

Algorithmique Avancée

Metz Numeric School
M2I



Programme

- Introduction
- Algorithmique génétique



Introduction

Théorie biologique :

- Sur une population donnée
 - Chaque spécimen est unique
 - Certains survivent mieux dans un environnement donné
 - Les autres moins bien



Introduction

Théorie biologique :

- La reproduction de deux spécimens permet :
 - D'introduire des mutations (nouvelles propriétés)
 - De renforcer certains traits (taille, vitesse, ...)
 - De combiner certaines compétences



Introduction

Algorithmique évolutionniste :

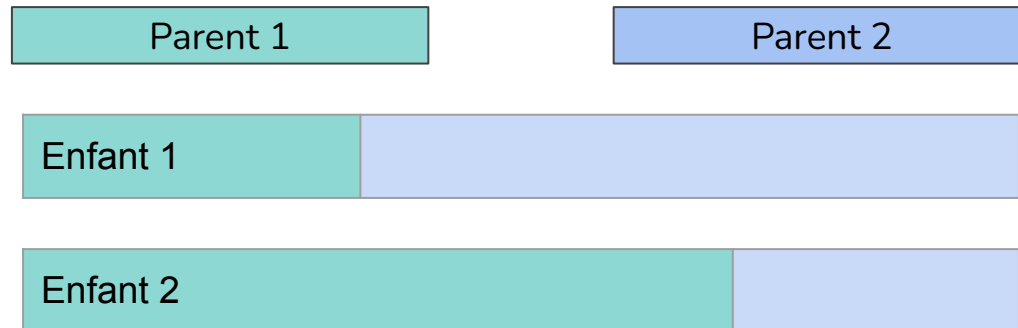
- Appliquer les principes d'évolution biologiques
 - Faire évoluer une population de solutions
 - Sélectionner les meilleures
 - Les faire se reproduire
 - Introduire des mutations



Algorithmes génétiques

Algorithmique génétiques :

- Représenter le problème sous forme de chaînes (encodage)
- Se concentrer sur la recombinaison (crossover)





Introduction

Algorithmique génétique :

- Problématique combinatoire
- Représentations discrètes (valeurs entières par exemple)

Algorithmique évolutionniste :

- Optimisation continue
- Le crossover est moindre face à la mutation (voir absent)
- Mécanismes de sélection différent



Application

Cas du Voyageur itinérant (Travelling Salesman Problem)

Étape 1 : Encodage des paramètres

Le parcours des villes est représentable sous forme d'un tableau

- On dit que les villes sont les gènes
- Le tableau est un chromosome
- Le premier et le dernier élément sont fixes (ville de départ)



Application

Cas du Voyageur itinérant (Travelling Salesman Problem)

Étape 2 : Calculer le score d'adaptation (Fitness)

- Dans notre cas, l'objectif est de réduire la distance
- Le score de fitness est donc l'inverse de la distance entre les villes



Application

Cas du Voyageur itinérant (Travelling Salesman Problem)

Étape 3 : Survival of the fittest

- Sélectionner les X chromosomes les plus performants
- Procéder à la reproduction et au croisement des survivants
 - Attention, les chromosomes doivent être viables
 - On n'aura qu'une seule fois chaque ville



Application

Cas du Voyageur itinérant (Travelling Salesman Problem)

Étape 4 : Mutation

- Selon un taux fixé, on va :
 - permuter aléatoirement 2 villes
 - inverser une séquence



Application

Cas du Voyageur itinérant (Travelling Salesman Problem)

Étape 5 : Élitisme

- Parfois, plutôt que de remplacer toute une population
 - On conserve les meilleurs éléments dans la génération suivante



Application

Cas du Voyageur itinérant (Travelling Salesman Problem)

Quand est-ce qu'on s'arrête ?

- Soit une fois que les résultats sont stables
- Soit après un nombre d'itérations données



Application

Cas du Voyageur itinérant (Travelling Salesman Problem)

Au lieu d'une solution en $O(n!)$, on retombe sur une solution $O(n^2)$
(parcours du nombre d'entrée et fonction de fitness)



Application

A vous d'implémenter la solution dans le langage de votre choix