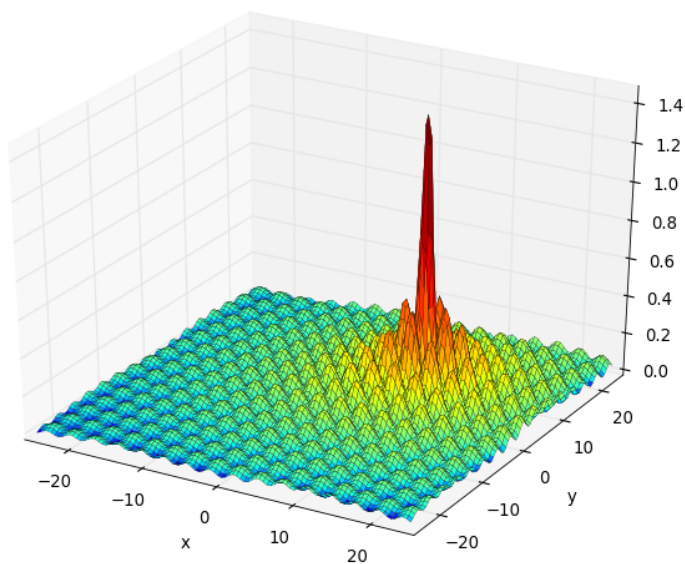


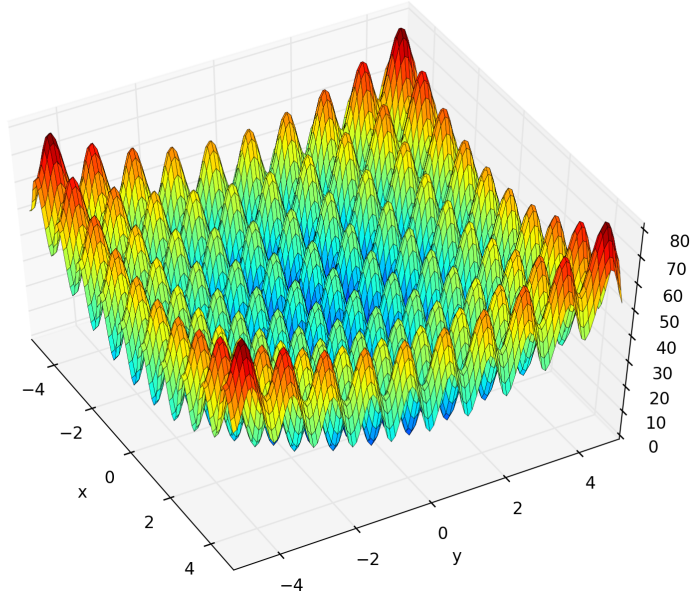
Optimisation numérique grâce aux algorithmes génétiques

Trouver le minimum (ou le maximum) d'une fonction numérique est un des problèmes pour lesquels l'efficacité des algorithmes génétiques a été empiriquement démontrée.

Etant donné une fonction $f : D_1 \times \dots \times D_n \mapsto \mathbb{R}$, il s'agit de trouver un vecteur $\bar{x} = (x_1, \dots, x_n)$ de n paramètres tel que $f(\bar{x})$ soit minimal (ou maximal). Les méthodes d'optimisation analytique, comme la descente en gradient, ont notamment tendance à échouer lorsque f n'est pas convexe, c'est-à-dire lorsqu'elle admet des minima (ou maxima) locaux.

Pour vous en convaincre, regardez les courbes des fonctions suivantes (où $n = 2$).





Le TD propose de résoudre ce problème d'optimisation pour des fonctions réputées difficiles, grâce au paradigme génétique, en adaptant les structures et sous-programmes déjà écrits lors du TP précédent. La principale difficulté de ce TP sera le choix de l'opérateur de croisement.

1 Exercices

Voici 2 fonctions à minimiser :

- $f(\bar{x}) = 10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)]$, avec $D_i = [-5.12, 5.12]$.

f a pour minimum global 0 lorsque $x_i = 0$, pour $0 < i \leq n$.

- $f(\bar{x}) = \sum_{i=1}^n [-x_i \sin \sqrt{|x_i|}]$, avec $D_i = [-500, 500]$.

f a pour minimum global $-418.9829n$ lorsque $x_i = 420.9687$, pour $0 < i \leq n$.

1.1 Encodage

Un individu peut être représenté par un vecteur $\bar{x} = (x_1, \dots, x_n)$ de paramètres, avec $\bar{x} \in D_1 \times \dots \times D_n$, et $D_i \subset \mathbb{R}$.

Modifiez la représentation d'un individu utilisée pour le TP précédent en conséquences.

Modifiez également le sous-programme générant la population initiale.

1.2 Fitness

Si \bar{x}^J est le vecteur de paramètres correspondant à l'individu J , alors la fitness pour J est :

- $f(\bar{x}^J)$ si la fonction doit être maximisée
- $-f(\bar{x}^J)$ si elle doit être minimisée.

Apportez là aussi les modifications nécessaires à votre code.

1.3 Sélection de paires d'individus

La fitness $fit(J)$ d'un individu peut être négative ici. Afin de sélectionner une paire d'individus au sein de la génération en cours, on pourra utiliser une fitness ajustée $fit'(J)$.

Si J_{last} est le plus mauvais individu de la génération en cours, alors :

$$fit'(J_{last}) = 0$$

Et pour tout J de la génération en cours :

$$fit'(J) = fit(J) - fit(J_{last})$$

Apportez les modifications nécessaires à votre code.

1.4 Croisement

Pour le croisement, vous pouvez :

- Choisir un des 4 opérateurs proposés en annexes.
- Définir votre propre opérateur de croisement.

1.5 Mutation

Fixez un taux de mutation t_m relativement faible (par exemple $t_m = 0.05$). Pour chaque x_i , tirez un nombre aléatoire $r = U(0, 1)$, et si $r \leq t_m$, remplacez x_i par une valeur aléatoire dans D_i .

En complément, vous pouvez implémenter l'amélioration présentée en annexe.

2 Annexes

2.1 Croisement

Soient P_1 et P_2 les individus parents, F_1 et F_2 leurs deux enfants, et \bar{x}^J (resp. x_i^J) le vecteur de paramètres (resp. le i ème paramètre) de l'individu J .

$\|JK\|$ dénotera la distance entre J et K dans l'espace des paramètres, et, par abus de notation $\|JK\|_i$ désignera $|x_i^J - x_i^K|$

Voici quelques possibilités pour le crossover, avec leurs avantages et inconvénients (non exhaustif).

2.1.1 Croisement de séquences

Il s'agit du croisement classique entre séquences (de bits, de caractères, ... ou bien dans ce cas de réels), utilisé par exemple ici (équations 4 et 5).

Soit α un entier choisi aléatoirement entre 0 et n .

F_1 hérite des α premiers éléments de P_1 , et des $n - \alpha$ derniers éléments de P_2 .

F_2 hérite des α premiers éléments de P_2 , et des $n - \alpha$ derniers éléments de P_1 .

- avantages :
 - Fidèle à l'approche génétique.
- inconvénients :
 - Pertinent ici seulement à partir d'un nombre n assez élevé de dimensions pour l'espace des paramètres.

2.1.2 "Croisement arithmétique"

Cette solution est notamment décrite ici (équations 10 et 11).

$$x_i^{F_1} = \alpha x_i^{P_1} + (1 - \alpha) x_i^{P_2} \quad (1)$$

$$x_i^{F_2} = \alpha x_i^{P_2} + (1 - \alpha) x_i^{P_1} \quad (2)$$

avec $\alpha = U(0, 1)$

- avantages :
 - $x_i^{F_j} \in D_i$
- inconvénients :
 - Dans l'espace des paramètres, F_1 et F_2 seront localisés à l'intérieur d'un "hyperrectangle" dont les extrémités opposées sont P_1 et P_2 . Par conséquent, à moins d'appliquer un taux de mutation élevé, les individus vont se concentrer au fil des générations vers le centre de l'espace des paramètres.

2.1.3 "Croisement heuristique"

Solution décrite ici (équations 12 et 13).

Supposons que $fit(P_1) \geq fit(P_2)$. P_1 sera alors conservé dans la génération suivante ($F_1 = P_1$), tandis qu'un nouvel individu F_2 sera généré à partir de P_1 , mais en "s'éloignant" de P_2 , cad en se déplaçant dans l'espace des paramètres linéairement à $\overrightarrow{P_2 P_1}$.

$$x_i^{F_1} = x_i^{P_1} \quad (3)$$

$$x_i^{F_2} = x_i^{P_1} + \alpha(x_i^{P_1} - x_i^{P_2}) \quad (4)$$

avec $\alpha = U(0, 1)$

- avantages :
 - Dans l'espace des paramètres, F_2 est nécessairement localisé à l'extérieur de l'"hyperrectangle" défini par P_1 et P_2 .
- inconvénients :
 - Pas de garantie que $x_i^{F_2} \in D_i$. Une solution consiste à générer un nouvel α jusqu'à ce que $\forall i : x_i^{F_2} \in D_i$.

2.1.4 Distance bornée pour chaque x_i

$$x_i^{F_1} = x_i^{P_1} + \alpha \|P_1 P_2\|_i \quad (5)$$

$$x_i^{F_2} = x_i^{P_2} + \alpha \|P_1 P_2\|_i \quad (6)$$

avec $\alpha = U(-0.5, 0.5)$.

Ainsi, quel que soit α :

$$\|F_k P_1\|_i + \|F_k P_2\|_i \leq 2 \|P_1 P_2\|_i$$

- avantages :
 - Dans l'espace des paramètres, F_1 et F_2 peuvent être localisés à l'extérieur de l'"hyperrectangle" défini par P_1 et P_2 .
- inconvénients :
 - Il est possible que $x_i^{F_1} \notin D_i$, ou bien que $x_i^{F_2} \notin D_i$ (mais pas les deux). Ici aussi, une solution consiste à générer un nouvel α , jusqu'à ce que $\forall i : x_i^{F_1} \in D_i \wedge x_i^{F_2} \in D_i$.
 - Le rapport $\frac{\|F_1 P_1\| + \|F_2 P_2\|}{\|P_1 P_2\|}$ tend à augmenter rapidement avec n .

2.2 Mutation

Cette amélioration est décrite ici.

Il s'agit de faire varier le taux de mutation t_m à chaque génération, en fonction des *fitnesses* de la génération précédente. Soient J_{top} le meilleur individu de la génération précédente, J_{med} l'individu médian de la génération précédente, et $fit(J_{top})$ et $fit(J_{med})$ leurs *fitnesses* respectives. t_m peut alors être caractérisé comme inversement proportionnel à $fit(J_{top}) - fit(J_{med})$. Autrement dit, plus les (meilleures) *fitnesses* de la génération précédente sont proches, plus le taux de mutation sera important.