

# IA : Approximation de fonction par algorithme génétique

## Plan d'action

- Taille de l'espace de recherche

L'espace de recherche choisit pour résoudre le problème a été d'initialiser une population de 100 combinaisons (a, b, c), que je modifie à chaque itération. Etant donné que j'ai besoin d'entre 150 et 200 itérations afin d'arriver à une solution stable, et que je garde les 20 meilleures combinaisons à chaque itération, l'espace de recherche total est d'environ :

$$E = 100 + 80.n_{150 < n < 200}$$

- Fonction Fitness

La fonction fitness que j'ai réalisé consiste à prendre les 3 valeurs a, b et c en paramètres puis en parcourant l'ensemble des valeurs des i et des T(i) de l'échantillon fournit dans le fichier csv, je calcule la température en fonction des 3 paramètres puis je la compare avec celle de l'échantillon. La Fitness retourne ensuite la moyenne des valeurs absolues des écarts.

- Opérateurs mis en œuvre et processus de sélection

J'ai fait une fonction qui rassemble les étapes de sélections, de croisements et de mutations. Je commence donc par sélectionner les 20 meilleurs individus de ma population. La sélection est réalisée en triant le dictionnaire *Combinaison/Fitness* (représentant ma population) par valeur de Fitness croissante.

Ensuite vient l'étape où je reconstruis ma population à partir de ces 20 meilleurs individus. J'ai fait une fonction qui crée un individu en croisant les paramètres a, b et c de deux éléments parents choisis aléatoirement. Enfin, je choisis 10 enfants aléatoirement dans ceux que je viens de créer, puis je change un des 3 paramètres au hasard, en y associant une nouvelle valeur afin de faire évoluer les paramètres de la population.

- Taille de la population et nombre d'itérations nécessaires

La population est tout le temps composé de 100 individus, et il faut entre 150 et 200 itérations du processus décrit ci-dessus avant d'atteindre une solution stable des paramètres a, b et c.

- Temps de run

Le programme met en moyenne 1.67 secondes afin de trouver la solution sur 200 itérations.

- Différentes solutions

Tout d'abord, j'avais fait une première solution mais sans effectuer l'étape de mutation sur les dix enfants comme expliqué ci-dessus. Cependant, je me suis rendu compte que si les paramètres solutions ne se trouvaient pas dans les 100 premiers individus, alors la solution qui serait retournée serait la meilleure de cette population mais pas la solution la plus précise possible. Cette étape de mutation permet donc d'amener de la diversité à la population et permet de laisser constamment une chance au programme de trouver une solution avec de nouveaux paramètres encore plus précise à chaque itération.

J'ai donc aussi fait une méthode qui gardait seulement les 20 meilleurs éléments puis qui remplissait la population avec 80 individus complètement nouveaux sans croisement ni mutation. Cependant, le phénomène de convergence est donc plus long car en suivant ce raisonnement, nous pouvons aussi créer des individus malheureusement très éloignés des paramètres solutions.