

Algorithmique génétique

École Centrale Paris
Module Algo-Prog

Joan Hérisson

Ingénieur de Recherche
Université d'Évry-Val-d'Essonne
Laboratoire iSSB (CNRS FRE3561, Genopole[®])

Table des matières

Introduction	5
1 Optimisation	7
1.1 Méthodes Monte-Carlo	8
1.1.1 Exemple : Estimation de la surface d'un lac	9
1.1.2 Qualité du tirage aléatoire	9
2 Les algorithmes génétiques	11
2.1 Description	11
2.2 Applications	12
2.3 La méthode en détail	13
2.3.1 L'encodage des variables	13
2.3.2 La génération initiale	15
2.3.3 L'évaluation	15
2.3.4 La sélection	15
2.3.5 Le croisement	18
2.3.6 La mutation	19

Introduction

La résolution d'un problème d'optimisation consiste à explorer un espace de recherche afin de maximiser (ou minimiser) une fonction donnée. Par exemple, si l'on cherche le parcours le plus court qui visite une unique fois tous les sommets d'un graphe, l'espace de recherche sera composé de l'ensemble des permutations (compatibles avec le graphe¹) de la liste des sommets, et la fonction à minimiser sera la somme des arcs reliant les sommets en suivant l'ordre de la liste courante. On parle alors de *fonction objectif*.

Les complexités (en taille ou en structure) relatives de l'espace de recherche et de la fonction à *optimiser* conduisent à utiliser des méthodes de résolution radicalement différentes. En première approximation, on peut dire qu'une méthode déterministe est adaptée à un espace de recherche petit et complexe et qu'un espace de recherche grand nécessite plutôt une méthode de recherche stochastique.

La Biologie est l'étude du vivant. Le vivant s'exprime sous des formes extrêmement variées et à de nombreuses échelles. Ainsi, la biologie couvre un très large spectre qui va du niveau moléculaire, en passant par celui de la cellule, puis de l'organisme, jusqu'au niveau de la population et enfin de l'écosystème. Ces différents niveaux montrent que le domaine du vivant est complexe et fortement hiérarchisé.

*En quoi l'étude du vivant peut nous aider à résoudre
un problème d'optimisation ?*

Même s'il n'existe pas de preuve générale de la Théorie de l'Évolution de C. Darwin [4], on peut constater que le vivant s'adapte sans cesse à son environnement. Même si des espèces meurent, d'autres apparaissent et la vie n'a ainsi jamais été menacé sur Terre. Il semblerait donc que le monde du vivant optimise constamment la fonction vie des organismes qui le composent. En effet, il est capable de générer des (ensembles de) solutions biologiques qui sont (1) viables, c'est-à-dire qu'elles peuvent elles-même engendrer le vivant

1. Deux sommets se suivent dans la liste si et seulement s'ils sont adjacents dans le graphe

(hérédité) ; et (2) adaptées car elles remplissent la fonction précédente dans un contexte donné (évolution).

Exemple : une population de lapins est constituée d'éléments qui courent de plus en plus vite au fur et à mesure que les renards les chassent.

La clé de cette adaptabilité réside dans l'apparition de petites variations (que l'on appelle mutations) dans le code génétique des individus. Générées lors du processus héréditaire et à l'échelle d'une population, ces petites variations suffisent pour expliquer l'évolution et l'adaptation des espèces. La chasse des renards induit un stress environnemental sur les lapins qui verront alors leur évolution dirigée pour répondre à ce stress. Si les lapins deviennent trop rapides, alors cela deviendra un stress pour les renards qui devront s'adapter afin de se nourrir.

Les algorithmes évolutionnistes (ou évolutionnaires) s'inspirent donc de la Théorie de l'Évolution afin d'optimiser une fonction objectif dans un temps raisonnable. Du fait de l'utilisation de processus aléatoires, ce sont des algorithmes stochastiques.

L'usage d'un algorithme évolutionniste est adapté à une exploration rapide et globale d'un espace de recherche de taille importante et est capable de fournir plusieurs solutions. Dans le cas où l'ensemble des solutions admissibles est complexe (i.e. il est difficile d'isoler une solution admissible), l'admissibilité peut être rendue intrinsèque à la représentation choisie ou intégrée à la génération des chromosomes (mutation, croisement) ou à la fonction à optimiser (on attribue une mauvaise adaptation à une solution non admissible).

Nous verrons dans la section 2.3 comment appliquer un algorithme génétique au problème du voyageur de commerce.

NOTE : Ce cours reprend quelques éléments du mémoire de thèse de doctorat de V. Magnin [6].

Chapitre 1

Optimisation

Pour optimiser un système, on peut employer la méthode dite d'« essai et erreur ». Cette méthode consiste à tester plusieurs solutions possibles en affectant des valeurs différentes aux paramètres du système que l'on souhaite optimiser. Le nombre de solutions testées est souvent très grand et on arrête le processus quand notre système représente une solution satisfaisante. C'est pourquoi, dans la plupart des cas, un problème d'optimisation se divise naturellement en deux phases : recherche des solutions admissibles puis recherche de la solution à coût optimal localement parmi ces dernières.

Afin de trouver une solution satisfaisante en un minimum de temps, il convient d'analyser le problème afin de définir les variables que l'on va faire fluctuer, l'espace de recherche dans lequel sont définies ces variables, l'objectif on souhaite atteindre, et la méthode d'optimisation la plus appropriée.

Les **variables** du système à faire évoluer sont définies par l'utilisateur. Le nombre de variables à optimiser pour un objectif donné peut être très petit comme dans un système à une variable ou lorsque l'utilisateur sait exactement quelle variable faire évoluer afin d'optimiser son système, ou très grand comme dans un système comportant de nombreuses variables et que l'utilisateur ne sait pas exactement quelles variables faire évoluer pour que son système s'optimise.

Ensuite, il s'agit de délimiter l'**espace** dans lequel vont évoluer les variables du système. Lorsque l'espace est infini, il faut le rendre fini, ce qui se fait mécaniquement lorsque le problème est traité par un ordinateur.

Une fois définis les variables et l'espace dans lequel elles sont définies, il convient d'élaborer la fonction qui va permettre de mesurer la qualité de la solution proposée. Cette fonction est appelée fitness fonction (**fonction d'adaptation**). Cette fonction représente la pierre angulaire du processus d'optimisation car c'est elle qui va donner la mesure d'une solution et donc

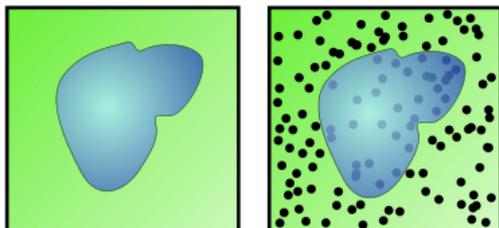
qui va faire de cette solution une bonne solution ou pas. Lorsque l'objectif à atteindre est unique, la fonction de fitness est égale au paramètre à optimiser. Si on cherche par exemple à optimiser la consommation de carburant d'un véhicule, la fonction sera égale à la consommation. Dans certains cas, le problème doit satisfaire plusieurs objectifs (parfois concurrents). On pourra alors définir une fonction pour chaque objectifs et les combiner au sein d'une fonction de fitness globale ou chaque fonction d'objectif unique est pondérée [Fonseca, 1993] en fonction du souhait de l'utilisateur. Par exemple, lors d'un trajet d'un véhicule, on cherche à optimiser la consommation de carburant et le temps de parcours.

Dernier point du processus d'optimisation : le choix de la **méthode d'optimisation**. Si l'évaluation de la fonction est très rapide ou que sa forme peut être connue, on choisira une méthode **déterministe** (méthode du gradient par exemple qui parcourt l'espace de recherche à partir d'un point de départ et de manière quasi-continue). Quand le temps de calcul de la fonction devient grand ou encore que la fonction est non-dérivable, on choisira plutôt une méthode **non-déterministe**. Ce type de méthode explore efficacement l'espace de recherche grâce aux tirages de nombres aléatoires. On peut citer : les méthodes de Monte Carlo qui consiste à évaluer la fonction en un grand nombre de points ; les méthode hybrides (par exemple Monte Carlo combinée avec le gradient) ; les algorithmes évolutionnaires (qui simulent l'évolution biologique d'une population d'individus). La figure 1.1 illustre les différentes méthodes d'optimisation sur un espace de recherche donné.

1.1 Méthodes Monte-Carlo

Les méthodes de Monte-Carlo sont des méthodes d'optimisation non-déterministes qui consistent en des simulations expérimentales ou informatiques de problèmes mathématiques ou physiques, basées sur le tirage de nombres aléatoires. Elles se basent sur l'utilisation d'expériences répétées pour évaluer une quantité, résoudre un système déterministe : estimer une surface ou un volume, résolution d'équations aux dérivées partielles et de systèmes linéaires, résolution de problèmes d'optimisation (recuit simulé). Elles sont généralement simples à mettre en oeuvre et permettent d'affecter une distribution de probabilités à des variables-clés d'un problème.

1.1.1 Exemple : Estimation de la surface d'un lac



On veut estimer la superficie d'un lac. Traçons une zone dont on connaît la superficie (le rectangle ou le carré sera plus simple pour le tirage aléatoire) et qui contient l'intégralité du lac. Pour trouver l'aire du lac, on procède à X tirs

balistiques aléatoires sur cette zone. On compte ensuite le nombre N de boulets qui sont restés sur le terrain ; le nombre de boulets qui sont tombés dans le lac est alors $X - N$. Il suffit ensuite d'établir un rapport entre les valeurs :

$$\frac{SUPERFICIE_{terrain}}{SUPERFICIE_{lac}} = \frac{X}{X - N}$$

$$\Rightarrow SUPERFICIE_{lac} = \frac{X - N}{X} \times SUPERFICIE_{terrain}$$

Par exemple, si le terrain fait 1000 m^2 et que sur 500 tirs, 100 projectiles sont tombés dans le lac, alors une estimation de la superficie du plan d'eau est de : $\frac{1000 \times 100}{500} = 200 \text{ m}^2$.

1.1.2 Qualité du tirage aléatoire

La qualité du tirage aléatoire est primordiale sur la qualité du résultat. En effet, si le tirage est biaisé (donc pas vraiment aléatoire), le résultat sera lui aussi biaisé car on aura attribué un poids plus important dans les zones ayant reçu plus de valeurs. Pour information, un ordinateur ne peut produire qu'une suite de nombres pseudos-aléatoires.

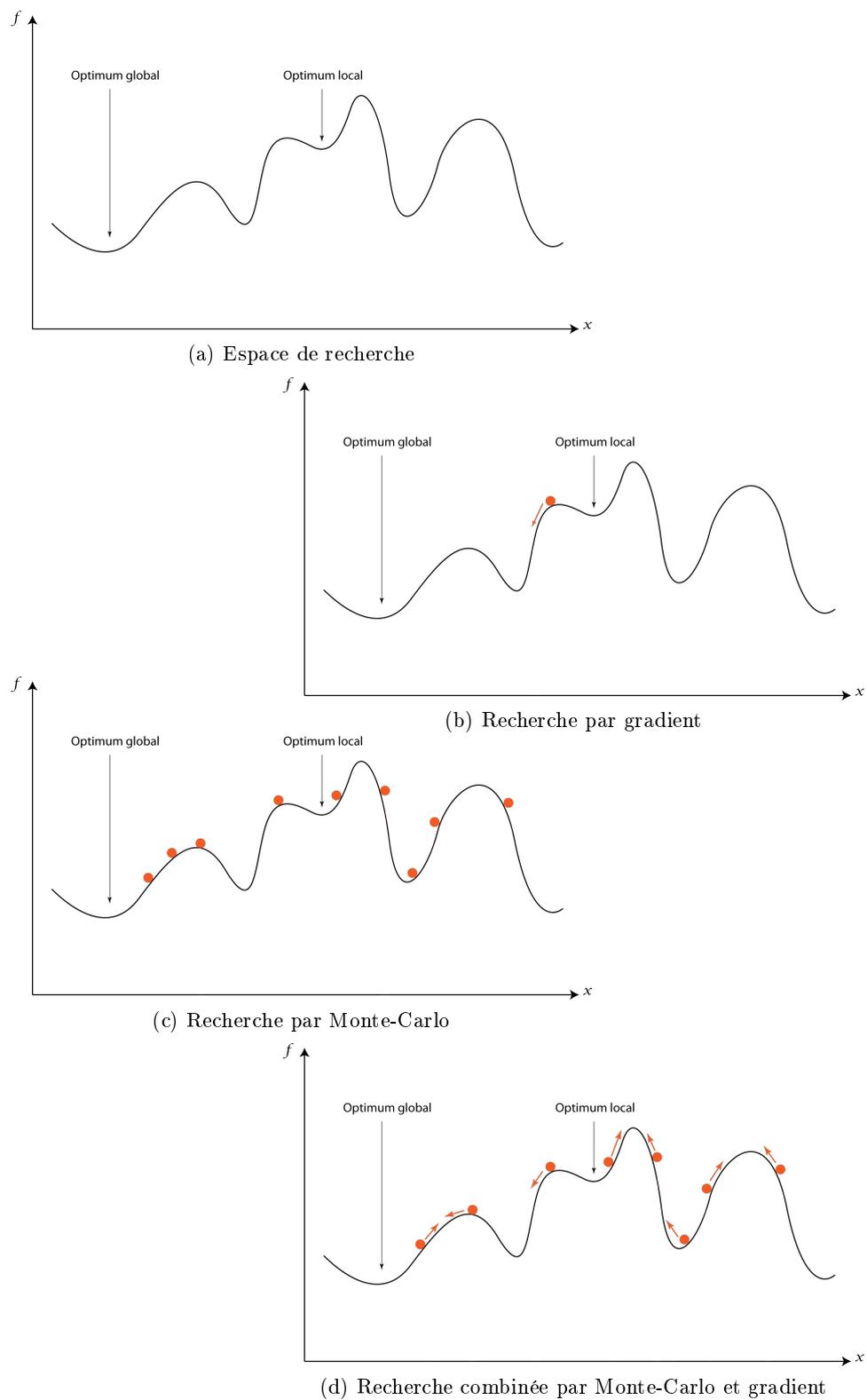


FIGURE 1.1 – Espace de recherche et différentes méthodes de recherche.

Chapitre 2

Les algorithmes génétiques

2.1 Description

Les algorithmes génétiques tentent de simuler le processus d'évolution naturelle suivant le modèle darwinien dans un environnement donné. Ils utilisent un vocabulaire similaire à celui de la génétique naturelle. Cependant, les processus naturels auxquels ils font référence sont beaucoup plus complexes. On parlera ainsi d'individu dans une population. L'individu est représenté par un chromosome constitué de gènes qui contiennent les caractères héréditaires de l'individu. Les principes de sélection, de croisement, de mutation s'inspirent des processus naturels de même nom.

Pour un problème d'optimisation donné, un individu représente un point de l'espace d'états, une solution potentielle. On lui associe la valeur du critère à optimiser, son adaptation. On génère ensuite de façon itérative des populations d'individus sur lesquelles on applique des processus de sélection, de croisement et de mutation (cf. figure 2.1). La sélection a pour but de favoriser les meilleurs éléments de la population pour le critère considéré (les mieux adaptés), le croisement et la mutation assurent l'exploration de l'espace d'états.

On commence par générer une population aléatoire d'individus. Pour passer d'une génération k à la génération $k + 1$, les opérations suivantes sont effectuées. Dans un premier temps, la population est reproduite par sélection où les bons individus se reproduisent mieux que les mauvais. Ensuite, on applique un croisement aux paires d'individus (les parents) d'une certaine proportion de la population (probabilité P_c , généralement autour de 0.6) pour en produire des nouveaux (les enfants). Un opérateur de mutation est également appliqué à une certaine proportion de la population (probabilité P_m , généralement très inférieure à P_c). Enfin, les nouveaux individus sont

évalués et intégrés à la population de la génération suivante. Classiquement, on fait en sorte de conserver une population de taille constante.

Plusieurs critères d'arrêt de l'algorithme sont possibles : le nombre de générations peut être fixé a priori (temps constant) ou l'algorithme peut être arrêté lorsque la population n'évolue plus suffisamment rapidement.

On peut remarquer que l'algorithme ci-contre ne nécessite aucune connaissance du problème. L'algorithme traite des données qui lui sont fournies en entrée pour produire des données de sortie qu'il ré-ingère pour produire de nouvelles données et enfin fournir une ou plusieurs solutions considérées bonnes.

De manière générale, si l'algorithmique évolutionnaire ne permet pas de trouver à coup sûr la solution optimale de l'espace de recherche, elle permet en revanche de fournir une solution qui s'en approche et qui est souvent meilleure que celles fournies par des approches plus classiques.

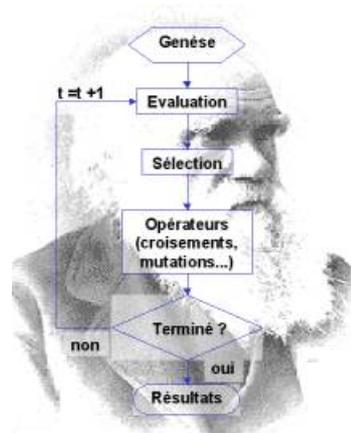


FIGURE 2.1 – *Algorithme évolutionnaire*

2.2 Applications

Les applications des algorithmes génétiques (AG) sont multiples : optimisation de fonctions numériques difficiles (discontinues, multimodales, bruitées), traitement d'image (alignement de photos satellites, reconnaissance de suspects), optimisation d'emplois du temps, optimisation de design, contrôle de systèmes industriels [2], apprentissage des réseaux de neurones [9], etc. Les AG peuvent être utilisés pour contrôler un système évoluant dans le temps (chaîne de production, centrale nucléaire) car la population peut s'adapter à des conditions changeantes. En particulier, ils supportent bien l'existence de bruit dans la fonction à optimiser. Ils peuvent aussi servir à déterminer la configuration d'énergie minimale d'une molécule ou à modéliser le comportement animal.

Les AG sont également utilisés pour optimiser des réseaux (câbles, fibres optiques, mais aussi eau, gaz), des circuits VLSI [2], des antennes [1]. Ils peuvent être utilisés pour trouver les paramètres d'un modèle petit-signal à partir des mesures expérimentales [7]. Des commutateurs optiques adiabatiques ont été optimisés à l'aide des Stratégies d'Evolution (autres AE) chez SIEMENS AG [8]. On envisage l'intégration d'AG dans certaines puces élec-

troniques afin qu'elles soient capables de se reconfigurer automatiquement en fonction de leur environnement (*Evolving Hardware* en anglais).

Pour utiliser un algorithme génétique sur un problème d'optimisation on doit donc disposer d'un principe de codage des individus, d'un mécanisme de génération de la population initiale et d'opérateurs permettant de diversifier la population au cours des générations et d'explorer l'espace de recherche.

2.3 La méthode en détail

Cette section décrit chaque étape d'un algorithme génétique avec ses variantes ainsi que la manière de l'appliquer au problème du voyageur de commerce. Ci-dessous, la description du problème donnée par Wikipédia.

Le problème du voyageur de commerce consiste, étant donné un ensemble de villes séparées par des distances données, à trouver le plus court chemin qui relie toutes les villes. Il s'agit d'un problème d'optimisation pour lequel on ne connaît pas d'algorithme permettant de trouver une solution exacte en un temps polynomial. De plus, la version décisionnelle de l'énoncé : « *pour une distance D , existe-t-il un chemin plus court que D passant par toutes les villes ?* » est connue comme étant un problème NP-complet.

En développant une approche basée sur l'algorithmique génétique, l'objectif est d'approcher la solution optimale pour un coût faible.

2.3.1 L'encodage des variables

Les algorithmes génétiques ne travaillent pas sur un individu isolé mais sur une population d'individus. D'ailleurs, la première étape d'un algorithme génétique est de générer une population d'individus (cf. section 2.1). Et par analogie avec la Biologie, chaque individu est composé d'un (voire plusieurs, cf. section 2.3.6) chromosome appelé aussi génotype [5] et qui représente une solution du problème. Un chromosome représente donc un élément de l'espace de recherche considéré. Chaque chromosome comporte plusieurs gènes qui représentent les variables du problème. L'encodage des variables est donc fondamental dans l'efficacité de l'algorithme.

On peut encoder les gènes de plusieurs manières différentes : réels, entiers, binaire, Gray... Il conviendra de choisir l'encodage qui convient le mieux au problème traité. En général, on encode les variables en notation binaire (cf. figure 2.3) qui a pour avantage de pouvoir facilement coder tout type de variables : entiers, réels, booléens, chaîne de caractères... Seules les fonctions d'encodage/décodage seront à instancier en fonction du type de variable.

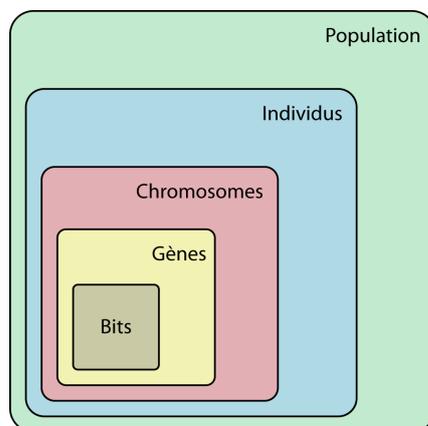


FIGURE 2.2 – Organisation hiérarchique des éléments manipulés par un algorithme génétique

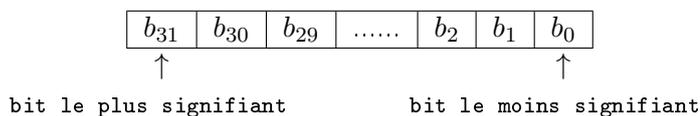


FIGURE 2.3 – Encodage des variables

Chaque gène (chaque paramètre du dispositif) est codé par un entier long (32 bits). À chaque variable réelle x_i on associe donc un entier long

$$g_i = \sum_{j=0}^{31} b_j \cdot 2^j \quad 0 \leq g_i \leq g_{max}, \forall i \in [1, n]$$

Les fonctions de codage et décodage s'écrivent de la manière suivante :

$$g_i = \frac{x_i - x_{imin}}{x_{imax} - x_{imin}} \cdot g_{imax}$$

$$x_i = x_{imin} + (x_{imax} - x_{imin}) \cdot \frac{g_i}{g_{max}}$$

📌 Application au Voyageur de Commerce Encodage

Chaque solution (ou individu) représente un chemin dans le graphe des villes. Chaque gène représente une ville qui peut être représentée par un simple entier. Un chromosome sera donc une suite d'entiers.

2.3.2 La génération initiale

La genèse de la population est la première étape d'un algorithme génétique. C'est à partir de cette population de départ que vont se faire toutes les opérations de l'algorithme et dont dépend le résultat final. Afin de ne pas introduire de biais dans la population initiale, on générera les individus primaires de manière aléatoire. La taille de la population initiale résulte d'un compromis entre qualité de la réponse de l'algorithme et temps de calcul. Quand on augmente le nombre d'individus de départ, on augmente la probabilité de tendre vers un optimum global et on augmente également le temps de calcul car l'algorithme traite un plus grands nombre d'individus.

2.3.3 L'évaluation

Afin de pouvoir comparer les solutions entre elles, il convient de pouvoir les évaluer quantitativement par rapport à l'objectif fixé. Pour ce faire, chaque solution (individu) devra être traitée par la fonction de *fitness* afin de renvoyer un score qui permettra d'ordonner l'ensemble des individus de la population. Dans le cas du voyageur de commerce, la fonction prendra en entrée la suite d'entiers représentant le chemin à suivre et retournera une valeur scalaire qui représentera le coût total du chemin. Ainsi, il sera facile de pouvoir comparer deux individus entre eux et de savoir lequel propose le chemin le moins coûteux.

2.3.4 La sélection

Une fois la population ordonnée par rapport à l'objectif fixé, l'opération de sélection va permettre de choisir les individus qui présentent le plus fort potentiel afin de leur appliquer les deux opérateurs : croisement et mutation. L'idée consiste à sélectionner une partie de la population (en général la moitié) et de la faire se reproduire pour retrouver la taille initiale. Il existe plusieurs méthodes de sélection, quelques-unes sont décrites ci-dessous.

Élitisme

Cette méthode consiste à sélectionner plusieurs individus en fonction de leur score obtenu par la fonction de fitness. Les individus sont triés de manière décroissante selon leur score, puis une partie des individus (en général la moitié) sont sélectionnés pour se reproduire afin de former la génération suivante (cf. figure 2.4).

Cette méthode est simple mais induit une forte pression de sélection qui entraîne souvent la convergence prématurée de l'algorithme.

Sélectionnés								Rejetés						
469	354	352	287	244	215	201	176	154	133	87	65	34	32	14

FIGURE 2.4 - **Sélection par élitisme**
Exemple de sélection de 50% des individus par élitisme

Sélection par roulette

On choisit les individus par un tirage aléatoire en considérant que plus un individu est performant, plus il a de chances d'être sélectionné. Cela revient à réserver pour chaque individu une tranche d'une roulette de jeu de taille proportionnelle à sa performance (cf. figure 2.5). Un individu ayant un score élevé occupe ainsi une surface importante de la roulette. Puis on fait courir une bille et on sélectionne l'individu sur lequel la bille a arrêté sa course. Ainsi, les meilleurs individus ont de grandes chances d'être sélectionnés et peuvent être sélectionnés plusieurs fois. La limite de cette méthode réside dans la sur-représentation potentielle d'un individu qui serait bien meilleur que les autres et qui pourrait représenter par exemple 90% de la roulette (cf. figure 2.5). Ce phénomène pourrait entraîner une stagnation du processus d'évolution.

Sélection par rang

Cette méthode de sélection reprend le principe de la sélection par roulette en limitant l'effet indésirable cité ci-dessus. Les individus sont ordonnés en fonction de leur performance et leur part représentative est proportionnelle à ce rang plutôt qu'à leur performance comme c'est le cas dans la sélection par roulette. Par conséquent, les écarts entre les meilleurs individus et les moins bons sont amortis et on évite ainsi l'écueil de la sur-représentativité (cf. figure 2.6). En revanche, cette réduction des écarts tend à augmenter le temps de convergence de l'algorithme.

Sélection par tournoi

Cette méthode consiste à organiser des tournois entre deux individus pris au hasard dans la population, celui qui remporte le tournoi (celui qui a le meilleur score produit par la fonction fitness) est sélectionné pour se reproduire. De plus, on peut jouer sur la pression de sélection en introduisant une probabilité de remporter le tournoi. On introduit ainsi de la variabilité

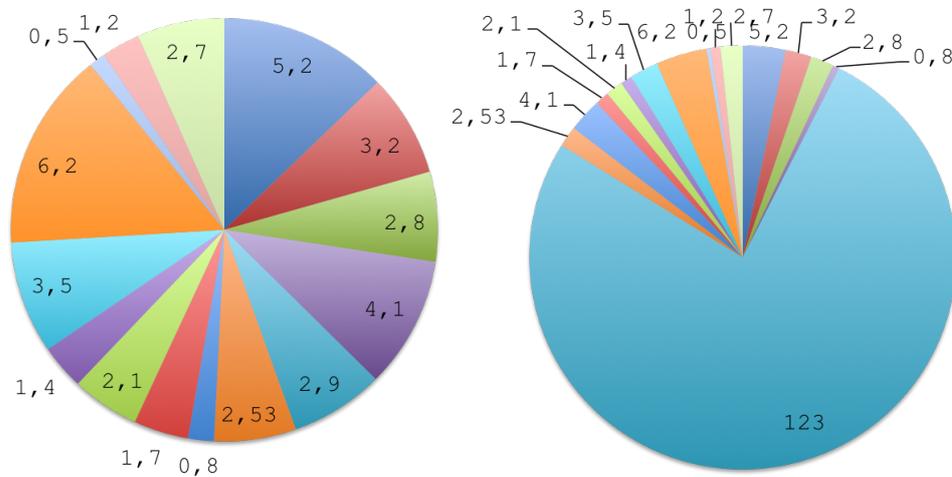


FIGURE 2.5 – Sélection par roulette

Deux exemples de représentation des individus en forme de roulette dans laquelle on va faire courir une bille. Les nombres représentent les scores de chaque individu. Un individu ayant un score élevé a ainsi plus de chances d'être sélectionné. On voit dans la roulette de droite qu'un individu ayant un score bien au-dessus des autres scores est sur-représenté pour la sélection.

dans la sélection qui fait partie d'un processus naturel d'évolution. On notera qu'un individu peut participer à plusieurs tournois.

On peut également créer des méthodes hybrides à partir des méthodes existantes. Par exemple, on pourrait sélectionner par tournoi les individus parents de la prochaine génération et y ajouter le meilleur individu de la génération courante s'il n'a pas été sélectionné.

Une fois la sélection des individus effectuée, il convient de générer de nouveaux individus afin de maintenir la population constante. Pour ce faire, il convient d'introduire de la diversité biologique dans le processus en appliquant deux opérations de manipulation génétique : le croisement et la mutation.



Application au Voyageur de Commerce

Sélection

Toute méthode de sélection peut être appliquée. On pourra débiter par la sélection par élitisme en raison de sa simplicité de mise en œuvre. Néanmoins, la correction du TD implante la méthode par tournoi.

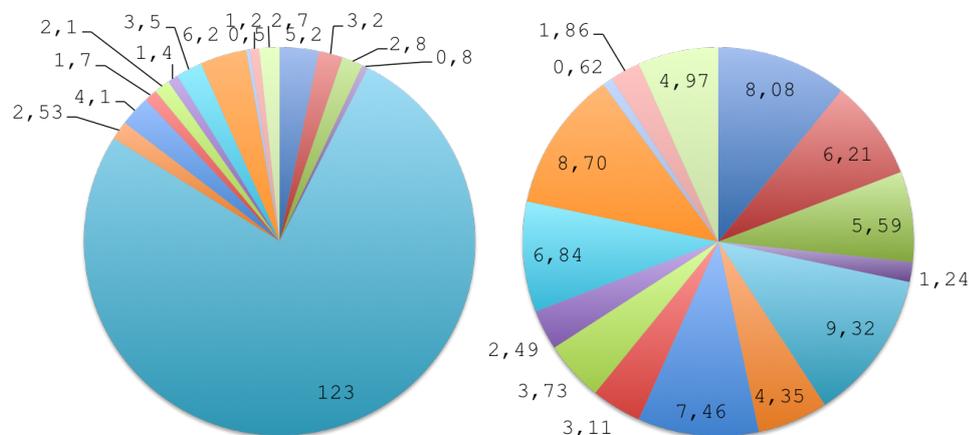


FIGURE 2.6 – Sélection par rang

À gauche, l'exemple de la représentation des individus d'après leur score, à droite celle des individus d'après leur rang. Cette dernière représentation lisse les inégalités et répond au problème de la sur-représentation des scores très élevés.

2.3.5 Le croisement

L'opération de croisement (*crossing-over*) permet de générer deux individus nouveaux à partir de deux individus déjà existants dans la population. Cette opération permet l'exploration de l'espace de recherche. En effet, si on considère deux gènes améliorables par mutation, ils n'ont qu'une faible probabilité de cohabiter au sein d'un même individu lors de l'opération de mutation du processus courant. Or, une opération de croisement permet de combiner rapidement les deux gènes améliorés dans la descendance de deux parents contenant chacun un des gènes mutants. L'opérateur de croisement assure donc le brassage du matériel génétique et l'accumulation des mutations favorables.

Croisement en un point

Pour chaque couple, on choisit au hasard un point de croisement. Chaque chromosome, scindé en deux parties, va échanger une de ses parties avec l'autre chromosome du couple. Pour plus d'efficacité, on pourra effectuer le croisement au niveau binaire (cf. figure 2.7). Ainsi, un chromosome peut être coupé au milieu d'un gène.

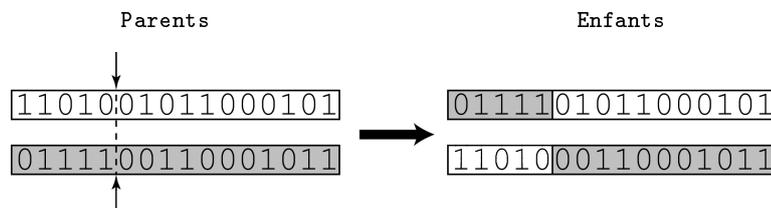


FIGURE 2.7 - **Opération de croisement de deux chromosomes en un point**
 Les deux enfants héritent chacun d'une partie de chaque parent.

Croisement en deux points

Pour le croisement en deux points, on choisit deux points de croisement au hasard. Chaque chromosome échange une partie (sur les trois du découpage) avec l'autre chromosome du couple. Cette méthode est réputée plus efficace que la précédente [3]. D'autres formes de croisement existent : du croisement en k points jusqu'au croisement uniforme.

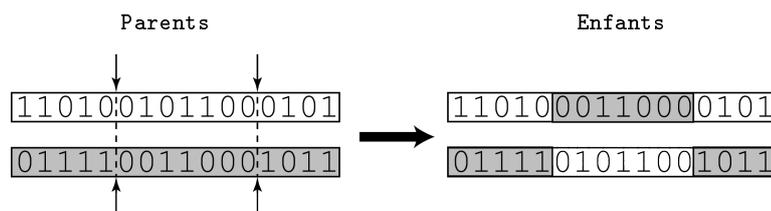


FIGURE 2.8 - **Opération de croisement de deux chromosomes en deux points**
 Les deux enfants héritent chacun d'une partie de chaque parent.

Application au Voyageur de Commerce Croisement

Les deux méthodes de croisement peuvent être appliquées. Néanmoins, il conviendra de gérer les redondances éventuellement générées dans la liste des villes (toutes les villes doivent être visitées et une ville ne peut être visitée une seule fois).

2.3.6 La mutation

Les mutations génétiques interviennent en permanence et à tous les niveaux d'organisation dans les organismes vivants. Par exemple, pour la copie d'un brin d'ADN, la cellule utilise une machinerie qui lit le brin d'ADN à copier et qui construit en temps réel une copie de ce brin. Lors de l'opération de copie, de petites erreurs se produisent et modifient légèrement la copie par

rapport à la source. Ces erreurs sont appelées mutations et leur conséquence varie selon l'emplacement de la mutation : une mutation dans un gène a plus d'impact que dans une séquence non-codante.

Au sein de l'algorithme génétique, nous allons introduire des mutations pour reproduire ce phénomène biologique qui représente un facteur essentiel dans l'évolution des espèces. Une mutation se définit comme la modification aléatoire d'un paramètre du dispositif (ou inversion d'un bit en représentation binaire, cf. figure 2.9). Ainsi, en introduisant du bruit au sein des individus, on évite la stagnation du processus d'évolution car l'opération de mutation permet une exploration locale et globale au sein de l'espace des solutions. En effet, comme dans le processus naturel, la mutation peut avoir des effets variables selon leur localisation. Une mutation peut donc permettre de s'extraire d'un optimum local vers lequel la population entière converge (dérive génétique) [Beasley, 1993b]. Enfin, d'un point de vue mathématique, les mutations garantissent que l'optimum global peut être atteint.

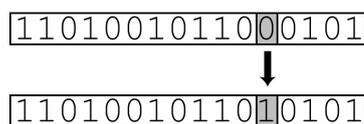


FIGURE 2.9 – Mutation d'un bit dans le code d'un gène

Il existe plusieurs méthodes pour réaliser l'opérateur de mutation. En effet, il n'est pas question de muter tous les bits d'un gène, ce qui reviendrait à casser le processus d'optimisation pour laisser régner l'aspect aléatoire sur les données. En général, la probabilité de mutation P_m par bit et par génération est comprise entre 0,001 et 0,01. On peut aussi choisir $P_m = \frac{1}{L}$ où L est le nombre de bits formant le chromosome. Nous pouvons faire varier ces probabilités d'un gène à l'autre et/ou au fil des générations. On peut imaginer un chromosome qui contient une probabilité de mutation propre à chaque variable et qui serait elle-même soumise au processus d'évolution. L'individu contiendrait alors un chromosome codant les données (gènes) et un chromosome codant les probabilités de mutation de chaque variable. Cette méthode est connue sous le nom d'auto-adaptation et est très efficace lorsque l'environnement subit une évolution rapide. En effet, un environnement instable nécessite une capacité d'adaptation beaucoup plus forte (taux de mutation plus élevée) qu'un environnement stable.



Application au Voyageur de Commerce

Mutation

La mutation sur les numéros de villes doivent s'effectuer au sein de l'ensemble des villes à visiter. Afin de gérer les redondances, la mutation consistera en un échange entre 2 villes au sein de l'individu.

Bibliographie

- [1] A. Reineix A., D. Eclercy, and B. Jecko. Fdtd/genetic algorithm coupling for antennas optimization. *Annales de Télécommunications*, 52(9-10), 1997.
- [2] D. Beasley, D.R. Bull, and R.R. Martin. An overview of genetic algorithms : Part 1, fundamentals. *University Computing*, 15(2) :58–59, 1993.
- [3] D. Beasley, D.R. Bull, and R.R. Martin. An overview of genetic algorithms : Part 2, research topics. *University Computing*, 15(4) :170–181, 1993.
- [4] Charles Darwin. *On the Origin of Species*. London : John Murray, 1859.
- [5] J.H. Holland. *Adaptation In Natural And Artificial Systems*. University of Michigan Press, 1975.
- [6] Vincent Magnin. *Contribution à l'étude et à l'optimisation de composants optoélectroniques*. PhD thesis, Université Lille I, 1998.
- [7] R. Menozzi and A. Piazzzi. Hemt and hbt small-signal model optimization using a genetic algorithm. In *EDMO 97*, number 13-18, London, November 1997.
- [8] R. Menozzi and A. Piazzzi. Shaping the digital optical switch using evolution strategies and bpm. *IEEE Photonics Technology Letters*, 9(11) :1484–1486, 1997.
- [9] J.M. Renders. *Algorithmes génétiques et réseaux de neurones*, 1995.