Projet n^{o}1 : Algorithmes génétiques

1 Introduction : les lois de l'évolution de Darwin

Les algorithmes génétiques sont des algorithmes destinés à rechercher le minimum d'une fonction $J : \Omega \to \mathbb{R}$ en utilisant les lois de Darwin d'évolution des espèces. A noter que l'ensemble Ω peut correspondre à divers objets mathématiques (discrets ou continus). Ils consistent à étudier l'évolution d'une population d'éléments (appelés aussi individus), choisie au départ aléatoirement parmi les éléments de Ω. En utilisant les principes de sélection, croisement et mutation, il est en effet observé qu'après un certain nombre de générations, la population se concentre vers le minimum global de J sur Ω. Les algorithmes génétiques binaires sont les premiers algorithmes génétiques ayant été introduits historiquement. Dans ce cas, l'ensemble Ω est constitué de chromosomes, correspondant à une suites de N bits prenant la valeur 0 ou 1. L'objectif de la section 2 est d'écrire un tel algorithme et de l'appliquer à des problèmes simples. Les algorithmes génétiques réels ont ensuite été introduits pour traiter plus facilement le cas où Ω est un sous ensemble de $\mathbb{R}^n$. Il est demandé dans la section 3 d'écrire de tels algorithmes et de comparer leur performance avec ceux des algorithmes binaires.

2 Algorithme génétique binaire

On construit ici avec Scilab un algorithme génétique binaire basé sur l'évolution d'une population de `Npop` individus (ou chromosomes), chacun d'eux formés d'une chaîne de N bits de 0 ou de 1. Par exemple, si $N = 6$, chaque chomosome C s'écrit $C = (b_1, b_2, b_3, b_4, b_5, b_6)$ avec $b_i \in \{0, 1\}$.

Les trois procesus d'évolution Darwiniens sont les suivants :

(i) **Sélection** : la sélection est basée sur le rang de chaque individu dans la population et sur le principe de tirage aléatoire biaisé. En classant les individus de 1 à `Npop` en fonction de leur valeur par la fonction J (l'individu N correspond à celui qui a la plus petite valeur par J), une nouvelle population de `Npop` individus est créée en tirant à chaque fois l'individu i avec la probabilité

$$p_i = \frac{i}{\displaystyle\sum_{i=1}^{Npop} i}$$

.

(ii)**Croisement, cas binaire** : le croisement permet de créer deux nouveaux individus (appelés enfants) à partir de deux autres individus (appelés parents) choisis aléatoirement après la sélection. Il consiste à tirer aléatoirement un entier entre 1 et $N-1$ et à échanger, avec une probabilité p_c, les i premières valeurs des deux parents. Par exemple, si $i = 3$, $N = 6$ et les parents valent respectivement $C_1 = (0, 1, 0, 1, 1, 1)$ et $C_2 = (1, 1, 0, 0, 0, 1)$, les deux enfants vaudront $(1, 1, 0, 1, 1, 1)$ et $(0, 1, 0, 0, 0, 1)$. Cette opération est répétée avec $Npop/2$ couples de parents.

(iii) **Mutation, cas binaire** : la mutation est une opération consistant à modifier chaque enfant créée précédemment. Elle revient à échanger chaque bit d'un chromosome donné (0 donne 1 ou 1 donne 0) avec une probabilité p_m.

1. Construire un algorithme génétique binaire reprenant les principes décrits plus haut et ayant pour paramètre d'entrée $(Npop, Ngen, N, p_c, p_m)$.

2. Appliquer cet algorithme génétique à la recherche du minimum de la fonction $f(C)$ = 'nombre de 1 dans un chromosome C' dont le minimum global est évidemment $C = (0, ..., 0)$ avec les données suivantes : $(Npop, Ngen, N, p_c, p_m) = (40, 100, 30, 0.6, 0.04)$

3. L'algorithme précédent peut être utilisé dans le cas de la minimisation d'une fonction réelle à n variables sur un domaine du type $D = [a, b]^n$. Il suffit en effet de discrétiser tout réel $x = (x_1, ..., x_n)$ dans D en un chromosome $C = (C_1, ..., C_n)$ de taille $N = nL$ où chaque sous-chromosome $C_i = (b_i^1, b_i^2 ..., b_i^L)$ de taille L représente la décomposition binaire du réel x_i sur L bits :

$$x_i = a + \frac{(b-a)}{2^L - 1} \sum_{j=1}^{L} b_i^j 2^{L-j}$$

Ecrire une version de l'algorithme précédent permettant de minimiser la fonction $J(x) = ||x||^2$ sur $[-2, 2]^n$ avec $(n, Npop, Ngen, L, p_c, p_m) = (4, 100, 100, 10, 0.6, 0.04)$.

3 Algorithme génétique réel

Les algorithmes génétiques réels sont adaptés à la recherche du minimum d'une fonction $J : \Omega \subset \mathbb{R}^n \to \mathbb{R}$. Par rapport aux algorithmes génétiques binaires, le seul changement concerne les deux principes de croisement et de mutation. La sélection reste pour sa part inchangée.

(ii) **Croisement, cas réel** : dans ce cas, à partir de deux parents $x_1 \in \Omega$ et $x_2 \in \Omega$, deux enfants x_1' et x_2' sont crées avec une probabilité p_c en selectionnant aléatoirement $\alpha \in [0, 1]$, et en écrivant : $x_1' = \alpha x_1 + (1 - \alpha) x_2$ et $x_2' = (1 - \alpha) x_1 + \alpha x_2$.

(iii) **Mutation, cas réel** : dans ce cas, à partir d'un enfant $x \in \Omega$, on crée, avec une probabilité p_m, un nouvel élement $x' = x + \sigma u$ où u est un vecteur aléatoire suivant une loi normale centrée réduite et σ est un réel strictement positif fixé.

4. Ecrire un algorithme génétique réel permettant de rechercher le minimum d'une fonction définie sur $\Omega = \mathbb{R}^n$.

5. Appliquer cett algorithme à la recherche du minimum de la fonction $J(x) = ||x||^2$ sur $[-2, 2]^4$ avec avec $(n, Npop, Ngen, p_c, p_m, \sigma) = (4, 100, 100, 0.6, 0.9, 0.1)$. Comparer les résultats avec ceux obtenus avec un algorithme génétique binaire.

6. Appliquer cet algorithme à la recherche du minimum de la fonction dite de Rastrigin à 2 variables :

$$J(x_1, x_2) = \sum_{i=1}^{2} \left(x_i^2 - \cos(2\pi x_i) \right) + 2$$

sur $[-5, 5]^2$ qui possède de nombreux minimas locaux mais un seul minimum global en $x = (0, 0)$. On pourra utiliser par exemple $(n, Npop, Ngen, p_c, p_m, \sigma) = (4, 100, 100, 0.6, 0.9, 0.1)$.

Illustrer ces résultats dans le cas $n = 2$ en montrant graphiquement comment se comporte la population des individus au cours des générations.

Comparer les résultats avec ceux obtenus avec la routine Scilab de minimisation `optim`. Commenter.