Algorithme génétique: application au problème du voyageur de commerce.

Mathieu Mandret

Contents

1 Introduction

1.1 Le problème du voyageur de commerce

On énonce la situation suivante: Un commerçant doit se rendre dans une liste de villes données. Il doit passer une seule fois par chaque ville et revenir à sa ville de départ à la fin de son voyage. On veut pouvoir savoir dans quel ordre il doit visiter les villes pour parcourir le moins de distance possible. Mais il se pose un problème d'explosion combinatoire, en effet, pour n villes, il existe $n!$ ordres de parcours.
Quelques exemples:

Pour $n = 10$ on a $n! = 3628800$, donc pour un parcours contenant 10 villes, il existe plus de 3 millions de parcours possibles.

Pour $n = 30$ on a $n! \simeq 2.65 \times 10^{32}$

Une approche déterministe n'est donc pas envisageable, il faudrait générer toutes ces permutations puis les évaluer unes par une pour trouver la meilleure ce qui prendrait un temps considérable.

On observe que la fonction $f : x \to x!$ croît extrêmement vite:

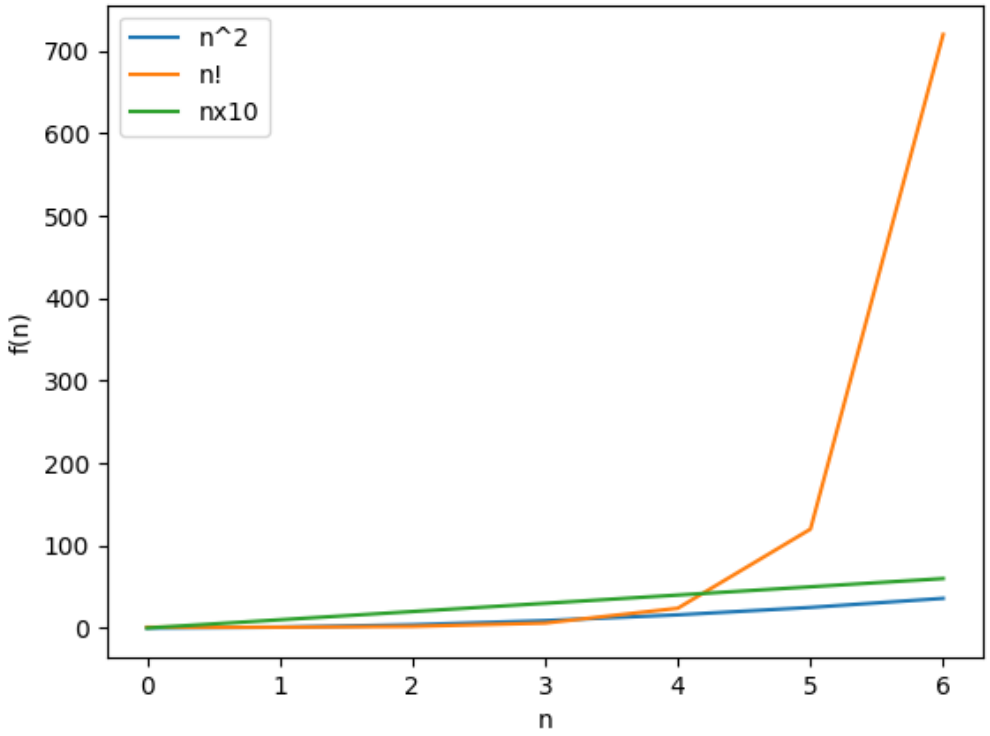

Figure 1: Croissance des fonctions $n^2, n * 10$ et $n!$

1.2 Les algorithme génétiques

On peut toutefois espérer trouver une solution approchée à ce problème avec un algorithme génétique.

C'est un type d'algorithme évolutionniste dont le but est de trouver une solution approchée à un problème d'optimisation. L'algorithme génétique est inspiré de la théorie de l'évolution qui dit que:

- Une espèce connaitra forcémment des variations aléatoires
- Si la variation est génante pour l'individu, il ne se reproduira pas ou peu, et cette variation disparaitra
- Si cette variation est avantageuse, il se reproduira plus et elle se diffusera dans les générations futures.

Dans un algorithme génétique, on retrouve toujours les composantes suivantes:

1.2.1 L'individu

C'est simplement une solution potentielle au problème.

1.2.2 La population

Une population est un ensemble d'individus divers. Analogiquement à la biologie, c'est une espèce.

1.2.3 La "fitness"

C'est une valeur associée à chaque individu, elle permet de quantifier à quel point une solution est adaptée au problème. Et des opérations permettant de faire évoluer une population vers une génération meilleure:

1.2.4 L'évaluation

Elle consiste à analyser tous les individus de la population pour associer à chacun une valeur de fitness.

1.2.5 La séléction

C'est la méthode qui permet de choisir dans la population 2 parents pour générer un individu fils. Si on fait le parallèle avec la théorie de l'évolution, cette opération représente la selection naturelle, les individus avec les meilleures caractéristiques, et donc la meilleure fitness, on plus de chance de survivre, ce qui est matérialisé par le fait qu'ils ont plus de chance d'être séléctionnés pour se reproduire et transmettre leurs caractéristiques au générations futures.

1.2.6 Le croisement, ou "crossover"

Croiser 2 individus représente le processus de reproduction dans la nature. Il revient à créer un individu fils en combinant 2 parents, les caractéristiques du fils seront alors un mélange aléatoire de celles des parents.

1.2.7 La mutation

Un mutation est un changement aléatoire des caractéristiques d'un individu.

Pour générer une solution approchée, un algorithme génétique suit le déroulement suivant:

1. On génère une population d'individus.
2. On les évalue
3. On selectionne les meilleurs individus qui seront les parents de la prochaine génération
4. On les croise pour créer les individus fils
5. On fait muter une partie de ces fils

On peut ensuite répéter ces étapes autant de fois qu'on le souhaite, jusqu'a obtenir un résultat satisfaisant, il faut alors une condition d'arrêt, qui peut être par exemple une valeur de fitness cible ou un nombre limite de générations. On peut modéliser ce déroulement avec un diagramme:

2 Application au problème du voyageur de commerce.

On utilisera donc un algorithme génétique pour une approximation du chemin le plus courant reliant n villes. **L'individu** sera alors un chemin, et sa valeur de **fitness** sera sa longueur.

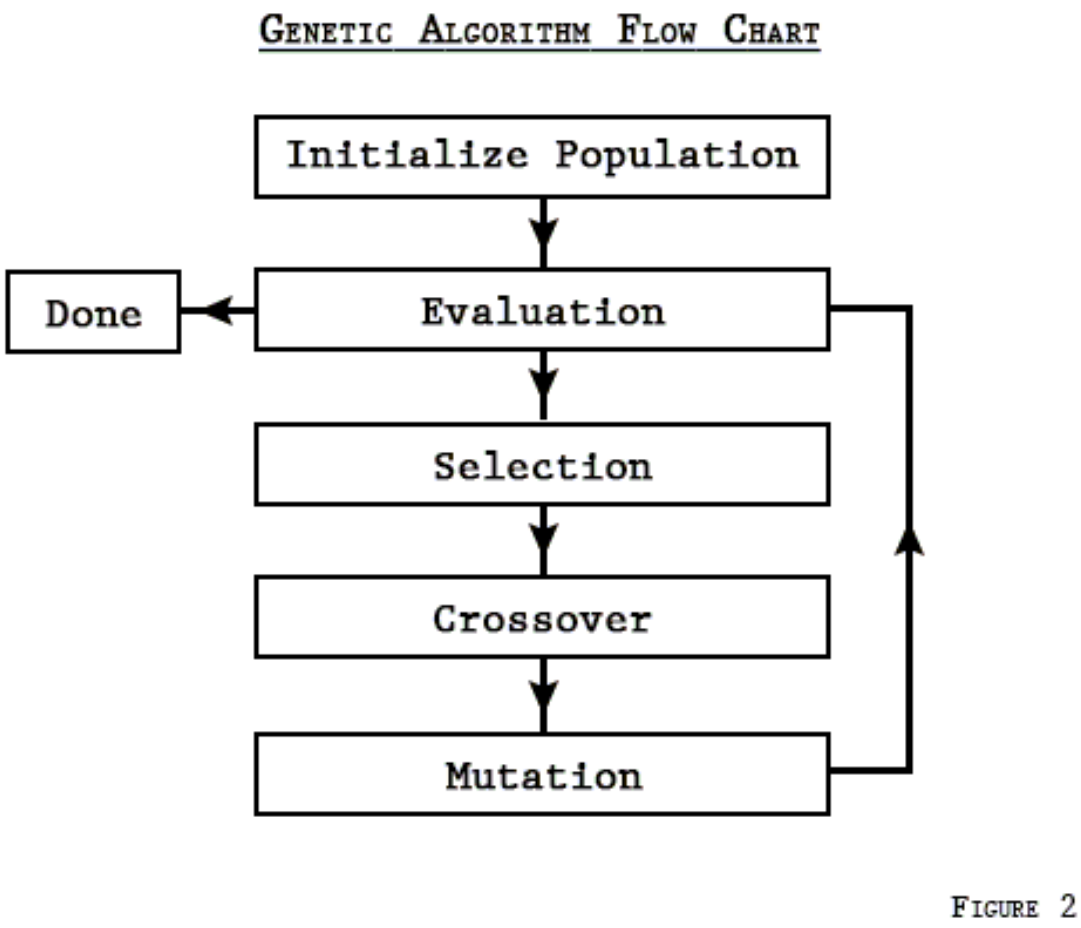

Figure 2: Déroulement d'un algorithme génétique. Source:becominghuman.ai

2.1 Représenter les entités

Une ville est représentée par ses coordonnées X et Y. Un chemin est une liste ordonnée de villes, sa fitness est sa longueur. Il est aussi possible de représenter chaque individu par une chaine binaire, permettant de stocker un très grand nombre d'invidus avec une chaine de longueur relativement petite, par exemple, pour une longueur $l = 100$, on peut représenter $2^{100} = 1.27 \times 10^{31}$ individus. Mais pour des raisons de clarté et de simplicité d'implémentation, les solution à des problèmes de combinatoire utilisent en général directemment des représentations directes des solutions.

La population sera un ensemble de chemin.

L'algorithme est implémenté dans le paradigme orienté objet avec les classes suivantes:

- Ville
- Chemin
- Population
- Client

Où le client est l'application principale permettant de choisir les paramètres et de visualiser l'évolution de la population. Ce qui nous donne l'architecture suivante:

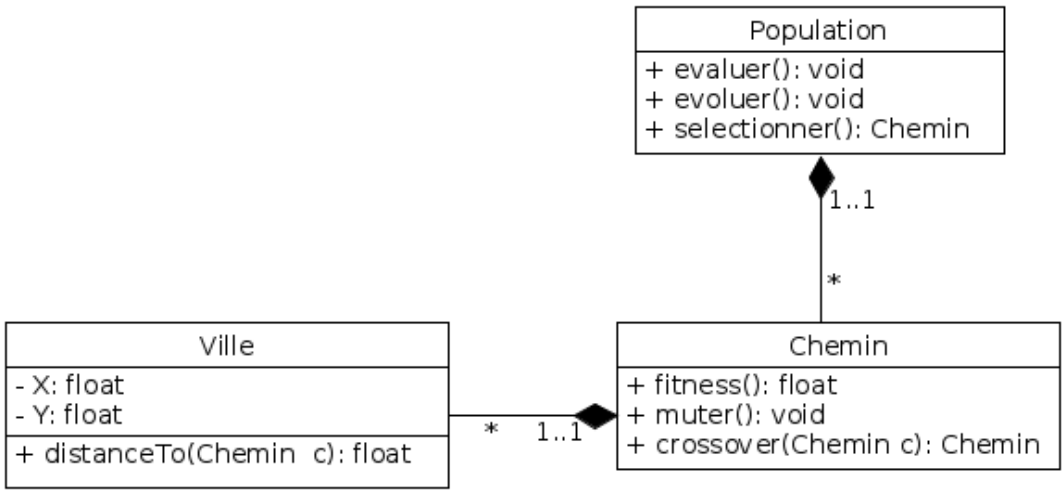

Figure 3: Diagramme de classe UML

2.2 Quelques méthodes

Il existe plusieurs façons de faire chaque opération dans un algorithme génétique.

2.2.1 La selection

La selection est implementée par 2 méthodes: par roulette et par tournoi. La selection par roulette à donner à chaque individu une change d'être selectionné proportionnelle à sa fitness. Quant à la selection par tournoi, elle consiste à prendre une sous partie de la population et d'en prendre le meilleur individu.

2.2.2 La mutation

Il y a aussi 2 méthodes de mutation dans l'algorithme: la méthode *swap* qui échange juste la position de 2 villes dans un chemin, et la méthode *scramble* qui mélange toutes les villes entre 2 points d'un chemin.

2.2.3 L'évolution

La paramètre pouvant varier dans la méthode d'évolution est l'elitisme. Si l'elitisme est activé, on conserve une partie des meilleurs parents dans la génération suivante.

3 Les résultats

Pour un nombre de ville relativement petit, on arrive rapidement à un bon resultat, par exemple, voici le résultat pour 19 villes disposées le long d'un cercle afin que l'on puisse connaitre à l'avance le chemin le plus court.

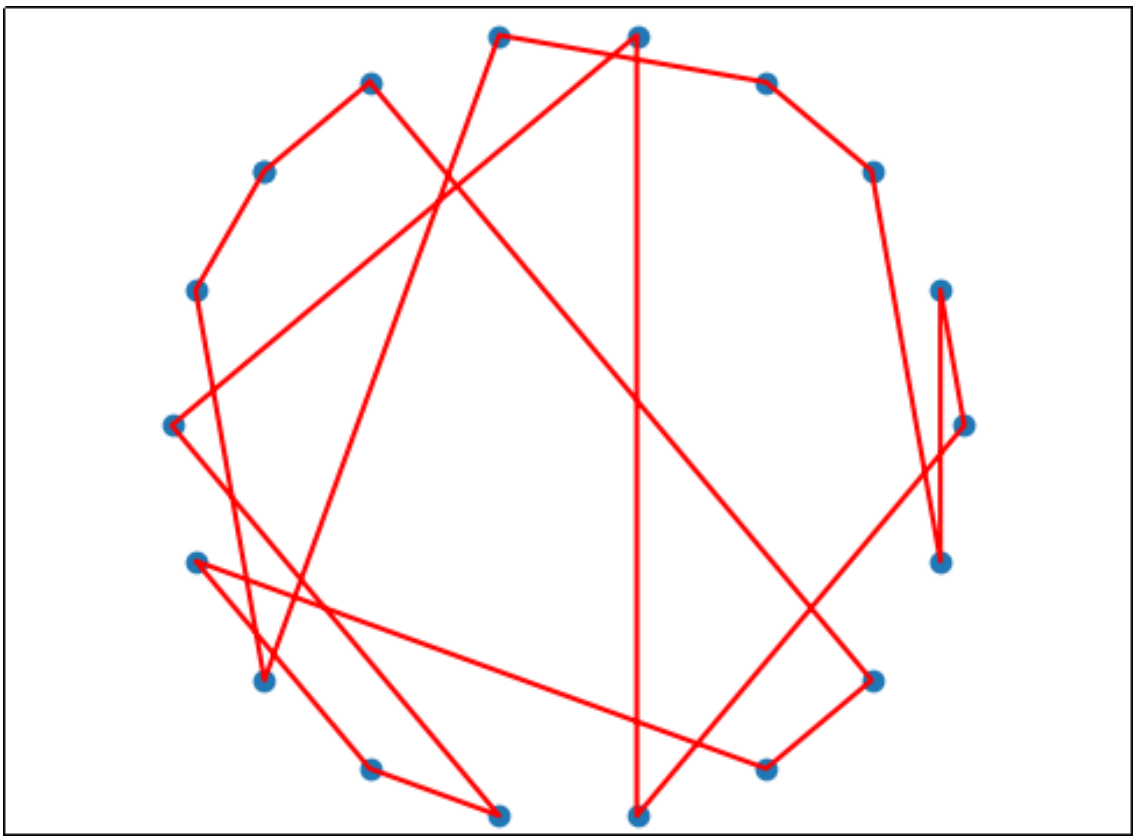

Figure 4: Meilleur chemin reliant 20 villes à la première génération

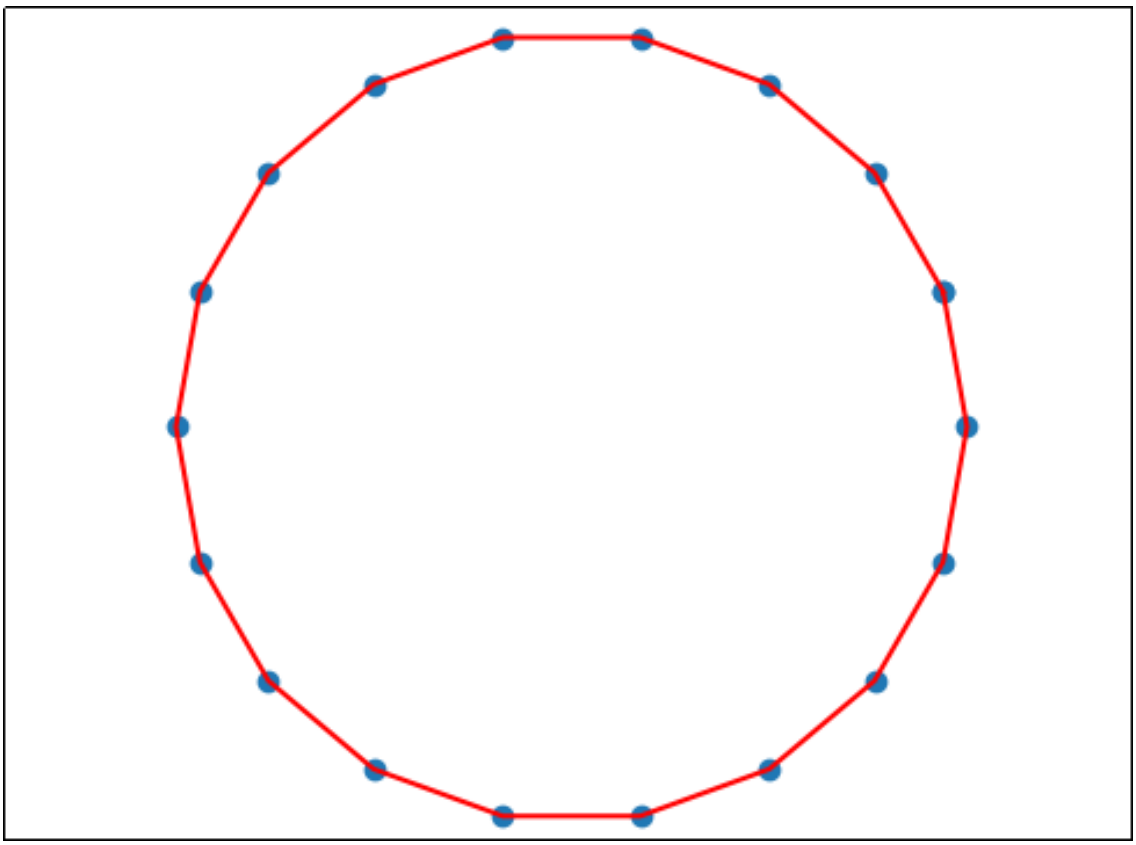

Figure 5: Chemin obtenu après 200 génération, avec une population de 80 individus et un taux de mutation de 3%

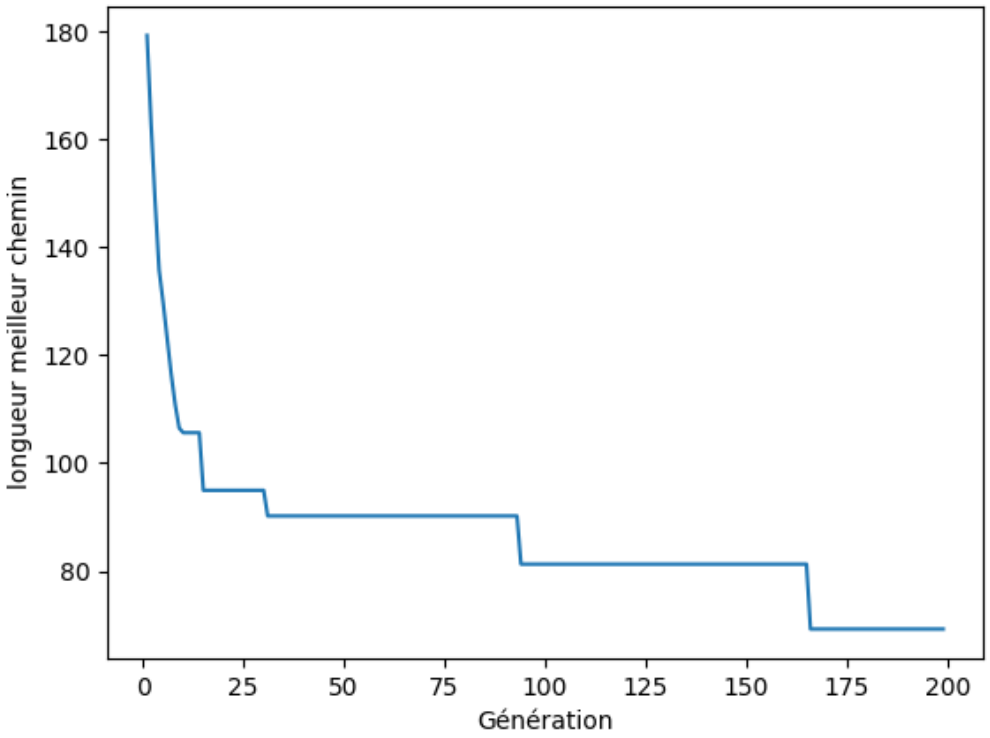

Figure 6: Evolution de la longueur du meilleur chemin en fonction des générations