Solving an optimization problem with a genetic algorithm

Prerequiste

- Python 3
- Matplotlib (optional)

Explanations

In Darwin's evolution theory, a population of individuals follows natural selection : the surviving subjects of a population are the most adaptable to their environment. They are able to reproduce to conceive the next generation of individuals of the population which will be stronger due to a better genetic material. A genetic algorithm uses this idea by following a pattern in three steps for one generation of a population :

- The selection phase : the strongest individuals of the population remain. A **fitness function** needs to be computed to be able to find the "fittest" subjects. Only a fraction of them is selected (**selection rate**). This simulates natural selection.
- The crossover phase : two individuals from the previous step are chosen to conceive one child. Their genetic material is mixed according to a **crossover function**. This simulates reproduction.
- The mutation phase : the genetic material of each child of the previous step can be altered through a **mutation rate**. This simulates gene mutation of DNA.

This process repeats itself over a significant amount of iterations, each one representing a new generation. After a certain amount of generations, one can hope having an ultimate population composed of environment-resistant individuals. The algorithm stops when a **criteria** is met.

How to proceed

The goal of the exercice is to use a genetic algorithm to find the content of a target sentence.

- The target sentence is "I use a genetic algorithm to solve an optimization problem"
- The genes (genetical material) are the characters composing the sentence.
- The fitness function returns the ratio of the number of the right characters at the right place in the sentence and the number of characters in the sentence.
- The selection rate defines the n % of the fittest individuals that are kept for reproduction.

- The individuals reproduce by pair. The crossover function is designed such that the offspring get one part of the first parent's genes and one part of the second parent's genes.
- The mutation fonction randomly alters one gene (or character) of the child with a defined probability (mutation rate)

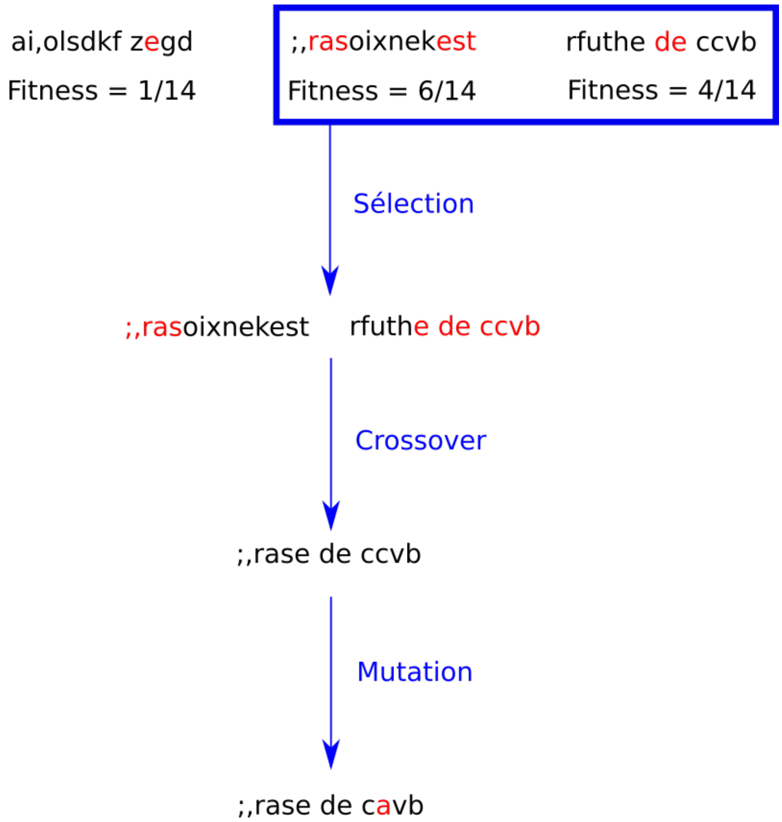


Figure 1: Example of the genetic algorithm's process

For instance, here is some outputs of the algorithm trying to guess the target sentence with a population of 500, a selection rate of 0.5 (50% of the fittest individuals are kept) and a mutation rate of 10% after 583 generations (only a few seconds running) :

```
I\\9.uga >enyTFc aIZoyik0m~xo/8oB|eaan+g;fimizaq*oxzp\\Hb."\r. |     Fitness: 0.423728813559322 | Generation: 22
I HseUa >enetvc algo<ithm7to soHveaan+gifimizat*oxzp\\Hb."m. |     Fitness: 0.6779661016949152 | Generation: 52
I Hse<a genetvc algoyithm7to soHve~an+gitimizat*oxzp\\Hbl"m. |     Fitness: 0.7288135593220338 | Generation: 89
I Hsega genetic algorithm to soHve an oitimizat*o, p\\zbl"m. |     Fitness: 0.847457627118644 | Generation: 133
I HseVa genetic algorithm to soHve an oitimizat*o8 probl"m. |     Fitness: 0.8813559322033898 | Generation: 184
I Hse a genetic algorithm to solve an optimization probl"m. |     Fitness: 0.9661016949152542 | Generation: 264
I use a genetic algorithm to solve an optimization problem. |     Fitness: 1.0 | Generation: 583
```

Experiment with the algorithm for various selection rate, mutation rate and population size.

Going further (optional)

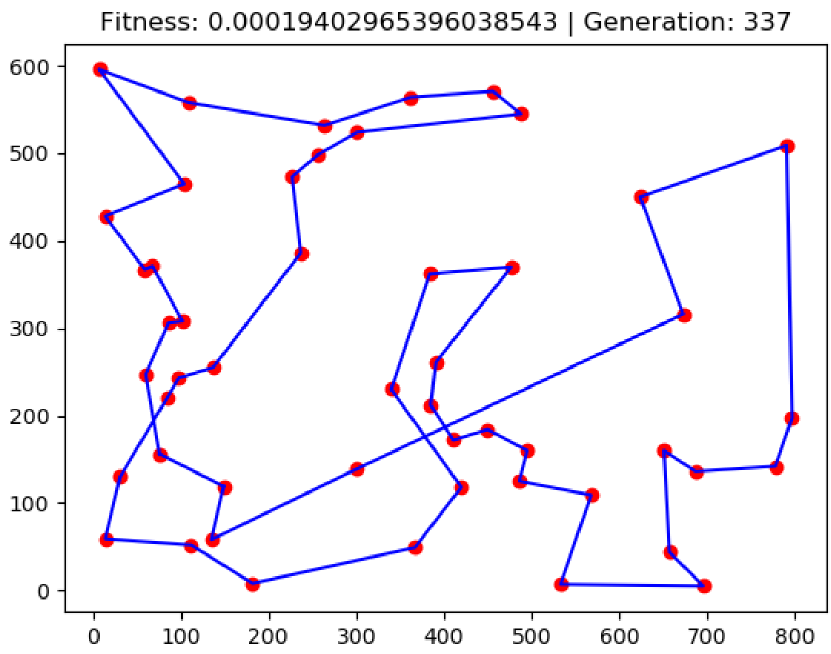
Let's try to solve another problem. Be a set of cities at various distance from each others. A postman needs to visit each city as fast as possible. With the ecological crisis and the global warning, the postmain wants to **find the shortest path to visit each city only once.**

The problem is translated as follows :

- A population is a set of travelling paths through n cities. A individual is a path through n cities. A gene is the city's id. For instance, for 10 cities, an individual (path) could be 5-7-1-3-4-8-2-0-6-9 and a population is a set of paths with different permutations.
- The fitness is the total distance of the path.
- The mutation function switchs two cities (genes) of a path (individual).
- The reproduction function is designed such that an offspring gets in place a subset of the first parent's path. Then, each city of the second parent's path is added to the child's path without duplicate :
  Parent 1 : 738649125
  Parent 2 : 123456789
  Offspring : 128649357

Since the solution here is unknown, a stopping criteria needs to be set. A way to check your algorithm is to scatter cities in circle. The shortest path should draw the circle perimeter.

For instance, here is a result with a population of 500, a selection rate of 0.5 and a mutation rate of 10% after 337 generations (only a few seconds running) :



Experiment with the algorithm for various selection rate, mutation rate, population size, stopping criteria and number of cities per path.