



Ceci est un extrait électronique d'une publication de
Diamond Editions :

<http://www.ed-diamond.com>

Retrouvez sur le site tous les anciens numéros en vente par
correspondance ainsi que les tarifs d'abonnement.

Pour vous tenir au courant de l'actualité du magazine, visitez :

<http://www.gnulinuxmag.com>

Ainsi que :

<http://www.linux-pratique.com>

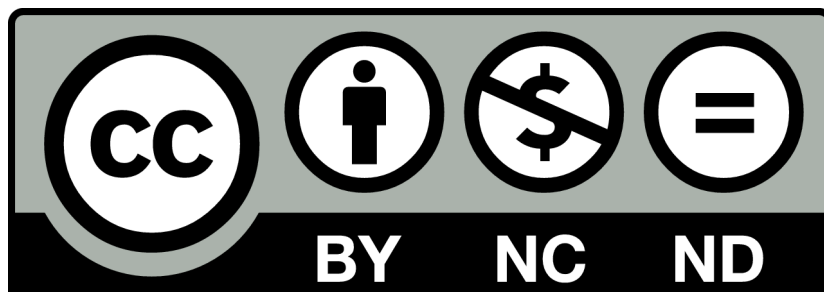
et

<http://www.miscmag.com>



Ceci est un extrait électronique d'une publication de Diamond Editions

<http://www.ed-diamond.com>



Creative Commons

Paternité - Pas d'Utilisation Commerciale - Pas de Modification 2.0 France

Vous êtes libres :

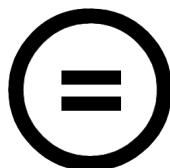
- de reproduire, distribuer et communiquer cette création au public.



Paternité. Vous devez citer le nom de l'auteur original de la manière indiquée par l'auteur de l'oeuvre ou le titulaire des droits qui vous confère cette autorisation (mais pas d'une manière qui suggérerait qu'ils vous soutiennent ou approuvent votre utilisation de l'oeuvre).



Pas d'Utilisation Commerciale. Vous n'avez pas le droit d'utiliser cette création à des fins commerciales.



Pas de Modification. Vous n'avez pas le droit de modifier, de transformer ou d'adapter cette création.

A chaque réutilisation ou distribution de cette création, vous devez faire apparaître clairement au public les conditions contractuelles de sa mise à disposition.

- Chacune de ces conditions peut être levée si vous obtenez l'autorisation du titulaire des droits.
- Rien dans ce contrat ne diminue ou ne restreint le droit moral de l'auteur ou des auteurs.

Ceci est le Résumé Explicatif du Code Juridique. La version intégrale du contrat est attachée en fin de document et disponible sur :

<http://creativecommons.org/licenses/by-nc-nd/2.0/fr/legalcode>

Découvrez les algorithmes évolutionnaires

Introduction

Pour les inventeurs, l'imitation de la nature est une tentation assez forte, même si elle ne conduit pas toujours au succès : les avions ne battent pas des ailes, malgré les tentatives infructueuses de nombreux pionniers du vol humain. En informatique en général, et en intelligence artificielle (IA) en particulier, de nombreux chercheurs se sont inspirés et s'inspirent encore des mécanismes naturels en espérant reproduire algorithmiquement les performances des systèmes qu'ils copient. Dans de nombreux cas, cette démarche s'avère fructueuse, à condition de s'affranchir de la copie naïve pour ne conserver que l'inspiration. Il ne s'agit pas en effet de simuler de façon réaliste certains phénomènes naturels (bien que cela soit passionnant pour d'autres applications), mais de tenter de découvrir les règles de haut niveau qui déterminent le comportement global d'un système et le rendent intéressant.

J'ai expliqué dans [1] comment certains chercheurs se sont inspirés de l'apparente intelligence collective des essaims et autres colonies d'insectes pour mettre au point des algorithmes de calcul de chemins optimaux très efficaces (et plus généralement des algorithmes d'optimisation pour certains problèmes complexes). Les solutions mises en œuvre ne peuvent prétendre simuler le comportement d'une fourmilière, par exemple, mais en retiennent les éléments essentiels : pas de planification centralisée, communications locales, phéromones, etc. Un mécanisme particulièrement fascinant est celui de la sélection naturelle qui "pilote" l'évolution des êtres vivants (au sens large du terme). Charles Darwin constate en effet [2] que les êtres vivants les plus adaptés à leur milieu retirent de cette adaptation un avantage reproductif qui leur permet d'augmenter la

La sélection naturelle, découverte par le génial Charles Darwin, est un phénomène fascinant. Elle constitue le moteur de l'évolution naturelle dont les réussites sont impressionnantes, tant la vie est capable de s'adapter à des milieux inhospitaliers comme les grands fonds marins ou la banquise. Pourtant, les mécanismes de la sélection naturelle sont extrêmement simples et il est donc tentant de les reproduire informatiquement en espérant construire de cette manière des mécanismes aussi complexes, beaux et intelligents que les organismes produits par la nature. Le but de cet article est de partir d'un exemple très simple de vie artificielle pour présenter les algorithmes évolutionnaires, un ensemble de techniques d'optimisation très performantes pour certains problèmes complexes.

probabilité de survie de leur lignée.

En d'autres termes, un être adapté à son milieu a plus de chance d'avoir des descendants viables qu'un être moins adapté. Comme les mécanismes de l'hérédité assurent la transmission des caractéristiques innées à la descendance, les "héritiers" d'un être adapté à son milieu le seront aussi, ce qui renforce l'avantage reproductif au fil des générations. Sur le long terme, une population s'adapte ainsi naturellement et mécaniquement à son milieu. Le résultat est édifiant : nous sommes le produit de l'évolution naturelle et nous sommes relativement intelligents (!) alors que les vestiges préhistoriques nous montrent que nos très lointains ancêtres l'étaient vraisemblablement moins. Même si cette évolution a pris des millions d'années, la tentation est grande pour le chercheur en IA de vouloir reproduire les mécanismes de la sélection naturelle afin de faire évoluer un programme pour le rendre intelligent.

Plus généralement, l'élément fondamental de la théorie de l'évolution est l'idée d'adaptation au milieu. En effet, celle-ci peut être interprétée de façons très diverses en prenant des exemples dans la nature. On peut penser aux diverses stratégies de camouflage utilisées par certains insectes (comme certains phasmes qui ont une forme de feuille), aux stratégies reproductives comme celle du coucou qui pond ses oeufs dans les nids d'autres oiseaux, etc.

Ces stratégies se traduisent par un comportement, une morphologie, une couleur, etc. qui donnent un avantage à certains individus. En fonction du problème (ne pas être mangé, ne pas dépenser de l'énergie à couvrir ses oeufs, etc.), l'évolution a produit des réponses par sélection naturelle. En ce sens, on peut voir celle-ci comme un mécanisme d'optimisation : par son intermédiaire, la nature

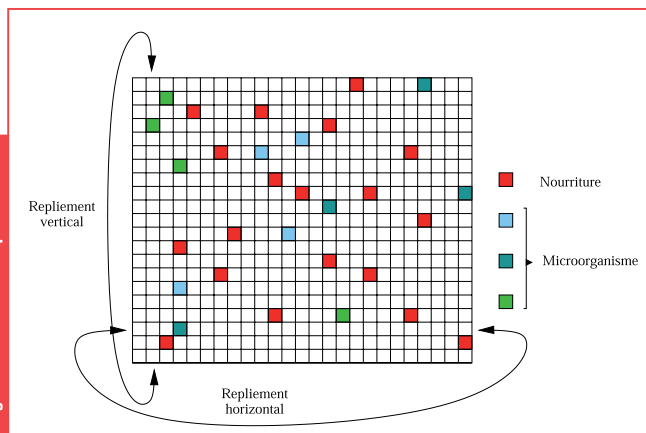
explore efficacement l'ensemble des solutions possibles à un problème et détermine la plus adaptée.

Cette vision motive les **algorithmes évolutionnaires** dont le but est de résoudre des problèmes d'optimisation en reproduisant informatiquement les mécanismes de la sélection naturelle. Je vais présenter dans cet article la forme générale des algorithmes évolutionnaires, en commençant par une illustration extrêmement simple, l'évolution artificielle d'une colonie de micro-organismes.

Une colonie de micro-organismes artificiels

Principes

Je développe ici un modèle très proche de celui présenté dans [1], en remplaçant les fourmis par des micro-organismes (le lecteur trouvera une implémentation en Java du modèle à l'URL suivante : <http://apiacoa.org/software/alife>). Il s'agit en effet d'étudier un ensemble de micro-organismes évoluant dans un milieu très simple. Pour faciliter la mise en œuvre informatique, le milieu est modélisé de façon discrète : le monde dans lequel vivent les micro-organismes est ainsi constitué d'une grille de cases. Pour éviter les problèmes liés aux limites de la grille, on considère qu'elle correspond à un monde torique : les cases du bord droit de la grille communiquent avec celles du bord gauche, alors que les cases du bord supérieur sont reliées à celles du bord inférieur, comme indiqué sur la figure 1. Chaque case peut contenir de la nourriture et/ou un nombre arbitraire de micro-organismes. La nourriture apparaît magiquement dans les cases de façon aléatoire, les paramètres du hasard étant réglés de sorte à produire un flux de nourriture adaptable en fonction de la pression environnementale qu'on souhaite exercer.



Les micro-organismes étudiés sont eux aussi réduits à leur plus simple expression. Ils possèdent un système sensoriel très limité : si la case dans laquelle un micro-organisme est positionné contient de la nourriture, ce dernier en prélève une quantité fixe. Par contre, il est incapable d'observer le contenu des cases voisines. De ce fait, ses déplacements se font de manière aléatoire. Plus précisément, chaque micro-organisme est orienté. On peut considérer qu'il "regarde" une des huit cases voisines de celle qu'il occupe, même s'il est en fait incapable de voir ce que cette case

voisine contient. Le déplacement aléatoire consiste alors à choisir une nouvelle direction en tournant à partir de l'orientation actuelle, puis à passer dans la case voisine indiquée par cette direction. Chaque case possède 8 voisines et on considère donc 8 rotations possibles, de -3 (trois cases dans le sens anti-horaire) à 4 (4 cases dans le sens des aiguilles d'une montre).

Pour représenter l'influence du milieu, nous introduisons un élément essentiel, la consommation d'énergie. Chaque micro-organisme est en effet caractérisé par son niveau d'énergie. Chaque déplacement induit une consommation d'une unité d'énergie alors qu'une unité de nourriture consommée apporte 30 unités d'énergie : après avoir mangé, un micro-organisme peut donc se déplacer 30 fois avant de revenir au niveau d'énergie de départ. Même si une case peut contenir plus d'une unité de nourriture, on considère que le micro-organisme ne peut assimiler qu'une unité par tour. Bien entendu, si le niveau d'énergie d'un micro-organisme atteint zéro, il meurt. Enfin, j'ai choisi de rendre les déplacements obligatoires : le temps est représenté de façon discrète, et à chaque unité de temps, tous les micro-organismes se déplacent simultanément d'une case. De ce fait, la consommation énergétique d'un micro-organisme est d'une unité d'énergie par unité de temps. Pour survivre, il doit passer par une case contenant de la nourriture au moins une fois par trente unités de temps. L'âge du micro-organisme est obtenu simplement en comptant le nombre d'unités de temps écoulées depuis sa naissance.

Comme nous l'avons vu en introduction, la sélection naturelle repose sur la notion d'avantage reproductif. Il nous faut donc traduire la pression de l'environnement induite par la consommation d'énergie en termes de reproduction. Nous choisissons la règle suivante : un micro-organisme ne pourra se reproduire que s'il a atteint l'âge de 400 unités de temps et que son niveau d'énergie est d'au moins 400 unités. L'idée sous-jacente est qu'un micro-organisme qui atteint l'âge de 400 est relativement efficace dans sa recherche de nourriture, ce qui est confirmé par le seuil sur le niveau d'énergie. Les valeurs choisies sont assez arbitraires, nous reviendrons sur ce point dans la suite de l'article.

Le code génétique

Le support de l'évolution des êtres vivants est l'ADN. Celui-ci code de façon chimico-physique toutes les informations nécessaires au développement d'un être vivant et à son comportement futur (je n'entre pas ici dans les débats sur l'inné et l'acquis en ne m'intéressant qu'à des organismes simplistes pour lesquels ces débats n'ont pas lieu d'être). Les **algorithmes génétiques** sont une version des algorithmes évolutionnaires dans laquelle une forme de code génétique informatique est utilisée pour coder le comportement des individus dont l'évolution est simulée.

Notre exemple vise la simplicité, c'est pourquoi nous limitons les possibilités d'évolution des micro-organismes à la stratégie de déplacement aléatoire. Comme nous l'avons précisé précédemment, tout se joue dans le choix de la

rotation effectuée par le micro-organisme avant son passage de la case actuelle à la suivante. Le choix étant aléatoire, un micro-organisme est donc caractérisé par les probabilités associées aux huit rotations possibles. Considérons par exemple les probabilités données dans le tableau 1. Un organisme caractérisé par ces valeurs va se déplacer de façon assez aléatoire car il n'a pas une forte probabilité de conserver la direction précédente (la rotation 0 est choisie avec une probabilité de 0.2, soit seulement une fois sur 5).

Tableau 1 : Probabilités de rotation

Rotation	-3	-2	-1	0	1	2	3	4
Probabilité	0.1	0	0.15	0.2	0.15	0.1	0.3	0

Comme le comportement d'un micro-organisme découle mécaniquement des probabilités, nous considérons celles-ci comme son code génétique. Chaque micro-organisme sera donc caractérisé par un tableau qu'il transmettra à sa descendance.

Différentes représentations informatiques sont envisageables pour ce tableau. J'ai choisi d'utiliser un tableau de 8 **bytes** (en Java), en me limitant aux valeurs comprises entre -8 et 8. Le tableau de **bytes** est converti en un tableau de probabilités en deux temps. Dans un premier temps, on construit le tableau des 2^k , où k désigne les entiers contenus dans le tableau de **bytes**. Ensuite, on normalise ces valeurs en les divisant par leur somme (cf le tableau 2 pour un exemple). On pourrait obtenir le même résultat en stockant de façon plus compacte les entiers ou en stockant directement des probabilités. Le choix de la représentation n'est cependant pas anodin et peut en fait avoir une influence importante sur le comportement de l'algorithme.

Reproduction, hérédité et mutation

La sélection naturelle fonctionne grâce à l'hérédité, c'est-à-dire la transmission des caractéristiques innées d'un parent à ses enfants. Dans notre application, il s'agit donc de recopier le code génétique (le tableau des probabilités) d'un micro-organisme dans celui de ses enfants. Comme nous l'avons indiqué précédemment, un micro-organisme ne peut se reproduire que s'il possède une énergie supérieure à 400 unités et est âgé d'au moins 400 unités de temps. Dès que ces conditions sont remplies, le micro-organisme se reproduit par division : il disparaît et est remplacé par deux descendants, qui se partagent à parts égales l'énergie de leur parent.

Le point important concerne la détermination du code génétique des enfants. Si nous nous contentions de recopier celui du parent, aucune progression ne serait possible. En effet, comme le comportement d'un micro-organisme est totalement déterminé par son code génétique, les deux enfants se comporteraient alors exactement comme leur parent et seraient donc tout aussi bien ou mal adapté au milieu. Pour introduire de la variabilité et permettre à l'algorithme d'explorer divers codes génétiques, on modifie le code génétique aléatoirement, reproduisant ainsi le

mécanisme naturel de **mutation génétique**. Dans la nature, l'ADN est tellement complexe que des erreurs de copie peuvent se produire : le code d'un descendant n'est pas toujours strictement identique à celui de son parent. Des mécanismes de redondance, de correction automatique, etc. conservent le taux de mutation à un niveau très faible et les descendants sont en général très proches génétiquement de leurs parents. Dans notre application, les mutations sont la seule source de variabilité car la reproduction que nous étudions est asexuée (nous reviendrons sur ce point plus tard). Il est donc important d'avoir un niveau élevé de mutation. J'ai choisi ici d'imposer une mutation par reproduction. Les enfants d'un micro-organisme ont donc toujours (ou presque) un code génétique différent de celui de leur parent. La mutation est implémentée très simplement : on choisit au hasard une rotation puis on modifie la probabilité concernée de sorte à la diminuer pour un des enfants et à l'augmenter pour l'autre. Avec notre implémentation en nombre entier, nous nous contentons d'ajouter 1 à l'un des deux codes et d'enlever 1 à l'autre. Le tableau 2 donne un exemple de code génétique d'un parent.

Tableau 2 : Code génétique du parent

Rotation	-3	-2	-1	0	1	2	3	4
Code génétique	2	4	-2	1	-3	3	-2	-1
Probabilité	12,85%	51,41%	0,80%	6,42%	0,40%	25,70%	0,80%	1,60%

Les tableaux 3 et 4 donnent le code génétique modifié par mutation pour les deux enfants. On constate que la modification d'un seul entier (qui joue le rôle d'un gène en quelque sorte) a un impact sur l'ensemble des probabilités, à cause de la normalisation. Dans la nature, les interactions entre gènes sont classiques et cet effet est donc acceptable.

Tableau 3 : Code génétique du premier enfant

Rotation	-3	-2	-1	0	1	2	3	4
Code génétique	2	5	-2	1	-3	3	-2	-1
Probabilité	8,49%	67,90%	0,53%	4,24%	0,26%	16,98%	0,53%	0,11%

Tableau 4 : Code génétique du second enfant

Rotation	-3	-2	-1	0	1	2	3	4
Code génétique	2	3	-2	1	-3	3	-2	-1
Probabilité	17,30%	34,59%	1,08%	8,65%	0,54%	34,59%	1,08%	2,16%

Dans certaines situations, le code d'un des enfants est identique à celui du parent car la probabilité qui devrait être modifiée est au maximum ou au minimum choisis dans l'application (à savoir 8 et -8). Ceci n'a aucune influence importante sur le déroulement de l'algorithme.

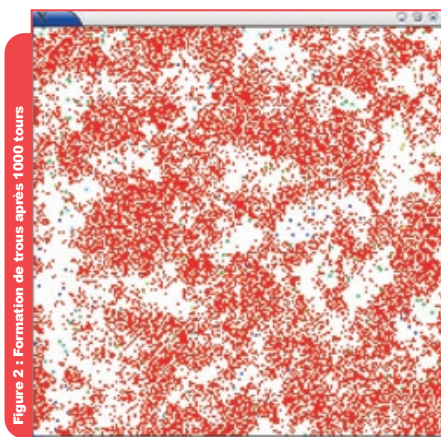
Évolution de la colonie

Initialement, on crée une colonie constituée de micro-organismes dont le code génétique est déterminé

aléatoirement. Les positions de départ des micro-organismes sont choisies aléatoirement, de même que celles des unités de nourriture. La colonie évolue ensuite selon les règles présentées dans ce qui précède. Le temps étant discret, on peut considérer que l'évolution se fait tour par tour. Un tour commence par l'apparition de nourriture à des emplacements aléatoires (répartis uniformément). Ensuite, tous les micro-organismes se déplacent simultanément, ce qui produit une diminution de leur énergie d'une unité et l'augmentation de leur âge d'une unité. Les micro-organismes qui sont dans une case contenant de la nourriture consomment alors une unité de celle-ci. On teste ensuite l'état de chaque micro-organisme : on supprime ceux qui n'ont plus d'énergie et on fait se reproduire ceux qui ont au moins 400 unités d'énergie et qui sont âgés d'au moins 400 tours de jeu. Le tour est alors terminé.

Les paramètres numériques de l'évolution sont assez difficiles à régler et j'ai dû procéder par tâtonnement. J'ai par exemple choisi d'insérer initialement de la nourriture dans 6% des cases du monde. L'énergie et l'âge minimaux pour la reproduction ont été déterminés par essais successifs, de même que l'énergie apportée par une unité de nourriture. La quantité de nourriture à introduire à chaque tour est plus simple à déterminer. En effet, une population de n micro-organismes consomme exactement n unités de nourriture. Il suffit donc de choisir la taille idéale de la population (j'ai choisi 50 pour un monde de 200×200 cases), puis de diviser ce nombre par le nombre représentant la quantité d'énergie apportée par une unité de nourriture pour savoir combien de nourriture doit être ajoutée par tour pour être à l'équilibre. En pratique, il est souhaitable d'en ajouter un peu plus pour éviter des phénomènes de famine.

Résultats pratiques

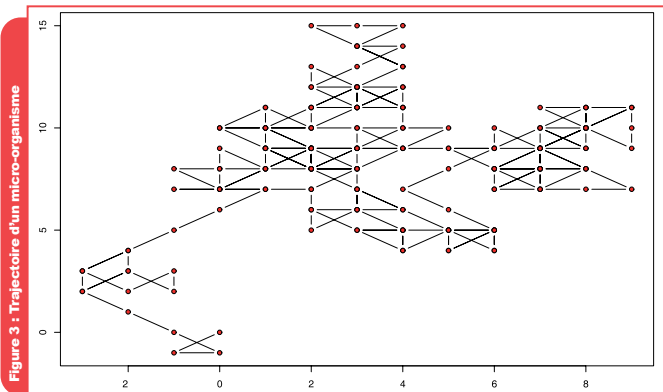


Initialement, les micro-organismes ne sont pas très efficaces dans leur collecte de nourriture. En effet, les codes génétiques choisis au hasard correspondent très souvent à des mouvements quasi aléatoires qui font rester les micro-organismes dans une zone très confinée autour de leur point de naissance. Comme la nourriture est répartie aléatoirement, la zone autour d'un micro-organisme qui fait du sur place a tendance à se vider rapidement, ce qui entraîne la mort de son hôte.

La figure 2 est une capture d'écran d'une implémentation du modèle, après environ 1000 tours. Contrairement à ce qui est indiqué plus haut, j'ai choisi ici une initialisation avec énormément de nourriture pour bien montrer la création de zones vides autour des micro-organismes. Pour le comprendre, considérons l'exemple de code génétique aléatoire donné par le tableau 5.

Tableau 5 : Code génétique aléatoire

Rotation	-3	-2	-1	0	1	2	3	4
Code génétique	2	5	-7	7	4	-7	8	6
Probabilité	0,80%	6,40%	0,02%	25,60%	3,20%	0,02%	51,20%	12,80%



Ce code est assez défavorable car la probabilité de tourner de 3 huitième est de 51,2 %, alors que celle de continuer tout droit est de 25,6 %. La figure 3 donne un exemple de trajectoire correspondant au point de départ (0,0) et à 200 déplacements. On constate que la trajectoire n'est pas très satisfaisante car la surface explorée est faible : le micro-organisme repasse souvent par les mêmes cases.

Tableau 6 : Code génétique obtenu après 2500 tours

Rotation	-3	-2	-1	0	1	2	3	4
Code génétique	-3	1	-5	8	-8	-5	1	-7
Probabilité	0,05%	0,77%	0,01%	98,39%	0,00%	0,01%	0,77%	0,00%

Au fur et à mesure de l'évolution de la population, on constate que les mouvements des micro-organismes deviennent moins désordonnés. Par exemple, au bout d'environ 2500 tours, on peut voir apparaître un micro-organisme dont le code génétique est donné par le tableau 6. Pour ce dernier, les probabilités sont telles que le déplacement sera quasi-linéaire, ce qui n'est pas une solution parfaite car j'ai choisi de travailler dans un monde carré et le micro-organisme va donc repasser continuellement par les mêmes cases. D'ailleurs, après quelques centaines de tours, ce type de code génétique disparaît.

Avec un peu de patience, on finit par obtenir un comportement très intéressant. On constate en effet qu'après 15000 tours, les micro-organismes ont des trajectoires beaucoup moins erratiques et semblent se diriger en ligne brisée, avec quelques tournants. Si on considère par exemple le code génétique donné par le tableau 7, on comprend rapidement pourquoi. Tout

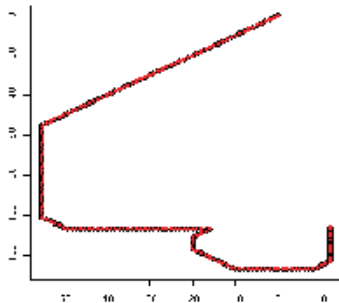
d'abord, les probabilités de faire un demi-tour complet sont très faibles. De plus, malgré une probabilité très forte de continuer tout droit, on effectue un tournant dans 4 % des déplacements, ce qui est suffisant pour éviter un comportement cyclique de type balayage d'une diagonale.

Tableau 7 : Un code génétique obtenu après 20000 tours

Rotation	-3	-2	-1	0	1	2	3	4
Code génétique	-1	-3	-5	7	2	-3	-8	-7
Probabilité	0,38%	0,09%	0,02%	96,39%	3,01%	0,09%	0,00%	0,01%

La figure 4 donne un exemple de trajectoire obtenue avec un tel code génétique sur 200 tours. On constate qu'on obtient à la fois une bonne exploration (le micro-organisme ne repasse pas par le même point) et un élément aléatoire minimal (la trajectoire n'est pas une droite).

Figure 4 : Trajectoire d'un micro-organisme évolué



Une fois ce stade atteint, il n'y a plus beaucoup d'évolution. La population continue bien entendu à se modifier en raison des mutations lors de la reproduction, mais on semble atteindre une sorte d'optimum, ce qui est relativement logique. On ne voit pas en effet comment améliorer la collecte de nourriture si on ne sort pas du modèle de déplacement aléatoire. Il semble donc que l'évolution artificielle obtenue grâce aux algorithmes proposés fonctionne de façon satisfaisante en produisant un code génétique bien adapté au milieu.

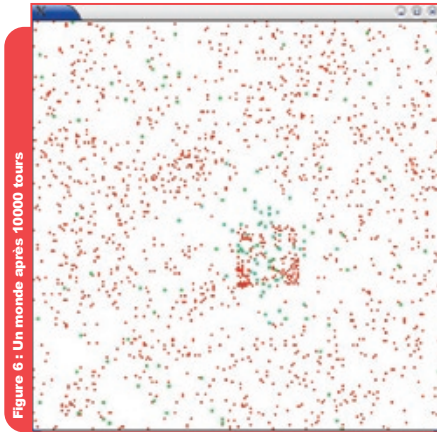
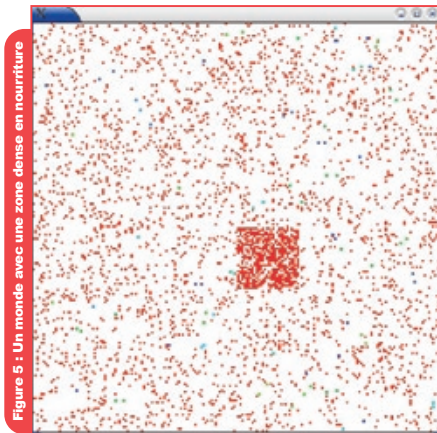
L'exploration aléatoire

Les micro-organismes évoluent-ils vraiment ?

En étudiant l'exemple proposé dans la première partie de cet article, on peut se poser légitimement la question suivante : l'adaptation au milieu observée est-elle réelle ou est-ce en fait le fruit d'une sorte de coup de chance ? En d'autres termes, peut-on espérer obtenir une adaptation au milieu et donc une optimisation pour d'autres problèmes avec une telle méthode ?

Avant de voir une justification de l'algorithme extrêmement simple mis en œuvre dans l'application proposée, étudions l'effet d'une modification du milieu dans celle-ci. Pour l'instant, nous avons seulement considéré une nourriture répartie de façon aléatoire dans le monde. Pour être précis, cette répartition était uniforme, c'est-à-dire qu'elle ne privilégiait aucune partie du monde par rapport aux autres.

On peut changer la situation en créant une zone spéciale dans le monde, de petite taille (par exemple 30x30 cases) dans laquelle la nourriture est très dense. Par exemple, on ajoute à chaque tour deux unités de nourriture dans cette zone, alors qu'on en ajoute autant au reste de la carte. La figure 5 représente une telle situation.



La population évolue alors de façon assez différente. Au bout d'une dizaine de milliers de tours, on retrouve bien entendu des organismes qui se déplacent plus ou moins en ligne droite, mais on découvre aussi un autre groupe de micro-organismes. Ceux-ci restent confinés dans la zone à forte densité de nourriture et s'y portent très bien. La figure 6 représente le monde après environ 10000 tours. On constate la présence des deux lignées : les micro-organismes centrés sur la zone dense en nourriture, de couleur cyan, et ceux qui sont présents un peu partout, de couleur verte.

Pour ce faire, le code génétique des micro-organismes "cyan" a évolué de façon à ce qu'ils tournent en permanence. On observe par exemple le code génétique donné par le tableau 8.

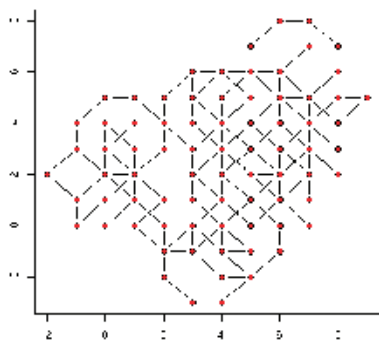
Tableau 8 : Un code génétique d'un micro-organisme tourneur

Rotation	-3	-2	-1	0	1	2	3	4
Code génétique	4	5	8	0	1	-3	-3	3
Probabilité	5,08%	10,15%	81,21%	0,32%	0,63%	0,04%	0,04%	2,54%

Les probabilités induites par ce code entraînent une rotation de -1 dans plus de 80 % des cas, ce qui confine le micro-

organisme dans une zone très petite, comme l'illustre la figure 7 (trajectoire sur 200 tours). Dans cette figure, le point important est l'échelle : les deux cents mouvements se font sur une zone de 10x10 cases, alors que pour la figure 3 la zone est de 15x10 et de 120x60 pour la figure 4. On constate ainsi que sans aucune modification de l'algorithme, la population s'adapte à des milieux différents. Il est donc clair que l'algorithme génétique proposé réalise une optimisation.

Figure 7 : Trajectoire d'un micro-organisme "tourneur"



Le hasard comme moteur d'optimisation

Même s'il est facile de constater que l'algorithme fonctionne, les raisons de son efficacité sont *a priori* assez mystérieuses. Pourtant, l'explication mathématique est très simple : l'algorithme génétique que nous avons proposé est une forme d'exploration aléatoire de l'espace des solutions du problème d'optimisation étudié. Comme toujours, le langage mathématique est obscur pour le non-spécialiste et quelques explications complémentaires sont nécessaires.

L'optimisation

Un problème d'optimisation se représente mathématiquement grâce à deux objets, un ensemble S et une fonction f de S dans l'ensemble des réels. L'ensemble S est l'espace des solutions du problème. Il contient toutes les solutions possibles. La fonction f associe à chaque solution x prise dans S une valeur notée $f(x)$. Le but de l'optimisation est de trouver un élément de S qui maximise f , c'est-à-dire un x dans S tel que $f(x)$ soit plus grand que $f(y)$ pour n'importe quel autre y dans S .

Considérons un exemple simple : je dispose d'un grand carton rectangulaire dans lequel je souhaite entasser le plus possible de livres pour mon déménagement. Bien entendu, mes livres viennent de divers éditeurs et ont donc des formats assez variés. De plus, mon carton n'est pas adapté, au sens où ses dimensions ne sont pas exactement celle d'un livre par exemple. Mon espace de solutions est l'ensemble de toutes les configurations de livre possibles : 15 livres de poche et 2 grands formats, ou encore 4 grands formats, etc. Cet espace tient bien sûr compte de la disposition des livres dans le carton. La fonction f associe à une disposition des livres le nombre de livres qu'elle permet de placer dans le carton. L'optimisation consiste alors à trouver la meilleure disposition.

Les problèmes d'optimisation sont extrêmement importants en pratique et il est impossible de faire une liste des domaines dans lesquels ils jouent un rôle central tant celle-ci serait longue. On peut citer par exemple l'apprentissage automatique (le *machine learning*) dans lequel on tente de faire apprendre une tâche à un ordinateur. Ce cadre général regroupe les réseaux de neurones, mais aussi les moteurs d'inférence dont les règles sont construites automatiquement à partir d'exemples, certaines méthodes de *data mining*..., en fait une part considérable de l'intelligence artificielle. Dans ce contexte, le problème d'optimisation consiste à régler les paramètres (au sens large du terme) du système à partir d'exemples bien construits : c'est la phase d'apprentissage des réseaux de neurones, par exemple.

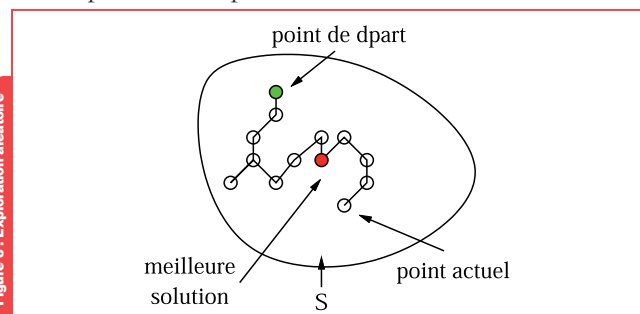
De façon implicite, notre monde et ses micro-organismes forment un problème d'optimisation. L'espace des solutions est l'ensemble de tous les codes génétiques possibles, c'est-à-dire l'ensemble des séquences de 8 entiers compris chacun entre -8 et 8 (ce qui donne 15^8 possibilités (environ 2,5 milliards) pour le code, mais avec des équivalences en termes de probabilités à cause de la normalisation). La fonction f à optimiser est ici totalement implicite car aucun score n'est associé directement à un code génétique. En fait, f est représentée par l'adaptation du micro-organisme au milieu, c'est-à-dire à la qualité de la collecte de nourriture.

Le rôle du hasard

Optimiser la fonction f revient donc à explorer l'espace S . Il existe de très nombreux algorithmes d'optimisation et nous aurons l'occasion d'en rencontrer régulièrement dans la rubrique IA. Un algorithme particulièrement étonnant est celui qui consiste à se déplacer aléatoirement dans S . Le principe est assez simple et se résume en l'algorithme suivant (illustré par la figure 8) :

1. partir du point x_0 , choisi au hasard dans S
2. à l'étape i :
 1. calculer $f(x_i)$ et conserver x_i si la valeur obtenue est plus grande que la meilleure valeur trouvée jusqu'à présent
 2. définir x_{i+1} comme un "voisin" de x_i , choisi aléatoirement
 3. passer à l'étape $i+1$

Figure 8 : Exploration aléatoire



Bien entendu, quelques détails manquent comme un critère d'arrêt (un point très délicat, en général on fixe un nombre d'étapes *a priori* en fonction des ressources disponibles),

une technique pour choisir le point de départ et une autre pour passer d'un point à un autre. Le point important à retenir est que si les choix aléatoires vérifient certaines propriétés mathématiques, on démontre que l'algorithme fonctionne, c'est-à-dire qu'il finit par découvrir la solution du problème.

Les conditions mathématiques visent à assurer que l'exploration est complète : en partant du point de départ, on doit avoir une probabilité non nulle d'atteindre n'importe quel point de l'espace des solutions en un temps fini.

Le lien avec notre algorithme génétique est assez évident : nous partons d'un code génétique aléatoire et chaque naissance correspond à l'exploration d'un nouveau code, voisin du parent (une seule valeur est modifiée) et déterminé aléatoirement (par mutation). Il existe cependant des différences fondamentales. Dans les algorithmes évolutionnaires, on utilise toujours une **population** de solutions envisageables au lieu d'une seule : nous avons ici plusieurs micro-organismes.

De plus, la conservation de la solution optimale est remplacée par une **compétition** entre les individus de la population : seuls les individus adaptés au milieu peuvent avoir une descendance, l'idée étant de privilégier les voisins d'une bonne solution. La similitude avec l'exploration aléatoire est cependant suffisamment forte pour que l'algorithme fonctionne et qu'on obtienne des micro-organismes adaptés à leur milieu.

Comme nous l'avons déjà indiqué, la fonction f à optimiser dans notre exemple est totalement implicite et nous ne sommes pas capables de la calculer directement, ce qui complique la tâche de l'algorithme.

Un véritable algorithme évolutionnaire De la vie artificielle...

L'algorithme génétique que nous avons proposé pour faire évoluer notre population de micro-organismes étudiée présente quelques déficiences. Son principal problème est l'aspect implicite du problème d'optimisation, ce qui a des conséquences néfastes.

Comme nous n'avons pas une définition précise de l'adaptation au milieu, nous ne pouvons pas calculer explicitement la fonction f . De ce fait, nous ne pouvons pas comparer directement les solutions entre elles. La compétition entre les micro-organismes n'est donc pas organisée directement, mais obtenue comme conséquence du fonctionnement du monde.

Le réglage des paramètres de celui-ci, comme par exemple l'énergie et l'âge minimaux pour la reproduction, est de ce fait assez délicat et je n'ai pu déterminer des valeurs intéressantes que par l'intermédiaire de nombreux essais.

De plus, les micro-organismes se reproduisent de façon asexuée, sans brassage de gènes, ce qui réduit l'efficacité de la sélection naturelle.

En raison du caractère implicite de la fonction à optimiser, l'exemple que j'ai proposé s'apparente en fait plus à de la **vie artificielle** [3][4] qu'à un algorithme évolutionnaire. L'idée était en effet de simuler un phénomène naturel de façon très simplifiée et d'observer le comportement obtenu.

L'interprétation en termes d'optimisation permet à la fois de justifier (très partiellement) les résultats observés, mais aussi de se faire une idée du potentiel des méthodes s'inspirant de la sélection naturelle. Nous allons maintenant glisser de la vie artificielle aux algorithmes évolutionnaires.

aux algorithmes évolutionnaires

Comme souvent quand on s'inspire de la nature en informatique, la première étape consiste à simplifier radicalement le modèle.

La difficulté principale dans notre application est que le but n'est pas explicitement formulé. Au contraire, dans un algorithme évolutionnaire, le but est de résoudre un problème d'optimisation, explicitement donné par un espace de solutions S et une fonction f , appelée en général la **fonction d'adaptation** (*fitness* en anglais) ou de performance. L'algorithme est alors le suivant :

- 1. Construction d'une **population** initiale, c'est-à-dire d'un ensemble fini d'éléments de S , P_0
- 2. À la génération i :
 - 1. Sélection d'une sous-population de P_i , R_i , les parents par l'intermédiaire de la fonction f
 - 2. Production des enfants E_i à partir des parents R_i par croisement, mutation, etc.
 - 3. Construction de la génération $i+1$, P_{i+1} , à partir de E_i et de P_i
 - 4. Arrêt éventuel de l'algorithme si P_{i+1} est satisfaisante

En fait, on constate que cet algorithme est celui utilisé par les éleveurs ! Au lieu de laisser la population évoluer naturellement grâce à une pression sélective mal maîtrisée, l'éleveur sélectionne les animaux en fonction de certaines caractéristiques qui l'intéressent (c'est la fameuse fonction f).

Il croise ensuite les animaux sélectionnés en espérant obtenir des descendants dont les caractéristiques seront une combinaison de celles des parents, ce qui fonctionne pour certaines d'entre elles. C'est d'ailleurs en croisant des pois que Gregor Mendel a fondé la génétique [5][6] !

J'ai présenté ici le modèle général des algorithmes évolutionnaires et il est clair que de nombreux détails manquent pour une mise en œuvre effective, essentiellement sur quatre points :

- 1. comment choisir la population initiale ;
- 2. comment sélectionner les parents ;
- 3. comment produire les enfants à partir des parents ;
- 4. comment produire la nouvelle population à partir de l'ancienne et des enfants.



Il manque aussi un critère d'arrêt, mais comme je l'ai déjà indiqué pour l'exploration aléatoire, on se contente en général de fixer *a priori* un nombre d'étapes, ici de générations.

Nous pourrions facilement nous transformer en éleveur pour nos micro-organismes, à condition de choisir une fonction f . Nous devons donc adopter le regard de l'éleveur et décider ce qui nous intéresse dans les micro-organismes. Si vous voulons par exemple qu'ils soient bien nourris, un choix possible pour f consiste à placer un micro-organisme dans un environnement confiné contenant une quantité de nourriture connue et fixée à l'avance.

On laisse le micro-organisme se déplacer pendant un certain temps puis on considère que la valeur de f pour le micro-organisme est exactement le niveau d'énergie après les déplacements. Reste alors à régler les différents points généraux évoqués plus haut.

La sélection

Il y a en fait deux sélections : celle des parents, puis celle qui permet de produire la nouvelle génération. Les sélections se basent exclusivement sur les valeurs de f et peuvent donc être évoquées dans le cadre des algorithmes évolutionnaires, sans qu'on précise le reste des mécanismes. Il existe de nombreuses procédures de sélection et nous nous contenterons d'évoquer les trois méthodes principales :

■ 1. la sélection dite **élitiste** : dans cette procédure, on ne conserve que les meilleurs individus, au sens de la fonction d'adaptation. Par exemple, seuls les meilleurs individus deviennent des parents ;

■ 2. la sélection par **roulette** : dans cette procédure, on fait intervenir le hasard, un individu ayant d'autant plus de chance d'être choisi qu'il a une bonne valeur au sens de la fonction d'adaptation. L'implémentation est délicate car si on se base sur les valeurs relatives de f (on calcule alors la probabilité de choisir x en divisant $f(x)$ par la somme des valeurs de f pour tous les individus considérés), on peut avoir un phénomène de domination de quelques individus par rapport aux autres. Si on se base au contraire sur le rang (le meilleur a une probabilité x d'être choisi, le second y , etc.) sans tenir compte des différences de valeur, on peut avoir le phénomène inverse ;

■ 3. la sélection par **tournoi** : dans cette procédure, on choisit au hasard T individus (sans tenir compte de la fonction d'adaptation), puis on applique une sélection roulette ou élitiste à cet ensemble pour choisir l'individu gagnant.

Le problème de la sélection étant réglé, nous allons maintenant nous intéresser aux algorithmes génétiques qui permettent de proposer des solutions simples pour les deux autres composants importants d'un algorithme évolutionnaire : le choix initial de la population et surtout la production des descendants.

Les algorithmes génétiques

La représentation génétique

Comme je l'ai indiqué précédemment, un algorithme génétique est un algorithme évolutionnaire dans lequel l'ADN est le support de l'hérédité. Informatiquement, on représente donc les solutions de l'ensemble S par un code génétique artificiel.

Historiquement, on considérait un chromosome numérique constitué d'une chaîne de bits de taille fixée, ce qui amène à distinguer le **génotype** du **phénotype**. En effet, l'espace S n'est pas en général directement compatible avec cette représentation, on doit effectuer un codage pour passer de S à un espace de représentation sous forme de chaîne de bits. Le **phénotype** désigne alors ce qui est directement observable, c'est-à-dire les éléments de l'espace S des solutions.

Le **génotype** désigne de son côté une représentation des éléments de S adaptée à l'exploration par algorithme évolutionnaire. L'idée est de copier d'une façon simple les mécanismes naturels de reproduction avec échange de matériel génétique et mutation.

Dans notre exemple initial, le phénotype est constitué des probabilités des 8 rotations possibles, soit donc de 8 nombres réels vérifiant des propriétés spécifiques aux probabilités : les nombres sont compris entre 0 et 1, et leur somme est 1. Comme je l'ai déjà indiqué, plusieurs représentations informatiques sont possibles pour le génotype.

Moyennant quelques précautions, on pourrait par exemple travailler directement avec les probabilités et donc avoir un génotype identique au phénotype. On pourrait aussi représenter les probabilités en codage binaire, par exemple avec une représentation en virgule fixe sur 1 octet par probabilité, soit une chaîne de 64 bits. J'ai choisi un codage simple dans lequel chaque probabilité est représentée sous la forme 2^k , où k est un entier compris entre -8 et 8.

L'intérêt de ce codage est que les différences entre les probabilités sont très importantes car l'échelle n'est pas linéaire mais exponentielle : ajouter 1 à un entier revient grossièrement à multiplier par 2 la probabilité correspondante, ce qui peut changer radicalement le comportement du micro-organisme. Le point important à retenir est que la représentation est cruciale dans les performances de l'algorithme génétique et qu'elle est assez difficile à choisir...

La population initiale

Le choix de la population initiale est fortement lié au choix du codage génétique. En effet, s'il s'agit d'une chaîne binaire par exemple, il est assez facile la choisir aléatoirement de façon uniforme, c'est-à-dire sans privilégier certains codes par rapport à d'autres, ce qui est la meilleure chose à faire quand on ne possède pas d'information particulière sur la fonction f .

Cependant, l'uniformité du génotype n'implique pas nécessairement celle du phénotype.

Dans notre exemple, plusieurs codes génétiques correspondent à une même probabilité à cause de la normalisation (par exemple le code (1,1,1,1,1,1,1) est équivalent au code (2,2,2,2,2,2,2)) : en fait, le nombre de génotypes distincts qui codent un même phénotype dépend de celui-ci, ce qui signifie qu'une initialisation uniforme sur les génotypes ne le sera pas sur les phénotypes.

Les phénotypes représentés par beaucoup de génotypes distincts auront plus de chance d'être choisis que les autres. Cela n'a pas de conséquences néfastes sur notre application, mais dans certains cas, il faut être très attentif à l'initialisation sous peine d'obtenir un algorithme assez mauvais.

La reproduction : croisements et mutations

Le but de la représentation par génotype est la mise en œuvre de la production des enfants. L'évolution naturelle montre que la reproduction sexuée est la plus efficace en termes d'adaptation au milieu car elle engendre du brassage génétique.

Celui-ci évite une population trop uniforme, qui est en général fragile (à cause de sa sensibilité à l'apparition d'une maladie nouvelle, par exemple). Pour obtenir informatiquement ce brassage, il faut pouvoir mélanger des génotypes. Pour ce faire, on définit un opérateur de croisement (*crossover*) qui s'inspire de ceux qu'on observe dans la nature entre chromosomes.

Considérons notre codage par entier et deux parents donnés par les tableaux 9 et 10. Une méthode de croisement classique consiste à choisir aléatoirement une position de croisement puis à produire deux enfants obtenus en utilisant le début du premier parent combiné avec la fin du deuxième et vice-versa.

Tableau 9 : Premier parent

2	4	-2	1	-3	5	2	1
---	---	----	---	----	---	---	---

Tableau 10 : Deuxième parent

1	2	-1	-4	7	0	-2	-1
---	---	----	----	---	---	----	----

Les tableaux 11 et 12 donnent les génotypes produits par le croisement à la position 3.

Tableau 11 : Premier enfant

2	4	-2	-4	7	0	2	-1
---	---	----	----	---	---	---	----

Tableau 12 : Deuxième enfant

1	2	-1	1	-3	5	2	1
---	---	----	---	----	---	---	---

Ce croisement s'applique sans difficulté à des chaînes binaires. Pour un génotype constitué d'une liste de nombres réels, on peut procéder par barycentre étendu, mais je n'entrerai pas dans les détails ici.

Il existe en fait de nombreux opérateurs de croisement, adaptés au génotype et leur étude constitue un thème de recherche actif.

Le croisement ne suffit cependant pas à assurer une bonne diversité dans la population et une exploration de l'espace des solutions. C'est pourquoi on doit aussi introduire des mutations.

Comme nous l'avons vu dans l'exemple des micro-organismes, il s'agit simplement de modifier aléatoirement un génotype. La technique utilisée pour notre représentation des probabilités est la plus classique : on choisit aléatoirement un élément du code génétique, puis on modifie sa valeur.

Pour une chaîne binaire, on remplace un 1 par un 0 et vice-versa. Pour une liste de nombres réels, on ajoute une petite valeur aléatoire, positive ou négative. Comme pour les croisements, il existe de nombreux opérateurs de mutation, certains étant très évolués : on considère par exemple des mutations dont l'importance est ajustée automatiquement par la sélection naturelle elle-même !

Mise en œuvre

En pratique, pour appliquer un algorithme génétique à un problème d'optimisation, il faut d'abord bien définir ce dernier, c'est-à-dire l'ensemble S et la fonction f . Ensuite, il faut s'assurer qu'un algorithme génétique est bien utile.

En effet, ces algorithmes sont lents et ne se justifient que si d'autres algorithmes d'optimisation plus efficaces ne sont pas applicables.

C'est le cas par exemple quand la fonction f n'est pas régulière (i.e. un petit mouvement dans S entraîne des grandes modifications dans la valeur de f) ou quand S est discret (par exemple si S est constitué de listes d'entiers).

Une fois ces précautions prises, il faut choisir le codage génétique adapté au problème, ce qui s'avère particulièrement délicat. On peut commencer par un codage binaire ou un codage par nombres réels, mais il faut s'attendre à devoir expérimenter.

Du codage découlent l'initialisation et les opérateurs pour la reproduction. Il reste donc au final à choisir les stratégies de sélection pour les parents et la nouvelle génération, ainsi que les paramètres numériques associés, comme le taux de mutation, la taille de la population, etc.

Conclusion

Le modèle simple d'évolution artificielle présenté dans cet article montre qu'on peut reproduire informatiquement la sélection naturelle et obtenir une adaptation d'organismes à leur milieu.

En s'inspirant des techniques classiques utilisées par les éleveurs, des chercheurs ont proposé des algorithmes évolutionnaires, qui utilisent la sélection artificielle pour optimiser des fonctions complexes.

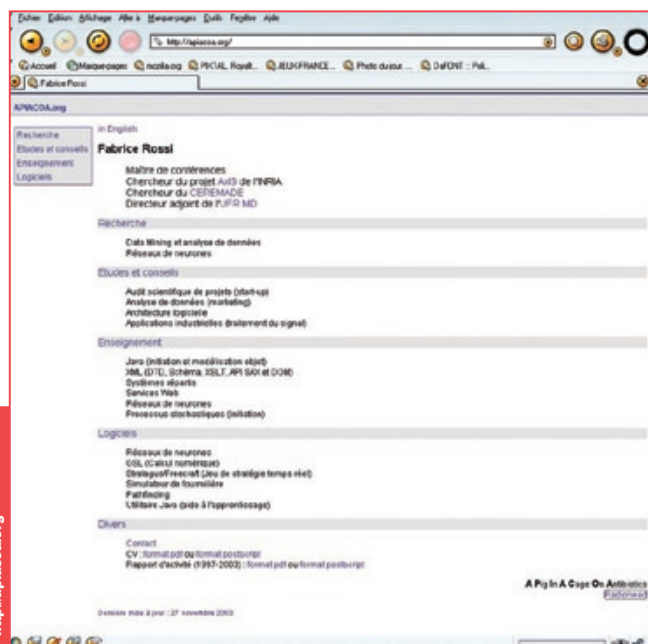
La rapide présentation de ces algorithmes qui conclut cet article montre que les algorithmes évolutionnaires forment

un cadre général qu'il faut spécialiser pour une application particulière. Cette spécialisation est difficile et demande de l'expérience, à la fois sur les algorithmes génétiques eux-mêmes, mais surtout sur le problème considéré. C'est le prix à payer pour accéder à des techniques qui permettent la résolution de problèmes qu'on ne sait tout simplement pas résoudre autrement. Dans de futurs articles de la rubrique IA, nous reviendrons sur les algorithmes génétiques pour les mettre en œuvre sur des problèmes plus concrets que l'exemple présenté ici.

Entre-temps, pour s'initier à la pratique des algorithmes évolutionnaires, je suggère de commencer par l'algorithme génétique le plus classique dans lequel on utilise un codage par chaîne binaire. Pour ce faire, le cours en ligne de Vincent Magnin [7] constitue une bonne base. Pour les lecteurs plus ambitieux (ou téméraires...), les cours de Marc Schoenauer [8], directeur de recherche à l'INRIA et chercheur renommé dans le domaine (il est rédacteur en chef de l'*Evolutionary Computation Journal*), sont une mine d'informations. Ils s'adressent par contre à un public d'ingénieurs et sont donc d'un niveau mathématique adapté à ce public.

Il existe une quantité quasi infinie de bibliothèques *open source* permettant de mettre en œuvre des algorithmes génétiques. Parmi celles-ci, je pense qu'il est important de retenir celles qui ont été produites par des chercheurs reconnus du domaine, comme GALib [9] et EO [10]. On peut d'ailleurs les utiliser avec une sur-couche comme EASEA [11], qui simplifie grandement le développement. Tout ceci étant en C++, les amoureux du C (?) préféreront certainement GAUL [12].

Fabrice Rossi
Projet AxIS (INRIA)
Fabrice.Rossi@apiacoa.org
<http://apiacoa.org>



Références

[1] Fabrice Rossi - *L'intelligence artificielle : des fourmis dans votre Linux*, GNU/Linux Magazine France, No 51, Juin 2003.

[2] Charles Darwin *L'origine des espèces*, 1859. Texte du domaine public, disponible en ligne à l'URL suivante : <http://abu.cnam.fr/cgi-bin/go?espece1>.

[3] Le *Open Directory Project* contient une section dédiée à la vie artificielle (URL: http://dmoz.org/World/Fran%27ais/Informatique/Intelligence_artificielle/Vie_artificielle/) dont les liens sont assez intéressants. La version anglaise accessible depuis l'URL proposée en suivant le lien "anglais" est comme toujours beaucoup plus riche et est recommandée par l'*International Society of Artificial Life (ISAL)*, <http://alife.org/>.

[4] Le site de Jean-Philippe Rennard (URL: <http://www.rennard.org/alife/french/entree.html>), recommandé dans [3] contient quelques textes sur la vie artificielle, quelques applets assez intéressantes (des fourmis dans l'esprit de [1] et des algorithmes génétiques appliqués à l'évolution de formes) et des liens bien choisis.

[5] Le site <http://www.mendelweb.org/> est consacré à Gregor Mendel et contient entre autres une traduction anglaise de son article fondateur sur les pois.

[6] Une présentation rapide des travaux de Mendel est disponible à l'URL suivante : <http://www.infoscience.fr/histoire/portrait/mendel.html>.

[7] Cours de Vincent Magnin, URL : <http://www.eudil.fr/~vmagnin/coursag/index.html>.

[8] Cours de Marc Schoenauer, URL : <http://www.cmap.polytechnique.fr/~marc/enseignement.html>.

[9] GALib : <http://lancet.mit.edu/ga/>

[10] EO : <http://eodev.sourceforge.net/>

[11] EASEA : <http://minimum.inria.fr/evo-lab/EVO-easea.html>

[12] GAUL : <http://gaul.sourceforge.net/>

Creative Commons

Paternité - Pas d'Utilisation Commerciale - Pas de Modification 2.0

Creative Commons n'est pas un cabinet d'avocats et ne fournit pas de services de conseil juridique. La distribution de la présente version de ce contrat ne crée aucune relation juridique entre les parties au contrat présenté ci-après et Creative Commons. Creative Commons fournit cette offre de contrat-type en l'état, à seule fin d'information. Creative Commons ne saurait être tenu responsable des éventuels préjudices résultant du contenu ou de l'utilisation de ce contrat.

Contrat

L'Oeuvre (telle que définie ci-dessous) est mise à disposition selon les termes du présent contrat appelé Contrat Public Creative Commons (dénommé ici « CPCC » ou « Contrat »). L'Oeuvre est protégée par le droit de la propriété littéraire et artistique (droit d'auteur, droits voisins, droits des producteurs de bases de données) ou toute autre loi applicable. Toute utilisation de l'Oeuvre autrement qu'explicitement autorisée selon ce Contrat ou le droit applicable est interdite.

L'exercice sur l'Oeuvre de tout droit proposé par le présent contrat vaut acceptation de celui-ci. Selon les termes et les obligations du présent contrat, la partie Offrante propose à la partie Acceptante l'exercice de certains droits présentés ci-après, et l'Acceptant en approuve les termes et conditions d'utilisation.

1. Définitions

- a. « **Oeuvre** » : oeuvre de l'esprit protégeable par le droit de la propriété littéraire et artistique ou toute loi applicable et qui est mise à disposition selon les termes du présent Contrat.
- b. « **Oeuvre dite Collective** » : une oeuvre dans laquelle l'oeuvre, dans sa forme intégrale et non modifiée, est assemblée en un ensemble collectif avec d'autres contributions qui constituent en elles-mêmes des oeuvres séparées et indépendantes. Constituent notamment des Oeuvres dites Collectives les publications périodiques, les anthologies ou les encyclopédies. Aux termes de la présente autorisation, une oeuvre qui constitue une Oeuvre dite Collective ne sera pas considérée comme une Oeuvre dite Dérivée (telle que définie ci-après).
- c. « **Oeuvre dite Dérivée** » : une oeuvre créée soit à partir de l'Oeuvre seule, soit à partir de l'Oeuvre et d'autres oeuvres préexistantes. Constituent notamment des Oeuvres dites Dérivées les traductions, les arrangements musicaux, les adaptations théâtrales, littéraires ou cinématographiques, les enregistrements sonores, les reproductions par un art ou un procédé quelconque, les résumés, ou toute autre forme sous laquelle l'Oeuvre puisse être remaniée, modifiée, transformée ou adaptée, à l'exception d'une oeuvre qui constitue une Oeuvre dite Collective. Une Oeuvre dite Collective ne sera pas considérée comme une Oeuvre dite Dérivée aux termes du présent Contrat. Dans le cas où l'Oeuvre serait une composition musicale ou un enregistrement sonore, la synchronisation de l'oeuvre avec une image animée sera considérée comme une Oeuvre dite Dérivée pour les propos de ce Contrat.
- d. « **Auteur original** » : la ou les personnes physiques qui ont créé l'Oeuvre.
- e. « **Offrant** » : la ou les personne(s) physique(s) ou morale(s) qui proposent la mise à disposition de l'Oeuvre selon les termes du présent Contrat.
- f. « **Acceptant** » : la personne physique ou morale qui accepte le présent contrat et exerce des droits sans en avoir violé les termes au préalable ou qui a reçu l'autorisation expresse de l'Offrant d'exercer des droits dans le cadre du présent contrat malgré une précédente violation de ce contrat.

2. Exceptions aux droits exclusifs. Aucune disposition de ce contrat n'a pour intention de réduire, limiter ou restreindre les prérogatives issues des exceptions aux droits, de l'épuisement des droits ou d'autres limitations aux droits exclusifs des ayants droit selon le droit de la propriété littéraire et artistique ou les autres lois applicables.

3. Autorisation. Soumis aux termes et conditions définis dans cette autorisation, et ceci pendant toute la durée de protection de l'Oeuvre par le droit de la propriété littéraire et artistique ou le droit applicable, l'Offrant accorde à l'Acceptant l'autorisation mondiale d'exercer à titre gratuit et non exclusif les droits suivants :

- a. reproduire l'Oeuvre, incorporer l'Oeuvre dans une ou plusieurs Oeuvres dites Collectives et reproduire l'Oeuvre telle qu'incorporée dans lesdites Oeuvres dites Collectives;
- b. distribuer des exemplaires ou enregistrements, présenter, représenter ou communiquer l'Oeuvre au public par tout procédé technique, y compris incorporée dans des Oeuvres Collectives;
- c. lorsque l'Oeuvre est une base de données, extraire et réutiliser des parties substantielles de l'Oeuvre.

Les droits mentionnés ci-dessus peuvent être exercés sur tous les supports, médias, procédés techniques et formats. Les droits ci-dessus incluent le droit d'effectuer les modifications nécessaires techniquement à l'exercice des droits dans d'autres formats et procédés techniques. L'exercice de tous les droits qui ne sont pas expressément autorisés par l'Offrant ou dont il n'aurait pas la gestion demeure réservé, notamment les mécanismes de gestion collective obligatoire applicables décrits à l'article 4(d).

4. Restrictions. L'autorisation accordée par l'article 3 est expressément assujettie et limitée par le respect des restrictions suivantes :

- a. L'Acceptant peut reproduire, distribuer, représenter ou communiquer au public l'Oeuvre y compris par voie numérique uniquement selon les termes de ce Contrat. L'Acceptant doit inclure une copie ou l'adresse Internet (Identifiant Uniforme de Ressource) du présent Contrat à toute reproduction ou enregistrement de l'Oeuvre que l'Acceptant distribue, représente ou communique au public y compris par voie numérique. L'Acceptant ne peut pas offrir ou imposer de conditions d'utilisation de l'Oeuvre qui altèrent ou restreignent les termes du présent Contrat ou l'exercice des droits qui y sont accordés au bénéficiaire. L'Acceptant ne peut pas céder de droits sur l'Oeuvre. L'Acceptant doit conserver intactes toutes les informations qui renvoient à ce Contrat et à l'exonération de responsabilité. L'Acceptant ne peut pas reproduire, distribuer, représenter ou communiquer au public l'Oeuvre, y compris par voie numérique, en utilisant une mesure technique de contrôle d'accès ou de contrôle d'utilisation qui serait contradictoire avec les termes de cet Accord contractuel. Les mentions ci-dessus s'appliquent à l'Oeuvre telle qu'incorporée dans une Oeuvre dite Collective, mais, en dehors de l'Oeuvre en elle-même, ne soumettent pas l'Oeuvre dite Collective, aux termes du présent Contrat. Si l'Acceptant crée une Oeuvre dite Collective, à la demande de tout Offrant, il devra, dans la mesure du possible, retirer de l'Oeuvre dite Collective toute référence au dit Offrant, comme demandé. Si l'Acceptant crée une Oeuvre dite Collective, à la demande de tout Auteur, il devra, dans la mesure du possible, retirer de l'Oeuvre dite Collective toute référence au dit Auteur, comme demandé.

- b. L'Acceptant ne peut exercer aucun des droits conférés par l'article 3 avec l'intention ou l'objectif d'obtenir un profit commercial ou une compensation financière personnelle. L'échange de l'Oeuvre avec d'autres Oeuvres protégées par le droit de la propriété littéraire et artistique par le partage électronique de fichiers, ou par tout autre moyen, n'est pas considéré comme un échange avec l'intention ou l'objectif d'un profit commercial ou d'une compensation financière personnelle, dans la mesure où aucun paiement ou compensation financière n'intervient en relation avec l'échange d'Oeuvres protégées.
- c. Si l'Acceptant reproduit, distribue, représente ou communique l'Oeuvre au public, y compris par voie numérique, il doit conserver intactes toutes les informations sur le régime des droits et en attribuer la paternité à l'Auteur Original, de manière raisonnable au regard du médium ou au moyen utilisé. Il doit communiquer le nom de l'Auteur Original ou son éventuel pseudonyme s'il est indiqué ; le titre de l'Oeuvre Originale s'il est indiqué ; dans la mesure du possible, l'adresse Internet ou l'Identifiant Uniforme de Ressource (URI), s'il existe, spécifié par l'Offrant comme associé à l'Oeuvre, à moins que cette adresse ne renvoie pas aux informations légales (paternité et conditions d'utilisation de l'Oeuvre). Ces obligations d'attribution de paternité doivent être exécutées de manière raisonnable. Cependant, dans le cas d'une Oeuvre dite Collective, ces informations doivent, au minimum, apparaître à la place et de manière aussi visible que celles à laquelle apparaissent les informations de même nature.
- d. Dans le cas où une utilisation de l'Oeuvre serait soumise à un régime légal de gestion collective obligatoire, l'Offrant se réserve le droit exclusif de collecter ces redevances par l'intermédiaire de la société de perception et de répartition des droits compétente. Sont notamment concernés la radiodiffusion et la communication dans un lieu public de phonogrammes publiés à des fins de commerce, certains cas de retransmission par câble et satellite, la copie privée d'Oeuvres fixées sur phonogrammes ou vidéogrammes, la reproduction par reprographie.

5. Garantie et exonération de responsabilité

- a. En mettant l'Oeuvre à la disposition du public selon les termes de ce Contrat, l'Offrant déclare de bonne foi qu'à sa connaissance et dans les limites d'une enquête raisonnable :
 - i. L'Offrant a obtenu tous les droits sur l'Oeuvre nécessaires pour pouvoir autoriser l'exercice des droits accordés par le présent Contrat, et permettre la jouissance paisible et l'exercice licite de ces droits, ceci sans que l'Acceptant n'ait aucune obligation de verser de rémunération ou tout autre paiement ou droits, dans la limite des mécanismes de gestion collective obligatoire applicables décrits à l'article 4(e);
- b. L'Oeuvre n'est constitutive ni d'une violation des droits de tiers, notamment du droit de la propriété littéraire et artistique, du droit des marques, du droit de l'information, du droit civil ou de tout autre droit, ni de diffamation, de violation de la vie privée ou de tout autre préjudice délictuel à l'égard de toute tierce partie.
- c. A l'exception des situations expressément mentionnées dans le présent Contrat ou dans un autre accord écrit, ou exigées par la loi applicable, l'Oeuvre est mise à disposition en l'état sans garantie d'aucune sorte, qu'elle soit expresse ou tacite, y compris à l'égard du contenu ou de l'exactitude de l'Oeuvre.

6. Limitation de responsabilité. A l'exception des garanties d'ordre public imposées par la loi applicable et des réparations imposées par le régime de la responsabilité vis-à-vis d'un tiers en raison de la violation des garanties prévues par l'article 5 du présent contrat, l'Offrant ne sera en aucun cas tenu responsable vis-à-vis de l'Acceptant, sur la base d'aucune théorie légale ni en raison d'aucun préjudice direct, indirect, matériel ou moral, résultant de l'exécution du présent Contrat ou de l'utilisation de l'Oeuvre, y compris dans l'hypothèse où l'Offrant avait connaissance de la possible existence d'un tel préjudice.

7. Résiliation

- a. Tout manquement aux termes du contrat par l'Acceptant entraîne la résiliation automatique du Contrat et la fin des droits qui en découlent. Cependant, le contrat conserve ses effets envers les personnes physiques ou morales qui ont reçu de la part de l'Acceptant, en exécution du présent contrat, la mise à disposition d'Oeuvres dites Dérivées, ou d'Oeuvres dites Collectives, ceci tant qu'elles respectent pleinement leurs obligations. Les sections 1, 2, 5, 6 et 7 du contrat continuent à s'appliquer après la résiliation de celui-ci.
- b. Dans les limites indiquées ci-dessus, le présent Contrat s'applique pendant toute la durée de protection de l'Oeuvre selon le droit applicable. Néanmoins, l'Offrant se réserve à tout moment le droit d'exploiter l'Oeuvre sous des conditions contractuelles différentes, ou d'en cesser la diffusion; cependant, le recours à cette option ne doit pas conduire à retirer les effets du présent Contrat (ou de tout contrat qui a été ou doit être accordé selon les termes de ce Contrat), et ce Contrat continuera à s'appliquer dans tous ses effets jusqu'à ce que sa résiliation intervienne dans les conditions décrites ci-dessus.

8. Divers

- a. A chaque reproduction ou communication au public par voie numérique de l'Oeuvre ou d'une Oeuvre dite Collective par l'Acceptant, l'Offrant propose au bénéficiaire une offre de mise à disposition de l'Oeuvre dans des termes et conditions identiques à ceux accordés à la partie Acceptante dans le présent Contrat.
- b. La nullité ou l'inapplicabilité d'une quelconque disposition de ce Contrat au regard de la loi applicable n'affecte pas celle des autres dispositions qui resteront pleinement valides et applicables. Sans action additionnelle par les parties à cet accord, lesdites dispositions devront être interprétées dans la mesure minimum nécessaire à leur validité et leur applicabilité.
- c. Aucune limite, renonciation ou modification des termes ou dispositions du présent Contrat ne pourra être acceptée sans le consentement écrit et signé de la partie compétente.
- d. Ce Contrat constitue le seul accord entre les parties à propos de l'Oeuvre mise ici à disposition. Il n'existe aucun élément annexe, accord supplémentaire ou mandat portant sur cette Oeuvre en dehors des éléments mentionnés ici. L'Offrant ne sera tenu par aucune disposition supplémentaire qui pourrait apparaître dans une quelconque communication en provenance de l'Acceptant. Ce Contrat ne peut être modifié sans l'accord mutuel écrit de l'Offrant et de l'Acceptant.
- e. Le droit applicable est le droit français.

Creative Commons n'est pas partie à ce Contrat et n'offre aucune forme de garantie relative à l'Oeuvre. Creative Commons décline toute responsabilité à l'égard de l'Acceptant ou de toute autre partie, quel que soit le fondement légal de cette responsabilité et quel que soit le préjudice subi, direct, indirect, matériel ou moral, qui surviendrait en rapport avec le présent Contrat. Cependant, si Creative Commons s'est expressément identifié comme Offrant pour mettre une Oeuvre à disposition selon les termes de ce Contrat, Creative Commons jouira de tous les droits et obligations d'un Offrant.

A l'exception des fins limitées à informer le public que l'Oeuvre est mise à disposition sous CPCC, aucune des parties n'utilisera la marque « Creative Commons » ou toute autre indication ou logo afférent sans le consentement préalable écrit de Creative Commons. Toute utilisation autorisée devra être effectuée en conformité avec les lignes directrices de Creative Commons à jour au moment de l'utilisation, telles qu'elles sont disponibles sur son site Internet ou sur simple demande.

Creative Commons peut être contacté à <http://creativecommons.org/>.