

Mathématiques - Recherche opérationnelle  
TP noté Python  
**Voyageur de commerce**

Benjamin LE GALLO  
Steven PELLLET  
C2

5 juin 2018

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Algorithmes génétiques</b>	<b>3</b>
2.1	Principe des algorithmes génétiques . . . . .	4
2.2	Sélection . . . . .	4
2.3	Croisement . . . . .	4
2.4	Mutation . . . . .	4
2.5	Adaptation . . . . .	4
<b>3</b>	<b>Fonctionnement (Python)</b>	<b>5</b>
3.1	Étapes . . . . .	5
3.1.1	Initialisation . . . . .	5
3.1.2	Générations (itérations) . . . . .	5
3.1.3	Résultat . . . . .	5
3.2	Exécution . . . . .	6

# Chapitre 1

## Introduction

Le voyageur de commerce est un **problème d'optimisation** qui, étant donné une liste de villes, et des distances entre toutes les paires de villes, **détermine un plus court chemin qui visite chaque ville une et une seule fois et qui termine dans la ville de départ**.

C'est un problème algorithmique célèbre, qui a généré beaucoup de recherches et qui est souvent utilisé comme introduction à l'algorithmique ou à la théorie de la complexité. Il présente de nombreuses applications que ce soit en planification et en logistique, ou bien dans des domaines plus éloignés comme la génétique (en remplaçant les villes par des gènes et la distance par la similarité).

Résoudre un problème d'optimisation, c'est trouver une valeur des paramètres d'un problème qui donne une solution meilleure que les autres. Une façon de trouver les meilleurs paramètres consiste à essayer toutes les valeurs possibles. Lorsque le nombre de paramètres possibles est grand cela peut se révéler impraticable.

Une technique mise au point dans les années 1960 est inspirée de la sélection naturelle : il s'agit des **algorithmes génétiques**.

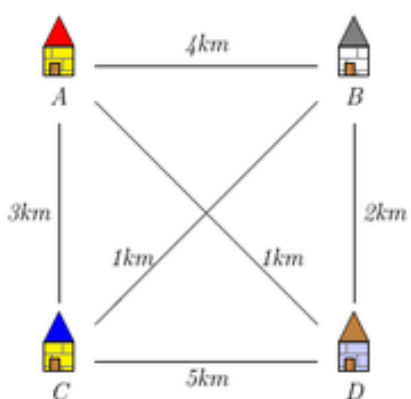


FIGURE 1.1 – Instance du problème

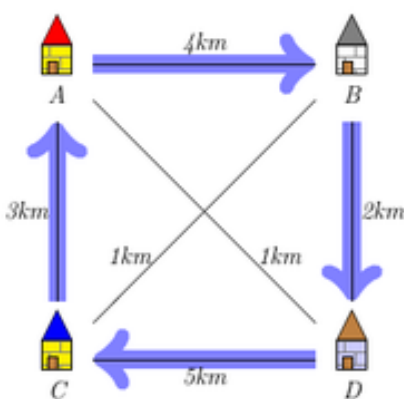


FIGURE 1.2 – Possible solution du problème

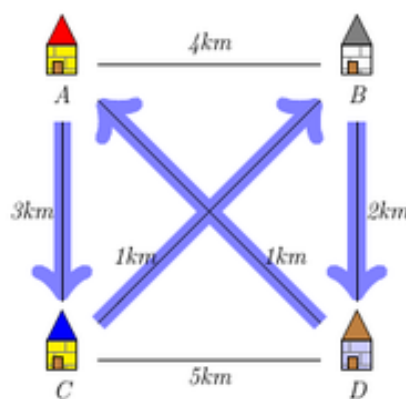


FIGURE 1.3 – Solution optimale du problème

# Chapitre 2

## Algorithmes génétiques

La **sélection naturelle** permet de sélectionner les individus les mieux adaptés à l'environnement (Darwin). Cette sélection par **reproduction** : les individus sélectionnés se reproduisent en mélangeant leurs gènes, et ce sont les individus les mieux adaptés qui se reproduisent le plus. Ils transmettent leurs gènes (qui sont supposés être le mieux adaptés) à leur descendance. Ainsi, sur une population nombreuse et un grand nombre de générations, les gènes les mieux adaptés seront de plus en plus représentés dans la population. D'autres processus d'évolution du matériel génétique existent dans la nature : les **mutations**.

Les algorithmiques génétiques sont **une manière de mimer le processus de sélection naturelle** décrit plus haut. On suppose que l'on connaît une fonction-objectif qui à **chaque individu associe un nombre qui évalue sa performance** (par convention : plus la valeur est basse, meilleure est la performance). On cherche un individu qui a une valeur de la fonction-objectif minimale.

Chaque individu **dépend d'un certain nombre de paramètres** que l'on peut coder dans une chaîne (par exemple binaire) de longueur  $l$ . Générer tous les individus possibles est hors de portée car on se place dans un cas où le nombre total de paramètres possibles est très grand.

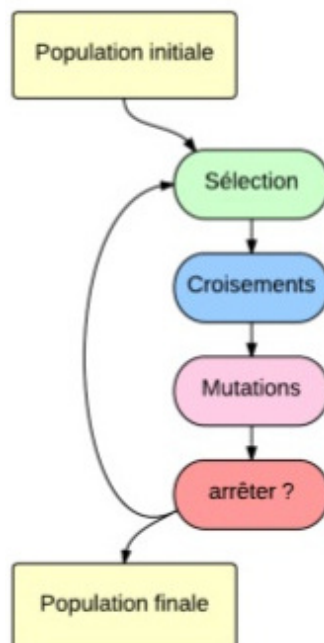


FIGURE 2.1 – Illustration de l'algorithme génétique

## 2.1 Principe des algorithmes génétiques

Un individu est **une solution du problème**. Une population représente **tous les individus, toutes les solutions**.

## 2.2 Sélection

La méthode par sélection permet **d'éliminer des individus, les solutions les moins adaptés**.

## 2.3 Croisement

La méthode par croisement consiste à croiser deux individus, deux solutions qui permet **de produire de nouveaux**.

## 2.4 Mutation

La méthode par mutation consiste **à modifier aléatoirement un individu, une solution**.

## 2.5 Adaptation

La méthode d'adaptation consiste **à évaluer un individu, une solution par rapport au problème à résoudre**.

# Chapitre 3

## Fonctionnement (Python)

### 3.1 Étapes

#### 3.1.1 Initialisation

Voici les étapes d'initialisation nécessaire avant de commencer les itérations pour ne garder que les meilleurs solutions.

- **Création de la map**, elle représente la distance à parcourir entre chaque ville
- **Définition de la population** (villes), à partir du nombre de villes et de solutions voulues, nous obtenons une population de base
- **Générer des solutions**, créer un certain nombre de solutions (définis par l'utilisateur), ces solutions représentent l'ordre des villes à parcourir pour obtenir une solution optimal

#### 3.1.2 Générations (itérations)

Ces itérations permettent de sélectionner les meilleures solutions puis d'en créer de nouvelles à partir de ces dernières. L'objectif est de se rapprocher de la solution optimal !

- **Sélection par élitisme**, elle consiste à retourner une population ( $m$ ) plus performante
- **Sélection par roulette**, elle consiste à choisir au hasard deux solutions et ne conservera que la meilleure des deux pour ensuite retourner une population ( $m$ ) plus performante
- **Croisement**, permet de créer de nouvelles solutions se rapprochant des meilleures solutions actuelles, pour obtenir une population ( $2m$ ).
- **Mutation**, elle échange deux villes dans une solution, elle permet elle aussi de se rapprocher de la solution optimal

#### 3.1.3 Résultat

- **Sélection la meilleure solution**, il s'agit de ne garder que la meilleur solution, la solution optimal
- **Ordre des villes**, il s'agit d'afficher l'ordre des villes à parcourir avec la distance optimal
- **Calcul de la distance parcourue**, à partir de la map et de la meilleure solution, elle retourne la distance à parcourir

## 3.2 Exécution

Voici un exemple de résolution du problème du voyageur de commerce par un utilisateur.

```
Écrire le nombre de villes à parcourir
> 7
Écrire le nombre de solutions que vous voulez obtenir
> 15
Écrire le nombre d'itérations (générations)
> 8

    1. Sélection par élitisme
    2. Sélection par roulette (aléatoire)

Choisissez une méthode de sélection
> 1

Écrire le taux de mutation (entre 0 et 100 (%))
> 50
```

FIGURE 3.1 – Initialisation des variables

```
**RÉSULTATS**
Distance a parcourir: 12 km
Ordre des villes: [1 6 5 4 7 3 2]
```

FIGURE 3.2 – Résolution du problème du voyageur de commerce