Algorithmes génétiques

Selvaraj Ramkumar

26 avril 2007

Résumé

Les algorithmes génétiques appartiennent à une catégorie d'algorithmes appelés métaheuristiques, dont l'objectif est de repérer une solution approchée à un problème d'optimisation en un temps raisonnable, lorsqu'il n'existe pas de méthode exacte pour le résoudre ou que la durée de la phase de calcul est trop longue à l'échelle de la vie humaine. Les algorithmes génétiques s'inspirent de la théorie de la sélection naturelle développée au XIX^e siècle par le biologiste Charles Darwin. Les principes de base de cette théorie sont appliqués sur une population initiale de solutions potentielles formant un espace de recherche pour le problème donné.

Cette population initiale est générée à l'aide de trois opérateurs : croisement, mutation, sélection. Les deux premiers sont des opérateurs d'exploration de l'espace, tandis que le dernier fait évoluer la population vers les optima d'un problème.

Table des matières

1 Introduction

La recherche opérationnelle est une discipline jeune ; elle se consacre depuis les prémices de la seconde guerre mondiale à l'élaboration de méthodes performantes permettant de résoudre des problèmes d'optimisation difficiles, et en particulier, des problèmes d'optimisation combinatoire. Un problème est dit combinatoire[1] lorsqu'il englobe un nombre conséquent de solutions admissibles parmi lesquelles une solution optimale ou approchée doit être trouvée. Beaucoup d'applications pratiques touchent aux problèmes combinatoires, notamment en logistique et en production. Comment choisir l'emplacement d'un dépôt ou dans quel ordre visiter les clients d'une même tournée pour réduire au mieux les déplacements, voici deux exemples de problèmes qui se posent fréquemment et qui malgré leur apparente banalité constituent de redoutables défis à la fois pour le gestionnaire de terrain et pour le chercheur.

En effet, on remarque aisément que, malgré le nombre fini de solutions, la stratégie de résolution naïve qui consiste à les énumérer toutes, en les évaluant à chaque fois et en retenant la meilleure, est impraticable car le nombre de solutions grandit généralement de manière exponentielle avec la taille du problème. Il est raisonnable d'utiliser cette technique pour des problèmes de petite taille. D'autre part, l'explosion combinatoire est telle que l'augmentation de la puissance des ordinateurs ne change pas radicalement la situation.

Face au caractère fondamentalement "intraitable" de ces problèmes, il est essentiel de pouvoir en fournir des solutions relativement bonnes en des temps corrects, en développant des méthodes, souvent appelées heuristiques. Ces algorithmes sont généralement spécifiques à chaque problème et leur performance est évaluée empiriquement par comparaison avec d'autres approches sur des bibliothèques de problèmes connus ou des batteries de problèmes générés.

Les métaheuristiques sont généralement des algorithmes stochastiques itératifs, qui progressent vers un optimum par échantillonnage d'une fonction objectif. Les métaheuristiques les plus classiques sont celles fondées sur la notion de parcours. Dans cette optique, l'algorithme fait évoluer une seule solution sur l'espace de recherche à chaque itération. La notion de voisinage est alors primordiale.

Dans la famille des métaheuristiques, il existe aussi les algorithmes génétiques qui manipulent un ensemble de solutions en parallèle, à chaque itération. Je me suis intéressé à ces algorithmes. Je commencerai mon exposé en définissant la notion d'algorithme et en rappellant les principes de base de la sélection naturelle de Darwin. En second lieu, par analogie avec la théorie

[1] Définition tirée du site internet http ://fr.wikipedia.org/

darwinienne, j'expliquerai le fonctionnement des algorithmes génétiques. En outre, je parlerai des applications pratiques que permet l'implémentation de tels algorithmes. De plus, j'appliquerai les principes de base des algorithmes génétiques à l'exemple du voyageur de commerce. Enfin je comparerai les algorithmes génétiques avec d'autres méthodes telles que les algorithmes colonie de fourmis.

2 Concepts de base

2.1 Algorithme

Un algorithme est un moyen de présenter la résolution par calcul d'un problème à une personne ou à une machine. Plus spécifiquement, il s'agit d'un énoncé, dans un langage bien défini, d'une suite d'opérations permettant de résoudre par calcul un problème.

2.2 Heuristique

Les heuristiques sont souvent, à la différence des algorithmes, tirées de l'expérience ou d'analogies, plutôt que d'une analyse scientifique trop complexe. Elles trouvent leur place dans les algorithmes qui nécessitent l'exploration d'un grand nombre de cas, car celles-ci permettent de réduire leur complexité moyenne en examinant d'abord les cas qui ont le plus de chances de donner la réponse.

2.3 Problème d'optimisation

Il s'agit d'un problème dont on cherche la solution optimale. En d'autres termes, on cherche soit un minimum, soit un maximum. Typiquement un problème de maximisation très général peut être formalisé de la manière suivante :

$$\max F(x),\ x \in X$$

où X est l'ensemble de solutions au problème de maximisation et où F désigne une fonction à valeurs réelles mesurant les solutions désignées par x dans X. Il faut maximiser la fonction F sur l'ensemble X de toutes les solutions admissibles.

De manière analogue, on peut formaliser un problème de minimisation.

2.4 Voisinage

La notion de voisinage est primordiale pour la conception d'heuristiques. Pour les problèmes combinatoires, le voisinage a un impact important sur le comportement des métaheuristiques.

Une structure de voisinage est une fonction V qui associe un sous-ensemble de X à toute solution $x \in X$. Une solution $x' \in V(x)$ est dite voisine de x.

Une solution $x \in X$ est un maximum local relativement à la structure de voisinage V si $f(x) \geq f(x')\ \forall x' \in V(x)$.

Une solution $x \in X$ est un maximum global si $f(x) \geq f(x')\ \forall x' \in X$.

3 La sélection naturelle de Darwin

Avant de détailler le fonctionnement d'un algorithme génétique, il serait utile de rappeler quelques principes de base de la sélection naturelle, nécessaires à la compréhension de l'exposé.

La théorie de la sélection naturelle permet de comprendre l'action de l'environnement sur l'évolution des populations. Dans une population d'individus, un caractère présente des variations. Certains individus portent des variations qui leur permettent d'être mieux adapté à leur environnement. Ils ont donc plus de chances de survivre. Ils ont donc une meilleure probabilité de se reproduire. Leur descendance est donc plus nombreuse, et portera cette variation si elle est héréditaire. La conséquence logique est que cette variation héréditaire verra sa fréquence augmenter dans la génération suivante. Et ainsi de suite... On parle d'avantage sélectif. Les variations qui présentent un désavantage sélectif, voient leur fréquence diminuer au fil des générations, et en général finissent par disparaître.

4 Description de l'algorithme

4.1 Canevas de l'algorithme

Les algorithmes génétiques ont été conçus par J. Holland en simulant l'évolution des espèces, la théorie développée par Charles Darwin. De Jongh a ensuite appliqué ces algorithmes à l'optimisation de fonctions à valeurs réelles. Par analogie avec la théorie darwinienne, les solutions les plus adaptées tendent à demeurer plus longtemps dans l'espace de recherche et donc à se reproduire plus aisément pour former éventuellement de nouveaux individus plus performants.

Algorithme 1 : Canevas de base d'un algorithme génétique

Data : Soit $X^0 \subseteq X$, une population initiale générée aléatoirement
while *la règle d'arrêt n'est pas satisfaite* **do**
  Soit $X^{(n)} \subseteq X$, la population courante
  Evaluer le degré d'adaptation de chaque individu
  Sélectionner dans $X^{(n)}$ un ensemble de paires de solutions de haute qualité;
  Appliquer à chacune des paires de solutions sélectionnées un opérateur de "croisement" qui produit une ou plusieurs solutions "enfants";
  Remplacer une partie de $X^{(n)}$, formée des solutions de basse qualité, par des solutions "enfants" de haute qualité;
  Appliquer un opérateur de mutation aux solutions ainsi obtenues;
  Les solutions éventuellement mutées constituent la population $X^{(n+1)}$;
end

Chaque individu de la population est codé de manière appropriée en fonction de la nature du problème. Cet algorithme de base tournera tant que la règle de satisfaction ne sera pas atteinte. Cette règle de satisfaction dépend du degré de précision que l'on souhaite. Pour chaque individu, on mesure et on attribue une note correspondant à son degré d'adaptation au problème. Ensuite, on choisit aléatoirement $\frac{n}{2}$ couples d'individus en fonction des notes attribuées précédemment. Notons qu'il est possible que le meilleur individu n'ait pas été sélectionné. Dans ce cas, il est copié d'office dans la génération intermédiaire à la place d'un individu choisi aléatoirement. Chaque couple donne naissance à deux individus enfants. On peut aussi simuler des mutations en remplaçant au hasard un caractère par un autre.

L'amélioration de la qualité de la population est très rapide au début (recherche globale) ; mais elle devient de plus en plus lente à mesure que le temps passe (recherche locale). Il y a convergence, quand la valeur moyenne de la fonction d'adaptation a tendance à se rapprocher de celle de l'individu le plus adapté. Il y a une uniformisation croissante de la population.

Le temps de calcul des algorithmes génétiques croît en général en $n \ln n$, n étant le nombre de variables.

4.2 Le codage des individus d'une population

Les algorithmes génétiques opèrent sur un échantillon d'individus appartenants à une population. Ces individus sont caractérisés par un chromosome, c'est-à-dire une séquence de gènes. Notons que, dans la littérature, les notions "individu", "chromosome" et "séquence" sont utilisées indifféremment ;

ce sont des synonymes. Dans un souci de clarté, j'utiliserai de préférence le terme "individu". Néanmois dans certains passages (croisement et mutation), j'utiliserai le terme "chromosome" par analogie avec la sélection naturelle de Darwin.

Ils sont ensuite introduits dans des structures de données appropriées en vue d'optimiser leur traitement. L'ensemble de ces chromosomes forme donc la population. Chaque chromosome représente un point de l'espace de recherche des solutions.

L'efficacité d'un algorithme génétique dépend du choix du codage d'un chromosome. Trouver une structure de données et un codage adéquats est dès lors un des objectifs les plus importants.

Pour illustrer l'importance d'un choix adéquat, on peut considérer l'exemple suivant. On peut désigner des personnes par leur nom, leur prénom, leur numéro de téléphone, leur profession, leur adresse, $\cdots$. Lorsqu'on veut envoyer une lettre à une personne, on n'a besoin que du nom de la personne et de son adresse. Pour pouvoir téléphoner à une personne, on n'a besoin que de son numéro de téléphone. À chaque usage correspondra une structure de données plus appropriée qu'une autre.

En organisant les données d'une certaines manière, on favorise leur traitement automatique, efficace et rapide. Adopter une structure de données appropriée pour un traitement informatique peut également contribuer à décomplexifier de manière significative une application informatique et ainsi participer à la diminution du taux d'erreurs.

Cependant, choisir un type de codage adapté ne peut pas être effectué de manière systématique dans l'état actuel des connaissances. Selon les chercheurs dans ce domaine, la méthode actuelle à appliquer consiste à choisir le codage qui semble le plus naturel en fonction du problème à traiter et à développer ensuite l'algorithme de traitement.

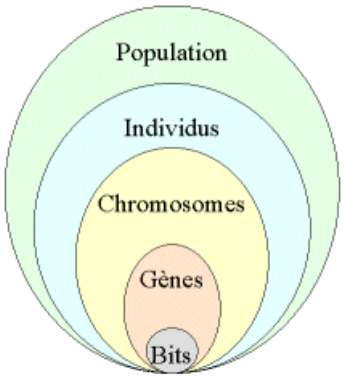

Fig. 1 – Les cinq niveaux d'organisation d'un algorithme génétique.

Voici une description de certains types de codage les plus couramment utilisés.

4.2.1 Codage binaire

Le codage binaire est un codage élémentaire dont le principe consiste à coder la solution selon une chaîne de bits. Une chaîne de bits est une suite de chiffres, chacun d'entre eux pouvant prendre la valeur 0 ou 1. La structure de données traditionnellement utilisée est un tableau, appelé aussi vecteur, de variables booléennes. Chaque composante $x_j, j = 0, ..., N$ de ce vecteur est une valeur booléenne prise par la variable. Ce type de codage est le plus utilisé.

4.2.2 Codage à caractères multiples

Par opposition au codage binaire, une autre manière de coder les chromosomes d'un algorithme génétique est le codage à l'aide de caractères multiples. Souvent, ce type de codage est plus naturel que le codage binaire. C'est d'ailleurs celui-ci qui est utilisé dans de nombreux cas poussés d'algorithmes génétiques.

4.2.3 Codage sous forme d'arbre

Ce codage utilise une structure arborescente ; un arbre est une structure de données munie d'une racine de laquelle peuvent être issus un ou plusieurs fils. Un de leurs avantages est qu'ils peuvent être utilisés dans le cas de problèmes où les solutions n'ont pas une taille finie. En principe, des arbres de taille quelconque peuvent être formés par le biais de crossing-over et de mutations.

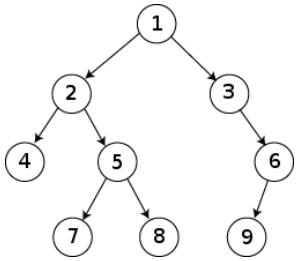

FIG. 2 – Exemple d'arbre

Le problème de ce type de codage est que les arbres résultants sont souvent difficiles à analyser et que l'on peut se retrouver avec des arbres «solutions» dont la taille sera importante, alors qu'il existe des solutions plus simples

et plus structurées à côté desquelles sera passé l'algorithme. De plus, les performances de ce type de codage, par rapport à des codages en chaînes sont encore mal connues. En effet, ce type d'expérience ne fait que commencer et les informations disponibles sont trop rares pour se prononcer.

4.3 Evaluation et sélection

L'évaluation d'un individu ne dépendant pas de celle des autres individus (on peut alors paralléliser cette phase), le résultat fourni par la fonction d'évaluation va permettre de sélectionner ou de refuser un individu pour ne garder que les individus ayant le meilleur coût en fonction de la population courante : c'est le rôle de la fonction de fitness. Cette méthode permet de s'assurer que les individus performants seront conservés, alors que les individus peu adaptés seront progressivement éliminés de la population. En d'autres termes, elle permet, à partir d'un chromosome, de calculer le coût d'un point de l'espace de recherche, de quantifier numériquement la validité de la solution qu'il représente et de mesurer la santé et le degré d'adaptation d'un individu à son environnement. Traditionellement cette fonction est croissante avec la qualité de la solution.

Dans un problème de maximisation, il peut s'agir de la fonction qu'on cherche à optimiser, appelée aussi fonction objectif. Pour des raisons d'efficacité de l'algorithme, on peut être amené à choisir la fonction d'évaluation de façon plus sophistiquée, mais elle sera toujours croissante en la valeur de l'objectif dans un problème de maximisation.

On peut alors définir une fonction de sélection qui, se basant sur l'évaluation précédemment établie, va identifier statistiquement les meilleurs individus d'une population, sélectionner une sous-population à partir de la population parente, éliminer les individus les moins performants, et générer à leur place les plus performants, simulant ainsi le schéma de la "sélection naturelle" de Darwin.

Plus formellement, la sélection tant des individus de "haute qualité" que ceux de "basse qualité" comporte généralement un aspect aléatoire. Chaque individu x_i de la population parmi laquelle se fait la sélection, se voit attribuer une probabilité p_i d'être choisi d'autant plus grande que son évaluation est haute (basse dans le cas d'une sélection de "mauvais" individus). On tire un nombre r au hasard (uniformément) entre 0 et 1. L'individu k est choisi de telle façon que

$$\sum_{i=1}^{k-1} p_i < r \leq \sum_{i=1}^{k} p_i$$

Dans la littérature, il existe de nombreuses méthodes de sélection, complexes ou non, adaptées à certains types de problèmes. Voici quelques-unes de ces méthodes :

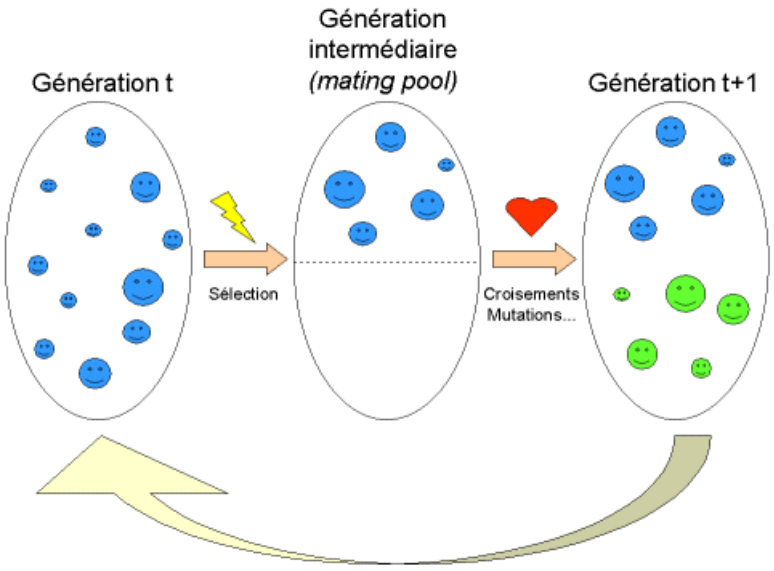

FIG. 3 – Sélection

4.3.1 Sélection par roulette

Il s'agit de la méthode la plus courante. Les individus parents sont sélectionnés proportionellement à leur performance. Meilleur est le résultat fourni par l'évaluation d'un individu, plus grande est sa probabilité d'être sélectionné. Le nombre de fois qu'un individu sera sélectionné est égal à son évaluation divisée par la moyenne de l'évaluation de la population totale. Plus exactement, la partie entière représente le nombre de fois qu'il sera sélectionné, et la partie flottante la probabilité qu'il aura d'être sélectionné à nouveau.

On peut comparer cette méthode de sélection à une roulette de casino sur laquelle sont placés tous les individus de la population, la largeur allouée à chacun des individus étant en relation avec leur valeur d'évaluation. Cette roulette est représentée par la figure suivante.

Ensuite, la bille est lancée et s'arrête sur un individu. Les meilleurs individus peuvent ainsi être tirés plusieurs fois et les plus mauvais ne jamais être sélectionnés. Cela peut être simulé par l'algorithme suivant :

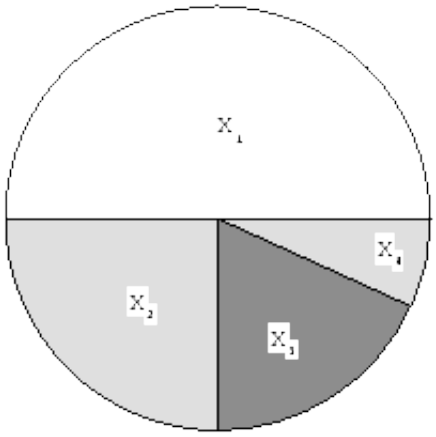

FIG. 4 – Probabilité de chaque individu placé sur une roulette de casino.

Algorithme 2 : Sélection par roulette

On calcule la somme $S1$ de toutes les fonctions d'évaluation d'une population;
On génère un nombre r entre 0 et $S1$;
On calcule ensuite une somme $S2$ des évaluations en s'arrêtant dès que r est dépassé;
Le dernier individu dont la fonction d'évaluation vient d'être ajoutée est sélectionné;

4.3.2 Sélection par rang

La sélection par roulette présente des inconvénients lorsque la valeur d'évaluation des individus varie énormément. En effet, on risquerait d'atteindre une situation de stagnation de l'évolution. Imaginons le cas où 90% de la roulette est allouée à l'individu qui a la meilleure évaluation, alors les autres individus auront une probabilité très faible d'être sélectionnés.

La sélection par rang trie d'abord la population par évaluation. Ensuite, chaque individu se voit associé un rang en fonction de sa position. Ainsi le plus mauvais individu aura le rang 1, le suivant 2, et ainsi de suite jusqu'au meilleur individu qui aura le rang N, pour une population de N individus. La sélection par rang d'un individu est identique à la sélection par roulette, mais les proportions sont en relation avec le rang plutôt qu'avec la valeur de l'évaluation. Avec cette méthode de sélection, tous les individus ont une chance d'être sélectionnés. Cependant, elle conduit à une convergence plus lente vers la bonne solution. Ceci est dû au fait que les meilleurs individus ne diffèrent pas énormément des plus mauvais.

4.3.3 Sélection steady-state

Le principe de base consiste à garder une grande partie de la population à la génération suivante. A chaque génération sont sélectionnés quelques individus, parmi ceux qui ont les meilleures évaluations, pour créer un ensemble d'individus enfants. Les chromosomes ayant les évaluations les plus faibles sont ensuite remplacés par les nouveaux. Le reste de la population survit à la nouvelle génération.

4.3.4 Sélection par tournoi

Soit une population de m individus. On forme m paires d'individus. Ensuite, il faut déterminer un nouveau paramètre, à savoir la probabilité de victoire du plus fort. Cette probabilité représente la chance que le meilleur individu de chaque paire soit sélectionné. En principe cette probabilité doit être grande. Par expérience, une valeur idéale se situe entre 70% et 100%. A partir des m paires, on détermine ainsi m individus pour la reproduction.

4.3.5 Elitisme

A la création d'une nouvelle population, il y a de grandes chances que les meilleurs chromosomes soient modifiés, et donc perdus après les opérations d'hybridation et de mutation. Pour éviter cela, on utilise la méthode élitiste. Elle consiste à copier un ou plusieurs des meilleurs chromosomes dans la nouvelle génération. Ensuite, on génère le reste de la population selon l'algorithme de reproduction usuel. Cette méthode améliore considérablement les algorithmes génétiques, car elle permet de ne pas perdre les meilleures solutions.

4.4 Croisement

Le phénomène de croisement est une propriété naturelle de l'ADN[2]. Il a pour objectif d'enrichir la diversité de la population en manipulant les composantes des individus, c'est-à-dire les chromosomes. C'est par analogie avec la notion d'hydridation de Darwin qu'a été conçu cet opérateur. Classiquement, les croisements sont envisagés avec un couple d'individus parents et génèrent deux enfants.

Cet opérateur favorise l'exploration de l'espace de recherche. Considérons deux gènes A et B pouvant être améliorés par mutation. Il est peu probable que les deux gènes améliorés A' et B' apparaissent par mutation chez un

[2]Acide désoxyribonucléique : ADN est le support de l'hérédité car il constitue le génome des êtres vivants et se transmet en totalité ou en partie lors des processus de reproduction

même individu. Par contre l'opérateur de croisement permettra de combiner rapidement A' et B' dans la descendance de deux parents portant chacun un des gènes mutants. Il est alors possible que la présence simultanée des deux gènes produise un individu encore plus adapté. L'opérateur de croisement assure donc le brassage du matériel génétique et l'accumulation des mutations favorables. En termes plus concrets, cet opérateur permet de créer de nouvelles combinaisons des paramètres des composants.

Voici deux méthodes de croisement classiques : le croisement en un point et le croisement en deux points. Notons qu'il existe une multitude de croisements en fonction de la nature du problème à traiter ; on pourrait aussi considérer le croisement en k points ou croisement uniforme. Je n'aborderai pas ces croisements ici.

4.4.1 Croisement en un point

On choisit au hasard un point de croisement, pour chaque couple. Notons que le croisement s'effectue directement au niveau binaire, et non pas au niveau des gènes. Un chromosome peut donc être coupé au milieu d'un gène, sauf si un bit représente un gène.

Considérons des chromosomes constitués de M bits. Initialement on tire aléatoirement une position de croisement. On échange ensuite les deux sous-chaînes terminales de chacun des deux chromosomes parents $P1$ et $P2$, ce qui produit deux nouveaux chromosomes enfants $E1$ et $E2$.

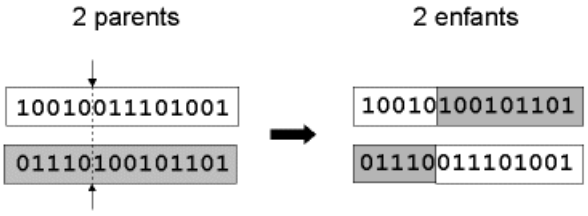

Fig. 5 – Croisement en un point.

Une fois la génération intermédiaire à moitié remplie, les individus sont aléatoirement répartis en couples hermaphrodites[3]. Les chromosomes des parents sont alors copiés et recombinés de façon à former deux descendants possédant des caractéristiques issues des deux parents. On forme ainsi la génération $t + 1$.

[3]Individu morphologiquement mâle et femelle, soit alternativement soit simultanément. Par exemple un escargot

4.4.2 Croisement en deux points

On choisit aléatoirement deux points de croisement. On échange ensuite les deux sous-chaînes situées entre les deux points de croisement de chacun des deux chromosomes parents $P1$ et $P2$, ce qui produit deux nouveaux chromosomes enfants $E1$ et $E2$.

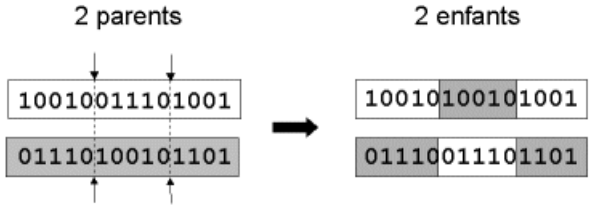

FIG. 6 – Croisement en deux points.

Cet opérateur est généralement jugé plus efficace que le précédent. Néanmoins aucune différence notable n'a été constatée dans la convergence de l'algorithme que nous avons utilisé comme illustration.

4.5 Mutation

Une mutation est l'inversion d'un bit aléatoire dans un chromosome. Un bit valué à 1 prendra la valeur 0 et réciproquement. Les mutations jouent le rôle de bruit en introduisant des innovations et empêchent l'évolution de se figer. Elles assurent en outre une recherche aussi bien globale que locale, selon le poids et le nombre des bits mutés. Plus un individu a été muté, plus la recherche s'éloignera du voisinage local de ce même individu non muté. De plus, elles garantissent mathématiquement que l'optimum global peut être atteint.

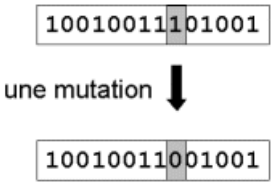

FIG. 7 – Mutation.

D'autre part, une population trop petite peut s'homogénéiser à cause des erreurs stochastiques[4] : les gènes favorisés par le hasard peuvent se répandre au détriment des autres. Une telle situation risque d'aboutir à des résultats qui ne seront pas forcément optimaux. Cet mécanisme de l'évolution, qui

[4] aléatoires

existe même en l'absence de sélection, est connu sous le nom de dérive génétique. Les mutations permettent de contrebalancer cet effet en introduisant constamment de nouveaux gènes dans la population.

Il existe de nombreuses méthodes simulant une mutation. Souvent la probabilité de mutation pm par bit et par génération est fixée entre $0,001$ et $0,01$. On peut prendre également une valeur $pm = \frac{1}{l}$ où l est le nombre de bits composant un chromosome. Il est possible d'associer une probabilité différente à chaque gène. Ces probabilités peuvent être fixes ou évoluer dans le temps.

Si, dans un environnement stable, il est préférable d'avoir un taux de mutation faible, la survie d'une espèce dans un environnement subissant une évolution rapide nécessite un taux de mutation élevé, permettant une adaptation rapide. Les taux de mutation d'une espèce dépendent donc de leur environnement. Après divers essais, il semblerait que la meilleure stratégie à adopter soit la méthode d'auto-adaptation des probabilités de mutation.

Lors de la genèse, les probabilités de mutation sont posées égales à $0,1$. Cette valeur semble bonne par expérience. Au cours du déroulement de l'algorithme, les gènes et les individus ayant des probabilités de mutation trop élevées ont tendance à disparaître. De même, les gènes ayant des probabilités de mutation trop faibles ne peuvent pas évoluer favorablement et tendent à être supplantés. Les probabilités de mutation dépendent donc du gène considéré et de la taille de la population. De plus, elles évoluent au cours du temps. Il y a donc auto-adaptation des probabilités de mutation.

4.6 Optimisation

Cette phase, relativement importante, n'est pas assez mise en valeur dans la littérature concernant les algorithmes génétiques. L'optimisation est l'utilisation d'une méthode de recherche locale, appliquée à chaque individu lors de sa création. Chaque individu est un minimum local, on ne risque donc pas de passer à côté d'une bonne solution.

4.7 Convergence de l'algorithme

L'amélioration de la population est très rapide au début (recherche globale) et devient de plus en plus lente à mesure que le temps passe (recherche locale). Le bruit dans la moyenne est essentiellement dû aux mutations.

Exemple de convergence de l'AG. On a reporté la valeur de la fonction d'adaptation de l'individu le plus adapté de chaque génération (trait), et la moyenne des fonctions d'adaptation (pointillés), pour une population de 200 individus.

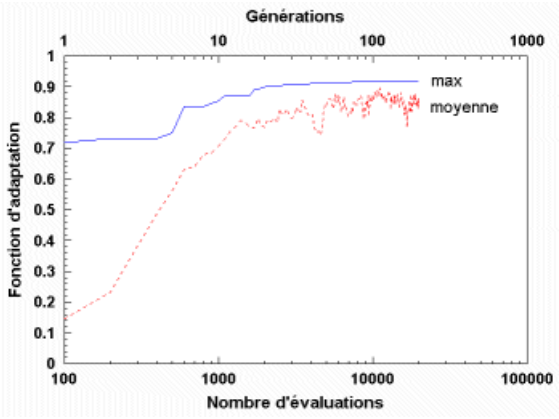

FIG. 8 – Exemple de convergence d'un algorithme génétique.

On voit que la valeur moyenne de la fonction d'adaptation a tendance à se rapprocher de celle de l'individu le plus adapté. Cela correspond à une uniformisation croissante de la population.

Un des intérêts des algorithmes génétiques est que le temps de calcul ne croît pas exponentiellement avec le nombre n de variables, mais plutôt en $n \ln n$. D'autre part, ce temps de calcul est proportionnel au temps de calcul de la fonction d'adaptation, donc du modèle utilisé, et à la taille de la population.

5 Les applications

5.1 Domaines d'application

Les algorithmes génétiques sont de plus en plus utilisés. On les retrouve désormais dans l'industrie, le monde de l'entreprises ou la recherche. Ils permettent de résoudre un large éventail de problèmes difficiles. Voici quelques exemples : ordonnancement des tâches dans un atelier de production, gestion du traffic aérien, coloration de cartes géographiques avec un minimum de couleurs, extraction de données pertinentes (appelée aussi data mining).

Un exemple illustrant les principes de base d'un algorithme génétique est décrit dans le point suivant.

5.2 Voyageur de commerce

5.2.1 Enoncé du problème

Un voyageur de commerce doit visiter n villes données en passant par chaque ville exactement une fois. Il commence par une ville quelconque et termine en

retournant à la ville de départ. Les distances entre les villes sont connues. Quel chemin faut-il choisir afin de minimiser la distance parcourue ? La notion de distance peut être remplacée par d'autres notions comme le temps qu'il met ou l'argent qu'il dépense : dans tous les cas, on parle de coût.

Ce problème, en apparence banal, se complexie de manière factorielle[5] quand le nombre de villes croît linéairement. Voici un tableau illustrant l'accélération fulgurante des chemins possibles et du temps d'énumération. (On suppose qu'une solution est trouvée toutes les microsecondes.)

Villes	Possibilités	Temps de calcul
5	12	12 microsecondes
10	181440	0.18 seconde
15	43 milliards	12 heures
20	$60E + 15$	1928 années
25	$310E + 21$	9.8 milliards d'années

FIG. 9 – Explosion combinatoire en fonction du nombre de villes pour le problème du voyageur de commerce

Notons qu'il existe énormément de variantes et de paramétrages qui affectent grandement les résultats.

Voyons maintenant plus en détail chaque phase de l'algorithme :

5.2.2 Codage

Un codage usuel est une liste ordonnée, par prochaine destination, des noms des n villes. Un individu correspond donc à un trajet respectant les hypothèses du problème. Par exemple pour 8 villes, un individu x sera codé sous la forme

C	B	D	A	F	H	G	C

FIG. 10 – Un individu désignant un trajet

où chaque lettre correspond à une ville

5.2.3 Evaluation

On mesure la longueur de chacun des trajet et on attribue à chacun des individus une note proportionnelle à la longueur mesurée. Le meilleur individu

[5]La factorielle d'un entier naturel n, notée $n!$, est le produit des nombres entiers strictement positifs inférieurs ou égaux à n

correspond à l'individu ayant la longueur de trajet la plus courte.

5.2.4 Sélection

On utilise une stratégie élitiste où les individus sont classés selon leur adaptation. Le premier individu correspond à l'individu ayant la longueur de trajet la plus courte. On crée ensuite un tableau d'entiers, qui comporte autant d'éléments que d'individus dans la population. Chaque entier du tableau prend une valeur croissante entre 1 et max, où max est une borne supérieure. L'intervalle entre les chiffres du tableau est plus grand au début et plus petit à la fin. Il suffit de tirer un chiffre au hasard et de le comparer au chiffre du tableau pour connaître l'individu sélectionné.

Pour que chaque individu ait la même probabilité d'être séléctionné, il suffirait de positionner les chiffres du tableau à un intervalle de $\frac{max}{n}$ où n est le nombre d'individus de la population.

5.2.5 Recombinaison

La règle suivante doit être respectée : si un chemin entre deux villes se trouve dans les deux individus parents, il faut qu'il se trouve dans l'individu enfant.

On utilise le croisement en deux points. Soient deux individus x et y.

$x =$

A	B	C	D	E	F	G	H

$y =$

B	E	F	H	A	D	G	C

Soient 2 et 5, les deux points de croisement.

$x =$

A	B	C	D	E	F	G	H

$y =$

B	E	F	H	A	D	G	C

En croisant les individus, on obtiendrait :

$x' =$

A	B	F	H	A	F	G	H

$y' =$

B	E	C	D	E	D	G	C

Or ce ne sont pas des tours corrects. En effet, trois villes apparaissent deux fois et trois autres n'apparaissent pas. Ces individus ne respectent pas les conditions du problèmes.

Le croisement dit OX permet de contourner cette difficulté. Cet opérateur, conçu spécialement pour les listes ordonnées implémente l'idée suivante.

Les deux solutions parentes sont "préparées" avant l'échange des séquences situées entre les deux positions choisies au hasard.

Dans l'exemple, la zone d'échange de x est préparée à accueillir la séquence des ville F, H, A de y. Pour ce faire, on remplace chacune des villes F, H et A dans le vecteur de x par une place vide symbolisée par un signe $*$.

$x =$

*	B	C	D	E	*	G	*

FIG. 11 – Individu x préparé pour le croisement OX

On commence à la droite de la zone d'échange et on tasse les villes restantes du tour dans l'ordre du tour x en "oubliant" les *.

$x =$

D	E	*	*	*	G	B	C

FIG. 12 – Tassement des villes dans l'ordre à partir du deuxième point de croisement

Les * se retrouvent dès lors dans la zone d'échange, alors que l'ordre de parcours des autres villes n'a pas été changé. On procède de même pour y.

$y =$

B	*	F	H	A	*	G	*

$y =$

H	A	*	*	*	G	B	F

FIG. 13 – Application du croisement OX sur l'individu y

On procède alors à l'échange des séquences, ce qui donne les deux enfants x' et y'.

$x' =$

D	E	F	H	A	G	B	C

$y' =$

H	A	C	D	E	G	B	F

FIG. 14 – Individus x' et y' enfants générés par le croisement OX

Un certain nombre de paires d'enfants sont ainsi générées et remplacent une partie des parents choisis parmi les moins performants.

5.2.6 Mutation

Voici une liste non exhaustive des mutations possibles pour le problème du voyageur de commerce :

– Echange de deux villes

– Renversement de parcours entre deux villes (assez efficace en particulier s'il est appliqué parfois deux ou trois fois à la suite)
– Remplacement d'une liaison entre deux villes par une liaison entre la première ville et la ville la plus proche

6 Autres techniques

Les problèmes cités précédemment sont solubles via d'autres méthodes. Je m'intéresserai ici uniquement aux méthodes qui s'inspirent de phénomènes biologiques.

– **Réseaux neuronaux** : Un réseau de neurones est un modèle de calcul dont la conception est très schématiquement inspirée du fonctionnement de vrais neurones. Les réseaux de neurones sont généralement optimisés par des méthodes d'apprentissage de type statistique, si bien qu'ils sont placés d'une part dans la famille des applications statistiques, qu'ils enrichissent avec un ensemble de paradigmes permettant de générer de vastes espaces fonctionnels, souples et partiellement structurés, et d'autre part dans la famille des méthodes de l'intelligence artificielle qu'ils enrichissent en permettant de prendre des décisions s'appuyant davantage sur la perception que sur le raisonnement logique formel.
  Les réseaux de neurones, en tant que système capable d'apprendre, mettent en oeuvre le principe de l'induction, c'est à dire l'apprentissage par l'expérience.
– **Algorithme colonie de fourmis** : L'idée originale est issue de l'observation du comportement collectif d'exploitation de la nourriture chez les fourmis. En effet, celles-ci, bien qu'ayant individuellement des capacités cognitives très limitées, sont capables collectivement de résoudre le problème de la découverte du plus court chemin entre une source de nourriture et leur nid. Notons que cet algorithme a été développé à l'ULB par le professeur Marc Dorigo de la faculté des Sciences Appliquées.
– **Algorithme glouton** : On appelle algorithme glouton un algorithme qui suit le principe de faire, étape par étape, un choix optimal local, dans l'espoir d'obtenir un résultat optimal global. Dans les cas où l'algorithme ne fournit pas systématiquement la solution optimale, il est appelé une heuristique gloutonne.

7 Conclusion

Les algorithmes génétiques s'inspirent de la théorie de la sélection naturelle de Darwin. Par analogie avec le monde biologique, l'algorithme génétique fait évoluer un échantillon d'individus à l'aide de trois opérateurs principaux : sélection, croisement, mutation.

Ils admettent une grande liberté dans la configuration des paramètres et dans l'implémentation des différents traitements. Pour atteindre des performances optimales, adapter les paramètres de l'algorithme et introduire des méthodes spécifiques au problème traité sont indispensables. On peut également réaliser des algorithmes hybrides mêlant différentes méthodes.

Ils synthétisent des solutions nouvelles et originales à des problèmes connus sans idées préconçues, si ce n'est les paramètres et l'espace de recherche qu'on leur impose. Ils permettent donc d'insuffler un peu de créativité dans l'ordinateur, cette machine qui en est si cruellement dépourvue !

8 Perspectives

Les algorithmes génétiques ont de beaux jours devant eux. Ils ont l'énorme avantage de pouvoir être appliqués sur un large éventail de domaines de recherche de solution. Ils sont utilisés lorsqu'il n'est pas nécessaire d'obtenir la solution optimale, qui prendrait par exemple trop de temps et de ressources pour être calculée. Ils constituent donc une nouvelle voie de recherche ; de nouvelles applications apparaissent et apparaîtront sûrement.

Références

[Aup] A. Aupetit. Les méthodes de résolution du problème du voyageur de commerce. `http://labo.algo.free.fr/pvc/algorithme_genetique.html`.

[Bon95] C. Bontemps. Principes mathématiques et utilisations des algorithmes génétiques. `http://www.toulouse.inra.fr/centre/esr/CV/bontemps/WP/AlgoGene.html`, 1995.

[eJT] Marc Pirlot et Jacques Teghem. *Résolution de problèmes de RO par les métaheuristiques*. Université de Mons-Hainaut.

[Her] A. Hertz. *Métaheuristiques*. Ecole Polytechnique Montréal.

[Mag06] V. Magnin. Optimisation et algorithmes génétiques. `http://magnin.plil.net/spip.php?rubrique8`, 2006.

[Pir06] Marc Pirlot. *Introduction aux métaheuristiques*. Université de Mons-Hainaut, 2006.

[Ren] J. Rennard. Introduction aux algorithmes génétiques. `http://www.rennard.org/alife/french/gavintr.html`.

[Tai] É. D. Taillard. In *Optimisation approchée en recherche opérationnelle*.

[Tol03] Y. Coueque J. Ohler S. Tollari. Algorithmes génétiques pour résoudre le problème du commis voyageur. http://sis.univ-tln.fr/~tollari/TER/AlgoGen1/, 2003.

[Vin06] P. Vincke. *Recherche opérationnelle*. Université Libre de Bruxelles, 2006.

[Wika] Wikipedia. Algorithme de colonies de fourmis. http://fr.wikipedia.org/wiki/Algorithme_de_colonies_de_fourmis.

[Wikb] Wikipedia. Algorithme glouton. http://fr.wikipedia.org/wiki/Algorithme_glouton.

[Wikc] Wikipedia. Algorithme génétique. http://fr.wikipedia.org/wiki/Algorithme_g%C3%A9n%C3%A9tique.

[Wikd] Wikipedia. Algorithmique. http://fr.wikipedia.org/wiki/Algorithmique.

[Wike] Wikipedia. Heuristique. http://fr.wikipedia.org/wiki/Heuristique.

[Wikf] Wikipedia. Réseau de neurones. http://fr.wikipedia.org/wiki/R%C3%A9seau_de_neurones.

[Wikg] Wikipedia. Sélection naturelle. http://fr.wikipedia.org/wiki/S%C3%A9lection_naturelle.