École Supérieure d'Ingénieurs Léonard de Vinci, La Défense
Datascience et IA, S6 2021

Approximation de fonction par Algorithme génétique

Flavien DESEURE–CHARRON

21/03/2021

1 Introduction

L'objectif de ce devoir est de déterminer la température à la surface d'une nouvelle étoile inconnue, à tout instant, à partir de quelques observations.
On sait que la température suit une fonction dîte de *Weierstrass*, définit ainsi:

$$t(i) = \sum_{n=0}^{c} a^n \times \cos(b^n \pi i)$$

Avec $t(i)$ la température de l'étoile à un instant i donnée, avec pour ensemble de paramètres:

$$\begin{cases} A = \{\ a \in \mathbb{R} \mid a \in]0,1[\ \} \\ B = \{\ b \in \mathbb{N} \mid b \in\ [1,20]\ \} \\ C = \{\ c \in \mathbb{N} \mid c \in\ [1,20]\ \} \end{cases}$$

Pour y arriver, il faut donc trouver le couple (a,b,c) approprié.
Pour ce faire, nous avons à notre disposition un ensemble de température mesuré à plusieurs instants i.

2 Espace de recherche

L'espace de recherche a pour taille $]0,1[\times 20 \times 20$.

Cependant, en python la précision maximale est de $10^{15} - 1$.
Il y a donc $(10^{15} - 1) \times 20 \times 20 = 4 \times 10^{16} - 1$ possibilités

3 Fitness

J'ai choisi pour fitness la fonction suivante:

$$f(i, t_i, a, b, c) = \sqrt{\sum_{i=1}^{n} (t(i, a, b, c) - t_i))^2}$$

où t est la fonction de *Weierstrass* avec les paramètres a,b,c entrés en paramètre et n le nombre d'observations.

Cette fonction correspond à la norme 2 du vecteur de la différence entre les observations et les valeurs trouvées avec les paramètres a,b,c.

Le choix de la norme 2 (par rapport à la norme 1) a pour avantage de pénaliser les écarts et donc d'obtenir une fitness plus optimisée.

4 Opérateurs

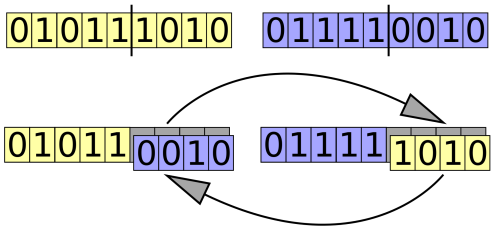
Afin de trouver les meilleurs paramètres pour la fonction, j'ai décidé d'utiliser des opérateurs différents pour les nombres entiers b et c et le nombre réel a.

Je présenterais donc les opérateurs de manière séparée.

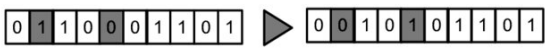
4.1 Nombres entiers

Les nombres entiers sont convertis en binaire puis mis côte à côte afin d'obtenir un chromosome.

1. Croisement
   Méthode utilisée: *Two point crossover*.
   Cela consiste à choisir deux points aléatoirement dans les chromosomes de chaque parent et d'inverser les séquences de gènes entre ces deux points.



2. Mutation
   Méthode utilisée: *Multiple flip bit mutation*.
   Cela consiste à choisir plusieurs gènes aléatoirement dans le chromosome et à inverser leurs valeurs (0 si 1 ou 1 si 0).



4.2 Nombre réel

1. Croisement
   Méthode utilisé: *Simulated binary crossover.*
   Cette méthode imite les propriétés du *Single point crossover*. Une de ses propriétés est que la moyenne des valeurs des parents est égale à celle des valeurs des enfants.

   Les deux enfants sont créés à partir des deux parents à l'aide de la formule suivante:

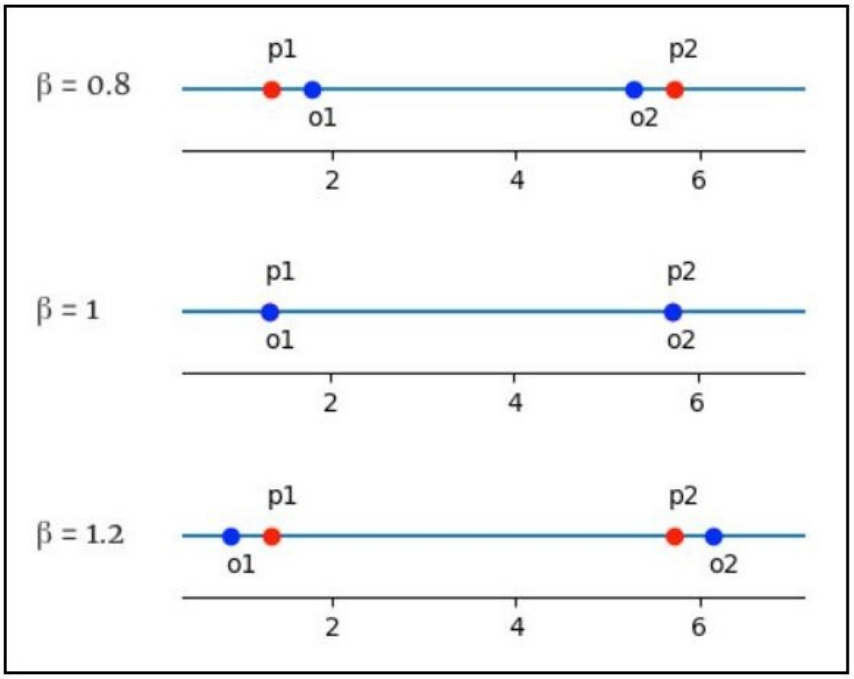
$$offspring_1 = \frac{1}{2}[(1+\beta)parent_1 + (1-\beta)parent_2]$$
$$offspring_2 = \frac{1}{2}[(1-\beta)parent_1 + (1+\beta)parent_]2$$

   Où β, le facteur d'étalement, est defini ainsi:

$$\beta = \begin{cases} (2u)^{\frac{1}{\eta+1}} & \text{si } u \leqslant 0.5 \\ [\frac{1}{2(1-u)}]^{\frac{1}{\eta+1}} & \text{sinon.} \end{cases}$$

   Avec les paramètres suivants:

   - η : compris entre 2 et 5: (2: plus de variations entre les parents, 5: moins de variations)
   - u : uniformément distribuée sur l'intervalle [0, 1]



2. Mutation
   La méthode consiste à multiplier le parent par un facteur compris entre 0.95 et 1.05.

5 Sélection

Pour la sélection, je choisis les meilleurs et moins bons individus.
La proportion est la suivante:

$$\begin{cases} Meilleurs\ individus : \frac{5}{15} \\ Pire\ individus : \frac{2}{15} \\ Nouveaux\ individus : \frac{8}{15} \end{cases}$$

J'ai pris la décision d'utiliser cette méthode assez simple, car elle à l'avantage de garder les meilleurs individus à chaque génération et d'être peu gourmande en temps de calcul tout en apportant, pour mon cas, des résultats équivalent à d'autres algorithmes (roulette wheel, stochastic universal sampling, tournament, ranked-based,...).

6 Solution

J'ai trouvé les paramètres suivants :

$$\begin{cases} a = 0.3415035442650213 \\ b = 15 \\ c = 3 \end{cases}$$

1. Taille de la population: 100 individus
2. Nombre d'itérations moyen pour converger: 57 itérations
3. Temps d'exécution sur 30 itérations:
   - **Seuil pour la fitness: 0.19**
     - Moyenne: 6.5 secondes
     - Écart-type: 4.5 secondes
   - **Seuil pour la fitness: 0.1813**
     - Moyenne: 21 secondes
     - Écart-type: 27 secondes

   Note: Le temps d'exécution est assez important car la précision demandée est élevée. En diminuant les exigences (par exemple avoir une précision de 10^{-2}), ce temps est beaucoup plus faible.

7 Discussion

J'ai testé différentes configurations pour ce problème, chacune ayant ses avantages et ses inconvénients.

7.1 Version Binaire

Description:
J'ai testé une méthode consistant à convertir a,b et c en binaire et les mettre côte à côte pour former un chromosome.
Note: Pour a, on choisit une précision (10^{-2} par exemple).
Avantages:

- Rapide
- Très précis si l'on connaît la précision à l'avance

Inconvénients:

- Peu stable
- Peu précis si l'on ne choisit pas la bonne précision

7.2 Modification des Hyperparamètres

7.2.1 Taille de la population

En augmentant la taille de la population (tout en diminuant proportionnellement le nombre d'itération), on n'obtient pas de meilleurs performances

7.2.2 Proportion pour la sélection

Différentes proportions on été testées, notamment la suppression des individus les moins bons (élitisme). Cela s'avère peu efficace (divergence).

7.2.3 Opérateur croisement

J'ai testé le *single point crossover*. Cependant, cette méthode était moins efficace que le *two point crossover*.

7.2.4 Opérateur mutation

J'ai testé la mutation non-uniforme de Michalewicz. Cependant, le ratio précision de la solution et performances était trop faible pour justifier son utilisation.

7.2.5 Fitness

Autre fitness possible:

$$f(t_i, i, a, b, c) = \frac{\sum |t(i, a, b, c) - t_i|}{\sum |t_i|} * 100$$

Elle correspond à un pourcentage d'erreur.