NUMERICAL OPTIMIZATION

examen de rattrapage, 2heures, sur machine

Le programme Scilab ou Matlab demandé doit être envoyé à l'adresse `laurent.dumas@upmc.fr` *au terme de l'épreuve en indiquant le titre ECP2009 et en donnant un nom au fichier* `ECP2009prenom-nom.sci`

Les algorithmes génétiques binaires sont les premiers algorithmes génétiques ayant été introduit historiquement. Ils sont basés sur l'évolution d'une population de chromosomes, constitués d'une suites de bits prenant la valeur 0 ou 1.

L'objectif est ici d'écrire un tel algorithme et de l'appliquer à des problèmes simples.

A. Construction d'un algorithme génétique binaire

1. Construire avec Scilab ou Matlab 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 Darwiniens sont les suivants :

(i) Sélection : comme dans le cas de l'algorithme génétique réel vu en cours, la sélection est basée sur le rang de chaque individu dans la population et sur le principe de roulette.

(ii) Croisement : il consiste à choisir 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 $(0, 1, 0, 0, 0, 1)$.

(iii) Mutation : elle consiste à échanger chaque bit d'un chromosome donné (0 donne 1 ou 1 donne 0) avec une probabilité p_m.

Un principe d'élitisme similaire à celui utilisé dans le cas réel est rajouté, à savoir que le meilleur individu de la population est forcément conservé à la génération suivante.

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 : $(Npop, Ngen, N, p_c, p_m) = (40, 100, 30, 0.6, 0.04)$

B. Utilisation de l'algorithme précédent pour la minimisation d'une fonction réelle

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 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 $(n, Npop, Ngen, L, p_c, p_m) = (4, 100, 100, 10, 0.6, 0.04)$.