

Projet Algorithmes Génétiques I

Coloration de graphe

1 Présentation du projet

Vous allez devoir réaliser dans ce projet en binômes (+ ou - 1 accepté si justifié), un algorithme génétique permettant de résoudre le problème de la coloration d'un graphe, consistant à partir d'un graphe donné, de déterminer le nombre chromatique de ce graphe.

Le nombre chromatique d'un graphe est le nombre minimum de couleurs distinctes nécessaires afin de colorier l'ensemble des sommets du graphe de telle sorte que deux sommets adjacents aient une couleur différente

2 Pourquoi un algorithme génétique

La résolution de ce problème à partir de 3 couleurs n'est pas possible de manière efficace avec des parcours exhaustifs de l'ensemble des possibilités. Nous devons donc nous rabattre sur un algorithme fournissant une solution approchée dans un temps raisonnable, ici un algorithme génétique.

3 Modélisation du problème

Nous allons utiliser la représentation d'un graphe issue de la banque de données <http://mat.gsia.cmu.edu/COLOR04/>, qui nous permettra d'avoir accès à de nombreux graphes exemples de différentes tailles et connexités. Ceux-ci nous permettront également de valider notre algorithme sur des données communes.

4 Codage du graphe

Vous allez commencer par construire un graphe à partir d'un fichier de cette banque de données. Vous choisirez au choix la représentation matricielle ou celle en liste de successeurs.

5 Initialisation de la population

Vous allez ensuite engendrer une population initiale via un algorithme glouton, consistant simplement à tirer un sommet au hasard et de calculer une solution en utilisant un parcours profond.

Exercice 1 : Première séance

- Ecrire une fonction C permettant de construire le graphe à partir d'un fichier texte.
- Choisir un codage pour représenter une solution.
- Ecrire une fonction d'évaluation de votre solution.
- Ecrire l'algorithme glouton décrit ci-dessus afin d'avoir une méthode pour engendrer votre population initiale.

6 Fonction de sélection

Maintenant que nous avons engendrer une population d'individus solutions à notre problème de coloriage de graphe, nous allons sélectionner certains de ces individus afin d'engendrer de nouveaux individus par croisement.

Exercice 2 : Seconde séance

- Ecrire une première fonction de sélection triviale, correspondant à la sélection des k meilleurs individus.
- Ecrire une seconde fonction de sélection correspondant à une sélection proportionnelle, sélectionnant un individu avec une probabilité dépendant de sa fonction d'évaluation
- Ecrire une troisième fonction de sélection selon le principe du tournoi avec ou sans remise.
- Ecrire enfin une dernière fonction de sélection sur le principe de la roulette.

7 Fonction de Croisement

Nous avons sélectionné une partie de notre population afin de créer de nouveaux individus à partir de couples issus des individus sélectionnés. Nous allons donc maintenant écrire les fonctions permettant d'effectuer ce croisement.

Exercice 3 : Troisième séance

Ecrire une fonction permettant de

- choisir 2 individus parmi les individus sélectionnés.
- placer un ou plusieurs points de coupure sur ces individus
- générer les 2 nouveaux individus par remplacement des gènes compris entre les points de coupure.

8 Fonction de Mutation

Afin de diversifier les solutions et d'éviter la stagnation dans un optimum local, nous allons maintenant effectuer une dernière transformation consistant à muter certains gènes de nos solutions en fonction d'un coefficient de mutation généralement inférieur à 0,3.

Exercice 4 : Troisième séance (suite)

Ecrire une fonction permettant de muter une solution.

9 Boucle générale

Nous allons maintenant réutiliser l'ensemble des fonctions précédentes afin d'écrire la boucle générale consistant à:

1. Sélectionner k individus parmi la population courante
2. Appliquer votre fonction de croisement sur cette population
3. Appliquer votre fonction de mutation sur la population issue du croisement.

Exercice 5 : Troisième séance (fin)

Implémenter cette fonction générale afin de tester votre algorithme, en paramétrant le plus possible votre algorithme (coefficient de croisement, coefficient de mutation, taille de la population, type de sélection, cas d'arrêt, ...)