

Introduction à l'intelligence artificielle

Partie I : Optimisation bio-inspirée

TP 2 : Initiation à l'algorithmique génétique

Wahabou ABDON
wahabou.abdou@u-bourgogne.fr
2024 - 2025

On souhaite utiliser un algorithme génétique pour trouver le plus grand nombre (décimal) que l'on peut coder sur n bits. Cet exemple est purement académique et à intérêt pédagogique, la solution optimale pouvant être facilement devinée. L'algorithme génétique devra apprendre à identifier la structure des solutions intéressantes. Dans ce cas précis, l'algorithme devra "se rendre compte" que les solutions ayant des bits de poids forts à 1 sont de bons individus.

Exercice 1 : Représentation des solutions (codage) et définition de la fonction objectif

1. Quelle représentation utiliseriez-vous pour coder vos solutions
 - a) Quelles sont vos variables de décision (gènes) ?
 - b) Quelles sont les valeurs possibles pour chaque gène (valeur minimale, valeur maximale) ?
 - c) À quoi correspond votre chromosome ?
2. Quelle est la taille de l'espace de recherche ?
3. Définissez la fonction objectif du problème posé.

Exercice 2 : Prise en main du framework d'algorithme génétique

Pour vos travaux pratiques un squelette d'algorithme vous est proposé. Il comprend quelques modules fournis, et d'autres à implémenter. Dans un premier temps, vous êtes invités à exécuter le programme fourni afin de comprendre son fonctionnement. Pour ce faire vous devez exécuter le code fourni dans la classe `Main`. Si vous le souhaitez, vous pouvez modifier les valeurs des paramètres suivants (dans la classe `Algorithme_genetique`) :

- taille de la population ;
- nombre de génération ;
- probabilité de croisement ;
- probabilité de mutation.

Exercice 3 : Implémentation

1. Écrivez une classe permettant de réaliser une sélection basée sur une roulette biaisée (vous pouvez vous inspirer de la classe `Selection_aleatoire`).
2. Modifiez le choix de l'opérateur de sélection dans la classe `Algorithme_genetique` afin de tester votre roulette biaisée.
3. Écrivez une classe permettant de réaliser un croisement 2-points (vous pouvez vous inspirer de la classe `Croisement_1point`).
4. Modifiez le choix de l'opérateur de croisement dans la classe `Algorithme_genetique` afin de tester votre croisement 2-points.