

Algorithmes Évolutionnaires — Master 2 MIAGE IA²

Travaux dirigés N^o 1 : *MaxOne*

Andrea G. B. Tettamanzi
Université côte d'Azur
andrea.tettamanzi@univ-cotedazur.fr

Année universitaire 2020/2021

Résumé

On va coder en Python (ou un autre langage de programmation de votre choix) un algorithme génétique simple pour résoudre le problème dit « *MaxOne* ».

1 Introduction

Le problème dit « *MaxOne* » est un problème jouet, qui consiste à maximiser le nombre de « 1 » dans une chaîne de chiffres binaires de longueur n . Bien évidemment nous connaissons la solution à l'avance (c'est la chaîne 111...1), mais nous souhaitons que l'algorithme génétique la découvre tout seul, en ayant comme seul indice une fonction de *fitness* qui renvoie le nombre de « 1 » dans une solution candidate.

2 Consignes

1. Prédisez la squelette de votre algorithme génétique en suivant ce modèle :

```
#!/usr/bin/env python3

import random

MAXGEN = 100 # Nombre maximum de générations
POPSIZE = 100 # La taille de la population
N = 20 # la longueur d'un génome
PMUT = 0.01 # probabilité d'une mutation
PX = 0.6 # probabilité d'une recombinaison

random.seed()
pop = []
for i in range(POPSIZE):
    # TO DO (initialisation)

for g in range(MAXGEN):
    pop = selection(pop)
    for ind in pop:
        mutation(ind)
    for i in range(0,POPSIZE,2):
        if random.random() < PX:
            crossover(pop[i],pop[i + 1])
```

2. Codez la fonction qui réalise l'opérateur de mutation, qui prend un individu en entrée et inverse chacun de ses « gènes » avec une probabilité *PMUT*.
3. Codez la fonction qui réalise l'opérateur de croisement, qui prend deux individus en entrée, choisit un point de croisement au hasard et échange les deux sous-chaînes à sa droite.
4. Codez la fonction de *fitness*, qui prend un individu en entrée et renvoie le nombre de « 1 » qu'il contient.
5. Codez la fonction qui réalise l'opérateur de sélection proportionnelle à la *fitness* (suggestion : simulez une roulette comme expliqué en cours).
6. Instrumentez votre code avec des `print` à des endroits stratégiques pour pouvoir vérifier qu'il marche correctement et suivre l'évolution. Par exemple, affichez la *fitness* cumulative à chaque génération, signalez si la solution a été trouvée, affichez la population, etc.
7. S'il vous reste du temps, jouez avec les paramètres pour étudier leur influence sur le comportement de l'algorithme.

Rendez votre code et vos observations dans un archive zippé par courriel.