

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider Biskra
Faculté des Sciences Exactes, des sciences de la Nature et de la Vie
Département d'Informatique

N° d'ordre :.....
Série :.....



Mémoire

Présenté en vue de l'obtention du diplôme de Magister en Informatique

Option: **Synthèse d'images et vie artificielle**

Titre :

LA PHYLOGENESE POUR LA CREATION DE CREATURES ARTIFICIELLES

Par :

M^{elle} OUANNES Nesrine

Soutenu le : 04/05/2010

Devant le jury :

Dr. F. CHERIF	MCA	Université de Biskra	Président
Pr. N. DJEDI	PR	Université de Biskra	Rapporteur
Dr. M. C. BABAHENINI	MCA	Université de Biskra	Examineur
Dr. A. MOUSSAOUI	MCA	Université de Sétif	Examineur
Dr. Z. E. BAARIR	MCA	Université de Biskra	Examineur

Remerciements

Je remercie, avant toute personne, mon encadreur Monsieur le Professeur NourEddine DJEDI, pour ses précieux conseils et surtout pour sa disponibilité et sa grande aide.

Je remercie, également, Monsieur le Professeur Yves DUTHEN et le Professeur Jean Pierre JESSEL pour leur accueil au sein de l'IRIT et plus spécialement Monsieur le Docteur Hervé LUGA qui m'a bien dirigé et m'a aidé tout au long de mon premier et deuxième séjour au sein de l'équipe VORTEX de l'IRIT à Toulouse (France).

Je tiens aussi à exprimer mes sincères remerciements aux membres de mon jury, Dr CHERIF Foudil, Dr BABAHENINI Med Chaouki et Dr MOUSSAOUI A/Ouahab, pour les conseils qu'ils m'ont émis durant la lecture de mon mémoire ainsi que les remarques qu'ils m'ont fournies durant la soutenance de mon magister.

Je remercie également Dr. Nicolas LASSABE et Dr. Stéphane SANCHEZ pour leurs conseils et leurs remarques pertinentes.

Je tiens aussi, à remercier Monsieur Nicolas CHAUMONT de l'université de Californie pour ses précieux conseils.

Je remercie, enfin ma famille dans tout son ensemble pour l'aide et le soutien qu'on m'a donné et surtout pour l'appui de ma mère qui m'as bien motivé pour faire mon mieux.



Table de matières

RÉSUMÉ	12
ABSTRACT	7
INTRODUCTION GÉNÉRALE	1
VIE ARTIFICIELLE	8
1. LES ORIGINES DE LA VIE ARTIFICIELLE.	8
1.1. HISTOIRE DE LA VIE ARTIFICIELLE.	8
1.2. QU'EST CE QUE LA VIE ?	11
1.3. DEFINITION DE LA VIE ARTIFICIELLE.	12
2. EVOLUTION	14
2.1. APPARITION DE L'ÉVOLUTION ARTIFICIELLE.	14
2.2. ÉCART ENTRE L'ÉVOLUTION ET L'ÉVOLUTION ARTIFICIELLE.....	14
2.3. QU'EST CE QUE L'ÉVOLUTION?.....	15
2.4. LA SELECTION NATURELLE.....	17
3. SYSTEMES EVOLUTIONNISTES.	18
3.1. HISTOIRE DES ALGORITHMES EVOLUTIONNISTES DE DARWIN A HOLLAND.	18
3.2. PRINCIPE DES ALGORITHMES EVOLUTIONNISTES	21
3.3. LES ALGORITHMES GENETIQUES.....	23
Le reste stochastique.....	26
La roulette pipée.....	26
La sélection par rang.....	27
La sélection par tournoi.....	27
Croisement à un point de coupe.....	29
Croisement à deux points de coupe.....	29
Croisement uniforme.....	30



3.4.	LA PROGRAMMATION GENETIQUE.....	32
3.5.	LES STRATEGIES D'ÉVOLUTION	34
4.	LES RESEAUX DE NEURONES.	34
4.1.	DEFINITION DE RESEAUX DE NEURONES.....	35
4.2.	LES RESEAUX DE NEURONES NATURELS.....	35
4.3.	LES RESEAUX DE NEURONES ARTIFICIELS.....	36
4.4.	L'APPRENTISSAGE DANS LES RESEAUX DE NEURONES.....	38
4.5.	LES RESEAUX A PROPAGATION AVANT.....	38
5.	LES L-SYSTEMES.	41
6.	EVOLUTION DE CONTROLEURS POUR UN AGENT AUTONOME.	42
6.1.	LES GENERATEURS DE CONTROLE DE PATTERN.....	43
6.2.	LES SYSTEMES DE CLASSEURS.....	44
6.3.	RESEAUX DE NEURONES EVOLUTIFS.....	45
7.	CONCLUSION.	47
	LES CREATURES ARTIFICIELLES	48
1.	DEFINITION DE CREATURES ARTIFICIELLES	48
2.	MORPHOGENESE ARTIFICIELLE.....	49
2.1.	ÉVOLUTION DE LA MORPHOLOGIE SEULE.....	50
2.2.	ÉVOLUTION DE LA MORPHOLOGIE ET DU CONTROLEUR.....	50
2.2.1.	LES CREATURES DE KARL SIMS.....	50
2.2.2.	LES CREATURES DE KOMOSINSKI.....	55
2.2.3.	LES CREATURES DE NICOLAS LASSABE.....	55
2.2.4.	LES CREATURES DE THOMAS MICONI.....	56
2.2.5.	LES CREATURES DE NICOLAS CHAUMONT.....	57
2.3.	ÉVOLUTION DU CONTROLEUR AVEC UNE MORPHOLOGIE FIGÉE.....	58
2.3.1.	LES CREATURES DE BONGARD ET LIPSON.....	58
2.3.2.	LES CREATURES DE LARY GRITZ.....	59
2.3.3.	LES TRAVAUX DE HERVE LUGA.....	60
2.3.4.	AUTRES TRAVAUX.....	61
2.4.	ÉVOLUTION DU CONTROLEUR AVEC UNE MORPHOLOGIE DE BIPÈDE.....	62
2.4.1.	LES BIPÈDES DE KRISTER WOLFF.....	62
2.4.2.	LES HUMANOÏDES DE MAURIZIO CONVENTI.....	63
2.4.3.	LES ROBOTS HUMANOÏDES DE MALACHY EATON.....	64

3. EMBRYOGENESE ARTIFICIELLE.....	64
3.1. LES CREATURES D'ARTURO CHAVOYA.....	65
3.2. LES CREATURES DE SYLVAIN CUSSAT-BLANC.....	66
4. LA ROBOTIQUE EVOLUTIONNISTE.....	67
4.1. LE PROJET GOLEM DE HOD LIPSON.....	67
4.2. LES ROBOTS DE GREGORY S. HORNBLY.....	69
4.3. LES ROBOTS DE VICTOR ZYKOV 1.....	70
4.4. LES ROBOTS DE VICTOR ZYKOV 2.....	71
5. SYNTHESE.....	72
6. CONCLUSION.....	72
LE MODELE PROPOSE.....	74
1. ENVIRONNEMENTS.....	75
A. LES ECOSYSTEMES.....	75
B. LES ENVIRONNEMENTS TRIDIMENSIONNELS.....	76
C. L'ENVIRONNEMENT CHOISI.....	77
2. LES MOTEURS PHYSIQUES.....	77
2.1. OPEN DYNAMICS ENGINE.....	78
3. BIBLIOTHEQUE GRAPHIQUES.....	82
3.1. OPENGL (OPEN GRAPHIC LIBRARY).....	82
4. ARCHITECTURE GLOBALE DU SYSTEME.....	82
4.1. ENVIRONNEMENT SIMULE.....	83
4.2. LA MORPHOLOGIE DE LA CREATURE PROPOSEE.....	84
4.3. LE CONTROLEUR DE L'HUMANOÏDE.....	88
4.3.1. <i>Les réseaux de neurones récurrents.....</i>	<i>88</i>
4.3.2. <i>Ensemble des senseurs et d'effecteurs.....</i>	<i>91</i>
4.4. STRATEGIES D'EVOLUTION (SE).....	93
4.4.1. <i>La création de la population.....</i>	<i>96</i>
4.4.2. <i>L'élitisme.....</i>	<i>97</i>
4.4.3. <i>Evaluation.....</i>	<i>97</i>
4.4.4. <i>Sélection.....</i>	<i>99</i>
4.4.5. <i>Modification.....</i>	<i>99</i>
4.4.6. <i>Remplacement.....</i>	<i>101</i>
4.5. PARAMETRES D'UN ALGORITHME EVOLUTIONNAIRE.....	101

4.5.1.	<i>Taille de la population.</i>	102
4.5.2.	<i>Conditions d'arrêt.</i>	102
4.5.3.	<i>Croisement.</i>	102
5.	RECAPITULATIF.	103
6.	CONCLUSION.	104
	IMPLÉMENTATION ET RÉSULTATS.	105
1.	LE LANGAGE DE PROGRAMMATION.	105
2.	LE MOTEUR PHYSIQUE DE L'APPLICATION.	106
3.	L'ARCHITECTURE GLOBALE DE L'IMPLEMENTATION.	107
4.	LES ALGORITHMES UTILISES.	108
4.1.	LA FONCTION DE BOUCLE DE SIMULATION.	109
4.2.	LA FONCTION DE L'ALGORITHME EVOLUTIONNAIRE.	110
4.3.	UN EXEMPLE D'UNE SIMULATION SOUS ODE.	111
5.	LES EXPERIMENTATIONS.	112
5.1.	MISE EN ŒUVRE DE LA SIMULATION.	113
5.1.1.	ENVIRONNEMENT VIRTUEL.	113
5.1.2.	CREATURE ARTIFICIELLE.	114
5.2.	CONDITIONS INITIALES DE L'APPLICATION.	114
6.	RESULTATS OBTENUS.	115
6.1.	LOCOMOTION SUR UN ESPACE PLAT.	115
6.1.1.	ANALYSE DES RESULTATS.	115
6.2.	UNE LOCOMOTION AVEC EVITEMENT D'OBSTACLES.	117
6.2.1.	ANALYSE DES RESULTATS.	118
7.	LA DISCUSSION DES RESULTATS.	119
7.1.	DISCUSSION DES RESULTATS DE LOCOMOTION SUR UN ESPACE PLAT.	119
7.2.	DISCUSSION DES RESULTATS DE LOCOMOTION AVEC EVITEMENT D'OBSTACLES.	121
7.3.	LA VARIATION D'ANGLES DE JOINTURES DU ROBOT.	121
8.	EVALUATION DU MODELE.	123
8.1.	COMPARAISON AVEC D'AUTRES MODELES.	124
9.	CONCLUSION.	126
	CONCLUSION GÉNÉRALE	116

PERSPECTIVES	118
BIBLIOGRAPHIE	131

Liste de figures

Figure 1: Les boucles de LANGTON : un automate cellulaire auto-réplicateur.	9
Figure 2 : Théorie synthétique de l'évolution.	21
Figure 3: Un chromosome selon Holland.....	24
Figure 4: Exemple de roulette pipée pour une population de 4 individus en fonction de Psel.	27
Figure 5: Croisement avec un point de coupe.	29
Figure 6: Croisement avec deux points de coupe.....	29
Figure 7: Croisement uniforme.	30
Figure 8: Mutation appliquée à un individu.	30
Figure 9 : Un chromosome fondé sur les S_expressions.	32
Figure 10: La mutation sur les arbres.....	33
Figure 11 : Le croisement génétique portant sur les arbres sous forme de S_Expressions.33	33
Figure 12: Schématisation d'un neurone biologique. Le neurone est une cellule qui possède des prolongements de 2 types : Un axone et des dendrites.....	36
Figure 13: Le neurone formel de McCulloch et Pitts est une abstraction mathématique du neurone biologique. Il décide de son éventuelle activation en appliquant une fonction de transition ϕ à la somme Σ des potentiels qu'il reçoit en entrée.	37
Figure 14: Le scalaire w_{ij} représente le poids associé à la connexion entre les neurones i et j	38
Figure 15: Le perceptron de Rosenblatt utilisé pour la reconnaissance des formes.....	39
Figure 16: Illustration du problème de la séparation linéaire de la fonction logique XOR.	40
Figure 17: Dans le perceptron multicouche, des couches intermédiaires sont insérées entre les couches d'entrée et de sortie.	41
Figure 18: Quelques images obtenues des L-Systèmes.....	42

Figure 19 : Système de classifieurs de type LCS.	45
Figure 20: Plante virtuelle. (a) [Jac, 1996]. (b) [Tou, 2003]. (c) [Bor, 2008].	50
Figure 21: (a) Exemples de graphes de génotype et des morphologies correspondantes de créatures de Karl Sims. (b) Le cycle des effets entre le cerveau, le corps de la créature et l'environnement.	53
Figure 22: Les créatures évoluées de Karl Sims peuvent sauter, nager, marcher, en compétition ou suivre un point lumineux.	54
Figure 23: Créature de Framessticks.	55
Figure 24: Les créatures de Nicolas Lassabe face à des situations complexes.	56
Figure 25 : Les créatures de Thomas Miconi inspiré de celles de Karl Sims	57
Figure 26: Les Créatures de Miconi qui mènent un combat afin de s'accaparer un cube..	57
Figure 27: Les créatures de Nicolas Chaumont qui se déplacent et qui catapultent un bloc.	58
Figure 28: (a) Morphologie de la créature comportant quatre pattes, huit actionneurs et un capteur de contact par patte. (b) Réseau de neurones reliant les capteurs aux actionneurs. (c) La progression de l'erreur à travers les générations. (d) En blanc, le chemin suivant quand les capteurs de phéromones sont captifs et en noir quand ils sont désactivés [Lip, 2005].	59
Figure 29: (a) : Le mouvement de saut de la lampe sachant qu'un programme de contrôleur est produit par se mouvement. (b) : La lampe peut apprendre à éviter des obstacles quand des contraintes additionnelles sont ajoutées au système.	60
Figure 30: (a) L'utilisation des algorithmes génétiques pour fournir la trajectoire d'un bras articulé qui sera défini par un ensemble de segments de longueur fixe ou variable et des différents angles formés entre ces segments. (b) L'interaction Temps-Réel.(c) Les bras articulés vont donc devoir coopérer afin d'emmener le palet de son point de départ à son objectif.	61
Figure 31: (1). (a) et (b) Cliff et Miller font co-évoluer deux créatures, la jaune doit apprendre à fuir, la rouge à rattraper la jaune, (c) Cédric Sanza. (2). (a) : Panzoli a fait évoluer un agent contrôlé par un réseau de neurones, qui doit être capable de suivre une lumière. (b) : Une autre créature de Panzoli, utilisant le même type de contrôleur mais étendu à des effecteurs différents. (c) Olivier Heguy.	62
Figure 32 : Les meilleurs résultats obtenus par les bipèdes de Krister Wolff sur un environnement 3D. [WBH+, 2006].	63

Figure 33: Les humanoïdes de Maurizio Conventi apprennent à marcher sur un espace plat au bout de 200 générations [CF, 2006].	63
Figure 34 : Les robots humanoïdes de Malachy Eaton [ED, 2006].	64
Figure 35: Travaux d'Yves Duthen et Arturo Chavoya [CD, 2007] l'utilisation des AG pour faire évoluer les réseaux de normalisation artificielle (ARNs) pour les modèles colorés désirés : (a) modèle carré de Trois-couleur. (b) modèle carré de Quatre-couleur. (c) un modèle du Drapeau français avec un poteau.	66
Figure 36: Les travaux d'Yves Duthen et Sylvain Cussat-Blanc [SLD, 2008] système de transfert de substrats artificiel. (1). La créature se développe pour créer la structure du système de transfert et commence le transfert. (2). La croissance de l'étoile de mer. (3). La croissance de la méduse.	66
Figure 37: (a) Schéma d'illustration d'un robot évolutif. (b) Des instances différentes de la génération entière. (c) Trois évolutions des robots, dans la simulation (d) les trois robots reproduits en réalité physique en utilisant un prototype rapide [LP, 2000].	68
Figure 38: (a) Schéma d'illustration de la séquence de construction. (b) Le robot résultant avec des joints actionnés. (c) Trois exemples de robots produits par l'évolution des l-systèmes. (d) Instanciation physique des exemples précédents. [HLP, 2003]	69
Figure 39: Changements morphologiques dans les robots de molécule dans la réalité [ML, 2004]	70
Figure 40: Robot de Victor Zykov 2.	71
Figure 41: Les types de joints [ODE, 2006]	80
Figure 42: L'architecture globale du système.	83
Figure 43: Le modèle de la créature humanoïde simulé par ODE.	86
Figure 44: Le réseau de neurones RNN.	89
Figure 45: La courbe de la fonction sigmoïde bipolaire.	90
Figure 46: L'ensemble de senseurs associés à la créature simulée.	93
Figure 47 : Le format du chromosome utilisé.	93
Figure 48: La boucle de l'algorithme évolutionnaire.	94
Figure 49: Boucle d'évaluation d'un individu de la population.	97
Figure 50: Evaluation d'un individu de la population sur le simulateur.	98
Figure 51: Un diagramme pour le processus de croisement en un seul point.	100
Figure 52: Les angles Alpha en rouge et Beta en bleu.	106
Figure 53 : L'architecture globale du système implémenté.	107

Figure 54: A gauche l'environnement simulé avec la créature, un obstacle (table) et un mur. A droite l'environnement simulé avec la créature seulement.	114
Figure 55: Les résultats moyens de locomotion sur un espace plat.	116
Figure 56 : Les meilleurs résultats de locomotion sur un espace plat.....	117
Figure 57: Des résultats d'évitement d'obstacles (tables).....	119
Figure 58: Graphe des valeurs des fitness par rapport au nombre de générations pour une locomotion sur un espace plat simple.....	120
Figure 59 : Les graphes de variation d'angles de jointures du robot sur 10 secondes de simulation.	123

Liste de Tables

Tableau 1: Probabilités de sélection dans le cas de la roulette pipée et de la sélection par rang.....	27
Tableau 2: Synthèse sur les travaux de créatures artificielles.....	73
Tableau 3: Les paramètres de simulation de la créature humanoïde.....	86
Tableau 4 : Les paramètres des stratégies d'évolution.....	95

Résumé

La simulation par ordinateur de la locomotion humaine est devenue une préoccupation de taille ces dernières années. Cette préoccupation est liée au désir d'employer les êtres humains en tant qu'acteurs synthétiques dans des environnements virtuels. Beaucoup de chercheurs ont proposé des mécanismes divers de contrôle du mouvement pour simuler la locomotion des créatures humanoïdes. Cependant, la plupart des systèmes d'animation basés sur ces mécanismes de contrôle sont plutôt adaptés à la reproduction de la marche humaine sur une terre plate, sans obstacles.

Le but principal de notre étude est de proposer une solution nouvelle au problème important de la marche dans divers environnements. Nous présentons, dans ce mémoire, une proposition d'un système d'animation capable de reproduire la marche humaine sur des terrains divers. Pour cela, nous avons adopté une approche de contrôle biologiquement inspirée et basée sur la théorie connexionniste (les réseaux de neurones artificiels, ANNs). Une autre approche suit le paradigme de l'évolution artificielle, i.e l'utilisation des algorithmes évolutionnaires, pour optimiser un contrôleur donné, qui est un réseau de neurones récurrents (RNNs).

La marche anthropomorphe abstraite pour une créature humanoïde simulée a été réalisée au moyen de l'évolution artificielle du modèle de réseaux de neurones récurrents (RNNs). L'approche a été étudiée pour des simulations de la dynamique d'une créature artificielle représentée par un corps tridimensionnel rigide possédant 15 degrés de liberté. Le modèle de contrôleur (RNN) subit des modifications structurelles au niveau de ses poids par le biais d'un algorithme évolutionnaire. Des modèles stables et normaux de démarche ont été obtenus, avec une vitesse maximum de marche d'environ 0,94 m/s.

Abstract

Computer animation of human locomotion has become popular in recent years because of the desire to use human beings as synthetic actors in three-dimensional simulation environments. Researchers have proposed various motion control mechanisms to simulate human-like figure locomotion. However, most of the animation systems based on these control mechanisms are only suitable for animating human walking on flat ground, without obstacles.

The main purpose of this study is to provide a new solution to the important problems of walking in various environments. In this dissertation, we present research into building an animation system which is capable to simulate human walking on varied terrain. For that we adopted an approach of order which is biologically inspired and is based on artificial neural networks (ANNs) theory. Another approach follows the paradigm of the artificial evolution, i.e the use of evolutionary algorithms (EAs), to optimize a given controller, who is a recurrent neural network (RNN).

Anthropomorphic walking for a simulated humanoid creature has been realized by means of artificial evolution of a recurrent neural network (RNN). The approach has been investigated through full rigid-body dynamics simulations in 3D of the creature with 15 degrees of freedom. The model of controller (RNN) undergoes structural modifications on the level of its weights by using an evolutionary algorithm. Stable, natural gait patterns were obtained, with a maximum walking speed of around 0,94 m/s.

ملخص

اكتسبت عملية التمثيل المعلوماتي لحركة الإنسان اهتماما متناميا خلال السنوات الأخيرة. ويرتبط هذا الاهتمام ارتباطا وثيقا بالاستعمال المتصاعد للأفلام المتحركة الاصطناعية مع التمثيل الاصطناعي للعناصر الفاعلة داخل البيئة الافتراضية الثلاثية الأبعاد.

و في هذا الإطار اهتم العديد من الباحثين، في هذا المجال بالتحديد، بهذه الإشكالية و ذلك باقتراحهم لآليات متعددة تسمح بمباشرة مراقبة و توجيه حركات التنقل المرتبطة بالعناصر الافتراضية الفاعلة داخل هذه البيئة الافتراضية. بيد أن الأغلبية من هذه الآليات المتخصصة في مراقبة و توجيه العناصر الافتراضية تنحصر في الفضاءات الخالية من العقبات.

الهدف الرئيسي من هذه الدراسة يتمثل في توفير حل للمشاكل المرتبطة بتنقل المخلوقات الافتراضية في المجالات المركبة و الغير العادية. و لتحقيق هذا الحل، قمنا بتبني طريقة تحكم ملهمة بيولوجيا قائمة على أساس الشبكات العصبية الاصطناعية. أما النهج الآخر المتبع فيتعلق بنموذج تطوري اصطناعي و ذلك باستخدام خوارزميات تطويرية(الخوارزميات الجينية) و المراقبة باستعمال الشبكات العصبية الاصطناعية المتكررة (RNN).

تمكنا من استحداث عملية المشي المجردة للمخلوقات الاصطناعية الشبيهة بالإنسان مستعملين في ذلك مبدأ التطور الاصطناعي لنموذج الشبكات العصبية الاصطناعية المتكررة. وقد اتبعنا لتحقيق ذلك منهجا مرتبطا باستغلال الديناميكية المتعلقة بالمخلوقات الاصطناعية الممثلة بجسم ثلاثي الأبعاد يمتلك 15 درجة حرية. أما بالنسبة لنموذج المراقبة و المتمثل في شبكة عصبية اصطناعية، فيخضع لتعديلات هيكلية على مستوى أقاله بواسطة الخوارزميات الجينية.

و تبعا لكل ما سلف، تمكنا من الحصول على نماذج طبيعية و مستقرة تسمح استحداث حركة المشي بسرعة تضاهي 0,94 م/ث.



Introduction générale

La vie artificielle est probablement un des plus vieux rêves de l'homme. De tout temps, en effet, l'homme n'a cessé de s'en inspirer pour créer, recréer ou reproduire les mécanismes de la nature. D'abord, il s'est servi de ce que la nature lui fournissait pour tenter ensuite de trouver le moyen de se débrouiller par lui-même pour obtenir ce qu'il désirait.

La vie est un des premiers concepts que l'homme ait tenté de maîtriser, l'agriculture et l'élevage sont en effet apparus très tôt. Mais c'est peut-être aussi la dernière chose qui lui résistait et qui ne cessera, probablement, à lui résister tellement les mécanismes qui la composent sont complexes et difficile à cerner.

Jusqu'à présent, tous les fruits de la technologie humaine n'étaient qu'une série d'outils qui l'aidaient à réaliser ce qu'il avait imaginé. La vie artificielle va lui fournir des outils qui l'aideront à imaginer ce qu'il pourrait réaliser.

En définitive, la question qu'on se pose est toujours la même : Est-ce que tout ça ne va pas se retourner contre nous ? Et finalement, à quoi ça sert ?

1. Définitions.

Avant de parler de vie artificielle, il faudrait définir le concept de la vie.

En fait, la vie peut être définie comme étant un Ensemble des phénomènes (nutrition, assimilation, croissance, reproduction,...) communs aux êtres organisés et qui constituent leur mode d'activité propre, de la naissance à la mort. (...) (LAROUSSE)

D'après cette définition, le fait d'être vivant implique :

- La capacité de *nutrition*, autrement dit la recherche de sources d'énergie.
- La capacité d'*assimilation*, c'est à dire la transformation de cette énergie en vue de reconstituer et de développer son corps matériel.

- La capacité de *croissance*, la faculté de se développer, de grandir.
- La capacité de *reproduction*, qui est très importante. En effet, sans reproduction, pas d'évolution possible. Et le fait de se reproduire induit également le fait de mourir pour une simple question d'espace vital.

Mais le qualificatif le plus important est certainement celui lié à la capacité d'organisation. Cela signifie qu'une entité vivante est un amas d'éléments liés et organisés selon des structures bien définies.

1.1. Intelligence artificielle.

L'intelligence artificielle est une science dont le but est de doter une machine d'une pensée similaire à la pensée humaine ou simplement d'en donner l'illusion.

1.2. Vie artificielle.

La vie artificielle est une tentative de reproduire le « vivant » dans une machine, dans des conditions initiales définies et à partir de mécanismes simples. Les principes de la vie artificielle sont :

La reproduction qui est la base de la vie. Quelque chose qui ne se reproduit pas n'est pas vivant.

Pour que la reproduction soit possible, il faut que le plan de l'organisme soit communiqué aux descendants. La seule manière connue est d'avoir un code génétique où sont stockées toutes les informations nécessaires à la survie de l'organisme. Les généticiens qualifient ce phénomène vital de transmission du patrimoine génétique de génération en génération.

L'autoréparation qui indique que si l'organisme possède son plan, il doit être capable de se reformer à partir d'une seule cellule. C'est un peu comme l'embryon humain qui n'est composé que d'une seule cellule qui se divise de plus en plus rapidement. Cependant, la différenciation cellulaire n'intervient pas, ou du moins pas encore, dans la vie artificielle. Toutes les cellules sont normalement identiques. Certaines formes plus complexes, incluant la différenciation cellulaire, peuvent aussi exister (présence d'un «centre de la pensée » ou de «bras » par exemple).

D'un autre côté, les mutations doivent être possibles et les nouvelles formes qui ne sont pas adaptées doivent disparaître, ce qui conduit au processus d'évolution et



d'amélioration de l'espèce. Les croisements, eux, ne peuvent avoir lieu qu'en cas de reproduction de type sexuée.

A partir du moment où un organisme se reproduit, et ce dans un espace limité, le milieu de la vie arrive assez rapidement à saturation et certains organismes doivent être éliminés. Dans la vie réelle, ce sont les vieux ou les malades. Par opposition, dans la vie artificielle, ce sont les individus les moins adaptés au milieu dans lequel ils se trouvent, par manque de place pour se développer ou par manque d'énergie. [Heu, 1994].

2. Où trouve-t-on la vie artificielle ?

On peut penser que les explications précédentes sont purement théoriques. La réponse est en fait négative et on peut considérer aujourd'hui que la vie artificielle est partout autour de nous.

Ce ne sont plus des hommes qui conçoivent les avions, qui les empêchent de s'écraser ou de trop tanguer lorsqu'ils passent dans des trous d'air, qui choisissent dans quel ordre les pièces seront assemblées dans les voitures. Ce sont des algorithmes génétiques qui s'en occupent à leur place.

Ce sont eux aussi qui font de la reconnaissance vocale, de la reconnaissance des visages et autres systèmes de sécurité.

Ce sont eux qui étudient le comportement des foules dans les immeubles et c'est grâce à eux que la circulation automobile est plus fluide, car il suffisait, par exemple, simplement, de modifier légèrement un virage sur telle bretelle de l'autoroute pour que les bouchons n'aient plus de raisons pour se former.

N'oublions pas les petits robots qui vont dans les endroits non fréquentables pour l'homme et qui doivent pouvoir se débrouiller seuls : les sondes envoyées pour explorer la planète Mars, qui ne peuvent pas être commandées en temps réel (plus d'une minute de décalage pour les ondes envoyées de la Terre vers Mars et autant pour le retour) et doivent donc être capables de réagir de manière autonome à une situation nouvelle. On pense aussi à utiliser ce type de robots pour exécuter certaines opérations dangereuses comme par exemple dans les zones contaminées des centrales de l'ex-république soviétique Tchernobyl.

On peut affirmer aujourd'hui, qu'on les retrouve partout et tendent à envahir davantage notre quotidien après avoir envahi le secteur de l'industrie de pointe.



3. Les applications.

Si la vie artificielle est apparue par la convergence de toutes les sciences, elle peut fournir un nombre illimité de solutions inédites et ce dans tous les domaines.

Les algorithmes génétiques pourraient par exemple servir à trouver de nouvelles molécules dans la recherche pharmaceutique grâce à leurs possibilités de mutations infinies.

La vie artificielle, peut également, servir d'outil d'optimisation puisque dans un environnement donné, la ou les meilleures solutions finissent toujours par émerger sans aucune intervention extérieure [Les, 1999].

4. Contribution.

Le domaine de la vie artificielle est en étroite relation avec les espaces virtuels peuplés de créatures artificielles pour lesquelles il est possible de générer les morphologies et les aptitudes en fonction de leurs objectifs. Grâce à des techniques combinant les algorithmes génétiques et les réseaux de neurones, des systèmes créés par Karl Sims [Sim, 1994a, Sim, 1994b], produisent par émergence des créatures capables de se déplacer (marcher, nager, ...) et d'agir en utilisant des comportements adaptés à leurs environnements.

Ces travaux constituent des axes de recherche récents, et notre travail s'inscrit dans cette optique. Notre objectif principal étant de générer automatiquement un système de contrôle d'une créature artificielle avec une forme spécifiée, un humanoïde en l'occurrence. Le contrôleur utilisé (réseau de neurone) va permettre à cet humanoïde de faire évoluer ses capacités à se déplacer dans l'environnement simulé (i.e un comportement de locomotion) pour converger vers une marche réaliste semblable à celle de l'être humain.

4.1. But de cette étude.

À la lumière de la montée subite récente de l'intérêt pour des applications articulées autour des environnements virtuels, beaucoup de recherches ont été consacrées à la résolution de problèmes de manœuvre d'humains virtuels dans des mondes tridimensionnels virtuels simulés, et particulièrement à la locomotion humaine. Cependant, la plupart des approches d'animation basées sur ces études peuvent reproduire uniquement la marche sur un support plat, sans obstacles. Cette défaillance, liée à la possibilité de



locomotion sur des environnements inégaux, réduit et limite inexorablement leur exploitation dans des environnements virtuels.

4.2. Le modèle.

La marche représente le moyen le plus commun de se déplacer et représente une des activités essentielles de notre vie quotidienne. La marche humaine peut être décrite comme la capacité de générer des mouvements sans heurt, fortement coordonnés, rythmiques grâce auxquels le corps peut se déplacer pas à pas dans une direction spécifiée. Elle exige la participation simultanée de tous les joints inférieurs des membres dans un modèle complexe du mouvement.

Pour répondre à cette préoccupation, nous avons proposé un modèle tirant profit des techniques bio-inspirées dans le but de générer des mouvements de locomotion ou d'évitement d'obstacles réalisés par une créature humanoïde capable d'évoluer dans un environnement dans lequel la physique est simulée. Les mouvements de cet humanoïde sont contrôlés à un niveau microscopique par une technique connexionniste qui est le réseau de neurones, l'aspect évolution est assuré par une des techniques évolutionnaires.



Plan du mémoire

Après une brève introduction au domaine de la vie artificielle que nous décrivons dans la première partie, nous présentons les termes les plus répondus dans notre domaine d'étude, tel que la vie, l'évolution et la sélection naturelle. Nous effectuons ensuite un tour d'horizon sur les principales techniques évolutionnaires (telles que les algorithmes génétiques) et les techniques d'apprentissage telles que les réseaux de neurones, les systèmes de classifieurs et les générateurs de contrôle de patterns (CPGs) ainsi que les générateurs de morphologies tels que les L-systèmes.

La section suivante présente l'état de l'art sur la génération de créatures artificielles par des méthodes évolutionnistes. La synthèse effectuée sur ses travaux, nous permettra d'établir un modèle de créatures artificielles ainsi qu'un environnement permettant leur évaluation.

La seconde partie traite notre contribution à la simulation évolutionniste de créatures artificielles autonomes. Nous présentons un modèle de contrôleur basé sur les réseaux de neurones de type récurrents (RNNs). Les créatures artificielles contrôlées ont plusieurs comportements à optimiser (par les stratégies d'évolution) tel que la locomotion ou l'évitement d'obstacles.

Enfin, le dernier chapitre permet d'évaluer notre modèle et de présenter les premiers résultats. Nous montrons dans un premier temps les capacités d'évolution de notre système de contrôle afin qu'il émerge vers un comportement de locomotion. Dans un second temps, nous testons l'adaptation de notre contrôleur pour un comportement d'évitement d'obstacles.

Nous concluons en citant les perspectives sur l'ensemble des directions qui nous semblent intéressantes à investir pour obtenir des créatures artificielles certes de plus en plus complexes mais possédant plus de capacités de réalisation de tâches par évolution.

Chapitre 1 : Vie artificielle

Vie Artificielle

Dans ce chapitre, nous introduisons le domaine de la vie artificielle en donnant des définitions sur les termes les plus utilisés en vie artificielle tels que la vie, l'évolution et la sélection naturelle en présentant également une brève description sur les techniques des systèmes évolutionnistes en détaillant celles que nous allons utiliser dans notre système, les stratégies d'évolution et les réseaux de neurones, notamment.

1. Les Origines de la vie artificielle.

1.1. Histoire de la vie artificielle.

Depuis une quinzaine d'années, des chercheurs d'horizons divers contribuent à ce nouveau domaine qu'ils ont baptisé Vie artificielle (*Artificial Life*). Plus qu'une discipline scientifique au sens strict, la vie artificielle est un regroupement de travaux hétéroclites, ayant pour point commun de s'inspirer directement et explicitement des caractéristiques du vivant. La révolution scientifique du XXe siècle a bousculé les interprétations de certains chercheurs sur la vie. L'idée a progressivement émergé que la vie, considérée comme un mécanisme, comme un processus, pouvait être à la portée de la connaissance humaine. En 1987 s'est tenue à Santa Fe au Nouveau Mexique la première conférence internationale sur la vie artificielle. Son jeune organisateur, Christopher Gale LANGTON, déclarait alors en prélude :

« 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 [Lan, 1989]. »

L'objectif de LANGTON était davantage plus ambitieux:

« Le but ultime de la vie artificielle serait de créer la « vie » dans un autre substrat, idéalement un substrat virtuel où l'essence de la vie aurait été abstraite des détails de sa mise en œuvre dans quelque substrat que ce soit. Nous aimerions construire des modèles qui sont si semblables au vivant qu'ils cesseraient d'être des simulations de la vie pour en devenir des exemples [Lan, 1986].»

Cette citation illustre ce que l'on qualifie maintenant *d'interprétation forte de la vie artificielle*. Comment LANGTON a-t-il pu conclure de la faisabilité d'un tel programme? Ceci apparaît d'autant plus difficile qu'à ce jour, nous ne disposons toujours pas d'une définition incontestée de la vie. C'est donc d'abord un problème philosophique qu'il posait.

Pour LANGTON, la vie est une affaire d'organisation et de structure, non de substance. Dans ce contexte, la possibilité théorique de réaliser de nouvelles instances du vivant existerait. Son raisonnement trouve ses racines dans les travaux des grands logiciens de la première moitié du XXe siècle avec en particulier Alonzo CHURCH et Alan TURING.

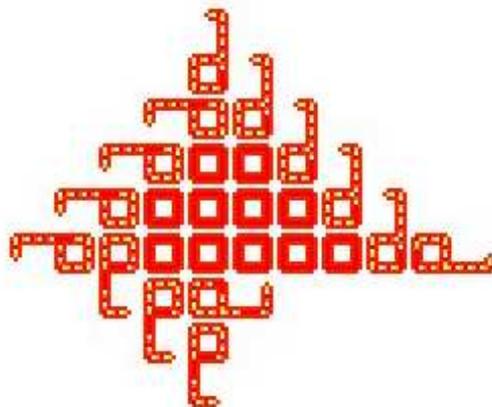


Figure 1: Les boucles de LANGTON : un automate cellulaire auto-réplicateur.

En 1936, Alan TURING a proposé le concept de *Machine Universelle*. Il montrait alors comment, un système logique minimum permettait de déterminer toute fonction calculable. Ce travail, qui est à l'origine de l'ordinateur (l'ordinateur n'étant rien d'autre qu'une machine universelle de Turing finie), a donné lieu à la *thèse de Church/Turing*.



Dans sa version de base, elle énonce que toute fonction calculable l'est par une machine de TURING. Cette thèse a été élargie dans ce que l'on qualifie de «version physique» selon laquelle une machine de TURING peut réaliser toute opération de traitement de l'information réalisable par un système physique. L'interprétation faible de cette thèse suppose ainsi que tout système, pouvant être décrit comme un automate, peut être simulé par une machine de TURING. Selon l'interprétation forte, une machine de TURING peut reproduire le comportement de tout système physique. D HOFSTADTER a parlé de « version microscopique » :

« Le comportement des éléments constitutifs d'un être vivant peut être simulé par un ordinateur. Autrement dit, une description suffisamment élaborée de l'état interne et de l'environnement local de n'importe quel élément (on suppose généralement qu'il s'agit d'une cellule) permet de faire calculer son comportement de façon arbitrairement précise par [...] une fonction récursive générale [Hof, 1999]. »

Les ordinateurs, c'est-à-dire des machines universelles de TURING finies, sont capables de traiter tout algorithme fini. Si on leur fournit la description logique d'une machine, c'est-à-dire les déterminants de son organisation, ils seraient capables de la « réaliser ». On arrive ainsi à l'idée selon laquelle « les ordinateurs sont des instances de processus biologiques [Sob, 1996] », cette thèse relevant de l'option philosophique. Pour être admise, elle suppose que l'on accepte que la réalité d'un système relève strictement de son organisation et qu'elle est donc indépendante de son support physique, c'est ce qu'on appelle le *fonctionnalisme*. Trouvant son origine dans l'intelligence artificielle, il suppose ainsi qu'il serait possible de reconstruire la logique d'un cerveau humain en dehors du substrat neuronal.

La vie est affaire d'organisation et non de substance ; tout fait organisationnel physique ou non est réalisable par une machine de TURING. Ces deux assertions, profondément discutables l'une comme l'autre, sont au cœur de l'interprétation forte de la vie artificielle.

C'est sur ces bases que certains affirment avoir créé de nouvelles formes de vie. Le plus célèbre d'entre eux, *Thomas RAY*, a même écrit un article intitulé « Comment j'ai créé la vie dans un univers virtuel [Ray, 1993] ». On pourrait penser qu'il s'agisse là de l'une de ces envolées lyriques que l'on rencontre parfois chez les théoriciens de l'informatique, mais RAY est d'abord un biologiste. Spécialiste de la forêt tropicale, il a consacré toute la première partie de sa carrière à la biologie, ce n'est que plus tard qu'il



s'est intéressé à l'informatique. A ce niveau, une question mérite d'être posée : Comment un biologiste a-t-il pu aboutir à une telle conclusion ?

Des outils inspirés du vivant.

De par leurs implications philosophiques, les tentatives de recréation de la vie illustrent le côté le plus fascinant de la vie artificielle. Dans la pratique, ces travaux sont pourtant loin d'être les plus importants, ou les plus représentatifs de la « discipline ». La vie artificielle ne se limite pas aux seules tentatives de construction de nouvelles instances de la vie ou à la biologie théorique. Elle s'inspire davantage des propriétés du vivant pour proposer des constructions originales, tant algorithmiques que physiques, dotées de capacités étonnantes et aptes à résoudre des problèmes difficiles vis-à-vis desquels les approches plus traditionnelles rencontrent de difficultés réelles. C'est le vaste domaine des *artefacts biomimétiques* (algorithmiques et robotiques), dont voici quelques exemples [Ren, 2003].

- Les algorithmes évolutionnaires ;
- Les L-systèmes ;
- Les essaims de particules ;
- L'optimisation par colonie de fourmis ;
- Les machines auto-répliquantes et autoréparables ;
- Les robots.

Les deux premiers exemples seront décrits en détails à la fin de ce chapitre.

1.2. Qu'est ce que la vie ?

Dans le langage courant, l'attribut artificiel peut avoir deux significations. La première évoque le fait que l'objet en question est une imitation d'un mécanisme ou d'un processus qui est difficile ou impossible à recopier ("fleur artificielle"). Le deuxième sens suggère que la synthèse de l'objet est faite par l'homme, alors que l'objet lui-même est aussi naturel que le phénomène original ("lumière artificielle"). Ces deux utilisations possibles correspondent à deux points de vue philosophiques incompatibles quant à la question de la vie artificielle. Selon l'interprétation faible, les travaux en vie artificielle sont une manière d'étudier la biologie en créant des modèles de processus variés et associés aux organismes vivants dans l'ordinateur ou sur d'autres supports artificiels. Cependant, ces modèles ne sont pas vraiment vivants. L'interprétation forte énonce qu'en instanciant des processus



typiques du vivant dans un médium artificiel, le système en question mérite d'être appelé vivant tout comme les organismes naturels.

En d'autres termes, les défenseurs du premier point de vue voient les modèles informatiques comme des simulations de la vie, alors que le deuxième groupe leur accorde le potentiel d'être des réalisations de la vie [Pat, 1988].

Ce différend est étroitement lié à la façon de considérer la question suivante : est-ce que la vie dépend du support physique dans lequel elle est implémentée, ou bien est-ce qu'elle est essentiellement un processus indépendant de la matière qui peut se dérouler dans tout médium qui possède la structure pour le supporter?

Ainsi, le débat entre les deux points de vue ne pourra pas être résolu sans une définition nette et précise de la notion de la vie. On reconnaît la vie quand on la voit, dit-on quelquefois. Par contre, à ce jour la science ne dispose toujours pas d'une définition incontestable. Certaines sont essentiellement empiriques et reposent sur l'observation de la nature. D'autres sont conceptuelles et cherchent à fournir une base théorique [Ren, 2002].

1.3. Définition de la vie artificielle.

Il est possible de trouver deux principales définitions de la vie artificielle. Selon Langton [Lan, 98], *"la vie artificielle a été définie comme étant l'étude de systèmes construits par l'homme qui présentent des comportements caractéristiques des systèmes vivants naturels."*

J. Doyne Farmer et Alleta d'A. Belin [FB, 1989] proposent une autre définition sous forme de propriétés caractérisant un système de vie artificielle dont voici la liste :

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. A l'opposé de la vie, un système de vie artificielle peut être reparti sur plusieurs endroits : par exemple, un robot et un ordinateur peuvent effectuer les calculs via une communication sans fil. Même à l'intérieur d'un ordinateur, rien ne garantit que les octets de ce système soient tous regroupés.

Si bien que les définitions précédentes peuvent paraître complètes selon les caractéristiques du vivant que l'on prend en compte. Pour éviter une définition de la vie artificielle sous forme de propriétés, il est possible de décrire la vie comme étant un processus qui s'auto entretient. La définition suivante a été donnée par Humberto Maturana et Francisco Varela [MV, 1980]:

"Il s'agit de concevoir un être vivant comme une organisation auto-poïétique. Cela caractérise le fait qu'un être vivant est un réseau de transformations dynamiques, fabricant ses propres composants (métabolisme) et qui construit une barrière topologique (membrane) qui, à son tour, est la condition nécessaire du fonctionnement en tant qu'unité du réseau de transformations qui l'a engendrée. Ce qui signifie que les êtres vivants sont continuellement entraînés de s'autoproduire."

Un nombre d'auteurs ont mis en avant la dynamique évolutionnaire des organismes vivants. Ainsi, le célèbre biologiste Ernst Mayr proposa de considérer une population d'entités comme vivante si "elle a la propriété de multiplication, d'hérédité et de variation" [May, 1975]. Dans le domaine de la vie artificielle, la définition évolutionnaire la plus influente est sans doute celle du philosophe Mark Bedau [Bed, 1996]. Pour lui, un système devrait être considéré comme vivant s'il est capable d'évolution ouverte, c'est-à-dire s'il exhibe un potentiel illimité de trouver de nouvelles solutions imprévisibles aux changements de l'environnement (*adaptation*).

Vu que c'est l'espèce et non l'individu qui évolue, c'est donc l'espèce qui est vivante au sens premier, l'individu ne l'est que par son appartenance à l'espèce.

Il est généralement reconnu en biologie que la vie sur Terre a émergé via une lignée de molécules répliquatives primitives puis de proto-organismes dont l'existence brouille la frontière entre le vivant et l'inerte. Ainsi, cette définition est séduisante par le fait qu'elle accorde, contrairement aux deux approches précédentes, l'existence de différents degrés de la vie en fonction du degré d'adaptabilité du système. Cependant, il est difficile d'accepter que la vie soit une propriété au niveau des populations [Bor, 2008].



2. Evolution.

2.1. Apparition de l'évolution artificielle.

L'évolution est un terme qui appartient au domaine de la biologie. Depuis le début de la vie, qui commençait comme une réaction chimique particulière entre les composants organiques qui ont causé la naissance d'une diversité gigantesque d'entités, l'exposition des caractéristiques et des comportements complexes a déconcentré les spécialistes humains.

Le pouvoir de l'évolution est plus visible dans des structures améliorées et appliquées par des organismes vivants vers les deux niveaux microscopique et macroscopique.

L'évolution artificielle est une tentative d'utiliser ce pouvoir pour la production des entités artificielles. L'idée d'utiliser les ordinateurs pour l'étude des concepts liés à la vie et à l'évolution est presque plus vieille que les ordinateurs eux mêmes. John Von Neumann a montré comment le calcul d'automates universels de Turing peut être appliqué à la création d'entités autoreproductibles. Sa considération sur le processus évolutionnaire l'a conduit à formaliser les mécanismes d'hérédité et d'autoreproduction et à examiner la question de la complexité de l'évolution [McM, 2000]. Peu de temps après, à priori dans un effort non relié, les ingénieurs ont utilisé les techniques évolutionnaires pour accomplir les tâches pratiques d'optimisation et de conception. Ces deux approches de l'évolution artificielle (théorique et conceptuelle d'un côté, et pratique, orientée machine de l'autre) ont évolué étroitement sur des lignées parallèles de la recherche en tant que vie artificielle et qu'optimisation évolutionniste.

2.2. Ecart entre l'évolution et l'évolution artificielle.

Malheureusement, il semble qu'il existe un manque de connexion entre les études de l'évolution dans la nature et dans les ordinateurs, ce qui peut être la cause du manque d'étude de fondements théoriques de l'évolution artificielle. Un résultat notable de ce manque de connexion est un désaccord entre ce qui est prévu à partir de l'évolution, et ce qui est exactement obtenu grâce à elle. Cela dénote l'impression, selon laquelle les systèmes d'évolution artificielle et de vie artificielle doivent souvent compter sur une



supposition implicite, des règles ADHOC et des mécanismes qui ont été soulignés par Taylor [Tay, 1999]:

« Le résultat de [notre] travail, et la considération de la littérature existante dans les systèmes évolutionnistes artificiels, amène à la conclusion que les modèles de la VA (...) sont absents pour un nombre de raisons théorique et méthodologique. C'est souligner que les considérations théoriques explicites doivent guider le concepteur de tels modèles, s'ils ont à avoir une valeur scientifique ».

Et Further (Citing Patee [Pat, 1989]):

« Je suis certainement pas la première personne qui a fait une critique sur les modèles de la vie artificielle sur ses raisons [...] Par exemple, Howard Pattee prévient que «les simulations qui sont dépendantes de paramètres ADHOC et de règles de but-spécial et de contraintes pour leurs imitation ne peuvent pas être utilisées pour supporter des théories de la vie» [Pat, 1989] [...]: La sensation d'adéquation de ce genre de système est une conséquence directe de ce manque d'étude de fondements théoriques. L'espace des paramètres incontrôlables de plusieurs d'entre eux peut aussi être attribué à ce manque de directions. Comme résultats de cette faiblesse, même si les comportements intéressants sont observés dans ces systèmes, nous sommes dans l'incapacité d'expliquer pourquoi. »

2.3. Qu'est ce que l'évolution?

L'évolution n'est pas compliquée ni difficile à comprendre, au moins dans ses principes de bases. Ses principes peuvent être résumés en un couple de phrases directes. En fait, l'évolution est relativement simple et nécessite peu d'explications particulières. Ceci, en effet, est peut être la difficulté la plus importante quand on raisonne sur le concept d'évolution. Bien que les principes de bases soient simples, ils sont aussi délicats, et très générateurs d'erreurs.

Afin d'asseoir notre recherche, nous allons chercher une réponse satisfaisante à la question: "qu'est ce que l'évolution?"

2.3.1. L'évolution est un changement héréditaire.

L'évolution dénote n'importe quel changement dans les fréquences des caractéristiques héréditaires d'une population d'entités pouvant se reproduire, en donnant de nouvelles générations. En pratique, on peut la regarder comme une stéréographie pour la «descente



avec modification», le propre terme de Darwin¹. Ce changement peut consister dans la perte de caractéristiques existantes, la création d'une nouvelle caractéristique, ou l'alternance fréquente des caractéristiques existantes.

Il est à noter que, le terme « évolution » par lui-même ne porte aucune implication d'adaptation, d'amélioration ou de fitness. L'évolution dans son sens général est un simple changement héréditaire. La nature et la direction de ce changement dépendent des forces qui la conduisent; ces forces peuvent être adaptatives (ex: la sélection naturelle), mais aussi sans fondement (mouvements génétiques aléatoires), ou même potentiellement nocives (ex: l'endogamie à travers une perte d'une diversité causé par les mouvements génétiques). Il est important de distinguer entre l'évolution d'une part, et les mécanismes qui causent ou qui conduisent cette évolution, d'autre part.

Dans un contexte biologique, l'évolution est essentiellement la négation de la stabilité des espèces: les espèces changent (se transforment) au cours du temps, plutôt que de rester fixes pendant toute leur vie, comme retenu communément par Lamarck et Darwin.

2.3.2. Les conditions de l'évolution.

En général, l'évolution présente trois conditions à vérifier, nommées variation, multiplication et hérédité². Pour rendre ces termes plus précis, on peut les considérer dans un ordre différent. Considérons d'abord la multiplication.

La multiplication est une explication de soi-même dans le contexte de la biologie, mais elle peut être utile pour générer le concept: ça signifie essentiellement que de nouveaux individus sont constamment créés, et que ces individus peuvent d'une manière ou d'une autre être associés avec certains individus déjà existants. Les nouveaux individus sont appelés «fils» ou «descendants», les individus existants avec ceux qui leur sont associés sont appelés «parents». La multiplication, ou la reproduction, est simplement la création de nouveaux individus, associés à un ensemble de parents existants. Les parents de certains individus, leurs propres parents, les parents de leurs parents, et ainsi de suite, sont les ancêtres de ces individus. Tous les individus qui ont certains individus comme l'un de leurs ancêtres sont les descendants de ces individus.

¹ Il peut être intéressant de préciser que le mot « évolution » n'apparaît pas n'importe où dans les premières éditions de l'origine des espèces.

² Ces trois conditions ont une longue histoire: Wallace [Wal, 1875] les mentionne déjà sous cette forme.



La variation implique que les descendants ne seront pas exactement porteurs des caractéristiques de leurs parents, ils seront de temps en temps différents de leurs parents, en exhibant certaines caractéristiques que leurs parents et frères ne possèdent pas.

L'Hérédité est le facteur crucial qui sépare l'évolution de la seule variation, la méthode doit permettre certaines de ces variations qui doivent ensuite être transmises vers les descendants, qui exhiberont aussi ces spécificités et différentes caractéristiques. Prendre simultanément ces trois conditions permet aux caractéristiques héréditaires de la population d'être modifiées au cours du temps. De plus, cela permet à l'évolution d'avoir lieu.

Dans ce qui suit, nous ne différencions pas toutes les voies possibles d'hérédité: génétique, épi-génétique (relative à l'épigenèse) culturel, etc. En pratique, l'évolution biologique est liée seulement aux caractéristiques biologiques, transmises à travers la reproduction biologique. Un autre domaine, tel que l'évolution culturelle, peut avoir des fonctionnements différents.

Darwin n'a pas inventé le concept d'évolution (contrairement à celui de stabilité des espèces), bien que son travail fût, de manière expérimentale, dans la fabrication du point de vue généralement acceptable plutôt que les marges de la théorie qui ont été faites au préalable. La contribution fondamentale de Darwin fut d'avoir identifié un mécanisme réalisable pour l'évolution, cela peut expliquer la quantité prodigieuse d'une complexité adaptative observable dans la nature sans invoquer des forces surnaturelle. Ce mécanisme, est bien entendu, la sélection naturelle.

2.4. La sélection naturelle.

2.4.1. Qu'est ce que la sélection naturelle ?

L'évolution est un changement dans les caractéristiques héréditaires d'une population. Lors de la reproduction d'individus, de nouvelles caractéristiques apparaissent, quelques caractéristiques existantes sont perdues et d'autres sont plus ou moins généralisées dans toute la population. Ces changements dans la fréquence peuvent être causés par divers facteurs, incluant des fluctuations aléatoires. La sélection naturelle est une des forces qui influencent la direction de ce changement.

Le mécanisme général de la sélection naturelle s'explique facilement: lors des interactions avec l'environnement, certains caractères phénotypiques héréditaires ont un



impact sur le succès de la reproduction (la «fitness») de leur porteur, ou des fils de leur porteur, ou pour d'autres entités qui possèdent aussi ce caractère. Mais comme ces caractères sont hérités, ils sont eux-mêmes dupliqués à travers la reproduction. Comme résultat, les caractères phénotypiques hérités qui, à travers leurs interactions avec l'environnement et le facteur de succès de la reproduction directe ou indirecte de leurs porteurs, deviendront mécaniquement de plus en plus communs dans la population. Inversement, les caractères phénotypiques hérités qui empêchent le succès de la reproduction des entités qui les portent voient leurs fréquences diminuer. A la fin, de tels caractères peuvent devenir fixés (c-à-d: être présent dans chaque entité dans la population), ou au contraire disparaître entièrement de la population.

De plus, la sélection naturelle provient de la variance dans la fitness des entités, qui est causée par l'interaction entre les caractères phénotypiques hérités et l'environnement. Elle peut donc être conçue comme une variance dans la fitness causée par les caractéristiques héréditaires. Alternativement, le terme «sélection naturelle» peut aussi se référer à un ensemble de forces dans l'environnement qui effectue la réplication des entités en agissant sur leurs caractères phénotypiques hérités. Ces deux points de vue désignent deux perspectives sur le même processus.

Ainsi, aux trois exigences préalables de l'évolution, la sélection naturelle ajoute des exigences supplémentaires: la variance héritée du succès de la reproduction, liée à l'interaction entre l'environnement et le caractère phénotypique hérité. Si ces quatre exigences sont vérifiées, on peut alors parler de la sélection naturelle issue de l'évolution darwinienne.

3. Systèmes évolutionnistes.

3.1. Histoire des algorithmes évolutionnistes de Darwin à Holland.

La découverte des principes de bases de l'évolution des espèces et de ses mécanismes a été un sujet polémique entre scientifiques entre le début du XIX^{ème} et le milieu du XX^{ème} siècles. Jusqu'à la fin du XIX^{ème} siècle, la grande majorité des naturalistes admettaient que les espèces n'évoluent pas.



Pourtant, dès 1809, Jean-Baptiste Lamarck, naturaliste français et fondateur du transformisme, émit l'hypothèse d'une évolution des organes d'un animal en fonction de ses besoins. L'adaptation des espèces se faisant selon l'influence du milieu de vie [LAM, 1809].

En 1859, Charles Darwin, naturaliste britannique, publia son ouvrage « l'origine des espèces par voie de sélection naturelle » [Dar, 1859]. Dans son livre, il tenta de défendre le transformisme mais sans reprendre les hypothèses de Lamarck.

Il émit l'idée selon laquelle, dans chaque espèce, la nature sélectionne les meilleurs individus. Chaque individu donnant naissance à beaucoup plus d'individus qu'il n'en peut survivre, cela entraîne la destruction d'un grand nombre d'entre eux et la conservation des plus aptes. C'est ce qu'il appela la sélection naturelle. Par le jeu de la sélection, les caractères avantageux se propagent peu à peu dans la population, et au bout d'un certain nombre de générations, l'espèce se trouve modifiée et mieux adaptée aux conditions d'existence. Ceci a permis d'arriver à la constatation selon laquelle les espèces se transforment.

Cette théorie a été appelée le *Darwinisme* et le courant reprenant ses concepts *l'évolutionnisme*.

En 1901, De Vries, Botaniste néerlandais, exposa sa théorie du *mutationnisme* selon laquelle les variations responsables de l'évolution ne se faisaient pas dans le temps mais de façon soudaine et se reproduisaient dans l'œuf. Il appela cela la mutation.

Aujourd'hui, on admet qu'ils avaient tous les trois en partie raison. Les bases de l'évolution étaient posées:

- L'adaptation des espèces sous la pression de leur environnement : Les individus mal adaptés disparaissent.
- La sélection des meilleurs individus : La reproduction est un combat entre individus.
- Le croisement pour assurer la pérennité des espèces.
- La mutation qui peut créer des variations génétiques néfastes ou bénéfiques pour l'individu.

D'autres chercheurs se sont intéressés aux principes biologiques des mécanismes de l'évolution.

Dès 1948, les scientifiques avaient remarqué qu'une substance colorait particulièrement bien le noyau de la cellule et faisait apparaître des filaments.



Dès 1866, Gregor Johann Mendel, botaniste autrichien, fut également le fondateur de la génétique moderne. Il démontra l'existence de facteurs héréditaires par des expériences sur des petits pois.

En 1902, Walter Sutton proposa le chromosome comme site des caractères transmissibles, constituant le lieu physique de chaque caractéristique d'un être vivant.

En 1911 Wilhelm Ludvig Johannsen nomma gène l'unité élémentaire de support de l'hérédité.

En 1962, James Dewey Watson, biologiste américain, Francis Harry Compton Crick et Maurice Hugh Frederick Wilkins, biologistes britanniques, obtinrent le prix Nobel pour leur découverte de la structure d'ADN. [Ber, 2001].

Ces découvertes vont être reprises et utilisées dans un autre domaine scientifique: l'informatique.

Dès 1960, de nombreux chercheurs essayèrent d'appliquer les connaissances sur l'évolution naturelle à des problèmes informatiques. Nils Aall Baricelli et A. S. Fraser tentèrent de simuler l'évolution naturelle par des programmes informatiques [Bar, 1957, Bar, 1962, Fra, 1962]. Mais ces tentatives furent infructueuses, probablement à cause de l'utilisation exclusive de la mutation.

En 1975, Johann Holland proposa une première solution [Hol, 1975] qu'il a appelée algorithme génétique.

Tous ces chercheurs, qui ont exploré les processus d'adaptation des systèmes naturelle, ont permis la découverte de cette nouvelle technique d'exploration. Les mécanismes de base de ses algorithmes sont très simples mais les raisons, pour lesquelles cette technique fonctionne si bien, sont plus complexes et subtiles. Les fondements mathématiques des algorithmes génétiques ont été exposés par Goldbreg [Gol, 1989].

La simplicité de mise en œuvre, l'efficacité et la robustesse sont les caractéristiques les plus attrayantes de l'approche proposée par Holland. La robustesse de cette technique est due à une sélection intelligente des individus les plus proches de la solution du problème [Ber, 2001].



3.2. Principe des algorithmes évolutionnistes

En publiant l'origine des espèces [Dar, 1859], Darwin exposa pour la première fois sa théorie de l'évolution, désormais communément admise. Celle-ci repose sur les observations suivantes :

- Il existe au sein de chaque espèce de nombreuses variations, chaque individu étant différent ;
- Les ressources naturelles étant finies, il naît rapidement plus d'être vivants que la nature ne peut en nourrir; il en résulte une lutte pour l'existence entre les organismes ;
- Les individus survivants possèdent des caractéristiques qui les rendent plus aptes à survivre : Darwin baptisa ce concept *sélection naturelle* ;
- Les organismes survivants transmettent leurs avantages à leur descendance. L'accumulation au cours des générations des petites différences entre chaque branche généalogique crée de nouvelles espèces de plus en plus *aptés* à survivre.

Ces quatre principes de la théorie darwinienne ont été progressivement étendus pour incorporer les découvertes des lois de l'hérédité [Cas, 1903]) puis de la structure de l'ADN [CW, 1953] et, plus généralement, de l'ensemble de la génétique moderne (figure 2).

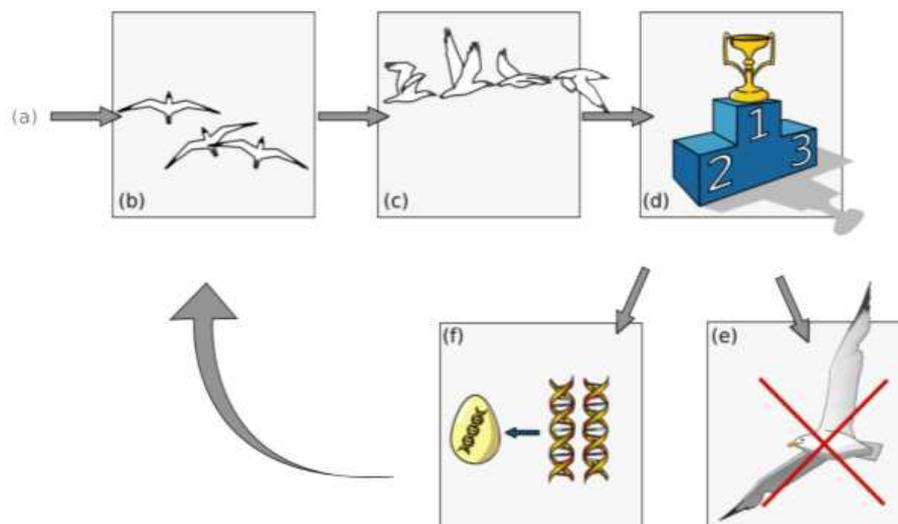


Figure 2 : Théorie synthétique de l'évolution.

Cette théorie synthétique de l'évolution a été baptisée *néo-darwinisme* et correspond aux bases de l'approche en vigueur dans la communauté scientifique. Dans le langage génétique moderne, les molécules d'ADN, constituées d'une suite de nucléotides,



constituent le support du génotype – l'ensemble des gènes – dont l'expression est dénommée le phénotype. Bien qu'il n'existe pas de définition précise de la notion de gène qui soit universellement reconnue, on peut considérer qu'un gène est une séquence de nucléotides codant pour un caractère particulier, le plus souvent une protéine.

Simuler l'évolution sur un ordinateur afin de créer des machines plus « intelligentes » anima Turing dès 1948. Mais il faut attendre le milieu des années 1960 pour que se développent les premiers algorithmes appliquant les concepts du néo-darwinisme à l'informatique. Ces recherches ont mené à quatre grandes familles d'algorithmes évolutionnistes [Gol, 1989; Bae, 1997]: les algorithmes génétiques [Hol, 1975] (qui seront décrits en détails sans la prochaine section), les stratégies d'évolution [Sch, 1993], la programmation génétique [Koz, 1992] et la programmation évolutionniste [WFO, 1966]. Toutes ces techniques reposent sur les mêmes principes de base (figure 2).

Tout d'abord, elles travaillent sur une population, initialement aléatoire, d'individus représentant chacun une solution au problème traité. Par la suite, chaque solution est représentée par un génotype et s'exprime sous la forme d'un phénotype.

Enfin, il est nécessaire de mettre au point une fonction d'adaptation, *la fitness*, qui permet de distinguer les phénotypes les plus performants. Ces derniers ont une probabilité plus importante de léguer à leur descendance leur génotype. Les règles régissant cette transmission de gènes sont décrites sous la forme d'opérateurs génétiques. Trois opérateurs sont requis :

- l'opérateur des élections qui décrit la manière dont les candidats sont choisis pour la reproduction;
- l'opérateur de croisement (*cross-over*) ou de recombinaison qui correspond à la méthode utilisée pour mélanger les génotypes des parents;
- l'opérateur de mutation qui décrit la fréquence et la nature des changements affectant le génotype lors de la transmission des gènes.

Il existe une multitude de versions de chaque opérateur, plus ou moins spécialisées pour le problème traité. (Des détails pourront être trouvés dans [Deb, 2001] ou [Gol, 1989]) (Voir également la prochaine section) :

- La roulette biaisée, qui permet de sélectionner les individus proportionnellement à leur fitness;



- la roulette biaisée sur le rang, sélectionnant les individus en fonction de leur classement, éliminant ainsi les effets dus à des différences de fitness trop grandes dans la roulette biaisée classique;
- le tournoi, dans lequel on tire aléatoirement n individus en choisissant celui qui possède la meilleure fitness.

Les autres opérateurs, croisement et mutation, dépendent du codage utilisé. Dans le cas des algorithmes génétiques, les solutions potentielles sont représentées par des chaînes binaires (par exemple 0001). L'opérateur principal est alors le croisement auquel s'ajoute la mutation pour accroître la diversité la population. Dans le cadre de la programmation génétique, une solution est encodée sous la forme d'un programme. Le programme est exécuté afin de permettre le calcul de la fitness. Le génotype est vu comme un arbre dans lequel chaque nœud représente une sous-fonction, une structure de contrôle, un opérateur, ou un terminal. L'opérateur principal est alors là aussi le croisement, implémenté sous la forme d'échanges de sous-arbres. À l'inverse, la programmation évolutionniste et les stratégies d'évolution utilisent essentiellement l'opérateur de mutation. Le premier cas correspond à l'évolution de programmes, le second à celle d'une liste de paramètres. Actuellement, la distinction entre ces quatre approches est de plus en plus floue, le génotype étant souvent constitué d'un mélange de structures complexes (arbres, graphes, listes de paramètres, ...). La différence entre ces quatre approches est donc essentiellement d'ordre historique et sont maintenant considérés comme des instances particulières des *algorithmes évolutionnistes* [DJo, 2002]. Du point de vue de la théorie de l'optimisation, ces algorithmes constituent des méta-heuristiques d'optimisation globale, au même titre que le recuit simulé [KGV, 1983] et la recherche Tabou [GLT, 1997]. Comme les autres méta-heuristiques, leur principal intérêt repose dans le peu de contraintes imposées à la fonction à optimiser. Notamment, cette dernière n'a pas besoin d'être dérivable, contrairement aux méthodes à base de gradient. De plus, l'utilisation d'une population est une mesure efficace pour éviter certains minima locaux et celle-ci peut être efficacement exploitée pour optimiser plusieurs objectifs simultanément.

3.3. Les algorithmes génétiques.

Les algorithmes génétiques sont des méthodes adaptatives qui peuvent être utilisées pour résoudre des problèmes d'optimisation ou de recherche de solutions. Leurs principes de



base, dans le cadre de l'optimisation mathématique, ont été rigoureusement définis par John Holland dès 1975 [Hol1975]. Cependant, la puissance de calcul des ordinateurs de l'époque n'était alors pas suffisante pour l'utilisation des algorithmes génétiques sur des problèmes réels de grande taille. Leur développement s'est accéléré à partir de 1989, notamment grâce aux travaux de synthèse présentés par David Goldberg [Gol, 1989].

3.3.1. Les opérations sur l'algorithme génétique.

3.3.1.1. Génome des individus.

Traditionnellement, le génome des individus d'un algorithme génétique est constitué par un chromosome unique de longueur fixe et composé de gènes issus de l'alphabet binaire $\{0,1\}$. On fait souvent référence au chromosome d'un individu par l'appellation "chaîne de bits". Cette représentation a été choisie par Holland car elle permet un support simple des opérations évolutionnistes et elle est, de plus, applicable à un grand nombre de problèmes. On peut en effet coder simplement à l'aide d'une chaîne de bits l'ensemble des entiers ou des réels, et donc optimiser un grand nombre de fonctions mathématiques.

1	1	0	1	1	0	0	0	1	0	1	1
---	---	---	---	---	---	---	---	---	---	---	---

Figure 3: Un chromosome selon Holland.

L'utilisation des chaînes de bits pour former le code génétique des individus implique la nécessité d'une fonction de traduction du génome en valeurs utilisables par l'opérateur d'évaluation. Bien que cette fonction soit relativement simple dans le cas des entiers et des réels, elle peut devenir rapidement complexe dans le cas du codage de structures complexes comme, par exemple, des réseaux de neurones [Gru, 1994].

Afin de permettre à l'algorithme génétique de converger, le génome des individus doit permettre une représentation complète de toutes les solutions possibles du problème. De plus, toutes les solutions codées à l'aide du ou des chromosomes des individus doivent correspondre à des solutions réalisables, c'est-à-dire des points de l'espace de recherche (principe de *validité*). Il est cependant possible d'adapter un algorithme génétique afin qu'il évite toute solution invalide.

Enfin, Goldberg [Gol, 1989] donne deux principes de base pour la représentation des solutions. Tout d'abord, le code génétique des individus doit contenir ou permettre la formation de *blocs de construction* contenant une information ayant du sens. Un *bloc de*



construction est un groupe de gènes qui ont co-évolué de façon à ce que son introduction dans un chromosome entraîne probablement une augmentation de la *fitness* d'un individu. L'hypothèse de la construction des blocs est qu'un algorithme génétique réalise, en premier, la formation du plus grand nombre possible de blocs différents pour ensuite les recombinaison afin d'obtenir les *fitness* maximales. Ensuite, Goldberg émet l'idée que l'alphabet utilisé pour construire les chromosomes doit être le plus petit permettant une expression naturelle du problème posé, ou plutôt de la forme de sa solution.

3.3.1.2. Création de la population initiale.

La première étape d'un algorithme génétique consiste à créer une population initiale à l'aide d'individus générés aléatoirement parmi les valeurs des domaines de définition du problème à résoudre. Toutefois et afin d'optimiser la convergence de l'algorithme, il est possible de restreindre ces domaines à des intervalles pertinents en fonction de la solution recherchée comme par exemple dans le cas d'optimisation d'angles orientés [Lug, 1997].

3.3.1.3. Evaluation.

Dans la nature, la capacité de survie d'un individu s'exprime par son adaptation à son environnement. Cette pression exercée continuellement sur les espèces provoque, par disparition des plus faibles, une sélection des meilleurs individus. Dans le cadre d'un algorithme génétique, l'adaptation d'un individu sera traduite par une mesure de sa capacité à répondre au problème posé. Cette mesure est appelée *fitness* ou fonction de notation. Cette fonction est dépendante du problème et sera définie par l'utilisateur.

A chaque début de génération, tous les individus de la population sont notés afin de permettre aux techniques de *sélection* de les comparer.

La convergence de l'algorithme génétique est conditionnée par la notation. Celle-ci ne doit pas être trop sévère afin de ne point privilégier les individus les plus forts car ceci peut entraîner une convergence rapide qui peut s'avérer néfaste par la suite. En effet, au début d'une simulation, les individus générés aléatoirement sont vraisemblablement éloignés de la solution optimale. Ainsi, s'ils sont favorisés trop tôt, l'algorithme risque de converger vers un optimum local et non vers la solution optimale. Inversement, la fonction de notation ne doit pas non plus être trop lâche: si la pression de l'environnement est insuffisante, la convergence sera trop lente ou inexistante.



3.3.1.4. Sélection.

La phase de *sélection* consiste à allouer des opportunités de reproduction à chaque individu. Le principe est de copier les individus de la population dans une population intermédiaire faisant office de bassin de reproduction ("mating pool"), les individus les plus adaptés étant recopiés plusieurs fois alors que les plus faibles auront tendance à ne pas être copiés. Dans un contexte de remplacement strict de la population, de génération en génération, la taille du bassin de reproduction est identique à celle de la population d'origine. Une fois la population intermédiaire constituée, des paires d'individus sont choisies aléatoirement afin de se reproduire (phase de *reproduction*), leur éventuelle progéniture constituant la nouvelle génération. Cette opération sera répétée jusqu'à ce que le bassin de reproduction soit vide. Le comportement d'un algorithme génétique dépend de la façon dont les individus sont sélectionnés pour faire partie du bassin de reproduction.

Il existe de nombreux opérateurs de *sélection* mais on peut identifier deux stratégies distinctes. La première consiste à réaliser une mise à l'échelle des *fitness* de chaque individu et de se servir de ces valeurs pour déterminer son nombre de copies dans le bassin de reproduction. Les méthodes les plus courantes utilisant cette stratégie sont le reste stochastique, la roulette pipée et la sélection par rang. La seconde stratégie permet de sélectionner directement les reproducteurs potentiels sans passer par un calcul modifiant les notes des individus. La méthode la plus courante dans ce cas étant la sélection par tournoi.

Le reste stochastique.

Le nombre de copies d'un individu x est égal à la partie entière du rapport entre sa fitness f et la fitness moyenne \bar{f} de la population. Cette méthode est surtout utilisée pour des populations de petite taille car elle est moins biaisée que la roulette pipée, dans ce cas.

La roulette pipée.

Dans le cas de la sélection par roulette pipée, la population intermédiaire est assimilée à une roue divisée en secteurs. Chaque secteur est occupé par les copies d'un individu unique et la taille des secteurs est proportionnelle à la valeur P_{sel} calculée par la formule:

$$P_{sel} = \frac{f(i)}{\sum_{i=1}^n f(i)} \quad \text{avec } f(i) \text{ la fitness de l'individu } i, \\ n \text{ le nombre d'individus de la population.}$$



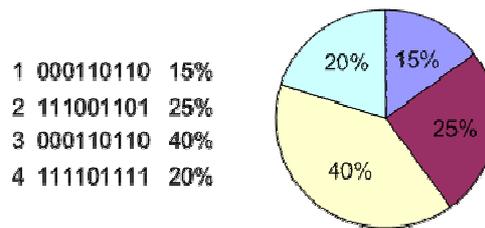


Figure 4: Exemple de roulette pipée pour une population de 4 individus en fonction de Psel.

La sélection par rang.

La méthode de sélection par rang permet de limiter la prédominance d'un individu extrême qui peut apparaître lors de l'utilisation de la roulette pipée. Les individus sont classés en fonction de leur fitness originale: généralement, les individus les plus forts sont de rang 1 et les plus faibles de rang n , n étant la taille de la population. Un pourcentage de sélection est ensuite défini en fonction du rang des individus. Le tableau 1 fournit un exemple de sélection par rang pour six individus. Avec cette méthode de sélection, tous les individus ont une chance d'être sélectionnés. Cependant, elle conduit à une convergence plus lente vers la bonne solution car les meilleurs chromosomes ne diffèrent pas énormément des plus mauvais.

Individus	1	2	3	4	5	6	Total
Probabilité de sélection initiale (roulette pipée)	70%	6%	10%	4%	8%	2%	100%
Rang	6	3	5	2	4	1	21
Probabilité de sélection modifiée (sélection par rang)	29%	14%	24%	9%	19%	5%	100%

Tableau 1: Probabilités de sélection dans le cas de la roulette pipée et de la sélection par rang.

La sélection par tournoi.

La version la plus simple de la sélection par tournoi est le tournoi binaire: une paire d'individu est choisie au hasard dans la population, celui qui a la fitness la plus forte est copiée dans le bassin de reproduction. Ceci est répété jusqu'à ce que le bassin de reproduction soit plein. On peut étendre le nombre de participants au tournoi à n individus, n étant supérieur ou égal à 2.

Cette méthode de sélection est plus élitiste que les précédentes car la probabilité de sélection d'un individu mal adapté (inférieur à la moyenne) est faible. La pression sélective



de cette technique étant conditionnée par le nombre de participants au tournoi, on peut toutefois facilement contrôler l'élitisme en limitant le nombre de participants.

3.3.1.5. Reproduction.

La modification d'un individu à l'aide de l'opérateur de *reproduction* se fait par l'application de deux opérateurs : le *croisement* et la *mutation*. Le rôle de ces opérateurs et d'assurer simultanément une exploration locale de l'environnement (valeurs proches de celles présentes dans la population) ainsi qu'une exploration globale de l'espace de recherche (zones inconnues, non encore visitées par l'algorithme).

A partir du bassin de reproduction, les individus sélectionnés sont aléatoirement groupés par paires. Chaque paire subit l'opérateur de croisement avec une probabilité t_{crois} appelée taux de croisement. L'ensemble des nouveaux individus issus du croisement et des individus non croisés constitue une population intermédiaire. Un opérateur est appliqué sur chaque élément de cette population de mutation avec la probabilité t_{mut} appelée taux de mutation. La population, après application de l'opérateur de mutation, constitue la nouvelle génération.

3.3.1.5.1. Croisement.

Le croisement a pour but d'assurer le brassage des caractères génétiques des individus. La recombinaison des gènes des parents ayant pour but de produire des individus potentiellement mieux adaptés au problème. La combinaison de l'effet de préservation et de concentration des meilleurs individus de l'opérateur de sélection, couplé à l'amélioration générale de la population due au croisement, va permettre à l'algorithme de converger plus ou moins rapidement vers une solution optimale. Les opérateurs de croisement les plus courants sont le croisement à un point de coupe, le croisement à deux points de coupe et le croisement uniforme. Il est important de noter que, hormis la comparaison entre le croisement à 1-point et celui à 2-points, il n'existe pas à priori de croisement qui soit meilleur qu'un autre. En réalité, le choix de l'utilisation d'un type de croisement dépend de la nature du problème, de la structure du chromosome et de l'évolution souhaitée de l'algorithme génétique. De plus, il est fréquent d'utiliser des opérateurs de croisement alternatifs et des opérations additionnelles (inversion de gènes, réorganisation, contrôle de validité des enfants générés) afin d'adapter le croisement au problème pour obtenir une meilleure convergence de l'algorithme.



Croisement à un point de coupe.

C'est le croisement historique des algorithmes génétiques. Son principe est de couper en un point choisi aléatoirement les chromosomes des individus parents, puis d'échanger les parties après ce point de coupe afin de former les enfants [San, 2004].

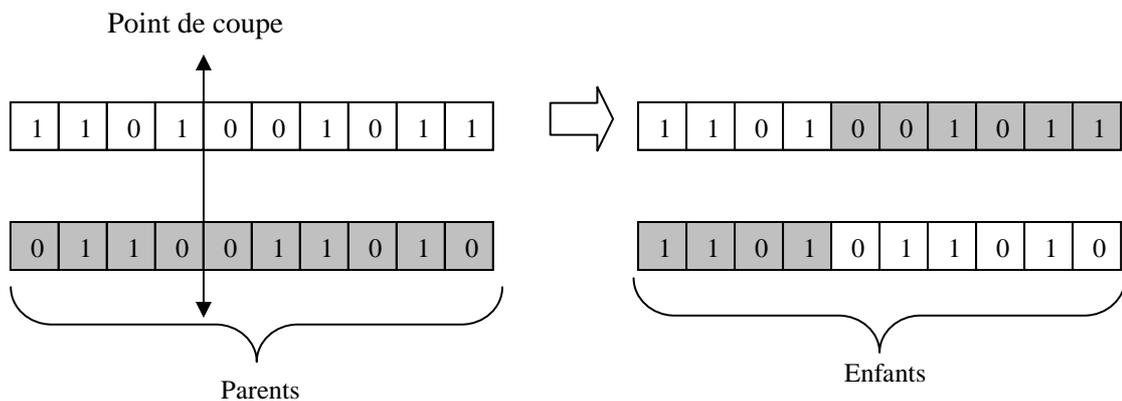


Figure 5: Croisement avec un point de coupe.

Croisement à deux points de coupe.

Afin d'améliorer les performances du croisement, différents algorithmes utilisant plus d'un point de coupe ont été conçus. DeJong [DeJ, 1975] a étudié l'efficacité des croisements avec points de coupe multiples et en a conclu que, si deux points de coupe améliorent la convergence, un nombre plus élevé réduit les performances de l'algorithme. Le problème étant que l'utilisation de nombreux points de coupe perturbe la formation de séries des bits pertinentes à l'intérieur du chromosome ("building blocks").

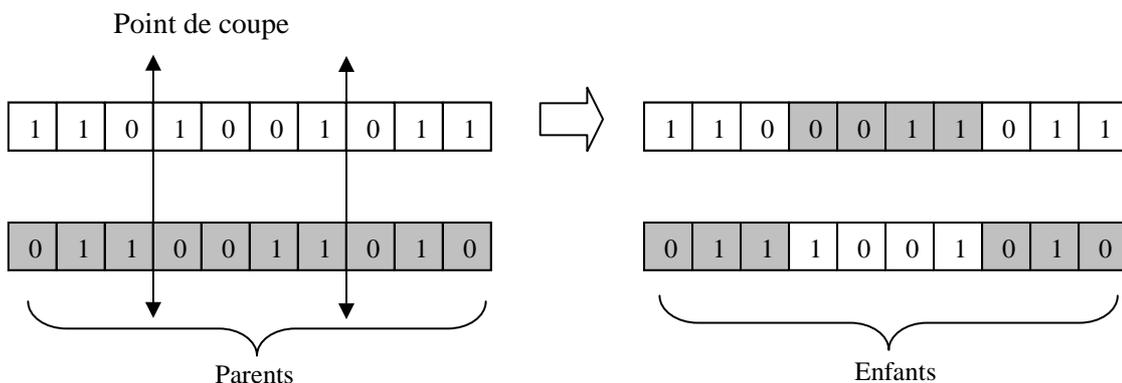


Figure 6: Croisement avec deux points de coupe.



Croisement uniforme.

Le croisement uniforme est radicalement différent des croisements à un ou plusieurs points de coupe. Un masque de croisement est aléatoirement constitué. Le premier enfant est formé en copiant les gènes du premier parent dans le cas d'un bit de masque correspondant égal à 1 et ceux du second dans le cas contraire (bit de masque égal à zéro). Le second enfant est obtenu en inversant les parents. Habituellement, un masque de croisement est généré pour chaque paire de parents.

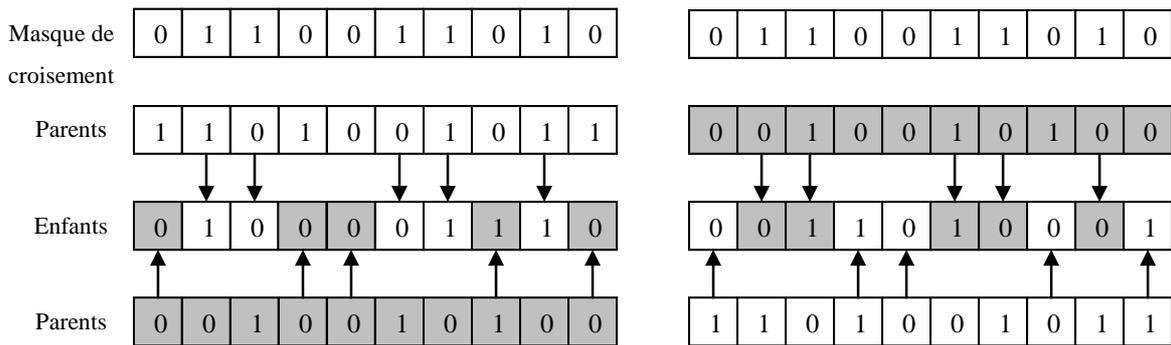


Figure 7: Croisement uniforme.

3.3.1.5.2. Mutation.

Le rôle de la mutation est de maintenir la diversité génétique au sein de la population d'individus. Cet opérateur va perturber aléatoirement le code génétique d'un individu et, ainsi, limiter la dérive génétique induite par un opérateur de sélection trop élitiste (élimination des solutions peu prometteuses a priori mais pouvant s'avérer primordiales pour arriver à la solution optimale). Le taux de mutation est généralement peu élevé et l'importance de cet opérateur par rapport à la sélection ou au croisement pourrait paraître faible pourtant. Bien que secondaire, son rôle est nécessaire à la convergence de l'algorithme.

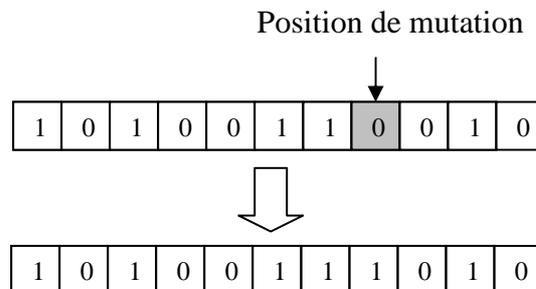


Figure 8: Mutation appliquée à un individu.



3.3.1.6. Critère d'arrêt

Le critère d'arrêt intuitif d'un algorithme génétique est obtenu lorsque la solution définie par le génome d'un individu est la solution optimale connue du problème posé. Cependant, une telle solution n'est pas connue dans la plupart des problèmes pour lesquels cette méthode d'optimisation est utilisée. Afin de pallier cette difficulté, de savoir si l'algorithme a effectivement trouvé la solution optimale, plusieurs techniques sont couramment utilisées. La première consiste à arrêter l'algorithme au bout d'un certain nombre de générations. La seconde est de considérer qu'on a obtenu une solution optimale lorsque le meilleur individu n'a pas été amélioré depuis un certain nombre de générations, ce qui signifie que l'algorithme converge. Enfin, une dernière technique arrête l'algorithme lorsque la diversité génétique au sein de la population n'est plus suffisante pour explorer plus profondément l'espace de recherche des solutions [San, 2004].

3.3.2. Les paramètres d'un algorithme génétique.

3.3.2.1. La taille de la population.

Les conditions de convergence changent avec la taille de la population : lorsque celle-ci est grande, sa diversité augmente ce qui diminue la convergence vers un optimum local, le temps d'exécution de chaque génération augmente. Consécutivement, la recherche risque de s'effectuer d'une façon redondante et l'efficacité de l'algorithme est globalement affectée. Par contre, si la taille de la population est petite, alors la probabilité de s'attarder sur des minima locaux est grande. La taille selon le cas se situe entre 25 et 100.

3.3.2.2. Le taux de croisement.

Il détermine la proportion des individus qui sont croisés parmi ceux qui remplaceront l'ancienne génération. L'opérateur de croisement est appliqué avec une probabilité P_c . Plus cette valeur est élevée, plus de nouvelles structures (individus) sont introduites dans la nouvelle génération, les structures performantes étant trop fréquemment détruites. Par contre, si ce taux est trop bas, la population n'évolue pas assez vite. En général, P_c varie entre 0.25 et 0.70.



3.3.2.3. Le taux de mutation.

L'opérateur de mutation est appliqué avec une probabilité P_m ; si ce taux est grand alors la recherche devient purement aléatoire, la population est diversifiée et l'AG perd de son efficacité. Si au contraire ce taux est faible, la population est moins diversifiée et en plus il y a risque de stagnation. Des études empiriques conseillent, pour l'obtention de bons résultats, une fréquence qui se situe autour d'une mutation tous les 1000 bits [Las, 2004].

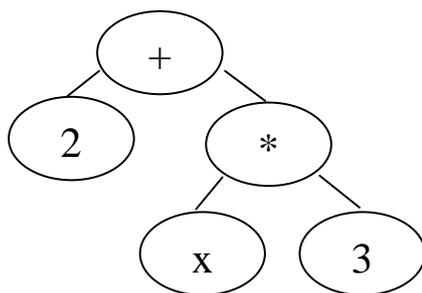
A noter que ces paramètres dépendent étroitement du type de problème à résoudre.

3.3.2.4. Le fossé des générations.

K. DE JONG [Sys, 1989] a proposé de ne pas remplacer l'ensemble de la population à chaque génération et pour cela, il a introduit la notion d'écart entre les générations, un nombre compris entre 0 et 1 qui indique la proportion de parents qui sont remplacés par leurs descendants. Si ce taux est à 1, l'ensemble de la population est remplacé.

3.4. La programmation génétique.

Cette méthode introduite par [Koz, 1992] est fondée sur une nouvelle représentation des individus composant la population. Les chromosomes sont ici constitués de $S_Expressions$ et non plus de chaînes (figure 9). Cette représentation sous la forme d'arbres permet d'utiliser des structures non linéaires et de taille variable; elle se place donc en une approche de plus haut niveau par rapport aux algorithmes génétiques.



Dans cette approche, la valeur d'un individu est fournie directement par interprétation de l'arbre qui le compose. Ici, l'individu correspondra à la fonction $f(X) = 2 + (X*3)$.

Figure 9 : Un chromosome fondé sur les $S_expressions$.

Effectuer un parallèle entre des opérations sur les arbres et les opérations précédemment décrites est relativement aisé. Les opérations de modification des individus telles que le croisement (figure 10) ou la mutation (figure 11) seront donc une transposition directe de



celles vues précédemment. La structure arborescente apportant simplement un remplacement de sous arbres au lieu d'un échange de morceaux de chaînes.

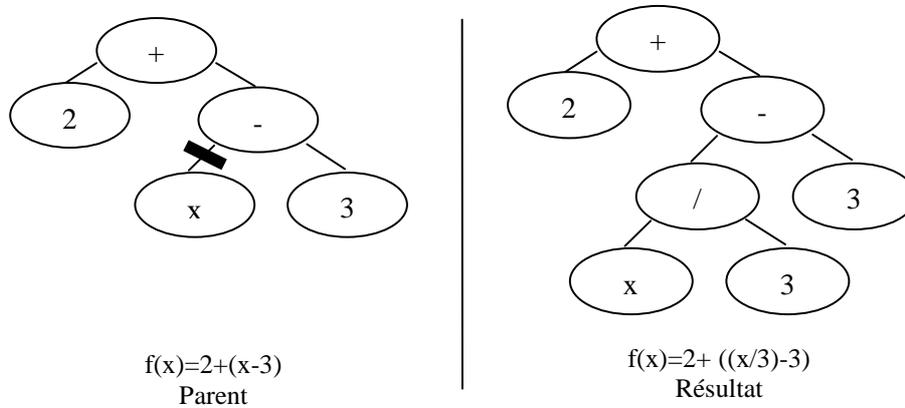


Figure 10: La mutation sur les arbres.

Dans cette approche, la mutation est beaucoup moins employée que dans les algorithmes génétiques car le croisement allié à la taille variable des individus suffit à ne pas perdre les informations nécessaires à la solution (maintient d'une cohérence et d'une diversité dans la population).

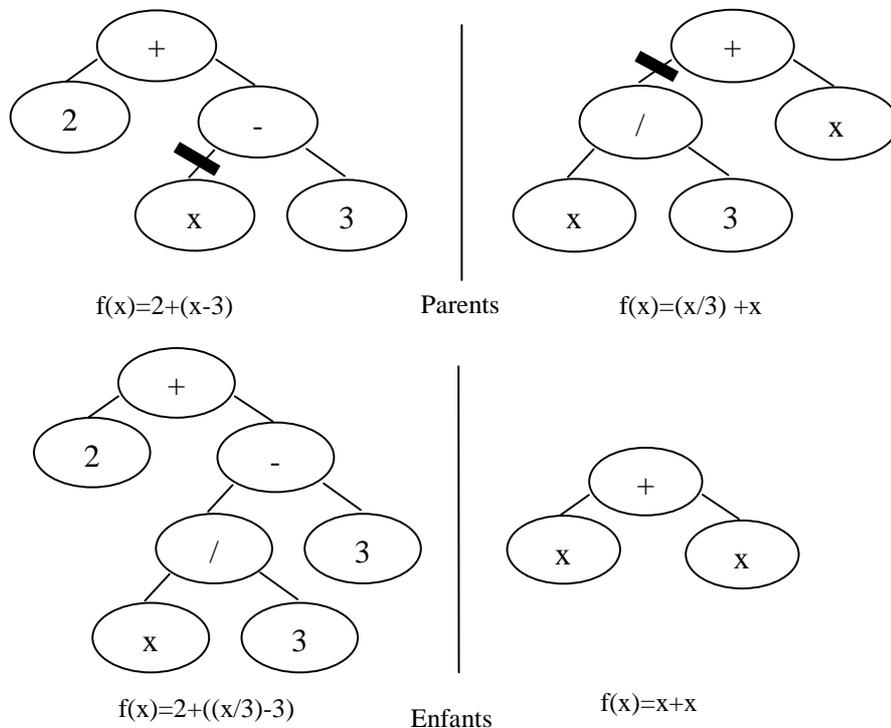


Figure 11 : Le croisement génétique portant sur les arbres sous forme de S_Expressions.



Avec cette méthode, la recherche de la structure des individus nécessite un premier stade: la définition d'un alphabet permettant de décrire l'espace des solutions du problème. Cet alphabet doit répondre à un certain nombre de contraintes. Il doit être suffisant (c'est à dire permettre de résoudre le problème) et fermé (chaque élément du langage peut être argument de chaque autre). Cet alphabet est de haut niveau et va donc permettre une interprétation directe des individus par lecture de leur arbre [Lug, 1997].

3.5. Les stratégies d'évolution

Les stratégies d'évolution forment une famille de métaheuristiques d'optimisation. Elles sont inspirées de la théorie de l'évolution et appartiennent, à ce titre, à la classe des algorithmes évolutionnaires.

La méthode fut initialement proposée par Ingo Rencherberg, en 1965 [Rec, 1965], à l'université technique de Berlin, en Allemagne. Elle est, à ce titre, la première véritable méta-heuristique et le premier algorithme évolutionnaire, bien avant le recuit simulé ou les algorithmes génétiques. La méthode fut ensuite développée durant la fin des années 1960, principalement par les travaux de Ingo Rechenberg, P. Bienert et Hans-Paul Schwefel sur la conception de profils aérodynamiques.

Par la suite, les stratégies d'évolutions (evolution strategies, abrégé ES), furent utilisées pour résoudre des problèmes d'optimisation continus, discrets, contraints, multi-objectifs, etc.

Dans sa version de base, l'algorithme manipule itérativement un ensemble de vecteurs de variables réelles, à l'aide d'opérateurs de mutation et de sélection. L'étape de mutation est classiquement effectuée par l'ajout d'une valeur aléatoire, tirée au sein d'une distribution normale. La sélection s'effectue par un choix déterministe des meilleurs individus, selon l'échelle de valeur de la fonction objectif.

4. Les réseaux de neurones.

Avant 1943, l'intelligence artificielle était symbolique. Le raisonnement et l'intelligence étaient perçus comme des manipulations de symboles. Cette année là, Warren McCulloch et Walter Pitts présentèrent leur modèle de neurone formel [McP, 1943] et l'intelligence devient computational. En démontrant qu'un réseau de leurs neurones formels est capable



de simuler toutes les opérations de la logique booléennes (ET, OU, NON, etc.), ils entendent prouver d'une part que l'intelligence provient uniquement du fonctionnement des neurones dans le cerveau, et d'autre part que les neurones mettent effectivement en œuvre des processus de raisonnement logiques³. En 1949, Donald Hebb [Heb, 1949] présente à son tour son modèle de synapse qui permet de rendre compte de processus d'apprentissage mis en œuvre par le cerveau.

4.1. Définition de réseaux de neurones.

Un réseau de neurones artificiels (RNA) est constitué par l'interconnexion d'unités élémentaires de traitement de l'information, dont le principe de fonctionnement est calqué sur celui des neurones biologiques. Un réseau possède des capacités de stockage et de traitement de l'information, imputables aux poids des connexions entre les neurones. Ces poids sont calculés par un processus d'adaptation également appelé apprentissage.

4.2. Les réseaux de neurones naturels.

Le système nerveux de la plupart des espèces naturelles est constitué d'un réseau de cellules spécialisées appelées neurones. Les neurones sont des constituants élémentaires de la plupart des organes de réflexion des systèmes intelligents. Ils permettent de traiter des informations à un niveau élémentaire, leur interconnexion fournissant le support de la complexité. Le fonctionnement d'un neurone est ainsi assez simple alors que celui résultant de l'interconnexion d'un millier voire de milliards de telles cellules est très complexe.

Ces cellules très spécialisées sont constituées de trois parties principales:

- Les dendrites qui vont servir de capteurs, permettant au neurone d'acquérir des informations.
- Le corps de la cellule qui va traiter cette information.
- L'axone qui va leur permettre d'émettre de l'information une fois le traitement effectué. Ce dernier est connecté à des effecteurs tels que des muscles ou aux dendrites d'autres neurones.

³ Ces travaux sont surtout la base de l'approche biomimétique, qui consiste à s'inspirer des structures et de la dynamique d'organismes vivants pour produire des machines intelligentes. Cette approche prendra de l'importance plus tard avec la vie artificielle.



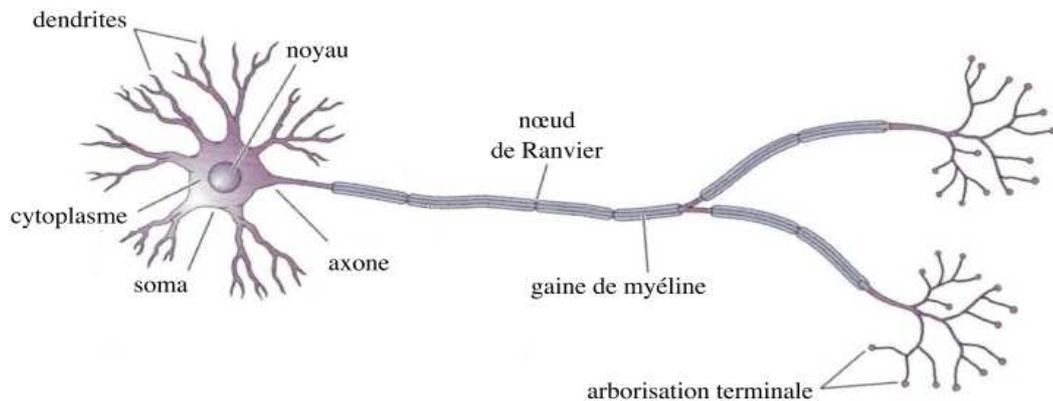


Figure 12: Schématisation d'un neurone biologique. Le neurone est une cellule qui possède des prolongements de 2 types : Un axone et des dendrites.

Le fonctionnement apparent d'un neurone est de fait relativement simple: il agit comme un intégrateur qui va propager sur son axone l'information présente sur ses dendrites si la somme de ces informations atteint un certain seuil. De ce fait, ce fonctionnement simplifié ne suffit pas à expliquer tout le fonctionnement d'un neurone car de nombreux paramètres tels que la fréquence des signaux reçus semble prendre une part non négligeable dans le déclenchement et la forme de l'impulsion produite par l'axone.

La connexion des différents neurones produits un réseau dont la topologie va déterminer le comportement. Le cerveau humain est ainsi constitué de 1011 neurones. Ce nombre important ne permet évidemment pas une interconnexion complète de tous les neurones du système nerveux (La boîte crânienne devrait avoir une taille de 10 kilomètres pour loger un tel cerveau).

4.3. Les réseaux de neurones artificiels.

L'objectif des réseaux de neurones artificiels est donc de recopier et reproduire les mécanismes liés au fonctionnement des réseaux de neurones naturels.

Le neurone artificiel.

On simule donc le comportement d'un neurone en calculant ses valeurs de sortie en fonction de ses valeurs d'entrée. Les entrées et les sorties étant bien entendu liées à d'autres neurones ou aux capteurs et effecteurs de l'entité considérée. [Lug, 1997]



Le neurone formel de McCulloch et Pitts.

Le neurone formel [McP, 1943] (figure 13) est aussi une abstraction mathématique du neurone biologique. A l’instar de son homologue biologique, il est le maillon élémentaire de la transmission du signal au sein du réseau.

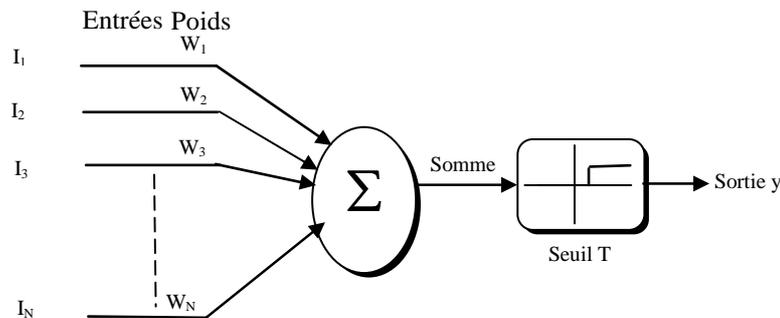


Figure 13: Le neurone formel de McCulloch et Pitts est une abstraction mathématique du neurone biologique. Il décide de son éventuelle activation en appliquant une fonction de transition ϕ à la somme Σ des potentiels qu’il reçoit en entrée.

Le neurone formel accepte plusieurs entrées E_i provenant d’autres neurones. Ces entrées sont des connexions afférentes qui proviennent d’autres neurones, si bien que la valeur E_i de chaque entrée correspond à la valeur de la sortie S du neurone afférent correspondant. Chaque connexion afférente est pondérée par un réel $w_i \in \mathbb{R}$. Fondamentalement, ce poids représente la “taille” de la synapse, en d’autres termes la quantité de neuromédiateurs présents dans l’axone du neurone pré-synaptique ainsi que la quantité de récepteurs dans la dendrite du neurone post-synaptique. Les connexions inhibitrices sont représentées par un poids $w_i < 0$.

Le corps du neurone est composé de deux parties. La première calcule la somme des potentiels entrants, pondérée par les poids synaptiques. La seconde implémente une fonction de transfert, qui prend cette somme en paramètre et renvoie une valeur de sortie.

Le neurone formel de McCulloch et Pitts est aujourd’hui encore le plus couramment utilisé dans les réseaux de neurones. Cela étant dit, du fait de son extrême simplification, il n’est pas exempt de critiques et d’autres modèles existent.



4.4. L'apprentissage dans les réseaux de neurones.

Les connaissances dans un réseau ne sont pas codées dans les neurones mais dans la configuration du réseau, autrement dit dans les poids des connexions entre les neurones. Le principe d'apprentissage repose sur la modification de ces poids.

Considérons une connexion entre deux neurones i et j (figure14). Les changements s'opèrent sur le poids de cette connexion et de manière incrémentale, ou dichotomique.

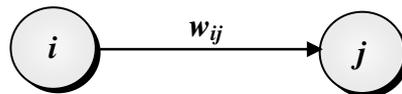


Figure 14: Le scalaire w_{ij} représente le poids associé à la connexion entre les neurones i et j .

Classiquement, il existe deux grandes classes d'apprentissage : l'apprentissage supervisé et l'apprentissage non supervisé⁴. Le premier fait évoluer les poids des connexions en fonction d'une estimation de l'erreur en réponse du réseau. Le deuxième ne possède pas une telle estimation et les changements s'opèrent uniquement en fonction des entrées, à partir de considérations statistiques, par exemple.

4.5. Les réseaux à propagation avant.

Les réseaux à propagation avant ou *feedforward networks* sont des réseaux en couches, c'est à dire dans lesquels les neurones sont répartis dans des couches, et dont l'influx nerveux se propage uniquement de l'entrée vers la sortie.

Le perceptron simple.

Le perceptron simple ou mono-couche de Frank Rosenblatt [Ros, 1958] est le premier réseau de neurones fonctionnel muni d'une méthode d'apprentissage. Dans sa conception première, illustrée par la figure 12, il permet de reconnaître des lettres de l'alphabet, cependant, il peut être utilisé dans n'importe quelle tâche de classification.

⁴ Il n'aura pas échappé au lecteur la similitude des propriétés de l'apprentissage d'un réseau de neurones et de l'adaptation d'un agent à son environnement



Il est composé d'une couche de neurones d'entrée, appelée rétine, à laquelle on fournit les informations sur l'item à reconnaître. Tous les neurones de la rétine sont reliés par des liens pondérés à une couche de sortie où chaque neurone est associé à une classe d'item. Enfin, une procédure d'apprentissage basée sur la correction d'erreurs permet d'"entraîner" le réseau, autrement dit de régler le poids de chaque lien, jusqu'à ce que celui-ci puisse discriminer les classes de tous les items qu'on lui présente. Les neurones sont munis de fonction d'activation de type seuil.

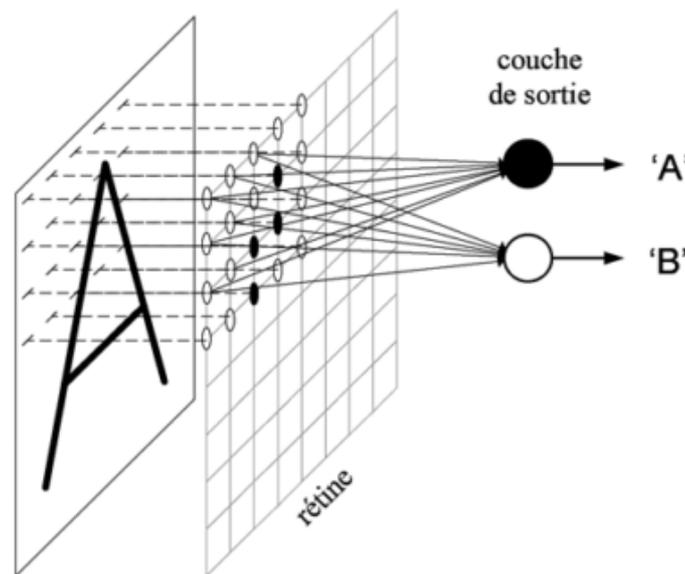


Figure 15: Le perceptron de Rosenblatt utilisé pour la reconnaissance des formes.

Les perceptrons multi-couches.

Deux ans plus tard, Bernard Widrow et Ted Hoff présentèrent ADALINE (Adaptive Linear Neuron) [WH, 1960], une évolution du perceptron de Rosenblatt où les fonctions d'activation des neurones sont désormais linéaires. La règle d'apprentissage utilisée, la règle de Widrow-Hoff, est à la base de la puissance de ce nouveau réseau. Il s'agit d'une adaptation de la méthode des moindres carrés totaux, qui ramène l'apprentissage à une descente de gradient. Son principe, succinctement décrit par l'algorithme 1, est le suivant : Pour chaque neurone j de la couche de sortie, et pour chaque neurone i de la rétine, on applique la règle du delta. Ainsi, pour chaque couple de neurones $\{i, j\}$, le poids de la connexion w_{ij} entre les deux neurones est modifié dans le sens d'une réduction de l'erreur entre l'état d'activation désiré de j et son état d'activation réel.



Données : R la constante d'apprentissage;
pour chaque neurone j de la couche de sortie **faire**
 pour chaque neurone i de la rétine **faire**
 $w_{ij} += R.(d_j - a_j).a_i$
 fin
fin

Algorithme 1 : Algorithme d'apprentissage d'ADALINE.

Dans leur ouvrage, Marvin Minsky et Seymour Papert soulèvent l'incapacité des perceptrons à résoudre des problèmes non linéairement séparables [MP, 1968]. Grâce à l'illustration présentée par la figure 13, on comprendrait qu'il n'est pas possible de trouver une droite permettant de discriminer les deux classes vraie et faux de la fonction logique XOR.

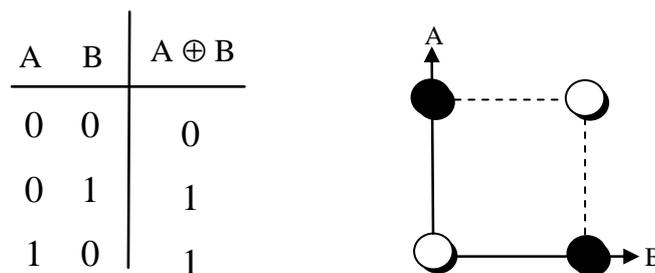


Figure 16: Illustration du problème de la séparation linéaire de la fonction logique XOR.

Ils présentèrent leur modèle de perceptrons multicouches (PCM) censé pallier ce problème en rajoutant une ou plusieurs couches cachées entre la rétine et la couche de sortie. Cependant, ils soutinrent que la majorité des limitations du perceptron simple s'appliquent également au perceptron multicouche.

Le pavé lancé par Minsky et Papert dans le domaine du connexionnisme a eu pour incidence de stopper ou en tout cas de freiner toute recherche sur les perceptrons pendant une décennie. Au début des années 1980, on assiste donc à l'avènement d'un nouveau type de réseau, le réseau associatif, illustré dans les sections suivantes par Hopfield, Kohonen et Grossberg. Malgré cela, le PCM gagnera enfin ses lettres de noblesses quand l'algorithme de rétro-propagation sera proposé par David Rumelhart [RHW, 1986]. Cet algorithme est une extension de l'algorithme 1 au perceptron multicouche. Le principe reste identique :



propager une erreur calculée de la couche de sortie vers les couches cachées afin de tendre vers une erreur minimale au cours des itérations.

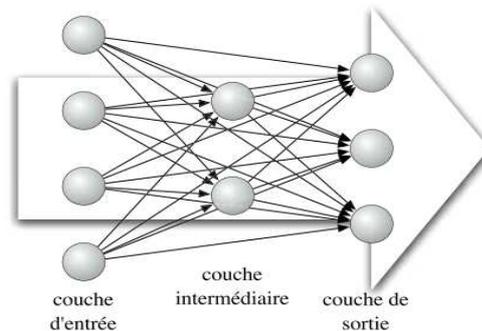


Figure 17: Dans le perceptron multicouche, des couches intermédiaires sont insérées entre les couches d'entrée et de sortie.

Il s'agit d'un apprentissage lent mais très efficace. Malgré son âge, le PCM compte encore parmi les réseaux les plus utilisés. Cependant, il n'est plus l'ultime apanage des réseaux à propagation avant, certaines alternatives récentes s'étant montrées plus innovantes. [Pan, 2008].

A l'heure actuelle, les NEAT [SM, 2002] sont considérés comme les plus performants car ils ont de surcroît la particularité d'adapter leur topologie en fonction de la complexité de l'information à traiter. Ces réseaux montrent qu'il est possible d'augmenter significativement les performances du PCM traditionnel en modifiant légèrement son principe de fonctionnement.

5. Les L-Systemes.

Les processus de la morphogenèse restent mystérieux. Malgré les progrès de la génétique, de nombreux mécanismes nous restent inconnus. A. LINDENMAYER (1925-1989) a proposé une méthode de description formelle de la structuration des plantes. Basée sur une forme récursive de grammaire générative, elle a été approfondie et mise en œuvre graphiquement par P. PRUNSINKIEWICZ dans les années quatre-vingt. Les procédures utilisées sont simples, mais nous ne les décrivons pas ici. Contentons nous de présenter quelques-unes des images obtenues⁵.

⁵ Ces figures ont été réalisées avec L-System 4 de T. Perz : <http://www.geocities.com/tperz/L4Home.htm>. On trouvera des images époustouflantes ici : <http://www.cpsc.ucalgary.ca/Research/bmv/vmm-deluxe/TitlePage.html>.



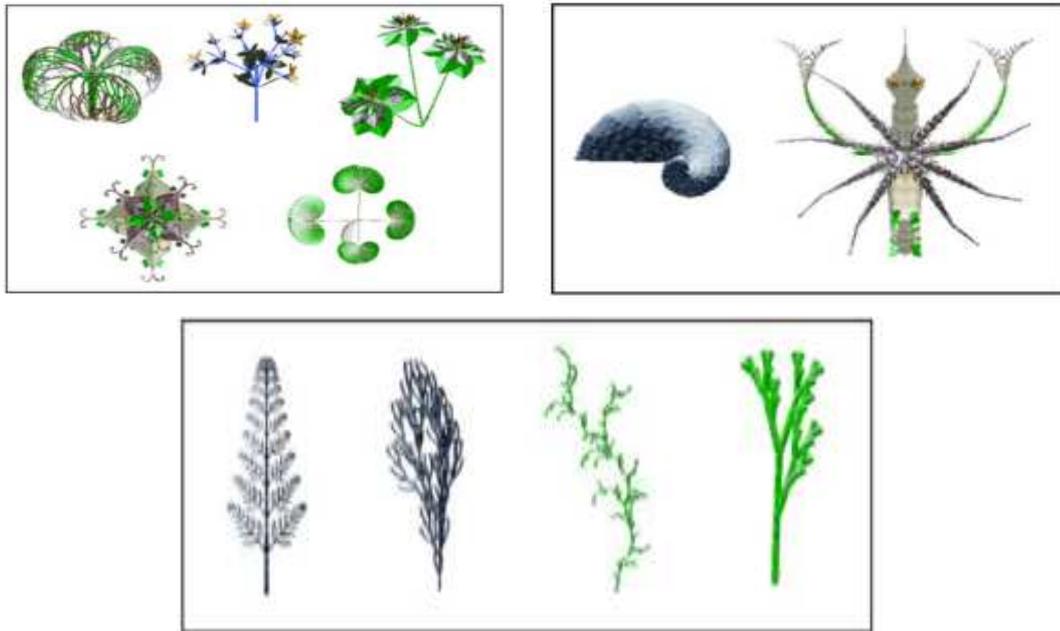


Figure 18: Quelques images obtenues des L-Systèmes

L'inspiration biomimétique de ces figures est indéniable. Elles ne sont pourtant que l'interprétation graphique de chaînes de caractères construites à partir d'un algorithme récursif. Il est très difficile de cerner le lien entre la morphogenèse végétale et les L-systèmes. De même que les paysages engendrés par les fractales n'ont en aucune manière, malgré leur ressemblance avec la réalité, suivi une évolution comparable aux paysages naturels, les similitudes entre plantes et L-systèmes ne signifiant nullement que les génomes végétaux décrivent un mécanisme identique. Les L-systèmes n'en confortent pas moins l'hypothèse selon laquelle les processus morphogénétiques mettent en œuvre des procédures répétitives, probablement souvent récursives. [Ren, 2003].

6. Evolution de contrôleurs pour un agent autonome.

Les méthodes évolutionnistes semblent bien adaptées à la génération des contrôleurs pour agents⁶ autonomes: après tout, des agents biologiques ont été produits par un algorithme évolutionniste. Il est clair que pour une telle tâche, le noyau du problème se situe non

⁶ Ici le mot "agent" doit être pris dans le sens le plus général de "quelque chose qui agit", qui correspond à son étymologie. Il implique seulement que l'agent considéré doit exécuter un certain genre d'action dans un environnement donné.



seulement dans le choix d'une méthode évolutionniste particulière, mais également dans le choix d'une représentation adéquate pour le contrôleur. [MC, 2005].

6.1. Les générateurs de contrôle de Pattern.

Une manière simple pour contrôler une articulation dans le corps d'un agent est de spécifier un pattern d'activation sous forme de valeurs d'oscillation.

Ces valeurs sont alors appliquées aux actionneurs qu'ils contrôlent, ayant pour résultat les mouvements d'oscillation. Le rôle du contrôleur est de fournir un pattern adéquat vis à vis de la tâche actuelle (habituellement une tâche locomotive). Cette méthode, qui peut sembler un peu artificielle en premier lieu, est en fait enracinée profondément chez l'animal neurobiologique, duquel provient l'origine du terme générateur centrale de patterns (CPG).

A titre d'exemple, Ijspeert et ses collègues [IHW, 1999] ont utilisé les modèles biologiques pour faire évoluer une lamproie de natation. La lamproie se compose de plusieurs segments, chacun contenant quelques neurones inter-couplés. Ces neurones peuvent produire un Pattern d'oscillation dans un segment singulier. Les segments interconnectés ensemble sont capables de synchroniser leurs oscillations dans une vague de mouvements coordonnés. L'information proprioceptive (c-à-d. entrée de l'état d'un corps) permet à la lamproie de maintenir un mouvement dans un environnement désordonné (courants, etc.). Les auteurs ont pu faire évoluer avec succès les connections neuronales d'une lamproie artificielle.

En utilisant le même cadre, le même auteur a fait une grande avancée évolutionniste en traitant un groupe entier de poissons, et a par ailleurs pu passer de la Lamproie au Salamandre tout en conservant pour la lamproie des capacités de nager dans l'eau. Il a également fait évoluer des contrôleurs pour la locomotion quadrupède sur la terre [IHW, 1999b].

Des Patterns d'oscillation complexes peuvent également être générés plus directement en spécifiant simplement un ensemble de composants de Fourier. Dans un tel modèle, on fait évoluer les montants de CPG en faisant évoluer un ensemble de paramètres numériques de la série de Fourier qui décrit un pattern périodique d'activation. Une polarisation permet de s'assurer que des composants de basse fréquence sont favorisés. Cette approche élégante, décrite par Arnold [Dir, 1997], a été appliquée aux tâches



locomotives simples pour les organismes 3D articulés (tout à fait semblables à ceux de Sims [Sim, 1994], voir 2^{ème} chapitre) et a produit un éventail de comportements intéressants : rampement, saut et bond.

6.2. Les systèmes de classeurs.

Dans cette partie, nous allons décrire le fonctionnement du système de classifieurs de type LCS (Learning Classifier System [Gol, 1989] [Hol, 1986]) qui correspond à la version ultérieure à CS-1. En effet, le LCS intègre un nouvel algorithme de répartition des crédits appelé « Bucket Brigade » [Hol, 1980] que nous détaillerons dans les paragraphes suivants.

6.2.1. Présentation.

Un système de classifieurs est un système adaptatif qui comprend des règles syntaxiques de la forme <condition>::<action> appelées classifieurs. Avec ce type de système, une règle est activée si sa partie condition est satisfaite. Elle peut déclencher directement une action ou bien d'autres conditions qui aboutiront finalement à une action. Dans ce cas, les parties "action" intermédiaires sont perçues comme des conditions. L'évolution de la base de règles est guidée par des récompenses provenant de l'environnement. Le système de classifieurs intègre un algorithme génétique comme moteur de cette évolution. L'apprentissage continu permet l'émergence de règles efficaces et leur réutilisation au moment approprié.

Un système de classifieurs est constitué de six composantes principales (figure 16) :

- une interface d'entrée composée de capteurs ;
- une base de règles (liste de classifieurs) ;
- une liste de messages ;
- un système de répartition des crédits (l'algorithme "Bucket Brigade") ;
- un algorithme génétique ;
- une interface de sortie composée d'effecteurs.



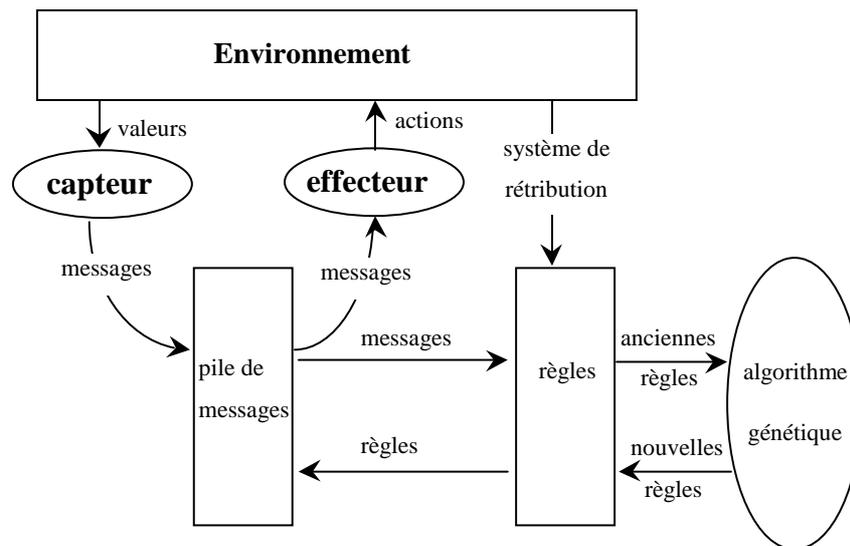


Figure 19 : Système de classifieurs de type LCS.

Un tel système est appelé système de production car il utilise un algorithme basé sur des règles. Son fonctionnement est comparable à celui des systèmes experts [Cap, 1998], à la différence près que les systèmes de classifieurs utilisent un alphabet fini, donc une syntaxe beaucoup plus simple qui autorise des opérateurs génétiques et donc une exploration très vaste des différentes combinaisons [San, 2001].

Les systèmes de classeurs sont à l'heure actuelle employés dans un nombre croissant de problèmes et notamment pour des environnements dynamiques [Har, 1992]. En effet, ces systèmes mettent en évidence leurs capacités à s'adapter dans un environnement en perpétuelle évolution. De plus, leurs aptitudes temps-réel (l'inférence sur des règles aussi simples est quasi-instantanée) leur procure un grand champ d'application dans tous les domaines afférents à la simulation [Dik, 1993] et notamment en robotique et en avionique.

6.3. Réseaux de neurones évolutifs.

L'application des algorithmes évolutionnistes pour la génération des réseaux de neurones semble une idée simple. Théoriquement, la flexibilité des algorithmes évolutionnistes (qui exigent seulement une fonction de fitness représentant un indicateur de progrès) pourrait permettre de surmonter les limitations de la formation arrière de propagation, particulièrement quand elle concerne les architectures complexes et récurrentes.



Toutefois, il est admis que l'application des algorithmes évolutionnistes aux réseaux de neurones n'est pas sans problèmes, spécialement en ce qui concerne l'opérateur de croisement. Les paramètres des réseaux de neurones sont fortement complexes et interdépendants, ainsi le croisement de deux réseaux efficaces est sujet à la production de descendants plus pauvres (cette interdépendance des paramètres dans une représentation génétique est connue comme épistasie des algorithmes évolutionnistes).

D'autre part, deux réseaux de neurones qui correspondent aux architectures équivalentes peuvent être très différents, juste parce que les neurones semblables se produisent dans des ordres différents dans les deux réseaux; ceci peut poser des problèmes pour des opérateurs de recombinaison (par exemple l'application du croisement, entre deux réseaux, commandée différemment, perd certains nœuds et en reproduit d'autres). C'est essentiellement le problème de la permutation, également connu sous le nom de problème « de convention de concurrence ». Cependant, ce problème s'est avéré moins grave que prévu [Han, 1992].

Toujours, un certain nombre d'auteurs ont incité à abandonner l'utilisation exclusive du croisement, et à compter exclusivement sur les opérateurs basés mutation. Un aperçu impressionnant de l'application des techniques évolutionnaires aux réseaux neurologiques a été décrit par Yao [Yao, 1999] (qui favorisent également l'évolution exclusive de la mutation). Une autre manière de procéder est de faire co-évoluer coopérativement des neurones dans un réseau. Cette méthode a été appliquée de plusieurs manières par des collaborateurs de Miikkulainen, par exemple [GM, 1997, MM, 1996].

Une manière de préserver des structures dans l'évolution du réseau neurologique, particulièrement quand les architectures doivent évoluer ainsi que leurs poids, est de ne permettre la recombinaison qu'entre les nœuds équivalents. On peut noter que si tous les réseaux commencent à partir des réseaux semblables et élémentaires, alors n'importe quelle structure différente doit provenir d'une mutation (l'addition, ou la soustraction d'un ou plusieurs nœuds). Dans ce contexte, les nœuds équivalents sont des nœuds qui correspondent à des versions différentes du même nœud original. Une seule manière pour détecter ceci consiste, à chaque fois qu'un nœud est créé par une mutation, à l'étiqueter avec une identification unique ID (par exemple un nombre séquentiel simple), et tous ses descendants qui portent ce nœud devraient garder cette étiquette. De cette façon il sera possible de ne recombiner que les nœuds semblables. C'est, essentiellement, l'idée véhiculée par NEAT, ou « *Neuro-Evolution of Augmenting Topologies* » [SM, 2002].



Cette idée, qui a utilisé l'étiquette, fournit, non seulement une manière de contrôler le croisement; elle permet également, d'abord de comparer deux réseaux et de déterminer leur degré de « condition parentale », c-à-d. leur proximité généalogique. Ceci permet, de ce fait, de fournir une manière objective de classer des individus dans les espèces (que NEAT permet de réaliser). Il pourrait également permettre de tracer le progrès évolutionniste en déterminant les additions les plus bénéficiaires (celles qui sont fréquentes dans le patrimoine héréditaire courant) et celles qui ne le sont pas.

7. Conclusion.

Dans ce chapitre, nous avons décrit les différentes techniques qui peuvent être utilisées dans le domaine de la vie artificielle et dans la génération des créatures artificielles. Nous pouvons ainsi conclure que les algorithmes génétiques sont moins adaptés à une utilisation en environnement dynamique que la programmation évolutionniste mais ils permettent, néanmoins, de définir des réseaux de neurones ou des automates cellulaires qui peuvent être très utiles pour la définition de contrôleurs (notamment en robotique [Gru, 1994] [Col, 1992]).

Dans le chapitre qui suit, nous présenterons les différents travaux sur les trois axes principaux de la génération de créatures artificielles qui sont : la morphogénèse, l'embryogénèse et la génération de contrôleur avec une morphologie fixée qui est notre domaine d'investigation.



Chapitre II : Les Créatures Artificielles

Les créatures artificielles

Dans ce chapitre, nous allons définir les créatures artificielles. De même, nous allons présenter quelques travaux sur celles-ci. Concernant le choix des travaux, celui-ci est basé sur leur popularité, ces travaux ayant été fait récemment. A la fin de ce chapitre, nous nous sommes proposé de présenter une synthèse de ses travaux selon un certain nombre de critères que nous définirons.

1. Définition de créatures artificielles

Nous définissons une créature artificielle évolutionniste comme étant une entité virtuelle ou réelle possédant une morphologie et un comportement issu de processus évolutionnistes. En nous référant à la définition de la vie artificielle de J. Doyne Farmer et Alleta d'A. Belin [FB, 1989], nous établissons la définition des créatures artificielles et des propriétés minimales d'une créature artificielle :

L'être humain a contribué au processus d'apparition de toutes créatures artificielles ;

1. Une créature artificielle est une unité ;
2. Une créature artificielle possède une morphologie ;
3. Une créature artificielle doit interagir avec son environnement.

Les propriétés suivantes ne sont pas indispensables :

4. Une créature artificielle est autonome ;
5. Une créature artificielle s'auto produit ;
6. Une créature artificielle peut se reproduire elle-même ;

7. Une créature artificielle est une unité ;
8. Une créature artificielle s'auto répare ;
9. Une créature artificielle possède des mécanismes d'adaptation.

Une créature artificielle virtuelle est une créature simulée qui a une représentation 2D/3D dans un environnement virtuel [LLD, 2008].

2. Morphogénèse artificielle.

Alors que la simulation comportementale concerne uniquement des morphologies figées, la morphogénèse permet de faire évoluer la morphologie et le comportement des créatures en assemblant différents blocs fonctionnels. Ceci permet de générer des créatures mieux adaptées à leur environnement en se focalisant sur des comportements de plus bas niveau. Ainsi, alors que la simulation comportementale va plutôt travailler sur le comportement global d'humanoïdes ou d'avatars, l'évolution conjointe de la morphologie et du comportement permettra de retrouver par évolution des comportements basiques tels que la marche ou la nage.

Le comportement de ces créatures est souvent basé sur des réseaux de neurones ou des systèmes de classeurs. L'évolution de la morphologie implique une représentation logique de celle-ci permettant son codage dans un génome. Ainsi, un algorithme évolutionnaire pourra faire évoluer cette morphologie et son comportement dans l'environnement.

Les travaux sur la morphogénèse ont pour objectif de générer des créatures artificielles adaptées à leur environnement. Par exemple, dans ses travaux, Karl Sims [Sim94a] utilise des blocs possédant différentes propriétés telles que la taille, la forme, la position des différents capteurs de contact, etc. Sur un modèle relativement similaire, Komosinski proposa Framstick [KU, 1999], des bâtonnets remplacent ici les blocs mais les créatures fonctionnent sur le même principe que celui utilisé par les créatures de Sims. Elles utilisent un réseau de neurones pour coordonner leurs mouvements.

Nicolas Lassabe a approfondi les travaux de Karl Sims en utilisant des environnements plus complexes [LLD, 2007]. Les créatures de Lassabe sont capables de monter un escalier ou de faire de la planche à roulette. Il existe également un nombre non négligeable de travaux sur la morphogénèse que nous allons détailler par la suite dans ce chapitre [SLD, 2008].



2.1. Evolution de la morphologie seule.

Il y a peu de travaux où l'on fait évoluer seulement la morphologie d'une créature sans lui donner un comportement. Certains travaux en embryogenèse se sont consacrés uniquement au processus de développement des formes comme ceux de Chavoya, Doursat et Fleischer [CD, 2007a]; [CD, 2007b]; [Dou, 2006]; [Dou, 2007a]; [Dou, 2007b]; [Fle, 1996], que nous développerons dans la section 3.

D'autres travaux se sont, par ailleurs, consacrés à l'évolution de plantes virtuelles mais qui ont été classés à part car une équivoque existerait quant à la considération des plantes artificielles en qualité de créatures artificielles, du fait qu'elles ne génèrent pas de mouvements car elles sont immobiles [Bor, 2008], [Jac, 1996a], [Tou, 2003].

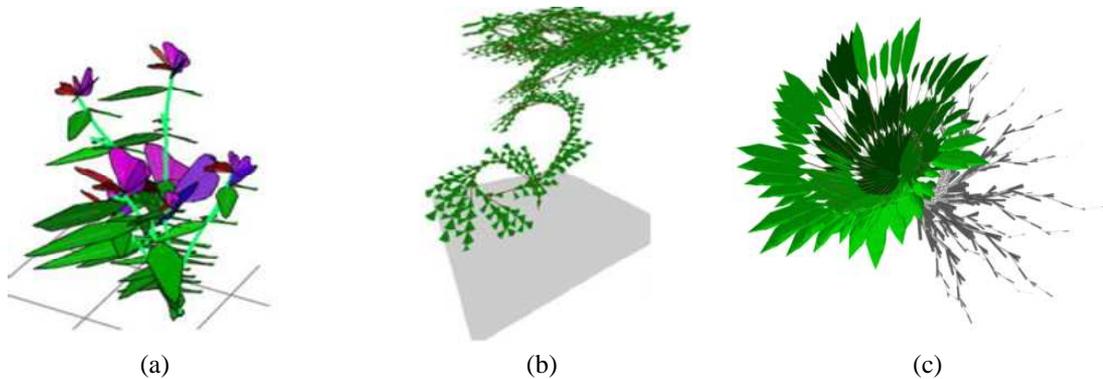


Figure 20: Plante virtuelle. (a) [Jac, 1996]. (b) [Tou, 2003]. (c) [Bor, 2008].

2.2. Evolution de la morphologie et du contrôleur.

2.2.1. Les créatures de Karl Sims.

Une compensation classique, dans le domaine de l'infographie et de l'animation, est celle de la complexité par rapport au contrôle. Il est souvent difficile de construire des entités virtuelles intéressantes ou réalistes et d'assurer et maintenir toujours leur contrôle. Parfois, il est difficile de construire tout un monde virtuel complexe, s'il est nécessaire de concevoir, modéliser, et assembler chaque composant. Un exemple de cette compensation est celui du contrôle cinématique par rapport à la simulation dynamique. Si nous fournissons directement les positions et les angles pour les objets mobiles, nous pouvons contrôler chaque détail de leur comportement, mais il pourrait être difficile de réaliser des mouvements physiquement vraisemblables. Si, autrement, nous fournissons des forces et



des couples et simulons la dynamique résultante, le résultat semblera probablement correct, mais il peut être également très difficile de réaliser le comportement désiré, tout particulièrement car les objets que nous voulons contrôler sont devenus plus complexes. Des méthodes ont été développées pour contrôler dynamiquement, avec succès, des objets spécifiques pour ramper, marcher, ou même courir [McZ, 1990] [Mil, 1988] [RH, 1991], mais un nouvel algorithme de contrôle doit être soigneusement conçu chaque fois qu'un nouveau comportement ou morphologie sont désirés.

Le travail de Sims est lié à ces projets, mais diffère sur plus d'un titre. Dans les travaux précédents, des systèmes de contrôle ont été produits pour les structures fixes qui étaient conçues par l'utilisateur, mais ici des créatures entières peuvent évoluer: l'optimisation détermine les morphologies de la créature aussi bien que son système de contrôle. En outre, dans les travaux de Sims, les corps des créatures sont tridimensionnels possédant des mécanismes physiques. La structure physique tridimensionnelle d'une créature peut s'adapter à son système de contrôle, et vice versa, puisqu'ils évoluent ensemble. Les systèmes nerveux des créatures sont également complètement déterminés par optimisation: le nombre de nœuds internes, la connectivité et le type de fonction que chaque nœud neuronal exécute, est inclus dans la description génétique de chaque créature. Ces nœuds peuvent se développer dans la complexité pendant qu'une évolution se produit. Simultanément, ceux-ci enlèvent la nécessité pour qu'un utilisateur fournisse n'importe quelle information spécifique sur la créature telle que la forme, la taille, les contraintes de joint, les senseurs, les actionneurs, ou les paramètres neuronaux internes. Ainsi, ici un processus de développement est employé pour produire des créatures et leurs systèmes de contrôle, et permet à des composants semblables, y compris leurs circuits neuronaux locaux, d'être définis une fois et répliqués, au lieu d'exiger que cela se fasse séparément. Cette approche est liée aux L-systèmes, aux grammaires graftales, et aux techniques instanciées d'objets [Har, 1992] [Kit, 1990] [Lin, 1968] [MSA, 1989] [Smi, 1984].

Il est opportun d'utiliser les termes biologiques *génotype* et *phénotype* quand on discute évolution artificielle. Un *génotype* est une représentation codée d'une solution possible d'individus ou de problèmes. Dans les systèmes biologiques, un génotype est composé habituellement d'ADN et contient les instructions pour le développement d'un organe. Les algorithmes génétiques emploient typiquement des populations de génotypes qui se composent de séries de nombres binaires ou de paramètres. Ceux-ci sont lus pour produire *les phénotypes* qui sont alors évalués selon quelques critères de fitness et



reproduits selon un processus sélectif. De nouveaux génotypes sont générés en copiant, mutant, et/ou combinant les génotypes des individus les plus adaptés. Pendant que le cycle se répète, la population devrait monter à des niveaux de comportements plus élevés traduits par un haut niveau de la fitness.

Dans les travaux de Sims, l'incorporation du phénotype d'une créature virtuelle est une hiérarchie des parties rigides tridimensionnelles articulées. La représentation génétique de cette morphologie est un graphe dirigé de nœuds et de connections. Chaque graphe contient les instructions de développement pour la croissance d'une créature, et fournit une manière de réutiliser des instructions pour générer des composants semblables ou récursifs dans une créature. Une hiérarchie de phénotypes des parties est faite à partir d'un graphe en commençant à un nœud racine défini et en synthétisant des parties à partir de l'information du nœud tout en traçant par les connections du graphe, ce dernier pouvant être récurrent. Les nœuds peuvent être reliés entre eux en cycles générant des formes récursives ou fractales. Ils peuvent également être reliés à un même fils plusieurs fois permettant, ainsi, de générer plusieurs fois la même forme dans la même annexe. Chaque nœud dans le graphe contient l'information décrivant une partie rigide. *Un type de joint* détermine les contraintes sur le mouvement relatif entre la présente partie et ses parents en définissant le nombre de degrés de liberté du joint et du mouvement permis. Les différents types de joint permis sont: *rigide, révoluté twist, universel, bend-twist, twist-bend, ou sphérique*. Le placement d'une partie d'un enfant par rapport à son parent est décomposé selon la *position, l'orientation, l'échelle* et la *réflexion*. Ainsi, chacun peut être muté indépendamment. La position de l'attachement est forcée d'être sur la surface de la partie du parent. Les réflexions causent des échelles négatives et permettent de générer des sous-arbres semblables mais symétriques. La figure 21 montre quelques topologies simples de graphes et les morphologies résultantes du phénotype.

Pour le contrôleur de la créature, c'est un «cerveau» virtuel qui en détermine le comportement. Le cerveau est un système dynamique qui accepte les valeurs de senseurs d'entrée et fournit les valeurs aux effecteurs de sortie. Les valeurs de sortie sont appliquées comme des forces ou moments de rotation aux degrés de liberté des joints du corps. Ce cycle est repris dans la figure 21. Les senseurs, les effecteurs, et les signaux internes des neurones sont représentés ici par des scalaires continuellement variables qui peuvent être positifs ou négatifs.



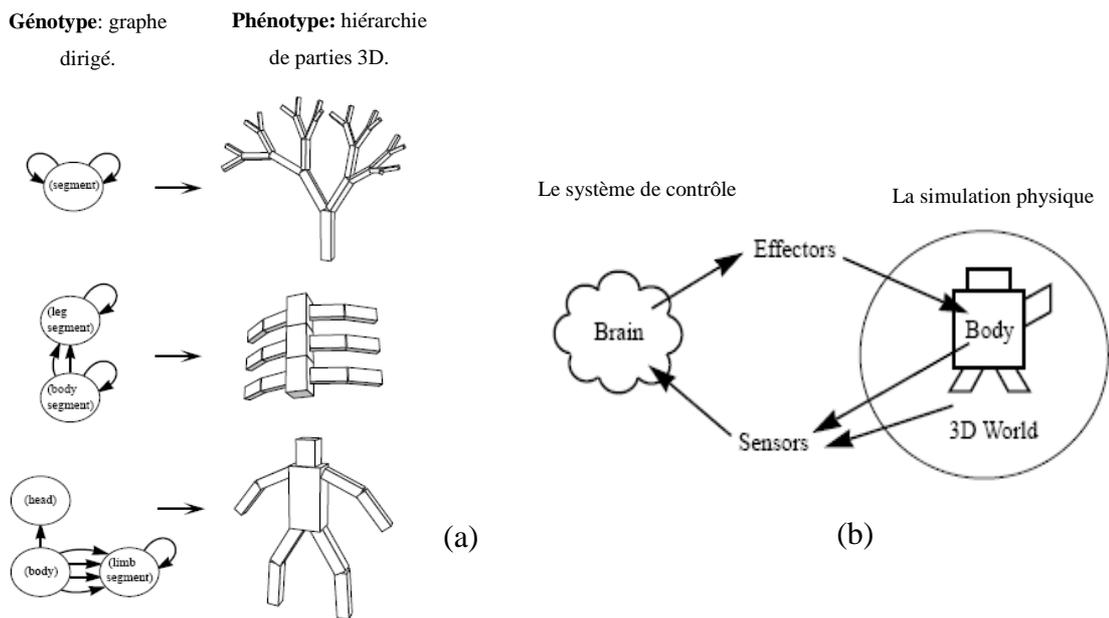


Figure 21: (a) Exemples de graphes de génotype et des morphologies correspondantes de créatures de Karl Sims. (b) Le cycle des effets entre le cerveau, le corps de la créature et l’environnement.

Il y a plusieurs composants de la simulation physique utilisée dans le travail de Karl Sims: dynamique articulée du corps, intégration numérique, détection de collisions, réponse aux collisions, frottements ainsi qu’un effet liquide visqueux facultatif.

Les créatures virtuelles peuvent évoluer grâce à un processus d’optimisation pour une tâche ou un comportement spécifique. Une créature est développée à partir de sa description génétique comme précédemment expliqué, elle est placée ensuite dans un monde virtuel dynamiquement simulé. Le cerveau (ou contrôleur) fournit les forces effectrices qui déplacent des parties de la créature, les senseurs rapportent les aspects du monde et le corps de la créature les transmettent au cerveau, après quoi, le comportement physique résultant de la créature est évalué. Après une durée donnée de temps virtuel, une valeur de fitness est assignée. Celle-ci correspond au niveau de succès de ce comportement. Si une créature possède une fitness élevée relativement au reste de la population, elle sera choisie pour la survie et la reproduction. [Sim, 1994]

Durant la simulation, les créatures sont évaluées pour une tâche spécifique dans l’environnement tridimensionnel. Les résultats obtenus sont les suivants, après plusieurs heures de calcul et plusieurs simulations sur une *Connexion Machine* (environ vingt fois plus puissant qu’un BI-XEON 3Ghz), l’évolution sélectionne différents individus pouvant se mouvoir en rampant, nageant ou en sautant (Figure 22). La créature la plus élaborée



peut suivre une source de lumière en nageant. Les résultats obtenus offrent une grande diversité de créatures ayant des stratégies très différentes.

Dans la seconde partie de ses travaux, il met directement les créatures en confrontation pour essayer de s'approprier un cube. Différentes stratégies intéressantes sont adoptées, notamment certaines éloignent le cube à l'opposé de leur adversaire ou alors tentent d'en garder la possession grâce à un membre articulé. Ces résultats sont spectaculaires, d'autant qu'ils n'ont jamais été reproduit avec un tel réalisme (figure 22).

En perspective de ces travaux, Karl Sims propose d'améliorer ses créatures afin qu'elles interagissent avec des environnements plus complexes. Il énonce aussi la possibilité de construire des robots à partir des créatures les plus évoluées. C'est ce que Lipson et Pollack réalisèrent avec le projet Golem [LP, 2000]. (Voir la section de robots évolutionniste).

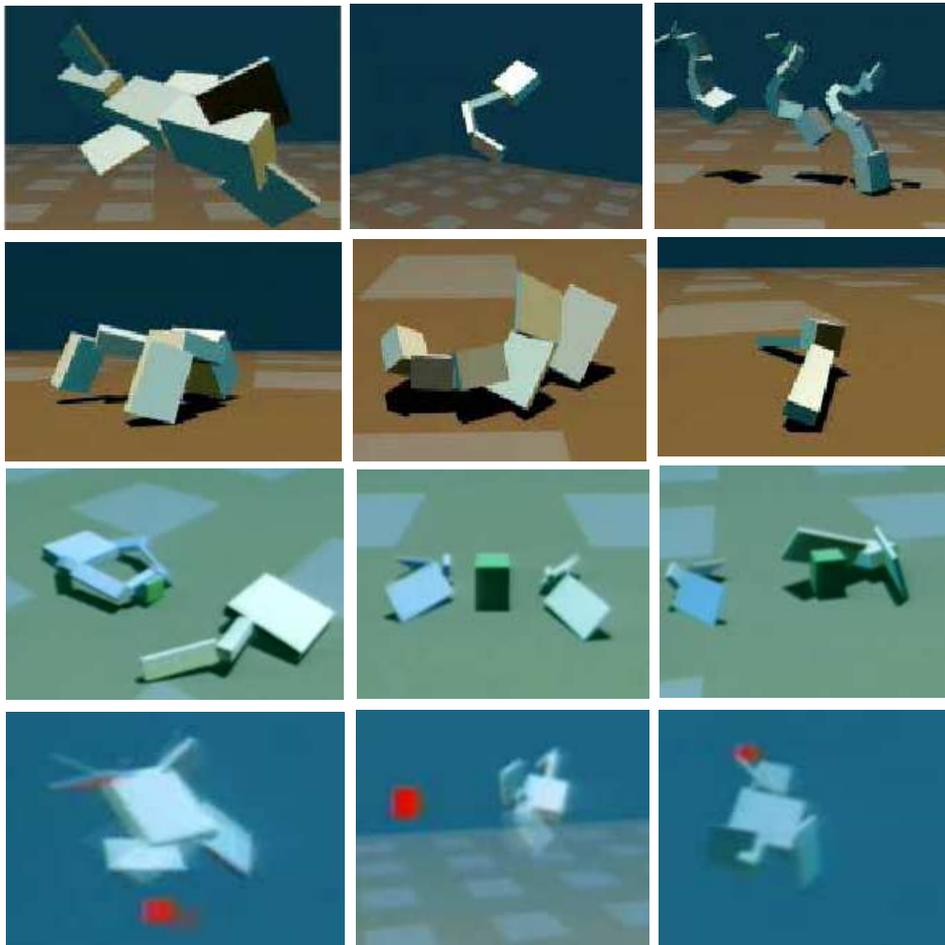


Figure 22: Les créatures évoluées de Karl Sims peuvent sauter, nager, marcher, en compétition ou suivre un point lumineux.



2.2.2. Les créatures de Komosinski.

A la différence de Karl Sims, Komosinski [Kom, 2000; Kom, 2005; AK, 2005; Kom, 2003; KU, 1999; KK, 2001; KKK, 200; KRV, 2001] crée un environnement plus riche dans lequel plusieurs créatures tentent de survivre. Pour cela, elles doivent s'approprier de l'énergie symbolisée sous la forme d'une sphère lumineuse. La morphologie des créatures est composée de tubes (*sticks*), qui peuvent avoir des caractéristiques différentes. Les capteurs sont plus nombreux mais le contrôleur est du même type que ceux de Karl Sims, à savoir des réseaux de neurones composés de fonctions mathématiques élémentaires. Les résultats ne sont pas aussi convaincants mais un langage approprié permet de créer facilement ces propres créatures (figure 23) et permet de les faire évoluer.

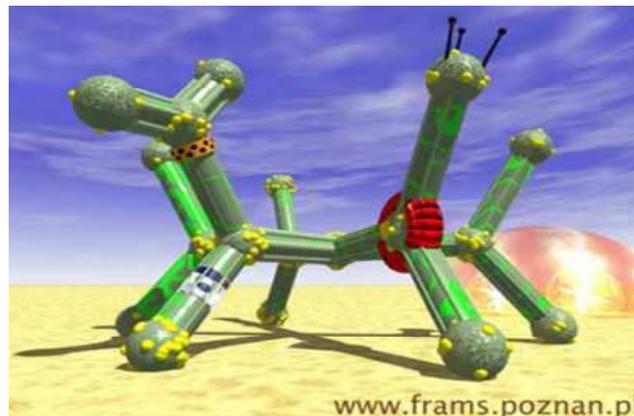


Figure 23: Créature de Framessticks.

2.2.3. Les créatures de Nicolas Lassabe.

Peu de temps après, Nicolas Lassabe et Yves Duthen [LLD, 2008] ont présenté des améliorations au niveau de l'évolution de la morphologie et du comportement complexe des créatures dans un environnement virtuel. L'auteur a proposé d'étudier l'évolution des créatures confrontées à des situations différentes de rampement ou de marche ou à des activités plus complexes comme s'élever ou patiner. Ces créatures utilisent des blocs solides tridimensionnels et une morphologie basée sur les *graphals* de Karl Sims [Sim, 1994b; Sim, 1994c; Sim, 1994a], et un nouveau type de contrôleur basé sur les systèmes de classeurs. Les résultats constituent une nouvelle étape vers des créatures adaptées à des environnements plus complexes.



En ce qui concerne le choix de son environnement, et pour faire évoluer ses créatures, il s'est orienté vers un environnement tridimensionnel simulant les lois de la physique. Son choix s'est donc porté sur *Breve* [Kle, 2003], un environnement basé sur un ODE et réalisé par Jon Klein. Il est dédié aux problèmes de simulations comme les systèmes particuliers, les systèmes multi-agents ou les simulations physiques.

Les figures suivantes, nous présentent les résultats graphiques de Lassabe concernant l'évolution des créatures face à différentes situations tel que ramper, marcher et pour des déplacements plus complexes comme la montée d'escalier, le franchissement de branchées, le déplacement sur un terrain accidenté, la coopération entre deux créatures poussant un bloc et l'interaction avec des objets mobiles.

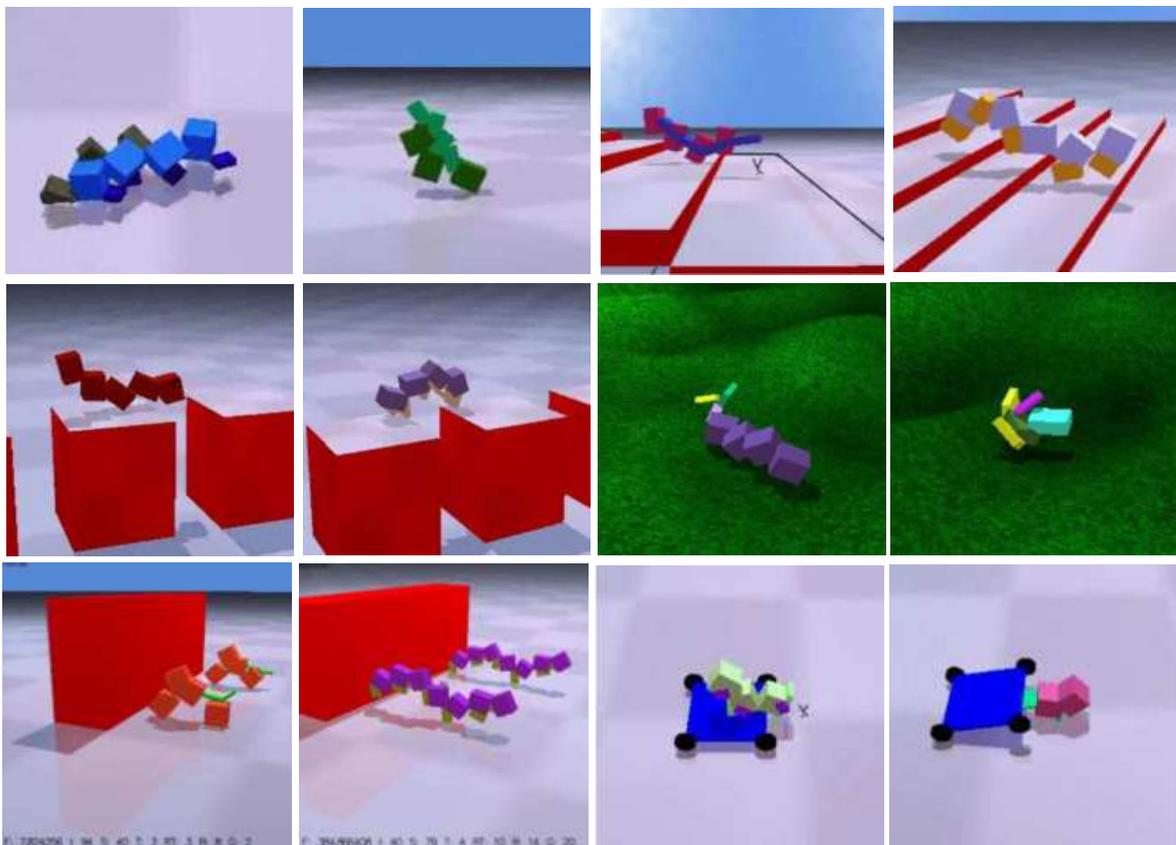


Figure 24: Les créatures de Nicolas Lassabe face à des situations complexes.

2.2.4. Les créatures de Thomas Miconi.

Le système présenté par Thomas Miconi [MC, 2005; MC, 2006a; MC, 2006b; Mic, 2007] est basé sur des créatures autonomes, dans un environnement tridimensionnel physiquement réaliste, semblables à celles présentées par Karl Sims [Sim, 1994c; Sim,



1994b]. Ces créatures sont des structures articulées composées de blocs rigides et commandées par des réseaux de neurones. Les différences principales avec Sims sont, premièrement, l'utilisation des neurones standards de McCulloch-Pitts (au lieu d'un ensemble de règles d'ad hoc, des neurones fonctionnels complexes) et, deuxièmement, un codage génétique amélioré et un système développemental.

Dans la figure 25, on trouve également quelques résultats des expériences faites avec ce système. Au meilleur de notre connaissance, ce travail est la première réplique des efforts de Sims pour réaliser des résultats comparables à ce dernier dans l'efficacité et la complexité. Il a réussi à améliorer les algorithmes de coévolution (figure 26). Miconi a pu, également, réaliser des créatures qui peuvent se combattre [Kom, 2000], [OH, 2004].

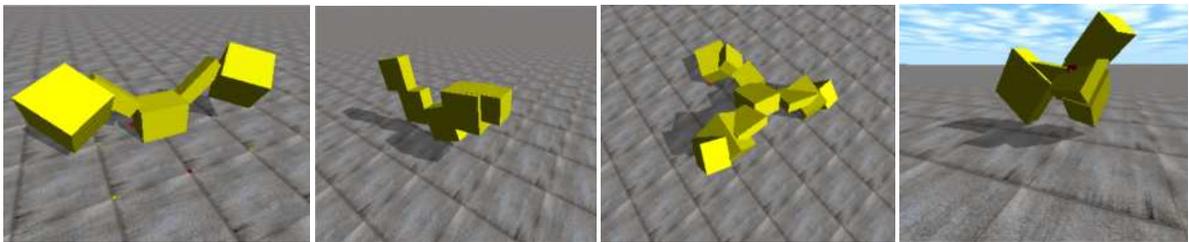


Figure 25 : Les créatures de Thomas Miconi inspiré de celles de Karl Sims

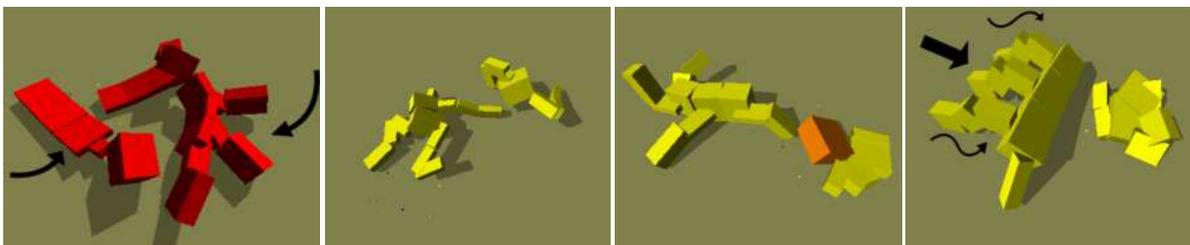


Figure 26: Les Créatures de Miconi qui mènent un combat afin de s'accaparer un cube.

2.2.5. Les créatures de Nicolas Chaumont.

Nicolas Chaumont [CEA, 2007] a également repris les créatures de Karl Sims en utilisant la même morphologie et le même type de contrôleur. Il a proposé, en plus, d'évaluer les créatures sur leur capacité à se déplacer, d'une part, ou à catapulter un bloc, d'autre part. En ce qui concerne les résultats (figure 27), il apparaît trois types de stratégies : certaines créatures lancent le bloc, d'autres le poussent alors que d'autres jonglent avec.



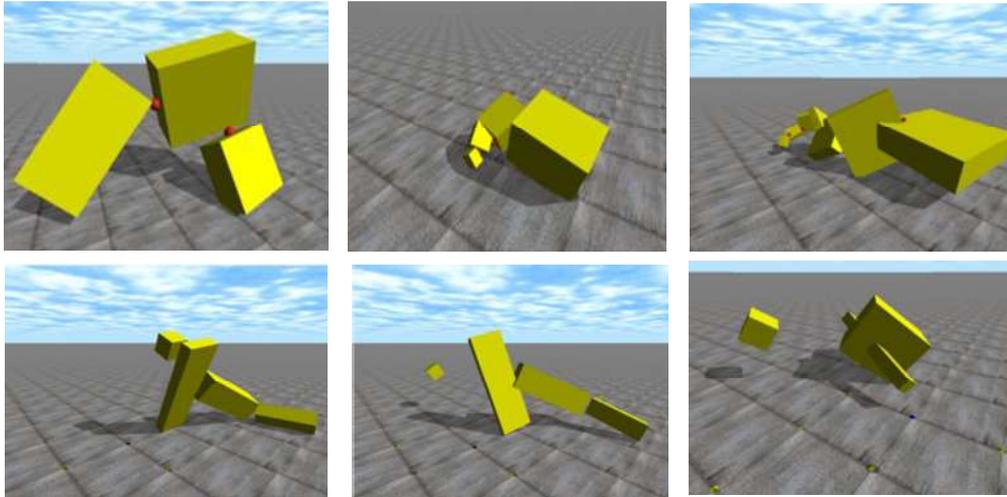


Figure 27: Les créatures de Nicolas Chaumont qui se déplacent et qui catapultent un bloc.

2.3. Evolution du contrôleur avec une morphologie figée.

De nombreux travaux sur l'évolution de contrôleurs pour une morphologie figée existent comme ceux par exemple de LjSpeert, Lee, van de Panne, Terzopoulos [Ljs, 2000]; [LHL 96]; [vdP, 1993]; [TTG, 1994]. En 1990, McKenna et Zeltzer sont parmi les premiers à réaliser un contrôleur pour une entité évoluant dans un environnement physique. Nofli S. and Floreano D. ont travaillé sur les robots à roues [NF, 2002].

2.3.1. Les créatures de Bongard et Lipson.

Les travaux de Bongard et Lipson [BL, 2004a; BL, 2004b et Lip, 2005] portent sur le contrôle de créatures à pattes. Ici, il n'est pas question d'imiter la marche mais de trouver par évolution un contrôleur qui permet à la créature de se déplacer (figure 28).

La créature appartient à un environnement physique réaliste. Elle possède quatre pattes munies de capteurs de contact pour détecter le sol, de capteurs d'angle au niveau des articulations et de deux capteurs de phéromones. Le comportement de la créature est généré par un réseau de neurones. Au commencement de la simulation, les neurones relient les capteurs aux actionneurs. La simulation comporte deux cents individus. Au bout de cinquante générations, le contrôleur produit arrive à contrôler la créature pour qu'elle se déplace en ligne droite ou si on le désire qu'elle suive la piste de phéromones. L'évolution a ainsi produit un contrôleur avec deux fonctions indépendantes.



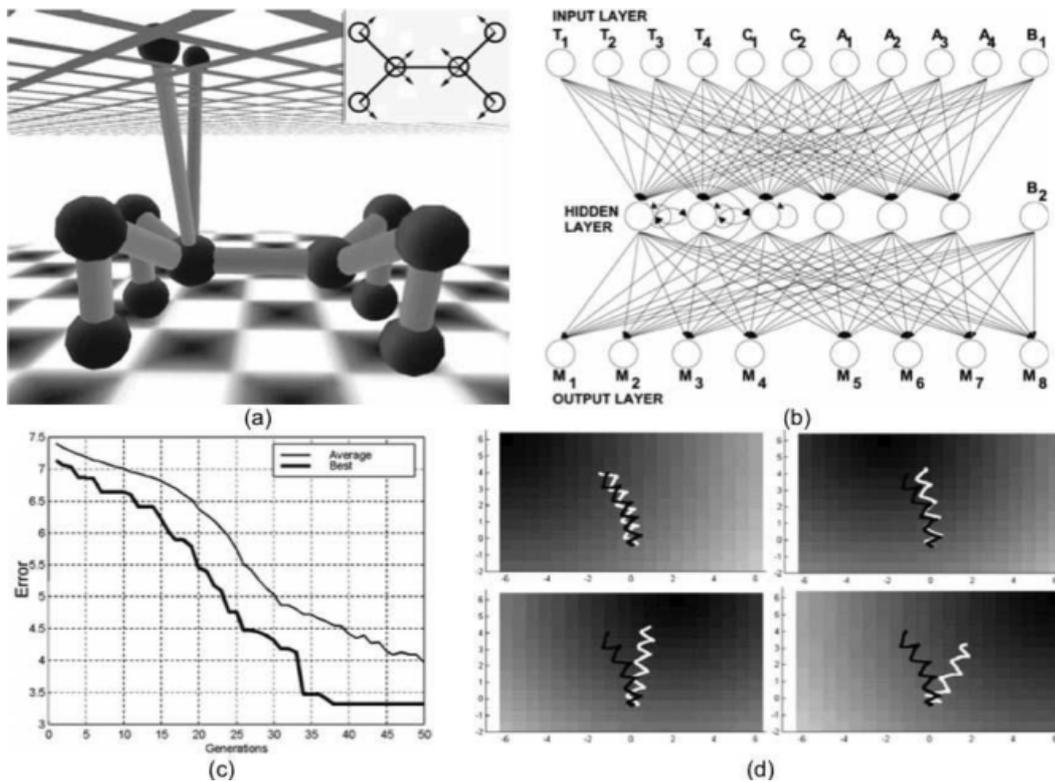


Figure 28: (a) Morphologie de la créature comportant quatre pattes, huit actionneurs et un capteur de contact par patte. (b) Réseau de neurones reliant les capteurs aux actionneurs. (c) La progression de l'erreur à travers les générations. (d) En blanc, le chemin suivant quand les capteurs de phéromones sont captifs et en noir quand ils sont désactivés [Lip, 2005].

2.3.2. Les créatures de Lary gritz.

Lary Gritz et James K. Hahn [GH, 1997] employèrent des techniques évolutionnistes (spécifiquement la programmation génétique) en tant que moyen pour la synthèse du contrôleur dans le cadre de l'animation d'un personnage tridimensionnel.

Des contrôleurs qui conduisent une simulation dynamique du personnage évoluent en utilisant les buts de l'animation comme fonction objective et ayant pour résultat le mouvement physiquement plausible.

Leurs résultats montrent (figure 29) une lampe articulée se déplaçant par saut, le contrôleur le plus évolué permet à la lampe d'éviter les obstacles sur son passage.



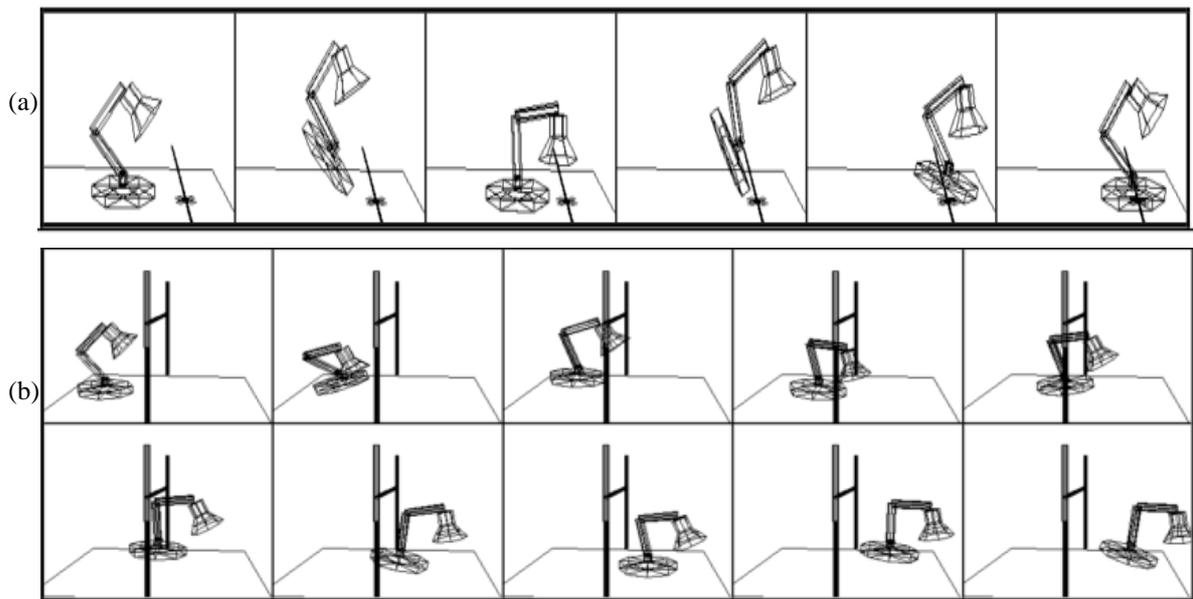


Figure 29: (a) : Le mouvement de saut de la lampe sachant qu'un programme de contrôleur est produit par se mouvement. (b) : La lampe peut apprendre à éviter des obstacles quand des contraintes additionnelles sont ajoutées au système.

2.3.3. Les travaux de Hervé Luga.

Hervé Luga [Lug, 1997] a exposé l'application des techniques de la vie artificielle pour la génération de comportements. Il s'est d'abord intéressé à la génération de contrôleurs simples puis à un assemblage de plusieurs contrôleurs différents pour fournir des comportements plus complexes et notamment coopératifs. L'ensemble de ses simulations ont permis de montrer qu'il est possible de générer automatiquement des comportements simples uniquement à partir de la description du problème et sans l'intervention de l'utilisateur autre que la fourniture des spécifications du problème à résoudre. Ces spécifications interviennent par le choix de la méthode utilisée pour résoudre le problème (algorithmes génétiques, programmation évolutionniste, systèmes de classeurs) et dans le choix d'une fonction de notation apte à rendre compte des différences d'adaptation entre les multiples solutions proposées par les individus de la population.

Un des intérêts majeurs mis en avant par ses travaux est la formidable capacité de ces algorithmes à fournir des comportements proches de ceux observés chez les êtres vivants.



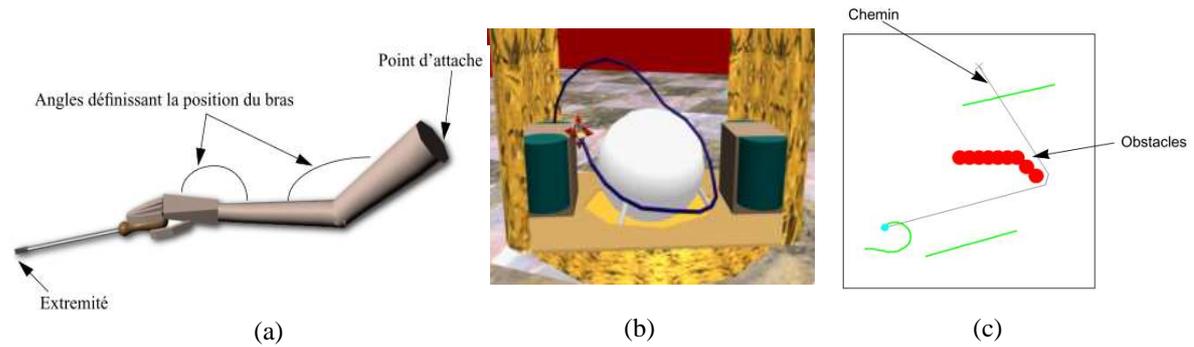


Figure 30: (a) L'utilisation des algorithmes génétiques pour fournir la trajectoire d'un bras articulé qui sera défini par un ensemble de segments de longueur fixe ou variable et des différents angles formés entre ces segments. (b) L'interaction Temps-Réel. (c) Les bras articulés vont donc devoir coopérer afin d'emmener le palet de son point de départ à son objectif.

2.3.4. Autres travaux.

Dave Cliff et Geoffrey Miller [CM, 1995], [CM, 1996] utilisèrent l'évolution pour obtenir des comportements simples de poursuite et de fuite (figure 31. (1) (a) et (b)). Les deux créatures sont contrôlées par des réseaux de neurones récurrents entraînés par une évolution génétique. Un point important de ce travail, consiste dans la notion de coévolution. En effet, les auteurs se sont proposé de faire co-évoluer les contrôleurs d'une proie et d'un prédateur où de faire co-évoluer les deux contrôleurs, ce qui signifie que l'évolution ne concerne pas chaque agent dans son environnement mais les 2 agents, l'un par rapport à l'autre. En définitive, ils ont utilisé les principes suivant : Poursuite/évasion, optimisation du comportement et gestion de l'énergie pour le déplacement.

Durant ses travaux de DEA, David Panzoli [Pan, 2003], a également eu l'occasion de faire évoluer génétiquement des réseaux de neurones pour le contrôle d'agents sur un plan (figures 31. (2). (a)) ou dans l'espace (28. (2). (b)).

De leur côté, Cédric Sanza [San, 2001] puis Olivier Heguy [Heg, 2003] ont utilisé des classifieurs pour produire des agents capables d'élaborer des stratégies dans le cadre de jeux de basket ou de football virtuels (figure 31. (1). (c) et (2). (c)).



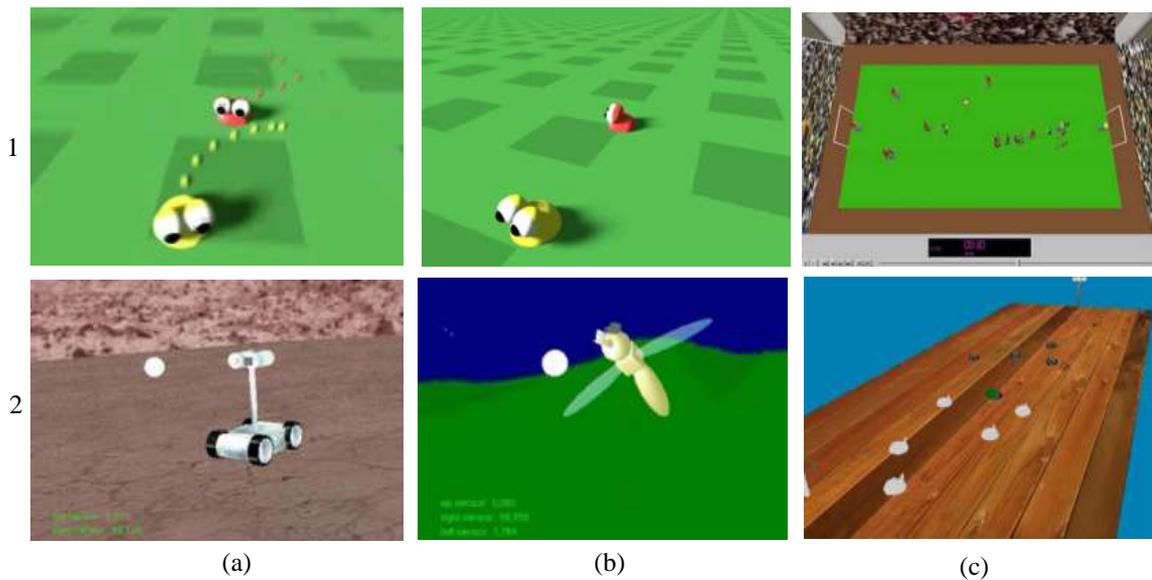


Figure 31: (1). (a) et (b) Cliff et Miller font co-évoluer deux créatures, la jaune doit apprendre à fuir, la rouge à rattraper la jaune, (c) Cédric Sanza. (2). (a) : Panzoli a fait évoluer un agent contrôlé par un réseau de neurones, qui doit être capable de suivre une lumière. (b) : Une autre créature de Panzoli, utilisant le même type de contrôleur mais étendu à des effecteurs différents. (c) Olivier Heguy.

2.4. Evolution du contrôleur avec une morphologie de bipède.

Un robot humanoïde (bipède) est un robot de forme humaine. La plupart des robots de science-fiction que nous voyons sur les écrans sont dotés d'une apparence humaine. C'est pourquoi, pour beaucoup de gens, le robot humanoïde est le robot par défaut. Du point de vue de la robotique, il est difficile d'affirmer qu'un robot créé effectuant les tâches de l'homme doit absolument être de forme humaine. Les avions, par exemple, n'ont pas l'apparence des oiseaux. Plus généralement, les tâches auxquelles un certain robot est destiné déterminent sa forme idéale.

2.4.1. Les bipèdes de Krister Wolff.

Le travail de Krister Wolff et Jimmy Pettersson [WBH+, 2006] concerne une étude sur la marche anthropomorphe abstraite pour un robot bipède simulé qui a été réalisée par le moyen de l'évolution artificielle d'un modèle de réseaux générateurs de patrons (CPGs). L'approche a été étudiée par des simulations dynamiques de rigide-corps 3D d'un robot bipède avec 14 degrés de liberté. Le modèle CPG a été employé comme unité d'oscillateur,



avec des chemins d'intercommunication entre les oscillateurs subissant des modifications structurales en utilisant un algorithme génétique. En outre, l'algorithme génétique a fait évoluer les poids des interconnexions dans le réseau de rétroaction de la structure prédéfinie. En outre, une structure support a été ajoutée au robot afin de guider le processus évolutionnaire vers des démarches humaines. Plus tard, cette structure a été enlevée, et la capacité du meilleur contrôleur à produire une démarche bipède sans aide de la structure support était vérifié. Des modèles stables et normaux de démarche ont été obtenus, avec une vitesse maximum de marche d'environ 0,9 m/s.

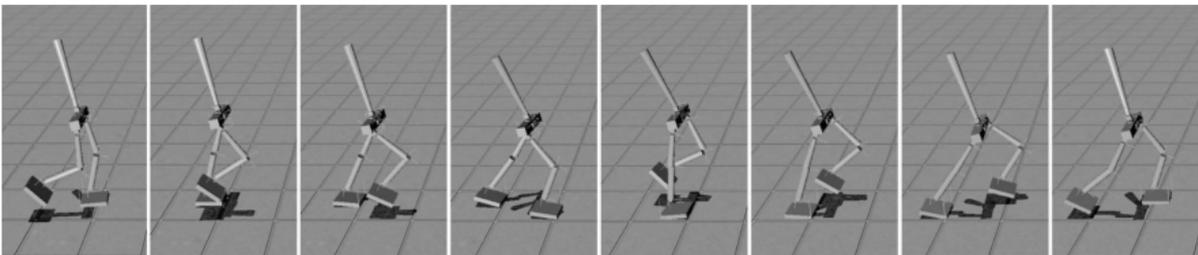


Figure 32 : Les meilleurs résultats obtenus par les bipèdes de Krister Wolff sur un environnement 3D. [WBH+, 2006].

2.4.2. Les humanoïdes de Maurizio Conventi.

Le travail de Maurizio Conventi s'adresse au problème de synthèse et d'animation de créatures bipède 3D dans un environnement simulé physiquement (utilisation du moteur physique OGRE⁷ et AGEIA PhysX), en utilisant des algorithmes génétiques codés.

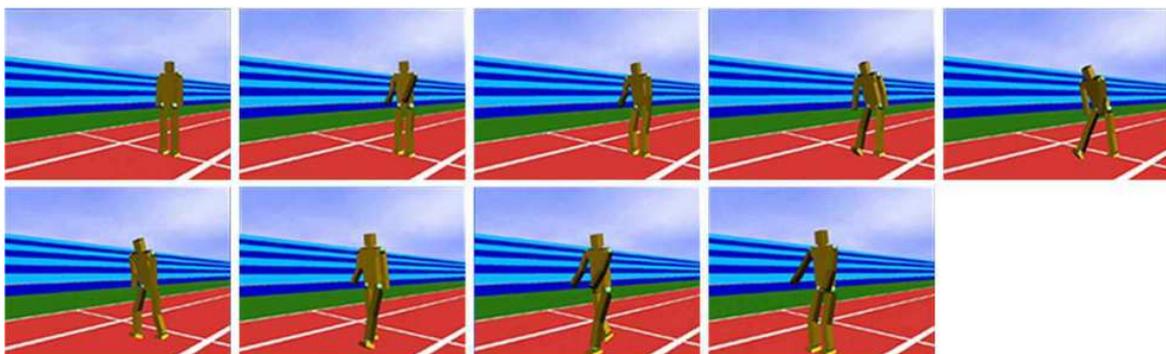


Figure 33: Les humanoïdes de Maurizio Conventi apprennent à marcher sur un espace plat au bout de 200 générations [CF, 2006].

⁷ <http://www.ogre3d.org/>.



Des créatures bipèdes sont représentées par une structure de corps rigides avec des jointures de 1DOF. De telles jointures sont contrôlées par les fonctions sinusoïdales dont les paramètres sont calculés par l'algorithme génétique. Le système qu'il a utilisé tient compte de la synthèse automatique de l'animation physiquement plausible, ainsi il convient aux utilisateurs non habiles. Il a présenté des résultats qui ont été obtenus en examinant et en comparant plusieurs opérateurs génétiques différents.

2.4.3. Les robots humanoïdes de Malachy Eaton.

Le travail de Malachy Eaton décrit une méthodologie, ainsi qu'une série d'expériences associée utilisant cette méthodologie, pour l'évolution du comportement de marche pour un robot humanoïde simulé possédant jusqu'à 20 degrés de liberté. Le travail a permis de faire évoluer les robots à l'effet d'apprendre à marcher en position droite et à démontrer une variété de différents comportements locomotifs. Ce travail diffère des travaux précédents par le nombre de degrés de liberté, jusqu'à 20 au total (bras, coudes, jambes, hanches, genoux, etc.), et démontre l'évolution automatique d'une locomotion entièrement droite pour un robot humanoïde simulé en utilisant le simulateur de robots mobiles qui est Webots⁸ [Mic, 2004].

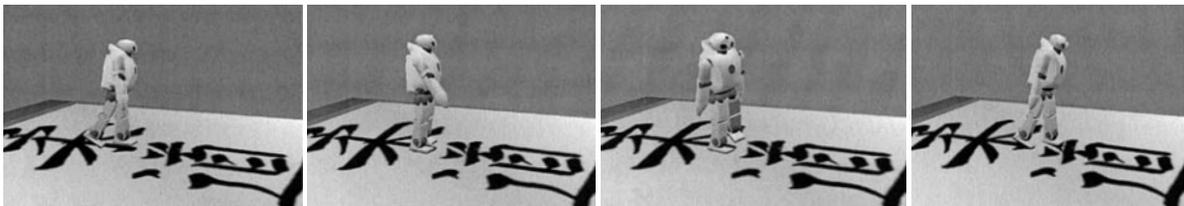


Figure 34 : Les robots humanoïdes de Malachy Eaton [ED, 2006].

3. Embryogénèse artificielle.

L'embryogénèse est certainement l'une des voies les plus prometteuses pour les créatures artificielles. Elle consiste à s'inspirer une fois de plus du vivant pour générer la morphologie, voire le contrôleur d'une créature.

⁸ <http://www.cyberbotics.com>. Commercial Mobile Robot Simulation Software



Le but n'est plus de générer directement une créature, mais de la développer à partir d'une cellule œuf. La créature va ainsi croître grâce à un régulateur de gènes.

Les premiers travaux d'embryogenèse artificielle furent ceux d'Hugo de Garis [dG, 1999]. En utilisant un automate cellulaire, il développa des formes bidimensionnelles. Les règles de cet automate furent déterminées à l'aide d'un algorithme génétique. Le but était de générer des formes ressemblant à des lettres.

Un autre but important de l'embryogenèse artificielle est la simulation des mécanismes de spécialisation cellulaire dont différents travaux existent sur ce sujet. Comme dans la nature, ils utilisent dans la plupart des cas un réseau de régulation de gènes (GRN pour Gene Regulation Network).

Dans la nature, les cellules d'un même organisme ont différentes fonctions, celles-ci étant décrites dans le génome de l'organisme et régulées par un réseau de régulation (GRN) [Dav, 2006]. Les cellules utilisent des signaux de l'environnement extérieur grâce à des récepteurs protéiques. Le GRN, lui aussi décrit dans le génome, utilise ces signaux pour activer ou inhiber la transcription de certains gènes en ARN messenger, le futur modèle de l'ADN de la cellule fille. L'expression de ces gènes dans la cellule fille donnera ses différentes capacités. [SLD, 2008]

Un des premiers travaux intéressants sur la croissance cellulaire est celui de Kurt Fleischer [Fle, 1996]. Plus récemment, René Doursat a proposé une méthode pour générer des structures basées sur l'embryogenèse [Dou, 2007b; Dou, 2006; Dou, 2007a]. Pour cela, il fait évoluer des réseaux régulateurs de gènes qui servent à la représentation des *patterns* et des formes de ces créatures.

3.1. Les créatures d'Arturo Chavoya.

Dans [CD, 2007], Arturo Chavoya a introduit un autre modèle dans lequel le système de régulation de gène est codé au début du génome. Il consiste en une série de sites inhibiteurs ou activateurs ainsi que le codage des différentes protéines de régulation. La production de chaque protéine est conditionnée par les sites d'inhibition et d'activation. La concentration finale de cette protéine détermine l'activation ou non d'une certaine fonction de la cellule. Si la concentration de la protéine est supérieure à un certain seuil, le gène est activé et de ce fait sa fonction correspondante l'est également (figure 35).



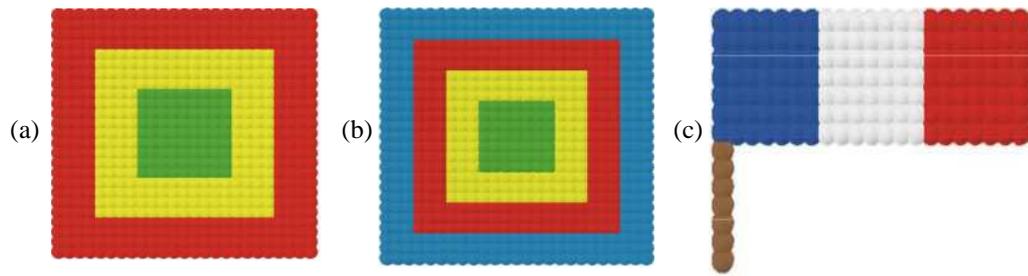


Figure 35: Travaux d'Yves Duthen et Arturo Chavoya [CD, 2007] l'utilisation des AG pour faire évoluer les réseaux de normalisation artificielle (ARNs) pour les modèles colorés désirés : (a) modèle carré de Trois-couleur. (b) modèle carré de Quatre-couleur. (c) un modèle du Drapeau français avec un poteau.

3.2. Les créatures de Sylvain Cussat-Blanc.

Les créatures d'Yves Duthen et Sylvain Cussat-Blanc sont considérées comme les plus récentes. Leur but est de créer un pont entre la morphogenèse artificielle et l'embryogenèse artificielle afin de produire des créatures virtuelles.

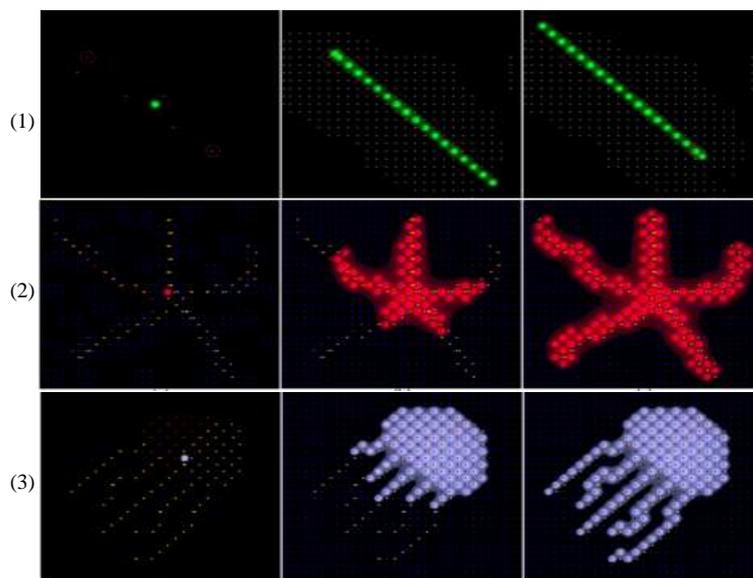


Figure 36: Les travaux d'Yves Duthen et Sylvain Cussat-Blanc [SLD, 2008] système de transfert de substrats artificiel. (1). La créature se développe pour créer la structure du système de transfert et commence le transfert. (2). La croissance de l'étoile de mer. (3). La croissance de la méduse.

Pour ce faire, ils ont fait l'hypothèse que les blocs ou les bâtonnets de l'approche morphogénétique pouvaient être considérés comme des organes, c'est à dire des parties de



la créature possédant une ou plusieurs fonctions. En utilisant des techniques de développement cellulaire pour faire croître la créature, ils ont voulu créer ces organes en partant d'une cellule unique. Pour ce faire, la cellule devra être capable de se spécialiser pour s'adapter à son environnement.

L'organisation des cellules en tissus (c'est à dire en groupes de cellules qui ont la même fonction) puis l'organisation des tissus permettra la création d'organes. Après avoir créé une bibliothèque d'organes, un mécanisme évolutionniste assemblera des organes afin de faire émerger une créature adaptée à son environnement [SLD, 2008]. Ses résultats sont présentés dans la figure 36.

4. La robotique évolutionniste.

A partir de la définition des créatures artificielles et des propriétés minimales de ces créatures qui sont décrites dans la section 1, on peut en déduire que les robots sont des créatures artificielles⁹.

La robotique évolutionnaire est un sous domaine du domaine de la robotique que dans lequel les algorithmes évolutionnaires sont utilisés pour la construction¹⁰ de robots [Weh, 2005], [NoI, 2000].

4.1. Le Projet Golem de Hod Lipson.

Les travaux de Lipson et Pollack [LP, 2000], [PHLF, 2003] sont certainement les plus finalisés par le fait que leurs créatures ont été par la suite effectivement construites (figure 37-d). Ce sont les seules à avoir suivi tout le processus d'évolution jusqu'à leur réalisation physique. Les composants virtuels doivent donc avoir la possibilité d'être reproduits sous forme physique. Pour cela, les créatures sont composées essentiellement d'éléments dit 1D de tubes et de pistons (figure 37). Elles ne comportent pas de capteurs. Les créatures peuvent se mouvoir en compressant leurs pistons. Les joints entre les pistons peuvent être rigides ou permettre un certain degré de liberté. Les actionneurs sont contrôlés par un réseau de neurones produisant des fonctions sinusoïdales et permettant des topologies récurrentes. L'évolution permet de modifier la morphologie des créatures ainsi que leurs contrôleurs.

⁹ En effet les robots sont une unité car ils peuvent avoir une indépendance énergétique par le biais d'une batterie. Ils interagissent physiquement avec l'environnement, ils peuvent être autonomes.

¹⁰ La construction peut évoluer le cerveau du robot, son corps, ou bien les deux.



Au début de la simulation, la population est de deux cents individus sans tubes et sans neurones. Les mutations sont alors la seule issue pour ajouter de nouveaux éléments aux créatures. La fonction de *fitness* détermine leur habilité à se déplacer en mesurant la distance parcourue par les créatures à partir de leur centre de gravité. La sélection permet aux créatures ayant le plus de potentiel de se reproduire. Ce processus est reconduit de trois cents à six cents fois. Après plusieurs dizaines de générations, les premiers mouvements de créatures apparaissent. Les patterns des morphologies et des neurones permettant de se mouvoir sont ensuite répandus dans la population par le processus de sélection et de croisement. L'évolution des populations de leur génotype fait apparaître des diversifications et des extinctions (figure 37-b).

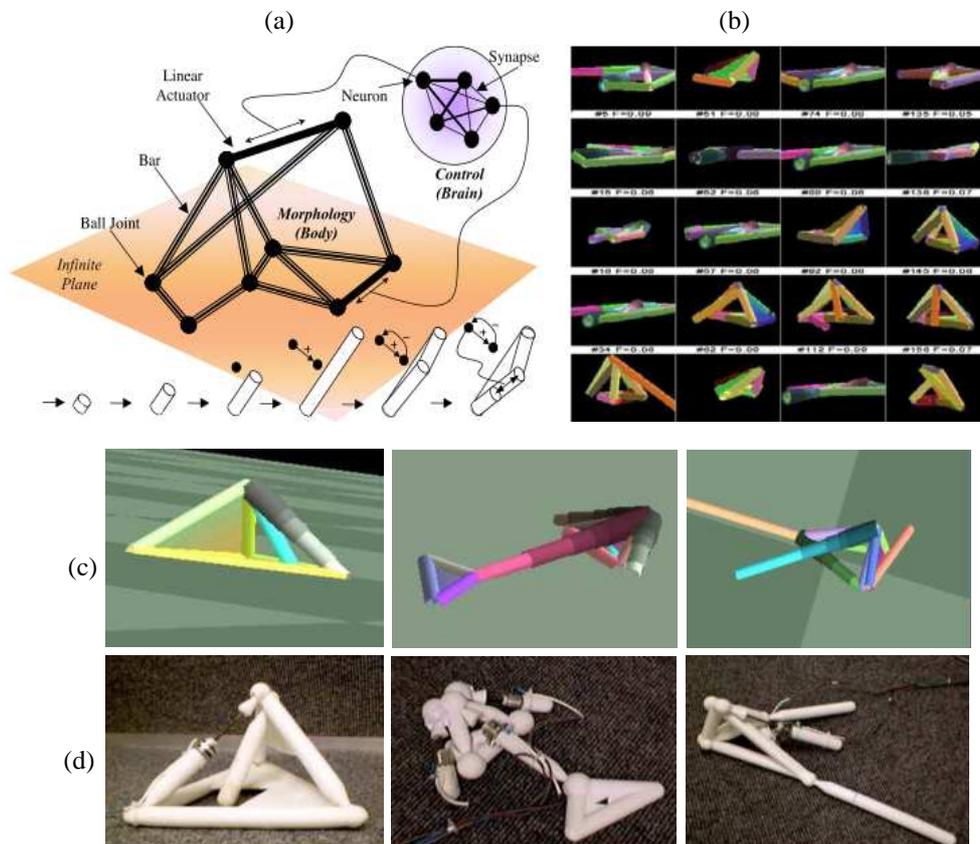


Figure 37: (a) Schéma d'illustration d'un robot évolutif. (b) Des instances différentes de la génération entière. (c) Trois évolutions des robots, dans la simulation (d) les trois robots reproduits en réalité physique en utilisant un prototype rapide [LP, 2000].

Les meilleures créatures à la fin de la simulation sont réalisées physiquement par une imprimante tridimensionnelle. Ces robots comportent généralement une vingtaine de pièces. Elles sont pour la plus part symétriques bien que cela n'ait été spécifié dans le



programme. Cela s'explique par le fait que les créatures symétriques sont plus performantes pour parcourir une longue distance. De plus, les créatures présentent une grande robustesse du fait que l'on change les tailles de leurs tubes.

4.2. Les robots de Gregory S. Hornby.

Gregory S. Hornby décrit un système qui utilise les l-systèmes comme un codage des algorithmes évolutionnistes (EA) pour la création de créatures artificielles. Les créatures générées par ce système ont des milliers de segments (Parties) et l'utilisation des l-systèmes donne des résultats de créatures avec un aspect visuel plus naturel. Pour faire évoluer les créatures mobiles, il s'est proposé d'utiliser la fonction de fitness représentée par une fonction de distance parcourue par le centre de gravité de la créature [HLP, 2003], le contrôleur utilisant un réseau de neurones. Le moteur physique a été réalisé pour obtenir de bonnes performances sur ce type de structures.

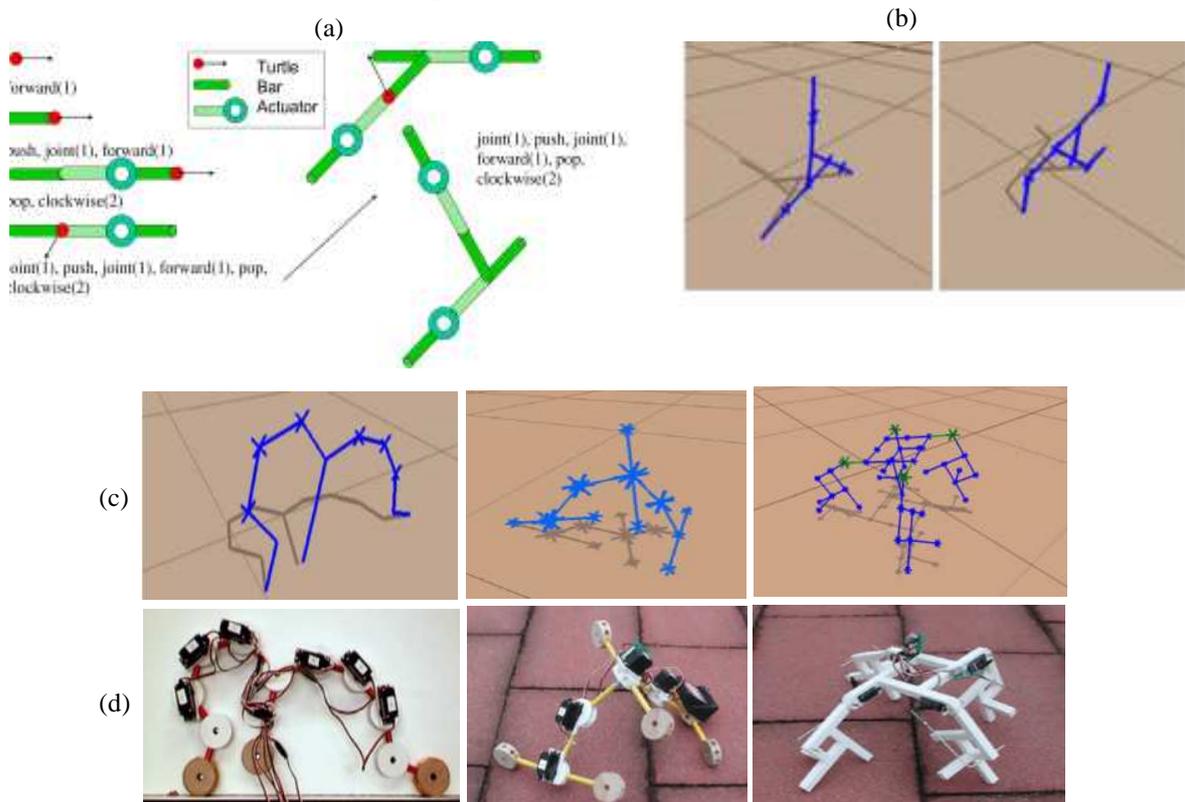


Figure 38: (a) Schéma d'illustration de la séquence de construction. (b) Le robot résultant avec des joints actionnés. (c) Trois exemples de robots produits par l'évolution des l-systèmes. (d) Instanciation physique des exemples précédents. [HLP, 2003]



Les simulations durent autour de 250 générations et les créatures ainsi obtenues sont d'une grande variété. De plus, les créatures ont pu être physiquement réalisées. Plus récemment, Hornby s'inspira de ces créatures pour faire évoluer des structures statiques comme des antennes pour les satellites. Ses résultats sont présentés dans la figure 38.

4.3. Les robots de Victor Zykov 1.

Un processus évolutif peut exiger un changement de la morphologie, ou plus encore, la production d'une nouvelle morphologie physique. Une approche pour la génération de nouvelles morphologies est d'utiliser les robots reconfigurables [YZD, 2002]. Des robots reconfigurables se composent de plusieurs modules qui peuvent être reliés, débranchés et réarrangés dans diverses topologies pour créer des machines avec des plans variables de corps. Les robots auto-reconfigurables peuvent réarranger leur propre morphologie, et s'adapter ainsi à la réalité physique. La figure 39 montre un exemple d'un robot auto-reconfigurable composé de huit cubes identiques [MMD, 2004]. Chaque cube peut pivoter autour de son axe, et se relie et se débranche à d'autres cubes en utilisant des électro-aimants sur ses faces [ML, 2004].

Bien que ce robot contienne seulement 8 unités, il est imaginable que de futures machines puissent être composées de centaines et de milliers de modules de plus en plus petits, permettant un contrôle et une flexibilité beaucoup plus grands dans le changement morphologique.

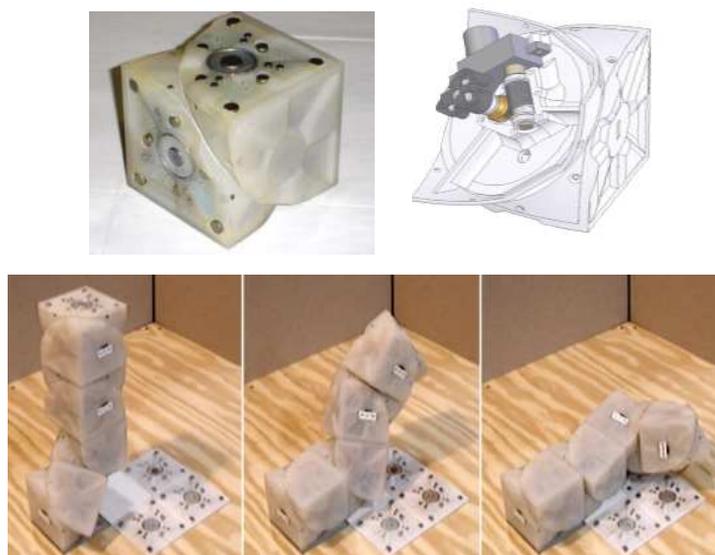


Figure 39: Changements morphologiques dans les robots de molécule dans la réalité [ML, 2004]



4.4. Les robots de Victor Zykov 2.

Les animaux soutiennent la capacité de fonctionner après des dommages en créant des comportements compensatoires qualitativement différents. Bien qu'une telle robustesse soit souhaitable pour des créatures artificielles, la plupart des machines échouent face aux dommages inattendus.

Victor Zykov [BZL, 2006] a décrit un robot qui peut récupérer de tels changements de façon autonome, par un auto-modeleur continu. Un robot à quatre jambes utilise le rapport action-sensation pour impliquer indirectement sa propre structure, il utilise alors ce modèle d'individu pour reproduire la locomotion vers l'avant (ou vers l'arrière). Quand une partie d'une jambe est enlevée, elle adapte les modèles d'individu, menant à la génération de démarches alternatives. Ce concept peut aider à développer des machines plus robustes.

La figure 40 montre le fonctionnement de ses robots.

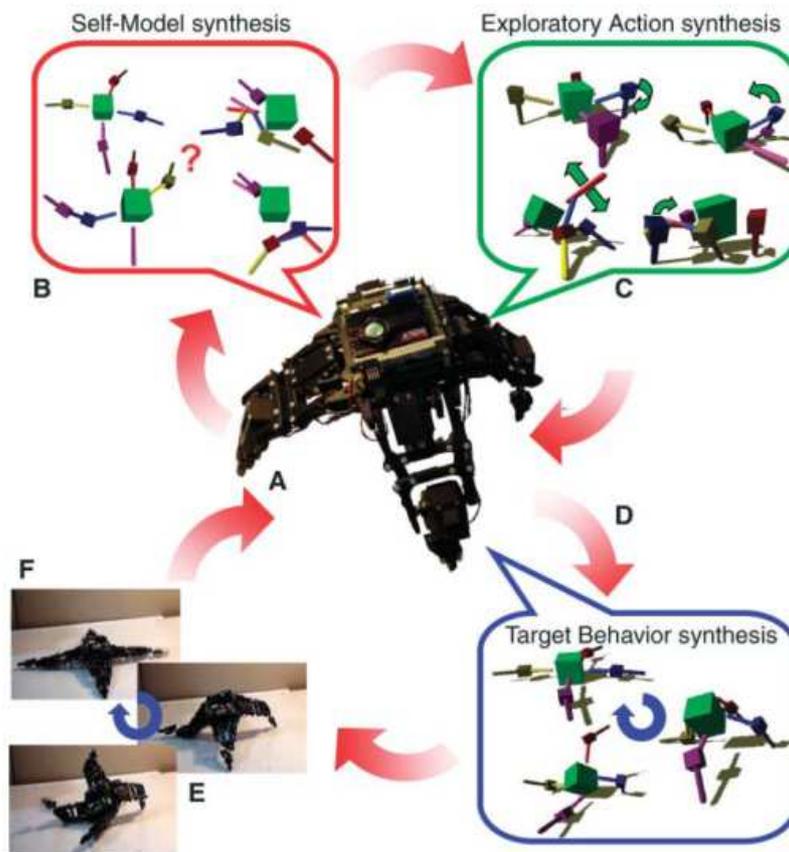


Figure 40: Robot de Victor Zykov 2.

Le robot fait un cycle continu à travers l'exécution d'actions. (A et B) synthèse du modèle d'individu. Le robot effectue physiquement une action (A). Au début, cette action est



aléatoire; plus tard, c'est la meilleure action trouvée en (C). Le robot produit alors plusieurs modèles d'individu pour assortir les données des senseurs rassemblées tout en effectuant les actions précédentes (B). Il ne connaît pas quel modèle est correct. (C) Synthèse d'action exploratoire. Le robot produit plusieurs actions possibles qui désambigüisent les modèles d'individu en concurrence. (D) Synthèse de comportement de cible. Après plusieurs cycles de (A) à (C), le modèle actuellement le meilleur est employé pour produire des ordres de locomotion par optimisation. (E) Le meilleur ordre de locomotion est exécuté par le dispositif physique. (F) Le cycle continue à l'étape (B) à raffiner plus loin des modèles ou à l'étape (D) pour créer de nouveaux comportements. [BZL, 2006]

5. Synthèse.

Dans cette partie, nous allons faire une synthèse des travaux présentés dans ce chapitre en mentionnant leurs catégories communes et les points forts de chacun ainsi que les nouveautés introduites par chaque type de créature (Tableau 2).

6. Conclusion.

En conclusion, nous avons, à l'occasion du présent chapitre, passé en revue l'essentiel des travaux dédiés à la simulation de créatures artificielles selon la forme et la nature du mouvement à reproduire. Le point commun entre tous ces travaux consiste en l'utilisation du terme évolution (les techniques évolutionnistes). Dans le prochain chapitre nous faisons une représentation détaillée sur notre contribution dans le domaine.



	Auteurs	Référence	Créature	Actions	Contrôleur	Morphologie	Points forts
Morphogénèse artificielle	<i>Karl Sims</i>	[Sims, 94]	Corps articulé (blocs)	Sauter, marcher, nager, suivre. Coévolution.	Réseaux de neurones avec des fonctions mathématiques	Parallélépipède des rectangles (Graphtals)	Le meilleur réalisme
	<i>Komosinski</i>	[Kom, 05]	Les Framssticks	Survivre	Pareil que Sims	Sticks	Un langage personnel, environnement
	<i>Nicolas Lassabe</i>	[LLD, 08]	Pareil que Sims	Marcher, ramper, monter un escalier, interagir avec des objets mobiles et coopérer.	Contrôleur basé sur le système de classeurs	Graphtals de Sims	Environnement complexe
	<i>Thomas Miconi</i>	[Mic, 07]	Pareil que Sims	Ramper, combattre. Coévolution.	Réseaux de neurones de type McCulloch-Pitts	Graphtals de Sims	Amélioration de la coévolution
	<i>Nicolas Chaumont</i>	[CEA, 07]	Pareil que Sims	Se déplacer et catapulte un bloc	Pareil que Sims	Graphtals de Sims	Catapultes
	<i>Josh Bongard</i>	[BL, 04]	Créatures à quatre pattes,	Déplacement	Réseaux de neurones avec deux fonctions différentes	Figée	Evolution du contrôleur
	<i>Lary Gritz</i>	[GH, 97]	Lampe	Sauter, Eviter des obstacles	Programmation génétique pour faire évoluer le contrôleur	Figée	Génération de programme avec des mouvements réalistes
	<i>Hervé Luga</i>	[Lug, 97]	Bras articulés	Recherche de trajectoire, coopération	Réseaux de neurones, systèmes de classeurs	Figée	Comportements proches de ceux des êtres vivants
	<i>Krister Wolff</i>	[WBH+, 2006]	Bipède	Locomotion	réseaux générateurs de paternes (CPGs)	Figée	Une meilleure locomotion
	<i>Maurizio Conventi</i>	[CF, 2006]	Humanoïde	Locomotion et lancement de balle	Des fonctions sinusoïdales	Figée	Comportement de lancement d'une balle
	<i>Malachy Eaton</i>	[ED, 06]	Humanoïde	Locomotion	Un contrôleur créé et modifié par C, C++ ou JAVA	Figée	Humanoïde avec 20DOF
Robotique évolutionnaire	<i>Hod Lipson</i>	[LP, 00]	Blocs construits à partir de tubes et de pistons	Ramper	Réseaux de neurones avec fonctions mathématiques	Tubes et pistons	Fabrication des robots à partir d'une évolution des créatures artificielles (imprimante 3D)
	<i>Gregory Hornby</i>	[HLP, 03]	Créature faite à partir de milliers de segments	Ramper	Réseaux de neurones avec fonctions mathématiques	L-Système	Robots à morphologie modulaire
	<i>Victor Zykov1</i>	[ML, 04]	Robot molécube	Changer la morphologie	Modulaire (contrôle de chaque cube)	Robots reconfigurables	Changement physique de la morphologie
	<i>Victor Zykov2</i>	[BZL, 06]	Robot à quatre pattes	Locomotion	Le rapport action-sensation	Conception d'un robot à partir d'une créature virtuelle	Robot adaptatif aux changements
Embryogénèse	<i>Arturo Chavoya</i>	[CD, 07]	Modèle coloré, triangles et cercles	Croissance d'une forme	les réseaux de normalisation artificielle (ARNs)	Cellule œuf	Modèle d'embryogénèse
	<i>Sylvain Cussat-Blanc</i>	[SLD, 08]	Etoile de mer et méduse	Croissance d'une étoile de mer et d'une méduse	Réseau de régulation de Gène	Cellule œuf	Utilisation du parallélisme

Tableau 2: Synthèse sur les travaux de créatures artificielles.



Chapitre III : Le Modèle Proposé

Le modèle proposé

Plusieurs aspects de la vie moderne impliquent l'utilisation de machines intelligentes capables de fonctionner sous l'interaction dynamique de leurs environnements. Pour cette raison, le domaine de la locomotion d'un bipède est d'intérêt spécial quand des créatures humanoïdes sont concernées. Actuellement, la recherche sur des créatures ou même des robots humanoïdes et bipèdes en locomotion constitue un des axes les plus passionnants dans le domaine de la robotique. On peut signaler plus de 50 projets s'intéressant aux robots humanoïdes de part le monde, sans oublier ceux dédiés exclusivement aux techniques de la marche des bipèdes¹¹.

D'un point de vue théorique, la réalisation du contrôle de la marche des robots bipèdes est plus difficile à traiter que les robots classiques. En effet, les robots classiques sont conçus pour maintenir leurs roues en contact permanent avec la surface de déplacement. La question fondamentale de la locomotion des bipèdes, indépendamment de la réalisation des mouvements relatifs aux liens du mécanisme, est de déterminer de quelle manière on peut préserver l'équilibre de marche dans le système [VBS, 2001]. Cette question a longtemps été la matière principale de beaucoup d'études scientifiques. C'est également le cas en ce qui concerne notre présente étude. Cette dernière est consacrée à l'évolution d'un contrôleur pour une créature appartenant à un environnement complexe, ce qui n'est pas le cas avec la majorité des travaux précédents.

Dans ce chapitre nous allons expliquer en détail la solution que nous proposons pour prendre en charge le problème de locomotion d'un humanoïde. Ainsi, nous allons décrire les différentes techniques de représentation de la morphologie de la créature simulée ainsi que le type de contrôleur choisi.

¹¹ Une liste étendue de projets est donnée sur le site www.androidworld.com.

1. Environnements.

a. Les écosystèmes.

A la différence des travaux d'évolution de créatures artificielles dans des environnements physiques réalistes, qui se focalisent sur des créatures dont une seule est généralement présente effectivement à la fois dans l'environnement, les écosystèmes font évoluer des créatures artificielles dans un environnement bidimensionnel ou tridimensionnel. Les écosystèmes artificiels s'intéressent aux stratégies et aux adaptations qui émergent de l'interaction d'une multitude de créatures. Dans les deux chapitres précédents nous n'avons pas traité ces derniers. Nous allons, par contre, à ce niveau donner quelques références sur les fameux travaux s'intéressant aux écosystèmes.

Nous nous sommes d'abord intéressés au projet **Tierra** développé par Thomas Ray tout au long de la décennie (1991-2001) [Ray, 1991a ; Ray, 1991b]. Dans ce projet, Ray a réalisé un écosystème dont l'environnement est la mémoire de l'ordinateur et dont les individus sont des programmes. Les programmes se disputent les ressources du microprocesseur et de la mémoire afin de survivre. Plusieurs autres projets se sont inspirés des travaux de Tierra malgré que les créatures utilisées (des organismes digitaux) n'aient, certainement pas, la possibilité d'acquérir les propriétés du vivant.

Nous avons également étudié le projet **Gene Pool** réalisé par Jeffrey Ventrella. Ce dernier a produit des créatures sous une sorte de microscope dans un environnement bidimensionnel simulant un laboratoire [Ven, 1994 ; Ven, 1996 ; Ven, 1998a ; Ven, 1998b ; Ven, 2005]. Ces créatures sont les *Swimbots*, l'environnement de Gene Pool est physique et simple, permettant de faire évoluer un grand nombre de créatures simultanément. Chaque créature (*Swimbot*) possède une quantité d'énergie qui diminue lors de ses déplacements et de sa reproduction.

Enfin, Metivier a proposé **Life Drop** [MLH, 2002], un écosystème dont l'environnement prend place dans une goutte d'eau artificielle et dont les créatures ont reçu la dénomination de *Biomorphs*.

Les travaux sur les agents autonomes se sont concentrés sur des environnements bidimensionnels négligeant, pour des raisons de complexité, la prise en charge d'environnements tridimensionnels.



b. Les environnements tridimensionnels.

Dans la prise en charge de la simulation de l'évolution de créatures artificielles tridimensionnelles, une gamme de travaux a été présentée dans le deuxième chapitre à l'occasion de laquelle nous nous sommes intéressés à analyser les mécanismes permettant de faire évoluer des créatures dans un environnement tridimensionnel. Tout ceci dans le but de munir ces créatures des différents mécanismes leur permettant de s'adapter à l'environnement dans lequel elles vivent et avec lequel elles interagissent. Selon plusieurs travaux et avec différents types d'environnements, nous avons pu conclure que les mouvements générés par évolution dépendent de l'environnement. Nous pouvons, ainsi citer, et à titre indicatif :

1- Les créatures de Karl Sims évoluant dans un espace plat où il fait co-évoluer la morphologie et le contrôleur afin de réaliser des mouvements de déplacement (*locomotion*) [Sim, 1994a ; Sim, 1994b ; Sim, 1994c]. De plus, ces créatures peuvent, également, simuler la nage dans l'eau. Les créatures de Shim évoluent pour adapter leurs ailes au *vol* [SK, 2007].

2- Les créatures de Nicolas Chaumont qui évoluent sur un espace plat et cela grâce à leur capacité de se déplacer (*locomotion*) ainsi qu'à leur capacité de *catapulter* des blocs. [CEA, 2007]

3- La fameuse lampe de Lary Gritz qui est, quant à elle, capable d'*éviter un obstacle* présent dans son environnement. [GH, 1997]

4- Les créatures développées par Nicolas Lassabe, qui évoluent dans un environnement complexe, ont été munies de capacités leur permettant de générer de nouveaux mouvements tel que la *montée d'escalier*, le *franchissement de tranchées*, le *déplacement sur un terrain accidenté* et le *déplacement en utilisant un objet mobile* [LLD, 2008]. Tous ces mouvements, qui sont différents et qui représentent beaucoup plus qu'une simple locomotion, sont réalisés sous la contrainte de l'environnement. Ceci signifie que le choix des mouvements et du type de créature dépend de la nature de l'environnement. La dualité se trouve également vérifiée, puisque nous pouvons considérer que pour une créature donnée nous aurons besoin de tels environnements.



c. L'environnement choisi.

Pour poursuivre notre simulation et pour accomplir notre but qui est l'évolution d'un contrôleur assurant le déplacement d'une créature tridimensionnelle, et selon la conclusion faite dans la section précédente, nous avons choisi un environnement physique 3D qui est un espace plat avec quelques obstacles (objets). Il y'a lieu de signaler, à ce niveau, que bien que notre créature soit en fait tridimensionnelle, notre but n'est pas de lui trouver un écosystème mais plutôt de la faire évoluer d'une manière autonome dans un environnement tridimensionnel. De ce fait, et partant de cette hypothèse, notre simulation nécessitera la disponibilité d'un environnement physique réaliste (3D) car c'est le plus à même de nous permettre de réaliser un monde virtuel proche du monde réel qui, de plus, devra permettre à notre créature, qui est en fait un humanoïde, de réaliser des mouvements réalistes proches des mouvements de l'humain.

Il y'a lieu, enfin de conclure que dès lors que l'étape du choix de l'environnement est réalisée, il sera question de trouver le moyen ou bien l'outil qui assure la simulation de toutes les caractéristiques nécessaires pour cet environnement, c'est le moteur physique.

2. Les moteurs physiques.

Plusieurs travaux cherchent à faire évoluer des créatures dans des environnements physiques réalistes en leur faisant reproduire des mouvements réalistes également. De ce fait, le seul moyen qui leur permet de garantir ce réalisme réside dans la simulation et ce en reproduisant les lois de la physique. Cette approche n'est pas nouvelle, du fait que Zeltzer [MZ, 1990] a fait une simulation dynamique (physique) pour l'évolution d'un bipède pour la locomotion. D'autres travaux s'y sont également intéressés [Ryn, 1987 ; TT, 1994 ; Tv, 2003] et à l'occasion desquels les chercheurs ont construit leur propre simulateur physique malgré que ces moteurs (simulateurs) physiques n'aient pas eu vraiment la qualité recherchée car ils étaient trop spécialisés. Le développement d'un modèle d'évolution de créatures artificielles dans un environnement 3D incluant la physique était alors un travail relativement complexe à réaliser. Actuellement et depuis les années 2000, nous disposons des outils (moteurs physiques libre) et du matériel (puissance des ordinateurs) nécessaires pour une bonne simulation de créatures artificielles quelque soit la forme simulée ainsi que les mouvements qu'on voudra lui faire reproduire.



Parmi les moteurs physiques qui sont à l'usage public (ou libre) nous pouvons citer, *Breve*¹², ODE¹³, Newton Dynamics, OGRE et bien d'autres, ceux là étant les plus répandus.

A titre indicatif, *Breve* développé en langage C, qui est un environnement basé sur ODE et réalisé par Jon Klein [Kle, 2003], est un environnement dédié principalement aux simulations de la vie artificielle. Pour simuler des modèles avec *Breve*, nous disposons de deux types de classes: *Steve* et *Python*.

Notre choix était en premier lieu retourné vers le moteur *Breve*. Cependant la réalisation des interfaces avec *Breve* représentait un travail fastidieux. Par ailleurs, *Breve* ne permettait pas d'implémenter tous les types de données, ainsi et à titre indicatif, pour inclure des algorithmes génétiques dans nos simulations, il fallait utiliser des *Plugins*.

En conséquence, notre choix c'est fixé sur le moteur physique le plus populaire qui est ODE (Open Dynamics Engine).

2.1. Open Dynamics Engine.

Open Dynamics Engine (ODE) est une bibliothèque logicielle open source classée dans la catégorie des moteurs physiques. Elle sert à simuler l'interaction physique de corps rigides. Elle est actuellement utilisée dans les jeux vidéo, les outils 3D et les outils de simulation.

ODE est disponible sur plusieurs plateformes (Linux, Mac OS ou Windows) et utilise une interface de programmation en C pour une plus grande compatibilité, bien qu'en interne, le code source soit écrit en C++. Cette bibliothèque possède plusieurs types de jointures et intègre un détecteur de collision avec friction. Le moteur utilise plusieurs intégrateurs en fonction de la précision et de la robustesse de la simulation désirées. Plusieurs primitives sont disponibles et le moteur peut gérer les surfaces constituées de triangles [ODE, 2006].

Nous utiliserons les 2 aspects du moteur ODE, à savoir le moteur physique et la gestion des collisions. ODE donnera un aspect « réel » à l'application.

¹² <http://www.spiderland.org/>

¹³ <http://ode.org/>.



A. Corps rigides.

Un corps rigide simulé avec ODE possède les propriétés suivantes :

Les propriétés dynamiques (changent avec le temps):

- Position du vecteur (x, y, z) du point de référence du corps, généralement le point de référence doit correspondre au centre de la masse du corps.
- La vitesse linéaire correspondante au point de référence, qui est un vecteur (v_x, v_y, v_z) .
- L'orientation d'un corps rigide, représentée par un Quaternion (q_s, q_x, q_y, q_z) ou une matrice 3×3 de rotation.
- Vecteur de vitesse angulaire (w_x, w_y, w_z) qui décrit comment l'orientation change avec le temps.

D'autres propriétés sont statiques (constantes dans le temps):

- La masse du corps.
- La position du centre de masse (centre d'inertie). Dans les simulations actuelles le centre de la masse et le point de référence doivent coïncider.
- La matrice d'inertie est une matrice 3×3 qui décrit la manière dont la masse du corps est divisée autour du centre de masse.

B. Intégration du temps.

Le processus d'une simulation d'un corps rigide nécessite l'intégration d'une contrainte très importante : le temps, en l'occurrence. Chaque *pas* (*step*) d'intégration incrémente le temps courant d'une valeur « *step size* » donnée qui fait ajuster l'état de tout le corps par la nouvelle valeur du temps.

C. Application des forces.

Pour chaque pas de simulation, l'utilisateur peut appeler une fonction qui applique une force au corps rigide. Cette force est ajoutée à une accumulation de forces qui sera remise à zéro après chaque *pas* d'intégration.

D. Application de jointures.

Il existe une relation qui est appliquée entre deux corps (primitives) qui peut avoir certaines positions et orientations par rapport aux autres. Cette relation est appelée



contrainte ou jointure. Il existe différents types de jointures pouvant relier les parties d'un corps rigide (primitives géométriques):

- Ball and socket (voir la figure 41.(1))
- Hinge (voir figure 41.(2)) ;
- Slider (voir figure 41.(3)) ;
- Universal (voir figure 41.(4)) ;
- Hing-2 (voir figure 41.(5)) ;
- Fixed ;
- Contact (voir figure 41.(6)) ;

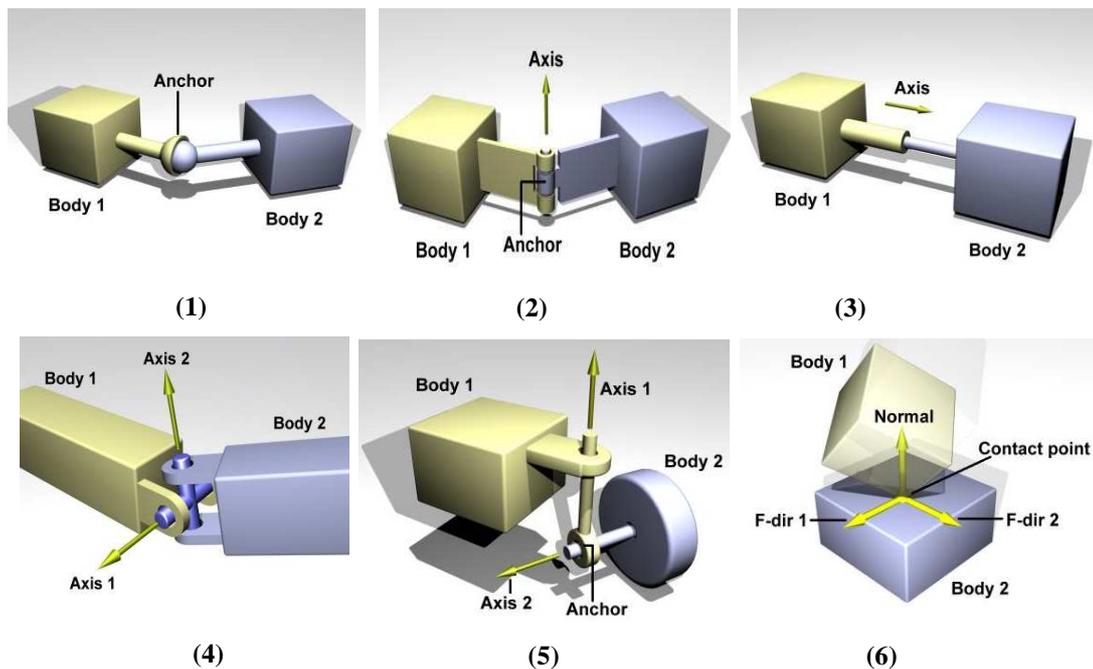


Figure 41: Les types de joints [ODE, 2006].

E. La détection de collision.

ODE possède deux composants principaux: un moteur de simulation de la dynamique et un moteur de détection de collisions. Le moteur de détection de collisions fournit à l'utilisateur des informations sur la forme de chaque corps. A chaque pas de temps, le moteur de détection de collisions détermine le corps qui présente une collision potentielle avec les autres et fournit ensuite à l'utilisateur l'information sur le point de contact. L'utilisateur crée alternativement des joints de contact entre les corps.



- **Geoms.**

Les objets géométriques (primitives géométriques, *Sphere*, *Box*, *Capped cylinder*, *Ray*, *Triangle mesh*, *Simple space*) abrégés par *Geoms* sont les éléments fondamentaux dans le système. Un *Geom* peut être une primitive géométrique ou un groupe de primitives géométriques qui représente un type spécifique de *Geom* appelé *Space*. *Space* est également similaire au concept *World* sauf qu'il est appliqué à la collision et non pas à la dynamique.

- **World.**

L'objet du monde est un récipient pour les corps rigides et les joints. Les objets appartenant à différents mondes ne peuvent pas agir l'un sur l'autre. Par exemple, les corps rigides de deux mondes différents ne peuvent pas se heurter. Tous les objets dans un monde existent au même point dans le temps. Ainsi, une des raisons d'employer les mondes séparés réside dans la simulation des systèmes à différents taux. La plupart des applications nécessiteront l'utilisation d'un seul monde, uniquement.

- **Space.**

Un espace est un *Geom* ne pouvant être placé et qui peut contenir d'autres *Geoms*. Il est semblable au concept de corps rigide du "*monde*", sauf qu'il s'applique à la collision et non pas à la dynamique.

F. Usage d'une carte graphique avec ODE.

ODE est un moteur logiciel, il est complètement indépendant de n'importe quelle bibliothèque graphique (i.e une bibliothèque associée à une carte graphique comme: *PhysiX*¹⁴). Cependant, les réalisations exploitant ODE utilisent OpenGL. Ainsi, et selon les besoins des utilisateurs de ce moteur, les efforts nécessaires à l'exploitation d'une bibliothèque avec une simulation basée ODE, seront exclusivement à la charge de l'utilisateur.

¹⁴ *La carte physiX d'Ageia (Nvidia), carte dédiée aux calculs physiques.*



3. Bibliothèque graphiques.

3.1. OpenGL (Open Graphic Library).

OpenGL est une API (Application Programming Interface) multi plate-forme pour le développement d'applications gérant des images 2D et 3D. Elle a été développée en 1989 par Silicon Graphics, puis portée sur d'autres architectures en 1993. Depuis 1992, sa spécification est surveillée par l'OpenGL Architecture Review Board (ARB).

Son interface regroupe environ 150 fonctions qui peuvent être utilisées pour afficher des scènes tridimensionnelles complexes en décrivant des objets (caméras, lampes, modèles 3D...) et les opérations que l'on peut effectuer pour les manipuler.

Elle à été implémentée sur la plupart des plates-formes informatiques actuelles (Linux et Unix, Windows, MacOS, BeOS...) et est disponible pour de nombreux langages de programmation (C, C++, Java, Fortran, Ada...).

Du fait de son ouverture, de sa souplesse d'utilisation et de sa disponibilité sur toutes ces plates-formes, elle est utilisée par la majorité des applications scientifiques, industrielles ou artistiques 3D et certaines applications 2D vectorielles.

4. Architecture globale du système.

Pour avoir une simulation des mouvements d'une créature 3D sur un environnement physique réaliste, nous avons utilisé des techniques permettant aux mouvements résultant d'avoir un haut niveau de réalisme, donc seule la simulation physique qui sert à donner une interaction complète avec l'environnement.

Le but de cette simulation est de s'inspirer de l'évolution pour trouver un contrôleur associé à une créature 3D simulée. Ce contrôleur aura pour but de garantir la réalisation des mouvements désirés sachant que ces derniers dépendent de la forme de la créature (sa morphologie) et de son environnement (simple, complexe).



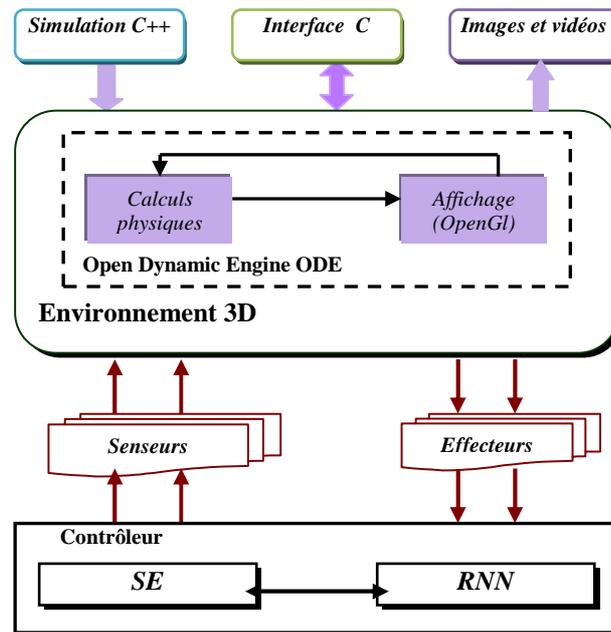


Figure 42: L'architecture globale du système.

La figure ci-dessus (figure 42) nous illustre l'interaction des différentes techniques utilisées pour trouver le contrôleur qui est un réseau de neurones de type RNN (Recurrent Neural Network). Son évolution est assurée par une des techniques évolutionnistes qui sont les stratégies d'évolution. Les deux principales techniques (le réseau de neurones et les stratégies d'évolution) coopèrent pour garantir les mouvements voulus alors qu'il y a une interaction dans les deux sens entre ces deux techniques. Ce contrôleur transfère, à chaque génération, les données depuis et vers l'environnement qui est simulé par un moteur physique, ce transfert étant assuré par des senseurs et des effecteurs.

Ainsi, et globalement, le système est un assemblage de deux parties, une concerne le cerveau de la créature (qui est son contrôleur) et la deuxième concerne *l'environnement* et la *créature*, sachant que ses deux derniers sont simulés par le moteur physique.

4.1. Environnement simulé.

L'environnement simulé, comme précédemment mentionné, est une condition (contrainte) nécessaire à la naissance de certains mouvements réalisés par la créature. Pour



cela, nous avons choisi un environnement complexe contenant divers objets de différentes formes et tailles. Pour être similaire à l'environnement réel, un environnement doit intégrer les notions de gravité, de masse ainsi que des capacités de traitement de collision.

L'environnement virtuel : la complexité de l'environnement joue un grand rôle dans le succès de l'évolution des organismes biologiques complexes. Pour réduire la complexité de la simulation, le moteur physique utilise le modèle de *Newton* ainsi que l'équation d'*Einstein*. En employant un environnement qui modélise le monde réel physique, tous les agents qui existent dans un tel environnement doivent être capables de gérer des mouvements qui sont reproductibles dans un environnement réel.

De plus, la créature (ou agent) qui vie dans cet environnement est composée, de série, de corps rigides qui sont interconnectés ensemble via des jointures. Le moteur physique agit directement sur les corps rigides qui composent la créature. L'agent lui même est sujet aux règles de la physique qui gouvernent l'environnement. Une jointure interconnecte deux corps rigides ou plus, elle peut être une jointure permanente (*Hinge, Ball et socket joint*) ou temporaire (*Contact joint*). Cela est appliqué, dans notre modèle de créature, au niveau des pieds de la créature (humanoïde) car une fois que ses pieds touchent le sol, les jointures du *contact* temporaire sont créées.

La gravité (accélération gravitationnelle) dans l'environnement virtuel est à $9,8\text{m/s}^2$. La friction au niveau du sol est nécessaire pour la locomotion qui est modélisée au niveau des joints temporaires qui sont créés lors d'une collision.

4.2. La morphologie de la créature proposée.

La morphologie détermine la forme que peut avoir une créature et la façon avec laquelle les différentes parties sont connectées. Il y'a plusieurs méthodes et techniques permettant la représentation d'une morphologie telles que les L-systèmes ou les graphes (employés par Carl Sims) et ceci dans le cas où la morphologie de la créature doit être modifiée par évolution. Dans le cas inverse, où la morphologie est figée ou prédéterminée, seul le contrôleur peut évoluer pour générer des mouvements au niveau d'une créature ou d'un robot.



Dans le domaine des robots autonomes, il existe plusieurs types différents de robots, les plus communs sont les robots à roues et les robots de marche. Ceci signifie que si on considère un robot, c'est d'une créature avec morphologie figée qu'il s'agit. Dans le sous-domaine des créatures de marche, il existe des créatures bipèdes¹⁵ et d'autres quadrupèdes¹⁶, aussi bien que des créatures avec plus de quatre membres locomoteurs. Des créatures bipèdes avec une forme approximativement humaine sont dites également des *créatures humanoïdes*. [Wah, 2005]

Des méthodes jusqu'ici diverses ont été proposées pour des bipèdes, quadrupèdes, hexapode¹⁷ et des créatures octopodes¹⁸ qui, malgré leur mobilité remarquable, peuvent présenter une instabilité élevée.

Il existe de nombreux domaines d'application pour des créatures avec des possibilités anthropomorphes¹⁹ de forme et de mouvement. De tels robots peuvent fonctionner dans certains secteurs qui ne sont pas accessibles pour les créatures à roues, tels que les escaliers ou les terrains naturels (accidentés), par exemple. En outre, les créatures bipèdes capables de locomotion ont la capacité d'agir sur l'environnement en utilisant le corps entier qui peut surmonter de grands obstacles existants sur leurs chemins.

Le modèle de créature utilisé ici est vu en tant que représentation d'un modèle d'humanoïde générique. Il a été créé à partir de primitives de corps et de joints disponibles sur ODE. Le modèle d'humanoïde se compose de 12 primitives géométriques de corps-rigide, et 13 joints : 8 *Hinge joint*, 2 *Ball and socket joint*, 2 *contact joint* et 1 *fixed joint*. Ces joints reliant les primitives du corps dans une structure articulée de corps-rigide.

Les primitives géométriques utilisées sont : 8 *Caped cylindres*, 3 *parallélépipèdes* et une *sphère*. La structure de l'humanoïde est définie en utilisant les chaînes multiples, à partir de ses pieds avec chaque lien décrit en termes de liens précédents. Cette composition a comme conséquence un modèle d'humanoïde possédant 15 degrés de liberté (DOF). La structure du corps rigide de notre humanoïde qui est simulé, avec l'organisation des liens entre les différentes primitives, est représentée par la figure 43 et le tableau 3.

¹⁵ Des créatures avec deux jambes.

¹⁶ Des créatures à 4 pattes.

¹⁷ Des créatures à 6 pattes.

¹⁸ Des créatures à 8 pattes.

¹⁹ Ressemblant à un humain en forme et attributs.



Comme motionné dans le tableau 3, les types de jointures qui sont utilisées dans notre modèle d'humanoïde sont : *hinge joint*, *ball & socket joint*, *Fixed joint* et *contact joint*²⁰.

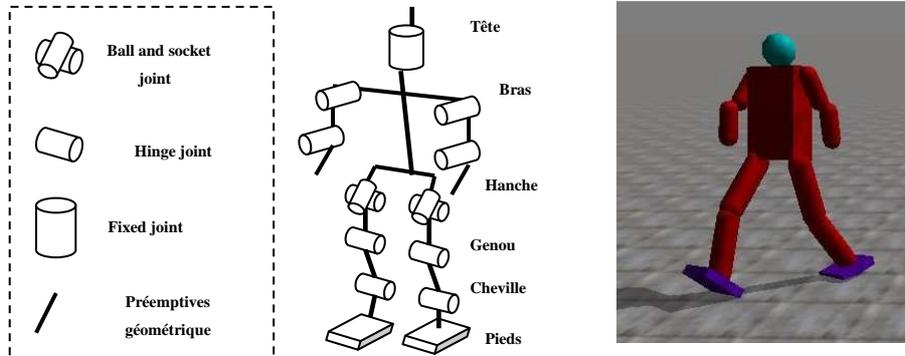


Figure 43: Le modèle de la créature humanoïde simulé par ODE.

Partie du corps	Jointure	Primitive géométrique
Tête	Fixed joint	Sphère
Bras	Hinge joint	Caped cylindre
Hanche	Ball and socket joint	Parallélépipède
Genou	Hinge joint	Caped cylindre
Cheville	Hinge joint	Caped cylindre
Pieds	Contact joint	Parallélépipèdes

Tableau 3: Les paramètres de simulation de la créature humanoïde.

Nous donnons ci-après une explication détaillée sur le fonctionnement de chacune des jointures utilisées et cela selon nos besoins de simulation.

- **Hinge Joint.**

La jointure de ce type, contraint le mouvement des deux parties reliées par cette jointure et cela pour les faire tourner autour d'un axe prédéterminé. Le joint de type *hinge*, considéré comme la jointure la plus simple, tient compte de seulement un degré de liberté. Le contrôleur de notre humanoïde (RNN) utilise seulement 2 senseurs :

- (1) angle *hinge* courant,

²⁰ *Hinge*, *Ball and socket*, *contact* et *fixed* joints sont les noms des types de jointures existantes dans ODE.



(2) le taux de l'angle *hinge* courant.

Notant $h(t)$ l'angle courant entre 2 parties du corps à l'instant t , et $r(t)$ le taux de l'angle *hinge* qui est défini comme la dérivée de l'angle *hinge* en fonction de temps :

$$r(t) = \frac{d}{dt} h(t) \quad (1)$$

La valeur retournée pour l'angle *hinge* sera entre $-PI$ et PI , cette jointure possède deux senseurs qui peuvent fournir l'angle courant et la vitesse angulaire du corps, la jointure possède également un effecteur qui affecte la vitesse désirée en tant que son entrée.

- **Ball & socket joint.**

Ce type de jointure est le plus complexe des jointures existantes dans ODE, cette jointure tient compte de 3 degrés de liberté autour des 3 axes. De plus, cette jointure fournit 2 senseurs de rétroaction pour chaque axe. Ces deux senseurs fournissent les données suivantes :

- (1) l'angle Ball & socket pour l'axe donné,
- (2) le taux de l'angle Ball & socket pour l'axe donné.

Le taux de l'angle pour un axe est calculé en prenant la dérivée en ce qui concerne la période de l'angle courant sur l'axe particulier. En raison des 3 degrés de liberté, les 2 senseurs peuvent fournir jusqu'à 6 informations au RNN.

- **Contact joint.**

Le joint de contact empêche le corps 1 et le corps 2 de s'interpénétrer au point de contact.

Il fait ceci en permettant seulement aux corps d'avoir une vitesse "sortante" dans la direction de la normale de contact.

Des joints de contact sont dynamiquement créés et supprimés en réponse à la détection de collision. Les joints de contact simulent le frottement au contact en appliquant des forces dans les 2 directions de frottement (friction) qui sont perpendiculaires à la normale [Smi, 2002]. Les joints de contact ont typiquement une vie liée au pas de temps. Ils sont créés et supprimés en réponse à la détection de collisions.



- **Fixed joint.**

Un joint fixe maintient une position et une orientation relatives fixes entre deux corps, ou entre un corps et l'environnement statique. L'utilisation de cette jointure n'est presque jamais une bonne idée dans la pratique, exceptée dans le cas où on procède à des corrections. Si on a besoin de deux corps pour être collés ensemble, il est préférable de les représenter comme un simple corps.

4.3. Le contrôleur de l'humanoïde.

Le contrôleur d'une créature est lié à sa morphologie et à ses capteurs qui lui permettent de percevoir le monde. Si nous voulons que le contrôleur émerge d'un assemblage d'éléments et non de la conception d'une intelligence et d'une expérience humaine, nous pouvons nous inspirer des mécanismes de la nature. Cette approche peut s'appliquer aux réseaux de neurones. En effet, l'assemblage de neurones permet de constituer une topologie dont un comportement pourra émerger. Les réseaux de neurones sont largement répondus dans l'ensemble des travaux traitant des créatures artificielles.

Il existe plusieurs autres méthodes pour représenter un contrôleur dont nous pouvons citer, parmi les plus connues : les fonctions logiques, les machines à états finis, les équations différentielles ou les réseaux de neurones. Dans le domaine des créatures artificielles, ce sont les réseaux de neurones qui sont principalement utilisés.

4.3.1. Les réseaux de neurones récurrents.

Les réseaux de neurones ont été employés pour notre humanoïde en tant qu'outil efficace pour la synthèse et l'adaptation de sa démarche. Un autre rôle important des systèmes connexionniste dans le contrôle des créatures artificielles tridimensionnelles réside dans la résolution de l'équilibre pendant le processus de marche sur le terrain avec différentes caractéristiques de l'environnement. Le type de réseau de neurones choisi pour contrôler le modèle d'humanoïde présenté ici est un réseau de neurone récurrent (RNN) [Hay, 1999].



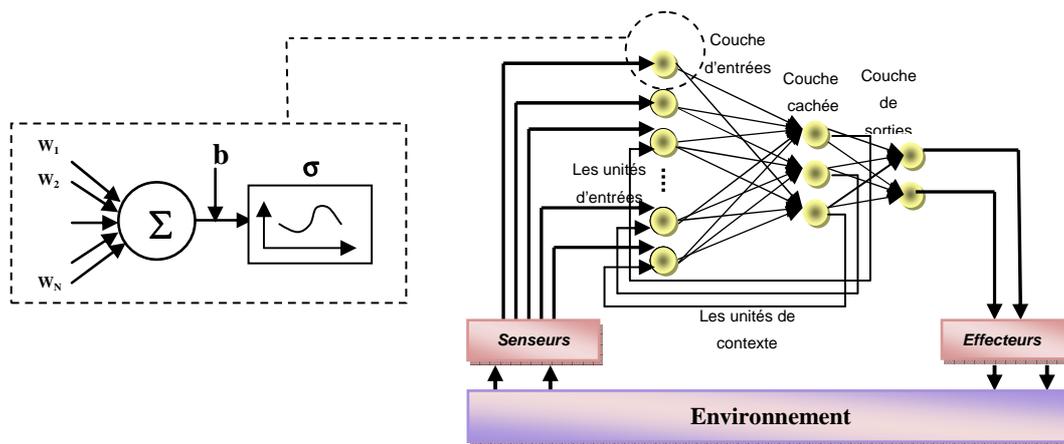


Figure 44: Le réseau de neurones RNN.

Le réseau RNN est un réseau de neurones multicouche entièrement connecté également connu sous le nom de réseau d'Elman. La topologie du réseau se compose de 4 couches : une couche d'*entrée*, une couche *cachée*, une couche de *sortie* et une couche de *contexte*. Le nombre de neurones contenus dans la couche d'entrée et la couche de sortie dépendent de la morphologie de l'humanoïde (ensemble de ses senseurs et de ses effecteurs), un exemple de RNN est abordé par la figure 44.

La raison d'un tel choix réside dans le fait que les RNNs utilisent la notion de temps et qu'ils sont plus puissants en terme de calcul. Dans un RNN, on utilise un simple modèle de neurones, où le comportement dynamique du $i^{\text{ème}}$ nœud dans le réseau est régi par l'équation suivante :

$$\tau_i \dot{y}_i + y_i = \sigma(b_i + \sum w_{ij} y_j + \sum w_{ij}^I I_j), \quad i=1, \dots, n \quad (2)$$

Où n est le nombre de neurones dans le réseau, τ_i sont les constantes de temps, y_i est la sortie (activité) du nœud j , w_{ij} sont les poids (synaptiques) reliant le nœud i au nœud j , w_{ij}^I est le poids d'une interconnexion entre le nœud d'entrée j et le nœud i , I_j est la $j^{\text{ème}}$ entrée externe du nœud i , et b_i est la limite polarisée, qui détermine la sortie du nœud en absence de l'entrée. $\sigma(\cdot)$ est une fonction sigmoïdale dont le but principal est de limiter l'activité des neurones à une gamme donnée. La fonction sigmoïdale d'activation du RNN choisie est la fonction sigmoïdale bipolaire (équation (2)) qui est utilisée au niveau des deux couches cachée et de sortie, où sa courbe est représentée par la figure 45.



$$\sigma(x) = \frac{2}{1+e^{-\alpha}} - 1 \quad (3)$$

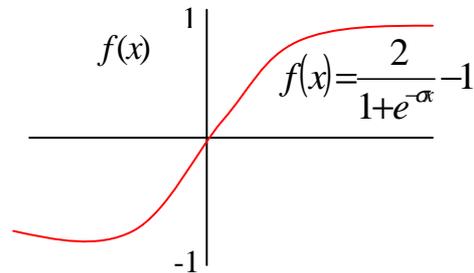


Figure 45: La courbe de la fonction sigmoïde bipolaire.

Le sigmoïde bipolaire est étroitement lié à la fonction hyperbolique de tangente, qui est souvent employée comme fonction d'activation lorsque la gamme désirée des valeurs de rendement est entre -1 et 1 . La tangente hyperbolique est :

$$h(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} \quad (4)$$

Les effecteurs prennent les valeurs d'entrée et les convertissent en une vitesse désirée en mesurant linéairement les entrées des effecteurs vers la gamme d'entrées suivantes $[-1,1]$. Le maximum force/couple de chaque type d'effecteur est prédéterminé. En utilisant les équations ci-dessous, le moteur physique peut rapidement déterminer l'accélération éprouvée par les corps attachés au joint basé sur la vitesse désirée réglée par le RNN. Les effecteurs utilisent un modèle simple des moteurs existants dans la réalité, avec deux paramètres:

- (1) La vitesse désirée,
- (2) La force maximum qui est disponible pour atteindre la vitesse désirée.

Les effecteurs peuvent également être employés pour modéliser les moteurs adaptés. De tels dispositifs sont souvent contrôlés en fixant la vitesse désirée, et peuvent développer, uniquement, une quantité maximum de puissance pour réaliser cette vitesse [Smi, 2002]. En utilisant le modèle adapté, nous pouvons efficacement réduire le nombre de sorties requises par le RNN juste à un par effecteur: la vitesse désirée.



Les joints de type « *hinge* » et « *ball and socket* » sont modélisés en utilisant la deuxième loi de Newton pour la rotation pour proposer les équations de base de couple en fonction de la force, de l'angle et du rayon qui est défini comme suit:

$$\text{Couple: } \tau = F \times r \times \sin \theta \quad (5)$$

Où τ est le couple, r représente le rayon, F est la force et θ est l'angle entre la ligne faite par r et F . L'équation de base de couple est utilisée pour calculer combien de couple est appliqué aux effecteurs des jointures de type « *hinge* » ou celui de type « *ball and socket* ».

L'autre force de rotation que le moteur physique doit considérer est le moment de l'inertie. Le moment de l'inertie d'un corps rigide représente la résistance au mouvement de rotation. Il dépend de l'axe qui est en rotation ainsi que de la masse de l'objet. Le moteur physique calcule le moment de l'inertie I , d'abord en divisant le corps rigide en plusieurs petits morceaux, ensuite en multipliant la masse m de chaque morceau par le carré de la distance depuis son axe de rotation r , et d'additionner ensuite tous ces produits ensemble :

$$\text{Moment d'inertie : } I = m \times r^2 \quad (6)$$

Les corps rigides de cette simulation ont une distribution homogène de la masse au centre du corps de la masse.

4.3.2. Ensemble des senseurs et d'effecteurs.

L'humanoïde est la seule entité dynamique dans tout le modèle et seul le contrôleur (RNN) assure les mouvements de cet humanoïde. Pour ce qui est du moyen d'interaction de cet humanoïde avec l'environnement nous utilisons un ensemble de senseurs et d'effecteurs.

L'ensemble des senseurs externes constituent la "*fenêtre*" de l'humanoïde avec l'environnement. L'utilisation de ces senseurs a pour but de collecter les informations depuis l'environnement et les effecteurs pour les transmettre au contrôleur (RNN). Les senseurs utilisés pour notre modèle de simulation de robot sont :

- Deux senseurs de contact au niveau des pieds droit et gauche pour indiquer si le pied est en contact avec le sol ou non (1 ou 0) ;



- Un capteur de vitesse angulaire au niveau de chaque articulation de type Hinge (mesuré entre -1,0 et 1,0 selon les limites minimum et maximum des angles) (i.e 8 capteurs de ce type);
- Un capteur de vitesse angulaire pour chaque degré de liberté pour les articulations de type Ball & socket (six capteurs).
- Un dernier capteur est celui de la direction indiquant l'axe de locomotion (axe x ou y).

La figure ci-dessous (figure 46) résume les différents capteurs utilisés dans notre modèle d'humanoïde ainsi que les lieux de chacun.

La simulation d'un humanoïde dans un environnement réaliste et complexe exige très probablement un ensemble de capteurs permettant à l'humanoïde d'éviter les obstacles de l'environnement et pour cela nous avons utilisé, en plus de l'ensemble de capteurs décrits ci-dessus, deux autres capteurs sur chaque pied et cela pour permettre à notre humanoïde de bien préciser les lieux des obstacles dans l'environnement.

Chaque articulation possède un moteur associé à elle connu sous le nom d'un effecteur. Un effecteur applique un couple aux degrés de liberté des articulations. Dans notre simulation, nous avons utilisé 18 effecteurs répartis comme suit :

- Un effecteur pour contrôler les muscles de chaque articulation. Cet effecteur détermine les vitesses angulaires que les muscles essaieront de maintenir à n'importe quel moment donné
- Au niveau des pieds, nous avons appliqué des vitesses linéaires pour le déplacement de toute la créature vers l'avant.

En tout, nous avons utilisé, pour notre modèle de réseau de neurone (i.e le contrôleur), 17 capteurs et 18 effecteurs. Ceci implique que le nombre de neurones va être grand et cela va complexifier la structure du RNN et induire des difficultés dans la réalisation de notre travail.



- 1 Senseur de contact
- 2 Senseur de vitesse angulaire pour hinge joint
- 3 Senseur de vitesse angulaire pour Ball & Socket joint
- 4 Senseur de direction
- 5 Senseur de vitesse linéaire

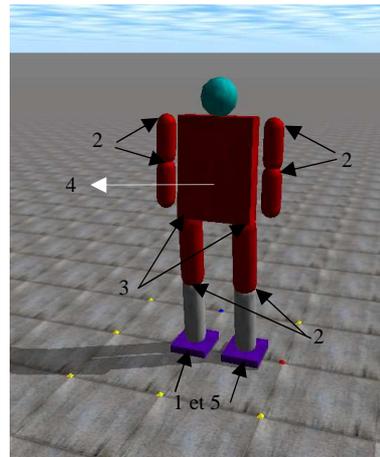


Figure 46: L'ensemble de senseurs associés à la créature simulée.

4.4. Stratégies d'évolution (SE).

Depuis l'introduction des algorithmes évolutionnaires dans les années 70, le nombre d'applications de ces algorithmes s'est développé solidement. Aujourd'hui, ces algorithmes sont employés dans des domaines aussi variés que la technologie, la biologie l'informatique, les finances, l'astrophysique, et naturellement la robotique. Un cas spécial de ces algorithmes consiste est lié aux stratégies d'évolution (SE) [Rec, 1994], dans lequel les variables du problème posé sont codées par un ensemble de nombres réels (c-à-d un vecteur) qui, une fois décodés, produisent le système qui doit être optimisé (par exemple le cerveau de l'humanoïde représenté, par un réseau de neurones).

Les stratégies d'évolution (SE) sont des techniques d'optimisation dérivées de la théorie évolutionniste de Darwin. Ils opèrent sur une population d'individus (solutions potentielles à un problème), mettant à jour la population au cours de générations. Chaque individu est représenté dans l'algorithme évolutionnaire par une liste de nombres et dans notre simulation nous avons utilisé les poids (i.e des valeurs réelles) du réseau de neurones RNN comme étant l'ensemble des gènes qui construisent le chromosome d'un individu de la population à faire évoluer. (Voir figure 47)

W_{11}	W_{12}	W_{13}	W_{14}	W_{15}	W_{16}	W_{31}	W_{32}	W_{33}
----------	----------	----------	----------	----------	----------	-------	----------	----------	----------	-------

Figure 47 : Le format du chromosome utilisé.



Nous avons utilisé l'algorithme évolutionnaire pour représenter les stratégies d'évolution dans le but d'optimiser les poids du réseau de neurones avec une architecture fixe et cela pour arriver à des valeurs permettant à notre humanoïde de se déplacer dans l'environnement tout en évitant les obstacles rencontrés. Pour cela, nous devons conjuguer une fonction de *fitness* de graduation prédéterminée qui est la distance parcourue au bout d'un temps fixe. La figure 48 représente la boucle selon laquelle l'algorithme évolutionnaire fonctionne de génération en génération. Cette figure spécifie que l'algorithme évolutionnaire est composé de trois parties formant une boucle. Son arrêt est dépendant de la fonction de *fitness* : c'est-à-dire si cette fonction est satisfaite et si les mouvements obtenus sont ceux désirés alors on peut interrompre la boucle et arrêter le processus.

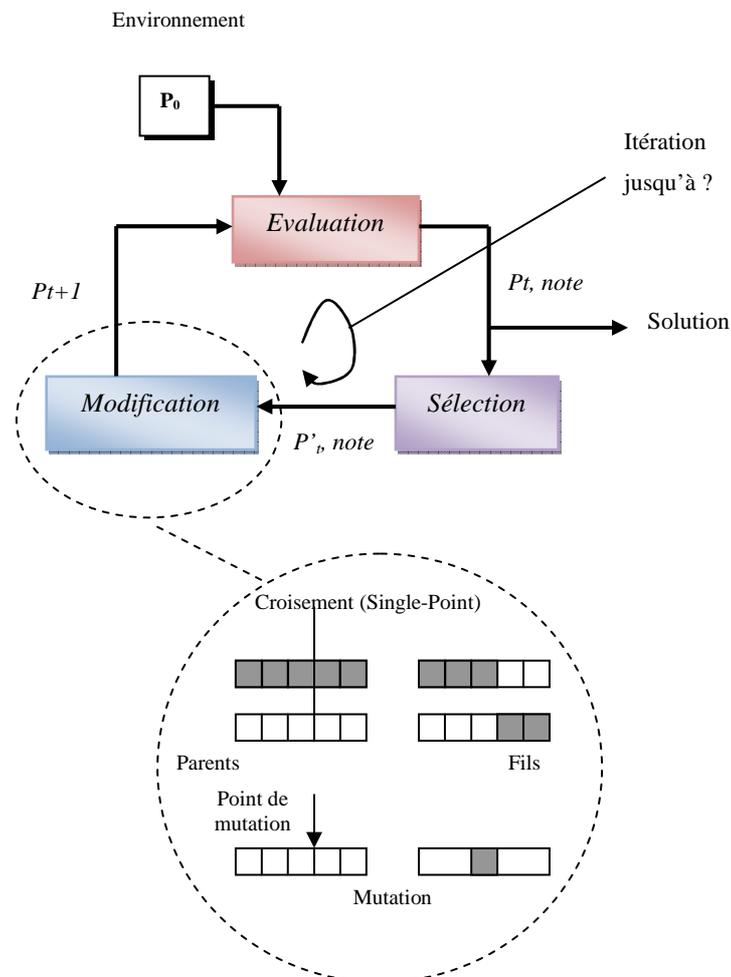


Figure 48: La boucle de l'algorithme évolutionnaire.



Pour les paramètres de l'algorithme évolutionnaire, nous avons utilisé le suivant (voire tableau 4) :

Taille de la population	100
Elitisme	20%
Taux de croisement	70%
Taux de mutation	1%
Type de sélection	La roue de la fortune
Taille du chromosome	1130 (45 nœuds)
Le nombre de générations	400 ou plus

Tableau 4 : Les paramètres des stratégies d'évolution.

Optimisation évolutionnaire du RNNs.

Dans les stratégies d'évolution qui sont des techniques d'optimisation évolutionnaire, les variables du problème sont codées dans une chaîne de longueur constante (i.e un vecteur). Cet algorithme agit directement sur le RNNs.

Le codage de nombres réels est employé, c.-à-d., tous les gènes prennent des valeurs réelles qui sont appropriées au poids du réseau de neurones. Après chaque évaluation, une valeur de la fitness est assignée à chaque individu. Les détails du processus d'évaluation sont décrits dans section 4.4.3.

Dans les SE, le remplacement générationnel est employé, ce qui veut dire que certains individus sont remplacés par leur fils dans chaque génération. La sélection des individus (pour la reproduction) est faite au moyen de la méthode de sélection du tournoi (Roulette Wheel Selection). En outre, l'élitisme est employé, comme suit : une copie simple du meilleur individu est transférée sans modification à partir d'une génération à celle qui la suit.

L'algorithme des stratégies d'évolution fonctionne dans notre cas comme suit :



1. création de la population initiale avec 100 individus;
2. Evaluation individuelle des 100 individus de la population comme suit :
 - a. Initialisation de la position de départ (centre de l'environnement $(0, 0, h)$, où h est la hauteur de lancement de la créature);
 - b. Activation du contact avec le sol, activation des senseurs et des effecteurs;
 - c. Initialisation du chromosome avec des poids aléatoires du réseau de neurones;
 - d. Dessin de la créature (Rendu + paramètres physiques);
 - e. Calcul de la fitness après un temps de simulation de 10 secondes.
3. Calcul du maximum et de la moyenne ainsi que du total des notes des individus évalués;
4. Sélection des meilleurs individus par la méthode de la Roulette pipée;
5. Modification des individus sélectionnés (croisement et mutation) et obtention des fils;
6. remplacement des parents par leurs fils;
7. Retour à l'étape 1.

4.4.1. La création de la population.

La population à faire évoluer, pour trouver une solution au problème posé, qui est la locomotion d'un humanoïde, est un ensemble d'individus (100 individus) où chaque individu représente la même forme physique de la créature mais avec un contrôleur différents pour chaque individu, ce qui signifie que le meilleur individu obtenu est le meilleur contrôleur pour doter l'humanoïde des facultés de la marche.



4.4.2. L'élitisme.

Bien qu'un individu convenable ait une grande probabilité d'être choisi pour la reproduction, il n'y a aucune garantie qu'il serait choisi. En outre, même s'il est choisi, il est probable qu'il serait détruit pendant le croisement ou la mutation. Afin de s'assurer que le meilleur individu ne serait pas perdu, il est commun de faire une ou plusieurs copies exactes de cet individu et de les placer directement dans la prochaine génération. Cette procédure est connue sous le nom de "*élitisme*". Dans notre cas, nous avons utilisé 20% des meilleurs individus élus de la population qui sont conservés pour la génération suivante. Tous les autres nouveaux individus (80%) sont obtenus via la séquence habituelle: sélection, croisement et mutation.

4.4.3. Evaluation.

L'évaluation consiste à attribuer à chacune des solutions une note qui correspond à son adaptation et sa conformité au problème. Pour la réaliser, on plonge l'individu dans le simulateur pendant un temps fixe en tenant compte des valeurs prises par les senseurs (telles que ceux de direction ou du contact avec le sol) et les valeurs reçues des effecteurs. Ainsi, avec les mouvements obtenus, on évalue les individus un par un. Cette étape est primordiale car elle sera prise en compte lors de la sélection des individus qui participeront à la prochaine génération. Elle influencera donc l'évolution globale des individus.

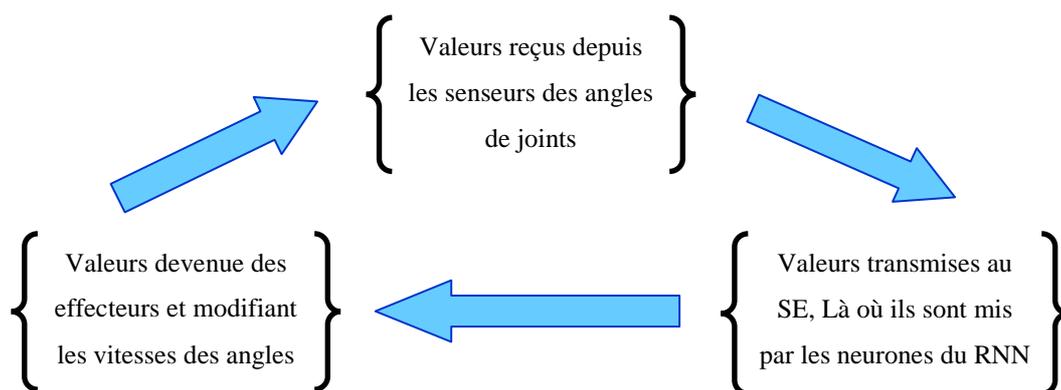


Figure 49: Boucle d'évaluation d'un individu de la population

La figure 49 représente la circulation des données qui proviennent des senseurs dont le rôle consiste à mesurer l'interaction avec l'environnement. Les données sont ensuite modifiées



par des neurones et cela en ajustant aléatoirement les poids des entrées des neurones car on utilise une fonction d'activation dans le réseau de neurones. En conclusion, les données modifiées sont exploitées au niveau de la jointure comme valeurs de la vitesse. Le cycle est repris une nouvelle fois encore quand les senseurs reçoivent de nouvelles données après que les jointures aient subi des déplacements.

- **La fonction de fitness.**

L'évaluation d'un individu de la population est réalisée en affectant une note à ce dernier, cette note étant la valeur de la fitness calculée par la fonction de fitness qui est dans notre cas la distance parcourue (équation 7). Cette fonction est appliquée selon certains critères qui sont :

- Le temps de simulation qui est fixé à 10 secondes pour chaque évaluation d'un individu dans le simulateur.
- La direction du mouvement qui est sur l'axe des x.

$$f(x) = \alpha(-x - x_0) - (|y - y_0|)^\beta \quad (7)$$

$$Fitness = \begin{cases} F(x) & F(x) > 0 \\ 0 & F(x) \leq 0 \end{cases}$$

- x_0, y_0 sont les deux coordonnées initiales du lieu de départ de la créature simulée.
- x et y sont les coordonnées de la position de la créature au moment t .
- α et β sont successivement des facteurs de multiplication et de pénalité.

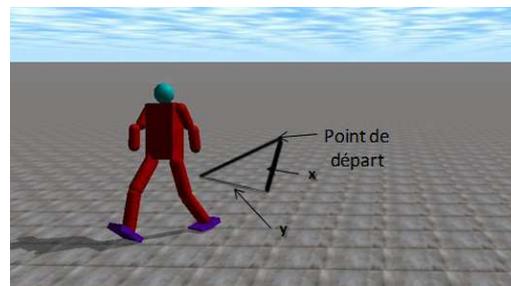


Figure 50: Evaluation d'un individu de la population sur le simulateur.



4.4.4. Sélection.

Pour déterminer quels sont individus les plus enclins à obtenir les meilleurs résultats, une sélection est opérée. Ce processus est analogue à un processus de sélection naturelle, les individus les plus adaptés gagnent la compétition de la reproduction tandis que les moins adaptés meurent avant la reproduction, ce qui améliore globalement l'adaptation.

La méthode de sélection que nous avons choisie est la méthode de *la roue de la fortune* (appelée aussi roulette pipée). Celle-ci permet d'introduire un peu de hasard en permettant la sélection d'individus totalement différents. Le principe est simple et consiste à associer à chaque individu de la population un poids en terme de *fitness*. La roue de la fortune consiste à tirer au hasard un nombre compris entre 0 et la somme de toutes les *fitness*. Ce nombre va permettre de déterminer quel individu va être sélectionné pour le croisement en parcourant la liste des individus dans l'ordre et en accumulant la *fitness* jusqu'à ce qu'elle soit supérieure ou égale au nombre tiré au hasard. Les individus qui ont les meilleures *fitness* auront le plus de chance d'être choisis. On peut, ainsi, choisir deux ou plusieurs fois le même individu.

4.4.5. Modification.

Au niveau de chaque génération, la population est mise à jour, l'opérateur de croisement utilisé étant le standard "*single-point*" tel que décrit par Goldberg [Gol, 1989], l'opérateur de mutation utilisé consiste à muter un gène par une perturbation gaussienne. Le croisement, est responsable de l'échange de l'information génétique entre les individus sélectionnés alors que la mutation, qui est un opérateur stochastique, est responsable de présenter le nouveau matériel génétique dans la population.

- **Croisement.**

Le croisement permet aux solutions spatiales de différentes régions de l'espace de recherche d'être assemblées dans une solution complète au problème posé. La procédure de croisement peut être appliquée de plusieurs manières. La version la plus simple est le croisement en un seul point (*single-point*) pour laquelle un seul point de croisement est



aléatoirement choisi, et la première partie du premier chromosome est jointe avec la deuxième partie du deuxième chromosome, comme illustré dans figure ci-dessous.

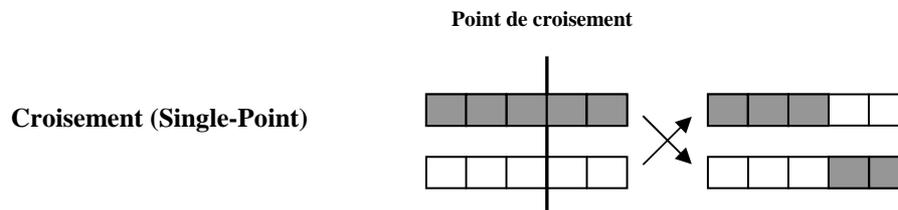


Figure 51: Un diagramme pour le processus de croisement en un seul point.

Tandis que le croisement joue un rôle très important, ses effets peuvent être négatifs si la taille de la population est petite, ce qui est presque toujours le cas dans l'évolution artificielle où la taille N de la population est typiquement de l'ordre de 30 à 1.000, par rapport aux populations de plusieurs milliers voire millions d'individus dans la réalité.

Un remède possible ne doit permettre le croisement ou la reproduction uniquement avec une certaine probabilité p_c . Dans ce cas là, quelques nouveaux individus sont formés à l'aide du croisement suivi de la mutation, et quelques individus sont formés en utilisant la reproduction, dans laquelle seules les mutations sont impliquées. Le p_c choisi dans ce cas est de 80%.

• Mutation.

Dans le domaine de l'évolution naturelle, la mutation joue un rôle permettant de fournir aux deux autres opérateurs principaux, sélection et croisement, le nouvel élément qui sera le point de départ de l'étape suivante. Le plus souvent, les mutations sont néfastes et nuisibles quand elles se produisent. Elles peuvent, cependant, apporter des avantages en fin de compte. Par exemple, quand l'environnement subit soudainement des changements tels que les individus sans mutation qui présenteront des difficultés de survie.

Dans les algorithmes évolutionnaires, la valeur de la probabilité de mutation p_{mut} est habituellement donnée par l'utilisateur au début du traitement.

Dans le cas des chromosomes utilisant le codage binaire (i.e l'algorithme génétique), la mutation consiste normalement à changer la valeur d'un gène en sa valeur complémentaire, c.-à-d. changer le 0 en 1 ou le 1 en 0, selon la valeur avant la mutation.



Dans le codage des nombres réels (i.e les stratégies d'évolution), les modifications obtenues en choisissant aléatoirement de nouvelles valeurs deviennent souvent trop grandes pour être utiles et donc une approche alternative, connue sous le nom de fluage de nombres réels, est fréquemment employée en échange. Dans le fluage de nombres réels, la valeur après mutation n'est pas totalement indépendante de la valeur avant la mutation comme dans les codages discrets. Au lieu de cela, la mutation portera sur la valeur précédente et le taux de fluage détermine à quelle distance la mutation peut prendre la nouvelle valeur.

En ce qui concerne l'opération de mutation que nous avons choisie dans notre simulation, le gène du chromosome qui va être muté possèdera une probabilité d'être muté. Ainsi, l'opération de mutation utilise la perturbation gaussienne plutôt qu'une mutation aléatoire. En perturbant les poids plutôt qu'en choisissant aléatoirement des valeurs pour les poids mutés, on converge vers un changement progressif, l'opération de mutation utilisée l'est avec un taux de 1% (i.e. 0.01).

4.4.6. Remplacement.

Le remplacement générationnel a été employé dans le cadre de notre travail, signifiant que tous les individus dans la génération évaluée ont été remplacés par un nombre égal de fils. Le remplacement générationnel n'est pas très réaliste d'un point de vue biologique. Dans la nature, les différentes générations coexistent et les individus apparaissent (et disparaissent) constamment, non nécessairement à des intervalles de temps spécifiques. En revanche, en remplacement générationnel, il n'y a aucune concurrence entre les individus de différentes générations.

4.5. Paramètres d'un algorithme évolutionnaire.

Le résultat final de l'application d'un algorithme évolutionnaire dépendra de certains paramètres dont la description va suivre.



4.5.1. Taille de la population.

Ce paramètre définit le nombre d'individus de la population par génération. Plus la population est grande, plus la diversité des individus générés au départ sera importante.

4.5.2. Conditions d'arrêt.

L'algorithme évolutionnaire fonctionne suivant une boucle qui se répète d'une génération à l'autre. Cette boucle ne s'arrête que si certaines conditions sont vérifiées. Le critère d'arrêt intuitif des stratégies d'évolution est obtenu lorsque la solution définie par le génome d'un individu est la solution optimale connue du problème posé. En d'autres termes, selon le cas envisagé, si les valeurs²¹ du génome correspondent au mouvement de locomotion d'un humanoïde on pourra dire donc ce cas que la solution optimale a été trouvée.

Cependant, une telle solution n'est pas connue dans la plupart des problèmes pour lesquels cette méthode d'optimisation est utilisée. Afin de pallier cette difficulté, de savoir si l'algorithme a effectivement trouvé la solution optimale, plusieurs techniques sont couramment utilisées, celles-ci ont été mentionnées dans l'état de l'art. En ce qui concerne notre cas, nous avons choisi la technique qui considère l'arrêt si on a obtenu une solution optimale et lorsque le meilleur individu n'a pas été amélioré depuis un certain nombre de générations, ce qui signifie que l'algorithme a effectivement convergé.

4.5.3. Croisement.

Lors d'un croisement, les trois paramètres suivants sont pris en compte :

- Nombre de points de croisements ;
- Taux de croisements ;
- Taux de mutations.

Remarque.

Lors du déplacement des créatures, il serait intéressant de calculer l'énergie dépensée afin de générer le mouvement qui dépense le minimum d'énergie. Cela est intéressant du point

²¹ Se sont les poids du RNN (contrôleur) qui se traduisent par des angles pour les différentes jointures formant le corps de la créature permettant à cette dernière de reproduire un mouvement.



de vue de l'animation des créatures afin de produire des mouvements plus réalistes. En ce qui concerne la robotique, les batteries des robots sont un obstacle en matière d'autonomie : en minimisant la dépense énergétique consacrée au déplacement, on pourrait augmenter leur autonomie de fonctionnement. De plus, calculer l'énergie dépensée est indispensable lorsqu'on réalise un écosystème basé sur la sélection naturelle, car elle détermine la survie de la créature.

Pour n'importe quel robot autonome qui porte sa propre source d'énergie (par exemple batteries), il est clairement important et primordial qu'il puisse se déplacer avec le minimum d'énergie possible. En nature, l'évolution permettrait d'optimiser la marche d'humain (et, en général, la locomotion animale), tout en optimisant l'énergie consommée. Tandis que nous appliquons l'évolution artificielle pour optimiser la démarche d'un robot simulé, nous devons préciser que ce problème d'optimisation diffère de l'optimisation réalisée par évolution naturelle.

5. Récapitulatif.

Tous au long de ce chapitre, nous avons proposé une architecture d'un système de contrôle d'une créature 3D qu'une technique évolutionnaire permet de faire évoluer. Ainsi, nous avons, dans un premier temps, choisi une créature tridimensionnelle dont la morphologie est préfixée (i.e. humanoïde) et qui est simulée grâce à l'exploitation du moteur physique ODE. Ce dernier permet d'obtenir une créature avec un niveau élevé du rendu, mais surtout le niveau de réalisme nécessaire à la reproduction du mouvement grâce aux paramètres physiques affectés à cette créature.

Nous avons, par la suite, exploité un réseau de neurones de type récurrent pour contrôler les mouvements de notre humanoïde.

Les mouvements, que le réseau de neurones permet de parfaire sont ensuite transmis vers les descendants d'une population étudiée et cela par l'intermédiaire d'une technique évolutionnaire : les stratégies d'évolution. En effet, les stratégies d'évolution, après avoir effectué leur processus d'évolution et après des centaines de générations, permet, à la créature, de réaliser des mouvements de marche dans un environnement



tridimensionnel possédant des obstacles dans lequel les lois de la physique sont reproduites.

6. Conclusion.

Pour conclure, il y'a lieu de noter que le système a été conçu dans le but de doter un humain virtuel d'une locomotion similaire à une locomotion naturelle et ce dans un environnement complexe. Pour ce faire, nous avons défini un processus nous permettant de simuler une créature tridimensionnelle dans un environnement tridimensionnel contenant certains objets qui sont positionnés sur le sol. Ce processus est tout à fait réaliste est similaire au processus réel car il intègre toutes les caractéristiques physiques et emploie toutes les lois de la dynamique dans le but de reproduire, pour les créatures simulées, des mouvements avec un haut degré de réalisme. Enfin, et conformément au domaine étudié (évolution artificielle) nous avons utilisé les stratégies d'évolution pour garantir le terme évolution.

Dans le chapitre suivant, nous allons discuter et donner les détails de l'implémentation de notre modèle d'évolution ainsi que les résultats obtenus tout en les analysant.



Chapitre IV :

Implémentation et Résultats

Implémentation et résultats

Ce chapitre a pour but de valider expérimentalement, à travers un ensemble de simulations, le modèle proposé dans le chapitre précédent. Nous disposons d'une créature artificielle tridimensionnelle munie d'un contrôleur. Nous faisons évoluer ce contrôleur par les stratégies d'évolution dans le but de converger vers la réalisation de certains mouvements par la créature. Ce contrôleur possède un ensemble de senseurs lui permettant de percevoir le monde et d'en extraire les informations qui sont nécessaires pour agir sur l'environnement complexe dans lequel il évolue.

Nous mettons, également, en exergue dans ce chapitre les résultats de deux expérimentations que nous avons réalisées. La première consistant à faire évoluer le contrôleur de l'humanoïde dans le but de réaliser une simple locomotion d'une créature tridimensionnelle dans un espace plat [ODD+, 2009a]. Quant à la seconde, elle s'intéresse à un deuxième aspect dans la locomotion, celui de l'évitement d'obstacles existants dans l'environnement de simulation [ODD+, 2009b].

1. Le langage de programmation.

Techniquement, le modèle permettant de faire évoluer le contrôleur de l'humanoïde est réalisé par le biais d'une application implémentée dans le langage de programmation C++. Nous avons choisi l'environnement de programmation Visual C++ pour, d'abord la qualité offerte par le C++, telle que la programmation orientée objet et pour l'interface qu'il fournit mais surtout pour sa compatibilité et ses capacités d'interfaçage avec le moteur

physique « ODE 0.11²² ». Le moteur ODE étant destiné, dans notre travail, à la réalisation des mouvements physiques ainsi que des aspects graphiques.

2. Le moteur physique de l'application.

L'intégration d'un moteur physique permet à une application de reproduire des aspects physiques réels, on parle alors de simulation.

Le développement d'un « moteur physique » de l'application permet de concentrer tous les appels à ODE en un module. L'utilisation de ce dernier ne requiert pas, donc, la maîtrise du moteur ODE. Il s'agit, en fait, de réunir dans une classe (ODEInterface) l'ensemble des primitives nécessaires au fonctionnement du moteur ODE.

Ainsi, les objets de l'environnement peuvent être matérialisés par des primitives géométriques existantes sous ODE (GeomBox sous ODE), tous ces objets étant positionnés dans l'environnement de simulation. Il faut, par conséquent, ajouter un plan statique (GeomGround sous ODE) qui constitue le sol de l'environnement. L'espace de collision s'applique sur ces objets, ce qui signifie qu'un corps rigide (la créature où un objet quelconque) ne pourrait en aucun cas les traverser.

Un mouvement physique ne pouvant être continu numériquement, il faut échantillonner une simulation. C'est cette idée qui introduit la notion de pas de simulation : faire un pas de simulation revient à avancer dans le temps la simulation, il faut pour cela définir la taille du pas (généralement 0.05 secondes environ). Plus la taille du pas est petite, plus la simulation sera précise, mais le temps d'exécution sera plus long. Dans notre cas de simulation nous avons choisi un pas de 0,05 secondes.

Idée de fonctionnement.

A chaque pas, un joueur virtuel fournit à la simulation les vitesses angulaires des angles d'inclinaison (alpha et beta), ces derniers sont alors modifiés en conséquence.

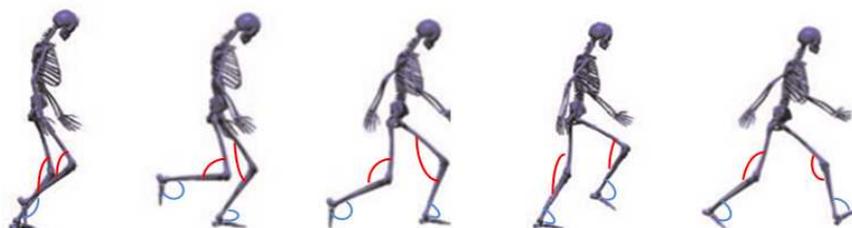


Figure 52: Les angles Alpha en rouge et Beta en bleu.

²² La dernière version du moteur ODE obtenue en 2009.



3. L'architecture globale de l'implémentation.

Le schéma ci-dessous montre les relations entre les différents techniques faisant partie du système global implémenté afin d'accomplir le rôle recherché dans ce travail, ce rôle étant l'évolution du contrôleur d'une créature 3D dans le but de réaliser des mouvements de locomotion dans un environnement physique réaliste.

Comme nous pouvons le remarquer dans le schéma, la créature 3D est un humanoïde dont le système de contrôle réside en son cerveau qui est un assemblage entre le réseau de neurones qui fait l'échange d'informations entre l'environnement physique et le cerveau de la créature, d'une part et les stratégies d'évolution, d'autre part. Ainsi, le génome de chaque individu est construit à partir de ce réseau. L'ensemble des individus formant la population de l'algorithme évolutionnaire se voit subir un processus évolutionnaire itératif pour reproduire le meilleur mouvement à exécuter au niveau du corps de l'humanoïde et plus précisément au niveau de ses pieds.

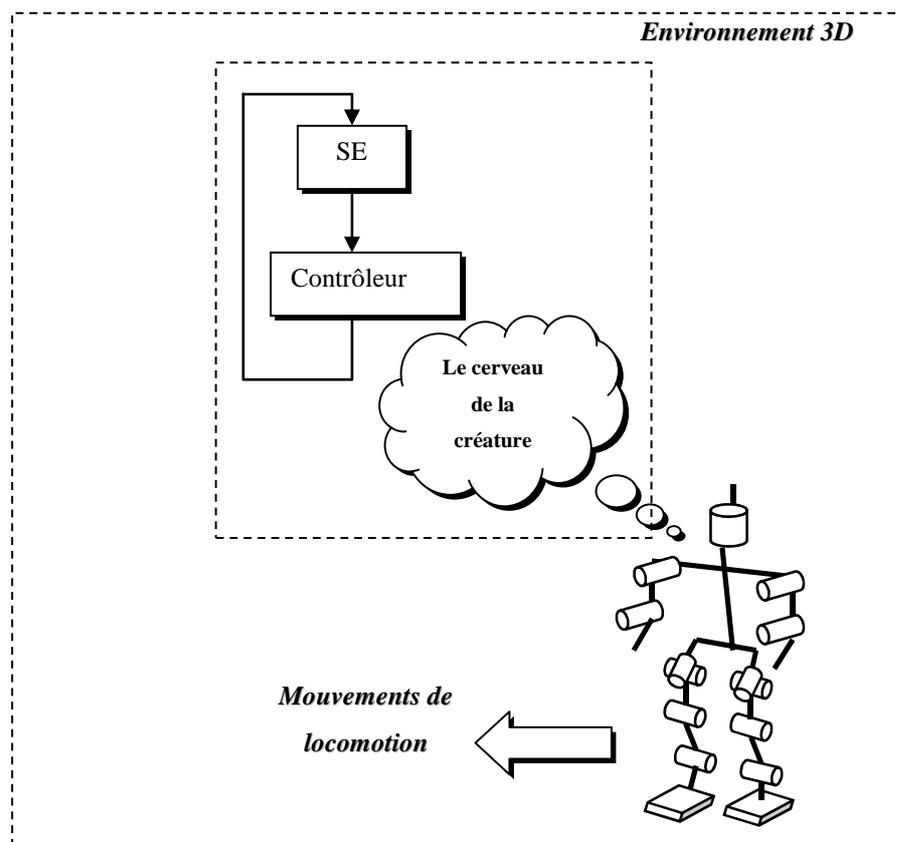


Figure 53 : Architecture globale du système implémenté.



Nous remarquons, ainsi, que le cerveau de l'humanoïde est le module principal qui nous garantit les mouvements de la créature dans son environnement, en application ce module est la « *classe mère* ».

Après avoir simulé la morphologie de la créature par la classe «*human*», des informations sont prises en considération pour coder le génome afin de faire évoluer la population par l'intermédiaire d'un algorithme évolutionnaire. Celui-ci étant également implémenté dans une classe qui est la classe «*Evolutionary Algorithm*». De même, et comme mentionné précédemment, le module cerveau de la créature nous permet de générer le meilleur contrôleur qui reproduit les mouvements recherchés. Ce même contrôleur est implémenté dans la classe nommée «*Reccurent Neural Network*».

De cette manière, notre application est implémentée grâce à un ensemble de classes qui appartiennent à la «*classe mère*». Cette «*classe mère*» fait appel aux différentes fonctions qui existent dans les autres classes telles que la fonction de simulation de la morphologie de la créature ou bien celle de l'algorithme évolutionnaire.

4. Les algorithmes utilisés.

L'application réalisée, dans le cadre de notre projet, est basée sur deux concepts très importants. Le premier étant la simulation graphique tridimensionnelle et physique de la créature et de son environnement (i.e. utilisation d'un simulateur physique qui est ODE) alors que le second aspect est celui de l'utilisation d'une technique évolutionnaire faisant partie du domaine de la vie artificielle.

Les stratégies d'évolution, que nous avons intégrées dans cette application, sont des algorithmes évolutionnaires fonctionnant de pair avec un réseau de neurones, le gène d'un chromosome de cet algorithme représentant le poids d'une interconnexion dans le réseau de neurones.

Ainsi, comme mentionné ci-dessus, l'algorithme évolutionnaire, le réseau de neurones et la morphologie sont des fonctions appartenant chacune à une classe spécifique. Ces fonctions sont exploitées par la fonction principale présentée ci-après, cette fonction est : `simLoop`²³ (voir fonction 1).

²³ La fonction de la boucle de simulation qui appartient à la classe mère, cette fonction permet de gérer la boucle prenant en charge le processus d'évolution.



4.1. La fonction de boucle de simulation.

La fonction 1 représente la boucle de simulation qui garantit le fonctionnement global de toute l'application. Nous remarquons que les fonctions présentées font appel à d'autres fonctions qui sont décrites dans d'autres classes dont nous décrivons les plus importantes. Par ailleurs, il y'a lieu de signaler que nous faisons également appel à des fonctions empruntées à l'environnement ODE.

Fonction 1 : La boucle de simulation

```
// simulation loop
static void simLoop (int pause)
{
    static int timer = 0;
    if (timer < TIMEOUT)
    {
        fonction de contact avec le sol;
        fonction d'obtention des données depuis les senseurs;
        fonction de transferts des données vers les effecteurs ;
        fonction de description du pas de simulation ;
        fonction de dessin de la créature ;

        timer++;
    }
    else
    if(i<POPSIZE)
    {
        fonction de calcul de la fitness d'un individu i ;
        fonction d'affichage de la valeur de la fitness calculée ;
        fonction de lancement de l'individu i sur le simulateur ;
        i++;
        timer = 0;
    }
    else
    {
        création d'une nouvelle population ;
        fonction de l'algorithme évolutionnaire ;
        affichage du max, moyenne et le total
        des fitness de la génération i ;

        for (int i=0;i<population.size();++i)
        {
            Fonction de suppression de la population ancienne ;
        }
        Remplacement de l'ancienne population par la nouvelle ;
        population = newPopulation;
        timer = 0;
    }
} // End of simulation loop
```

Remarque 1: le TIMEOUT et POPSIZE sont des constantes décrites ailleurs.



4.2. La fonction de l'algorithme évolutionnaire.

Concernant l'implémentation de l'algorithme évolutionnaire, celui-ci représente le processus de reproduction des individus présentés et cela dans le but de faire émerger de nouvelles espèces tout en gardant les meilleurs et ce dans le but d'optimiser la recherche de la solution (i.e. trouver le meilleur comportement). Nous avons ainsi décrit, dans la fonction de modification de l'algorithme évolutionnaire (fonction 2), les deux opérations de croisement et de mutation pour l'obtention des fils.

Fonction 2 : modification de l'algorithme évolutionnaire.

```
// Update the Evolutionary Algorithm
vector Genetic::epoch(vector<Organism<float> *> &population)
{
    int max_index = 0;
    TotalFitness = 0;
    AvgFitness = 0;
    MaxFitness = 0;
    m_Generation++;

    for (int i=0;i<population.size();i++)
    {
        double temp_fitness = population[i]->getFitness();
        TotalFitness += temp_fitness;
        if (temp_fitness > m_MaxFitness)
        {
            max_index = i;
            MaxFitness = temp_fitness;
        }
    }
    m_AvgFitness = m_TotalFitness/static_cast<double>(population.size());

    for (int i=0;i<population.size()-m_NumElite;i++)
    {
        fonction de sélection de la roulette pipée pour le parent 1 ;
        fonction de sélection de la roulette pipée pour le parent 2 ;
        fonction de reproduction pour le parent 1 et le parent 2;
    }
    if (m_NumElite > 0)
    {
        for(int j=0;j<m_NumElite;j++)
        {
            fonction pour garder les individus élus;
        }
    }

    for (int i=0;i<offspring.size();++i)
    {
        fonction pour recalculer les positions de la nouvelle population;
    }
}
```



```

    }
    retourner les fils;
} // End of Update the Genetic Algorithm

```

4.3. Un exemple d'une simulation sous ODE.

La troisième fonction présentée dans ce chapitre est celle de la modélisation d'un objet de l'environnement qui est représentée physiquement dans le but de matérialiser la présence d'un obstacle au moment de la navigation de l'humanoïde, cet objet étant une table carrée possédant quatre supports.

Nous avons choisi de représenter cette fonction dans le but d'illustrer le fonctionnement de l'environnement ODE ainsi que la manière dont nous pouvons l'exploiter pour simuler un objet.

Fonction3 : La construction d'un objet 3D.

```

void human<T>::constructObject(const dWorldID world, const dSpaceID
space, const dVector3 position)
{
    static dGeomID G1[3];
    static dGeomID G2[3];
    static dGeomID G3[3];
    static dGeomID G4[3];
    static dGeomID G5[3];

    //DESSINER UNE TABLE 1
    dsSetColor(1,1,0);
    dGeomBoxGetLengths (G1[0],ss);
    dsDrawBox (dGeomGetPosition(G1[0]),dGeomGetRotation(G1[0]),ss);

    dGeomBoxGetLengths (G2[0],ss);
    dsDrawBox (dGeomGetPosition(G2[0]),dGeomGetRotation(G2[0]),ss);

    dGeomBoxGetLengths (G3[0],ss);
    dsDrawBox (dGeomGetPosition(G3[0]),dGeomGetRotation(G3[0]),ss);

    dGeomBoxGetLengths (G4[0],ss);
    dsDrawBox (dGeomGetPosition(G4[0]),dGeomGetRotation(G4[0]),ss);

    dGeomBoxGetLengths (G5[0],ss);
    dsDrawBox (dGeomGetPosition(G5[0]),dGeomGetRotation(G5[0]),ss);

    //HAUT DE LA TABLE
    dMassSetBox (&m,1,3,1.5,0.2);
    dMassAdjust (&m,MM);
    G1[0] = dCreateBox (earth.getSpaceID(),3,2,0.2);
    dGeomSetBody (G1[0],B1[0]);
    //PIED1 DE LA TABLE

```



```

dMassSetBox (&m,1,0.2,0.2,1.3);
dMassAdjust (&m,MM);
G2[0] = dCreateBox (earth.getSpaceID(),0.2,0.2,1.3);
dGeomSetBody (G2[0],B2[0]);

//PIED2 DE LA TABLE
dMassSetBox (&m,1,0.2,0.2,1.5);
dMassAdjust (&m,MM);
G3[0] = dCreateBox (earth.getSpaceID(),0.2,0.2,1.5);
dGeomSetBody (G3[0],B3[0]);

//PIED3 DE LA TABLE
dMassSetBox (&m,1,0.2,0.2,1.5);
dMassAdjust (&m,MM);
G4[0] = dCreateBox (earth.getSpaceID(),0.2,0.2,1.5);
dGeomSetBody (G4[0],B4[0]);

//PIED4 DE LA TABLE
dMassSetBox (&m,1,0.2,0.2,1.5);
dMassAdjust (&m,MM);
G5[0] = dCreateBox (earth.getSpaceID(),0.2,0.2,1.5);
dGeomSetBody (G5[0],B5[0]);

dMatrix3 R,R2;

dGeomSetPosition (G1[0],-7,2,1.5);
dGeomSetPosition (G2[0],-8.4,2.9,0.75);
dGeomSetPosition (G3[0],-8.4,1.1,0.75);
dGeomSetPosition (G4[0],-5.6,2.9,0.75);
dGeomSetPosition (G5[0],-5.6,1.1,0.75);

dGeomSetRotation (G1[0],R);
dGeomSetRotation (G2[0],R);
dGeomSetRotation (G3[0],R);
dGeomSetRotation (G4[0],R);
dGeomSetRotation (G5[0],R);
}

```

5. Les expérimentations²⁴.

Le problème d'évolution des stratégies de locomotion d'une créature tridimensionnelle peut être résolu en utilisant des techniques évolutionnaires. Le but de notre application est de faire évoluer les stratégies locomotives d'un humanoïde dans un environnement physique complexe et réaliste. Pour ce faire, nous avons utilisé les stratégies d'évolution pour optimiser un réseau de neurones récurrent qui est le centre de contrôle de l'humanoïde à faire évoluer.

Les stratégies d'évolution utilisent la fonction de fitness pour faire converger l'évolution du contrôleur de l'humanoïde dans le sens du comportement désiré. La fonction

²⁴ "There is no such thing as a failed experiment, only experiments with unexpected outcomes."



de fitness détermine le meilleur mouvement et est obtenue grâce au score de la fitness la plus élevée (voir les détails dans la section de discussion). Les simulations présentées dans le chapitre 3 sont réalisées avec certaines conditions initiales et certains paramètres nécessaires pour le fonctionnement du processus d'évolution, ces conditions sont citées à la fin de cette section.

Par ailleurs, nous faisons évoluer la stratégie de locomotion par le biais d'un processus qui fait mouvoir le centre d'inertie de la créature, les résultats de cette simulation sont présentés selon deux cas d'étude dans la section suivante avec des images indiquant les résultats obtenus. Plusieurs jours de calculs ont été nécessaires pour que l'algorithme évolutionnaire puisse converger vers une bonne solution permettant de rendre cette simulation particulière possible mais surtout proche du réel. Les calculs ont été effectués sur un processeur Intel quadri-cœurs secondé par une mémoire centrale de 4 Go.

Notre but était de créer une simulation de créatures artificielles qui pourraient évoluer et développer différentes méthodes pour se déplacer tout en se rapprochant graduellement d'un mouvement réaliste proche de la locomotion humaine et ceci sans qu'elles doivent évoluer physiquement. Si le mouvement réalisé par la créature se rapproche le plus de celui d'une locomotion naturelle, nous considérons que nous avons atteint notre but. Dans le cas contraire, nous assignons à la solution une évaluation nulle.

5.1. Mise en œuvre de la simulation.

Les expérimentations reposent sur une simulation d'un environnement physique réaliste tridimensionnel qui contient certains objets considérés comme des obstacles (des tables simples et des cubes). La créature simulée est une forme physique bien spécifique qui est un humanoïde, ce dernier est la seule entité contrôlée dans toute la scène. Ainsi, son contrôleur est l'élément principal à faire évoluer pour garantir la convergence vers les mouvements désirés.

5.1.1. Environnement virtuel.

L'environnement virtuel de simulation est un espace plat simple ne possédant aucune dénivellation. Il s'agit donc d'une surface non bornée sur laquelle nous avons positionné d'une manière fixe certains obstacles et ceci pour étudier le comportement d'évitement.



Cet environnement possède toutes les caractéristiques physiques nécessaires pour réaliser une simulation réaliste.

5.1.2. Créature artificielle.

Du fait que l'environnement simulé possède les caractéristiques physiques nécessaires à la procédure de simulation, la créature artificielle devant s'y mouvoir doit les posséder également. La créature artificielle simulée est un humanoïde possédant 15 degrés de libertés avec un certain nombre de senseurs et d'effecteurs pour interagir avec l'environnement.

Cette créature, comme mentionné dans le chapitre précédent, possède une morphologie complexe. De ce fait, il est difficile de conduire son contrôleur dans l'optique de lui permettre de marcher dans cette environnement sans tomber et ce en exploitant seulement ses deux jambes, ces dernières devant rester simultanément ou alternativement en contact avec le sol. Ainsi, ceci va dépendre essentiellement du fonctionnement de son contrôleur, le mouvement de cette créature étant, en réalité et en définitive, dépendant des angles et de la vitesse des différentes articulations existantes au niveau de son corps.



Figure 54: A gauche l'environnement simulé avec la créature, un obstacle (table) et un mur. A droite l'environnement simulé avec la créature seulement.

5.2. Conditions initiales de l'application.

L'application est réalisée sous certaines conditions qui sont nécessaires au fonctionnement du processus d'évolution. De plus, comme décrit dans le premier chapitre, le domaine de la vie artificielle ne nécessite pas d'intervention de la part de l'utilisateur car toutes les tâches qui ont été réalisées par le programmeur ou ajustées par l'utilisateur sont désormais à la charge et sous la responsabilité du programme (algorithme évolutionnaire). Ainsi, nous



avons défini tous les paramètres ainsi que les valeurs initiales avant le déclenchement de la simulation.

La population simulée est formée de cent individus, le génome de chaque individu étant, dans un premier temps, généré aléatoirement. Par la suite, et à l'issue des générations suivantes, les valeurs des génomes dépendront des meilleurs patrimoines transmis.

L'évaluation des individus d'une population est réalisée en plongeant la créature, avec toutes ses caractéristiques physiques et morphologiques, dans le simulateur pendant un temps fixe que nous avons arrêté à 10 secondes (10,000 pas dans ODE) et avec une fonction de fitness qui est la même que celle utilisée par Karl Sims (distance parcouru). Cette fonction ne changera pas pour la deuxième expérimentation. Cependant, et pour le comportement d'évitement d'obstacle, nous devons réaliser des traitements spécifiques (voir section résultats).

La méthode de sélection utilisée est celle de la roulette pipée : 20% de la population sont élus alors que le reste des individus de la population sont reproduits avec un taux de 70% pour le croisement, d'une part et un taux 10% pour la mutation, d'autre part.

6. Résultats obtenus.

Pour ce qui est des expérimentations, nous nous sommes attelé à évaluer notre système par rapport aux buts que nous nous sommes assignés et qui concernent, en définitive, l'obtention de mouvements selon les deux cas d'études préalablement présentés: la locomotion sur un espace plat ainsi que la locomotion avec évitement d'obstacles.

6.1. Locomotion sur un espace plat.

Dans cette partie, nous présentons les résultats de notre expérimentation, relativement simple, et plus classique et dont le but est d'optimiser le déplacement d'une créature sur un espace plat afin de converger vers un mouvement de locomotion simulant la marche naturelle de l'humain, tout ceci en faisant coopérer un algorithme évolutionnaire et un réseau de neurones récurrent (RNNs).

6.1.1. Analyse des résultats.

Les résultats obtenus, à l'issue des premières générations, étaient loin d'être des mouvements de locomotion car la créature avait comme premier but d'apprendre à se



mettre debout. Si l'individu, au bout de son évaluation, tombe sur le sol, son score de fitness est mis à zéro et l'individu est éliminé de la population.

Ainsi, et après avoir appris à se mettre debout, la créature s'est vue assignée un second but, celui de marcher en se maintenant en équilibre pour engendrer des mouvements de locomotion plus réalistes. Ainsi, ce n'est qu'après quelques centaines de générations que nous avons pu arriver à des mouvements de locomotion plus équilibrés. De plus, les meilleurs individus sont marqués, dans un premier lieu, par leur grande valeur de fitness se traduisant par des mouvements plus réalistes. Cela peut être remarqué dès lors que la créature commence à générer des mouvements de marche sans avoir les deux pieds simultanément posés sur le sol.

En ce qui concerne la direction du mouvement, il est plus difficile d'obtenir une locomotion dans une direction spécifique, celle de départ étant aléatoire (figure 54). Comme présenté dans la figure 48, ce n'est qu'après plusieurs centaines de générations que les créatures les plus performantes finissent par transmettre des variantes de leurs gènes à toute la population. Ceci peut être remarqué dans l'ensemble des valeurs des fitness obtenues par les générations ou bien par une stabilisation dans la moyenne de la fitness de tous les individus d'une population (figure 59). (Voir la figure 55).

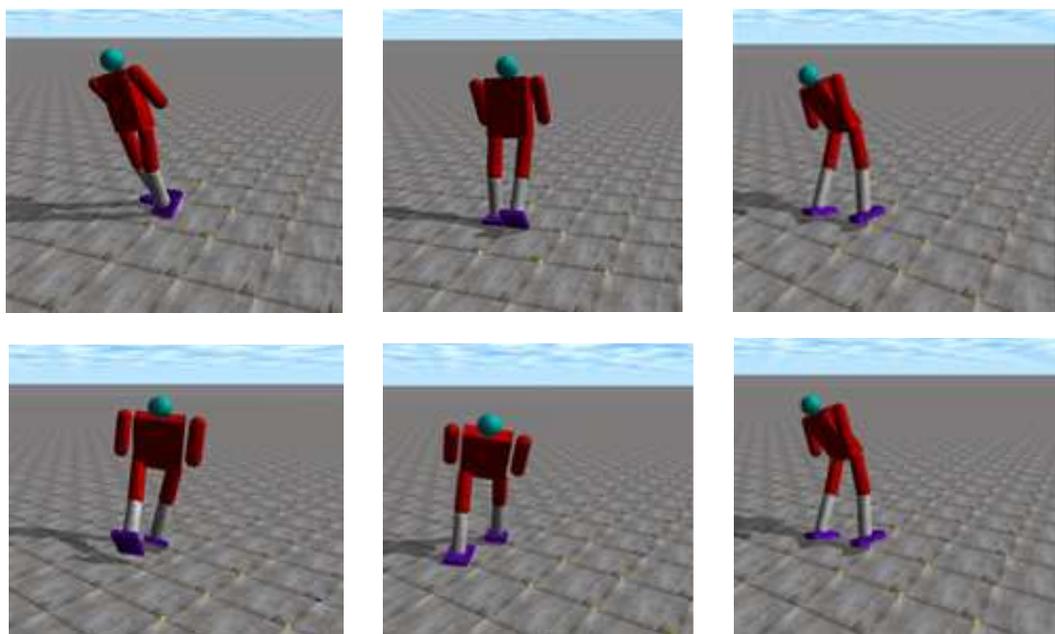


Figure 55: Les résultats moyens de locomotion sur un espace plat.



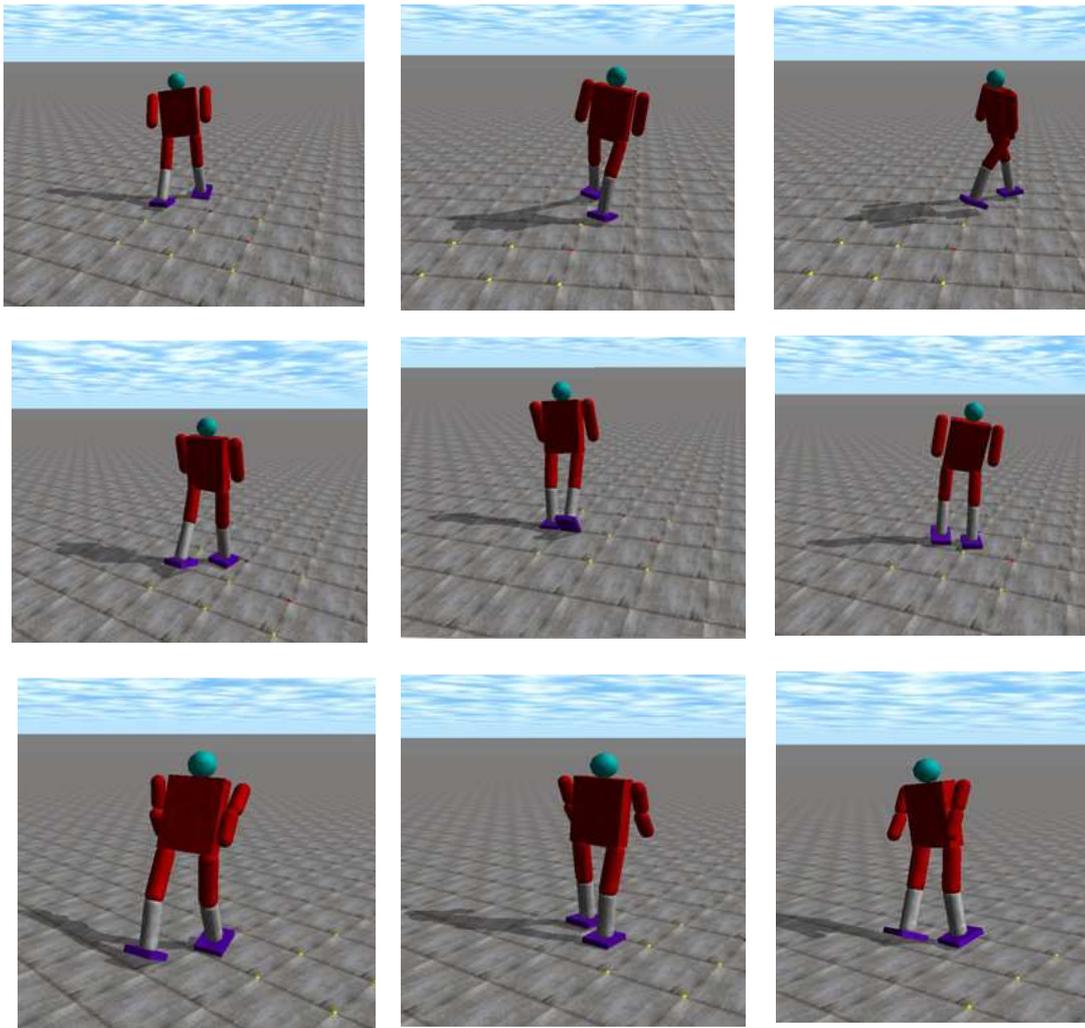


Figure 56 : Les meilleurs résultats de locomotion sur un espace plat

6.2. Une locomotion avec évitement d'obstacles.

En plus de la locomotion, nous avons choisi de faire évoluer le contrôleur de notre humanoïde pour qu'il puisse marcher dans un environnement plat contenant, par ailleurs, un ensemble d'obstacles. La raison d'un tel choix réside dans le fait que, dans la réalité, on rencontre rarement un espace parfait et même s'il est plat, il peut contenir des objets. En définitive, le fait que la créature choisie est un humanoïde implique l'utilisation d'un environnement plus adéquat.

La fonction de fitness utilisée dans ce cas est celle de la distance parcourue mais dans une direction aléatoire (équation ci-après) :



$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Dans certains cas, les mouvements peuvent changer et cela comme suit :

Si les senseurs utilisés ont détecté un obstacle (i.e. $\{s_1, s_2\} > \epsilon$) alors le robot va faire une rotation vers la gauche et vice-versa pour le cas des deux senseurs $\{s_3, s_4\}$ du pied gauche et pour détecter si l'obstacle est devant on effectue le test suivant :

1.

$$\bar{s} = \frac{1}{4} \sum_{i=0}^4 s_i > \epsilon .$$

- s_i est le senseur de 1 jusqu'à 4 car nous utilisons deux senseurs sur chaque pied.
 - ϵ est une valeur trop petite qui indique l'existence d'un objet.
 - \bar{s} est la moyenne des valeurs détectés par les quatre senseurs et qui indique que l'objet obstacle est en face à la créature.
2. S'agissant des résultats obtenus dans le cas de l'évitement d'obstacles, le robot exploite la première simulation (i.e. une simple locomotion) jusqu'à détecter un obstacle. Dans ce cas, le robot pivote pour changer la direction de locomotion et évidemment pour éviter l'obstacle (une table ou un cube), pour poursuivre ensuite son mouvement.

6.2.1. Analyse des résultats.

Les meilleurs résultats obtenus concernent une locomotion humaine avec un comportement d'évitement d'obstacle qui se réalise en effectuant des mouvements de pivotement pour changer la direction de locomotion. Notre créature artificielle (humanoïde) détecte, en apprenant la locomotion, un obstacle (figure 57). En réponse, elle change la direction de locomotion en pivotant son corps pour continuer ensuite à marcher dans la nouvelle direction. Tout à fait naturellement, dans le cas où la créature, en se déplaçant, ne détecte aucun obstacle, elle continue à marcher normalement. Dans le cas où cette même créature détecte un des obstacles existants sur son itinéraire, elle pivotera jusqu'à trouver un passage libre pour l'emprunter.



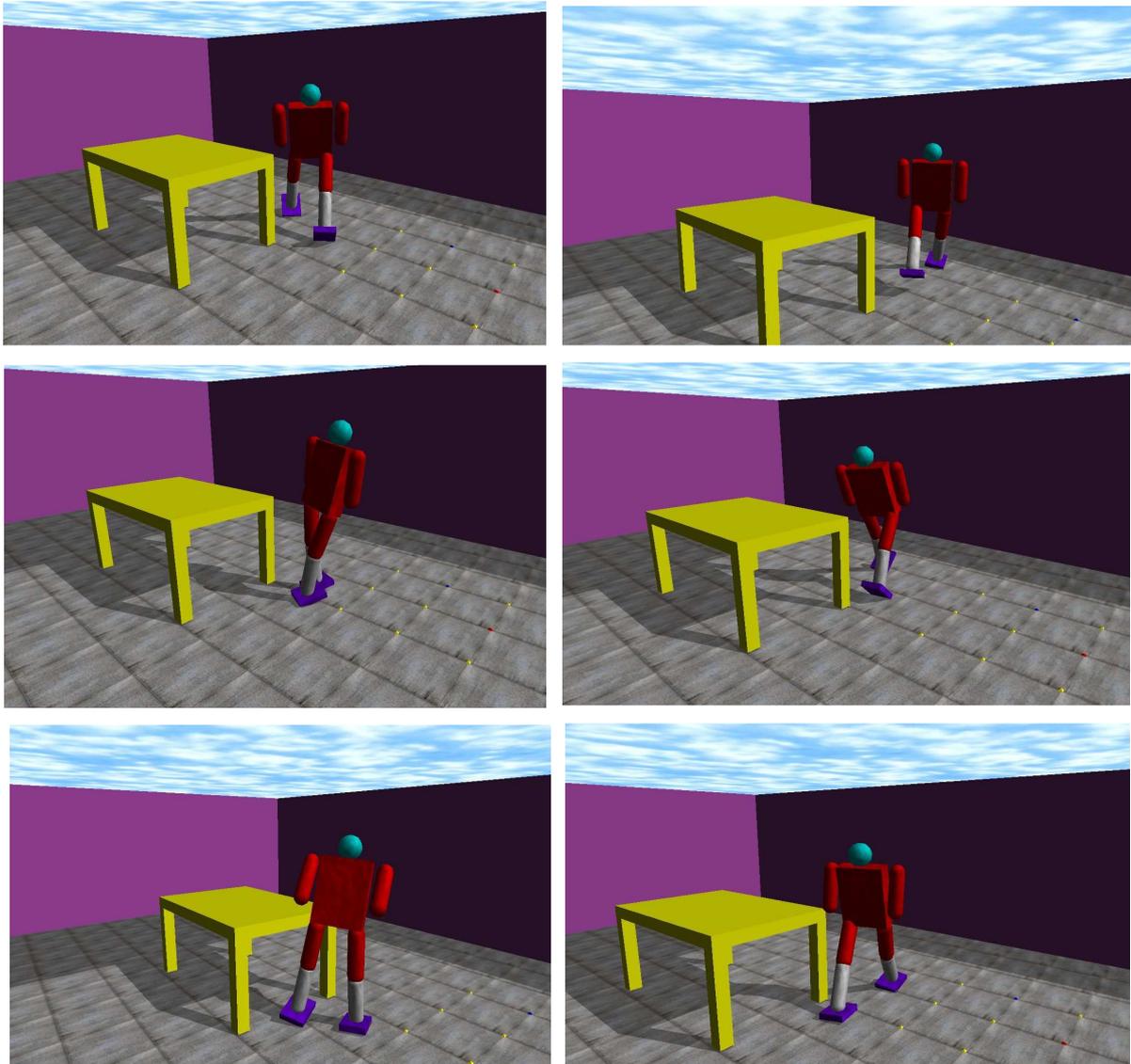


Figure 57: Quelques illustrations de résultats d'évitement d'obstacles (tables).

7. La discussion des résultats.

Dans cette partie, nous analysons les valeurs des fitness obtenues par rapport aux mouvements générés suivant le nombre de générations, et ceci selon les deux cas d'études réalisées.

7.1. Discussion des résultats de locomotion sur un espace plat.

La figure 58 montre les valeurs des fitness moyennes et maximales d'un robot humanoïde après plus de 200 générations. L'exécution est réalisée d'une manière continue pendant



plusieurs jours de simulation sur ordinateur, correspondant à une semaine de calcul environ.

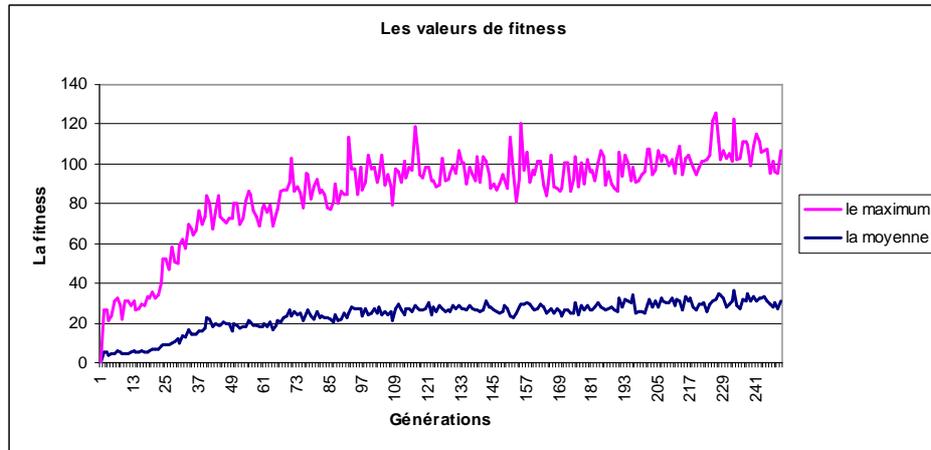


Figure 58: Graphe des valeurs des fitness par rapport au nombre de générations pour une locomotion sur un espace plat simple.

Le génome de la créature utilisée, dans cette expérience, utilise un chromosome qui se compose approximativement de 1130 gènes. Le robot humanoïde apprend avec succès à manœuvrer les bras et les jambes qui sont reliés à son corps par des jointures avec un seul degré de liberté ainsi que 3 degrés de liberté au niveau des hanches, tout ceci dans le but de produire des comportements locomotifs intelligents et efficaces.

Au début de l'évolution, toute la population des robots humanoïdes reproduit des mouvements faibles. Cela est dû à leurs sorties effectrices aléatoires qui auraient comme conséquence que la créature tombe rapidement sur le sol. Après environ une quarantaine de générations, certains de ces individus ont appris comment commander la direction de leur mouvement de locomotion initial en améliorant de ce fait leurs mouvements d'équilibre et de déplacement. Après chaque génération, l'amélioration progressive des capacités locomotives était relativement insignifiante. Cependant, la distance parcourue par les individus, pour chaque génération, augmente progressivement. Dans les 100 générations qui suivent les 40 premières, la population a rapidement amélioré ses points maximum de son corps physique pour manœuvrer ses joints et annexes en appliquant des forces variables aux effecteurs. Après ceci, la population a diminué pendant que le nombre de générations augmentait. Malgré cela, les individus ont rapidement appris à utiliser les deux jambes en les balançant simultanément en avant, en les plaçant sur terre et en les poussant de nouveau vers l'avant du corps. Au bout des 250 générations, les mouvements obtenus



sont bien meilleurs, et la population augmentait désormais presque linéairement avec le nombre de générations.

Les résultats pour cette simulation démontrent, à nouveau, les capacités du réseau de neurones à faire évoluer les solutions d'un problème donné vers des solutions robustes et intelligentes. Nous avons démontré les capacités des RNNs à générer des solutions, dans des temps raisonnables, pour une morphologie complexe.

7.2. Discussion des résultats de locomotion avec évitement d'obstacles.

Les résultats des autres travaux similaires, mais avec des morphologies différentes de celle de notre humanoïde, démontrent que la complexité du génome peut ne pas représenter un facteur important par rapport aux contraintes imposées par la morphologie des créatures tridimensionnelles. Les créatures qui évoluent vers des comportements locomotifs plus complexes exigent une connaissance particulière de l'environnement, telle que la faculté de prévoir la quantité de force exigée pour lever une jambe ou quel intervalle d'angle permettrait d'effectuer un pas normal. Par ailleurs, elles exigent des temps de calcul importants pour faire évoluer leurs capacités de locomotion vers des comportements locomotifs intelligents permettant l'évitement d'obstacles. En définitive, ils nécessitent beaucoup plus de temps à faire évoluer leurs modèles à l'effet d'inclure les mouvements liés à la dynamique de rotation dans le but de bien apprendre à éviter les obstacles détectés.

7.3. Variation d'angles des jointures du robot.

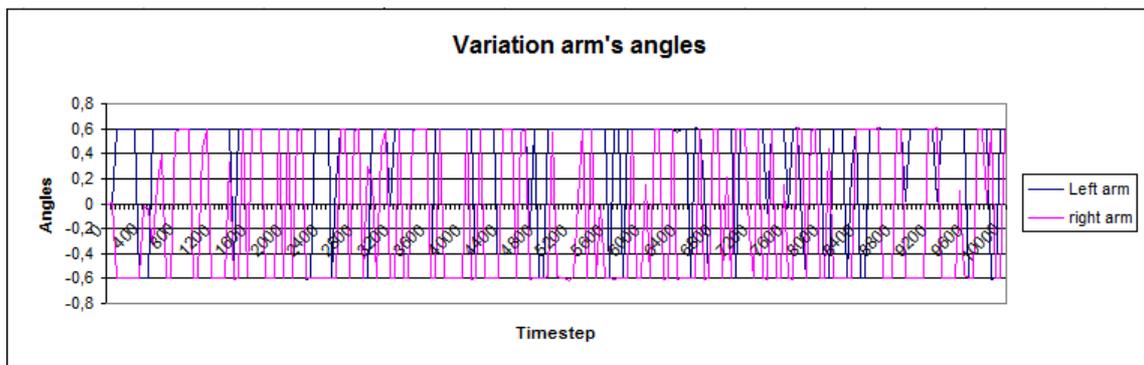
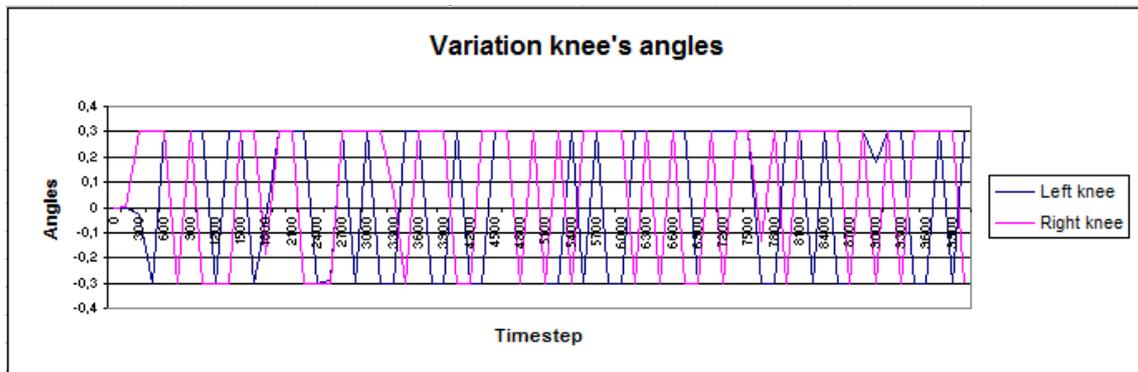
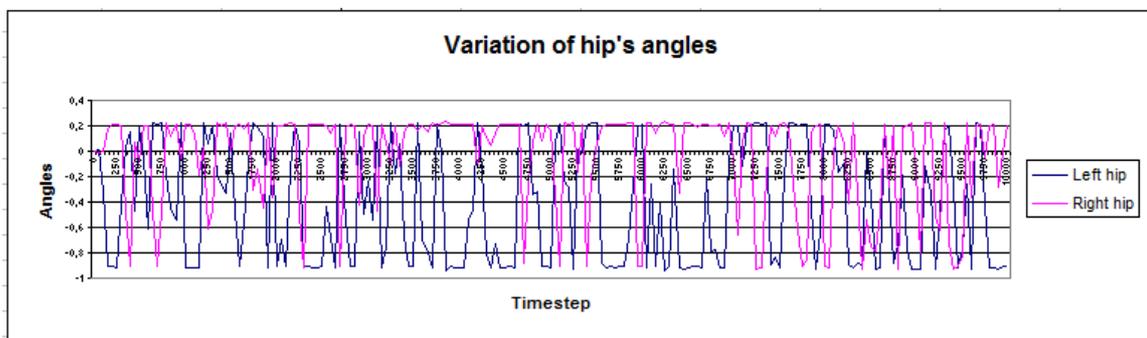
La figure 59 montre les variations d'angles des jointures du robot pendant sa locomotion relativement à une période de simulation de 10 secondes c-à-d 10 000 pas de simulation (time steps) sachant que nous avons choisi un pas de 0,05 secondes (50 time steps). Et pour avoir un équilibre des sorties effectrices du RNN nous avons fixé 50 comme temps de réponse du réseau de neurones RNN.

Les différents graphes de la figure 59 indiquent les transitions des jointures au niveau des chevilles, des genoux, des hanches et des bras respectivement droits et gauches du robot simulé. Ces graphes ont démontré que le robot a réussi à se déplacer en avançant pas à pas mais d'une manière non périodique (i.e. pas simultanément). Ceci est dû au fait



qu'à certains niveaux, le pied droit est positionné suivant son angle maximum mais le gauche est soit sur la valeur zéro ou bien sur la valeur minimale de l'angle. Ceci peut se voir surtout sur le premier graphe des variations des angles des hanches où le robot ne génère pas des mouvements complètement symétriques. Cependant, à partir de la septième seconde de simulation, les mouvements des hanches finissent par être symétriques.

Inversement aux variations des angles des hanches, les autres graphes représentent des alternations symétriques des angles du pied droit et gauche, notamment sur celui des chevilles. En général, nous remarquons sur les quatre graphes que les variations des angles du pied droit représentent dans la quasi-totalité des cas, l'inverse de celle du pied gauche.



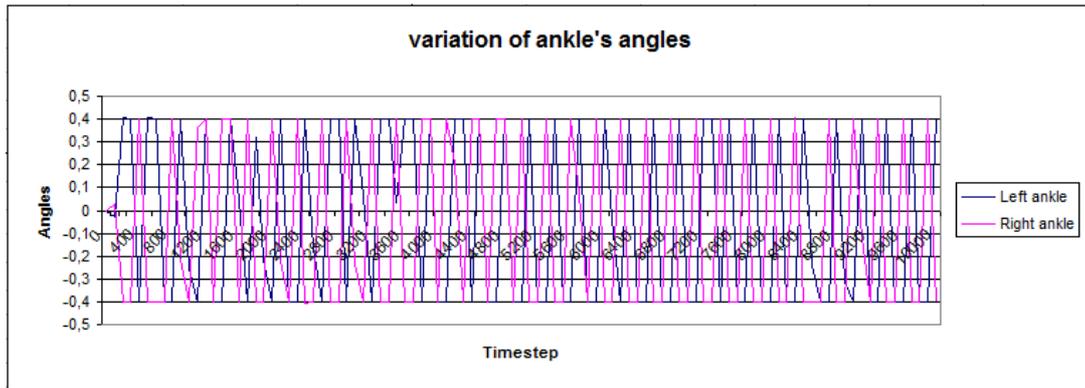


Figure 59 : Graphes de variation d'angles de jointures du robot sur 10 secondes de simulation.

Remarque.

Dans notre approche, dès lors que le comportement que l'on veut faire évoluer est complexe mais surtout nécessite des temps de simulation élevés, nous avons opté pour le choix de déclencher une deuxième exécution en réajustant les paramètres qui y sont liés.

8. Evaluation du modèle.

Le but de notre travail était de développer et de démontrer un cadre pour muner, par évolution, des créatures artificielles, de comportements intelligents réalistes.

L'environnement virtuel est modélisé pour être conforme à tout environnement naturel et ce dans le but d'offrir la possibilité de porter et embarquer les solutions adoptées sur des robots réels. Les résultats des expériences entreprises prouvent que l'évolution des comportements intelligents dans un environnement virtuel complexe contraint par les règles de la physique est faisable. Les créatures simulées ont démontré leur capacité d'apprendre et d'interagir par l'intermédiaire d'un modèle d'évolution artificielle. Le robot humanoïde apprend également à coordonner et chronométrer ses mouvements pour produire des comportements locomotifs efficaces, et de cette manière, il peut apprendre à s'adapter à son environnement. En outre, il peut faire évoluer son comportement à l'effet de reproduire des mouvements de locomotion efficaces sans une quelconque intervention externe. Bien que la morphologie de la créature soit prédéterminée, le robot humanoïde démontre sa robustesse et sa capacité à une meilleure utilisation de sa morphologie pour évoluer vers des comportements locomotifs de plus en plus efficaces.



Par ailleurs, les créatures artificielles font évoluer leur capacité à se servir de leurs modèles en commandant leurs structures physiques par l'application des forces et ce en se servant de leurs effecteurs. Enfin, les résultats obtenus dénotent que les systèmes neuronaux peuvent, dans beaucoup de cas, être une solution viable dans le processus d'évolution de créatures artificielles.

La majorité des travaux réalisés par les chercheurs travaillant dans le même domaine se sont focalisés, d'une manière générale, sur l'évolution des comportements de robot bipèdes moins complexes que les robots humanoïdes (comportement des membres inférieurs) et le peu de travaux traitant les robots humanoïdes n'ont pas abordé des comportements plus complexes que ceux d'une locomotion simple.

Nous concluons, qu'en plus des expérimentations réalisées, il serait plus intéressant d'aborder de nouvelles investigations dans le but de parfaire les mouvements simulés et de générer des comportements plus complexes tels que la montée d'escalier ou l'évitement d'objets mobiles. Ces mouvements seront vraisemblablement réalisés à un niveau microscopique et permettraient aux techniques évolutionnaires d'intervenir à un niveau élémentaire, rendant le processus de réalisation de tels comportements par un robot humanoïde, malgré sa morphologie complexe, plus facilement réalisables qu'avec des techniques classiques, ceci sans aborder la puissance de calcul nécessaire pour rendre cette entreprise réalisable.

8.1. Comparaison avec d'autres modèles.

D'un point de vue analytique, on peut constater qu'un nombre important de travaux sur les créatures artificielles focalisent leurs efforts sur le moteur physique et la morphologie au détriment d'un environnement complexe. Ceci provient certainement du fait que le moteur physique et la représentation de la morphologie sont indispensables à la réalisation de créatures artificielles évolutionnistes.

En ce qui concerne le réalisme des mouvements obtenus, des tests ont été faits sur la qualité des résultats obtenus par des créatures artificielles qui se focalisent sur l'évolution de la morphologie et du contrôleur et qui donnent souvent des résultats plus notables par rapport à celle effectuée par des créatures avec une morphologie figée.

L'ensemble des travaux réalisés sur des créatures avec une morphologie prédéfinie dénotent que la complexité d'une morphologie figée influe négativement sur le réalisme des mouvements obtenus ainsi que sur la complexité des mouvements réalisés par ces



créatures (i.e. comportements plus complexes qu'une locomotion). En d'autres termes, les travaux qui se focalisent sur les créatures bipèdes ou une autre forme plus simple, sont plus faciles à faire évoluer par rapport à celles avec une morphologie d'un humanoïde complet. Ceci est dû à la difficulté d'animer des créatures humanoïdes par des techniques évolutionnaires, l'ensemble des travaux existants dans cet axe sont tous dédiés à l'évolution d'un contrôleur simple pour permettre à une créature de se mouvoir dans un espace simple.

En conclusion à tous ceci, nous pouvons dire que la nouveauté apportée par notre travail consiste en la faculté de faire évoluer un contrôleur dans le but de permettre à un robot humanoïde d'éviter les obstacles.

Ainsi, les points forts de notre investigation sont :

- Le nouveau comportement réalisé par une créature avec une morphologie complexe ;
- L'environnement physique qui ne se résume pas en une simple surface bidimensionnelle mais incorporant un ensemble d'obstacles ;
- La morphologie complexe d'un robot bipède qui doit être contrôlée par une technique évolutionnaire ;
- L'utilisation d'un réseau de neurones qui possède la possibilité d'approcher n'importe quelle fonction mathématique ;
- Pour ce qui est de la convergence lors nos expérimentations, si on s'en tient aux nombres de générations calculées lors d'une évolution, on peut considérer que notre modèle est en accord avec les autres travaux du domaine.

Les points que nous pouvons améliorer, que cela soit à titres de nouveaux travaux ou à titre de nouvelles perspectives peuvent être résumés dans les points suivants :

- La morphologie de nos créatures peut être améliorée en incorporant plus de degrés de liberté et en ajoutant davantage de capteurs comme des gyroscopes ou des accéléromètres ;
- Le mouvement des créatures pourrait être amélioré en prenant en compte l'énergie dépensée lors des déplacements ;
- L'intégration de nouvelles contraintes spécifiques à la robotique et aux environnements réels dans l'optique de prise en compte des solutions générées à l'effet de les exploiter pour la conception de robots réels;#



- Exploitation du parallélisme dans le processus d'évolution dans l'application, notamment par l'exploitation de la puissance des GPUs actuels, intégrant, par ailleurs des moteurs physiques ;
- Intervention au niveau du code du moteur physique ODE pour utiliser une carte physique ;
- Utilisation des techniques d'évolution pour des parties bien précises dans le corps de la créature (faire évoluer deux contrôleurs, un pour chaque pied, par exemple).
- Incorporation dans la fonction de fitness de paramètres qualitatifs de la marche, celle que nous avons utilisée prend en compte l'évaluation de l'aspect qualitatif, uniquement (inclusion de l'aspect morphologique).

9. Conclusion.

Du point de vue, réalisation de l'action de locomotion, notre modèle est conçu pour développer des capacités d'apprentissage d'un robot humanoïde en utilisant un réseau de neurones comme technique d'apprentissage. Du point de vue évolution, les capacités d'apprentissage du comportement locomotif d'un individu sont transférées à ses descendants afin de le faire évoluer vers les meilleurs comportements de locomotion. Le comportement locomotif ainsi développé a été, par la suite, étendu pour réaliser un comportement d'évitement d'obstacles dans un environnement relativement complexe.

Les futurs développements devraient prendre en considération l'exploitation des avancements dans les techniques, certes plus complexes mais surtout plus efficaces, des réseaux de neurones. D'un autre côté, on pourra envisager l'exploration des procédés d'évolution neuronale tout en considérant la possibilité de faire évoluer les entités autonomes vers des comportements toujours plus complexes en incorporant les dimensions coopérative et concurrentielle entre plusieurs entités autonomes.



Conclusion générale

Ce travail s'inscrit dans le domaine de la vie artificielle pour la synthèse de créatures artificielles dans des espaces virtuels. Nos travaux ont porté sur la construction dynamique de comportements par évolution grâce à un système de contrôle.

Durant le 21^{ème} siècle, on s'attend à ce que les robots humanoïdes puissent jouer un rôle de plus en plus important, dans l'industrie mais également dans la vie de tous les jours (robots de ménage, ...). La concrétisation d'une telle situation passe par une complexification de ces mêmes robots qui constituera à son tour un défi formidable à lever par les chercheurs et les laboratoires de recherche. À mesure que la complexité des robots humanoïde augmente, il y aura un besoin fort de représentations flexibles et souples pour des comportements moteurs [Mit, 1995]. En plus d'une représentation flexible, une méthode efficace d'optimisation pour produire des comportements robustes réside dans l'utilisation des systèmes évolutionnaires bio-inspirés.

Le premier objectif de cette étude était de développer un système de contrôle via une évolution artificielle pour reproduire et simuler la marche humaine. Plusieurs techniques existantes ont été intégrées pour produire un outil flexible de synthèse de mouvements pour la réalisation des buts recherchés. La technique que nous avons choisie pour contrôler les mouvements de nos créatures artificielles consiste en un système connexionniste représenté par un réseau de neurones, les réseaux de neurones étant les systèmes les plus utilisés dans ce domaine.

Les possibilités de la marche sur un terrain inégal ou dans des environnements encombrés d'obstacles sont fondamentales dans notre vie quotidienne (escaliers, par exemple). Jusqu'ici, seuls quelques systèmes étaient capables de simuler la marche humaine sur des terrains accidentés (non plats). Cependant, la plupart de ces systèmes exigeaient davantage d'intervention de la part de l'utilisateur, qui peut constituer un travail pénible, et ne pouvaient reproduire habituellement la marche continue sur un terrain inégal d'une manière interactive. Notre contribution au niveau de l'évolution d'un comportement

adaptatif à de tels environnements a pu être réalisée sans aucune intervention extérieure et ce en exploitant des techniques totalement évolutionnistes.

Le grand intérêt porté sur les robots humanoïdes ces derniers temps est motivé par les nombreux avantages véhiculés par les robots bipèdes par rapport aux robots roulants. Tout d'abord, les robots humanoïdes (et les robots bipède en général) peuvent se déplacer dans des environnements très souvent inaccessibles aux robots roulants, tels que les cages d'escalier et les terrains extérieurs accidentés. En outre, leur forme, empruntée de celle de l'être humain, permet à de tels robots de fonctionner dans des environnements construits, tels que les maisons ou les manufactures qui sont, naturellement, adaptées aux êtres humains. Cet intérêt nous a encouragés et motivés dans notre choix d'une créature artificielle ayant une morphologie ressemblant à celle des humains (humanoïde). Par ailleurs, les robots humanoïdes (réels) sont à l'origine des études et recherches dans le domaine des créatures humanoïdes (virtuelles) et ce pour des raisons de simulation, principalement.

Ainsi, les mouvements de la créature sont réalisés grâce à un contrôleur constitué d'un réseau de neurones que nous avons fait évoluer par le biais d'un algorithme évolutionnaire. Le processus d'évolution est appliqué sur la structure des poids du réseau de neurones (RNN) qui sera traduite via des effecteurs à des angles dans les jointures de la créature. Le déplacement réalisé par ce contrôleur et l'algorithme évolutionnaire, est assuré par une fonction de fitness qui a pour rôle de faire converger le processus d'évolution vers le comportement recherché, cette fonction de fitness étant la distance parcourue au bout d'un temps fixé.

Les résultats obtenus par évolution concernent, dans un premier temps une locomotion simple sur un support plat. Par la suite, et en modifiant l'environnement en lui rajoutant quelques obstacles, nous avons pu simuler un comportement de locomotion avec évitement d'obstacles.

Les résultats que nous avons obtenus offrent de nombreuses perspectives pour les nouveaux types d'environnements qui conduisent à de nouveaux comportements (assimilation de la troisième dimension pour les environnements accidentés, ...). D'autres perspectives sont à envisager en matière de possibilité de description de la morphologie et du contrôleur de cette créature.

Perspectives

Dans cette partie du mémoire nous présentons quelques perspectives au sujet de travaux sur les créatures artificielles et plus précisément les perspectives sur des créatures humanoïdes. Nous commençons par identifier la perspective la plus importante qui est la réalisation du processus d'évolution des comportements intelligents d'un robot humanoïde par un algorithme évolutionnaire, sous la forme d'une machine qui sera réalisée physiquement et en utilisant un ensemble de senseurs et un 'cerveau' embarqué qui aura le rôle de contrôle de cette machine.

1. Morphologie.

Un des problèmes complexes et difficile à apprivoiser, lié au processus du mouvement, réside dans le fait que l'être humain possède, naturellement, plus de 200 degrés de liberté (DOF), même pour les figures humaines les plus simples, l'être humains en utilise une combinaison de plus de 30 degrés de liberté. Le contrôle de ces degrés de liberté, pour exprimer un mouvement désiré, représente en même temps un défi et une perspective de recherche très intéressants mais surtout utiles dans le domaine des humanoïdes.

Le contrôle d'un humanoïde avec plusieurs degrés de liberté implique l'utilisation de plus de neurones dans le réseau de neurone utilisé comme contrôleur. Le nombre de neurones augmentant, il aura pour effet de générer une contrainte difficile à contrôler surtout en termes de puissance de calcul, de tels problèmes peuvent être réglés par l'ensemble de perspectives présentées dans la section suivante.

2. Réseau de neurones.

Actuellement, le réseau de neurones artificiel récurrent utilise des poids évolutifs. Ainsi, la structure d'un tel réseau est fixe. Une structure évolutive d'un réseau de neurones peut fournir certains avantages tels que la convergence qui devrait être plus rapide pour la

génération d'une solution. Les algorithmes neuro-évolutionnaires, avec lesquels on commence par un réseau de structure minimal tout en complexifiant la structure progressivement selon le besoin, peuvent également remédier à la méthode d'essais-erreur qui détermine le nombre correct de nœuds dans les couches cachées et de contexte, permettant de commencer la simulation. Le réseau NEAT de Kenneth Stanley (2002) peut être utile pour un cas pareil.

Malgré le développement qui peut être réalisé en utilisant des réseaux de neurones évolutifs (tel que NEAT), le niveau de complexité d'un robot humanoïde avec plus de 30 degrés de liberté peut toujours poser un problème au niveau du contrôle des mouvements de tous le corps d'une telle créature. Pour cela, nous proposons une autre perspective au niveau de l'algorithme évolutionnaire.

3. Algorithme évolutionnaire.

L'algorithme évolutionnaire, dans un futur travail, peut calculer les paramètres pas pour toutes les jointures de la créature mais seulement pour une partie de son corps. Les jointures restantes peuvent être manipulées par l'utilisateur avec un outil. De cette façon, par exemple, l'algorithme évolutionnaire peut déplacer les jambes pendant la marche, la partie supérieure du corps humain peut être commandée complètement ou partiellement par l'utilisateur. Ce type d'interaction peut être également utile pour placer approximativement les paramètres des jointures qui seront raffinées avec l'algorithme évolutionnaire. Un dispositif intéressant, pour l'utilisateur, permettrait également de modifier, manuellement, une valeur du gène, pendant l'évolution. Les indications suggérées pour les travaux futurs peuvent aider l'algorithme évolutionnaire et permettre un meilleur contrôle des mouvements d'une créature humanoïde ou de n'importe quelle autre créature.

4. Environnement.

L'environnement dans lequel l'évolution se produit est extrêmement influant pour les résultats finaux. L'environnement virtuel place les mêmes contraintes sur des créatures simulées possédant une morphologie que celles de l'environnement normal (naturel). Il est probable, dans n'importe quelle réalisation physique, que les créatures artificielles et plus



précisément les créatures humanoïdes comme dans l'environnement naturel, rencontrent des obstacles qui sont, non seulement fixes, mais surtout mobiles. Afin de permettre à ces créatures d'apprendre à faire face aux obstacles, le cerveau de telles créatures doit être capable d'évoluer lui même pour produire des comportements plus adaptatifs

Une autre considération peut diriger les créatures à suivre un chemin sans le vouloir : un exemple pour une telle idée consiste à utiliser de petits couloirs avec une série de murs que la créature devrait parcourir, et cela seulement car elles sont obligées d'éviter l'ensemble de murs (i.e. obstacles) mais, d'une manière progressive, le comportement d'évitement pourra les conduire vers un nouveau comportement qui est le suivi d'un chemin pour trouver une sortie, par exemple.

Un autre type de contraintes, très présentes dans les environnements réels, sont les environnements inégaux ou accidentés. La locomotion dans de tels milieux est très difficile à réaliser surtout avec une créature humanoïde. Ainsi, il serait souhaitable de rajouter un ensemble de gyroscopes pour permettre à la créature de bien prévenir les lieux difficiles.

4.1. Ecosystème.

Des terrains accidentés ou des environnements encombrés d'obstacles fixes ou mobiles peuvent représenter une forte contrainte pour la naissance de comportements très délicats dans le domaine de la robotique avec une forme d'humanoïde. Le niveau de complexité de ces environnements peut être une raison pour les considérer comme des écosystèmes mais cela va dépendre de la nature des créatures choisies et ainsi de leurs comportements. Prenons, par exemple, une créature qui se développe à partir d'une cellule œuf par un processus évolutif qui s'appelle l'embryogenèse. Cette créature, une fois arrivée à former son corps matériel, va poursuivre son processus de vie pour se nourrir par d'autres créatures qui existent dans son environnement ou par d'autres type de nourritures. Cette nourriture sera transformée sous la forme d'énergie qui va lui permettre de se déplacer dans l'environnement, soit pour rechercher encore de la nourriture ou bien pour accomplir des tâches qui lui sont destinées telles que des comportements de poursuite ou d'attaque.

4.2. Ecosystème semi-évolucionnaire.

Une idée que l'on pourrait certainement développer dans le contexte des écosystèmes, est de choisir une créature qui construit sa morphologie conformément à son environnement.



Son passage, d'un milieu à un autre, induirait un changement de morphologie et de comportement.

Nous pouvons envisager, également, à peupler des écosystèmes par plusieurs entités qui peuvent s'attaquer entre elles. Ainsi, pour mieux diriger l'évolution de l'ensemble de ces comportements, nous pouvons offrir à l'utilisateur la possibilité d'intervenir, soit via un ensemble de paramètres à manipuler ou carrément d'agir sur cet environnement à travers son avatar.

Toutes ces idées, mais surtout celle de l'intervention extérieure de l'utilisateur dans le processus d'évolution, peuvent nous conduire à un aspect de jeux vidéo qui est dirigé par un processus évolutionnaire.

4.3. Jeu évolutionnaire.

L'intervention de l'utilisateur dans le processus évolutionnaire peut être une meilleure combinaison entre deux aspects qui sont contradictoires dans leur fonctionnement. Tout d'abord, l'aspect utilisateur qui est un aspect très réputé dans les simulations déterministes et aussi dans les jeux vidéo où l'utilisateur est le guide et est l'élément principal dans leur fonctionnement. Pour ce qui est de l'aspect évolutionnaire, cela se traduit par tout ce qui est imitation de la nature et de la vie.

La combinaison de ces deux aspects peut être effectuée pour plusieurs raisons. D'abord l'utilisateur peut assister un processus d'évolution comme il peut en déclencher un. C'est cette notion qu'on qualifie de combinaison de deux aspects contradictoires.

L'idée consiste à développer des jeux vidéo utilisant des techniques évolutionnistes. Ainsi, la réponse à une action ne se faisant plus d'une manière déterministe, mais d'une manière intelligente via un processus évolutionnaire et ne sera donc plus prédéfinie. Le joueur aura l'impression, à chaque fois qu'il joue à ce jeu, qu'il le fait pour la première fois. Si ceci serait réalisable, il ne pourrait l'être sans prendre en charge des problèmes de fonctionnement du processus évolutionnaire, principalement sur la question des temps de réponse.



Bibliographie

- [AK, 2005] Andrew Adamatzky and Maciej Komosinski. Artificial Life Models in Software. Springer-Verlag, 2005.
- [Bae, 1997] T. Baeck, D. B. Fogel, et Z. Michalewicz. Handbook of Evolutionary Computation. Institute of Physics Publishing, 1997.
- [Bar, 1957] Nils Aall Baricelli, Symbiogenetic Evolution Processes realized by Artificial Methods, *Methodos*, 9, p. 143-182, 1957.
- [Bar, 1962] Nils Aall Baricelli, Numerical Testing of Evolution Theories, Part II Preliminary Tests of Performance, Symbiogenesis and terrestrial life, *Acta Biotheoretica*, XVI, p.99-126, 1962.
- [Bed, 1996] Bedau, M. (1996). The nature of life. In Boden, M., editor, *The Philosophy of Artificial Life*. Oxford University Press, Oxford.
- [Ber, 2001] Alain Berro thèse « Optimisation multiobjectif et stratégies d'évolution en environnement dynamique », 18 décembre 2001
- [BL, 2004a] J. C. Bongard and Hod Lipson. Integrated design, deployment and inference for robot ecologies. In *Proceedings of Robosphere 2004*.
- [BL, 2004b] J. C. Bongard and Hod Lipson. Once more unto the breach, automated tuning of robot simulation using an inverse evolutionary algorithm. In *Proceedings of the Ninth Int. Conference on Artificial Life (ALIFEIX)*, 2004.
- [Bor, 2008] Stefan BORNHOFEN THESE « Emergence de dynamiques évolutives dans une approche multi-agents de plantes virtuelles », 30 septembre 2008
- [BZL, 2006] Josh Bongard, Victor Zykov, Hod Lipson Resilient Machines Through Continuous Self-Modeling 1,217 November 2006 VOL 314 SCIENCE www.sciencemag.org p: 1118-1121

- [Cas, 1903] W. E. Castle. Mendel's law of heridity. *Science*, 18(456):396–406, 1903.
- [Cap, 1998] Capus L., Tourigny N., "Road Safety Analysis: A Case-based Reasoning Approach", Transportation Research Board - 77th Annual Meeting (TRB'98), Washington, USA, Janvier 1998.
- [CD, 2007] CHAVOYA A., DUTHEN Y.: Evolving an artificial regulatory network for 2d cell patterning. *Proceedings of the 2007 IEEE Symposium on Artificial Life (2007)*, 47–53.
- [CEA, 2007] Nicolas Chaumont, Richard Egli, and Christoph Adami. Evolving virtual creatures and catapults. *Artificial Life*, 13(2):159–187, 2007.
- [CF, 2006] Maurizio Conventi¹, Marco Fratarcangeli^{1, 2}. Title: "Physically-based Animations of 3D Biped Characters Using Genetic Algorithms".
¹University of Rome "La Sapienza", Department of Computer and Systems Science, Italy et ²Linköping Institute of Technology, Department of Electrical Engineering, Sweden.
- [CM, 1995] Dave Cliff & Geoffrey F. Miller. Tracking the redqueen: Measurements of adaptive progress in co-evolutionary simulations. In Berlin Springer-Verlag (ed.), *Proceedings of the Third European Conference on Artificial Life*, pp. 200–218 (1995).
- [CM, 1996] Dave Cliff & Geoffrey F. Miller. Co-evolution of pursuit and evasion ii: Simulation methods and results. In USA MIT Press, Cambridge (ed.), *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, pp. 506–514 (1996).
- [CW, 1953] F. Crick et J. Watson. A Structure for Deoxyribose Nucleic Acid. *Nature*, 171 (737-738), 1953.
- [Col, 1992] Collins R. J., "Studies in Artificial Evolution.", Artificial Life laboratory Phd, University of California. Los Angeles, 1992.
- [Dar, 1859] Charles Darwin, *The Origin of Species*, 1859.
- [Dav, 2006] Davidson, E. H. (2006). *The regulatory genome: gene regulatory networks in development and evolution*. Academic Press.

- [Deb, 2001] K. Deb. Multi-objectives optimization using evolutionary algorithms. Wiley, 2001.
- [Dik, 1993] Dike B. A., Smith R. E., “Application of genetic algorithms to Air Combat Maneuvring”, TCGA Report No. 93002, August 1993.
- [Dir, 1997] Dirk Arnold. Evolution of legged locomotion. Master’s thesis, School of Computing Science, 1997.
- [dJo, 1975] De Jong K.A. “An analysis of the behaviour of a class of genetic adaptative systems” PH. D. thesis, Department of Computer and Communication Sciences, University of Michigan. 1975.
- [dJo, 2002] K. A. de Jong. Evolutionary Computation. The MIT Press, 2002.
- [dG, 1999] DE GARIS H.: Artificial embryology and cellular differentiation. In Evolutionary Design by Computers (1999), Peter J. Bentley e., (Ed.), pp. 281–295.
- [Dou, 2006] René Doursat. The growing canvas of biological development: multiscale pattern generation on an expanding lattice of gene regulatory networks. InterJournal Complex Systems, 2006.
- [Dou, 2007a] René Doursat. Organic Computing, chapter Organically Grown Architectures: Creating Decentralized, Autonomous Systems by Embryomorphic Engineering. Springer-Verlag, 2007.
- [Dou, 2007b] René Doursat. The self-made puzzle: Integrating self-assembly and pattern Formation under non-randomgenetic regulation. In 7th International Conference on Complex Systems, 2007.
- [ED, 2006] Malachy Eaton · Thomas J. Davitt, Title: “Evolutionary control of bipedal locomotion in a high degree-of-freedom humanoid robot: first steps. In Artif Life Robotics (2007) 11:112–115 DOI 10.1007/s10015-006-0411-y, July 22, 2006.
- [FB, 1989] D. F. Farmer and A. Belin. Artificial life: the coming evolution. In Artificial Life, pages815–840, 1989.

- [Fle, 1996] Kurt Fleischer. Investigations with a multicellular developmental model. In *Artificial Life V*, pages 229–236, 1996.
- [Fra, 1962] A. S. Fraser, Simulation of Genetic Systems, *Journal of theoretical biology*, 2, p. 329-364, 1962.
- [GH, 1997] Larry Gritz James K. Hahn Genetic Programming Evolution of Controllers for 3-D Character Animation Appeared in: Koza, J.R., et al. (editors), *Genetic Programming 1997: Proceedings of the 2nd Annual Conference*, pp. 139-146, July 13-16 1997, Stanford University. San Francisco: Morgan Kaufmann. 1997.
- [GLT, 1997] F. Glover, M. Laguna, et al. *Tabusearch*. Springer, 1997.
- [GM, 1997] Faustino Gomez and Risto Miikkulainen. Incremental evolution of complex general behavior. *Adaptive Behavior*, 5:317–342, 1997.
- [Gol, 1989] David E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, Massachusetts, 1989.
- [Gru, 1994] Gruau F. “Neural network synthesis using cellular encoding and the genetic algorithm”. Thèse de Doctorat, ENS Lyon-Université de Lyon I. 1994.
- [Han, 1992] P. J. B. Hancock. Genetic algorithms and permutation problems: a comparison of recombination operators for neural net structure specification. In D. Whitley, editor, *Procs of COGANN workshop, IJCNN, Baltimore. IEEE*, 1992.
- [Hay, 1999] HAYKIN, S., *Neural Networks: A comprehensive foundation*. Upper Saddle River, New Jersey: Prentice Hall, 2nd edition, 1999.
- [Har, 1992] Hart, J., “The Object Instancing Paradigm for Linear Fractal Modeling,” *Graphics Interface*, 1992, pp.224-231.
- [Heu, 1994] Livre : HEUDIN Jean Claude, « **La vie artificielle** », Editions Hermès, Paris, 1994
- [Hof, 1999] HOFSTADTER D., *Gödel, Escher, Bach*, Dunod, Paris, 1999.

- [Har, 1992] Harvey I., Husbands P., Cliff D. "Issues in evolutionary robotics" Cognitive Science Research Papers University of Sussex, July 1992.
- [Hol, 1975] John Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Harbor, 1975.
- [Hol, 1980] Holland J. H., "Adaptive Algorithms for Discovering and Using General Patterns in Growing Knowledge Bases", *International Journal for Policy Analysis and Informations Systems*, Vol 4, No 3, pp 245-268, 1980.
- [Hol, 1986] Holland, J. H., "Escaping Brittleness: The Possibilities of General-Purpose Learning Algorithms Applied to Parallel Rule-Based Systems", *Machine Learning: An Artificial Intelligence Approach, Volume II*, Morgan Kaufman Publishers, Los Altos, 1986.
- [Heb, 1949] Donald Hebb. *The Organisation Of Behavior*. Wiley (1949).
- [Heg, 2003] Olivier Heguy. *Architecture comportementale pour l'´emergence d'activités coopératives en environnement virtuel*. Ph.D. thesis, Université Paul Sabatier, Toulouse (2003).
- [IHW, 1999a] A.J. Ijspeert, J. Hallam, and D. Willshaw. Evolving swimming controllers for a simulated lamprey with inspiration from neurobiology. *Adaptive Behavior*, 7(2):151–172, 1999.
- [IHW, 1999b] A. J. Ijspeert, J. Hallam, and D. Willshaw. From lampreys to salamanders: evolving neural controllers for swimming and walking. In Pfeifer et al. [140], pages 390–399.
- [Ijs, 2000] Auke Jan Ijspeert. A 3d biomechanical model of the salamander. In *Virtual Worlds*, pages 225–234, 2000.
- [Jac, 1996] Jacob, C. (1996a). Evolution programs evolved. In Voigt, H., Ebeling, W., Rechenberg, I., and Schwefel, H., editors, *PPSN IV, Lecture Notes in Computer Science 1141*. Springer, Berlin.
- [Katic, 2003] D. Katic & M. Vukobratovic. Survey of intelligent control techniques for

humanoid robots. *Journal of Intelligent and Robotic Systems*, vol. 37, No. 2, page 117-141, 2003.

- [KGV, 1983] S. Kirkpatrick, C. D. Jr. Gelatt, et M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.
- [Kit, 1990] Kitano, H., “Designing neural networks using genetic algorithms with graph generation system”, *Complex Systems*, Vol.4, pp.461-476, 1990.
- [KK, 2001] Maciej Komosinski and Marek Kubiak. Taxonomy in A life. Measures of similarity for complex artificial organisms. In *Advances in Artificial Life. Lecture Notes in Artificial Intelligence 2159*, pages 685–694. Springer-Verlag, 2001.
- [KKK, 2001] Maciej Komosinski, Grzegorz Koczyk, and Marek Kubiak. On estimating similarity of artificial and real organisms. *Theory in Biosciences*, 120(34):271–286, 2001.
- [Kle, 2003] Jon Klein. Breve: a 3d environment for the simulation of decentralized systems and artificial life. In *ICAL2003: Proceedings of the eighth international conference on Artificial life*, pages 329–334, Cambridge, MA, USA, 2003. MIT Press.
- [Kom, 2000] Maciej Komosinski. The world of framsticks: Simulation, evolution, interaction. In *VW’00: Proceedings of the Second International Conference on Virtual Worlds*, pages 214–224, London, UK, 2000. Springer-Verlag.
- [Kom, 2003] Maciej Komosinski. The Framsticks system: versatile simulator of 3D agents and their evolution. *Kybernetes: The International Journal of Systems & Cybernetics*, 32:156–173, 2003.
- [Kom, 2005] Maciej Komosinsk. Framsticks: a platform for modeling, simulating and Evolving 3D creatures. In *Artificial Life Models in Software*, chapter 2, pages 37–66. Springer-Verlag, 2005.
- [Koz, 1992] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.

- [Koz, 1994] J.R. Kosa. *Genetique programming II :Automatic Discovery of reusable Programs* MIT Press, 1994.
- [KRV, 2001] Maciej Komosinski and Adam Rotaru-Varga. Comparison of different genotype encodings for simulated 3D agents. *Artificial Life Journal*, 7(4):395– 418, 2001.
- [KU, 1999] Maciej Komosinski and Szymon Ulatowski. *Framsticks: towards a simulation of a nature-like world, creatures and evolution*. In *Advances in Artificial Life. Lecture Notes in Artificial Intelligence 1674*, pages 261–265. Springer-Verlag, 1999.
- [LAM, 1809] Lamarck, J.B. *Philosophie Zoologique* (1809) trans. by Hugh Elliot as *Zoological Philosophy: An Exposition with Regard to the Natural History of Animals with introductory essays by David L. Hull and Richard W. Burkhardt Jr.*, Chicago, 1984.
- [Lan, 1986] LANGTON C.G., « *Studying Artificial Life with Cellular Automata* », *Physica D*, 22, p. 120-147, 1986.
- [Lan, 1989] LANGTON C.G., *Artificial Life I*, Addison-Wesley, 1989.
- [Lan, 1998] Chris Langton. *Artificial life*. In *Artificial Life*, pages 1–47, 1998.
- [Las, 2004] LASSOUAOUI Nadia, HAMAMI Latifa, NOUALI Nadia « *les algorithmes génétiques application à la la segmentation d’images* », *RIST* Vol.14 N°02, 2004.
- [Les, 1999] LESEUL Benoît 6eF, Année 1998-1999, *DE L’INTELLIGENCE ARTIFICIELLE A LA VIE ARTIFICIELLE*, Ecole Notre-Dame des Champs, rue Edith Cavell 143, 1180 Bruxelles.
- [LHL, 1996] HL Wei-Po Lee, John Hallam, and Henrik Hautop Lund. A hybrid gp/ga approach for co-evolving controllers and robot bodies to achieve fitness specified tasks. In *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation*, pages 84–389. IEEE Press, 1996.

- [Lin, 1968] Lindenmayer, A., "Mathematical Models for Cellular Interactions in Development, Parts I and II," *Journal of Theoretical Biology*, Vol.18, 1968, pp.280-315.
- [Lip, 2005] Hod Lipson. *Evolutionary robotics and open-ended design automation*, 2005.
- [LLD, 2007] LASSABE N., LUGA H., DUTHEN Y.: A New Step for Evolving Creatures. In *IEEE-ALife'07* (April 2007), IEEE, pp.243–251.
- [LLD, 2008] LASSABE N., LUGA H., DUTHEN Y.: « Morphogénèse et évolution des créatures artificielles ». Thèse de doctorat, Université de Toulouse, 2008.
- [LP, 2000] Lipson H., Pollack J. B. (2000) Automatic design and manufacture of artificial life forms. *Nature*, 406:974–978.
- [Lug, 1997] Luga Hervé. "Vie artificielle et synthèse d'images : Etude des mécanismes évolutionnistes pour la synthèse de formes et de comportements". Thèse de Doctorat, Université Paul Sabatier, Toulouse (France). 1997.
- [May, 1975] Mayr, E. (1975). *Evolution and the Diversity of Life*. Harvard University Press, Cambridge.
- [MC, 2005] T. Miconi and A. Channon. A virtual creature's model for studies in artificial evolution, 2005.
- [MC, 2006a]; T. Miconi and A. Channon. An improved system for artificial creatures evolution, 2006.
- [MC, 2006b]; T. Miconi and A. Channon. Then-strikes-out algorithm: A steady-state algorithm for coevolution. In the *IEEE Congress on Evolutionary Computation*, 2006.
- [Mic, 2007] Thomas Miconi. *THE ROAD TO EVERYWHERE* Evolution, complexity and progress in Nature and in computers. PhD thesis, The University of Birmingham, 2007.

- [McM, 2000] Barry Mc Mullin. John von Neumann and the evolutionary growth of complexity. *Artificial Life*, 6(4):347–361, 2000.
- [McP, 1943] Warren S. McCulloch & Walter Pitts. A logical calculus of ideas immanent in nervous activity. *Bulletin of mathematical biophysics* 5:115–133. Reprinted in McCulloch, W. S., *Embodiments of mind*. Cambridge, MA: MIT Press. (1943).
- [McZ, 1990] McKenna, M., and Zeltzer, D., “Dynamic Simulation of Autonomous Legged Locomotion,” *Computer Graphics*, Vol.24, No.4, July 1990, pp.29-38.
- [Mic, 2004] Michel O (2004) Webots: professional mobile robot simulation. *Int Journal of Adv Robotic Syst* 1:39–42;
- [Mil, 1988] Miller, G., “The Motion Dynamics of Snakes and Worms,” *Computer Graphics*, Vol.22, No.4, July 1988, pp.169-178.
- [Mit, 1995] K. Mitobe et al., Non-linear feedback control of a biped walking robot. In: *Proc. of the 1995 IEEE International Conference on Robotics and Automation*, pp.2865-2870, 1995.
- [ML, 2004] Malone E., Lipson H., (2004) “Functional Freeform Fabrication for Physical Artificial Life”, *Ninth Int. Conference on Artificial Life (ALIFE IX)*, *Proceedings of the Ninth Int. Conference on Artificial Life (ALIFE IX)*, pp.100-105.
- [MLH, 2002] M. Métivier, C. Lattaud, and J.C. Heudin. A stress-based speciation model in life drop. In *Proceedings of the 8th Int. Conf. On Artificial Life*, pages 121–126. MIT Press, 2002.
- [MM, 1996] David E. Moriarty and Risto Miikkulainen. Efficient reinforcement learning through symbiotic evolution. *Machine Learning*, 22:11–32, 1996.
- [MMDL, 2004] Mytilinaios E., Marcus D., Desnoyer M., Lipson H., (2004) “Designed and Evolved Blueprints For Physical Self-Replicating Machines”, *Proceedings of the Ninth Int. Conference on Artificial Life (ALIFE IX)*,

pp.15-20.

- [Mou, 2008] M. Jean-Baptiste MOURET, « Pressions sélectives multiples pour l'évolution de réseaux de neurones destinés à la robotique », thèse de doctorat, 5 décembre 2008.
- [MP, 1968] Marvin Minsky & Seymour Papert. Perceptrons : an Introduction to Computational Geometry, MIT Press, 1968.
- [MSA, 1989] Mjolsness, E., Sharp, D., and Alpert, B., "Scaling, Machine Learning, and Genetic Neural Nets," Advances in Applied Mathematics, Vol.10, pp.137-163, 1989.
- [Mul, 1998] Franck MULTON, « Contrôle du mouvements des humanoïdes de synthèse », Thèse de Doctorat, Université de Rennes 1, 16 Octobre 1998.
- [MV, 1980] Humberto Maturana and Francisco Varela. Autopoiesis and Cognition: The realization of the living. D.Reidel, Boston, 1980.
- [MZ, 1990] McKenna, M. & Zeltzer, D. (1990). Dynamic simulation of autonomous legged locomotion. Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1990 (pp. 29-38). Dallas, TX: ACM Press.
- [NF, 2002] S. Nolfi and D. Floreano. Synthesis of autonomous robots through evolution, 2002.
- [Nol, 2000] NOLFI, S. and FLOREANO, D., Evolutionary robotics. The MIT Press, 2000.
- [ODE, 2006] Open Dynamics Engine V0.5 user guide, Russell Smith, 2006.
- [OH, 2004]. M. J. T. O'Kelly and K. Hsiao. Evolving simulated mutually perceptive creatures for combat. In Ikegami T., Bedau M., Husbands P. and Watson R. A., editors, Proc. of Artificial Life IX, ed., pages 111–118, 2004.
- [ODD+, 2009a] N. OUANNES, N. DJEDI, Y. DUTHEN & H. LUGA, Evolution du contrôleur d'un robot humanoïde par un algorithme génétique dans un

- environnement complexe, Symposium IMAGE'2009, Algérie, 2009.
- [ODD+, 2009b] N. OUANNES, N. DJEDI, Y. DUTHEN & H. LUGA, Evolution du contrôleur d'un humanoïde pour des mouvements de locomotion, ICAI09 conference, Bordj Bou Arréridj, Algérie, 2009.
- [Pan, 2003] David Panzoli. Simulation comportementale par réseau de neurones et apprentissage par algorithme génétique. Master's thesis, IRIT, Toulouse, 2003.
- [Pan, 2008] David PANZOLI thèse « Simulation d'entités artificielles dans les environnements dynamiques virtuels ».septembre 2008.
- [Pat, 1988] Pattee, H. (1988). Simulations, realizations, and theories of life. *Artificial Life*, volume VI of Santa Fe Institute Studies in the Sciences of Complexity, pages 63– 77.
- [Pat, 1989] H. H. Pattee. Simulations, realizations, and theories of life. In C. G. Langton, editor, *Artificial life*, volume VI of Santa Fe Institute Studies in the Sciences of Complexity, pages 63–77, 1989.
- [PHLF, 2003] J. B. Pollack, G. S. Hornby, Hod Lipson, and P. Funes. Computer creativity in the automatic design of robots. *LEONARDO*, 2(36), 2003.
- [Ray, 1991a] Thomas S. Ray. Evolution and optimization of digital organisms. In Billing sley K. R., E. Derohanes, H. Brown, III [eds.], *Scientific Excellencein Supercomputing: The IBM 1990 Contest Prize Papers*, Athens, 1991.
- [Ray, 1991b] Thomas S. Ray. Aesthetically evolved virtual pets. In *Artificial Life 7 workshop proceedings*, pages158–161, 200.
- [Ray, 1993] RAY T.S., *How I Created Life in a Virtual Universe*, 1993, <http://www.isd.atr.co.jp/~ray/pubs/index.html>.
- [Ren, 2002] Rennard, J. (2002). *Vie artificielle, o`u la biologie rencontre l'informatique*. Vuibert Informatique, Paris, May, 1975.

- [Ren, 2003] Jean-Philippe RENNARD « Trois étapes de la vie artificielle : le canard de Vaucanson, les boucles de Langton et le Lt ». Avril 2003.
- [Rec, 1965] Rechenberg, I., Cybernetic Solution Path of an Experimental Problem, Royal Aircraft Establishment Library Translation, 1965
- [Rec, 1994] RECHENBERG, I., Evolutionsstrategien. Holtzmann-Froboog, 1994.
- [Rey, 1987] Reynolds,C.(1987). Flocks, herds, and schools: A distributed behavioural model. Proceedings of the 14st Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1987 (pp. 25-34). New York, NY: ACM Press.
- [RH, 1991] Raibert, M., and Hodgins, J.K., “Animation of Dynamic Legged Locomotion,” Computer Graphics, Vol.25, No.4, July 1991, pp.349-358.
- [RHW, 1986] David E. Rumelhart, Geoffrey E. Hinton, & Ronald J. Williams. Learning representations by back-propagating errors. Nature 323:533–536 (1986).
- [Ros, 1958] Frank Rosenblatt. The perceptron: probabilistic model for information storage and organisation in the brain. Psychological Review 65:386–408 (1958).
- [San, 2001] Cédric Sanza. Evolution d’entités virtuelles coopératives par système de classifieurs. Ph.D. thesis, Université Paul Sabatier (2001).
- [San, 2004] Stéphane Sanchez, « une architecture générique intégrant des mécanismes évolutionnistes pour la simulation comportementale d’acteurs virtuels», le 8 Décembre 2004.
- [Sch, 1993] H. P. P. Schwefel. Evolution and Optimum Seeking : The Sixth Generation. John Wiley & Sons, Inc. New York, NY, USA, 1993.
- [Sim, 1994a] Karl Sims. Evolving 3d morphology and behavior by competition. Artificial Life, 1(4):353–372, 1994.
- [Sim, 1994b] Karls Sims. Evolving 3d morphology and behaviour by competition. In

- Artificial Life IV, pagespp.28–39. Artificial Life, MIT Press, 1994.
- [Sim, 1994c] Karl Sims. Evolving virtual creatures. In SIGGRAPH 94, pages 15–22. ACM Press, 1994.
- [SK, 2007] SHIM, Y.-S. AND KIM, C.-H. 2007. Evolving physically simulated flying creatures for efficient cruising. *Artificial Life* 12, 4, 561–591.
- [SLD, 2008] Sylvain Cussat-Blanc, Hervé Luga et Yves Duthen Développement de créatures artificielles primitives possédant un métabolisme et une morphologie, AFIG 2008, Toulouse IRIT Presse.
- [SM, 2002] Kenneth O. Stanley and Risto Miikkulainen. efficient reinforcement learning through evolving neural network topologies. In *Procs GECCO 2002*. Morgan-Kaufmann, 2002.
- [Smi, 1984] Smith, A.R., “Plants, Fractals, and Formal Languages,” *Computer Graphics*, Vol.18, No.3, pp.1-10, July 1984.
- [Smi, 2002] How to make new joints in ODE Copyright © 2002 Russell Smith February 24, 2002
- [Sob, 1996] SOBER E., « Learning from Functionalism — Prospects for Strong Artificial Life », dans BODEN M.A. (ed.), *The Philosophy of Artificial Life*, Oxford Readings in Philosophy, 1996.
- [Sys, 1989] Syswerda G., « Uniform crossover in Genetic Algorithms », 3rd International Conference on Genetic Algorithms, 1989, USA, p. 2-9.
- [Tay, 1999] Tim Taylor. From artificial evolution to artificial life. PhDthesis, University of Edinburgh, 1999.
- [Tou, 2003] Toussaint, M. (2003b). The evolution of genetic representations and modular neural adaptation. PhD thesis, Ruhr-Universität, Bochum.
- [TT, 1994] Xiaoyuan Tu and Demetri Terzopoulos, *Artificial Fishes: Physics, Locomotion, Perception, Behavior* Department of Computer Science, University of Toronto, 10 King’s College Road, Toronto, Ontario,

Canada, M5S 1A4. 1994

- [TTG, 1994] Demetri Terzopoulos, Xiaoyuan Tu, and Radek Grzeszczuk. Artificial fishes: Autonomous locomotion, perception, behavior, and learning in a simulated physical world. *Artificial Life*, 1(4):327–351, 1994.
- [Tv, 2003] Faloutsos, P., vandePanne, M. & Terzopolous, D. (2003). Autonomous reactive control for simulated humanoids. *IEEE International Conference on Robotics and Automation* (pp. 917-924). Taipei, Taiwan: IEEE.
- [VBS, 2001] M. Vukobratovic, B. Borovac, and D. Surdilovic. Zero-moment point proper interpretation and new applications. In *Proceedings of the 2nd International Conference on Humanoid Robots, Humanoids'01*, Waseda University, Tokyo, Japan, 22-24 November, PP. 237-244, IEEE-RAS, Humanoid Robotics Institute, 2001.
- [vdP, 1993] Michiel van dePanne. Sensor-actuator networks. In *SIGGRAPH'93: Proceedings of the 20 th annual conference on Computer graphics and interactive techniques*, pages 335–342. ACM, 1993.
- [Ven, 1994] Jeffrey Ventrella. Explorations in the emergence of morphology and locomotion behavior in animated characters. In *Proceedings of the 4th International Workshop on the Synthesis and Simulation of Living Systems Artificial Life IV*, pages 436–441, Cambridge, MA, USA, 1994. MIT Press.
- [Ven, 1996] Jeffrey Ventrella. Sexual swimmers: Emergence morphology and locomotion behavior. In *Artificial Life IV*, pages 484–493, 1996.
- [Ven, 1998a] 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*, pages 178–186, Cambridge, MA, USA, 1998. MIT Press.
- [Ven, 1998b] Jeffrey Ventrella. Designing emergence in animated artificial life worlds. In *VW '98: Proceedings of the First International Conference on Virtual Worlds*, pages 143–155, London, UK, 1998. Springer-Verlag.

- [Ven, 2005] Jeffrey Ventrella. *Artificial Life Models in Software*, chapter Gene Pool: Exploring the Interaction Between Natural Selection and Sexual Selection. Springer-Verlag, 2005.
- [Wah, 2005] Evolutionary Robotics The Use of Artificial Evolution in Robotics. A tutorial presented at AMiRE2005. MattiasWahde Technical ReportTR-BBR-2005-002. Department of Applied Mechanics Chalmers University of Technology Göteborg, Sweden.
- [Wal, 1875] Alfred Wallace. The limits of natural selection as applied to man. In *Contributions to the Theory of Natural Selection*, chapter 10. Macmillan, 1875.
- [WBH+, 2006] Krister Wolff, Jimmy Pettersson, Almir Heralic, and Mattias Wahde. Title: "Structural Evolution of Central Pattern Generators for Bipedal Walking in 3D Simulation" to appear in *Proceedings of the 2006 IEEE International Conference on Systems, Man, and Cybernetics (SMC 2006)*, Taipei, Taiwan, October 2006.
- [Weh, 2005] WAHDE, M., "Evolutionary robotics: The use of artificial evolution in robotics", Tutorial presented at AMiRE2005, 2005. Available from: <http://www.me.chalmers.se/~mwahde/AdaptiveSystems/Tutorials.html>.
- [WH, 1960] Bernard Widrow & Marcian T. Ho. Adaptive switching circuits. *IRE WESCON Convention Record* pp. 96–104 (1960).
- [WFO, 1966] M. J. Walsh L. J. Fogel, A. J. Owens. *Artificial Intelligence Through Simulated Evolution*. Wiley, 1966.
- [Yao,1999] [Yao, 99] Yao. Evolving artificial neural networks. *Procs of the IEEE*, 87, 1999.
- [YZD, 2002] Yim, M., Zhang, Y. and Duff, D., "Modular Reconfigurable Robots, Machines that shift their shape to suit the task at hand," *IEEE Spectrum Magazine* cover article, Feb. 2002.