

REMERCIEMENTS

Nous tenons à remercier en premier lieu notre grand dieu qui nous a donné la force et le courage pour continuer et nous avoir aidées et éclairer le chemin pour réaliser de ce modeste travail.

*Nous exprimons toute notre gratitude et reconnaissance à notre encadreur **Md.OUNNAS Nesrine** pour ses merveilleux conseils et aides pendant notre travail de fin d'études.*

Nous remercions également nôtres collègues de l'option Intelligence artificielle pour leurs précieuses aides.

*Nous n'oublions pas tous ceux qui nous aidés de près ou de loin, ainsi que tous les employés et les étudiants **du Département D'informatique** pour leur aide.*

Seksaf Soundes

Table des matières

Introduction générale

Chapitre 1 : Les techniques de vie artificielle

Introduction.....	1
1. Vie artificielle.....	1
2. Les modèles artificiels.....	2
3. Les principales techniques liées à la vie artificielle	3
3.1. Automates cellulaires	3
3.2. Jeu de la vie	4
3.4. Les systèmes de classifieurs	6
3.5. Les Animats	7
4. Les algorithmes évolutionnistes	7
4.1. L'inspiration darwinienne.....	8
4.2. Principe d'un algorithme évolutionniste	8
Les composants	9
4.3. L'espace de recherche	9
4.4. Opérateurs génétiques	10
4.5. Les schémas d'évolution	14
5. Les réseaux de neurones.....	15
5.1. Apprentissage	16
Conclusion	17

Chapitre 2 : Les créatures artificielles

Introduction.....	1
1. Définition de créatures artificielles.....	2
2. Phylogenèse artificielle	3
2.1. Les créatures de Karl Sims	3
2.2. Créatures de Gregory Hornby.....	5
2.3. Framsticks	6
2.4. Créatures de Nicolas Lassabe	6
2.5. Autres travaux	7
3. Les écosystèmes	8
3.1. Les écosystèmes naturels.....	8
3.2. Les travaux sur les écosystèmes artificiels	9
3.2.1. Tierra	10
3.2.2. Gene Pool	11
3.2.3. Life Drop.....	11

3.2.4. Darwin's Park.....	12
3.2.5. PolyWorld	12
Conclusion	13

Chapitre 3 : Conception de système

Introduction.....	1
1. L'objectif de travail.....	1
2. conception global	1
3. Conception détaillé.....	2
3.1. L'environnement de simulation	2
3.2. Les créatures artificielles	3
3.2.1. L'entité source d'énergie (nourriture).....	4
3.2.2. L'entité fourrageur.....	4
3.3. Algorithme de mouvement de la créature prédatrice	6
3.4. L'évolution	7
3.5. Algorithme génétique	8
3.5.1. Pseudo code de l'algorithme génétique	8
Conclusion	10

Chapitre 4 : Implémentation et résultats

Introduction.....	1
1. Le langage de programmation	1
1.1. Présentation de langage utilisé.....	1
1.2. La Bibliothèque graphiques OpenGL (Open Graphic Library).....	2
1.3. Motivation du choix du Builder C++.....	2
2. Implémentation	2
2.1. L'environnement	2
2.2. Algorithme de construction de l'environnement.....	3
2.3. Les structures des Créatures artificielles.....	4
2.4. Description de l'application.....	5
2.5. Présentation de l'algorithme	5
3. Résultats obtenus.....	6
3.1. Interface graphique	6
Conclusion	10

Conclusion générale

Bibliographie

Introduction générale

Introduction générale

S'inspirer du vivant est de nos jours capital pour le développement de la technologie du futur et l'informatique du demain. Alliant informatique et biologie et naissant du croisement de ces deux disciplines, la synthèse de créatures artificielles est depuis peu un domaine de recherche qui présente un intérêt de plus en plus grandissant.

Le travail présenté dans ce manuscrit, consiste à adopter des termes de biologie et d'écologie et d'essayer de les reproduire artificiellement dans des créatures artificielles.

L'évolution des comportements de ces créatures artificielles doit nous fournir des individus évolués qui ont reçu un sous-ensemble de paramètres qui déterminent la stratégie de recherche de nourriture d'un individu (i.e., le comportement) de leurs antécédents sélectionnés ce qui implique que les descendants de chaque génération auront les meilleurs comportements (i.e. traduction des meilleurs paramètres). Les créatures simulées sont un ensemble de fourrageurs possédant un métabolisme.

Le mémoire est organisé comme suit :

Le premier chapitre est celui de la vie artificielle où on décrit et on définit les termes les plus utilisés par les chercheurs dans ce domaine. Le deuxième chapitre présente un état de l'art sur l'étude de la phylogénèse des créatures artificielles, de l'ontogénèse, et l'évolution dans les écosystèmes.

Après avoir présenté les travaux dans le domaine de la vie artificielle et les concepts biologiques, nous avons pu construire notre propre contribution en simulant les concepts de biologie des fourrageurs, et pour l'aspect évolution, nous avons proposé un codage spécifique qui décrit un ensemble de capacités pour chaque fourrageur dans l'écosystème qui sont : la capacité de vitesse, la capacité de déplacement et la capacité de consommation d'énergie. Le processus d'évolution doit conduire les meilleurs individus vers les capacités qui rendent leurs

mouvements plus efficaces et une fois que toute la population se comporte presque de la même façon on considère que la solution est trouvée.

Le dernier chapitre présente un ensemble d'expérimentations effectuées avec des paramètres fixes, ensuite on étudie et nous discutons les résultats obtenus qui sont considérés comme notable. Nous terminons par une conclusion générale et un ensemble de perspectives.

Chapitre 1

Chapitre 1 : Les techniques de vie artificielle

Introduction

La discipline qui est de nos jours le fer de lance pour les approches d'inspiration biologique est sans conteste, la vie artificielle. Cette discipline constitue le cadre dans lequel s'inscrit ce mémoire. Pour cela, nous l'introduisons dans ce chapitre en présentant ses définitions puis nous présenterons l'ensemble des techniques qui sont d'inspiration purement informatique (L-systèmes, automates cellulaire, etc.) et/ou biologique (algorithmes génétique, réseau de neurone, etc.), et qui sont maintenant attribués à la vie artificielle.

1. Vie artificielle

Il est difficile de donner une définition exacte de la vie artificielle. Plusieurs définitions ont déjà été proposées. Ainsi, selon (Christopher Langton): "La vie artificielle est donc l'étude de systèmes vivants naturels pour en trouver des principes afin de recréer sur des supports artificiels des phénomènes biologiques à partir de rien."

Ou encore, selon (J.-P. Rennard): "La vie : état de ce qui n'est pas inerte. La vie artificielle : domaine de recherche qui cherche à expliquer la définition précédente."

Enfin, on pourrait définir la vie artificielle par les caractéristiques que doivent posséder les systèmes de vie artificielle : "Ces systèmes doivent avoir été créés par l'homme, être autonomes, être en interaction avec leur environnement, faire émerger un ou des comportements. Ils peuvent aussi, mais cela n'est pas nécessaire, se reproduire et être capables de s'adapter" (J. Doyne Farmer).

La vie artificielle est un domaine scientifique d'étude et une forme d'art associé qui examinent les systèmes liés à la vie, ses processus et son évolution, grâce à l'utilisation de simulations avec des modèles informatiques, la robotique et la biochimie.

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 s'arrêtent.

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.

Ils sont partout et ce n'est pas fini. Aujourd'hui outils de conception et d'optimisation dans les industries de pointe, demain ils envahiront notre vie quotidienne.

2. Les modèles artificiels

Dans la vie artificielle il y a beaucoup des modèles qui sont inspirés de la biologie on peut l'appeler les modèles écologiques, par exemple :

1- Les algorithmes de colonies de fourmis s'inspirent du comportement réel des fourmis dans la nature.

Les fourmis interagissent entre elles par l'intermédiaire de leur environnement afin d'effectuer des tâches complexes. Malgré leur petite taille qui les empêche d'avoir une vue globale de leur environnement, les fourmis d'une colonie empruntent le chemin le plus court pour contourner un obstacle ou pour aller chercher de la nourriture.

2- Un organisme parasite ou parasitaire est simplement une forme de vie qui existe principalement hors de son hôte. Les parasites vivent une vie parallèle dans notre corps, se nourrissant de notre propre énergie et nos propres cellules. L'idée que les parasites ou les

prédateurs peuvent conduire à des effets positifs est bien connue en biologie. Les réactions parasites bénéfiques pourraient ne pas avoir été rares pendant les origines de la vie, pour la simple raison que les réactions parasites auto-catalytiques peuvent fournir un mécanisme de l'hérédité, augmentant la probabilité que de tels systèmes seraient présents dans le monde prébiotique.

3- Un des acteurs souvent trouvés dans les voies de transduction du signal est l'enzyme. Enzymes biologiques sont une classe spécifique de protéines qui catalysent les réactions chimiques qui se produisent dans un système cellulaire spécifique. Les molécules appelées acte de substrat en entrée et se lie à l'enzyme formant un complexe enzyme-substrat. Les molécules de substrat sont alors manipulées et modifiées par l'enzyme pour produire des molécules de sortie appelé produits ayant des propriétés différentes en général à partir du substrat. Comme toutes les enzymes sont des catalyseurs, le taux de réaction chimique et de l'expression dans la cellule est commandé à la fois par la concentration de molécules de substrat et le montant de la participation enzymatique.

3. Les principales techniques liées à la vie artificielle

Les techniques artificiels liés au domaine de vie artificielle qui dont sont d'une inspiration purement informatique sont : (L-systèmes, automates cellulaire, etc.) et/ou biologique (algorithmes génétique, réseau de neurone, etc.).

3.1. Automates cellulaires

Reprenant la théorie des automates autoreproducteurs de Von Neumann, Christopher Langton créa en 1984 le premier automate cellulaire.

Contrairement à l'entité 'cellule' du jeu de la vie possédant uniquement un état 'vivant' et un état 'mort', cet automate, plus complexe, est composé d'une centaine de cellules comprenant huit états possibles. Comme le montre la figure suivante, l'automate est constitué d'une membrane externe de cellules à l'état 2, enveloppant les cellules internes. Un certain nombre de ces cellules d'état 0, 1, 4 et 7, sont des programmes de reproduction capables de reproduire l'automate. Une reproduction complète d'un automate cellulaire s'effectue en 151 pas.

```
22222222
2170140142
2022222202
272    212
212    212
202    212
272    212
21222222122222
2071071071 11 11 2
22222222 22 22 2
```

Figure 1. La boucle de LANGTON dans sa forme initiale (structure initiale de 86 cellules).

La propriété émergente de ce type d'automates est la structure que prend une population d'automates reproduits à partir d'un unique automate. Après quelques générations seulement, l'organisme résultant reprend le même développement et la même architecture que les récifs de coraux. Les automates cellulaires font apparaître un autre principe de la vie artificielle : Le principe de similitude représente l'apparition d'une structure émergente de l'ensemble de l'organisme résultant, possédant une similitude de forme proche de certaines espèces naturelles.

3.2. Jeu de la vie

En 1970, le mathématicien John Conway, eu l'idée de créer un jeu avec un minimum de règles, où des cellules naissent, vivent et disparaissent dans un espace discret toroïdal à deux dimensions. Les cases de cet espace représentent les cellules du jeu et possèdent deux états possibles : vivant ou mort, c'est à dire allumé ou éteint. Après une initialisation aléatoire de l'environnement, l'état des cellules à un instant $t+1$ est déterminé par son état à l'instant t et l'application des deux lois suivantes :

- Une cellule naît si elle possède trois cellules voisines vivantes.
- Une cellule meurt si elle possède moins de deux cellules voisines vivantes (mort par isolement) ou plus de trois cellules voisines vivantes (mort par surpopulation).

Après quelques générations, il apparaît des comportements surprenant de la part du système. A partir de ces deux lois simples, certains types de formation de cellules émergent de la structure grouillante du départ. Ainsi la figure suivante met en évidence quatre structures différentes :

- La forme dite « beehive » ou nid d'abeilles, est une structure statique, trois nids sont présents au même endroit sur les deux états.
- La forme dite blinker ou clignotant, est une structure stable périodique à deux états.

- La forme dite « traffic » light ou feu de circulation, est une structure stable périodique à deux états, les deux présents sont facilement discernables puisqu'un état est une croix alors que le second est un carré aux arêtes tronquées. Le feu de droite est cependant une version déformée du feu standard.
- La forme dite « glider » ou planeur, est une structure mobile périodique à cinq états, elle a la faculté de glisser dans l'espace en conservant sa forme par période.

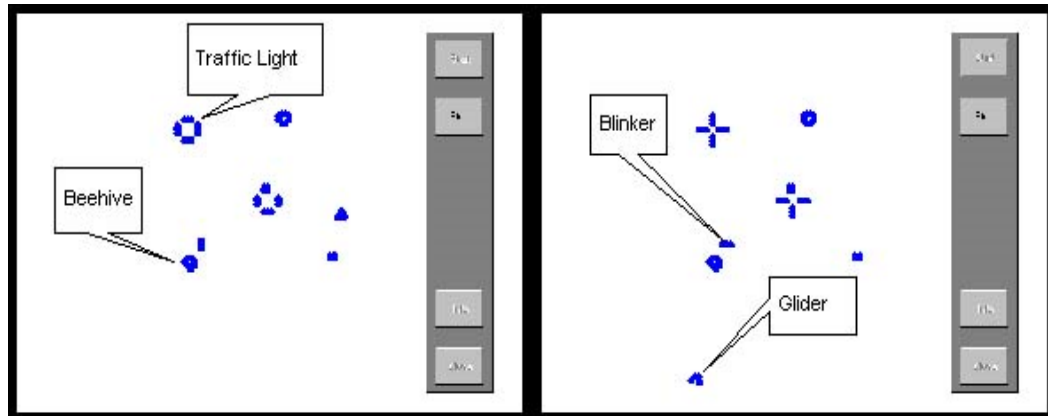


Figure 2. Les quatre différentes structures de jeu de la vie.

3.3. L-Systèmes

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 obtenues 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/vmmdeluxe/TitlePage.html>.)

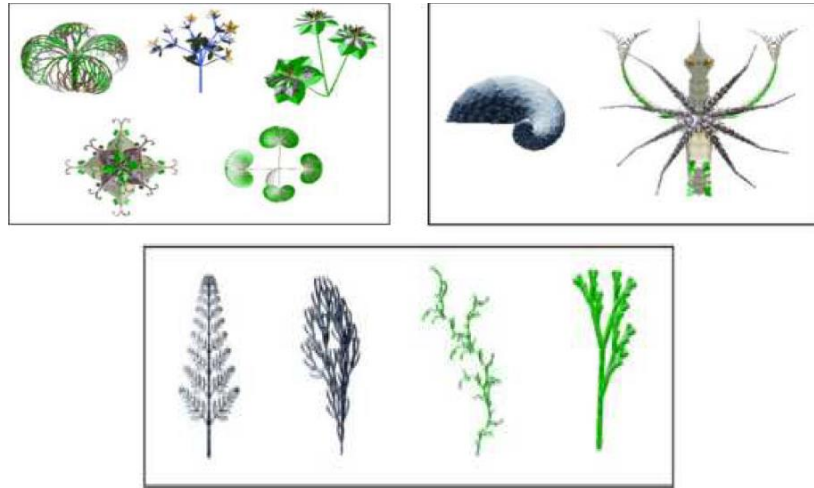


Figure 3. Quelques images obtenues des L-Systèmes.

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 signifient 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, 03].

3.4. Les systèmes de classifieurs

Les systèmes de classifieurs, dont les fondements théoriques sont une méthode mêlant apprentissage automatique et évolution. Les connaissances du système données sous forme de règles nommées classifieurs, sont apprises ou oubliées suivant leurs performances lors de leur utilisation. Ainsi, une des différences par rapport à un système expert standard est l'importance des règles, celles-ci ne sont ni définies par le programmeur, ni par les experts du domaine. La seconde différence principale avec les systèmes experts est la possibilité de déclencher des règles en parallèle, et non plus obligatoirement de manière séquentielle, c'est à dire les unes après les autres. Un système de classifieurs est composé de quatre parties :

- Une base de classifieurs représentant les connaissances du système.
- Une interface d'entrée composée de capteurs.
- Une interface de sortie composée d'effecteurs.

- Une liste de messages.

3.5. Les Animats

Bien que le principe d'un animal artificiel fût envisagé par L. Booker, le terme d'animat fut mentionné pour la première fois par S. Wilson lors de la première conférence internationale sur les algorithmes génétiques en 1985. Il y présenta son animat, animal artificiel, capable de découvrir de la nourriture au milieu d'arbres dans un monde discret de 18*58 cases, à l'image de l'environnement de la figure suivante.

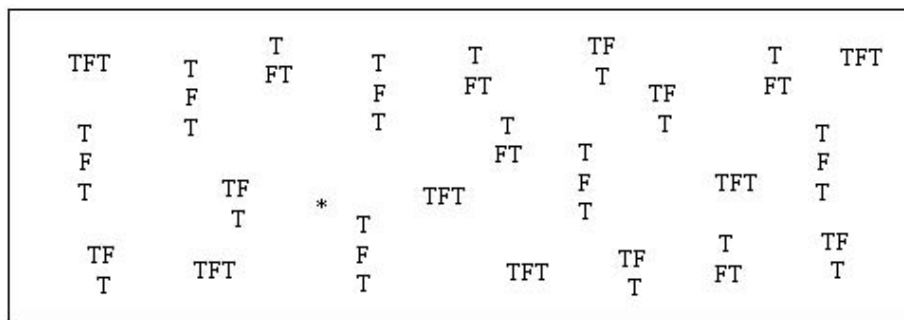


Figure 4. Tableau_animats (T : représente les arbres, F : représente la nourriture, * : représente l'animat).

4. Les algorithmes évolutionnistes

Les algorithmes évolutionnistes sont basés sur la théorie de l'évolution des espèces de Darwin: cela permet l'évolution et l'optimisation d'un automate. Le but est de trouver la solution d'un problème.

Ils font évoluer un ensemble de solutions dans le but de répondre à un problème donné. Ils utilisent des processus aléatoires pour trouver la réponse qui convient.

Les algorithmes évolutionnistes sont déjà anciens et ils sont issus historiquement de trois sections :

- les algorithmes génétiques (GA pour Genetic Algorithm)
- Les stratégies évolutives (ES pour Evolution Strategies)
- La programmation évolutive (EP Evolutionary Programming)

Les algorithmes génétiques ont été introduits et étudiés notamment par Holland et Goldberg. Ils s'inspirent de la théorie de l'évolution des espèces de Darwin.

4.1. L'inspiration darwinienne

Selon la génétique et la théorie de l'évolution

- Un enfant hérite son patrimoine génétique pour moitié de sa mère et pour moitié de son père.
- Les enfants ne sont pas identiques aux parents car des altérations des gènes peuvent se produire (mutations).
- Parmi les mutations, certaines peuvent être favorables et d'autres défavorables.
- Il naît beaucoup de descendants : mais seuls les individus les mieux adaptés pourront survivre et transmettre leurs gènes à leur tour à leur descendance.

4.2. Principe d'un algorithme évolutionniste

Les algorithmes génétiques (et les algorithmes évolutionnistes en général) s'inspirent de la théorie de l'évolution pour résoudre un problème d'optimisation (avec un espace de recherche et une fonction d'évaluation), un algorithme génétique fait évoluer itérativement une population de configurations. Les algorithmes génétiques sont les plus utilisés dans le calcul évolutionniste, développés par John Holland [Hol75] puis approfondis par David Goldberg [Gol89].

A chaque itération (génération) :

- Des opérateurs de variation (mutation et croisement) permettent d'engendrer de nouvelles configurations (les enfants) à partir des configurations de la population courante (les parents).
- Les configurations obtenues peuvent être bonnes ou non selon la fonction d'évaluation. Un opérateur de sélection élimine les configurations les moins bonnes.

L'algorithme peut être exprimé ainsi :

- Construction et évaluation d'une population initiale jusqu'à atteindre un critère d'arrêt :
- Sélection d'une partie de la population
- Reproduction des individus sélectionnés
- Mutation de la descendance
- Évaluation du degré d'adaptation de chaque individu
- Remplacement de la population initiale par une nouvelle population.

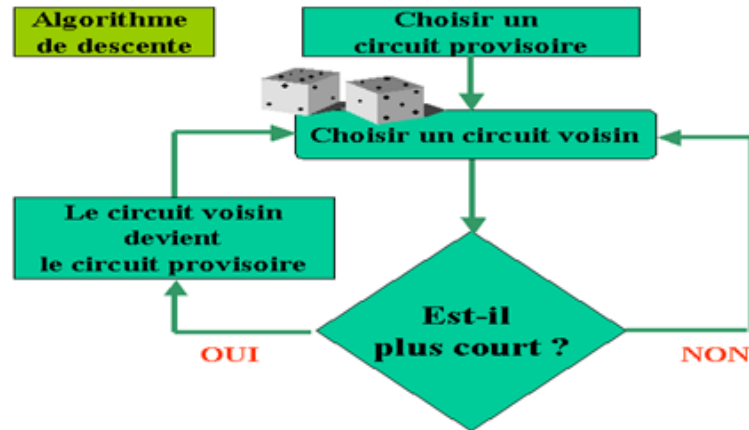


Figure 5. Exemple d'algorithme évolutionniste.

Les composants

Nous allons maintenant détailler les principaux composants d'un algorithme évolutionnaire, en donnant des exemples concrets.

On notera tout d'abord que l'on peut répartir les diverses étapes de l'algorithme de la section précédente en deux groupes : celles qui dépendent de l'espace de recherche (initialisation, opérateurs génétiques et évaluation) et celles qui n'en dépendent pas (sélection et remplacement).

L'étape d'évaluation, totalement spécifique au problème traité, et objet des sections sur la parallélisations.

4.3. L'espace de recherche

Il s'agit de la composante principale de l'algorithme -- qui est même en fait préalable aux autres. Dans de nombreux cas, l'espace de recherche est totalement déterminé par le problème (la fonction objectif). Mais il est toujours possible de transporter son problème dans un espace habilement choisi (changement de "variables") où sa résolution sera plus aisée. Cet espace, où seront appliqués les opérateurs génétiques, est alors appelé espace génotypique, et l'espace de recherche initial, dans lequel est calculée la performance des individus, est appelé espace phénotypique. Deux exemples d'espaces de recherche parmi les plus utilisés -- et détaillerons les composantes de l'algorithme dans ces deux cas particulier.

Les chaînes de bits (bitstring en anglais).

L'espace de recherche est ici $\Omega = \{0,1\}^N$ il s'agit de l'école des algorithmes génétiques, et la justification de l'utilisation intensive de cet espace de recherche particulier était fondé à la fois sur un parallèle encore plus précis avec la biologie (une chaîne de bits étant assimilée à un chromosome) et sur des considérations théoriques. Ce contexte reste toutefois très utilisé, et permet également une présentation aisée des divers composants de l'algorithme.

Les vecteurs de réels

C'est bien sûr le cas le plus fréquent en calcul numérique : Ω est un sous-ensemble d' \mathbb{R}^n , borné ou non. On parle alors aussi d'optimisation paramétrique.

Initialisation

Le principe général de l'initialisation est d'échantillonner le plus uniformément possible l'espace de recherche Ω .

Dans le cas des chaînes de bits, chaque bit de chaque individu est tiré égal à 0 ou à 1 avec une probabilité 0.5. Dans le cas de l'optimisation paramétrique, si $\Omega = \Omega[a_i, b_i]$ (cas borné), on tire uniformément chaque coordonnée dans l'intervalle correspondant. Par contre, si Ω n'est pas borné, il faut faire des choix. On pourra par exemple tirer mantisses et exposants uniformément.

4.4. Opérateurs génétiques

C'est durant cette étape que de nouveaux individus sont créés à partir des individus préalablement sélectionnés. On distingue les opérateurs de croisement (binaires, ou plus généralement n-aires) et les opérateurs de mutation, unaires. A noter que cette étape est toujours stochastique, c'est à dire que le résultat de l'application d'un opérateur dépend de tirages aléatoires.

Le croisement

L'idée générale du croisement est l'échange de matériel génétique entre les parents : si deux parents sont plus performants que la moyenne, on peut espérer que cela est dû à certaines parties de leur génotype, et que certains des enfants, recevant les "bonnes" parties de leurs deux parents, n'en seront que plus performants. Ce raisonnement, trivialement valable pour des fonctions performance linéaires, est extrapolé (et expérimentalement vérifié) à une classe plus étendue de fonctions, sans que les résultats théoriques aujourd'hui disponibles ne permettent de délimiter précisément la classe de fonctions pour lesquelles le croisement est utile.

Signalons enfin que tous les individus sélectionnés ne sont pas forcément croisés. Le croisement -- comme tous les opérateurs génétiques, est appliqué avec une certaine probabilité notée P_c .

Dans le cadre des chaînes de bits, les divers opérateurs de croisement échangent des bits (à position fixée) entre les parents. La figure 6 donne l'exemple du croisement à point. Un autre croisement consiste à tirer au sort pour chaque position (avec probabilité 0.5) de quel parent proviendra le bit correspondant chez chaque enfant.

$$\left. \begin{array}{l} (b_1, \dots, b_N) \\ (c_1, \dots, c_N) \end{array} \right\} \xrightarrow{P_c} \left\{ \begin{array}{l} (b_1, \dots, b_l, c_{l+1}, \dots, c_N) \\ (c_1, \dots, c_l, b_{l+1}, \dots, b_N) \end{array} \right.$$

Figure 6. Exemple de croisement à point.

Dans le cadre de l'optimisation paramétrique, on peut bien entendu appliquer les mêmes opérateurs d'échange de coordonnées. On peut également -- et c'est en général bien plus efficace -- "mélanger" les deux parents par une combinaison linéaire. On parle alors de croisement arithmétique :

$$(X,Y) \xrightarrow{P_c} \alpha X + (1 - \alpha) Y, \alpha = U[0,1]$$

Ou

$$(X,Y) \xrightarrow{P_c} \alpha_i X_i + (1 - \alpha_i) Y_i, \alpha_i = U[0,1]$$

La première version revient à choisir l'enfant uniformément sur le segment $[X Y]$ alors que la deuxième revient à tirer l'enfant uniformément sur l'hyper-cube dont $[X Y]$ est une diagonale. Remarquons que l'échange de coordonnées revient à choisir comme enfant un des sommets de ce hyper-cube.

La mutation L'idée directrice de la mutation est de permettre de visiter tout l'espace. Les quelques résultats théoriques de convergence des algorithmes évolutionnaires ont d'ailleurs tous comme condition l'ergodicité de la mutation, c'est-à-dire le fait que tout point de l'espace de recherche peut être atteint en un nombre fini de mutations.

Dans le cadre des chaînes de bits, on modifie aléatoirement certains bits. Par exemple, pour chaque parent (ou enfant du croisement), et pour chaque position l ,

$$(b_1, b_2, \dots, b_N) \xrightarrow{P_m} (b_1, b_2, \dots, 1-b_l, b_{l+1}, \dots, b_N)$$

Attention, la probabilité p_m est ici une probabilité par bit et non par individu.

Une autre possibilité est de prédéfinir un nombre de bits à modifier (généralement 1) et de choisir ensuite au hasard une position dans chaque individu et de modifier le bit correspondant. Cet opérateur est lui appliqué avec une probabilité p_m par individu, bien sûr.

Dans le cadre de l'optimisation paramétrique, la mutation la plus employée est la mutation gaussienne, qui consiste à rajouter un bruit gaussien à chaque variable (l'opérateur étant appliqué avec une certaine probabilité par individu p_m).

$$X_i := X_i + N(0, \sigma^2)$$

Tout l'art est alors dans le choix de la variance σ^2 du bruit gaussien utilisé. On peut évidemment demander à l'utilisateur de fixer cette valeur. Mais des études théoriques sur des fonctions simples ont démontré que cette valeur devrait décroître au fil des générations -- et il devient difficile de fixer une suite de valeurs décroissante qui soit synchrone avec l'éventuelle convergence de l'algorithme pour des fonctions quelconques.

L'idée originale pour résoudre ce problème est due aux pères des stratégies d'évolution (I. Rechenberg et H.-P. Schwefel). Elle consiste à considérer ces σ^2 comme des variables supplémentaires, et à les faire également évoluer.

Si l'espace phénotypique initial est, par exemple, $\square[ai, bi]$, l'espace génotypique sera $\square[ai, bi] \times \mathbb{R}^+$, chaque individu se voyant muni de sa propre variance σ^2 . La mutation devient alors

$$\sigma := \sigma \exp(\tau N(0, 1))$$

$$X_i := X_i + N(0, \sigma)$$

La variance σ^2 est modifiée par une perturbation log-normale -- elle doit rester positive, et son évolution doit être symétrique par rapport à un puisqu'elle intervient ensuite de manière multiplicative.

L'idée est qu'un individu ayant de bonnes performances après plusieurs mutations n'est pas arrivé où il est par hasard, et doit avoir une valeur de σ^2 adaptée à la zone de l'espace de recherche où il se trouve. Signalons qu'il existe une variante dans laquelle une variance est attachée à chaque variable. Enfin, dans la variante la plus générale, une matrice de corrélation complète évolue avec chaque individu.

Procédures de sélection

La partie darwinienne de l'algorithme comprend les deux étapes de sélection et de remplacement. Répétons que ces étapes sont totalement indépendantes de l'espace de recherche.

On distingue deux catégories de procédures de sélection ou de remplacement (par abus de langage, nous appellerons sélection les deux types de procédures) : les procédures déterministes et les procédures stochastiques.

Sélection déterministe

On sélectionne les meilleurs individus (au sens de la fonction performance). Si plus de quelques individus doivent être sélectionnés, cela suppose un tri de l'ensemble de la population -- mais cela ne pose un problème de temps calcul que pour des très grosses tailles de population.

Les individus les moins performants sont totalement éliminés de la population, et le meilleur individu est toujours sélectionné -- on parle alors d'élitisme.

Sélection stochastique

Il s'agit toujours de favoriser les meilleurs individus, mais ici de manière stochastique, ce qui laisse une chance aux individus moins performants. Par contre, il peut arriver que le meilleur individu ne soit pas sélectionné, et qu'aucun des enfants n'atteigne une performance aussi bonne que celle du meilleur parent ...

Le tirage de roulette est la plus célèbre des sélections stochastiques. Supposant un problème de maximisation avec uniquement des performances positives, elle consiste à donner à chaque individu une probabilité d'être sélectionné proportionnelle à sa performance. Une illustration de la roulette est donnée figure 7 : on lance la boule dans la roulette, et on choisit l'individu dans le secteur duquel la boule a fini sa course.

Le tirage de roulette présente toutefois de nombreux inconvénients, en particulier reliés à l'échelle de la fonction performance : alors qu'il est théoriquement équivalent d'optimiser $\square F + \square$ et F pour tout $\square > 0$, il est clair que le comportement de la sélection par roulette va fortement dépendre de \square dans ce cas. C'est pourquoi cette sélection est presque totalement abandonnée aujourd'hui.

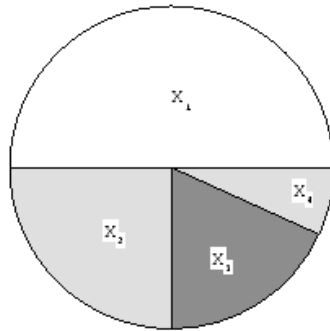


Figure 7. Sélection par tirage de roulette : cas de 4 individus de performances respectives 50, 25, 15 et 10. Une boule est "lancée" dans cette roulette, et l'individu dans le secteur duquel elle s'arrête est sélectionné.

La sélection par le rang consiste à faire une sélection en utilisant une roulette dont les secteurs sont proportionnels aux rangs des individus. La variante linéaire utilise directement le rang (P pour le meilleur, 1 pour le moins bon, pour une population de taille P), les variantes polynomiales remplaçant ces valeurs par i/P^α , $\alpha > 0$. Seuls comptent alors les positions relatives des individus entre eux, et non les valeurs -- arbitraires -- de la fonction F .

La sélection par tournoi n'utilise aussi que des comparaisons entre individus -- et ne nécessite même pas de tri de la population. Elle possède un paramètre T , taille du tournoi. Pour sélectionner un individu, on en tire T uniformément dans la population, et on sélectionne le meilleur de ces T individus. Le choix de T permet de faire varier la pression sélective, c'est-à-dire les chances de sélection des plus performants par rapport aux plus faibles. A noter que le cas $T=2$ correspond, en espérance et au premier ordre en fonction de P , à la sélection par le rang linéaire.

4.5. Les schémas d'évolution

On regroupe sous ce nom les ensembles sélection/remplacement, qui ne peuvent être dissociées lors des analyses du darwinisme au sein des algorithmes évolutionnaires. Un schéma d'évolution est donc la réunion d'une procédure de sélection et d'une procédure de remplacement. Toute combinaison des procédures présentées plus haut (et de bien d'autres encore) est licite. Toutefois, certaines combinaisons sont plus souvent utilisées, que ce soit pour des raisons historiques, théoriques ou expérimentales. Pour cette raison, les noms donnés sont souvent les noms des écoles historiques qui les ont popularisées -- mais gardons à l'esprit que ces schémas sont totalement indépendants de l'espace de recherche, alors que nous verrons que les écoles historiques travaillaient sur des espaces de recherche bien précis.

Schéma algorithme génétique générationnel (GGA)

Ce schéma utilise une sélection stochastique pour sélectionner exactement P parents (certains parents peuvent donc être sélectionnés plusieurs fois, d'autres pas du tout). Ces P parents donnent ensuite P enfants par application des opérateurs génétiques (avec probabilité donnée). Enfin, ces P enfants remplacent purement et simplement les P parents pour la génération suivante.

Schéma algorithme génétique stationnaire (Steady-state GA -- SSGA)

Dans ce schéma, un individu est sélectionné, généralement par tournoi, un second si le croisement doit être appliqué, et l'enfant résultant (après croisement et mutation éventuels) est réinséré dans la population en remplacement d'un "vieux" parent sélectionné par un tournoi inversé (le moins performant ``gagne").

Schémas stratégies d'évolution ((μ , + \square)-ES)

Deux schémas sont regroupés sous ces appellations. Dans les deux cas, l'étape de sélection est un tirage uniforme (on peut dire qu'il n'y a pas de sélection au sens darwinien). À partir d'une population de taille μ (notations historiques!), \square enfants sont générés par application des opérateurs génétiques. L'étape de remplacement est alors totalement déterministe. Dans le schéma (μ , \square)-ES, les meilleurs μ enfants deviennent les parents de la génération suivante, alors que dans le schéma (μ + \square)-ES, les meilleurs des μ + \square parents plus enfants sont les parents de la génération suivante.

5. Les réseaux de neurones

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

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

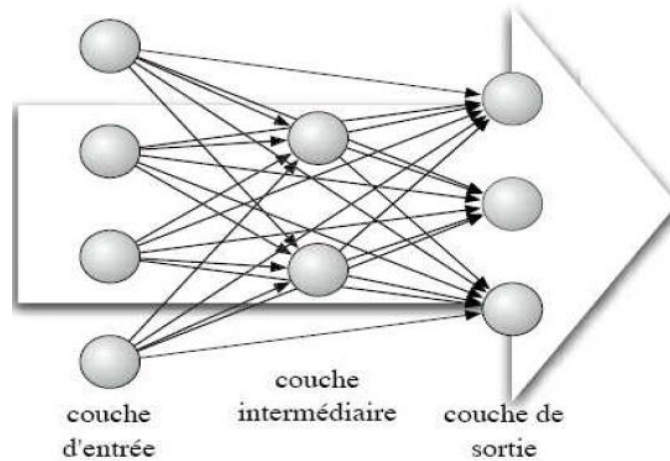


Figure 4. Exemple d'un réseau de neurone : le perceptron multi-couches.

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

5.1. Apprentissage

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

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

réseau une valeur rendant compte son adaptation à l'environnement. Le réseau apprend en essayant de maximiser cette valeur. Ici on parle de feedback évaluatif.

- L'apprentissage non supervisé ou auto-organisationnel. Cet apprentissage ne repose pas sur la notion de feedback. Il est basé sur des probabilités.

Conclusion

Dans ce premier chapitre, nous avons présenté des définitions des concepts de bases du domaine de notre recherche qui est la vie artificielle, ainsi que certain de ses techniques, nous envisageons présentés dans le deuxième chapitre les travaux qui portent sur l'évolution des créatures artificielles, ce qui est largement liés à notre modèle présenté par la suite.

Chapitre 2

Chapitre 2 : Les créatures artificielles

Introduction

Le développement des créatures artificielles s'est fait en plusieurs étapes. Dans les années 80-90, beaucoup de chercheurs ont travaillé sur la simulation du comportement de créatures artificielles. Partant d'une forme prédéfinie, l'évolution porte uniquement sur leur comportement. Après quelques années, on a commencé à faire évoluer la forme et le comportement des créatures pour qu'elles s'adaptent à leur environnement. [Sims 94]

Depuis une quinzaine d'années, un certain nombre de travaux tendent à produire des créatures basées sur une unité de base bien plus petite : la cellule artificielle. C'est en fait l'échelle de simulation des créatures qui a évolué au cours du temps : partant de la simulation du comportement d'une entité fixe, on en simule aujourd'hui le processus de développement cellulaire.

Il est intéressant de noter que l'évolution de l'échelle de simulation s'est faite conjointement à l'évolution de la puissance de calcul des machines : il faut aujourd'hui quelques secondes pour calculer le comportement macroscopique d'une créature, quelques heures pour évoluer la morphologie de celle-ci et quelques jours pour simuler son développement cellulaire. Si on revient une dizaine d'années en arrière, ces temps de calcul étaient décalés d'un niveau : il fallait quelques heures pour simuler le comportement d'une créature et quelques jours pour évoluer sa morphologie. Quant à simuler son développement cellulaire, cela semblait inaccessible !

La simulation comportementale a pour but de simuler le comportement d'agents dans un environnement virtuel. Elle est issue de l'informatique graphique. Les domaines d'application de la simulation comportementale sont principalement axés sur les jeux vidéo (pour les personnages non joueurs, présentes dans la figure 1).



Figure 1. Les Sims 3 (Electronic Arts) est un jeu très populaire dans lequel les personnages non joueurs évoluent dans un monde virtuel très complexe.

Les travaux sur les créatures artificielles sont classifiés en trois modèles :

- 1- L'Ontogenèse
- 2- La phylogénèse
- 3- La morphogénèse

Les travaux tendant à simuler le comportement des créatures, utilisent généralement des mécanismes d'apprentissage comme les réseaux de neurones et les systèmes de classeurs.

1. Définition de créatures artificielles

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 [FB89]

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.

2. Phylogenèse artificielle

La phylogenèse artificielle consiste à générer des créatures artificielles en réalisant l'évolution conjointe de la morphologie et du comportement. La morphologie est générée en utilisant des composants déjà prêts comme des blocs, des bâtonnets, des cylindres, comme organes de créatures. Le comportement d'une entité est défini en créant des liens entre ses capteurs et ses effecteurs. Ces derniers contrôlent les jointures entre les différentes parties de la créature et permettent ainsi de la faire bouger en modifiant les angles, vitesses, etc. appliqués à ces jointures. Dans ce type d'approche, on utilise généralement une fonction d'évaluation (fitness) pour diriger l'évolution. L'environnement simulé est tridimensionnel et intègre un simulateur physique. Les créatures sont plongées dans cet environnement et confrontées à des problèmes à résoudre qui constituent dans la plupart des cas leurs fonctions d'évaluation : par exemple si la tâche consiste à nager, la fonction d'évaluation serait la capacité de nager mesurée par exemple par la distance parcourue en nageant. [Dje10]

2.1. Les créatures de Karl Sims

Le mode de réalisation de phénotype d'une créature virtuelle est une hiérarchie de parties rigides en trois dimensions articulées.

La représentation génétique de cette morphologie est un graphe orienté de nœuds et de liaisons. Chaque graphique contient les instructions de développement pour la croissance d'une créature, et fournit une façon de réutiliser des instructions pour fabriquer des composants similaires ou récursifs dans la créature. Une hiérarchie de phénotype de pièces est réalisée à partir d'un graphique en commençant par un nœud racine et de la synthèse de parties définies à partir des informations de nœud tout en traçant par l'intermédiaire des liaisons du graphe. Le graphique peut être récurrent. Les nœuds peuvent se connecter à eux-mêmes ou en cycles pour former récursive ou fractale comme des structures. Ils peuvent également se connecter à un même enfant plusieurs fois pour faire les instances en double du même appendice.

Chaque nœud dans le graphe contient des informations décrivant une partie rigide. Les dimensions déterminent la forme physique de la partie. Un type commun détermine les contraintes sur le mouvement relatif entre cette partie et son parent en définissant le nombre de degrés de liberté de l'articulation et le mouvement autorisé pour chaque degré de liberté. Les différents types de joints sont autorisés: rigide, révoluté, torsion, universel, pliez-twist, twist-coude, ou sphérique. Joint-limites déterminent le point au-delà duquel la restauration des forces de ressort sera déployée pour chaque degré de liberté. Un paramètre récursif limite détermine combien de fois ce nœud devrait générer une partie de phénotype quand dans un cycle récursif. Un ensemble de neurones locaux est également inclus dans chaque nœud, et cela sera expliqué plus loin dans la section suivante. Enfin, un nœud comprend un ensemble de connexions à d'autres nœuds.

Chaque connexion contient également des informations. Le placement d'une partie enfant par rapport à son parent est décomposé en position, l'orientation, l'échelle et la réflexion, de sorte que chaque peut être muté de façon indépendante. La position de fixation est limitée à se trouver sur la surface de la pièce mère. Réflexions provoquent échelle négative, et permettent de sous arbres similaires, mais symétriques à décrire. Un drapeau de terminal ne peut causer une connexion à être appliqué que lorsque la limite est atteinte récursive, et permet queue ou comme composants d'occasion de se produire à la fin des chaînes ou des motifs répétitifs.

La figure 2 montre certaines topologies de graphes simples conçus à la main et résultant morphologies phénotype. On notera que les paramètres dans les nœuds et les connexions, tels que récursif limite ne sont pas représentés pour le génotype, même si elles affectent la morphologie du phénotype. Les nœuds sont anthropomorphique étiquetés comme «corps», «segment de la patte», etc., mais les descriptions génétiques ont fait pas de notion de catégories spécifiques de composants fonctionnels. [Sims94]

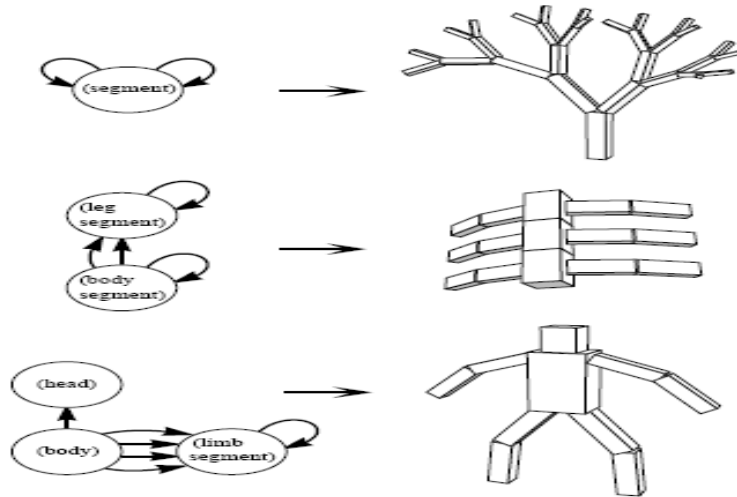


Figure 2. Exemples de graphes de génotype et des morphologies correspondantes de créatures de Karl Sims.

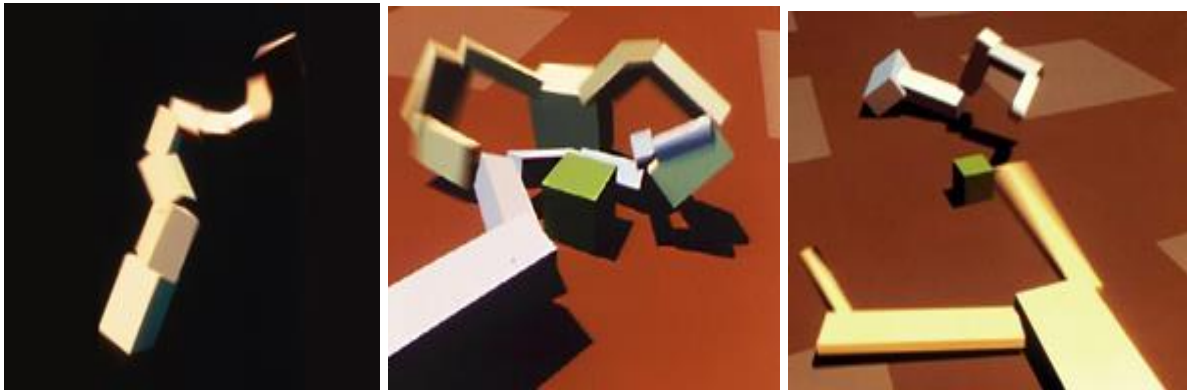


Figure 3. Créature de Karl Sims (à gauche : une créature nage ; à droite et au milieu : deux créatures en luttent pour s'emparer d'un cube).

2.2. Créatures de Gregory Hornby

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

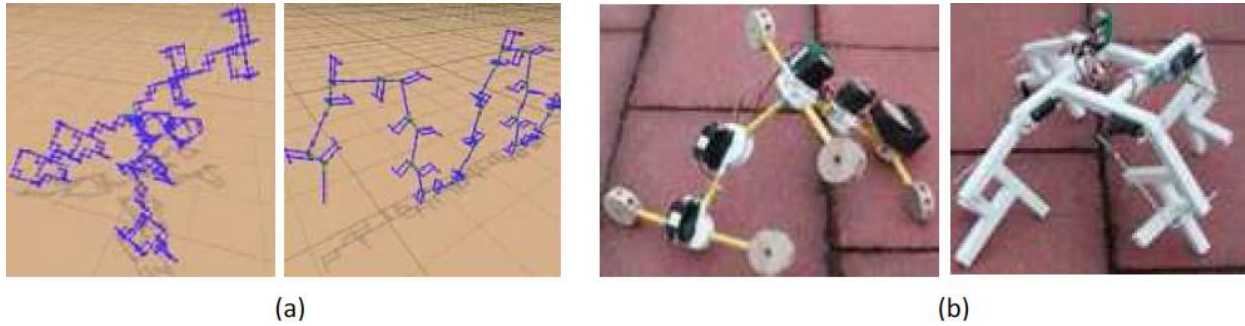


Figure 4. Créatures de Hornby. (a) Dans le simulateur. (b) Réalisation sous forme physique (robot).

2.3. Framsticks

Framsticks est un projet réalisé par Komosinski [Kom00, Kom03, Kom05, AK05], dans lequel les créatures sont assimilables à celles de Karl Sims. Elles évoluent cependant dans un environnement plus riche. Cet environnement contient des sphères énergétiques que les créatures doivent s'approprier pour survivre. De plus, un nombre plus grand de capteurs est spécifié (des capteurs pour le sens tactile, des capteurs pour le sens de l'équilibre, ..., etc.). En outre, l'évolution est dirigée soit par une fonction d'évaluation soit par la sélection naturelle.

La morphologie des créatures est composée de bâtonnets (sticks), chacun peut être spécialisé dans une fonction : assimilation, ingestion, ... Les mouvements sont accomplis par un réseau de neurones qui agit sur les jointures (muscles en rouge sur la figure 5).

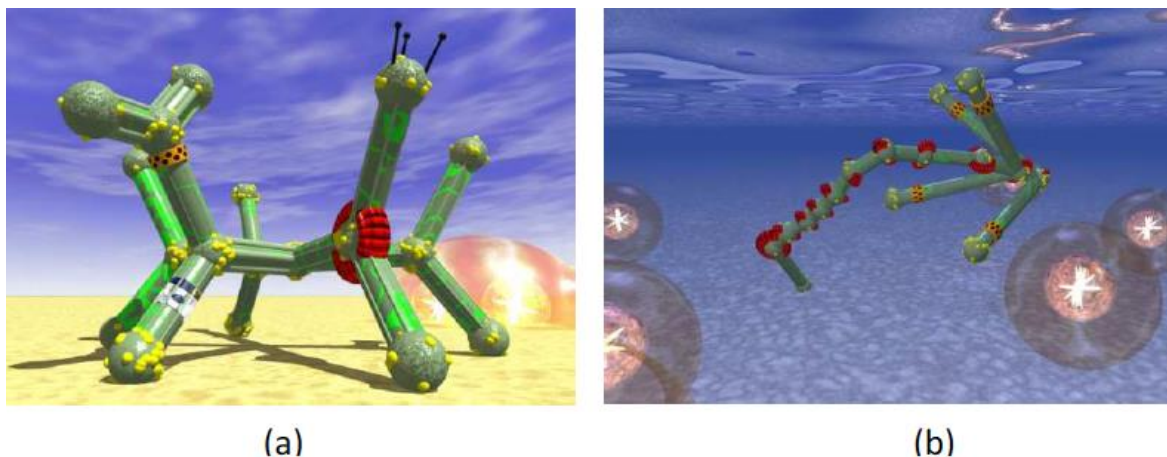


Figure 5. Framsticks. (a) créature terrestre. (b) Créature aquatique.

2.4. Créatures de Nicolas Lassabe

Lassabe [LLD07a, LLD07b, LLD07c] propose l'utilisation d'un nouveau type de contrôleur basé sur un système de classeurs. La morphologie de ses créatures est générée en utilisant un

graphe orienté comme dans le modèle original de Karl Sims.

Afin de valider son approche, il propose d'utiliser des environnements plus complexes. Pour cela, les créatures ont été confrontées à des situations mettant en jeu des objets statiques : escalier, murs, branchées. La figure 6 (a) présente des créatures essayant de monter un escalier. Ensuite, une autre série d'expérimentations a été menée dans le but d'étudier l'interaction entre les créatures et un objet mobile. La figure 6 (b) présente un ensemble de créatures se déplaçant avec un skateboard.

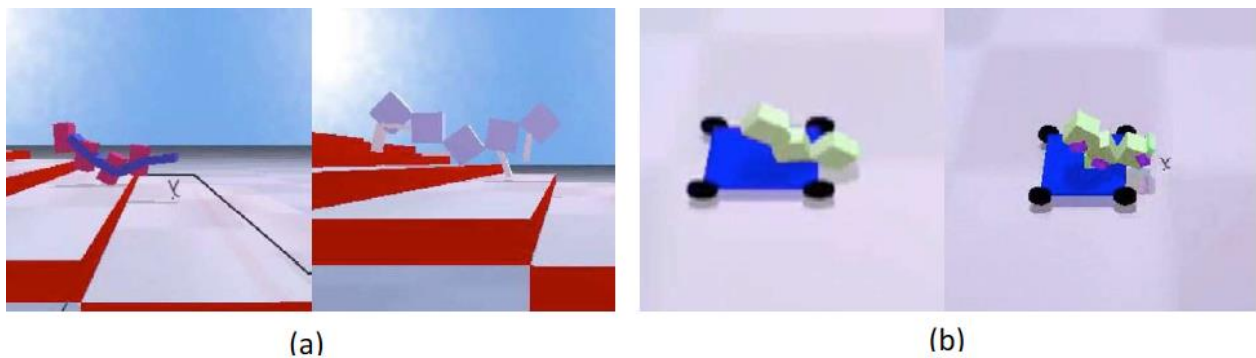


Figure 6. Créatures de Lassabe. (a) La montée d'escalier. (b) le déplacement avec un skateboard.

2.5. 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 7. (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 7. (2). (a)) ou dans l'espace (7. (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 7. (1). (c) et (2). (c)).

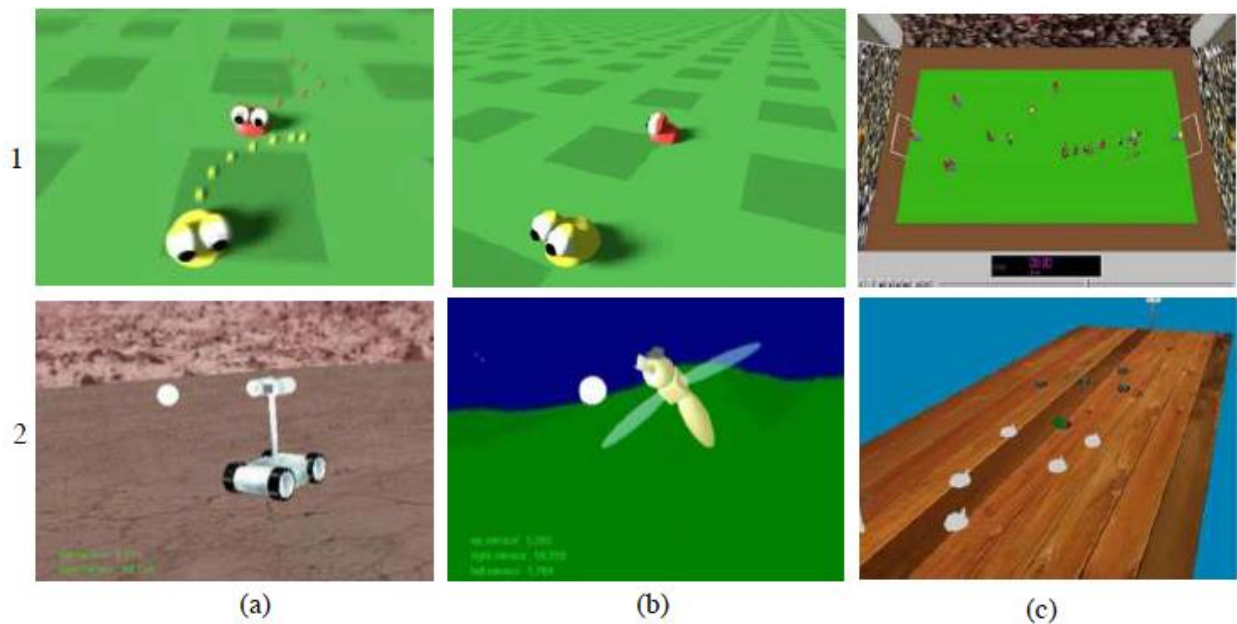


Figure 7. (1). (a) et (b) Cliff et Miller font co-évoluer deux créatures, la jaune doit apprendre à fuir, la rouge à rattraper la jaune, (c) Cédric Sanza. (2). (a) : Panzoli a fait évoluer un agent contrôlé par un réseau de neurones, qui doit être capable de suivre une lumière. (b) : Une autre créature de Panzoli, utilisant le même type de contrôleur mais étendu à des effecteurs différents. (c) Olivier Heguy.

3. Les écosystèmes

La notion d'écosystème, en général, est associée à l'écosystème naturel, où le biotique et l'abiotique se trouvent en équilibre sans intervention de l'homme. Toutefois, nous attirons l'attention sur le fait d'exister des écosystèmes artificiels créés par l'être humain et qui n'existent pas dans la nature. Dans notre approche que nous allons proposer dans le prochain chapitre, nous présentons un système qui utilise l'évolution artificielle pour le développement des comportements de créatures artificielles dans des écosystèmes virtuels.

3.1. Les écosystèmes naturels

Tout écosystème est constitué par l'association de deux composantes en constante interaction l'une avec l'autre : le biotope et la biocénose.

Le biotope est l'environnement physico-chimique, inerte, dit « abiotique » (littéralement « sans vie »). Il englobe toutes les composantes extérieures aux êtres vivants : le sol, l'air, l'eau, la lumière, le climat, etc.

La biocénose est la communauté vivante, constituée par la totalité des êtres vivants qui peuplent le biotope : bactéries, champignons, végétaux et animaux. [Encyclopédie Larousse]

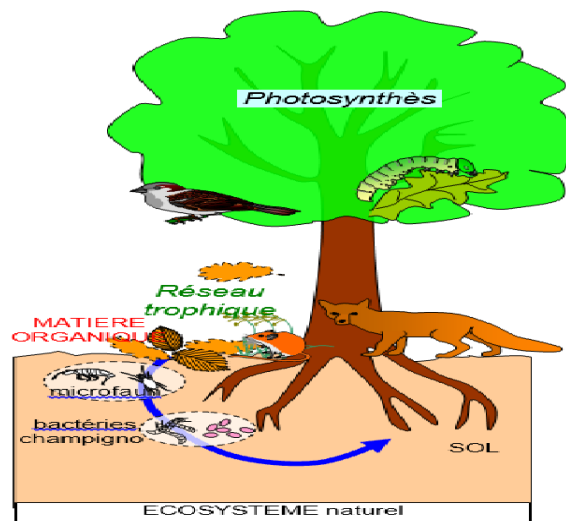


Figure 8. Écosystème naturel.

3.2. Les travaux sur les écosystèmes artificiels

Dans cette section, nous présentons l'ensemble des écosystèmes artificiels dont le but est de faire évoluer des créatures artificielles. Contrairement aux travaux qui se focalisent sur les créatures dont une seule est généralement présente à la fois dans l'environnement, les écosystèmes artificiels s'intéressent aux stratégies et aux adaptations qui émergent de l'interaction d'une multitude de créatures. Donc, un écosystème artificiel permet de faire co-évoluer un ensemble de créatures (plantes virtuelles comprises) en même temps dans un même environnement. L'évolution, ici, n'est pas guidée par une fonction d'évaluation (fitness), ce qui veut dire la sélection est naturelle. Les individus se disputent donc les ressources de l'environnement : A titre d'exemple, l'espace de la mémoire, comme dans Tierra [Ray91a; Ray91b]) ; la nourriture, comme dans Gene Pool [Ven96, Ven98a, Ven98b, Ven05] et Darwin's Park [BL07a, BL07b, BL08a, BL08b, BL08c]), etc. Ainsi, un individu meurt quand il n'a plus d'énergie ou d'espace pour survivre. Les individus les plus adaptés peuvent cependant survivre et faire passer leurs gènes à leurs descendants.

L'intérêt d'un tel système est d'essayer de comprendre et d'analyser l'évolution d'un écosystème naturel. En outre, L'avantage est que l'évolution est laissée libre et peut aboutir à des résultats intéressants tels que dans les travaux présentés ci-après.

3.2.1. Tierra

Durant les années 1991-2001, Thomas Ray développe le projet Tierra [Ray, 91a][Ray, 91b], en s'inspirant des programmes CoreWare et Darwin [McI et al.71]. Il propose de réaliser un écosystème dont l'environnement est la mémoire de l'ordinateur et les individus des programmes. Les programmes se disputent les ressources du microprocesseur et de la mémoire afin de survivre. Ils évoluent par mutation, croisement et répliation. Le génotype et le phénotype sont confondus car ils sont représentés par le programme ce qui limite l'évolution par rapport à une représentation indirecte. Ray espérait que simplement avec la mutation et le croisement il puisse produire au bout d'un certain temps un programme avec la capacité de s'auto-reproduire. Il abandonna finalement cette idée à la vue de la faible probabilité que cela apparaisse. Il initialisa donc son environnement avec des programmes ayant la capacité de se dupliquer. Ces organismes digitaux qui ne sont soumis à aucune fonction d'évaluation se battent pour la conquête des ressources de l'ordinateur comme la mémoire et le processeur.

Durant les simulations il observe des stratégies différentes. Certains programmes continuent à se répliquer comme le programme originel. D'autres n'ont plus la capacité de se dupliquer mais parasitent d'autre programme pour profiter de cette fonction. Il observe aussi que des programmes parasites sont parasités à leur tour par d'autres programmes

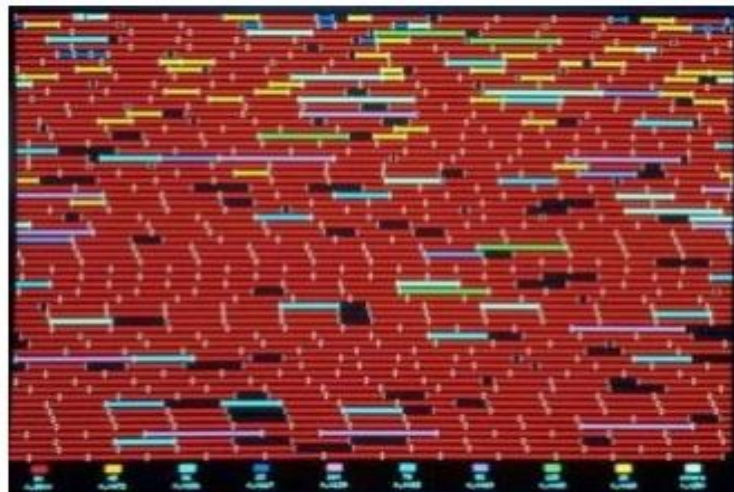


Figure 9. L'écosystème réalisé par Thomas Ray dont on voit ici une représentation de la mémoire peuplé de programmes. Ceux-ci ont des couleurs différentes selon leur taille, les programmes les plus petits en jaune sont des parasites qui profitent des capacités de répliations des autres programmes.

3.2.2. Gene Pool

Gene Pool est un écosystème aquatique bidimensionnel peuplé de créatures appelées swimbots [Ven96, Ven98a, Ven98b, Ven05]. La morphologie des créatures est composée de rectangles aux angles arrondis qui sont reliés par des articulations. Le projet Gene Pool réalisé par Jeffrey Ventrella, l'environnement de Gene Pool est physique et simple, permettant de faire évoluer un grand nombre de créatures simultanément. Chaque créature possède un état qui est décrit par un diagramme d'état comportant quatre possibilités et deux actions: recherche d'un partenaire ayant envie de se reproduire, se diriger vers le partenaire, rechercher de la nourriture, se diriger vers la nourriture, se reproduire et manger.

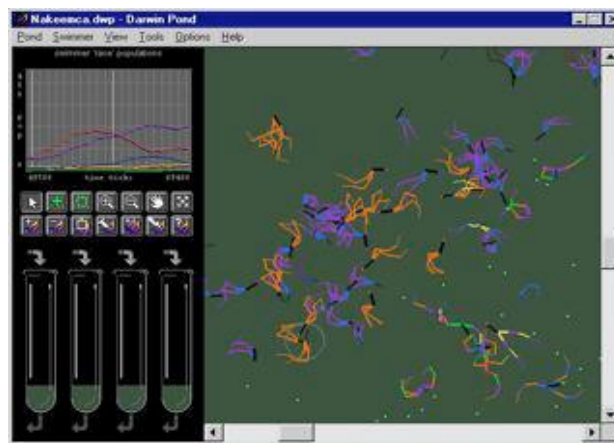


Figure10. Gene Pool, l'écosystème réalisé par Jeffrey Ventrella

3.2.3. Life Drop

Life Drop est un écosystème simulant une goutte d'eau artificielle, habitée par de petites créatures. Ces dernières sont inspirées des Biomorphs. L'environnement simule les forces physiques et prend en compte des paramètres comme le PH et la fluidité de l'eau. Chaque créature est un agent autonome décrit par un modèle en couches inspiré de l'architecture de Brooks. Le modèle implémenté comporte cinq couches : Génotype : composé de 32 bits codant toutes les caractéristiques de la créature; Métabolisme : contrôle les cycles essentiels comme le cycle de vie, de croissance, etc. ; Dynamiques : contrôle "l'incarnation" de la créature simulant l'impact "des conditions physiques" comme la fluidité de l'eau ; Comportement réactif : contrôle des comportements réactifs de base tels que l'évitement d'obstacle, la fuite d'un danger, etc. ; Comportement cognitif : contrôle des comportements tels que le choix d'un partenaire. En plus des paramètres génétiques, chaque créature obtient un ensemble additionnel de paramètres

après son développement. Ceux-ci incluent -entre autres- ses états comme “vivante“, “affamée“, “fécondée“, “stressée”...etc. Le modèle physique qui permet aux créatures de se déplacer dans l’environnement s’inspire de celui de [Rey ,99]. Les créatures se reproduisent entre elles suivant leur affinité génétique et leur niveau de stress.

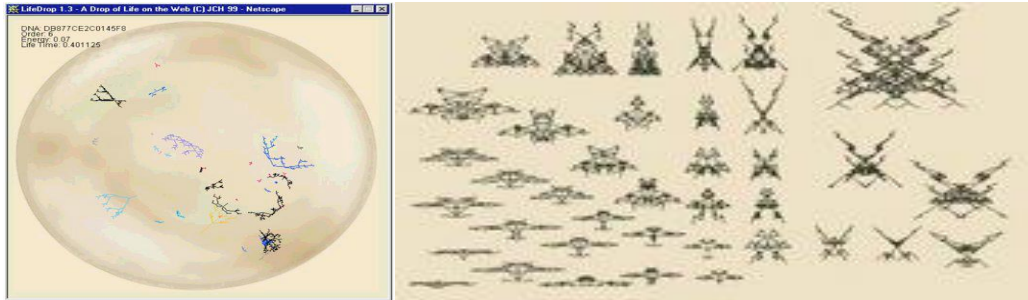


Figure 11. Evolution d’un écosystème artificiel dans une goutte d’eau. Il est peuplé de créatures appelées Biomorph.

3.2.4. Darwin’s Park

Darwin’s Park [BL07a, BL07b, BL08a, BL08b, BL08c] est une plateforme de simulation d’écosystèmes de plantes virtuelles. Chaque plante possède une morphologie et une physiologie. Les morphologies aériennes et souterraines sont chacune définie par un L-système. Quand à la physiologie, elle est définie par un modèle transport-résistance [Tho98] qui décrit les processus d’assimilation et d’allocation de ressources par un système d’équations différentielles.

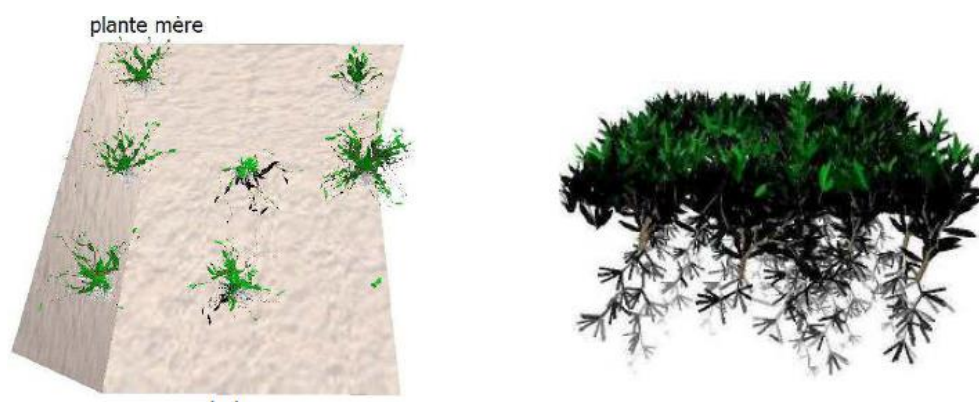


Figure 12. Darwin’s Park. (a) jardin virtuels. (b) Compétition sur le terrain virtuel.

3.2.5. PolyWorld

Il existe d’autres travaux sur les écosystèmes artificiels, parmi eux nous citons : PolyWorld

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

Conclusion

Dans ce chapitre nous avons introduit le concept de créatures artificielles, une citation de quelques travaux de créatures artificielles dans le domaine de vie artificielle, nous avons présenté aussi certains travaux sur les écosystèmes artificiels.

Chapitre 3

Chapitre 3 : Conception de système

Introduction

Dans les chapitres précédents, nous avons fait une étude théorique sur le domaine de recherche de vie artificielle d'où on a pu établir les points essentiels dans la création d'un modèle d'évolution de créatures artificielles dans un monde virtuel, les détails de ce modèle sont présenté dans le chapitre courant, et qui est organisé comme suit:

Une conception globale décrivant comment fonctionne notre modèle d'après ses principaux objectifs et quels sont les points qui entrent dans sa modélisation

Une deuxième partie consiste à détailler la conception précédente en décrivant les détails de chaque module de la conception.

Nous définissons par la suite les paramètres utilisés dans la simulation (soit dans le module de modélisation 3D ou bien dans le setup de l'algorithme génétique).

Dans ce chapitre nous suggérons un modèle nommé évolution de créatures fourrageurs qui a comme but, de trouver les meilleures créatures fourrageurs qui ont la plus grande espérance de vie.

1. L'objectif de travail

L'objectif de ce modèle est d'étudier l'aspect écologique tel que l'évolution, la survie et la reproduction des créatures de fourragement dans un écosystème artificiel simulé par ordinateur. Toutes les créatures ont des règles qui s'appliquent à leur comportement (c.-à-d. règles locales), et chacune des créatures va tenter de survivre et de prospérer.

2. conception global

Dans ce travail, on s'intéresse au problème de poursuite effectué par un fourrageur envers des nourritures dans un environnement virtuel. Nous avons choisi un simple environnement qui est

un espace limité, cet environnement est divisé en une grille à deux dimensions (structure matricielle), cet environnement contient deux types de créatures; une créature fourrageur et un ensemble fixe de nourritures.

Les créatures sont un ensemble des fourrageurs, il se déplace, consomme des nutriments et meurt. Pour réaliser ce processus, les fourrageurs ont besoin d'énergie pour survivre.

La figure suivante montre l'architecture globale de notre modèle ou l'environnement virtuel simulé qui contient des nourritures et des fourrageurs qui sont soumise à passer par un processus évolutionnaire pour interagir avec l'environnement.

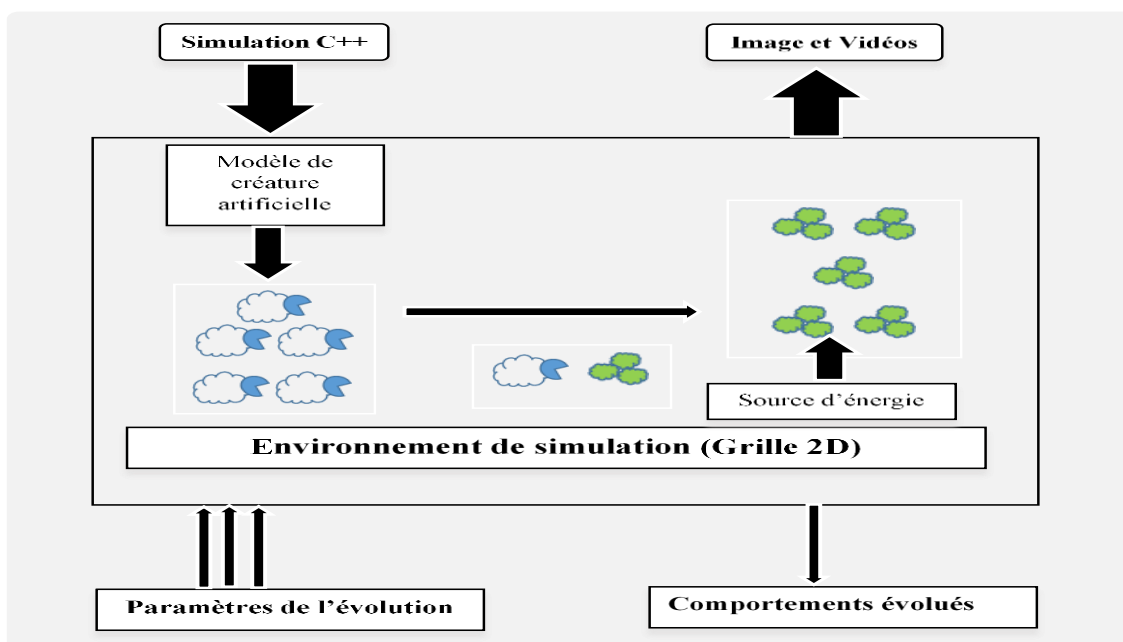


Figure 1. Architecture globale du système.

3. Conception détaillé

3.1. L'environnement de simulation

L'environnement est une grille de 2D qui est l'environnement local. Notre modèle est représenté graphiquement dans une scène 3D limitée (terrain plat) sans obstacle, 2 entités simulées une animés (fourrageur) et l'autre fixe (nourriture) tel que présenté dans la figure les lieux de la scène sont en 2D représenté sur une grille.

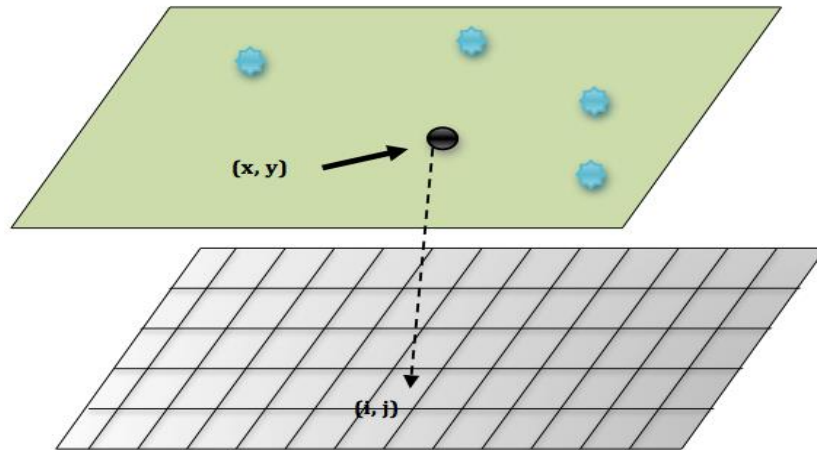


Figure 2. La créature fourrageur qui se déplace à travers l'espace 2D continu au-dessus d'une grille discrète de l'environnement, la cartographie de sa position de coordonnées (x, y) d'un indice (i, j) .

Représentation

1. Graphique

La représentation graphique de la créature fourrageur et nourriture :

On présente le prédateur par exemple par une sphère (noir).

On présente la proie par exemple par une étoile (bleu).

2. Logique

La représentation logique sera comme suit :

- Le vide par (0).
- Le fourrageur par (1).
- La nourriture par (2).

3.2. Les créatures artificielles

Les créatures artificielles simulées dans cet environnement sont le point focal de notre modèle. La capacité des créatures pour obtenir des ressources et éviter la mort détermine l'abondance et la dynamique de chaque population.

Chaque espèce est définie en termes de:

A. Caractéristiques hétérogènes (les variables)

A.1. Variables d'état individuels. Variables d'état des créatures comprennent:

1. de la biomasse, l'unité utilisée pour décrire le contenu énergétique d'un animal ou la taille,
2. un lieu dans l'environnement de simulation, à savoir dans une "cellule" à un discret (x,y), et
3. un âge qui est l'âge initial à la naissance (Age Initial) plus le nombre de mesures qu'il a vivant.

A.2. Stratégie comportementale Individuel. La stratégie comportementale d'un individu détermine comment cet individu répond à toutes les combinaisons possibles d'états perceptifs générés par les capacités sensorielles d'un individu. Les paramètres qui dictent la stratégie comportementale ne sont pas seulement potentiellement variables entre les individus d'une espèce, mais ils sont sujet à l'évolution, et donc leur répartition peut changer au fil du temps.

B. Les caractéristiques homogènes (fixes)

Caractéristiques générales. Plusieurs paramètres définissent les caractéristiques générales de chaque espèce. Par exemple, les paramètres qui décrivent le métabolisme (la perte de la biomasse), la mort en raison de certains facteurs, et certaines caractéristiques de recherche de nourriture, ces paramètres sont constants et ont la même valeur pour tous les individus d'une espèce donnée.

3.2.1. L'entité source d'énergie (nourriture)

Les nourritures sont représentées de façon aléatoire et fixe dans un environnement 3D.

3.2.2. L'entité fourrageur

Ce modèle a comme but principal d'évoluer les comportements d'une créature fourrageur qui doit trouver une des nourritures pour manger et qui vient dans le même environnement qu'elle, la nourriture choisie doit être la plus proche.

Les mouvements de la créature fourrageur sont réalisés sur une Grille 2D en utilisant sa propre structure de données et son métabolisme, ainsi que l'ensemble des règles définit ci-dessus.

A chaque pas de temps, tous les individus d'une population doivent choisir une action à exécuter à partir du répertoire d'actions. Le répertoire comprend trois actions, qui sont : ne rien faire, manger (après éventuellement le déplacement), et se déplacer. En particulier, quand un individu exécute l'action ne rien faire, il reste à son emplacement actuel (c.-à-d. la même cellule). Lorsqu'un fourrageur mange implique l'incrément de la biomasse de ce fourrageur.

Nous avons des caractéristiques aux niveaux espèce qui déterminent les différentes capacités de recherche de nourriture définies par des paramètres qui spécifier la gamme qu'un fourrageur peut rechercher et capturer leurs nutriment. Tout cela va permettre des stratégies plus efficaces de recherche de nourriture.

La possibilité d'examiner l'influence de ces paramètres est d'un grand intérêt, car elles influent sur l'efficacité alimentaire qui affecte fortement la dynamique des chaînes alimentaires.

Ce serait une simple question de modifier les paramètres du système (par évolution) de sorte que la consommation de nourritures dans une période de temps donné est dépendante de la densité et influencé par la consommation de nourritures potentielles à cette période de temps.

3.2.2.1. Les propriétés des fourrageurs

Les créatures fourrageur vont être évoluées pour, d'une part garantir leur survie dans l'environnement, et un autre part pour étudier le comportement de recherche de nourriture afin de voir l'effet des stratégies comportementales étudiées sur l'évolution des fourragement.

3.2.2.1.1. Le métabolisme

Le métabolisme primaire est responsable des cycles essentiels de la croissance, le développement, la reproduction et gère les états comme la faim ou le stress, la majorité de ces cycles sont affecté par les gènes et les interactions entre la créature et son mouvement.

Nous modélisons un environnement virtuel avec une population de telles créatures pour étudier l'évolution de certains comportements tel que recherche des nourritures limités par leur consommation d'énergie dans un environnement simulé.

Les créatures doivent survivre tout en maintenant leur niveau d'énergie qui est calculée en fonction de leur métabolisme.

La biomasse d'une créature est sa valeur initiale à la naissance (Biomasse initiale), ainsi que le montant cumulé des ressources dont elle a assimilés en mangeant, moins le montant perdu "métabolisme" et la reproduction. Un individu doit augmenter la biomasse afin de reproduire et d'éviter la mort par inanition

Le métabolisme représente la biomasse qui est perdue à chaque pas de temps par un facteur constant (perte de biomasse) indépendamment du comportement.

Action	La biomasse
Nouvel individu	25
Déplacer	-capacité de déplacement
Manger	+ (capacité de consommation*100)

Tableau 1. Représentant le métabolisme de la créature prédateur.

3.3. Algorithme de mouvement de la créature prédatrice

Les procédures ne rien faire, manger et déplacer qui représente les actions à être effectuées par des individus fourrageurs sont multipliées par les probabilités d'actions qu'on obtient à chaque pas d'animation par l'algorithme génétique qui sont calculés en fonction de l'état environnemental perçu.

- Perception globale. Explorer et tester l'existence ou non d'une nourriture au minimum.
- Perception locale. Tester les 8 cellules voisines de la cellule occupée par l'individu pour trouver la direction à prendre par cet individu qui est dépendante du positionnement de la nourriture trouvée.

Les actions de fourrageur

- Manger () : Prendre la position de la nourriture et l'effacer de l'environnement.
- Déplacer () : Changer de direction et de position pour se déplacer envers la nourriture la plus proche, les directions de déplacement possibles sont : Aller droite, Aller gauche, Aller haut, Aller bas.

Début

Tant que (il y a des nourritures dans l'environnement) faire

Calculer la distance ();

Si (la plus proche nourriture est en haut) alors

Déplacer vers haut ;

Si (la plus proche nourriture est en bas) alors

Déplacer vers bas ;

Si (la plus proche nourriture est en gauche) alors

Déplacer vers gauche ;

Si (la plus proche nourriture est en droite) alors

Déplacer vers droite ;

Si (le fourrageur atteint la nourriture cible) alors

Manger ();

Fin tant que.

Fin.

3.4. L'évolution

Désigne la transformation des espèces vivantes qui se manifeste par des changements de leurs caractères génétiques et morphologiques au cours des générations ou même comportementaux permettant à ces espèces de mieux s'adapter aux changements environnementaux.

3.5. Algorithme génétique

Les algorithmes génétiques sont basés sur la théorie de l'évolution de Charles Darwin qui considère que les êtres les plus adaptés à leur environnement survivent et propagent leurs caractéristiques. L'information à optimiser est représentée sous la forme de chromosome. Un génotype est le génome représentant une créature. Un phénotype représente les mouvements réalisés par la créature durant sa simulation.

Les algorithmes génétiques fournissent des solutions aux problèmes n'ayant pas de solution calculables en temps raisonnable de façon analytique ou algorithmique.

Sous cette méthode, des milliers de solutions (génotype) plus ou moins bonnes sont créées mimant l'évolution des espèces : les plus "adaptés", c'est-à-dire les solutions au problème qui sont les plus optimales survivent davantage que celles qui le sont moins et la population évolue par générations successives en croisant les meilleures solutions entre elles et en les faisant muter, puis en relançant ce procédé un certain nombre de fois afin d'essayer de tendre vers la solution optimale.

3.5.1. Pseudo code de l'algorithme génétique

Les étapes d'un algorithme génétique utilisé pour notre modèle sont les suivantes :

1. Création d'une population aléatoire (chromosome aléatoire) ;
2. Lancer la simulation des individus de la population sur la période de simulation (exécuter les variables d'état du chromosome).
3. Calculer la performance de chacun des individus en sélectionnant les meilleurs individus (ceux qui ont répondu avec la meilleure façon);
4. Choisir ces individus sélectionnés et voir leur paramètres et conditions pour lancer la reproduction ;
5. Effectuer des changements sur les chromosomes des individus en appliquant les deux opérations de croisement et de mutation.
6. Remplacer les parents par ses fils ;

Rappelons que dans ce travail, le but de l'algorithme génétique est d'évoluer une créature

fourrageur pour des comportements de poursuite, un bon génotype est un ensemble de paramètres qui provoque une créature pour effectuer le comportement souhaité, ce qui est de se nourrir des sources le plus que possible.

Chromosome

Un chromosome est une suite de génome. Chaque individu est représenté par un chromosome, et une population est un ensemble d'individus.

Les chromosomes sont des suites de gènes sont représentés par des tableaux de 8 cellule et les individus de notre espace de recherche sont représentés par des tableaux de chromosomes.

La création de la population

La première étape de l'algorithme génétique consiste à créer la population initiale. Cette première génération est choisie aléatoirement parmi les valeurs possibles de la population. La population à faire évoluer, pour trouver une solution au problème posé.

Sélection

Pour déterminer quels sont individus les plus enclins pour 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.

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 où un seul point de croisement est aléatoirement choisi, et la première partie du premier chromosome (parent 1) est jointe avec la deuxième partie du deuxième chromosome (parent 2), comme illustré dans figure ci-dessus.

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 est habituellement donnée par le concepteur au début du traitement. 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.

Conclusion

Ce travail met l'accent sur l'étude des comportements évolutifs qui visent à trouver le comportement du fourrageur en recherche des nutriments. L'étude présentée dans ce chapitre constitue la première étape réalisée à la construction d'un écosystème virtuel simulé. Nous visons à l'étude d'un modèle de fourrageur (avec l'évolution de population). Nous allons étudier la dynamique d'un système fourrageur, l'objectif du fourrageur est de trouver et capturer sa nourriture aussi efficacement que possible.

Dans le prochain chapitre nous allons tenter à présenter quelques une des expérimentations réalisées et les résultats obtenus en simulant des créatures artificielles et avec l'utilisation d'un algorithme génétique.

Chapitre 4

Chapitre 4 : Implémentation et résultats

Introduction

Dans ce chapitre, nous allons décrire la mise en œuvre des différentes étapes de notre système conçu dans le chapitre précédent. Nous commençons par présenter le langage de programmation qui a été utilisé dans le développement de notre application, ensuite nous détaillons les structures de données utilisées. Et enfin nous présentons l'algorithme principal que sur lequel le système se déroule, qui explique comment les créatures se déplacent dans l'environnement 3D et comment elles évoluent pour atteindre les sources d'énergie, ainsi que le fonctionnement du métabolisme. La dernière partie concerne les résultats obtenus, qui sont les comportements des créatures artificielles, une petite discussion est donnée à la fin de ce chapitre.

1. Le langage de programmation

1.1. Présentation de langage utilisé

Le langage de programmation utilisé est le C++. Avec le logiciel Borland C++ Builder v6.0 qui est le nouvel environnement de développement basé sur C++ proposé par Borland. Borland a repris l'interface et la bibliothèque de composants visuels pour l'adapter de puis le langage Pascal Orienté Objet vers C++ répondant ainsi à une large faction de programmeurs peu enclins à l'utilisation du Pascal qu'ils jugent quelque peu dépassé.

Le choix du langage utilisé repose sur plusieurs critères. Parmi ces critères, on peut trouver la connaissance du langage qui nous a conduits choisir l'environnement C++ Builder. On a essayé d'exploiter les caractéristiques de ce langage (unité, objet, fonction etc.....) pour implémenter les différents composants du système. Ce qui implique que chaque composant peut être implémenté en termes d'une unit, d'un objet ou d'une fonction. Et nous avons renforcé C++ Builder par la bibliothèque graphique Open GL.

1.2. La Bibliothèque graphiques OpenGL (Open Graphic Library)

OpenGL est une spécification qui définit une API multiplateforme pour la conception d'applications générant des images 3D (mais également 2D). Elle utilise en interne les représentations de la géométrie projective des objets. L'interface regroupe environ 250 fonctions différentes qui peuvent être utilisées pour afficher des scènes tridimensionnelles complexes à partir de simples primitives géométriques. Du fait de son ouverture, de sa souplesse d'utilisation et de sa disponibilité sur toutes les plates-formes, elle est utilisée par la majorité des applications scientifiques, industrielles ou artistiques 3D.

1.3. Motivation du choix du Builder C++

Nous avons choisi ce logiciel à cause des services qu'il nous offre, qui sont :

- Permet de concevoir l'interface utilisateur de manière particulière et interactive.
- Un langage spécifique pour la simulation multi-agents.
- Un langage adapté à modéliser les systèmes complexes.
- Un environnement modelant et programmable pour simuler des phénomènes sociaux et normaux.

2. Implémentation

La première étape dans la réalisation de notre système est le choix des différentes structures de données nécessaires pour l'implémentation des différents objets.

2.1. L'environnement

Le module environnement est chargé de la construction de l'environnement, il produit deux types de représentation: représentation matricielle et représentation graphique.

La représentation matricielle de l'environnement est représentée par une matrice. Chaque élément de cette matrice présente une cellule de l'environnement. Les cellules ont la structure suivante :

```
Typedef struct cellule {  
    Float x ;  
    Float y ;  
    Int v ; // si v= 1 existe fourrageur  
           // si v= -1 existe nourriture  
}cell ;
```

2.2. Algorithme de construction de l'environnement

Entrées : coordonnées de la scène (fournies par l'utilisateur)

Sortie : représentation matricielle, représentation graphique (des positions et rendu)

Début

```
Dessiner scène () ;  
  
Représentation matricielle ;  
  
Représentation graphique ;
```

Fin

L'environnement est un objet dont la classe est la scène.

```
Classe scène  
Début  
    // attributs  
    Centre ;  
    Grille ;  
    Les objets de la scène ;" "  
    // méthodes  
    Dessiner () ;  
    Changement de vue () ;  
Fin
```

a. **Dessiner scène () :**

Après l'exécution de cette méthode, on aura l'affichage de la grille et les différents objets de notre scène qui sont les fourrageurs et les nourritures.

```
Début  
[  
  Découpage de la scène ;  
  Pour (tous les objets de la scène) faire  
  [  
    Affichage des objets () ;  
  Fin  
] Fin
```

2.3. Les structures des Créatures artificielles

La structure du fourrageur

La structure de données du fourrageur se compose de plusieurs variables selon la description suivante :

```
Structure fourrageur  
{  
  Float x ;  
  Float y ;  
  Float quantite ;  
  Chrom ch ;  
} tind ;
```

```
Structure chrom  
{  
  Int capacite_detection [8] ;  
  Int capacite_deplacement [8] ;  
  Int capacite_consomation [8] ;  
} chrom ;
```

La structure des sources nourriture

La structure de données des sources de nourriture aussi se compose de plusieurs variables selon la description suivante :

```
Structure fourrageur  
{  
  Float x ;  
  Float y ;  
  Float quantite ;  
  } P ;
```

2.4. Description de l'application

Nous commençons notre expérimentation par l'initialisation des paramètres de simulation, tel que le nombre de créatures artificielles, leurs lieux, leurs quantités d'énergies initiales et le nombre de sources d'énergie dans l'environnement 3D de simulation. On lance la simulation par effectuant la boucle de l'algorithme génétique qui consiste à évaluer les individus lancés initialement à la fin de chaque génération qui durent 50 secondes, attribuer des notes qui sont calculés par la fonction de fitness, ensuite reproduire les créatures artificielles pour obtenir les meilleur au cours des générations, et ce sont ceux qui se comportent mieux en tant que créatures fourrageurs.

Chaque fourrageur a une capacité de consommation d'énergie et une capacité de déplacement, et lorsque les fourrageurs se déplacent leurs quantités d'énergie vont diminuer, et lorsqu'ils mangent la quantité d'énergie va augmenter.

La fonction de fitness utilisé est celle de la quantité d'énergie d'une créature artificielle, cette valeur signifie que ces créatures ou individus de la population à évoluer a une grande chance de survivre et les meilleur capacité pour se déplacer vers les sources d'énergie existante dans l'environnement, après le processus de sélection les individus choisi (sélectionné) vont être utiliser pour former les nouveaux individus par les deux opérations génétiques qui sont le croisement et la mutation, dès qu'on obtient les fils on va refaire le processus d'évaluation et de sélection jusqu'à obtention des meilleurs individus et que toute la population convergent vers les meilleurs solution.

2.5. Présentation de l'algorithme

L'algorithme génétique et les détails de son utilisation est présenté dans l'algorithme suivant :

Start :

Pour (chaque fourrageur) **faire**

Déplacer de façon aléatoire dans l'environnement ;

Si (il y a des nourritures dans l'environnement) **alors**

Calculer distance entre fourrageur et nourriture ;

Déplacer vers plus proche nourriture ;

manger nourriture ;

Fin si

Fin pour

Si (temps d'exécution de la génération == 20 secondes)

Si (tous les fourrageurs son consomme bien l'énergie et a le meilleur chromosome) **alors**

Arrêter l'exécution ;

Sion

Sélection meilleur individu comme parent 1 et en sélection parent 2 aléatoirement ;

Croisement ;

Mutation ;

Aller à Start ;

Fin Sinon

Fin Si

3. Résultats obtenus

3.1. Interface graphique

La barre d'outil

Elle regroupe des boutons où chacun a sa fonctionnalité comme présentée dans la figure 1.

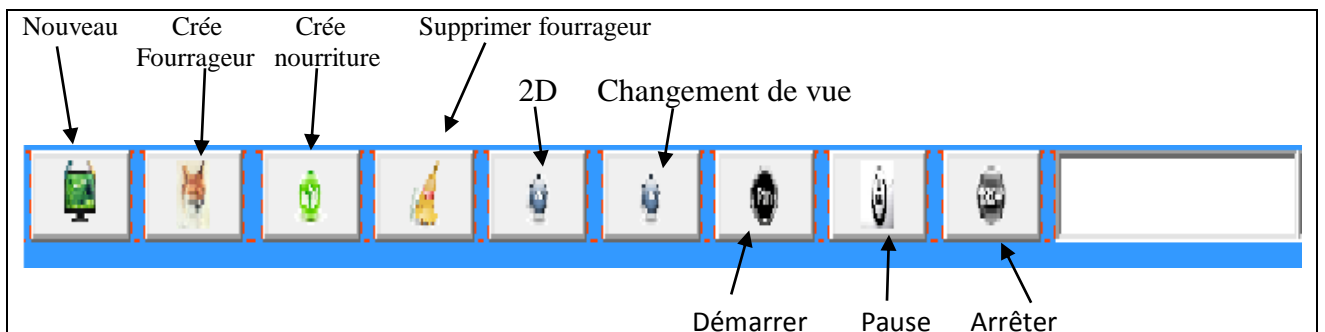


Figure 1. Représentation graphique de barre d'outil.

Zone d'affichage

1. Initialisation de l'environnement (2D et 3D)

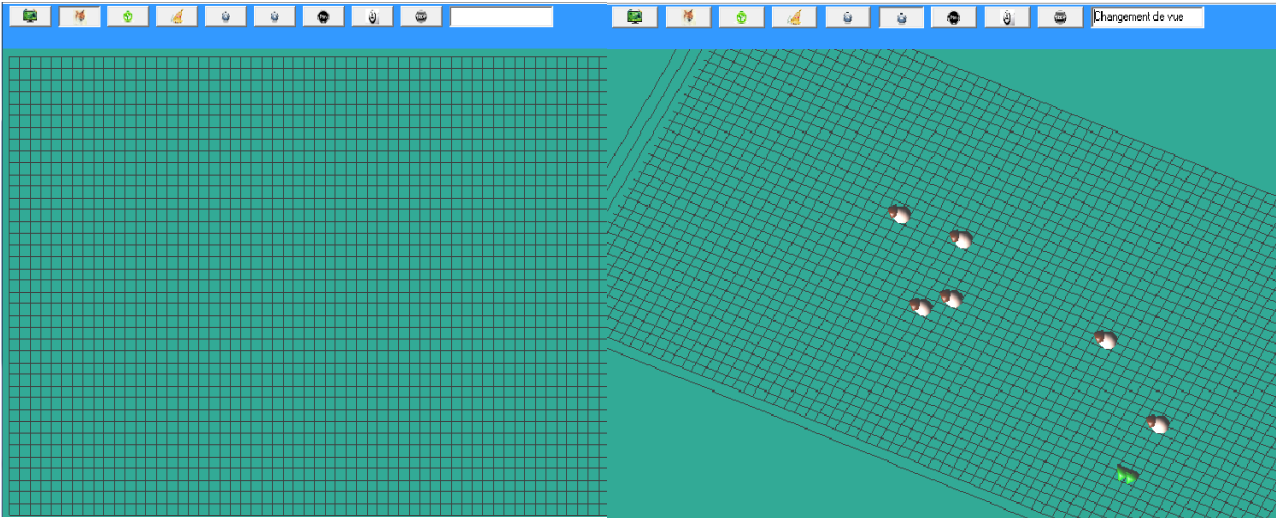


Figure 2. (a) : représentation graphique de l'environnement en 2D. (b) : représentation graphique de l'environnement en 3D.

2. Lancement de l'exécution

La première étape de l'exécution est la création des fourrageurs et des nourritures dans l'environnement (figure 3).

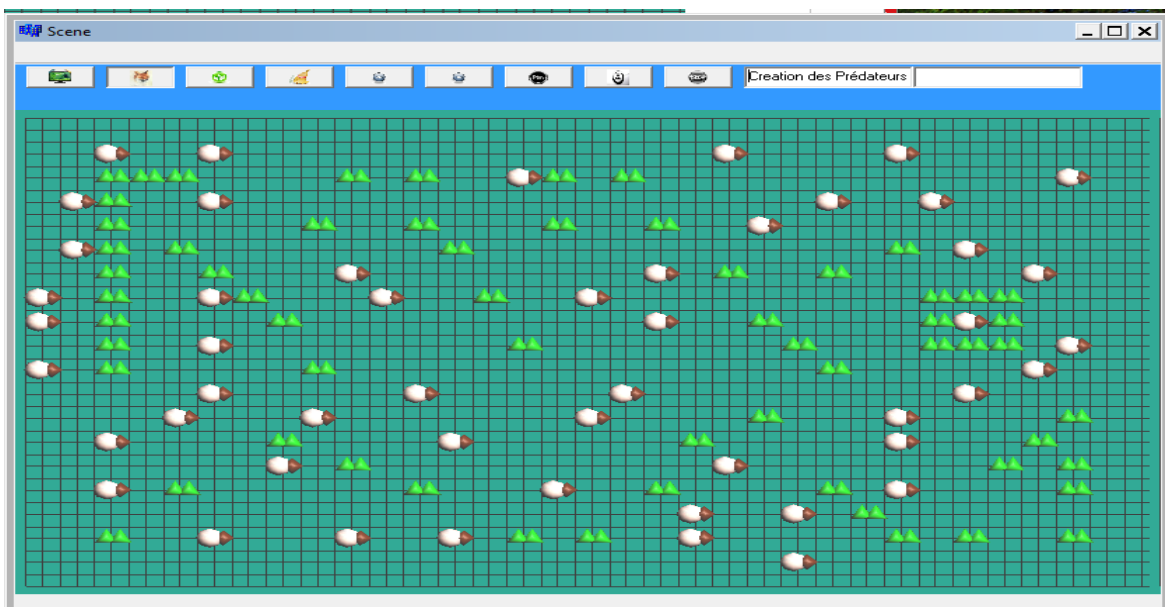


Figure 3. Création des créatures dans l'environnement de façon aléatoire.

Ensuite en va lancer l'animation (figure 4).

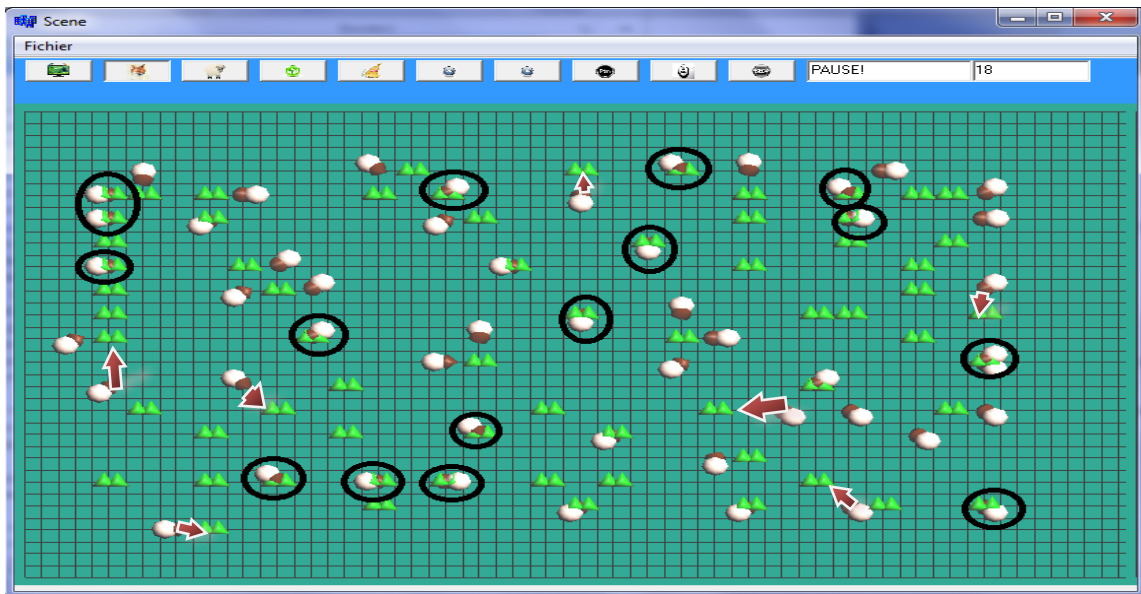


Figure 4. Les fourrageurs changer leur direction vers la plus proche nourriture, en cercle noir des fourrageurs sont arrivé au but (ils sont mangent).

Les fourrageurs commencent à percevoir son environnement, après avoir exécuté l'algorithme présenté ci-dessus, il se déplace en se dirigeant vers la plus proche source de nourriture existante dans l'environnement. Nous constatons qu'il y a des fourrageurs sont arrivés à la cible rapidement par rapport aux autres, et cela signifie que ses fourrageurs a des capacités meilleur de chromosome que les autre.

A la fin de chaque génération, on obtient le meilleur individu, en raison de la quantité d'énergie, mais pas nécessairement qu'il consomme un grand nombre de source d'énergie ou se déplacer en plusieurs endroits est le meilleur individu, alors en peut dire que le critère d'arrêt est de trouver un individu qui a une valeur de capacité de consommation et de déplacement optimale au pour maintenir leur survie et aussi la capacité de vitesse est évident (i.e. on a deux fourrageur en même place presque et la distance entre les deux fourrageur et la plus nourriture est même aussi alors qui a la meilleur capacité de vitesse c'est lui qui peut survivre.

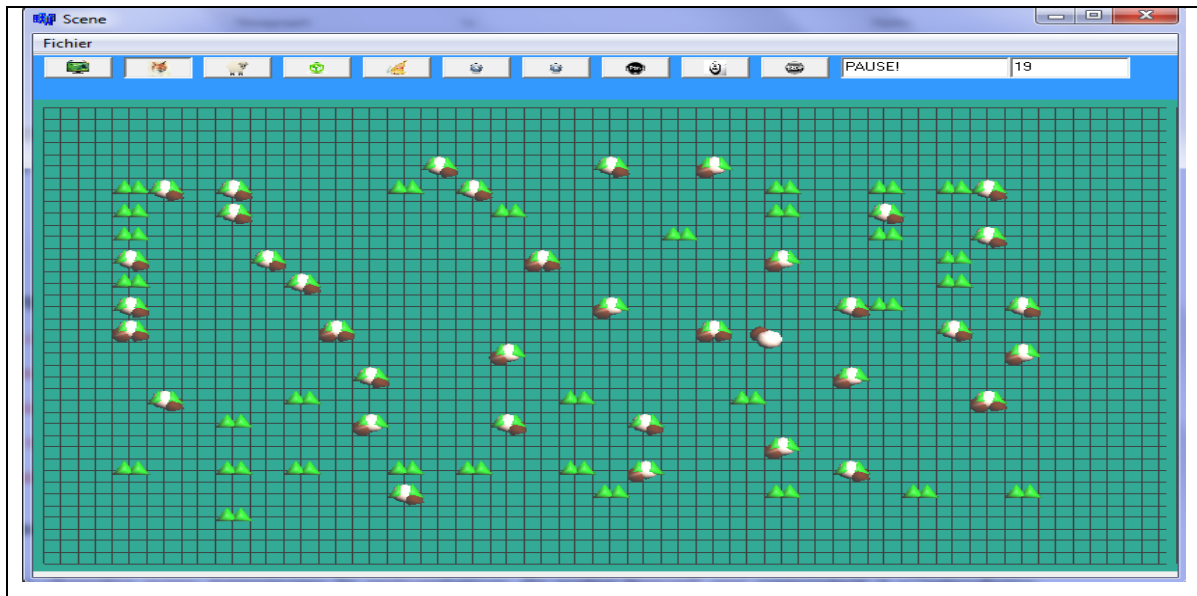


Figure 5. Génération 70 (plus de fourrageur sont arrivé au but).

En remarque après la génération 70 qu'il y a beaucoup de changement sur le comportement des fourrageurs et ses capacités de chromosome va augmenter.

L'utilisation d'un algorithme génétique et d'un processus qui durent plusieurs heures ou plusieurs jours continus de travail ne peut pas être évalué par la présentation des résultats finaux, mais il faut bien évaluer ces résultats et présenter plus de données sur le déroulement du processus tout entier.

Pour faire cela nous avons enregistré les valeurs de fitness expliquant la qualité des individus qui était traduite par des comportements et nous avons établis le graphe présenté dans la figure6 qui expliquent l'amélioration des comportements (tel que vue par les valeurs qui s'augmentent).

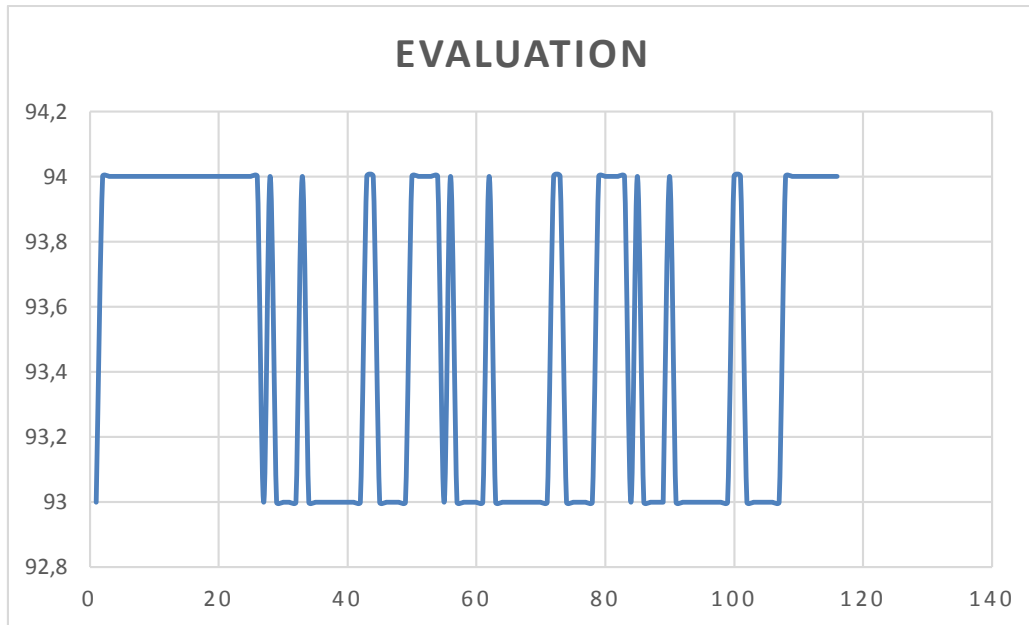


Figure6. Graphe d'évolution qui présente le changement des valeurs de fitness par rapport aux générations.

Conclusion

Avec ce chapitre nous terminons la présentation de notre travail qui consistait à s'introduire dans le domaine de vie artificielle et utiliser ses méthodes les plus utilisés (algorithme génétique) et de s'inspirer de la nature, et aussi de s'inspirer de certains travaux déjà réalisé dans ce domaine de recherche, ces résultats présentent la validité de notre modèle et nous mets dans le bon chemin pour pouvoir aller plus loin dans le domaine.

En conclusion nous avons présenté un modèle d'évolution de comportement de créatures artificielles qui sont des créatures fourrageurs qui devront chercher des sources de nourriture de la façon la plus efficace, afin d'utiliser cette énergie pour se reproduire dans son environnement.

Conclusion générale

Conclusion générale

Nous envisageons dans ce mémoire une contribution dans le domaine de vie artificielle qui consiste à simuler d'un modèle de création d'un écosystème artificiel avec deux créatures artificielles, et qui consiste à évoluer leur capacité de vie afin de pouvoir mieux se nourrir pour à la fin se reproduire et former une population et utiliser l'énergie pour maintenir la survie.

Dans cette partie de cette contribution nous avons proposé un modèle qui a un rôle très important dans la création d'un écosystème pareil, ce modèle, on met l'accent sur l'étude des comportements évolutifs qui produits qui visent à trouver le comportement du fourrageur en recherche des nourritures. L'étude présentée dans ce mémoire est vue comme la première étape réalisée à la construction d'un écosystème virtuel simulé. Nous avons étudié dans ce modèle des comportements de fourrageur (avec l'évolution d'une population), l'objectif du fourrageur est de maintenir leur survie.

Nous avons présenté nos résultats primaires obtenues de simulation de créatures artificielles, nous avons aussi à poursuivre plusieurs autres étapes pour compléter les résultats qu'un modèle comme le nôtre pourras fournir et cela en appliquant le processus d'évolution sur les créature fourrageur en utilisons le type de codage qu'on décrit dans notre modèle et même avec d'autres types de créatures artificielles.

Nous avons implémenté le modèle proposé et nous avons pu présenter et discuter les résultats obtenus qui sont considérés comme notable.

Nos futur travaux consistent à aller plus moins avec l'idée de simulation de fourrageur et de leur comportement et cela peut être appliqué en utilisons un modèle plus complexe et en exploitant des techniques avancé dans le domaine de vie artificielle.

Bibliographie

Bibliographie

- [AK,05] Andrew Adamatzky and Maciej Komosinski. *Artificial Life Models in Software*. Springer-Verlag, 2005.
- [CM, 95] 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, 96] 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).
- [Dje,10] Nedjma DJEZZAR. *L'Ontogenèse pour la création de créatures artificielles*. Université Med kheider Biskra, Algérie. 2010.
- [DU, 05] K. Doya and E. Uchibe. *The Cyber Rodent Project: Exploration of Adaptive Mechanisms for Self-Preservation and Self-Reproduction*. Adaptive Behavior, 2005.
- [EUD, 05] S. Elfwing, E. Uchibe, K. Doya, and HI Christensen. Biologically inspired embodied evolution of survival. *Evolutionary Computation*. The 2005 IEEE Congress, 2005.
- [Gol,89] Goldberg D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, Massachusetts, 1989.
- [Heg, 03] Olivier Heguy. *Architecture comportementale pour l'émergence d'activités coopératives en environnement virtuel*. Ph.D. thesis, Université Paul Sabatier, Toulouse (2003).
- [Hol,75] Holland J.H., *Adaptation in Natural and Artificial Systems*. Ann Arbor: The University of Michigan Press, 1975.
- [HP,01b] Gregory Hornby and Jordan B. Pollack. Evolving l-systems to generate virtual creatures. *Computers & Graphics*, 25(6):1041–1048, 2001.
- [HP,02] G. Hornby and J. Pollack. *Creating high-level components with a generative representation for body-brain evolution*, 2002.
- [Kom,00] Maciej Komosinski. The world of framsticks: Simulation, evolution, interaction. In VW'00: *Proceedings of the Second International Conference on Virtual Worlds*, Springer-Verlag, London, UK, pages 214–224, 2000.

- [Kom,03] 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,05] Maciej Komosinski. Framsticks: a platform for modeling, simulating and evolving 3D creatures. In *Artificial Life Models in Software*, Springer-Verlag, pages 37–66, 2005.
- [LLD,07a] Nicolas Lassabe, Hervé Luga, and Yves Duthen. Interaction of evolving artificial creatures. In *International Conference on Enactive Interfaces*, pages 137–140, 2007.
- [LLD,07b] Nicolas Lassabe, Hervé Luga, and Yves Duthen. A new step for evolving creatures. In *IEEE-ALife'07, IEEE*, pages 243–251, 2007.
- [LLD,07c] Nicolas Lassabe, Hervé Luga, and Yves Duthen. New skills for evolving artificial creatures. *International Journal of Information Technology and Intelligent Computing*, IEEEICIS, 2007.
- [MP, 43] Warren S. McCulloch and Walter Pitts. A logical calculus of ideas immanent in nervous activity. *Bulletin of mathematical biophysics*. Reprinted in McCulloch, W. S., *Embodiments of mind*, MIT Press Cambridge, MA, 5 :115–133, 1943.c
- [McI et al, 71] MCILROY, M. D., MORRIS, R., AND VYSSOTSKY, V. A. 1971. *Darwin, a game of survival of the fittest among programs*. Technical report, Bell Labs.
- [Pan, 03] David Panzoli. Simulation comportementale par réseau de neurones et apprentissage par algorithme génétique. Master's thesis, IRIT, Toulouse, 2003.
- [PHLF,03] J. B. Pollack, G. S. Hornby, Hod Lipson Lipson, and P. Funes. Computer creativity in the automatic design of robots. *LEONARDO*, 2(36), 2003.
- [Ray,91a] Thomas S. Ray. Evolution and optimization of digital organisms. In Billing sley K. R., E. Derohanes, H. Brown, III [eds.], *Scientific Excellence in Supercomputing: The IBM 1990 Contest Prize Papers*, Athens, 1991.
- [Ray,91b] Thomas S. Ray. Aesthetically evolved virtual pets. In *Artificial Life 7 workshop proceedings*, pages 158–161, 2000.
- [Ren,03] Jean-Philippe RENNARD « Trois étapes de la vie artificielle : le canard de Vaucanson, les boucles de Langton et le Lt ». Avril 2003.
- [Rey,99] Reynolds C.W. 1999. *Steering Behaviors For Autonomous Characters*. Presented at the 1999 Game Developer Conference.

- [San, 04]** Stéphane Sanchez, « une architecture générique intégrant des mécanismes évolutionnistes pour la simulation comportementale d'acteurs virtuels », le 8 Décembre 2004.
- [Sims,94]** Karl Sims. Evolving 3d morphology and behavior by competition. *Artificial Life*, 1(4):353–372, 1994.
- [Von,05]** H. Von Koch. Une méthode géométrique élémentaire pour l'étude de certaines questions de la théorie des courbes planes. *Acta Mathematica*, 30: 145–174. 1905.
- [Yae,94]** 1994, Computational genetics, physiology, metabolism, neural systems, learning, vision and behaviour or polyworld: Life in a new context, in C. G. Langton (ed.), *Artificial Life III*, Vol. XVII of SFI Studies in the Sciences of Complexity, Addison-Wesley, pp. 263–298.
- [YS, 08]** L. S. Yaeger and O. Sporns. Evolution of neural structure and complexity in a computational ecology 2008(ALIFE'08).IEEE Symposium on Artificial Life, 2008.