ainsi, l'alphabet V et l'ensemble des paramètres formels Σ sont définis comme dans un L-système paramétrique classique (section 2.2.3). Néanmoins, l'axiome ω et l'ensemble des productions P sont redéfinis. L'axiome est un mot non paramétré et les règles de production ne possèdent pas de pré-conditions mais ce sont les symboles terminaux qui possèdent des pré-conditions.

Une pré-condition $C(\Sigma)$ attachée à un terminal signifie que celui-ci ne peut être interprété que si sa pré-condition est vérifiée. Par exemple, le non terminal "d" ne peut être interprété en une action de division cellulaire que si les conditions de la division cellulaire sont vérifiées par la cellule concernée.

De cette manière, cette variante va permettre d'interpréter toutes les sous-chaines produites par l'application d'une règle dans une génération donnée jusqu'au dernier terminal qui ne vérifie pas sa pré-condition, alors que dans les L-systèmes paramétriques traditionnels toutes les sous-chaines générées par cette règle ne se seraient pas générées car la règle ne se serait pas déclenchée. De cette manière, toutes les cellules qui veulent se diviser dans cette génération pourront le faire si elles remplissent les pré-conditions.

Exemple.

Nous considérons l'exemple de l'APL-système (V, Σ, ω, P) où :

- $V = \{A, B, d, +, -, [,]\}$,
- $\Sigma = \{s_1, s_2, s_3, e\}$,
- $\omega = A$,
- $P = \{A \rightarrow -d(e > 4)d(s_1 > 3, s_2 \geq 1, s_3 > 0, e > 2) [B]A,$
 $B \rightarrow +d(s_1 > 1, s_2 \geq 5, e \geq 2)d(s_3 > 2, e > 1)\}$

Dans cet exemple, nous avons quatre paramètres formels représentant l'état interne de chaque cellule : les trois premiers paramètres correspondent à la quantité de chaque substrat cytoplasmique (s_0, s_1, s_2) . Le quatrième paramètre correspond à la quantité de l'énergie vitale (e) . La cellule mère est initialisée avec 48 unités du substrat 0, 160 unités du substrat 1, 32 unités du substrat 2 et 60 unités d'énergie vitale $((s_0, s_1, s_3, e) = (48, 160, 32, 64))$.

La cellule mère est placée dans le patch (4,6) et l'orientation de départ est au nord de cette cellule. Si on travaille sur le voisinage de Von Neumann donc on a 4 direction ce qui donne un angle de rotation de $\delta = 360^\circ/4 = 90^\circ$. Ainsi,

- “+“ : tourner à gauche de l'angle $\delta = 90^\circ$;
- “-“ : tourner à droite de l'angle $\delta = 90^\circ$.

L'exécution de cet APL-système jusqu' à la quatrième itération donne :

Génération 0 : **A** ;

Génération 1 : $-d(e > 4)d(s_1 > 3, s_2 \geq 1, s_3 > 0, e > 2) [B]A$;

Génération 2 : $-d(e > 4)d(s_1 > 3, s_2 \geq 1, s_3 > 0, e > 2) [+d(s_1 > 1, s_2 > 5, e \geq 1)d(s_3 > 2, e > 1)] - d(e > 4)d(s_1 > 3, s_2 \geq 1, s_3 > 0, e > 2)[B]A$;

Génération 3 : $-d(e > 4)d(s_1 > 3, s_2 \geq 1, s_3 > 0, e > 2) [+d(s_1 > 1, s_2 > 5, e \geq 1)d(s_3 > 2, e > 1)] - d(e > 4)d(s_1 > 3, s_2 \geq 1, s_3 > 0, e > 2)[+d(s_1 > 1, s_2 > 5, e \geq 1)d(s_3 > 2, e > 1)] - d(e > 4)d(s_1 > 3, s_2 \geq 1, s_3 > 0, e > 2)[B]A$;

La figure 4.11 illustre le développement de l'organisme multicellulaire correspondant. Afin de se focaliser sur le fonctionnement de notre APL-système, dans cet exemple nous avons

supposé que les cellules ne possèdent pas un métabolisme, et qu'elles n'ont aucune interaction avec l'environnement.

A l'instant t_0 la cellule mère reçoit la totalité de la séquence générée à la génération 3 (le génome de croissance). Elle lit cette séquence jusqu'au premier non terminal "d". Puis, elle vérifie les pré-conditions de la division cellulaire spécifiées par ce non terminale ("d" courant). Dans le cas où les pré-conditions sont vérifiées par cette cellule, celle-ci mettra une cellule fille vers l'orientation lue dans le génome de croissance (si, bien évidemment, cette direction correspond à une position non occupée par une autre cellule). Si aucune orientation n'a été lue avant la lecture du non terminale "d", alors la cellule fille sera mise dans la direction courante. Durant la division, la quantité de substrats cytoplasmique ainsi que l'énergie vitale seront partagées d'une manière égale entre la cellule mère et sa fille. Ainsi, le processus est répété jusqu'à la fin de l'interprétation du génome de croissance. A l'instant t_7 la cellule ne peut se diviser parce qu'elle n'a pas assez de substrat 0 et d'énergie vitale pour pouvoir se diviser.

Dans le cas où toutes les cellules spécifiées par la division cellulaire peuvent se diviser, c'est à dire dans le cas où la cellule mère est initialisée avec des quantités assez suffisantes d'énergie et de substrat cytoplasmique ou dans le cas où les cellules peuvent regagner de l'énergie vitale ou produire les substrats en utilisant leurs métabolismes, l'exécution de cet APL-système jusqu'à la génération 4 donnera l'organisme de la figure 4.12 (a).

Le L-système paramétrique classique équivalent à cet exemple s'écrit comme suit :

- $V = \{A, B, d, +, -, [,]\}$;
- $\Sigma = \{s_1, s_2, s_3, e\}$;
- $\omega = A(48,160,32,64)$;
- $P = \{A(s_1, s_2, s_3, e): (s_1 > 3, s_2 \geq 1, s_3 > 0, e > 4) \rightarrow -d(s_1/2, s_2/2, s_3/2, e/2)d(s_1/2, s_2/2, s_3/2, e/2) [B]A ;$
 $B(s_1 s_2 s_3, e): (s_1 > 1, s_2 \geq 5, s_3 > 2, e \geq 2)$
 $\rightarrow +d(s_1/2, s_2/2, s_3/2, e/2)d(s_1/2, s_2/2, s_3/2, e/2)\}$

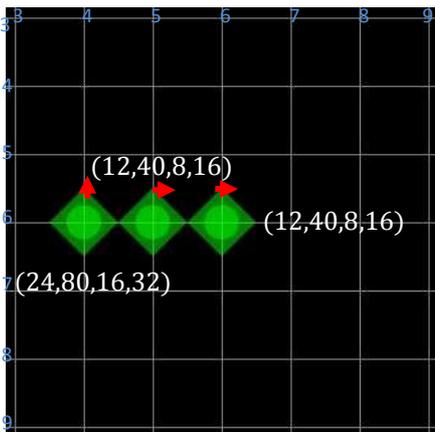
Le développement de l'organisme correspondant à l'exécution de ce L-système s'arrêtera à l'instant t_6 . A cet instant la cellule (8, 6) ne peut pas se diviser, car étant donné que cette cellule possède un état interne de (1.5,5,1,2) et le déclenchement de la deuxième règle impose *une quantité de substrat 1* $1 > 1$, *une quantité de substrat 2* ≥ 5 , *une quantité de substrat 3* ≥ 5 et *une quantité d'énergie vitale* ≥ 2 . Cette règle ne pourra être exécutée. Donc, l'exécution de ce L-système paramétrique équivalent au premier donnera l'organisme de la figure 4.12 (b).



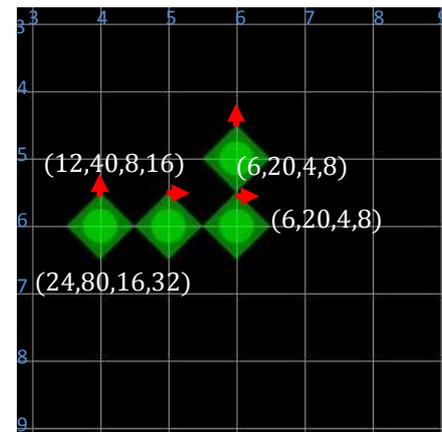
t_0 : la cellule (4,6) lit la chaîne " $-d(e > 4)d(s_1 > 3, s_2 > 1, s_3 > 0, e > 2)[+d(s_1 > 1, s_2 > 5, e \geq 1)d(s_3 > 2, e > 1)] - d(e > 4)d(s_1 > 3, s_2 \geq 1, s_3 > 0, e > 2)[+d(s_1 > 1, s_2 > 5, e \geq 1)d(s_3 > 2, e > 1)] - d(e > 4)d(s_1 > 3, s_2 \geq 1, s_3 > 0, e > 2)[B]A$ " et met une cellule fille à sa droite



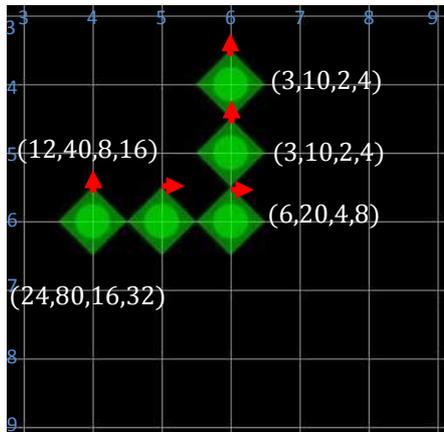
t_1 : la cellule (5,6) lit la chaîne " $d(s_1 > 3, s_2 > 1, s_3 > 0, e > 2)[+d(s_1 > 1, s_2 > 5, e > 1)d(s_3 > 2, e \geq 1)] - d(e > 4)d(s_1 > 3, s_2 \geq 1, s_3 > 0, e > 2)[+d(s_1 > 1, s_2 > 5, e \geq 1)d(s_3 > 2, e > 1)] - d(e > 4)d(s_1 > 3, s_2 \geq 1, s_3 > 0, e > 2)[B]A$ " et met une cellule fille dans la direction courante (droite)



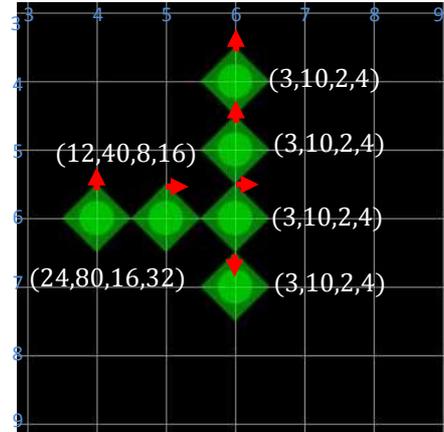
t_2 : la cellule (6,6) lit la chaîne " $+d(s_1 > 1, s_2 > 5, e \geq 1)d(s_3 > 2, e > 1)] - d(e > 4)d(s_1 > 3, s_2 \geq 1, s_3 > 0, e > 2)[+d(s_1 > 1, s_2 > 5, e \geq 1)d(s_3 > 2, e > 1)] - d(e > 4)d(s_1 > 3, s_2 \geq 1, s_3 > 0, e > 2)[B]A$ " et met une cellule fille à gauche de sa direction courante



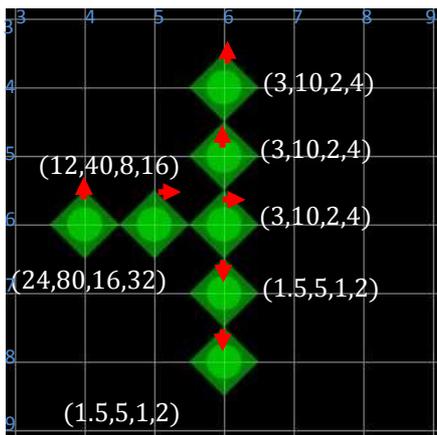
t_3 : la cellule lit la chaîne " $d(s_3 > 2, e > 1)] - d(e > 4)d(s_1 > 3, s_2 \geq 1, s_3 > 0, e > 2)[+d(s_1 > 1, s_2 > 5, e \geq 1)d(s_3 > 2, e > 1)] - d(e > 4)d(s_1 > 3, s_2 \geq 1, s_3 > 0, e > 2)[B]A$ ", met une cellule fille vers sa direction courante et dépile la position courante : cellule (6,6) (cellule à se diviser)



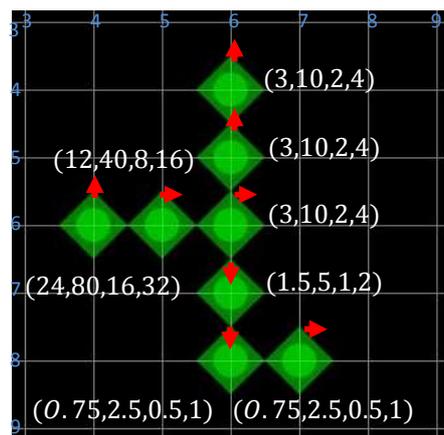
t_4 : la cellule (6,6) lit la chaîne $-d(4)d(s_1 > 3, s_2 \geq 1, s_3 > 0, e > 2)[+d(s_1 > 1, s_2 > 5, e \geq 1)d(s_3 > 2, e > 1)] - d(e > 4)d(s_1 > 3, s_2 \geq 1, s_3 > 0, e > 2)[B]A$ " et met une cellule fille à droite de direction courante



t_5 : la cellule (7,6) lit la chaîne $d(s_1 > 3, s_2 \geq 1, s_3 > 0, e > 2)[+d(s_1 > 1, s_2 > 5, e \geq 1)d(s_3 > 2, e > 1)] - d(e > 4)d(s_1 > 3, s_2 \geq 1, s_3 > 0, e > 2)[B]A$ " et met une cellule fille dans sa direction courante



t_6 : la cellule (8,6) lit la chaîne $[+d(s_1 > 1, s_2 > 5, e \geq 1)d(s_3 > 2, e > 1)] - d(e > 4)d(s_1 > 3, s_2 \geq 1, s_3 > 0, e > 2)[B]A$ " et met une cellule fille à gauche de sa direction courante



t_7 : la cellule (8,7) lit la chaîne $d(s_3 > 2, e > 1)] - d(e > 4)d(s_1 > 3, s_2 \geq 1, s_3 > 0, e > 2)[B]A$ " mais elle ne peut pas se diviser car $e \leq 1$ et $s_3 < 2$.

Figure 4.11- Exemple de développement d'une créature en utilisant APL-système. Les flèches rouges en haut de chaque cellule déterminent la direction courante. Les formules entre parenthèses se trouvant à proximité de chaque cellule déterminent l'état interne de celles-ci.

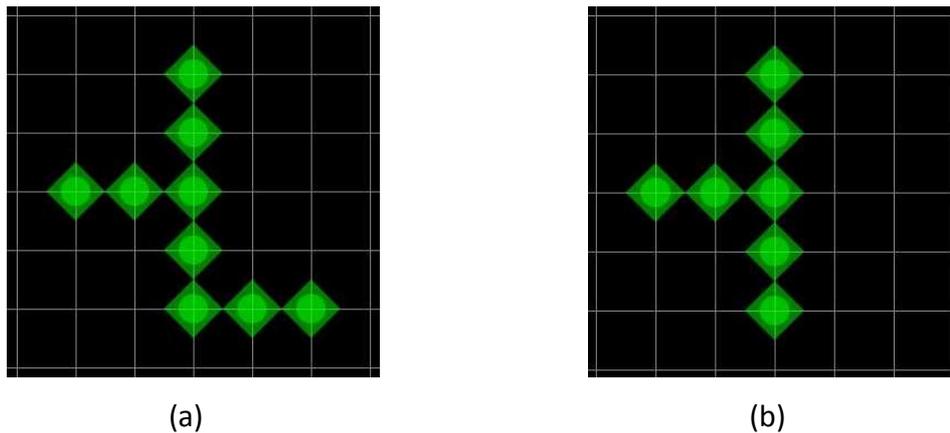


Figure 4.12 - Comparaison entre deux organismes développés par deux L-systèmes différents. (a) : Organisme produit par APL-système. (b) : Organisme produit par L-systèmes paramétriques classique équivalent.

En comparant les deux figures, on peut constater que L'APL-système permet l'expression d'un nombre plus grand de cellules.

Notons que pour faire des divisions avec des quantités de substrats non équitables entre la cellule mère et sa fille nous pourrions étendre notre APL-système en ajoutant des expressions arithmétiques aux non-terminaux. Ces expressions arithmétiques déterminent la quantité de substrat transférée à la cellule fille durant la division cellulaire.

4.6 Evolution.

Pour trouver la créature la plus adaptée à un problème spécifique, nous utilisons un algorithme génétique [Hol92]. Le fonctionnement de cet algorithme est basé comme tous les algorithmes évolutionnistes sur la boucle générationnelle de l'évolution présentée en section 2.3.2. Cependant, cet algorithme a été étendu avec l'idée que l'espace des solutions représente l'espace des organismes possibles et que le codage proposé est indirect (car nous y intégrons un processus développementale – le modèle de développement cellulaire – qui joue le rôle de médiateur entre le génotype utilisé et le phénotype développé (voir section 3.1.1)). On parle alors, d'une ontogenèse ; un modèle évolutionniste dans lequel les phénotypes subissent une phase de développement.

Cette section est ainsi dédiée à la présentation de notre modèle d'évolution et ceci en décrivant le modèle génétique de la créature, la démarche évolutionniste illustrée par une approche développementale et enfin les différents opérateurs de reproduction que nous utilisons pour explorer l'espace de recherche par diversité génétique.

4.6.1 Génotype.

Chaque créature multicellulaire artificielle est codée avec un génome qui se compose de deux chromosomes : le système de sélection d'actions, qui contient la liste des règles

permettant aux cellules de sélectionner l'action à exécuter à chaque moment de la simulation et l'APL-système ayant pour rôle de spécifier les règles de croissance.

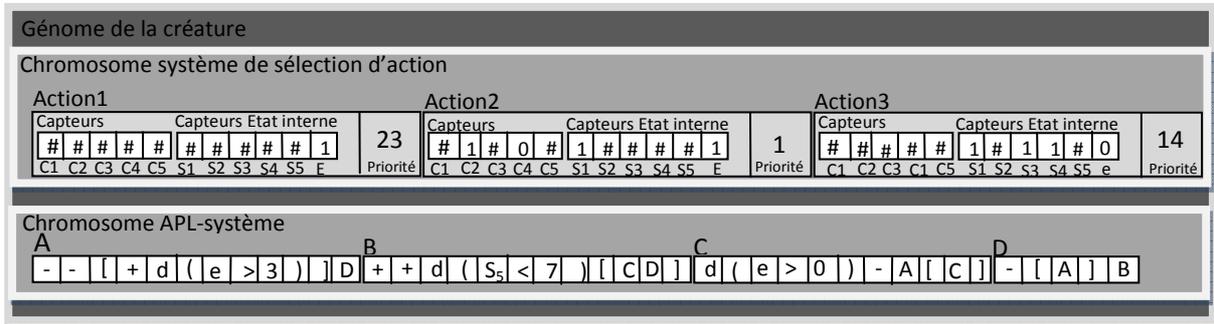


Figure 4.13 - Le génome d'une créature artificielle est composé de deux chromosomes : le sélecteur d'action et le APL-système.

Chromosome, système de sélection d'actions.

Ce chromosome est codé à l'aide d'un triplet (liste de capteurs membraneux, liste de capteurs d'état interne, priorité) pour chaque action possible dans laquelle :

- la liste de capteurs membraneux représente l'ensemble des capteurs membraneux de la cellule. Chaque capteur est défini sur un alphabet ternaire {0,1,#} où : "1" et "0" représente la présence ou l'absence respectivement du substrat environnemental courant. "#" signifie que la valeur de ce capteur n'est pas nécessaire pour l'activation de ce classeur.
- La liste de capteurs d'état interne en substrats et en énergie vitale est également définie sur un alphabet ternaire. "1" et "0" représentent la présence ou l'absence respectivement du substrat cytoplasmique courant ou de l'énergie vitale. "#" signifie que la présence ou l'absence de cet élément (substrat ou énergie) à l'intérieur de la cellule n'est pas nécessaire pour le déclenchement du classeur.
- La priorité représente la priorité de la règle courante dans l'ensemble des priorités. Elle est codée à l'aide d'un entier.

Dans le génome de la figure 4.13, nous avons un exemple de chromosome de sélection d'action. Les cellules ici peuvent exécuter trois actions Action1, Action2 et Action3 et possèdent cinq capteurs C1, C2, C2, C3, C4, C5 correspondant respectivement aux substrats S1, S2, S2, S3, S4 et S5. Ce chromosome produirait le système de sélection d'actions suivant :

$$(Energy = vrai) \rightarrow (action1) (23)$$

$$(Capteur2 = vrai) et (Capteur4 = faux)$$

$$et(Substrat1 = vrai) et (Energie = vrai) \rightarrow (action2)(1)$$

$$(Substrat1 = vrai)et(substrat3 = vrai) \\ et (substrat4 = vrai)et(Energie = faux) \rightarrow (Action3) (14)$$

Chromosome APL-système.

Ce chromosome est codé à l'aide d'une paire (prédécesseur, successeur) pour chaque règle. Dans un tel jeu de règles, le premier symbole prédécesseur est A , le second symbole B et ainsi de suite. Cette contrainte a été ajoutée dans l'optique de réduire, dans un premier temps, l'espace de recherche génétique pour une meilleure compréhension des résultats.

Ce chromosome contient également des paramètres numériques associés à chaque non terminal "d" représentant les quantités de substrat et d'énergie nécessaire pour l'interprétation de ce symbole.

Dans le génome de la figure 4.13, on a un exemple du chromosome APL-système. Ici, on a quatre règles de production. Ce chromosome produirait l'APL-système suivant :

- $V = \{A, B, d, +, -, [,]\}$;
- $\Sigma = \{s_1, s_2, s_3, e\}$;
- $P = \{A \rightarrow - - [+d(e > 3)]D ;$
 $B \rightarrow + + d(s_5 < 7)[CD] ;$
 $C \rightarrow d(e > 0) - A[C] ;$
 $D \rightarrow -[A]B\}$.

Le chromosome L-système peut contenir également des paramètres morphologiques comme la position de la cellule mère, le nombre d'itérations ou la longueur maximale d'une règle. Ainsi ces paramètres peuvent également subir un processus d'évolution.

4.6.2 Démarche évolutionniste.

L'algorithme génétique fonctionne de la façon suivante :

- 1- *construire une population de génomes composés des deux chromosomes : Le chromosome L-système et le chromosome système de sélection d'actions.*
- 2- *Tant que le critère de convergence n'est pas atteint*
 - 3- *Pour chaque génome généré*
 - 4- *Initialiser l'environnement avec certaines quantités de substrats environnementaux*
 - 5- *Lancer le processus de la diffusion*
 - 6- *Placer une cellule mère dans l'environnement et lui affecter le génome*
 - 7- *Laisser exécuter le processus développemental*
 - 8- *Evaluer le génome simulé en fonction de l'organisme développé : son phénotype*
 - 9- *Fin pour*

10- Appliquer l'opérateur de sélection sur les génomes évalués

11- Appliquer les opérateurs de reproduction (opérateurs génétiques : mutation et croisement) sur les génomes sélectionnés

12- Remplacer la population existante par la nouvelle population

13-Fin tant que

Étape 1 : Cette étape représente le point de départ de l'algorithme évolutionniste. Elle consiste à générer de façon aléatoire l'ensemble de gènes de chaque chromosome du génome. Pour le chromosome L-système, un gène est une règle de production. Pour le chromosome système de sélection d'actions, un gène est un classeur.

La taille de la population à utiliser par génération est définie d'une manière empirique mais en respectant certaines contraintes. En effet, elle ne doit pas être très grande pour des raisons de calculs, ni trop faible pour que l'algorithme soit capable de trouver une solution convenable.

Étape 2 et 13 : Comme expliqué dans 2.3.2, le critère d'arrêt de la boucle générationnelle peut être défini de plusieurs façon : arrêt après un certain temps limité, arrêt lorsque la fitness d'un individu a atteint un certain seuil fixé, arrêt de la convergence après un certain temps d'itération (fitness de la meilleur solution n'augmente plus)...etc. Dans le cadre des simulations qui seront présentées dans le chapitre qui suit, l'arrêt de l'algorithme est déterminé lorsqu'un nombre maximal de générations (itérations) est atteint. Ce nombre est déterminé de façon empirique.

Étape 3 et 9 : On simule chaque génome de la population.

Étape 4 : Spécification et initialisation de l'environnement. Les quantités de chaque substrat environnemental à diffuser dans l'environnement doivent être spécifiées avant le début de chaque simulation. Ces valeurs sont définies d'une façon empirique.

Étape 5: Ce processus s'exécute parallèlement et continuellement avec les processus de développement de chaque cellules et dès le début jusqu'à la fin de simulation.

Étape 6 : Une cellule mère est placée dans l'environnement et un génome lui est affecté. Ce génome dicte alors son développement. La cellule est placée généralement au centre de la grille, ce qui peut permettre une évaluation plus aisée de l'organisme formé. La cellule peut également être placée dans une position aléatoire dans la grille et dans ce cas là, sa position doit être précisée dans son code génétique.

Étape 7 : Le développement de l'organisme est simulé. Son comportement ou le comportement de chaque cellule le composant (division, transformation chimiques de substrats, apoptose) est le résultat de l'interprétation de son génome et en fonction de ses perceptions vis-à-vis des ressources de l'environnement. Le développement d'un organisme multicellulaire s'arrêtera dans l'un des deux cas suivant :

- Mort de toutes les cellules le composant.
- Écoulement d'un temps de simulation maximal.

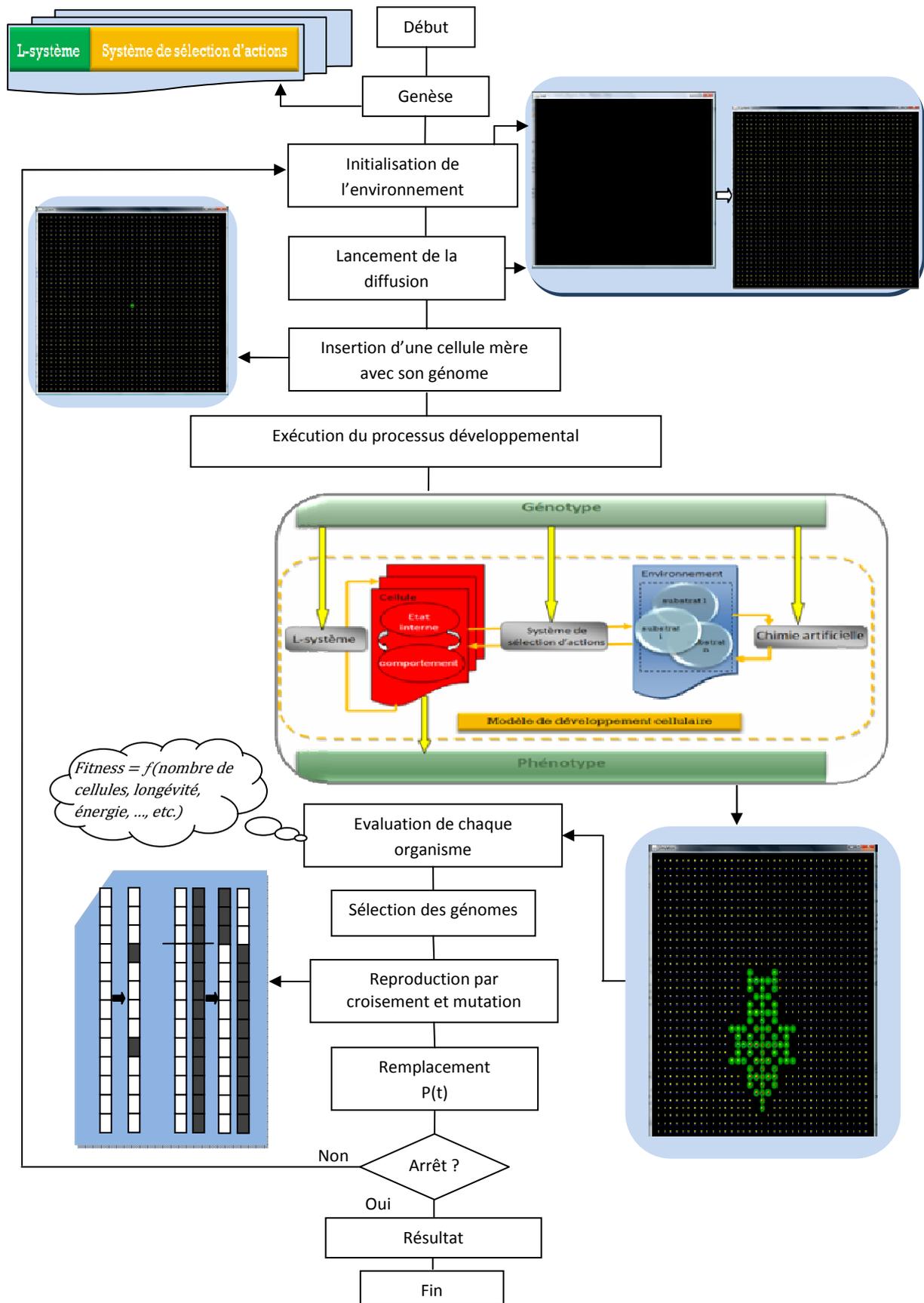


Figure 4.14 - La Boucle générationnelle de l'évolution illustrée par une approche ontogénétique.

Dans des simulations où l'expression du chromosome système de sélection d'actions est inhibée, la simulation du développement de l'organisme multicellulaire est arrêtée lorsque les cellules n'exhibent plus de comportements de division (fin d'interprétation du chromosome L-système). Dans certaines simulations mettant en œuvre les deux chromosomes, un certain temps est passé après l'arrêt des divisions afin de permettre à la diffusion des substrats dans l'environnement de se stabiliser et ainsi permettre l'expression des comportements d'apoptose.

Etape 8 : L'organisme développé est évalué. Ici, la fonction de fitness est automatisée, c'est à dire que l'évolution n'est pas interactive. Selon l'expérimentation, l'expression de l'un des deux chromosomes peut être inhibée. Comme on peut commencer avec un chromosome prédéfinie. Par exemple, on peut commencer par un chromosome L-système favorisant préalablement l'autosimilarité (trouvé par évolution).

Etape 10: Lorsque l'ensemble des génomes de la population a été évalué. L'opérateur de sélection est appliqué afin de sélectionner les individus pour la reproduction. Il existe différents techniques de sélection (section 2.3.2) : sélection à la roulette, sélection à la roulette avec reste stochastique, sélection par rang, ...etc. Toutes ces techniques traitent de manières différentes le dilemme entre exploration de l'espace de recherche par diversité génétique et exploitation des meilleurs individus par pression sélective. Cependant, il existe très peu de base théorique pour guider le choix de la méthode de sélection à utiliser. Pour cela, nous utilisons la sélection par tournoi car cette méthode se rapproche de la compétition en milieu naturel en considérant les performances relatives des individus en les comparants entre eux.

Un tournoi consiste à prendre aléatoirement un sous-ensemble d'individus et à en déterminer le meilleur. Le gagnant est retenu pour la création de la génération suivante. La taille de ce sous-ensemble représente la taille du tournoi.

Nous pouvons augmenter la pression sélective en augmentant le nombre des individus par tournoi. Cependant, une pression sélective trop forte provoque la convergence prématurée dans un optimum ou des optima locaux puisqu'elle tend à concentrer la recherche sur les meilleurs individus. En revanche, une pression sélective trop faible mène à une marche aléatoire des solutions et rend l'algorithme inefficace.

Etape 11 : Une fois les génomes sélectionnés, les opérateurs de reproduction sont appliqués indépendamment sur chacun des chromosomes des génomes. Nous définissons des opérateurs génétiques de mutation de et croisement pour chaque type de chromosome (section 4.6.3 et 4.6.4).

Les gènes compatibles sont croisés ensemble, chacun des opérateurs génétiques étant dépendant de la nature des gènes affectés (chaines de bits pour les classeurs, nombre réels ou caractères respectivement pour les paramètres et l'alphabet du L-système).

Il en va de même pour les opérateurs de mutation qui sont directement dépendants de la nature des gènes traités. Ces deux opérateurs sont appliqués dans le but de fournir une nouvelle population de génomes plus adaptée que leur parents et ainsi d'explorer l'espace possible des organismes.

Etape 12 : Une fois les individus sélectionnés se sont reproduits, une nouvelle population $P(t)$ est générée à partir des individus croisés et mutés et de l'ancienne population. Pour décider de la survie des solutions existantes et de la façon de choisir les nouveaux individus (créés par reproduction) pour faire partie de la nouvelle population. Il existe -d'une manière générale- trois stratégies de remplacement (section 2.3.2). Dans les simulations que nous présentons dans le chapitre qui suit, nous utilisons un remplacement générationnel avec élitisme.

Ainsi, la nouvelle population est entièrement formée à chaque génération. Et, étant donné que les opérateurs de reproduction s'appliquent avec des probabilités qu'il faut fixer, certains individus enfants peuvent être une copie exacte des parents qui n'ont pas été perturbés par les opérateurs génétiques de croisement et de mutation. Enfin, afin d'écartier la possibilité de perdre des solutions d'une pertinence déjà acquise, nous recopions le meilleur individu de chaque génération dans la population de la génération suivante (élitisme).

Bien que cet algorithme soit similaire à un algorithme génétique classique, il a été développé dans une optique embryogénique: les génomes des cellules ne codent pas une solution au problème, mais codent les interactions nécessaires à la croissance d'un organisme multicellulaire éventuellement capable de résoudre le problème.

La figure 4.14 représente notre démarche évolutionniste illustrée par une approche développemental. Notons que l'exécution du processus de diffusion et l'exécution du processus développemental par chaque cellule se fait en parallèle, malgré que dans le schéma de la figure 4.14, on voit que ces processus apparaissent d'une manière séquentielle.

4.6.3 Opérateurs génétiques du chromosome APL-système.

Les opérateurs génétiques des L-systèmes sont inspirés de la littérature [HP01a, EGHA02, BL08]. Afin d'explorer la totalité de l'espace de recherche, l'algorithme évolutionniste par diversité génétique, nous utilisons les deux opérateurs de reproduction: mutation et croisement.

D'une part, cinq opérateurs de mutation travaillent sur les chaînes successeurs de règles. Ceux-ci sont appliqués à chaque règle des L-systèmes avec la probabilité P_{symbol} . Afin d'éviter des imbrications erronées, les parenthèses sont ignorées lors de la sélection aléatoire des symboles.

Ces opérateurs sont définis dans le tableau ci-dessous.

Opérateur	Explication	Exemple
deleteSymbol	cet opérateur choisit un symbole aléatoirement de la chaîne successeur puis le supprime. Si l'action résulte dans une parenthèse vide, celle-ci est également supprimée.	$A \rightarrow dd + A \xrightarrow{\text{deleteSymbol}} A \rightarrow d + A$
insertSymbol	un symbole est inséré. La position dans la chaîne est entièrement aléatoire à	$A \rightarrow dd + A \xrightarrow{\text{deleteSymbol}} A \rightarrow d + A$

	l'exception des sites qui suivent un non-terminal. Le type du nouveau symbole dépend d'une probabilité P_{insert} . La probabilité de générer un module terminal aléatoire ainsi que celle de créer une commande de rotation est $P_{insert}/2$. Tandis que, la probabilité de tirer un non-terminal, aléatoirement entre les symboles prédécesseurs des règles existantes, s'élève à $1 - P_{insert}$. Pour maintenir la cohérence du génotype, un nouveau non-terminal est encadré par des parenthèses.	
permuteSymbols	Cet opérateur échange deux symboles adjacents, à moins que l'un des deux soit une parenthèse ou un non-terminal	$A \rightarrow dd + A \xrightarrow{\text{permuteSymbols}} A \rightarrow d + dA$
duplicateSymbol	Cet opérateur insère un symbole une seconde fois devant sa propre position dans la chaîne. Si le symbole généré aléatoirement est un non-terminal, la nouvelle instance est encadrée de parenthèses.	$A \rightarrow dd - A \xrightarrow{\text{duplicateSymbol}} A \rightarrow dd - [A]A$
replaceSymbol	Cet opérateur remplace un symbole par un autre. Le symbole remplaçant est sélectionné comme dans l'opérateur <i>insertSymbol</i> .	$A \rightarrow dd + A \xrightarrow{\text{replaceSymbol}} A \rightarrow -d + A$
mutateParameter	Cet opérateur remplace une condition portant sur un paramètre par une autre.	$A \rightarrow d(s_1 > 2) \xrightarrow{\text{mutateParameter}} A \rightarrow d(s_1 < 7)$

Tableau 4.3- Chromosome L-système : Opérateurs génétique de croisement.

D'autre part, deux opérateurs de croisement sont définis de la manière suivante:

Opérateur	Explication	Exemple
crossRules	cet opérateur travaille au niveau des règles entières et remplace au hasard, via un croisement discret, une règle d'un chromosome par celle de l'autre chromosome ayant le même symbole prédécesseur.	$ \begin{array}{l} A \rightarrow d+[B]A \\ B \rightarrow -[A]B \\ A \rightarrow -d+[A] \\ B \rightarrow [dB]-B \end{array} \xrightarrow{\text{crossRules}} \begin{array}{l} A \rightarrow d+[B]A \\ B \rightarrow [dB]-B \end{array} $
crossBrackets	cet opérateur se rapproche du croisement des sous-arbres de la programmation génétique. A l'instar de <i>crossRules</i> , il sélectionne l'une des deux	$ \begin{array}{l} A \rightarrow d+[B]A \\ B \rightarrow -[A]B \\ A \rightarrow d+[B]A \end{array} \xrightarrow{\text{crossBrackets}} \begin{array}{l} A \rightarrow d+[A] \\ B \rightarrow [dB]-B \end{array} $

	règles puis remplace avec une probabilité p_{bracket} une expression imbriquée dans la chaîne successeur par une expression imbriquée de la règle correspondante du deuxième chromosome. Au cas où $P_{\text{bracket}} = 0$ crossBrackets est identique à crossRules .	$B \rightarrow [dB]-B$
--	---	------------------------

Tableau 4.4- Chromosome L-système : Opérateurs génétique de mutation.

4.6.4 Opérateurs génétiques du chromosome système de sélection d'actions.

Pour le système de sélection d'action nous utilisons deux opérateurs : un opérateur de croisement et un opérateur de mutation.

Opérateur de croisement : Cet opérateur effectue un croisement mono-point opérant sur deux classeurs tirés aléatoirement à partir des deux systèmes de classeurs parents.

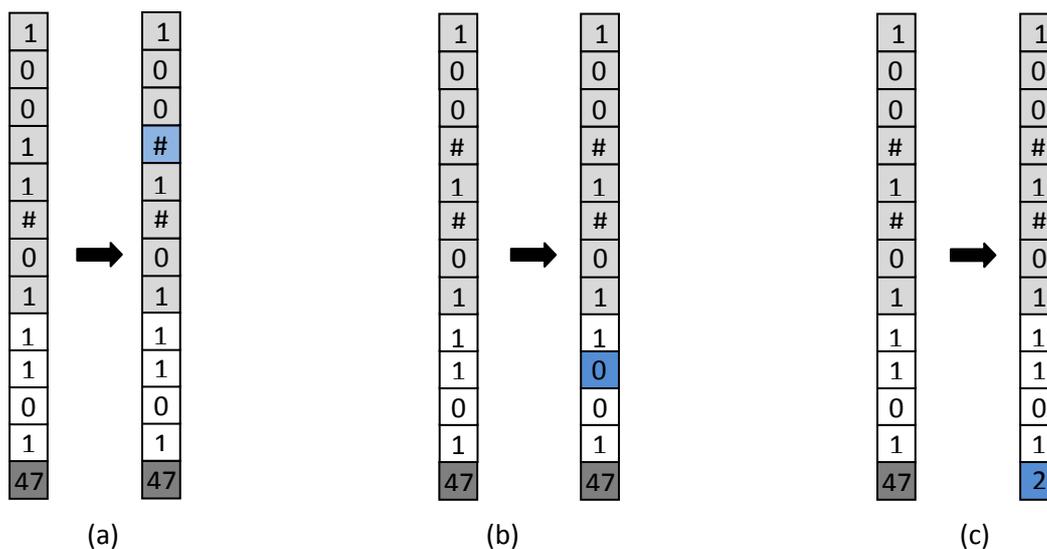


Figure 4.15- mutation dans le chromosome système de sélection d'actions. (a) La mutation porte sur une condition. (b) La mutation porte sur une action. (c) La mutation porte sur la priorité. La condition est définie sur un vecteur de 8 valeurs et l'action est définie sur un vecteur de 7.

Opérateur de mutation : Cet opérateur échange la valeur d'un des caractères de la chaîne chromosomique en l'une des autres valeurs possibles. Pour ce faire, on choisit une position dans la chaîne chromosomique et on teste.

- Si cette position est contenue dans la partie condition, la valeur remplaçante sera sélectionnée à partir de l'ensemble $\{0, 1, \#\}$.
- Sinon, si cette position est contenue dans la partie action, la valeur remplaçante sera choisie à partir de l'ensemble $\{0, 1\}$,

Sinon, c'est à dire la valeur à muter est une priorité, celle-ci sera remplacée par un nombre entier choisi aléatoirement entre 0 et 100.

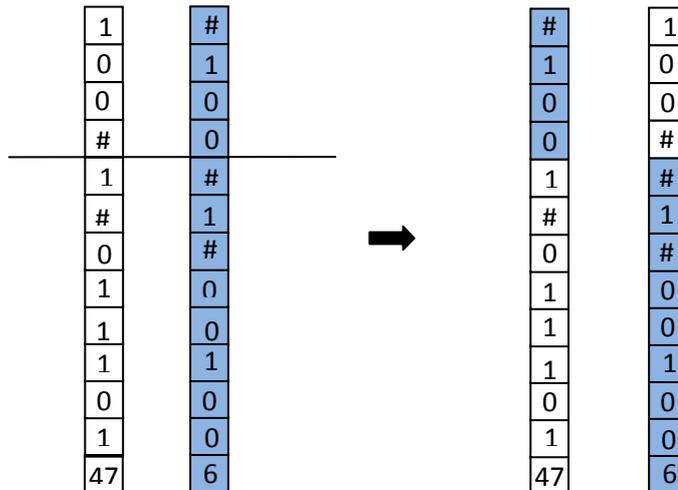


Figure 4.16 - Croisement dans le chromosome système de sélection d'actions.

4.7 Implémentation du modèle.

Dans notre implémentation du modèle, chaque organisme multicellulaire artificiel peut être vu comme un "organisme multi-agents". Chaque créature multicellulaire artificielle a été implémentée en tant qu'un système multi-agents où chaque cellule est un agent autonome comportant deux couches : une couche génétique (génome) et une couche comportementale (métabolisme et actions cellulaires). Il n'y a pas de communication directe entre ces agents mais ils coexistent dans un même environnement. Ainsi, le comportement d'un agent affecte l'environnement, ce qui peut avoir un effet sur le comportement des autres agents.

Ce modèle a été implémenté en utilisant le langage de programmation java et une architecture multithread. Toutes les cellules sont codées comme des threads indépendants qui s'exécutent en parallèle et partagent les mêmes ressources : les substrats environnementaux. Un thread est aussi utilisé pour chaque capteur de chaque cellule. Un thread supplémentaire est utilisé pour le système de diffusion et un autre pour le système de croissance codé dans l'APL-système.

Nous avons fait ce choix d'une architecture hautement parallèle vu le fort développement des architectures massivement parallèles telles que les machines multi-cœurs et multiprocesseurs. Cette parallélisation permet d'augmenter le nombre de tâches exécutées simultanément.

Notons qu'aucun mécanisme de synchronisation de processus (Producteur /consommateur, lecteur/rédacteurs,...) n'a été mise en place entre les threads. Les cellules sont complètement indépendantes et fonctionnent d'une manière totalement indépendante.

Seuls des verrous ont été mis au niveau des différentes ressources de l'environnement afin de garantir la cohérence des données.

4.8 Conclusion.

Ce chapitre nous a permis de présenter notre modèle de développement artificiel destiné à répondre à la partie ontogénétique de notre problématique. Ce modèle est fondé sur le couplage de deux approches reconnues par la littérature, à savoir, les approches grammaticales et celles basées sur la chimie cellulaire. Le chapitre suivant décrira un ensemble d'expériences effectuées avec ce modèle.

Chapitre 5

Expériences et résultats

“Il ne savait pas que c’était impossible, alors ils l’ont fait. ”, Mark Twain

Ce chapitre présente un ensemble d’expériences effectuées dans le cadre de la problématique générale de ce mémoire. La première section se concentre sur l’étude de l’évolution des L-systèmes pour générer la morphologie de nos créatures artificielles. Dans la deuxième section, nous développons deux organismes multicellulaires : une écrevisse et une chauve-souris artificielles. Une expérience supplémentaire a été ajoutée afin d’étudier les changements morphologiques de nos organismes multicellulaires en réponse aux variations de l’environnement. Nous verrons, dans la section qui suit, que nos organismes multicellulaires possèdent une propriété très intéressante qui est la capacité de se régénérer en cas de dommage. Nous présentons ensuite les limites de notre modèle. Puis nous menons une étude comparative permettant de mieux positionner notre modèle par rapport aux modèles de références présentés dans l’état de l’art. L’inspiration biologique sera discutée dans la septième section. Le chapitre est clôturé par un bilan discutant les capacités du modèle et son potentiel à répondre aux objectifs envisagés.

5.1 Développement d’organismes simples.

Dans cette section, nous voulons synthétiser des créatures multicellulaires avec le modèle de développement proposé dans le chapitre précédant. L’autosimilarité est utilisée comme critère de sélection pour le développement morphologique de nos organismes multicellulaires. Nous avons fait ce choix pour produire des créatures similaires à celles de Karl Sims [Sim94]. A long terme, elles pourront être plongées dans un simulateur physique et posséder un module comportemental de haut niveau leur permettant d’évoluer dans leur environnement et ce afin d’accomplir ce but, en produisant des créatures symétriques et possédant un métabolisme leur permettant de croître et survivre.

Les expérimentations évolutionnistes se font en deux temps pour réduire le temps de convergence de l’algorithme génétique et pour une meilleure compréhension des résultats obtenus par chaque expérimentation évolutionniste. Ainsi, dans un premier temps, nous

utilisons une fonction de fitness qui favorise l'autosimilarité bilatérale. Ensuite, une fois les meilleurs chromosomes décrivant la lignée cellulaire trouvés, la deuxième étape consiste à apprendre aux créatures à survivre dans l'environnement en développant leurs métabolismes. La fitness de chaque créature est calculée à la fin d'un temps de simulation maximale, en fonction de sa longévité ainsi que sa taille.

5.1.1 Expérimentation 1 : Génération de formes.

Le but de cette expérimentation est de générer des morphologies avec autosimilarité. Pour ce faire, nous avons étudié l'évolution des DOL-systèmes pour produire les bons chromosomes décrivant la lignée cellulaire.

Conditions expérimentales.

La taille de la grille est de 100*100, suffisamment grande pour développer des morphologies avec un nombre important de cellules. L'environnement n'est pas encore initialisé et les paramètres métaboliques n'ont pas été considérés dans cette expérience. La position de la cellule mère initiale est déterminée d'une manière empirique.

Paramétrage de l'algorithme génétique.

La fitness de chaque individu consiste à estimer l'équilibre du "poids" de sa morphologie. Ainsi, la somme des valeurs absolues des abscisses x , des coordonnées à gauche (et à droite) de l'axe vertical est calculée : $X_g(X_d)$. La meilleure valeur de fitness est attribuée aux structures les mieux équilibrées. En d'autres termes, les structures ayant un rapport gauche/droite égal à 1, ce qui revient à minimiser la différence entre X_g et X_d . La formule finale de fitness est donnée par la formule suivante:

$$fitness = |X_g - X_d|,$$

Dans toutes les expériences effectuées dans cette section, nous simulons 50 chromosomes par population durant 500 générations. Le temps de chaque simulation ne dépasse pas 50ms. Une expérience prend alors moins d'une demi-heure sur un PC avec un processeur à 2 cœurs cadencé à 2.00GHz.

Nous avons, par ailleurs, fait varier la taille du L-système pour chaque expérimentation. Cependant, cette taille, une fois déterminée, demeure fixe durant l'évolution. Le nombre maximal de règles par L-système est fixé à 4. Le premier prédécesseur de règle étant A, le dernier étant D. Dans le cas où on utilise une seule règle son symbole prédécesseur dérive vers le terminal d (cette dérivation est la dernière à s'exécuter). Quand on utilise plus d'une règle, tous les prédécesseurs se dérivent sur ϵ (la chaîne vide). Tous les opérateurs génétiques définis dans le chapitre précédent ont été utilisés pour la reproduction. Les probabilités d'application de ces opérateurs sont définies comme suit :

$$p_{insert} = 0.66, p_{symbol} = 0.1.$$

Résultats.

Chaque expérience produit une morphologie différente. La figure 5.1 montre un éventail de morphologies qui ont émergé. D'autres morphologies ainsi que les L-systèmes qui les ont engendrées sont consultables dans l'annexe A.

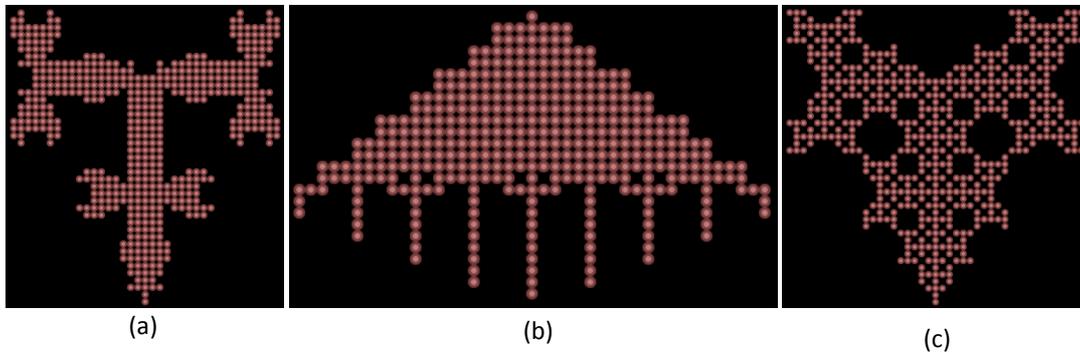


Figure 5.1- morphologies émergées de l'évolution du chromosome L-systèmes.

Effet des opérateurs génétiques mutation et croisement.

Dans la plus part des jeux de règles produits par évolution, nous avons pu remarquer qu'un taux d'apparition important de terminaux donne de meilleur résultats qu'un taux équivalent pour les non terminaux. De plus, il permet d'obtenir des morphologies avec un nombre plus important de cellules en seulement quelques itérations¹. Ainsi, les mutations de symboles non terminaux sont moins robustes. Pour cela, la probabilité de tirer un terminal a été élevée à *Pinsert* au lieu de *Pinsert/2*.

Pour la génération de morphologie avec autosimilarité, nous pouvons utiliser des opérateurs de mutation plus robustes. Par exemple, l'opérateur *insertSymbol* pourra être redéfini de la manière suivante : l'insertion d'un symbole tiré aléatoirement à partir de l'ensemble des symboles se fait deux fois sur deux positions symétriques de la règle. En d'autres termes, si le premier est inséré dans la position i en parcourant la chaîne successeur à partir du début, le deuxième est inséré dans la même position mais en parcourant la chaîne à partir de la fin (une variante de *duplicatesymbol*). Et de même pour l'opérateur *deleteSymbol*. On peut également commencer avec des chromosomes vierges (chaîne successeur vide) ou avec une taille très petite. Comme cette dynamique tend à converger plus rapidement vers les bonnes solutions, cela est critique et pose beaucoup de questions sur le dilemme exploration-exploitation.

L'opérateur *crossbraket* a été désactivé vue l'ineffectivité de cet opérateur (démontré par [Bor08]).

¹ Un taux important d'apparition de terminaux augmentera le taux d'apparition du non terminal d donc un nombre important de divisions peuvent avoir lieu durant une seule itération. En quelques itérations (3 à 4), la morphologie obtenue peut contenir rapidement une centaine de cellules.

Discussion.

Dans la genèse des chromosomes L-système, les chaînes successeurs de règles peuvent contenir des symboles inutiles² ou des symboles inaccessibles³ qui ralentiront la convergence de l'algorithme. Ainsi, une phase préparatoire d'élimination des symboles inutiles et inaccessibles pourra être d'une grande utilité.

Par ailleurs, selon la biologie évolutionnaire, le génome naturel possède une certaine quantité d'information sans fonction biologique connue, appelé "ADN-poubelle" (*junk DNA*) [Ohn72]. Plusieurs théories expliquant leur émergence ont été proposées [Bor08], telles que la "réplication égoïste" [Daw76], ou bien la duplication puis divergence de gènes existants [Ohn70]. Dans notre modèle, les règles de production inaccessibles ou inutiles peuvent être rapprochées de cette ADN-poubelle.

Bilan.

Bien qu'un angle de rotation fixé à 90° (voisinage de Von Neumann) rend la génération évolutionniste de morphologie plus complexes et particulièrement des morphologies organique (animale) un peu difficile, le but de cette expérience a été atteint.

Afin de développer d'autres morphologies, une étude des performances de chaque opérateur génétique sera d'une grande utilité.

5.1.2 Expérimentation 2 : développement du métabolisme.

Le but de cette expérimentation consiste à apprendre aux créatures à survivre dans l'environnement en développant leurs métabolismes et à étudier la capacité de ces créatures à utiliser ce métabolisme pour croître grâce à notre APL-système.

Le développement de l'organisme multicellulaire artificiel commence par une cellule mère initiale placée au milieu de l'environnement. Quand cette cellule se divise, elle donne naissance à une nouvelle cellule qui est mise dans la position spécifiée par l'APL-système contrôlant la croissance de cet organisme. Le développement se termine quand toutes les cellules constituant l'organisme multicellulaire meurent. Tous les nutriments d'un organisme multicellulaire morts sont rendus à l'environnement.

Conditions expérimentales.

Pour développer nos organismes multicellulaires, nous utilisons un modèle de chimie artificielle avec un ensemble de molécules comportant trois substrats $S = \{A, B, C\}$, avec :

² Un symbole inutile est un symbole qui ment après un certain nombre d'itération à lui-même.

³ Un symbole inaccessible est un symbole qui n'est jamais atteint par une dérivation.

- *A* : substrat environnemental bleu utilisé par l'organisme comme combustible. Ce substrat possède la propriété de se diffuser dans l'environnement à la vitesse d'une étape de diffusion toutes les 1000 millisecondes en temps simulé.
- *B* : substrat environnemental jaune utilisé également comme combustible et se diffusant dans l'environnement à la vitesse d'une étape de diffusion.
- *C* : substrat cytoplasmique rouge, produit par les cellules en utilisant les substrats environnementaux et utilisé par l'organisme comme matériel de division. Ce substrat possède la particularité de ne pas se diffuser dans l'environnement.

La grille est de taille 100*100. La propriété de diffusion de substrat permettra de répartir rapidement les deux substrats dans l'environnement.

L'ensemble des réactions de la chimie artificielle comporte deux réactions chimiques : $R = \{r1 : A + 2B \rightarrow C (-10), r2 : A \rightarrow + (60)\}$, avec :

- $r1 : A + 2B \rightarrow C (-10)$, produit une unité de substrat C en utilisant une unité de substrat A et 2 unités du substrat A. cette transformation consomme 10 unités d'énergie vitale,
- $r2 : A \rightarrow + (60)$, transforme une unité de substrat A en énergie vitale.

La cellule mère est initialisée avec 20 unités d'énergie vitale (afin qu'elle puisse survivre dans les premières instants des simulations) et 4 unité du substrat C. Chaque cellule peut effectuer les actions présentées dans le tableau 5.1. Ces actions lui permettent de :

- déclencher la transformation de substrats (algorithme de collision moléculaire stochastique),
- absorber ou rejeter du substrat A dans les quatre directions cardinales,
- absorber ou rejeter du substrat B dans les quatre directions cardinales,
- attendre une condition particulière de l'environnement.

Action	Coût énergétique	Exigences particulières
Absorb A from North Absorb A from South Absorb A from East Absorb A from West	4	La cellule doit contenir moins de 10 unités de substrat A
Absorb B from North Absorb B from South Absorb B from East Absorb B from West	4	La cellule doit contenir moins de 10 unités de substrat A

Evacuate A to North Evacuate A to South Evacuate A to East Evacuate A to West	3	La cellule doit contenir Au moins de 2 unités de substrat A
Evacuate B to North Evacuate B to South Evacuate B to East Evacuate B to West	3	La cellule doit contenir Au moins de 2 unités de substrat B
Transform substrate	-60/+10	1 unité de substrat A/ 1 unité de A et 2 unités de substrat B
Survive	2	-

Tableau 5.1-Liste des actions possibles des cellules pour développer leurs métabolismes.

Chacun des quatre coins de la cellule (nord, sud, est et ouest) possède un capteur pour le substrat A et un capteur pour le substrat B.

Evaluation de l'organisme et paramètres de l'algorithme génétique

Afin de se focaliser sur l'étude de l'évolution du système de sélection d'action et pour minimiser le temps de convergence de l'algorithme en réduisant son espace de recherche, nous utilisons le chromosome décrivant la morphologie données par la figure 5.1 (en réduisant le nombre d'itérations à 4). Ainsi, l'évolution de ce chromosome consistera seulement à faire évoluer ses paramètres métaboliques. Ainsi, seul l'opérateur génétique de mutation *mutateParameter* est activé. Nous utilisons comme paramètres : le substrat C (s_1) et l'énergie vitale (e). En effet, la cellule concernée par la division cellulaire ne pourra se diviser que si elle possède à l'intérieur de sa membrane les quantités de substrat C et d'énergie vitale spécifiées par le non terminal d (quantités nécessaire à la création d'une nouvelle cellule). Le coût énergétique de cette action est de 4 unités.

La fonction d'évaluation de l'organisme est divisée en deux sous-fonctions :

- La longévité de l'organisme *long* (durée de la simulation en millisecondes),
- La taille de l'organisme mesurée par le nombre de cellules le composant *nbCells*.

La fonction d'évaluation finale est donnée par la formule :

$$fitness = \frac{\alpha * long + \beta * nbCells}{\alpha + \beta}, \quad \text{avec } \alpha = 10, \beta = 100.$$

Les coefficients α , β de cette formule sont donnés pour prendre davantage en considération le but final de l'organisme qui est sa capacité de se développer que son métabolisme.

Le but de l'algorithme génétique est de maximiser cette fonction. Les paramètres de cet algorithme sont les suivants :

- sélection : tournoi à 6 avec élitisme,

- taux de mutation : 5%; taux de croisement : 65%,
- remplacement : individus les plus mauvais,
- taille de la population : 100 individus.

La durée de vie d'une créature multicellulaire est limitée par un temps de développement maximal (âge maximal de la créature). Ainsi, le critère d'arrêt de chaque simulation est déterminée par :

- mort de l'organisme cellulaire (absence de cellules dans l'environnement) ou ;
- expiration du temps max de développement.

Dans les premières générations, les simulations ne prennent que quelques secondes. N'ayant pas développé leur métabolisme, les cellules mères initiales meurent rapidement après consommation de l'énergie vitale qu'elles possèdent (20 unités). Au fil des générations, les cellules apprennent graduellement à utiliser les substrats environnementaux afin de survivre dans l'environnement. En état de cellule initiale, l'expression de l'APL-système commence par le non-terminal A. Celle-ci utilise les nutriments qu'elle possède (substrat cytoplasmique) pour pouvoir donner naissance à ses premières cellules filles. Ensuite, la croissance demande l'acquisition des ressources qui se trouvent dans l'environnement.

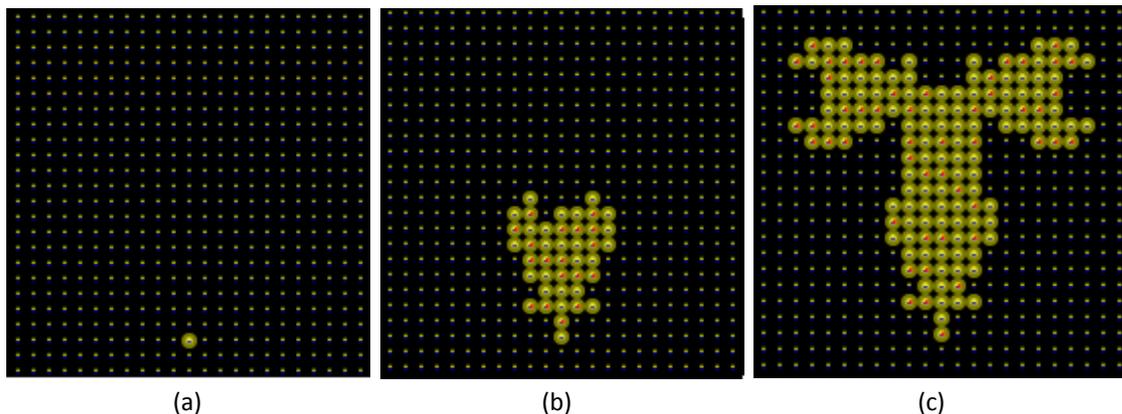


Figure 5.2- Croissance de d'un organisme (écrevisse artificielle) se composant de 146 cellules. (a) Début de la simulation. (b) L'organisme se développent en utilisant les ressource de l'environnement. (c) fin du développement.

Résultats.

La figure 5.2 montre les résultats de la meilleure créature produite par évolution (une écrevisse artificielle). Nous pouvons observer que la morphologie désirée est obtenue. On déduit donc que la créature a développé son métabolisme et que celui-ci a permis l'expression de tout le chromosome de croissance.

Il est intéressant d'étudier le système de sélection d'action produit par évolution. Celui-ci est donné par le tableau 5.2.

Condition	Action	Priorité
$(Sensor_{A_{North}} = 1)$	$\rightarrow (Absorb_{A_{North}})$	(56)
$(Sensor_{A_{South}} = 1)$	$\rightarrow (Absorb_{A_{South}})$	(48)
$(Sensor_{A_{East}} = 1)$	$\rightarrow (Absorb_{A_{East}})$	(70)
$(Sensor_{A_{West}} = 1)$	$\rightarrow (Absorb_{A_{West}})$	(63)
$(Sensor_{B_{North}} = 1)$	$\rightarrow (Absorb_{B_{North}})$	(21)
$(Sensor_{B_{South}} = 1)$	$\rightarrow (Absorb_{B_{South}})$	(13)
$(Sensor_{B_{East}} = 1)$	$\rightarrow (Absorb_{B_{East}})$	(43)
$(Sensor_{B_{West}} = 1)$	$\rightarrow (Absorb_{B_{West}})$	(37)
$(InternalState_A = 1)$	$\rightarrow (Evacuate_{A_{North}})$	(0)
$(InternalState_A = 1)$	$\rightarrow (Evacuate_{A_{South}})$	(5)
$(InternalState_A = 1)$	$\rightarrow (Evacuate_{A_{East}})$	(0)
$(InternalState_A = 1)$	$\rightarrow (Evacuate_{A_{West}})$	(0)
$(InternalState_B = 1)$	$\rightarrow (Evacuate_{B_{North}})$	(3)
$(InternalState_B = 1)$	$\rightarrow (Evacuate_{B_{South}})$	(11)
$(InternalState_B = 1)$	$\rightarrow (Evacuate_{B_{East}})$	(0)
$(InternalState_B = 1)$	$\rightarrow (Evacuate_{B_{West}})$	(0)
$(InternalState_A = 1) \text{ and } (InternalState_B = 1)$	$\rightarrow (Transform\ substrate)$	(88)
	$\rightarrow (survive)$	(16)

Tableau 5.2- Système de sélection d'action produit par le génome du meilleur organisme obtenu grâce à l'algorithme génétique.

Ce système de sélection montre que l'algorithme génétique associe correctement les données des capteurs aux actions devant être effectuées par la cellule. De plus, l'action de transformation de substrat est toujours prioritaire sur les autres actions ce qui permet d'accumuler l'énergie et les substrats nécessaires pour que l'organisme puisse suivre le schéma de croissance donné par le chromosome APL-système. Nous pouvons aussi remarquer que l'action de survie est prioritaire aux actions de rejet. Cela peut se traduire par le fait que les cellules ont appris à "préférer" ne rien faire que de rejeter les substrats nécessaires à leurs métabolisme et à leurs croissance.

Il est, également, intéressant d'étudier les paramètres métaboliques de l'APL-système de la meilleure créature : (V, Σ, ω, P) avec :

- $V = \{A, B, d, +, -, [,]\}$,
- $\Sigma = \{s_3, e\}$,
- $\omega = A$,
- $P = \{A \rightarrow Ad(s_3 \geq 3, e > 2)B[+A][-A];$

$B \rightarrow B[-d(s_3 > 2, e \geq 1)d(s_3 > 4, e \geq 2)][+d(s_3 > 2, e \geq 2)d(s_3 > 4, e \geq 3)]d(s_3 > 3, e \geq 2)B\}$

- Nombre d'itérations = 4.

Nous pouvons remarquer que les conditions des terminaux d produisant des parties symétriques portent sur des valeurs plus ou moins proches. Cela se traduit par le fait que les parties symétriques de l'écrevisse utilisent les mêmes conditions de division cellulaire.

Validation.

En observant les classeurs du système de sélection d'action découvert par l'algorithme génétique, nous pouvons remarquer qu'il est possible de produire d'autres créatures en utilisant le même chromosome. Pour vérifier cette hypothèse, nous avons développé un second organisme comportant un nombre plus grand de cellules : une chauve-souris artificielle (la morphologie donnée par la figure 5.3). Pour cela, nous utilisons le même environnement, les mêmes substrats et les mêmes actions. Et nous faisons une expérience supplémentaire dans laquelle on ne fait évoluer que les paramètres métaboliques.

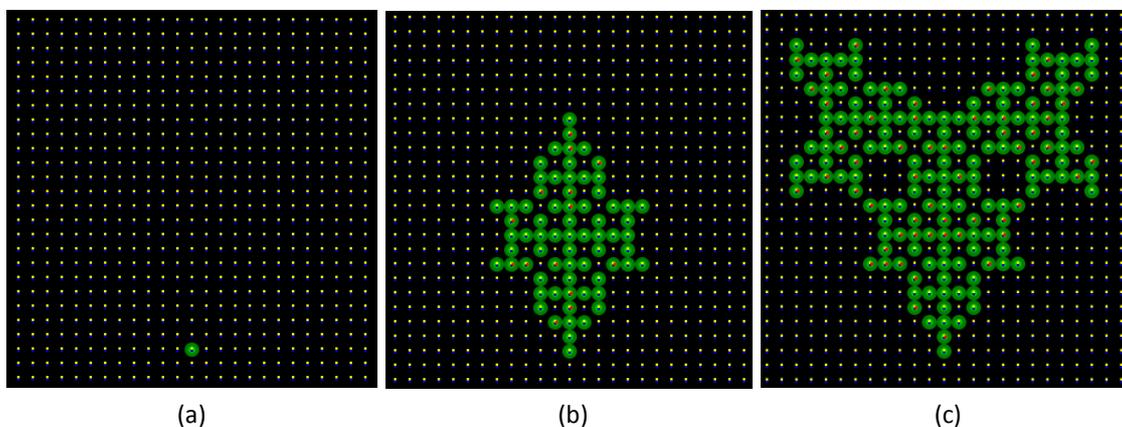


Figure 5.3- Développement d'un organisme (chauve-souris artificielle) se composant de 561 cellules. (a) Début de la simulation. (b) La chauve-souris se développent en utilisant les ressources de l'environnement. (c) la chauve-souris s'est développée.

La figure 5.3 montre le développement du meilleur organisme obtenu par évolution. L'APL-système produit par algorithme génétique est : (V, Σ, ω, P) avec :

- $V = \{A, B, d, +, -, [,]\}$,
- $\Sigma = \{s_3, e\}$,
- $\omega = A$,

- $P = \{A \rightarrow AA[+A][-A];$
 $A \rightarrow d(s_3 > 1, e \geq 0)B\},$
- Nombre d'itération = 5.

5.2 Expérimentation3 : développement dans des environnements asymétriques

Les expérimentations précédentes ont montré que nos créatures multicellulaires s'adaptent parfaitement aux variations de l'environnement. Si on utilise donc un génome qui devrait générer une créature artificielle symétrique et si on développe cette créature dans un environnement où les ressources environnementales seront réparties d'une manière asymétrique, on pourra avoir donc une créature asymétrique.

Afin de vérifier cette hypothèse, Le génome⁴ de la meilleure créature obtenu dans l'expérimentation 2 a été utilisé pour faire deux simulations. Dans la première, nous avons dispatché une quantité de substrats considérable à différents endroits de l'environnement pour qu'ils puissent se répartir de façon plus ou moins homogène sur toute la grille. L'organisme multicellulaire créé est représenté par la figure 5.4(a). Pour la deuxième simulation, nous avons initialisé l'environnement de façon à ce que les substrats bleus ne peuvent se diffuser que dans la partie droite de la grille. En utilisant le même génome, un deuxième organisme a été créé (figure 5.4(b)).

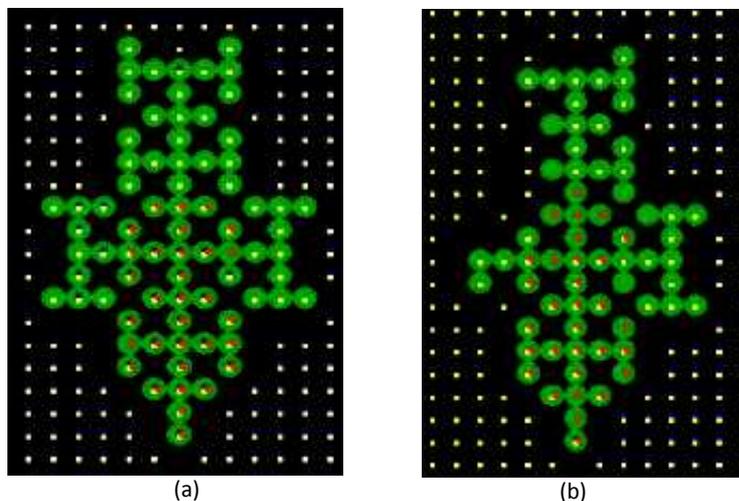


Figure 5.4 -(a) organisme multicellulaire créé dans un environnement équilibré. (b): organisme multicellulaire artificiel créé dans un environnement moins équilibré (insuffisance du substrat bleu).

En comparant les deux phénotypes, on peut constater clairement les différences de croissance entre les deux organismes créés avec le même génome dans deux environnements différents.

⁴ Nombre d'itération du chromosome L-système a été diminué à 3

Dans la deuxième simulation, l'insuffisance des ressources environnementales -ici les substrats bleus dans le côté gauche de la grille- influence directement le métabolisme de la créature :

- Mort cellulaire - avant la fin de la croissance - par manque d'énergie vitale ($A \rightarrow (60)$);
- Non capacité à se diviser par manque du substrat rouge ($A + 2B \rightarrow C (-10)$) ou d'énergie vitale.

Menant ainsi à une croissance réduite du côté gauche de la créature. Par conséquent, pour obtenir des créatures asymétriques en utilisant un génome favorisant la symétrie bilatérale, on aura qu'à les développer dans des environnements asymétriques.

5.3 Propriétés d'autoréparation.

Bien que notre modèle de développement cellulaire n'était pas conçu pour qu'il puisse exhiber cette propriété. Chaque organisme multicellulaire est capable de s'auto-réparer. En suivant toujours un mécanisme de lignée, chaque cellule mère pourra se souvenir de la position de ses cellules filles. Ainsi, elle pourra les régénérer au cas où celles-ci sont mortes par manque de l'énergie vitale.

Pour vérifier cette hypothèse, nous infligeons des blessures extérieures (réalisées par l'utilisateur) sur différents niveaux de la créature et nous étudions la capacité de celle-ci à s'auto-réparer. Cette propriété sera vérifiée en deux phases : après et au cours du développement de la créature. Pour cela, nous utilisons la créature développée en section 5.1.2 (la chauve souris artificielle).

5.3.1 Expérimentation4 : blessures Apres développement.

La figure 5.5 montre la régénération des cellules mortes par l'organisme après avoir infligé une blessure au centre de la créature et cela après la fin de son développement. Comme la croissance, la régénération aussi se fait par mécanisme de lignée. En effet, si la *cellule_i* est morte et si cette cellule a été créée par la *cellule_{i-1}*. La première ne pourra être recréée que si la seconde est en vie c.à.d. n'pas été blessée ou a été blessée mais régénérée par sa cellule mère.

Notons que si la blessure touche la cellule mère initiale. Celle-ci ne pourra être régénérée étant donné qu'elle n'a été créée par aucune autre cellule. Ainsi, par analogie au développement naturel, si l'œuf fécondé est endommagé, l'embryon ne se développera pas.

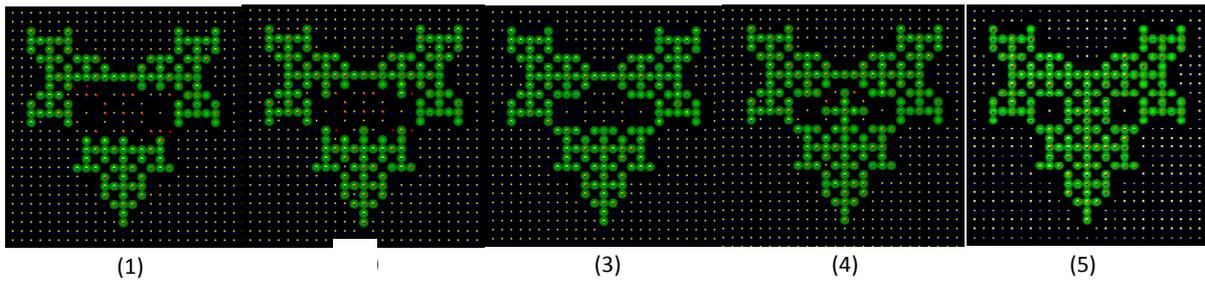


Figure 5.5- Blessures en fin du développement. (1) L'organisme après lui avoir infligé une blessure au centre. (2-5) L'organisme se régénère.

5.3.2 Expérimentation 5 : blessures en cours de développement.

Dans cette expérimentation la blessure est effectuée en cours du développement, et particulièrement sur les cellules nouvellement créées. Dans ce cas, la régénération des cellules mortes prend plus de temps étant donné que la division cellulaire demande de l'énergie et les cellules nouvellement créées n'ayant pas encore interagit avec l'environnement, n'en possèdent pas assez.

Après la fin de la régénération des cellules mortes et avant la fin du développement, nous procédons à nouveau à deux blessures simultanées sur deux endroits différents de l'organisme pour voir sa capacité de réagir à une telle situation. Une première blessure au centre bas de la créature, et une seconde en haut gauche Figure 5.6 (7). De la même manière l'organisme a pu s'adapter aux altérations du milieu Figure 5.6 (8-10). La branche latérale gauche de l'organisme mettra cependant plus de temps à se développer. Cela est dû au fait que le trou ici touche un nombre plus grand de cellules (29 cellules) que celui du centre (17 cellules).

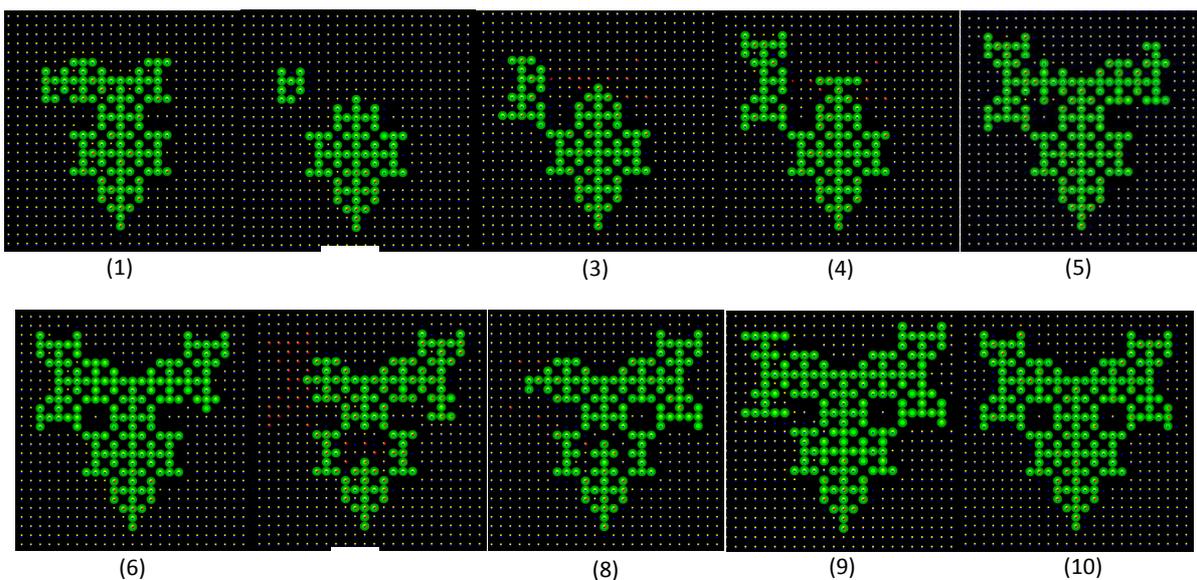


Figure 5.6-Blessures infligées au cours du développement. (1) la créature se développent. (2) Réalisation d'un trou centrale au niveau des cellules nouvellement créées. (3, 4, 5, 6) La créature se régénère. (7) la

réalisation simultanée de deux blessures : la première sur la branche latérale de la créature, la deuxième au centre bas de la créature. (8, 9, 10) La créature se régénère.

Nous avons également testé la régénération après réalisation des blessures consécutives sur le même endroit de la créature (c'est-à-dire après la régénération, on reproduit la même blessure au même endroit). L'expérience a montré que l'organisme pourra se réparer autant de fois qu'il possède de l'énergie nécessaire pour le faire.

L'autoréparation consomme de l'énergie vitale. A chaque régénération d'une cellule morte, on a dégradation du niveau de l'énergie (énergie nécessaire à une nouvelle division). De plus cette cellule a vécu un certain temps avant de se régénérer et donc son énergie s'est dégradée en fonction de sa longévité (énergie nécessaire à la survie). Pour réaliser des expériences avec des organismes blessés, l'environnement doit être initialisé avec les ressources nécessaires (déterminées d'une manière empirique en faisant la comparaison avec les ressources nécessaire au développement de l'organisme sain).

5.4 Vers la construction de drapeaux.

Dans l'expérimentation 1, au cours de développement de morphologies avec autosimilarité, nous avons obtenus trois chromosomes⁵ différents qui génèrent un phénotype en forme de drapeau (figure 5.7). Nous avons déduit que si on pourra simuler la différenciation cellulaire, plusieurs sortes de drapeau peuvent être développées avec notre modèle de développement cellulaire.

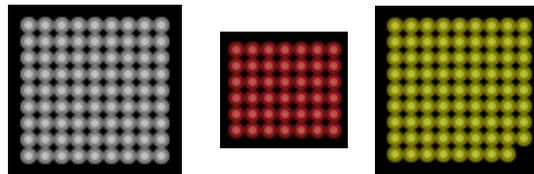


Figure 5.7-Morphologies en forme de drapeau.

Pour générer d'abord des formes 2D prédéfinies comme des carrés ou des rectangles. La fonction d'évaluation à utiliser sera la fonction de fitness classique utilisée pour les problèmes de génération de drapeaux :

$$fitness = (in - 1/2 out)/des,$$

Avec *in* représentant le nombre de cellules à l'intérieur de la forme désirée et *out* le nombre de cellules à l'extérieur de la forme désirée et *des* représentant le nombre total des cellules.

Afin de simuler la différenciation cellulaire, et pouvoir générer des formes 2D simples avec couleur et par la suite un drapeau Français, nous pouvons introduire un terminal "c" à la liste

⁵ Ces chromosomes sont consultables dans l'annexe.

de l'alphabet du L-système. Ce terminal représente le type de la cellule exprimé par une couleur.

5.5 Limites du modèle.

5.5.1 Complexité.

Les créatures que nous développons sont d'une grande complexité. Surtout si elles doivent respecter en même temps des contraintes morphologiques et fonctionnelles (adaptation avec l'environnement et développement du métabolisme). En effet, le fait d'ajouter un métabolisme utilisant des constituants de l'environnement, qui doit être découvert par la créature elle-même, rend la recherche d'un bon génome difficile. Du fait du temps de calcul nécessaire pour une simulation⁶ qui peut prendre jusqu'à 60 secondes, le calcul d'une génération composée d'une moyenne de 100 créatures peut prendre jusqu'à $100 * 60 = 6000 \text{ sec} = 100 \text{ minutes}$ par génération. Sachant qu'il faut au minimum entre 50 générations pour que l'algorithme converge, on arrive donc facilement à $100 * 50 = 5000 \text{ minutes} \approx 3 \text{ à } 4 \text{ jours}$ pour chaque expérimentation. Pour cela, l'une des principales améliorations de notre modèle consiste à paralléliser l'algorithme génétique.

5.5.2 Paramétrage.

La deuxième limite concerne l'espace des paramètres du modèle proposé. En effet, malgré la simplicité du modèle, celui-ci intègre un nombre important de variables, que ce soit au niveau de l'environnement ou des paramètres génétiques des organismes multicellulaires. Ces paramètres ont une grande influence sur la variabilité des résultats.

Etant donné qu'un organisme ne pourra se développer que si l'environnement le lui permet, les simulations sont très sensibles au paramétrage de l'environnement. Les composants de l'environnement à paramétrer sont :

- L'initialisation de l'environnement en substrats. La quantité de chaque substrat à diffuser dans l'environnement et sa répartition initiale influence directement la croissance des organismes. Par exemple, comme nous l'avons déjà montré dans l'expérimentation 3 (section 5.2), si certaines zones de l'environnement sont dépourvues de certains substrats nécessaires aux métabolismes de la créature, il y a de fortes chances que l'organisme ne se développe pas dans ces zones. Aussi, le nombre, la couleur de substrat ainsi que le type (environnemental, cytoplasmique) et

⁶ Sachant que ce temps augmente avec le nombre de cellule (threads) composant les créatures et le nombre de substrat environnementaux. Du fait que pour chaque substrat environnemental, la cellule possède un capteur associé qui est un thread aussi. Donc si on a 1000 cellules et deux substrats environnementaux, on $1000 * 2 * 4 + 1 = 8001$ threads qui s'exécutent en parallèles.

la fonction (combustible, matériel de division) sont des paramètres à définir soigneusement pour chaque simulation.

- La composition de la cellule mère : L'état interne de la cellule mère en énergie et en substrats influence aussi la croissance de l'organisme.
- Le coût de chaque action cellulaire (survie, division..) en énergie vitale est également un paramètre à définir avec soin. Par exemple, une action de division ayant un coût très faible, remplira l'environnement en cellules ce qui réduit la longévité de l'organisme en entier (surconsommation des ressources). Une action de division ; ayant un coût très fort, réduira la capacité de l'organisme à croître ou à se régénérer (s'auto-réparer) dans le cas où il a été blessé.
- La liste des réactions chimiques (transformation chimiques) valables dans l'environnement.
- Le nombre de capteurs qui a une relation avec le nombre de substrats.
- la vitesse d'exécution des threads : la vitesse de diffusion et la vitesse de réaction de chaque cellule, la vitesse de détection de chaque capteur sont des paramètres à définir avec attention pour chaque expérimentation. Par exemple, si la vitesse de détection des capteurs est inférieure à celle des réactions de cellules, cela ralentira le fonctionnement de ces dernières.

Le paramétrage de l'algorithme génétique (taux de mutation, de croisement,...) a été largement discuté dans l'état l'art. De plus on a les probabilités d'application des opérateurs génétiques du chromosome APL-système (*Pinsert*, *Psymbol*). Nous avons vu dans l'expérimentation 1 que ces paramètres influencent grandement la qualité des résultats obtenus.

5.6 Etude comparative.

Afin de situer notre modèle par rapport aux travaux de références cités dans ce mémoire, nous présentons dans cette section une étude comparative basée sur les critères suivants :

- Les propriétés des environnements des modèles de développements,
- Les mécanismes ou propriétés "biologiques" simulés,
- Les mécanismes impliqués dans le contrôle du développement,
- Les propriétés des organismes développées,

Pour ce faire, nous reprenons d'une manière plus détaillée le tableau de synthèse des travaux sur le développement artificiel présenté en section 3.4 (tableau 3.1), en le divisant en quatre tableaux.

5.6.1 Comparaison sur l'environnement.

L'environnement dans lequel nos créatures multicellulaire se développent est basé sur une grille torique bidimensionnelle. Cela nous permet par rapport aux autres modèles de développement cellulaire de créer des organismes avec un nombre plus important de cellules. En effet dans la moitié des modèles existants l'environnement est représenté par un espace 2D ou 3D discret. Cette simplification permet d'un côté d'utiliser des mécanismes plus aisés de diffusion chimique et ainsi réduire la complexité du modèle. D'un autre côté, réaliser des modèles plus simples de divisions cellulaires. L'autre moitié des modèles utilisent des environnements continus ce qui permet un plus grand réalisme dans le développement d'organismes.

Modèles	Environnement				
	2D	3D	continue	discret	Forces mécaniques
[De, 92, 99]	×			×	
[Kit90]			×		
[BKHS93]			×		
[Gru93, 94,95]			×		
[FB94, Fle96]	×	×	×		×
[Egg97]		×	×		×
[Rei99]		×			
[Ban03]		×			
[KLB04, KB04]			×		
[Mil04]	×				
[STK05]		×			×
[Cha07]	×	×		×	
[DBS07]	×			×	
[CLD08, CLD09]	×			×	
[DDCD09]	×			×	

Tableau 5.3-Comparaison sur les propriétés des environnements des modèles de développement.

Certains modèles et plus particulièrement ceux dans lesquels les environnements sont tridimensionnels et intégrant les forces physiques, permettent de simuler la propriété d'adhésion cellulaire ce qui permet d'obtenir une croissance plus réaliste. Cette propriété que nous n'avons pas prise en compte dans notre modèle sera intéressante à intégrer dans notre modèle si nous décidons l'étendre à la troisième dimension.

5.6.2 Comparaison sur les mécanismes et propriétés simulés.

Le tableau 5.4 compare les mécanismes et propriétés simulés par chaque modèle de développement. Dans la majorité des modèles sauf dans le cas particulier de la chimie artificielle, et les modèles de Kitano [Kit95] et Cussat-Blanc [CLD08, CLDa, CLDb, CLDc, CLDd] le métabolisme n'est pas simulé, ce qui consiste pourtant une fonction primordiale pour

chaque cellule biologique. Notre modèle grâce à l'intégration d'une chimie artificielle simplifiée, permet de simuler cet aspect.

La diffusion chimique quant à elle est omniprésente dans les modèles de développement. Par ailleurs, une majorité des modèles permet de simuler les interactions environnementales. Mise à part les modèles basés uniquement sur la chimie cellulaire, les interactions cellulaires, quant à elles, si elles existent sont indirectes. Notre modèle permet de simuler les interactions environnementales et les interactions cellulaires indirectes.

La régulation génétique est présente dans la majorité des modèles. Modélisé par un réseau artificiel de régulation, elle permet la régulation de la croissance des organismes multicellulaire afin de produire des formes particulières. Dans notre modèle la croissance est guidée par les règles de croissance d'un APL-système. En faisant abstraction sur les réseaux de régulation de gènes trouvés dans la nature, dans notre APL-système l'activation de gènes peut correspondre à l'apparition d'un non terminale qui crée référence à une règle non active. Comme l'activation, ou la non activation, d'une règle peut avoir un effet important sur le phénotype résultat, une règle de croissance peut être reliée au rôle d'un gène régulateur dans les RAR.

Modèles	Mécanismes ou propriétés simulés								
	Diffusion	Métabolisme	Lignée cellulaire	Régulation génétique	différentiation	spécialisation	Actions cellulaires	Interaction	Autoreparation
[De, 92, 99]				x					
[Kit90]			x						
[BKHS93]			x						
[Gru93, 94,95]			x						
[Kit95]		x							
[FB94, Fle96]	x		x	x				x	
[Egg97]	x			x	x			x	
[Rei99]	x			x					
[Ban03]	x			x					
[KLB04, KB04]				x					
[Mil04]					x				x
[STK05]	x			x	x		x	x	
[KNSQ06, KNS07]									
[CD07a,b]	x			x	x				
[DBS07]					x			x	x
[CLD08, CLD09abcd]	x	x		x		x	x	x	x
[DDCD09]	x	x	x				x	x	x

Tableau 5.4-Comparaison sur les mécanismes ou propriétés simulés.

La différenciation cellulaire est prise en compte par plusieurs modèles de développement ([Egg97], [Mil04], [STK05], [Cha07a, Cha07b]). Dans la plus part des cas, elle est exprimée avec une couleur. Dans la section 5.4, nous avons vu que cet aspect pourra être introduit

facilement dans notre modèle en ajoutant simplement un terminal c à la liste d'alphabet de l'APL-système. Cependant, la spécialisation cellulaire est peu prise en considération par les modèles de développement.

Enfin, l'autoréparation - aspect très important de la plus part des cellules biologiques⁷ - a été exhibée par de nombreux organismes multicellulaires développés récemment : modèles de Devert [DBS07], Cussat-Blanc [CLD08, CLD09a, CLD09b, CLD09c, CLD09d] et Miller [Mil04]. Notre modèle permet également de faire émerger cette propriété.

5.6.3 Comparaison sur le contrôle du développement.

Dans la plupart des modèles, la croissance est régulée par un réseau artificiel de régulation (RAR). Dans notre modèle, la croissance est basée sur une approche grammaticale simulée par APL-système. Malgré que le premier modèle utilise des mécanismes plus élémentaires (et plus proche également de la nature) pour contrôler le développement et permet d'une manière générale de définir des représentations génétiques plus expressives. Cependant, comme le note Dellaert et Beer [DB96], la perte de performance en terme de convergence n'est pas compensée par le gain potentiel d'expression. Afin de palier à ce problème, nous avons fait ce choix d'un modèle de croissance basé sur L-système qui nous a permis d'une part de simuler le mécanisme de lignée cellulaire. D'autre part, développer des organismes avec autosimilarité (concepts très intrinsèques aux L-systèmes) qui pourront facilement posséder, dans le futur, un module comportemental (basé sur des réseaux de neurones comme les créatures de Karl Sims [Sim94]) et accomplir quelques fonctions de locomotion dans des environnements où la physique est simulée.

Quelques travaux utilisent comme modèle de croissance des automates cellulaires contrôlés également par des réseaux artificiels de régulation. C'est le cas des modèles simulant le développement de drapeau comme le modèle pionnier de Garis [De, 92, 99] ou le modèle de Chavoya [CD07a, CD07b].

Les équations différentielles et les réseaux de neurones ([FB94, Fle96], [DBS07]) sont utilisés par les modèles simulant la croissance dans des environnements continus. Ces deux modèles permettent alors de simuler la croissance en déterminant d'une manière continue la façon dont la constitution interne de la cellule en protéine change en fonction de son environnement local (cellules voisines comprises) ainsi que de son état courant.

Les modèles basés uniquement sur la chimie artificielle ([OI99]) permettent, quant à eux, de simuler d'une manière plus réaliste les détails des mécanismes cellulaires. Cependant, la taille des organismes développés par ces modèles ne dépassent pas une dizaine de cellules.

Dans notre modèle, nous utilisons aussi une chimie artificielle afin de simuler le métabolisme des cellules. Cela a été utilisé également dans le modèle de Sylvain Cussat-Blanc [CLD08, CLD09a, CLD09b, CLD09c, CLD09d]. Notre modèle, à la différence de ce dernier, utilise un

⁷ Les cellules du système nerveux ne se régénèrent pas.

processus de division basé sur un mécanisme de lignée et les conditions de cette division cellulaire ne sont pas les mêmes durant toutes les phases de développement. De plus, elles sont déterminées d'une manière adaptative grâce à notre variante de L-système paramétrique.

Modèles	Mécanismes impliqués dans le contrôle du développement							
	Automate cellulaire	RAR	L-système	Réseau de neurones	Equations différentielles	Evolution génétique	Chimie artificielle	Autres
[De, 92, 99]	×	×				×		
[Kit90]			×					
[BKHS93]			×					
[Gru93, 94,95]			×					
[Kit95]								×
[FB94, Fle96]			×		×			
[Egg97]		×				×		
[Rei99]		×						
[OI99]							×	
[Ban03]		×				×		
[KLB04, KB04]		×						
[Mil04]								×
[STK05]		×						
[KNSQ06, KNS07]		×				×		
[CD07a, CD07b]	×	×				×		
[DBS07]				×				
[CLD08, CLD09]		×				×	×	
[DDCD09]			×			×	×	

Tableau 5.5- Comparaison sur les mécanismes impliqués dans le contrôle du développement.

Enfin, Comme la majorité des modèles, le développement de nos organismes multicellulaires est contrôlé par un algorithme évolutionniste.

5.6.4 Comparaison sur les organismes développés.

Le tableau 5.6 montre les propriétés des organismes développés : leur phénotype, leur taille et la forme des cellules les composants.

Il est intéressant d'observer la taille des organismes obtenus par chaque modèle. En effet, cette taille est très variable. Elle peut être de quelques cellules pour les modèles tentant de simuler d'une manière réaliste des mécanismes cellulaires comme la formation de la membrane et les détails de tous les échanges intra et extra membranaires. C'est le cas, des modèles de la chimie artificielle ([OI99]). De même, les modèles de réseaux artificiels de régulation ([Rei99], [Ban03]) se focalisent sur une seule cellule pour étudier les dynamiques de la régulation génétiques. Par ailleurs, certains modèles tentent de produire des drapeaux⁸. C'est le cas de la majorité des modèles ([KNSQ06, KNS07], [Mil04], [CD07], [DBS07]). D'autres, ayant des objectifs plus ambitieux, ils tentent de produire des organes ou

⁸ Rappelant qu'un drapeau signifie une forme 2D prédéfinie.

des structures imprévisibles comme une étoile de mère ([CLD08, CLD09a, CLD09b, CLD09c, CLD09d]), un organisme en forme de cigare, tube ou quadrupède ([STK05]). Dans ces derniers cas, la taille des organismes varie entre 20 à 650 cellules.

Notre modèle, grâce aux simplifications que nous avons réalisées, que ce soit sur la modélisation de l'environnement, sur le modèle de croissance et plus particulièrement sur le modèle de la chimie artificielle proposé, nous permet de développer des organismes avec une taille plus importante (plusieurs centaines de cellules). La seule limite étant la taille de la grille de diffusion et la quantité des ressources environnementales nécessaires au métabolisme et ainsi à l'espérance de vie de l'organisme.

Modèles	Organisme développé		
	phénotype	Taille	Forme de la cellule
[De, 92, 99]	drapeau	~ 200 cellules	carrée
[Kit90]	Réseaux de neurones	-	-
[BKHS93]	Réseaux de neurones	-	-
[Gru93, 94,95]	Réseaux de neurones	-	-
[Kit95]	Réseaux de neurones	-	-
[FB94, Fle96]	Formes, réseau de neurones	~ 30 cellules	Cercle \ sphère
[Egg97]	formes	~ 75 cellules	sphère
[Rei99]	graphe	1 cellule	-
[OI99]	-	~ 6 cellules	polygone
[Ban03]	graphe	1 cellule	-
[KLB04, KB04]	graphe	1 cellule	sphère
[Mil04]	drapeau		carrée
[STK05]	formes	~ 650 cellules	sphère
[KNSQ06, KNS07]	drapeau	1 cellule	
[CD07]	drapeau	~ 250 cellules	Cercle\ sphère
[DBS07]	drapeau	~ 60 cellules	Carré
[CLD08, CLD09]	Organes, étoile de mère, organe autoalimenté	~500 cellules	losange
[DDCD09]	Ecrevisse, chauve-souris, drapeau	Plusieurs centaine de cellules	losange

Tableau 5.6-Etude comparative : comparaison sur les organismes développés.

Pour représenter graphiquement les cellules, les modèles utilisent des sphères ([Egg97] [STK05]...), des cercles ([CD07]), des carrés ([De, 92, 99], [Mil04]) ou des losanges ([CLD08, CLD09a, CLD09b, CLD09c, CLD09d]), etc. Les polygones ([OI99]) sont utilisés par les modèles permettent la déformation des cellules lors de la croissance. Dans notre modèle, nous avons opté pour des losanges parce que c'est la plus petite forme géométrique possédant 4 coins (voisinage de Von Neumann). Ceci nous permettra de réduire le nombre de capteurs par

cellule⁹ qui sont en fait des threads qui s'exécutent en parallèle, ce qui réduit la complexité de la simulation.

Concernant les phénotypes développés, notre modèle permet de générer plusieurs formes telles que l'écrevisse ou la chauve-souris, De plus, nous avons montré que notre modèle pourra s'étendre facilement pour inclure la différenciation cellulaire et développer des drapeaux.

5.7 Inspiration biologique : discussion.

En nature, la transformation d'une cellule souche en un embryon est la conséquence d'une séquence complexe d'interactions génétiques, cellulaires et environnementales (une ontogénèse).

Par analogie au développement biologique, notre modèle est capable de modéliser les interactions chimiques, l'interaction cellulaire indirecte et la diffusion de substrats dans l'environnement. Pour ce qui est des interactions génétiques (régulation), dans les L-systèmes, l'activation de gènes correspond à l'apparition d'un caractère non-terminal qui crée une référence à une règle inactive. Ainsi, comme l'activation ou la désactivation d'une règle du L-système peut avoir un effet drastique sur le phénotype généré, chaque règle de production peut ainsi être reliée au rôle des gènes régulateurs dans les organismes naturels.

Mais cela ne signifie guère que la croissance des organismes multicellulaires naturels est guidée par un mécanisme similaire aux L-systèmes. Cela est juste une métaphore, car le but du modèle n'est en aucun cas de concevoir un système biologiquement plausible mais de créer un système qui rivalise le niveau de la complexité constaté dans la nature et qui peut servir par la suite comme modèle de conception de robots autonomes.

5.8 Bilan.

Le but de cette section est de répondre à deux questions :

- quelles sont les capacités de notre modèle ?
- A quel point ce modèle répond aux objectifs fixés (décrits dans section 4.1.1) ?

En effet, notre modèle montre une approche nouvelle du développement de créatures artificielles. Fondé sur le couplage des deux approches : grammaticales et chimiques, il a pu répondre à certains manques des modèles existants, tels que la combinaison de plusieurs mécanismes cellulaires dans un seul modèle de développement (diffusion chimique, lignée cellulaire, division, apoptose).

⁹ Si on a une cellule avec X coins et si l'environnement contient Y substrats, on aura besoin de X.Y capteurs pour chaque cellules.

De plus, nous avons pu intégrer une chimie artificielle simplifiée permettant à l'organisme de construire un métabolisme dépendant des propriétés de son environnement, fonctionnalité très importante de tout organisme biologique pourtant omise par la plus part des modèles existants. En effet, outre le modèle de Cussat-Blanc ([CLD08, CLD09a, CLD09b, CLD09c, CLD09d], aucun modèle ne prend en compte une idée de dépense énergétique nécessaire à l'accomplissement d'une action (croissance, division, différenciation).

En outre, L'étude de L-systèmes évolutifs a été largement explorée dans le domaine de recherche de la vie artificielle et notamment dans le développement de plantes virtuelles. Nous avons pu explorer la capacité de ces systèmes dans le contexte d'un modèle d'ontogenèse artificielle. De plus, nous avons proposé une variante de L-systèmes paramétriques simulant la croissance cellulaire d'une manière adaptative.

Grace à ce modèle, nous avons pu synthétiser des créatures multicellulaires possédant une morphologie générée par un APL-système et un métabolisme leur permettant de croître et agir. Ces créatures possèdent des morphologies auto-similaires, comme celles développées par Karl Sims [Sim94].

Par rapport aux objectifs fixés, nous n'avons donc pas abordé la partie phylogénétique de notre modèle. Cela fera donc l'une des perspectives de ce mémoire.

5.9 Conclusion.

Afin de valider notre modèle, ce chapitre nous a permis de présenter une série d'expérimentations qui ont été menées dans le but de produire des organismes multicellulaires possédant une morphologie et un métabolisme leur permettant de croître et agir.

Chapitre 6

Conclusion générale

*“There is nothing wrong with a good illusion
as long as one does not claim it is reality.”, Mark Twain.*

L'objectif principal de cette thèse était de proposer un modèle hybride ontogénétique et phylogénétique permettant de développer des créatures artificielles complètes, possédant un métabolisme, une morphologie et un comportement. Dans l'optique d'élaborer un modèle permettant de répondre à la problématique donnée, le premier pas de cette thèse consistait donc à identifier les éléments nécessaires pour la compréhension puis la formalisation d'un outil de recherche approprié.

Nous avons donc commencé par présenter le domaine de la vie artificielle, la discipline dans laquelle s'inscrit le cadre de ce mémoire. Puis les mécanismes et techniques sur lesquels ce champ de recherche est basé. Nous avons ensuite synthétisé les travaux sur la génération de créatures artificielles en les classifiant en trois catégories : phylogenèse, épigenèse et ontogenèse.

Le développement étant un processus biologique très compliqué, afin de comprendre puis d'établir un modèle artificiel de ce processus, nous avons mené une brève étude sur le développement naturel et sa place dans l'orientation de l'évolution des espèces, puis présenté quelques prérequis biologiques concernant ses principaux mécanismes. Nous avons, ensuite, collecté quelques définitions de l'ontogénèse artificielle que nous avons trouvés dans la littérature et présenté ses objectifs, avantages et inconvénients comme une sous discipline du calcul évolutionniste et comme étant une approche permettant de synthétiser des créatures artificielles.

L'étude s'est ensuite portée sur les travaux effectués par la communauté scientifique sur l'ontogenèse artificielle. De plus, nous avons effectué une synthèse de ces travaux en nous basant sur les classifications que nous avons trouvées dans la littérature. Enfin, les résultats des expérimentations permettent d'affirmer que nous avons réussi à apporter une réponse aux objectifs principaux fixés dans la problématique de ce mémoire.

En effet, nous nous sommes proposés dans ce mémoire, de développer un nouveau modèle de développement cellulaire fondé sur le couplage de deux approches reconnues dans la littérature, à savoir les approches grammaticales et celles basées sur la chimie cellulaire. D'une part, ce modèle simule les éléments majeurs d'un modèle de chimie cellulaire à savoir : les réactions chimiques d'une façon simplifiée, la diffusion de substrats dans l'environnement, les interactions cellulaires et environnementales ainsi que quelques actions cellulaires comme la division ou l'apoptose. D'autre part, la morphogenèse est fondée sur une approche grammaticale utilisant des règles de croissance codées dans un APL-système, variante de L-système paramétrique que nous avons proposée afin de simuler d'une manière adaptative la croissance de nos organismes multicellulaires en fonction de l'état interne des cellules et des composants.

Beaucoup plus bio-inspiré que biologiquement plausible, ce modèle est capable de produire des créatures artificielles à partir d'une cellule unique. Les créatures développées possèdent une morphologie générée par un mécanisme de lignée basé sur les L-systèmes et un métabolisme leur permettant de croître et agir. Dans la lignée des travaux de Sylvain Cussat Blanc, notre modèle est l'un des rares modèles de développement prenant en compte un métabolisme qui est trop souvent volontairement omis dans les modèles de la littérature et qui est pourtant primordial pour le fonctionnement de toute cellule biologique. En outre, notre modèle est capable de faire émerger la propriété d'autoréparation au cas où l'organisme est blessé après ou même au cours de son développement. Cette propriété, inhérente à tout organisme naturel, est de nos jours vérifiée par une majorité de travaux récents.

A notre connaissance, notre modèle représente la première utilisation des L-systèmes pour orienter la morphogénèse d'une créature multicellulaire artificielle. Nous avons fait ce choix afin d'explorer le potentiel de ces systèmes largement utilisés avec succès dans de nombreux domaines et particulièrement dans le contexte de recherche en vie artificielle.

Afin de répondre à notre objectif, développer des créatures similaires à ceux développés par Karl Sims, nous avons développé des créatures présentant des propriétés d'autosimilarité (écrevisse et chauvesouris artificielles). Nous avons montré que les L-systèmes sont bien adaptés pour la réalisation d'un tel objectif.

Par ailleurs, notre APL-système est une variante unique de L-systèmes paramétriques. Elle a la particularité d'être indépendante du modèle proposé et peut donc être utilisée dans une large gamme d'applications qui utilisent les L-systèmes paramétriques.

6.1 Perspectives

La principale amélioration de notre modèle consiste à pallier aux limitations présentées dans le chapitre précédant. Afin de réduire le temps de convergence de l'algorithme génétique, nous envisageons paralléliser cet algorithme en explorant les méthodes existantes dans la littérature [Sek04, HML05, BKS04].

Nous envisageons également étendre notre modèle en utilisant d'autres extensions des L-systèmes : les L-systèmes stochastiques afin de simuler le développement des cellules en fonction des cellules adjacentes, et les L-systèmes contextuels afin d'augmenter la variété de morphologies obtenues, en utilisant le même jeu de règles.

Le modèle de l'environnement pourrait être doté de nouvelles caractéristiques. N'intégrant à l'heure actuelle que les substrats, il est notamment envisagé d'intégrer d'autres paramètres comme la température, le niveau de PH...etc.

A long terme, nous comptons converger notre travail dans l'optique de générer des créatures artificielles complètes qui possèdent un métabolisme, une morphologie, et qui peuvent évoluer en comportement en satisfaisant quelques fonctions d'adaptation, telles que la capacité de locomotion. Pour ce faire, une fois un organisme multicellulaire artificiel créé avec le modèle proposé, il suffit de faire abstraction de ce modèle et de plonger cet organisme dans un simulateur physique. Les mouvements des créatures pourraient être accomplis à l'aide d'un réseau de neurones comme dans les travaux de Karl Sims.

L'objectif de la synthèse de nos créatures artificielles, consiste dans le fait qu'elles peuvent servir dans le futur comme modèles de conception de nano-robots autonomes. Par rapport aux créatures de Karl Sims, le métabolisme introduit dans notre modèle de développement cellulaire permet de fournir plus d'autonomie, à ces modèles de robots. Car le métabolisme est un moyen de se procurer de l'énergie. Ainsi, les substrats, par exemple, peuvent être considérés comme métaphores de batteries électriques que les robots peuvent récupérer à partir de l'environnement et s'en servir pour exécuter des tâches spécifiques.

Remarque.

Au cours de la rédaction de ce manuscrit, les scientifiques ont découvert que la théorie darwiniennes est fausse, malgré cela de nombreuses approches biologiques informatiques et même sociales sont fondées sur cette théorie, notre travail y compris. Et ils fonctionnent toujours... . On peut déduire que parfois certains défauts ou erreurs de fonctionnement permettent de développer des systèmes pertinents. On le constate aussi dans les créatures de Lassabe où une division par Zéro à conduit vers la meilleure créature.

C'est donc en faisant l'une de ses grandes erreurs que l'humanité a probablement réussi à faire ses plus grands exploits ...

Annexe A

Systemes de classeurs

"We are all agreed that your theory is crazy. The question which divides us is whether it is crazy enough to have a chance of being correct." Niels Bohr

Cet annexe vient en complément de la partie Systemes de classeurs présentée en section 2.4.2.

Structure.

Un système de classeurs est composé de :

- Une population d'individus.
- Chaque individu est lui-même soit un classeur, soit un ensemble de classeurs.
- Un individu possède une *qualité*, une *force* ou une *fitness*, qui est la valeur représentative de cet individu par rapport à la solution et qui évolue en fonction des rétributions de l'environnement.
- Un classeur possède une partie condition et une partie action : les classeurs sont qualifiés également de règles de production.
- Les règles de production sont des chaînes de caractères définies sur un alphabet.
- La partie condition représente les critères qui vont déterminer si le classeur peut être activé ou non à un instant t. Elle représente les capteurs du système de classeurs.
- La partie action sert à déterminer le comportement du système de classeurs. Elle représente les effecteurs du système de classeurs.

Formalisme de représentation

Formellement, la partie condition des classeurs est constituée d'un vecteur de tests. Chaque test portant sur un attribut spécifique. Il existe un test particulier, noté «#» et appelé «*don't care symbol*». Ce symbole représente le moyen mis en œuvre par les systèmes de classeurs

pour assurer la propriété de généralisation [sig07], et signifie que la condition stipulée par le classeur sera vérifiée quelle que soit la valeur de l'attribut correspondant.

Dans sa version originelle, un système de classeurs fonctionne uniquement avec des vecteurs de tests booléens. Chaque test de la partie Condition prend alors sa valeur dans l'ensemble $\{0, 1, \#\}$. Plus récemment, de nombreux systèmes ont été étendus pour prendre en compte des valeurs discrètes (voir par exemple dans [GMS05]) ou continues (voir par exemple dans [Wil01, HSD02]). Dans ce second cas, un test spécifie généralement un intervalle dans lequel doit se situer la valeur de l'attribut pour que la condition soit vérifiée. Il existe aussi des systèmes de classeurs dans lesquels chaque condition individuelle est exprimée sous la forme d'une expression symbolique¹ que doit vérifier l'attribut pour que la condition soit vérifiée (voir par exemple dans [AB99, LP99]).

Systèmes de classeurs basés sur la force : ZCS.

Le ZCS (Zeroth level Classifier System) a été présenté par Wilson [Wil94] comme système de classeurs de type Michigan. Ce système assimile la qualité ou la force d'un classeur à l'espérance de renforcement reçu en déclenchant le classeur concerné, donnant naissance à la lignée des systèmes « basés sur la force ». Chaque classeur se compose d'une condition, action et qualité $\langle C, A, f \rangle$. Dans ce type de système, La *fitness* des classeurs est directement égale à la qualité du classeur, c'est-à-dire à l'estimation de récompense attendue calculée par l'apprentissage par renforcement. Il précise les mécanismes suivants :

- 1/ **Sélection** : Une fois déterminé l'ensemble $[M]$ des classeurs activables, le choix de l'action activée s'effectue dans les ZCS selon une sélection à la roulette [Gol89].
- 2/ **Evaluation des classeurs** : Le mécanisme de propagation des rétributions est proche de l'algorithme *Bucket Brigade* [DB94]. Plus précisément, si on note a_t l'action choisie effectivement à partir de $[A]$, dans un premier temps, tous les classeurs qui ont proposé l'action a_{t-1} choisie lors du pas de temps précédent se partagent équitablement une fraction $\gamma \cdot \alpha$ de la somme des valeurs des classeurs de $[A]$. Dans un deuxième temps, tous les classeurs de $[A]$ se partagent équitablement une fraction α de la récompense r_t reçue pour l'exécution de a_t . Enfin, tous les classeurs de $[M]$ qui ne font pas partie de $[A]$ voient leur valeur réduite d'une taxe τ .
- 3/ **Evolution de la population**. L'évolution de la population de classeurs est commandée dans ZCS par un algorithme génétique et un opérateur de couverture.
 - A chaque pas de temps, il y a une probabilité p d'invocation de l'algorithme génétique. S'il est invoqué, il utilise la sélection à la roulette pour choisir deux classeurs dans l'ensemble de la population $[P]$. Deux descendants de ces classeurs sont engendrés à l'aide d'un opérateur de croisement à un point et d'un opérateur de mutation. Ces descendants reçoivent la moitié de la *qualité* de chacun de leurs parents. Ils remplacent au sein de la population deux autres

¹ S-expression.

classeurs choisis par un mécanisme de sélection à la roulette basé sur l'inverse de la *qualité*.

- L'opérateur de couverture est invoqué à chaque fois que l'ensemble [M] est vide ou contient des classeurs dont la *qualité* est considéré comme insuffisante.

Systèmes de classeur basés sur la précision : XCS.

Après ZCS, Wilson a imaginé un nouveau système plus subtil, baptisé « XCS » [Wil95], dans lequel la force représente cette fois la capacité du classeur à *prédire précisément* la quantité de renforcement, que son déclenchement permet de recevoir. XCS s'est avéré très performant et constitue le point de départ d'une lignée de systèmes « basés sur la précision ». L'idée fondamentale au cœur de XCS consiste à découpler les processus d'apprentissage par renforcement et d'évolution des règles. En effet, la *qualité f* du ZCS est décomposée ainsi:

- la Prédiction de paiement p représente la récompense attendue en effectuant l'action A dans les situations appariées par C ;
- l'Erreur sur la prédiction e représente l'écart entre la prédiction de paiement p et le paiement réel ;
- la Fitness fit est la fonction inverse de e , elle représente donc l'exactitude de la prédiction de paiement p .

Ce système de classeur précise les mécanismes suivants :

- 1/ Sélection :** La Sélection se base non plus sur la *Qualité f* du classeur du ZCS mais sur la précision de la *Prédiction de paiement p*. Cette méthode permet de sélectionner les classeurs n'ayant pas seulement une action optimale mais plutôt ceux qui permettent de donner la *qualité* de l'action proposée avec précision. Le mécanisme de gestion du compromis entre exploration et exploitation est géré de différentes manières. Parmi les mécanismes, on trouve le recours à *ϵ -greedy*, l'alternance d'essais de pure exploration et d'essais de pure exploitation [Sig07], ou bien le mécanisme de sélection à la roulette utilisé aussi dans ZCS.
- 2/ Evaluation des classeurs :** L'évaluation se fait grâce à un algorithme évolué de Q-Learning [Wat89]. Elle nécessite un calcul des prédictions (*Prediction Array PA*). Pour chaque action a_i présente dans [M], une prédiction $p(a_i)$ est calculée. Une fois la sélection effectuée, l'ensemble des actions précédentes $[A]_{-1}$ (l'ensemble [A] lors du dernier cycle) est modifié en utilisant une combinaison de la dernière rétribution de l'environnement et la prédiction maximale de la PA actuelle.
- 3/ Evolution de la population :** Comme dans ZCS, ce mécanisme utilise un algorithme génétique ainsi qu'un opérateur de couverture.
 - L'algorithme génétique intervient sur les règles se situant dans [A] et se base sur l'erreur de prédiction. La *Fitness fit* est la valeur sélective. Si l'erreur est grande alors la valeur sélective est faible et inversement. Butz [BP01] a montré que cette

méthode favorise le maintien dans [P] des classeurs les plus généraux. Le classeur est conservé tant que son erreur est faible. Les parents sont choisis selon une sélection à la roulette. Récemment, Butz [BSG03] a montré que l'on obtenait une augmentation significative de la performance du système XCS en choisissant l'action par un mécanisme de tournoi 20 au sein de [M]. Deux rejets de ces parents sont insérés dans la population globale, après application d'un opérateur de mutation soit libre, soit guidé par la perception courante. On parle de mutation guidée lorsqu'un bit ne peut pas muter vers une valeur opposée à celle qui est présente dans la perception courante.

- L'opérateur de couverture permet d'initialiser [P] avec un ensemble vide ou très réduit.

Systèmes de classeurs basés sur l'anticipation : ACS.

Le premier système de classeurs par anticipation conçu après les simplifications apportées par Wilson au domaine est ACS (*Anticipatory Classifier System*) [Sto98, BGS00] qui a donné lieu à une nouvelle lignée des systèmes de classeurs «basés sur l'anticipation». Ceux-ci se distinguent sensiblement des deux précédant. Ils se basent sur l'hypothèse que l'anticipation des conséquences engendrées par une action est primordiale pour définir un comportement. Nous présentons ici les principaux systèmes:

- Le système ACS ajoute une partie *effet* pour chaque règle, anticipant les changements du monde causés par la partie action sous la partie condition. La qualité d'un classeur est alors calculée sur la base de la qualité de l'anticipation des effets de la règle sur le monde. Ainsi le système crée un modèle du monde lui permettant de planifier ses actions. Il propose l'utilisation d'heuristiques à la place des algorithmes génétiques pour la construction des règles ;
- Le système YACS (*Yet Another Classifier System*) [GSS02] utilise le même formalisme que le système ACS avec des heuristiques différentes permettant d'améliorer la vitesse de l'apprentissage latent ;
- le système MACS (*Modular Anticipatory Classifier System*) [GMS05] est un formalisme pour l'apprentissage latent qui introduit les anticipations partielles, ses règles n'anticipent pas les valeurs de tous les capteurs en une seule fois. Ce système offre de nouvelles possibilités de généralisation et de performances.

Systèmes de classeurs hiérarchiques.

Les systèmes hiérarchiques proposent l'utilisation de plusieurs systèmes de classeurs s'échangeant des informations. Nous citons les deux principaux modèles :

- le système ALECSYS (*A LEarning Classifier SYStem*) [Dor95] est une structure hiérarchique permettant de décomposer une tâche complexe en un ensemble de tâches simples. Un système de classeurs joue le rôle d'arbitre dans le choix de l'activation d'un des autres systèmes qui apprennent les comportements de base ;

- le système OCS (*Organizational Classifier System*) [TTS99] est composé de différents agents possédant chacun un système de classeurs devant satisfaire une tâche commune. Le mécanisme de spécialisation et la communication entre agents permettant d'augmenter l'efficacité du système.

Annexe B

Morphologie et Règles de L-systèmes

“Data is not information, information is not knowledge, knowledge is not understanding, understanding is not wisdom.” Clifford Stoll

Cet annexe réunit les règles d’L-système obtenu dans le chapitre 5 durant l’expérimentation de la génération de formes (sections 5.1.1). Il présente également d’autres morphologies obtenues ainsi que les règles qui les ont engendrés.

A.1 Règles engendrant les morphologies présentées dans le chapitre Expérimentation.

Règles de la morphologie de la figure 5.1(a):

- $P = \{A \rightarrow AdB[+A][-A];$
 $B \rightarrow B[-dd][+dd]dB\}$
- Nombre d’itération = 5.

Règles de la morphologie de la figure 5.1(b):

- $P = \{A \rightarrow AA[+A][-A];$
 $A \rightarrow dA\},$
- Nombre d’itération = 5.

Règles de la morphologie de la figure 5.1(c):

- $P = \{A \rightarrow d[B]Ad;$
 $B \rightarrow [-ddC] + ddD]$
 $C \rightarrow d + dd - dddAd\}$
- Nombre d’itération = 12.

Règles de la morphologie de la figure 5.7(a):

- $P = \{A \rightarrow [A] + [A] + [A] + A$
 $A \rightarrow d[+d][-d]\},$
- Nombre d’itération = 8.

Règles de la morphologie de la figure 5.7(b):

- $P = \{A \rightarrow dB$
 $B \rightarrow -dB + +dC -$
 $C \rightarrow +dB - -dC +\},$
- Nombre d’itération = 12.

Règles de la morphologie de la figure 5.7(c):

- $P = \{A \rightarrow A + [+A - AA -$
 $AA - [-A + AA + AA]$
 $A \rightarrow d\},$
- Nombre d’itération = 4

A.2 Autres morphologies.

Règles de la morphologie de la figure 1:

- $P = \{A \rightarrow ddd[B]A$
 $B \rightarrow [-ddC] + dd$
 $C \rightarrow d + dd - dd + dA$
 $D \rightarrow d - dd + dd - dA\}$,
- Nombre d'itération = 12.

Règles de la morphologie de la figure 2:

- $P = \{A \rightarrow A[-AA + A][+dd - d]d$
 $A \rightarrow d\}$,
- Nombre d'itération = 4.

Règles de la morphologie de la figure 3:

- $P = \{A \rightarrow B + B + B + B$
 $B \rightarrow B[-B] - B + B[+B]B\}$,
- Nombre d'itération = 4.

Règles de la morphologie de la figure 4:

- $P = \{A \rightarrow ddd - [BC] + [BC]A$
 $B \rightarrow +dC$
 $C \rightarrow -dB\}$,

Nombre d'itération = 4.

Règles de la morphologie de la figure 5:

- $P = \{A \rightarrow AA[-A + +A][+A - -A]A$
 $B \rightarrow A\}$,
- Nombre d'itération = 3.

Règles de la morphologie de la figure 6:

- $P = \{A \rightarrow dd[B]A[B]$
 $B \rightarrow [-ddC] + dD$
 $C \rightarrow d + d - dd + d$
 $D \rightarrow d - d + dd - d\}$,
- Nombre d'itération = 12.

Règles de la morphologie de la figure 7:

- $P = \{A \rightarrow dBd$
 $B \rightarrow [-ddB] + ddB$
- Nombre d'itération = 12.

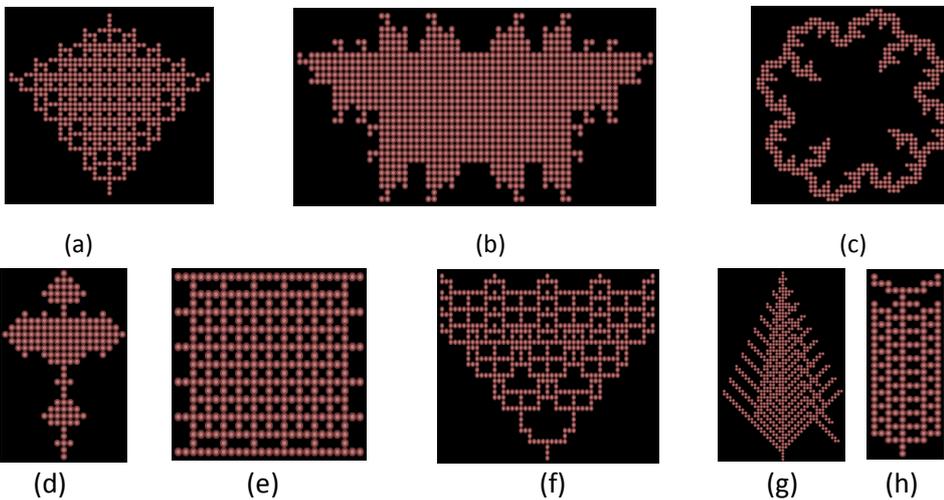


Figure A.1- Panel de morphologies obtenues par évolution.

Bibliographie

[SBLD04] Sanchez, S., Balet, O., Luga, H., and Duthen, Y. (2004). Autonomous virtual actors. LECTURE NOTES IN COMPUTER SCIENCE., pages 68-78.

[AB99] M. Ahluwalia and L. Bull. A Genetic Programming-based Classifier System . In W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, S. R. E. editors , Proceedings of the 1999 Genetic and Evolutionary Computation Conference Workshop Program, Morgan Kaufmann, pages 11-18, 1999.

[AH58] K. J. Arrow and L. Hurwicz. Studies in Linear and Non Linear Programming, chapitre Gradient Methode for Concave programming. Stanford University Press, 1958.

[AK96] P. Angeline and Jr. Kinnear, K., editors . Advances in Genetic Programming 2. MIT Press, Cambridge,1996.

[AK05] Andrew Adamatzky and Maciej Komosinski. Artificial Life Models in Software. Springer-Verlag, 2005.

[Ama05] Saoirse Amarteifio. Interpreting a Genotype-Phenotype Map with Rich Representations in XMLGE. A thesis for the degree of Master of Science. University of Limerick, 2005.

[Art00] W. Arthur. The concept of developmental re-programming and the quest for an inclusive theory of evolutionary mechanism. *Evolution & Development*, 2:49-57, 2000.

[Art02]: W. Arthur. The interaction between developmental bias and natural selection: from centipede segments to a general hypothesis. *Nature Heredity*, 89:239-246, 2002.

[Art04a]: W. Arthur. *Biased Embryos and Evolution*. Cambridge University Press, Cambridge,2004.

[Art04b]: W. Arthur. The effect of development on the direction of evolution: toward a twenty-first century consensus. *Evolution & Development*, 6:282–288, 2004.

[Bak87] J. Baker. Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the Second International Conference on Genetic Algorithms and their Applications*, Hillsdale, USA, pages 14–21. 1987.

[BGB07] J. BACARDIT, D. E. GOLDBERG and M. V. BUTZ. Improving the Performance of a Pittsburgh Learning Classifier System Using a Default Rule. *Lecture Notes In computer Science*. 4399 pages 291-307, 2007.

[BGS00] M. V. Butz, D. E. Goldberg and W. Stolzmann. Introducing a Genetic Generalization Pressure to the Anticipatory Classifier Systems Part I : Theoretical Approach. In Whitley et al.

Also Technical Report 2000005 of the Illinois Genetic Algorithms Laboratory. pages 34-41, 2000.

[Bed92] M.A. Bedau. Philosophical Aspect of Artificial Life. In F. Varela, & P. Bourguine (eds), *Towards A Practice of Autonomous Systems*. Bradford Books/MIT Press, Cambridge , MA, 1992

[Ben99]: Peter Bentley. *Evolutionary Design by Computers*. Academic Press Ltd, London, 1999.

[BK99] P. J. Bentley and S. Kumar. Three ways to grow designs: A comparison of Evolved embryogenies for an evolutionary design problem. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-1999)*, CA: Morgan Kaufmann ,San Francisco, pages 35–43,1999.

[BKHS93] E.J.W. Boers, H. Kuiper, B.L.M. Happel, and I.G. Sprinkhuizen-Kuyper. Designing modular artificial neural networks. Wijshoff, H.A. (ed), *Proceedings of Computing Science in The Netherlands*. Pages 87–96, 1993.

[BKS04] J. Branke, , Kamper, A., and Schmeck. Distribution of evolutionary algorithms in heterogeneous networks. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pages 923-934,2004.

[BL06a] S. Bornhofen and C. Lattaud. Evolutionary design of virtual plant. In *Proceedings of the 2006 International Conference on Computer Graphics and Virtual Reality*, Las Vegas, USA, pages 28-34, 2006.

[BL06b] S. Bornhofen, and C. Lattaud,. Life history evolution of virtual plants : Trading off between growth and reproduction. In Runarsson, T., Beyer, H., Burke, E., Merelo-Guervos, J., Whitley, D., and Yao, X., editors, *Lecture Notes in Computer Science 4193*, pages 808–817, 2006.

[BL07a] S. Bornhofen, and C. Lattaud. Evolution of virtual plants interacting with their environment. In *Proceedings of the 9th International Conference on Virtual Reality*, Laval, France, pages 172–176, 2007.

[BL07b] S. Bornhofen and C. Lattaud, Simulation de communautés de plantes et dynamique des populations. *Techniques et Sciences de l'Information*, 26 :391–420, 2007.

[BL08a] S. Bornhofen and C. Lattaud. Competition and evolution in virtual plant communities : A new modeling approach . *Natural Computing*, 2008.

[BL08b] S. Bornhofen and C. Lattaud, Evolving CSR strategies in virtual plant communities (accepted). In *Proceedings of Artificial Life XI*, Winchester, UK, 2008.

[BL08c] S. Bornhofen, and C. Lattaud. Neutral mutations and junk code in evolving L-systems (accepted). In *Proceedings of the International Conference on Genetic and Evolutionary Computation*, Atlanta, USA,2008.

[Bor08] S. Bornhofen. Emergence de dynamiques évolutives dans une approche multi-agents de plantes virtuelles. PHD thesis,université Paris-Sud, France,2008.

[BP01] M. Butz and M. Pelikan. « Analyzing the Evolutionary Pressures in XCS ». In L. Spector, E. D. Goodman, A.Wu,W. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. H. Garzon, E. Burke (eds), *Genetic and Evolutionary Computation Conference (GECCO'01)*, Morgan Kaufmann, San Francisco, California, USA, pages 935-942, 2001.

- [BP01]: J.C. Bongard and R. Pfeifer. Repeated structure and dissociation of genotypic and phenotypic complexity in artificial ontogeny. In L. Spector, E. D. Goodman, A. Wu, W. B. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. H. Garzon, & E. Burke (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference*. CA: Morgan Kaufmann, San Francisco, pages 829-836, 2001.
- [Bro91] Rodney A. Brooks. How to build complete creatures rather than isolated cognitive simulators. In *Architectures for Intelligence*, pages 225-240, 1991.
- [BSD06]: Cedric Buche, Cyril Septseault and Pierre De Loor: Proposition d'un modèle générique pour l'implémentation d'une famille de systèmes de classeurs. *Revue d'Intelligence Artificielle* 20(1): 63-88, 2006.
- [BSG03] M. V. Butz, K.Sastry, D. E. Goldberg. Tournament Selection : Stable Fitness Pressure in XCS, in E. Cantú-Paz, J. A. Foster, K. Deb, D. Davis, R. Roy, U.-M. O'Reilly, H.-G. Beyer, R. Standish, G. Kendall, S. Wilson, M. Harman, J. Wegener, D. Dasgupta, M. A. Potter, A. C. Schultz, K. Dowsland, N. Jonoska, J. Miller (eds), *Genetic and Evolutionary Computation (GECCO-2003)*, LNCS, Springer-Verlag 2724, pages 1857-1869, 2003.
- [CBLD07] S. Cussat-Blanc, H.Luga and Y. Duthen. A developmental model to simulate natural evolution. *International Conference on Computer Graphics and Artificial Intelligence 3IA*, 2007.
- [CBLD08]:Sylvain Cussat-Blanc, Hervé Luga, Yves Duthen. *Artificial Embryogeny and Grid Computing*. Technical Report, 2008.
- [CBLD08]: Sylvain Cussat-Blanc, Hervé Luga, and Yves Duthen. *From Single Cell to Simple Creature Morphology and Metabolism*. *Artificial Life*, ACM, Winchester, 2008.
- [CBLD09] Sylvain Cussat-Blanc, Herve Luga and Yves Duthen. *Cell2Organ: Self-Repairing Artificial Creatures Thanks to a Healthy Metabolism*. In Andy Tyrrell editor, 2009 *IEEE Congress on Evolutionary Computation (IEEE CEC 2009)*, IEEE Press, Trondheim, Norway, 2009.
- [CD06] : A. Chavoya Pena and Y. Duthen. Evolving cellular automata for 2d form generation. In *Proceedings of the Ninth International Conference on Computer Graphics and Artificial Intelligence 3IA'2006*, pages 129–137, 2006.
- [CD07a]: Arturo Chavoya Pena and Yves Duthen. Evolving an artificial regulatory network for 2D cell patterning. In *Proceedings of the 2007 IEEE Symposium on Artificial Life (CI-ALife'07)*, IEEE Computational Intelligence Society, pages 47–53, 2007.
- [CD07b]:Arturo Chavoya Pena and Yves Duthen. An Artificial Development Model for Cell Pattern Generation. In *Australian Conference on Artificial Life (ACAL 2007)*, Springer, Gold Coast, Australia, 2007.
- [CEA07] Nicolas Chaumont, Richard Egli, and Christoph Adami. Evolving virtual creatures and catapults. *Artificial Life*, 13(2):159–187, 2007.
- [CEA08] Nicolas Chaumont, Richard Egli and Christoph Adami . *CiteSeerX - Scientific Literature Digital Library and Search Engine*, 2008
- [Cho57] N. Chomsky. *Syntactic structures*. Mouton, The Hague,1957.

- [CLD08] Cussat-Blanc, S., Luga, H., and Duthen, Y. (2008b). From single cell to simple creature morphology and metabolism. In *Artificial Life XI*, pages 134-141. MIT Press, Cambridge, MA.
- [CLD09a] Cussat-Blanc, S., Luga, H., and Duthen, Y. Cell2organ : Self-repairing artificial creatures thanks to a healthy metabolism. In *Proceedings of the IEEE Congress on Evolutionary Computation (IEEE CEC 2009)*, 2009.
- [CLD09b] Cussat-Blanc, S., Luga, H., and Duthen, Y. Digital organ cooperation : Toward the assembly of a self-feeding organism. In *Proceedings of the European Conference on Artificial Life (ECAL'09)*, 2009.
- [CLD09c] Cussat-Blanc, S., Luga, H., and Duthen, Y. Making a self-feeding structure by assembly of digital organs. In *Proceedings of the Australian Conference on Artificial Life (ACAL'09)*, A paraître. ,2009.
- [CM96] D. Cliff, and G. Miller, Co-evolution of pursuit and evasion II : Simulation methods and results. In *From Animals to Animats IV : Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*. MIT Press- Bradford Books, Cambridge, MA, 1996.
- [CW53] F. Crick et J. Watson. A Structure for Deoxyribose Nucleic Acid. *Nature*, 171 :737-738, 1953.
- [Dar59]: C. Darwin. *The Origin of Species*. John Murray, London, 1859.
- [Daw86] : Dawkins Richard. *L'Horloger Aveugle*, R. Laffont, 1989. Édition originale : *The Blind Watchmaker*, Longman Scientific & Technical, 1986.
- [Daw76] Dawkins, R. (1976). *The Selfish Gene*. Oxford University Press, Oxford.
- [DB96] Frank Dellaert and R.D. Beer, A Developmental Model for the Evolution of Complete Autonomous Agents. In : *SAB '96*, 1996.
- [DB94] M. Dorigo and H. Bersini. A Comparison of Q-Learning and Classifier Systems. *From Animals to Animats 3 : Third International Conference on Simulation of Adaptive Behavior (SAB94)*, pages 248-255, 1994.
- [DDD09]: Nedjma Djezzar, Nour-Eddine Djedi, Sylvain Cussat-Blanc and Yves. Duthen. Un modèle bio-inspiré pour la synthèse de créatures artificielles. *Symposium international IMAGA'09*, Biskra, pages 53-62, 2009.
- [de 02]K. A. de Jong. *Evolutionary Computation*. The MIT Press, 2002.
- [de 99]: Hugo de Garis. Artificial embryology and cellular differentiation. In Peter J. Bentley, editor, *Evolutionary Design by Computers*. Morgan Kaufmann Publishers, Inc., San Francisco, USA, pages 281-295, 1999.
- [de 92]: Hugo de Garis, H. Artificial embryology : The Genetic Programming of an artificial embryo,. John Wiley, New York, pages 373–393, 1992.
- [Deb01] K. Deb. *Multi-objectives optimization using evolutionary algorithms*. Wiley, 2001.
- [DBS07] A. Devert, N. Bredeche, M. Schoenauer. Robust Multi-Cellular Developmental Design. D. Thierens et al., Eds., *Proceedings of GECCO'07*, ACM Press, 2007

- [Dor95] M. Dorigo. « Alecsys and the AutoMouse : Learning to Control a Real Robot by Distributed Classifier Systems ». *Machine Learning*, 19:209-240, 1995.
- [Dou09] Doursat, R. Facilitating evolutionary innovation by developmental modularity and variability. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 683-690. ACM, 2009.
- [DU05] K. Doya and E. Uchibe. The Cyber Rodent Project : Exploration of Adaptive Mechanisms for Self-Preservation and Self-Reproduction. *Adaptive Behavior*, 13(2) :149, 2005.
- [DZB01] P. Ditztrich, J. Ziegler, W. Banzhaf : Artificial chemistries - a review. *Artificial Life* 7, 3, pages 225–275, 2001.
- [Egg97]: P. Eggenberger. Evolving morphologies of simulated 3D organisms based on differential gene expression. In P. Husbands and I. Harvey, editors, *Proceedings of the Fourth European Conference on Artificial Life (ECAL97)*, MIT Press, pages 205–213. 1997.
- [EGHA02] M. Ebner, , A. Grigore, , A. Heffner, , and J. Albert. Coevolution produces an arms race among virtual plants. *Lecture Notes in Computer Science*, 2278 :316– 325,2002.
- [ES93] L. Eshelman and J. D. Schaffer. Real coded genetic algorithms and interval-schemata. In L.D. Whitley, editor, *Foundations of Genetic Algorithms 2* Los Altos, CA. Morgan Kaufman , pages 187-202, 1993
- [EVK95] A.E. Eiben, C.H.M. Van Kemenade , J.N. Kok . « Orgy in the Computer: Multi-Parent Reproduction in Genetic Algorithms. In *ECAL'95. Lecture Notes In Computer Science*, Springer-Verlag London, UK , 929: 934 – 945,1995.
- [EUD05] S. Elfving, E. Uchibe, K. Doya, and H. Christensen. Biologically inspired embodied evolution of survival. *Evolutionary Computation. The 2005 IEEE Congress*, 2005.
- [FB89]: D. Farmer and A. Belin. Artificial life: the coming evolution. In *Artificial Life*, pages 815-840, 1989.
- [FB94]: K. Fleischer and A.H. Barr. A simulation testbed for the study of multicellular development: The multiple mechanisms of morphogenesis. In Langton, C., editor, *Artificial Life III*, Addison-Wesley Redwood City, CA. pages 389–416, 1994.
- [FGV06] Dario Floreano, Robert L. Goldstone, and Alessandro Vespignani. *Proceedings of the Artificial Life X Conference (Alife 10)*, MIT Press, pages 15-21, 2006.
- [Fle96]: K. Fleischer. Investigations with a multicellular developmental model. In Langton, C. G. and Shimohara, T., editors, *Artificial Life V: Proceedings of the Fifth International Conference on the Synthesis and Simulation of Living Systems*, The MIT Press/Bradford Books, Cambridge, MA., pages 229–236,1996.
- [FOW66] L. Fogel, A. Owens, and M. Walsh. *Artificial Intelligence through Simulated Evolution*. John Wiley, New York, 1966.
- [Gar70] Gardner, M. Mathematical games. the fantastic combinations of John Conway's new solitaire game "life". *Scientific American*, 223(4) :120–123,1970.
- [GD91] D. Goldberg and K. Deb. A comparative analysis of selection schemes used in genetic algorithms. In Rawlins, G., editor, *Foundations in Genetic Algorithms*. Morgan Kaufmann, San Mateo, 1991.

- [Gea06]: Nicholas Geard . Artificial Ontogenies: A Computational Model of the Control and Evolution of Development School of Information Technology and Electrical Engineering. PHD thesis. The University of Queensland, 2006.
- [Gel95] M. Gell-Mann. "Le quark et le jaguar. Voyage au cœur du simple et du complexe", Albin Michel Sciences, 1995.
- [Ger04] Carlos Gershenson. Introduction to random boolean networks. In M. Bedau, P. Husbands, T. Hutton, S. Kumar, and H. Susuki, editors, Workshop and Tutorial Proceedings, Ninth International Conference on the Simulation and Synthesis of Living Systems (ALife IX), pages 160–173, 2004.
- [GMS05] P. Gérard, J.-A. Meyer, O. Sigaud « Combining Latent Learning with Dynamic Programming in MACS ». European Journal of Operational Research, 160:614-637, 2005.
- [GL97] F. Glover, M. Laguna and al. Tabu search. Springer, 1997.
- [Gol89] : D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, 1989.
- [Gru93] F. Gruau. Genetic Synthesis of Modular Neural Networks. Proceedings of the 5th International Conference on Genetic Algorithms, Morgan Kaufmann Publishers Inc, San Francisco, CA, USA Pages 318–325, 1993.
- [Gru94 a] F. Gruau. Genetic micro programming of neural networks. In K. E. Kinnear Jr., editor, Advances in GP, MIT Press, pages 495–518, 1994.
- [Gru94 b] F. Gruau. Neural network synthesis using cellular encoding and the genetic algorithm. Doctoral dissertation, Ecole Normale Supérieure de Lyon, France ,1994.
- [Gru95] F. Gruau, « Automatic definition of modular neural networks». Adaptive Behaviour, 3(2): 151-183, 1995.
- [GSS02] P. Gérard, W. Stolzmann, O. Sigaud. « YACS : a new Learning Classifier System with Anticipation ». Journal of Soft Computing : Special Issue on Learning Classifier Systems, 6(3-4):216-228, 2002.
- [Han92] J.S. Hanan, parametric L-system and their application to the modeling and visualization of plant. Dissertation, University of Regina,1992
- [Hay01] N. K. Hayles. Review of Silicon second nature: Culturing artificial life in a digital world, by Stefan Helmreich. Artificial Life, 7(4) :425–428, 2001.
- [Heg03] Olivier Heguy. Architecture comportementale pour l'émergence d'activités coopératives en environnement virtuel. Ph.D. thesis, Université Paul Sabatier, Toulouse, 2003.
- [Heu98] Jean-Claude Heudin. L'évolution au bord du chaos. Hermes, Paris, France, 1998.
- [HML05] Herrera, J., Huedo, E., Montero, R. S., and Llorente, I. M. A grid-oriented genetic algorithm. in Advances in Grid Computing - EGC 2005, pages 315-322,2005.
- [Hog00a] P. Hogeweg. Evolving mechanisms of morphogenesis: on the interplay between differential cell adhesion and cell differentiation. Journal of Theoretical Biology, 203:317–333, 2000.

- [Hog00b] P. Hogeweg. Shapes in the shadow: Evolutionary dynamics of morphogenesis. *Artificial Life*, 6:85–101,2000.
- [Hol62] J. Holland. Outline for a logical theory of adaptative systems. *Journal of the association of computing machinery*, 3 :297–314, 1962.
- [Hol75] J. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, 1975.
- [Hol76] J. H. Holland. « Adaptation ». *Progress in theoretical biology* (vol. 4), 1976.
- [Hol86] J. H. Holland, « Escaping Brittleness : The possibilities of general-purpose learning algorithms applied to parallel rule-based systems ». *Machine Learning, An Artificial Intelligence Approach* (volume II), Morgan Kaufmann, 1986.
- [HP01a] G. Hornby, and J. Pollack. The advantages of generative grammatical encodings for physical design. In *Proceedings of the Congress on Evolutionary Computation*, pages 600–607 , Seoul, Korea 2001.
- [HP01b] Gregory Hornby and Jordan B. Pollack. Evolving l-systems to generate virtual creatures. *Computers & Graphics*, 25(6):1041–1048, 2001.
- [HP02] G. Hornby and J. Pollack. Creating high-level components with a generative representation for body-brain evolution, 2002.
- [HR78] J. Holland and J. Reitman « Cognitive systems based on adaptive algorithms ». In D. A. Waterman, F. Hayes-Roth (eds), *Pattern-directed inference systems*, New York : Academic Press, 1978. Reprinted in : *Evolutionary Computation. The Fossil Record*. David B. Fogel (Ed.) IEEE Press, 1998. ISBN : 0-7803-3481-7.
- [HSD02] O. Heguy, C. Sanza, A. Berro, Y. Duthen. « GXCS : A generic Classifier System and its application in a Real Time Cooperative Behavior Simulations ». *International Symposium and School on Advanced Distributed System (ISADS'02)*, 2002.
- [Hut07] T.J. Hutton. Evolvable self-reproducing cells in a two-dimensional artificial chemistry . *Artificial Life*, 1(13), Pages 11 - 30, 2007.
- [Ing05] D. E. Ingber, Mechanical control of tissue growth: Function follows form. *Proceedings of the National Academy of Science, USA*, 102(33):11571–11572, 2005.
- [KA95] Shigeru Kondo and Rihito Asai, A reaction-diffusion wave on the skin of the marine angelfish *Pomacanthus*, *Nature* 376: 765-768, 1995.
- [Kau69] Stuart A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22 :437–467, 1969.
- [Kau71] S.A. Kauffman. Gene regulation networks: a theory for their global structure and behaviours. *Current Topics in Developmental Biology*, 6:145–182, 1971.
- [Kau93] Stuart A. Kauffman. *The Origins of Order : Self-Organization and Selection in Evolution*. Oxford University Press, 1993.
- [KBO3a]: Sanjeev Kumar and Peter J. Bentley. Computational Embryology: Past, Present and Future. *Computational embryology*. In *Advances in evolutionary computing : theory and applications*,. Springer-Verlag New York, Inc., New York, NY, USA, pages 461-477, 2003.

- [KB03b] S. Kumar and P.J. Bentley. An Introduction to Computational Development. In S. Kumar and P.J. Bentley, editors, *On Growth, Form and Computers*, chapter 1, Elsevier pages 1-44, 2003.
- [KB04] Paul Dwight Kuo and Wolfgang Banzhaf. Small world and scale-free network topologies in an artificial regulatory network model, Boston, USA, J. Pollack, M. Bedau, P. Husbands, T. Ikegami, R. Watson (Eds.) . *Artificial Life IX (ALIFE-9)*, MIT Press, Cambridge, pages 404-409, 2004.
- [KGV83] S. Kirkpatrick, C. D. Jr. Gelatt and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220 :671–680, 1983.
- [Kit90] Hiroaki Kitano. Designing neural networks using genetic algorithms with graph generation system. *Complex Systems*, 4 :461–476, 1990.
- [Kit95] Hiroaki Kitano. A simple model of neurogenesis and cell differentiation based on evolutionary large-scale chaos. *Artificial Life*, 2(1) ,1995.
- [KLB04] Paul Dwight Kuo, Andre Leier, and Wolfgang Banzhaf. Evolving dynamics in an artificial regulatory network model. In X. Yao, E. Burke, J.A. Lozano, J. Smith, J.J. Merelo-Guervos, J.A. Bullinaria, J. Rowe, P. Tino, A. Kaban, and H.-P. Schwefel, editors, *Proceedings of the Parallel Problem Solving from Nature Conference (PPSN-04)*, LNCS 3242, Birmingham, UK, Springer, Berlin , pages 571-580,2004
- [KNSQ06] Johannes F. Knabe, Chrystopher L. Nehaniv, Maria J. Schilstra, and Tom Quick. Evolving biological clocks using genetic regulatory networks. In Luis M. Rocha, Larry S. Yaeger, Mark A. Bedau, Dario Floreano, Robert L. Goldstone, and Alessandro Vespignani, editors, *Proceedings of the Artificial Life X Conference (Alife 10)*, MIT Press, pages 15-21, 2006.
- [KNS07] Johannes F. Knabe, Chrystopher L. Nehaniv, and Maria J. Schilstra. The essential motif that wasn't there : Topological and lesioning analysis of evolved generic regulatory networks. In *Proceedings of the 2007 IEEE Symposium on Artificial Life (CI-ALife'07)*. IEEE Computational Intelligence Society, pages 47-53, 2007.
- [KSN08] Knabe, J., Schilstra, M., and Nehaniv, C. (2008). Evolution and morphogenesis of differentiated multicellular organisms : autonomously generated diffusion gradients for positional information. *Artificial Life XI*, 11 :321.
- [Kom00] Maciej Komosinski. The world of framsticks: Simulation, evolution, interaction. In *VW'00: Proceedings of the Second International Conference on Virtual Worlds*, Springer-Verlag, London, UK, pages 214–224, 2000.
- [Kom03] Maciej Komosinski. The Framsticks system: versatile simulator of 3D agents and their evolution. *Kybernetes: The International Journal of Systems & Cybernetics*, 32:156–173, 2003.
- [Kom05] Maciej Komosinski. Framsticks: a platform for modeling, simulating and evolving 3D creatures. In *Artificial Life Models in Software*, Springer-Verlag, pages 37–66, 2005.
- [Koz92] J. Koza. *Genetic Programming : On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge. 1992.

- [Lan89] C.G. Langton. Proceedings of the Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems (ALIFE '87), Los Alamos, NM, USA, September 1987, Santa Fe Institute Studies in the Sciences of Complexity, Addison-Wesley, Ed, pages 15-18. 1989.
- [Lan98]: Chris Langton. Artificial life. In *Artificial Life*, pages 1–47, 1998.
- [Las08] Nicolas Lassabe. Morphogenèse et Evolution de Créatures Artificielles. Phd thesis, Université de Toulouse 1, France, 2008.
- [LG02] X. Llorà, J.M. Garrell. « Co-evolving Different Knowledge Representations with fine-grained Parallel Learning Classifier Systems ». In W. B. Langdon, E. Cantu-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M. A. Potter, A. C. Schultz, J. F. Miller, E. Burke, N. Jonoska (eds), *Proceeding of the Genetic and Evolutionary Computation Conference (GECCO2002)*, Morgan Kaufmann, 2002.
- [Lin68]: Aristid Lindenmayer. Mathematical models for cellular interaction in development, Parts I and II. *Journal of Theoretical Biology*, 18 :280–315, 1968.
- [Lin74] A. Lindenmayer. Adding continuous components to L-systems. In Rozenberg, G. and Salomaa, A., editors, *L Systems, Lecture Notes in Computer Science 15*. Springer, Berlin. 1974.
- [LL02] P. Larrañaga and J.A. Lozano. *Estimation of Distribution Algorithms : A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, 2002.
- [LLD07a] Nicolas Lassabe, Hervé Luga, and Yves Duthen. Interaction of evolving artificial creatures. In *International Conference on Enactive Interfaces*, pages 137–140, 2007.
- [LLD07b] Nicolas Lassabe, Hervé Luga, and Yves Duthen. A new step for evolving creatures. In *IEEE-ALife'07, IEEE*, pages 243–251, 2007.
- [LLD07c] Nicolas Lassabe, Hervé Luga, and Yves Duthen. New skills for evolving artificial creatures. *International Journal of Information Technology and Intelligent Computing, IEEE-CIS*, 2007.
- [LP99] P.-L. Lanzi, A. Perrucci. « Extending the Representation of Classifier Conditions Part II : From Messy Coding to S-Expressions », in W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, S. R. E. (eds), *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 99)*, Morgan Kaufmann, Orlando, FL, pages 345-352, 1999.
- [LP00] Hod Lipson and Jordan B. Pollack. Automatic design and manufacture of artificial life forms. *Nature*, 406, 2000.
- [LR75] A. Lindenmayer, G. Rozenberg, (eds.), *Automata, Languages, Development*, North-Holland, 1975.
- [LS08] S. Landau, and O. Sigaud. A Comparison between ATNoSFERES and LCSs on non-Markov problems. *Information Sciences*, pages 4482-4500, 2008.
- [LSS05] S. Landau, O. Sigaud, and M. Schoenauer. ATNoSFERES revisited. *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-2005* ACM Press, publisher, Washington DC, pages 1867-1874. 2005.
- [Lug09] Hervé Luga. Expérimentation virtuelles : vie artificielle pour la création de formes et de comportement. *Journées d'informatique musicales - JIM 2009*, Grenoble, France, 2009.

- [Man06] Daniel Mange .La Complexité, vertiges et promesses. Livre. IV De la vie in silico Entretien avec Daniel Mange. Editions le Pommier, France, 2006.
- [MB03] J. F. Miller and W. Banzhaf. Evolving the program for a cell: from french flags to boolean circuits. In S. Kumar and P. J. Bentley, editors, *On Growth, Form and Computers*. Academic Press, 2003.
- [MC05] T. Miconi and A. Channon. A virtual creatures model for studies in artificial evolution, 2005.
- [MC06a] T. Miconi and A. Channon. An improved system for artificial creatures evolution, 2006.
- [MC06b] T. Miconi and A. Channon. The n-strikes-out algorithm: A steady-state algorithm for coevolution. In the IEEE Congress on Evolutionary Computation, 2006.
- [Men07] : G. Mendel. « Les lois de l’hérédité recherches sur les végétaux ». Bulletin scientifique de la France et de la Belgique t.41 :371, 419, 1907. Tard de “Versuche über Pflanzen-Hybriden.” 1866 par A. Chappellier.
- [Mei95] Hans Meinhardt, *The Algorithmic Beauty of Sea Shells*, Springer, Berlin ,1995.
- [Mic07] Thomas Miconi. THE ROAD TO EVERYWHERE Evolution, complexity and progress in Nature and in computers. PhD thesis, The University of Birmingham, 2007.
- [Mil04] J. F. Miller. Evolving a self-repairing, self-regulating, french flag organism. In GECCO, pages 129–139, 2004.
- [MLH02] M .Métivier, C. Lattaud, and J.C. Heudin. A stress-based speciation model in life drop. In *Proceedings of the 8th International Conference On Artificial Life*, MIT Press, pages 121– 126, 2002.
- [Mou05] J.-B. Mouret. Concepts fondamentaux des algorithmes évolutionnistes. *Linux Magazine France*. Pages 34-41. 2005
- [MP43] Warren S. McCulloch and Walter Pitts. A logical calculus of ideas immanent in nervous activity. *Bulletin of mathematical biophysics*. Reprinted in McCulloch, W. S., *Embodiments of mind*, MIT Press Cambridge, MA, ,5 :115–133, 1943.
- [MP96] R. Mech, and P. Prusinkiewicz. Visual models of plants interacting with their environment. In *Computer Graphics Proceedings, ACM SIGGRAPH’96*, New Orleans, USA, pages 397–410, 1996
- [MT00] J. F. Miller and P. Thomson. Cartesian genetic programming. In P. et al, editor, *EuroGP’00, LNCS 1802*, Springer-Verlag, pages 121–132, 2000.
- [Nos03] H. Noser. A 3D Animation System for Functional Simulation, Visualization, and Control of Dynamic Processes Based on Parallel Rewriting, University of Zurich, 2003.
- [Ohn70] Ohno, S.. *Evolution by Gene Duplication*. Springer, New York, 1970.
- [Ohn72] : Ohno, S. So much “junk” DNA in our genome. *Brookhaven Symposia in Biology*, 23 :366–370, 1972.
- [Ovi92] Ovide. *Les Métamorphoses*. Gallimard, Paris. 1992

- [OI99] ONO N., IKEGAMI T. : Model of self-replicating cell capable of self-maintenance. In ECAL '99 : Proceedings of the 5th European Conference on Advances in Artificial Life, Springer-Verlag, pages 399–406. London, UK, 1999.
- [PHLF03] J. B. Pollack, G. S. Hornby, Hod Lipson Lipson, and P. Funes. Computer creativity in the automatic design of robots. *LEONARDO*, 2(36), 2003.
- [PL90] P. Prusinkiewicz, and A. Lindenmayer. *The Algorithmic Beauty of Plants*. Springer, Berlin, 1990.
- [Pru86] P. Prusinkiewicz. Graphical applications of L-systems. In *Proceedings of Graphics Interface '86 -Vision Interface '86*, CIPS ,pages 247-253, 1986.
- [Pru87] P. Prusinkiewicz. Applications of L-systems to computer imagery. In H. Ehrig, M. Nagl, A. Rosenfeld, and G. Rozenberg, editors, *Graph grammars and their application to computer science; Third International Workshop, Lecture Notes in Computer Science 291*, Springer-Verlag, Berlin, pages 534-548, 1987.
- [Ram07] Marco A. Ramos. *Etude et proposition d'un système comportementale autonome anticipatif*. Ph.D. thesis, Université de Toulouse I, 2007.
- [Ray91a] Thomas S. Ray. An approach to the synthesis of life. In *Artificial Life II*, pages 371–408, 1991.
- [Ray91b] Thomas S. Ray. Evolution and optimization of digital organisms. In Billingsley K. R., E. Derohanes, H. Brown, III (eds.), *Scientific Excellence in Supercomputing: The IBM 1990 Contest Prize Papers*, Athens, 1991.
- [Ray00] Thomas S. Ray. Aesthetically evolved virtual pets. In *Artificial Life 7 workshop proceedings*, pages 158–161, 2000.
- [RCNA03] S. Rasmussen, L. Chen, M. Nilsson & S. Abe. Bridging nonliving and living matter. *Artificial Life*, 3(9), pages 269 - 316, 2003.
- [Rei99]: Torsten Reil. Dynamics of gene expression in an artificial genome – implications for biological and artificial ontogeny. In *Proceedings of the 5th European Conference on Artificial Life (ECAL)*, Springer Verlag, New York, NY pages 457–466, 1999.
- [Rec73] I. Rechenberg. *Evolutions strategie : Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann Holzboog, Stuttgart, 1973.
- [Ren02] Jean-Philippe Rennard. *Vie Artificielle : Où la biologie rencontre l'informatique*. Vuibert Informatique, Paris, France, 2002.
- [Rey91] Craig W. Reynolds. Steering behaviors for autonomous characters. In *Game Developer Conference*, pages 763–782, 1999.
- [Sch95] H. P. Schwefel. *Numerical Optimisation Of Computer Models*. John Wiley & Sons, New York, 1981.1995-2nd edition,1995
- [Rya00] C. Ryan. *Automatic Re-engineering of Software Using Genetic Programming*. Kluwer Academic Publishers, Boston. 2000.
- [Sanza, 2001] Sanza, C. (2001). *Evolution d'entités virtuelles coopératives par système de classifieurs*. Ph.D thesis.
- [SBLD04] Sanchez, S., Balet, O., Luga, H., and Duthen, Y. (2004). Autonomous virtual actors. *LECTURE NOTES IN COMPUTER SCIENCE.*, pages 68-78.

- [Sch75] H. Schwefel. Evolutions strategie und numerische Optimierung. PhD thesis, Technische Universität, Berlin, 1975.
- [She97] M. Shelley. Frankenstein. Gallimard, Paris, 1997.
- [SJ91] W. Spears, and K.D. Jong. An analysis of multi-point crossover. In Rawlins, G., editor, Foundations of Genetic Algorithms. Morgan Kaufmann, San Mateo, 1991.
- [Sig07] O. Sigaud, Les systèmes de classeurs : un état de l'art. Revue d'Intelligence Artificielle Hermès, publisher, pages 75-106, 2007.
- [Sim91] Karl Sims. Artificial evolution for computer graphics. In SIGGRAPH, pages 319–328, 1991.
- [Sim94a] Karl Sims. Evolving 3d morphology and behavior by competition. Artificial Life, 1(4):353–372, 1994.
- [Sim94b] Karl Sims. Evolving 3d morphology and behavior by competition. In Artificial Life IV., Artificial Life, MIT Press, pages 28–39, 1994.
- [Sim94c] Karl Sims. Evolving virtual creatures. In Siggraph'94, pages 15–22, 1994.
- [SM03]: Kenneth O. Stanley, Risto Miikkulainen. A Taxonomy for Artificial Embryogeny. The MIT Press Journals.2003.
- [Smi80] S. Smith, «A learning system based on genetic algorithms», PhD. Thesis, Computer Science Department, University of Pittsburgh, 1980.
- [SQ79] A. L. Szilard and R. E. Quinton. An interpretation for DOL systems by computer graphics. The Science Terrapin, 4:8-13, 1979.
- [STK05]: Finlay Stewart, Tim Taylor, and George Konidaris. Metamorph Experimenting with genetic regulatory networks for artificial development. In ECAL, pages 108–117, 2005.
- [Sto98] W. Stolzmann. « Anticipatory Classifier Systems ». In J. Koza, W. Banzhaf, K. Chellapilla, K. Deb, M. Dorigo, D. B. Fogel, M. H. Garzon, D. E. Goldberg, H. Iba, R. Riolo (eds), Genetic Programming, Morgan Kaufmann Publishers, Inc., San Francisco, CA, pages 658-664, 1998.
- [Sys89] G. Syswerda. Uniform crossover in genetic algorithms. In Proceedings of the Third International Conference on Genetic Algorithms, Los Altos, USA, pages 2–9, 1989.
- [Tay00] Tim Taylor. Artificial life techniques for generating controllers for physically modelled characters. In Proceedings of the First International Conference on Intelligent Games and Simulation, 2000.
- [Tho98] J. Thornley. Modelling shoot :root relations : The only way forward ?. Annals of Botany, 81 :165–171, 1998.
- [TM00] Tim Taylor and Colm Massey. Recent developments in the evolution of morphologies and controllers for physically simulated creatures. Artificial Life, 7(1):77–87, 2000.
- [TT94] X. Tu, and D. Terzopoulos, Artificial fishes : Physics, locomotion, perception, behavior. In Proceedings of the 21st annual conference on Computer graphics and interactive techniques, pages 43-50. ACM New York, NY, USA, 1994.

- [TTS99] K. Takadama, T. Terano, k. Shimohara. « Can Multiagents Learn in Organization ? – Analyzing Organizational Learning-Oriented Classifier System ». IJCAI'99 Workshop on Agents Learning about from and other Agents, 1999.
- [Tur36] A.M Turing, “On Computable Numbers, with an application to the Entscheidungsproblem”, Proceedings of the Mathematical Society, Série 2, Vol. 42, pages 230-265. 1936.
- [Tur50] A.M Turing, “Computing Machinery and Intelligence”, Mind, N°59, 1950.
- [Tur52] : Alan M. Turing. The chemical basis of morphogenesis. Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences, 237(641) :37-72, 1952.
- [Ula62] S. Ulam. On some mathematical properties connected with patterns of growth of figures. In Proceedings of Symposia on Applied Mathematics. American Mathematical Society, 14: 215–224,1962.
- [Ven96] Jeffrey Ventrella. Sexual swimmers: Emergence morphology and locomotion behavior. In Artificial Life IV, pages 484–493, 1996.
- [Ven98a] Jeffrey Ventrella. Attractiveness vs. efficiency (how mate preference affects location in the evolution of artificial swimming organisms). In ALIFE: Proceedings of the sixth international conference on Artificial life, MIT Press, Cambridge, MA, USA. pages 178– 186, 1998.
- [Ven98b] Jeffrey Ventrella. Designing emergence in animated artificial life worlds. In VW '98: Proceedings of the First International Conference on Virtual Worlds, Springer-Verlag , London, UK pages 143–155,1998.
- [Ven05] Jeffrey Ventrella. Artificial Life Models in Software, chapter Gene Pool: Exploring the Interaction Between Natural Selection and Sexual Selection. Springer-Verlag, 2005.
- [Von66] J. Von Neumann. Theory of Self-Reproducing Automata. edited and compiled by A.W. Burks, University of Illinois Press, Urbana,1966.
- [Von05] H. Von Koch. Une méthode géométrique élémentaire pour l'étude de certaines questions de la théorie des courbes planes. Acta Mathematica, 30 :145–174. 1905
- [Wat89] C.J.C.H. Watkins. Learning with Delayed Rewards, PhD thesis, Psychology Department, University of Cambridge, England, 1989.
- [Whi89] D. Whitley. The genitor algorithm and selection pressure : Why rank-based allocation of reproductive trials is best. In Proceedings of the Third International Conference on Genetic algorithms, San Mateo, USA, pages 116–121, 1989.
- [Wil94] S. W. Wilson, « ZCS, a Zeroth level Classifier System », Evolutionary Computation, 2(1) :1-18, 1994.
- [Wil95] S.W. Wilson, « Classifier Fitness Based on Accuracy », Evolutionary Computation, 3(2): 149-175, 1995.
- [Wil96] W.S. Wilson, « Explore/exploit strategies in autonomy », From Animals to Animats 4 : Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (SAB96), pages 325-332, 1996.
- [Wil01] S.W. Wilson, « Function Approximation with a Classifier System », in L. Spector, G. E. D., A. Wu, W. B. Langdon, H. M. Voigt, M. Gen (eds), Proceedings of the Genetic and Evolutionary Computation Conference (GECCO01), Morgan Kaufmann, pages 974-981, 2001.

[Wol83] S. Wolfram, "Statistical mechanics of cellular automata," *Reviews of Modern Physics*, 55:601–644, 1983.

[Wol68] Lewis Wolpert. The French flag problem : a contribution to the discussion on pattern development and regulation. In C. Waddington, editor, *Towards a Theoretical Biology*, Edinburgh University Press, New York, NY, USA, pages 125-133, 1968.

[Yae94] L. Yaeger. Computational Genetics, Physiology, Metabolism, Neural Systems, Learning, Vision, and Behavior or Poly World : Life in a New Context. In *Sante Fe Institute studies in the sciences of complexity*. Addison-Wesley, 17: 263–263, 1994.

[YS08] L. S. Yaeger and O. Sporns. Evolution of neural structure and complexity in a computational ecology 2008(ALIFE'08). *IEEE Symposium on Artificial Life*, 2008.

