

IN104: Projet Informatique Genetic Algorithms.

Natalia Díaz Rodríguez. ENSTA ParisTech

April 1, 2019

1 Description of the project: Genetic algorithms

This is one of the projects proposed for IN104 course Projet Informatique¹. GA Project site here².

Genetic algorithms (GA) take inspiration from two concepts of the theory of evolution: the definition of a selection process and the use of random mutations to obtain a new set of solutions from solutions previously envisaged. The project aims to implement a genetic algorithm and use it to determine the largest subset of a set of relative integers such that the sum of the elements of this subset is 0. The project will be implemented in Python 3.

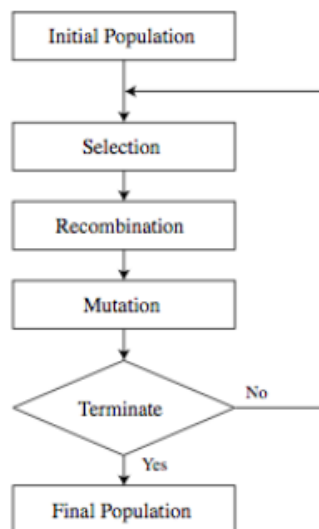


Figure 1: Example of genetic algorithms population evolution cycle

2 Evaluation

The components of the final grade will be aggregated to provide a final mark according to the ENSTA scale.

- **25% Source code:** features, tests, documentation, etc. Source code must be documented, the report including a link to the repository must be in a .zip file. The repository will contain the sources,

¹<https://synapses.ensta-paristech.fr/catalogue/ue/16/in104-projet-informatique> and Moodle: <https://csaclay-sandbox.mrooms.net/course/view.php?id=1158§ion=2>

²https://github.com/NataliaDiaz/IN_104-Projet-Informatique

as well as a plain text file README.md which indicates the actual operational features and limitations. Python 3 code should compile, the teaching assistants are not supposed to make significant corrections for it to compile: a code with few features, but that compiles and does not crash will be preferred to a more complete code but which is not directly operational!

- **25% Defense** (7-10 minutes presentation, 3-5 minutes questions): The formal quality of the presentation will be an important element. The defense needs to include a demonstration on the basis of the source code, an analysis of the difficulties encountered and implemented solutions. It will not include a presentation of the problem or the method of resolution that the teaching assistant obviously knows already well. The defense is open to everyone (subject to the acceptance of the pair that will present), the chronological order of defense will be given by the list of each group.
- **20% Practice Analysis.** In the report you will analyze and criticize the progress of your project, and its success and failure factors. This evaluation component also includes the oral treatment of these questions during the defense.
- **30% Continuous Progress** of the practical work (mid-term evaluation) during the practical lab sessions (based on git commits thorough the whole course).

3 Reporting and deadline

You will return **ONLY** a **SINGLE pdf** (max. 5 pages) report to Natalia Díaz Rodríguez (natalia.diaz (at) ensta (dot) fr). The report must include a link to the git repository named IN104_NameA_SurnameA_NameB_SurnameB (for all team members) with the project code inside a folder called GA. Please do not send extra files by email (these should be all in the repository link to include in the pdf). The report needs to be submitted, at the latest, 5 days before the defense day (happening the last day of the course). Late submissions will be penalized one point per day.

For this particular project we will evaluate the implementation of the methods *get_fitness()*, *select()*, *mutate()*, *crossover()*, plots based on the population size and the demo of the code running and changing diverse parameters. Implementation using a graphical interface library such as *tkinker* is optional. Further experiments evaluating the population evolution dynamics, parameter analysis plots and performance will be more valuable. The report should include insightful plots.

4 Useful links

- GIT:
 - Lecture 0 Intro to Git ³
 - Antonin Raffin tutorials - Intro to Git: <http://slides.com/antoninraffin/git> and Git intermediate: <http://slides.com/antoninraffin/git-intermediate>
 - <http://users.humboldt.edu/smtuttle/sl2cis492/492guide-to-git.pdf>
 - <https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf>
 - <https://github.com/git-tips/tips#everyday-git-in-twenty-commands-or-so>
 - <https://tutorialzine.com/2017/11/10-useful-git-tips>
- Install Python libraries and Master Python: http://musicinformationretrieval.com/python_basics.html
- Python Numpy <http://cs231n.github.io/python-numpy-tutorial/> and IPython tutorials <http://cs231n.github.io/ipython-tutorial/>

³https://github.com/NataliaDiaz/IN_104-Projet-Informatique

- Iterate fast installing Jupyter notebooks <http://jupyter.org/install> and get good at IPython: http://musicinformationretrieval.com/get_good_at_ipython.html
- The quartet of NumPy, SciPy, Matplotlib, and IPython is a popular combination in the Python world. Numpy Basics: http://musicinformationretrieval.com/numpy_basics.html
- Numpy Tutorial: http://scipy.github.io/old-wiki/pages/Tentative_NumPy_Tutorial