

# Cours scientifique - IN104: Projet informatique

Natalia Díaz Rodríguez, Vladimir Paun

2 avril 2018

## 1 Description of the project : Genetic algorithms

This is one of the projects proposed for IN104 course Projet Informatique <sup>1</sup>.

Genetic algorithms 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. Support slides on GA are in the website of the project <sup>2</sup>.

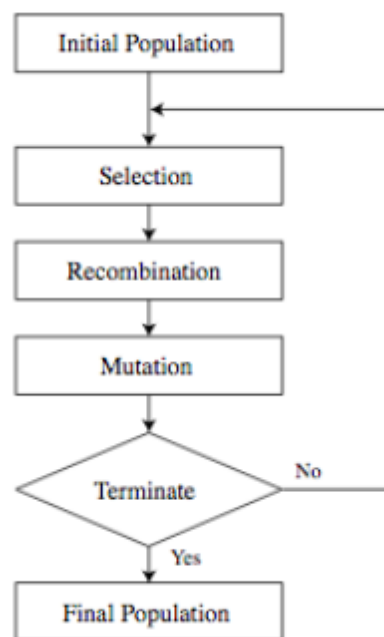


FIGURE 1 – Example of genetic algorithms population evolution cycle

## 2 Evaluation

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

1. <http://perso.ensta-paristech.fr/~paun/ENSTA/IN104/introduction.pdf>  
2. <https://perso.ensta-paristech.fr/~tbernard/Ens/IN104ProgDet.html>

<https://synapses.ensta-paristech.fr/catalogue/ue/16/in104-projet-informatique>

- 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, as well as a plain text file README.md which indicates the actual operational features and limitations. Python 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** (10 minutes presentation, 10 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** -Max 5 pages : You will return a critical report before the defense in which you analyze and criticize the progress of your project and its success and failure factors. This evaluation component also includes the oral treatment of this question during the defense.
- 30% **Progress** of practical work during the practical lab sessions (based on git commits).

For this particular project we will evaluate the implementation of key methods such as *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 tkinter is not needed ; instead, if there is extra time, extra points will be given for implementing features from neuroevolution (see blog post <sup>3</sup>). Further experiments evaluating the population evolution dynamics and performance will be as well valued.

### 3 Getting started

Test driven development is strongly recommended as a development cycle. Some of the files included in the project repository <sup>4</sup> :

- Presentation of the project : [IN104\\_slides\\_PAUN.pdf](#)
- Detailed subject : [IN104\\_AlgoGen\\_sujet\\_PAUN.pdf](#)
- Expected deliverable format : [AlgoGen\\_expected\\_format.pdf](#)
- Given test input data files : [inputSet.zip](#)
- Code for input set generation : [genEnsDep.py.zip](#)
- Both [gitlab.ensta.fr](#) and [github.com](#) can be used. However, the latter is recommended (in which case, the project created needs to strictly be set as private). If you can't create a private repo, follow instructions in Appendix.

### 4 Reporting and deadlines

A zip file containing the pdf report should be sent to Natalia Díaz Rodríguez (natalia.diaz (at) ensta (dot) fr). The report must include a link to the github (preferred) or gitlab repository with the code and must be called surname-first-team-member\_surname-second-team-member(.zip and .pdf, respectively). Deadlines are in the course website <sup>5</sup>.

### 5 Useful links

- Test Driven Development and Git versioning : <http://perso.ensta-paristech.fr/~paun/ENSTA/M0101/>
- GIT :

---

3. <https://eng.uber.com/deep-neuroevolution/>

4. [https://github.com/NataliaDiaz/IN\\_104-Projet-Informatique](https://github.com/NataliaDiaz/IN_104-Projet-Informatique)

5. <http://perso.ensta-paristech.fr/~tbernard/Ens/IN104ProgDet.html>

- 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](http://musicinformationretrieval.com/python_basics.html)
- Python Numpy <http://cs231n.github.io/python-numpy-tutorial/> and IPython tutorials <http://cs231n.github.io/ipython-tutorial/>
- Iterate fast installing Jupyter notebooks <http://jupyter.org/install> and get good at IPython : [http://musicinformationretrieval.com/get\\_good\\_at\\_ipython.html](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](http://musicinformationretrieval.com/numpy_basics.html), Numpy Tutorial : [http://scipy.github.io/old-wiki/pages/Tentative\\_NumPy\\_Tutorial](http://scipy.github.io/old-wiki/pages/Tentative_NumPy_Tutorial)
- Intro to genetic algorithms with an industry case (data-driven fashion design) : <https://multithreaded.stitchfix.com/blog/2016/07/14/data-driven-fashion-design/>

## 6 Appendix

### 6.1 Recap from pre-requirement lecture MO101 : Intro to Test Driven Development (TDD) and Git versioning control

This TD on TDD+GIT must follow the instructions<sup>6</sup>. It will be evaluated on the work done during the class (the game on add, commit, push and pull) and also, the work done until next following 7 days- i.e., until Tuesday 3rd April 23.59h. This is in order to ensure you have had time to understand the concepts behind and are in shape to put it in practice during the IN104 projet informatique.

Remember to add your ssh-key to your gitlab/github account (each of the team members) using [1]. There are some subtleties found to work easier in Github versus Gitlab ; therefore, github is recommended :

1. Create a new account as in [1] and a repository
2. Add your ssh-key as in [2].
3. Provide master access to your team member and me (username 'diaz') via the *Team* option settings of the project.

For those using gitlab, access denied problems should be solved by giving your partner (and your teaching assistant), *master* access instead of *developer* access.

An editor easier to use than *vi* is *Atom* (open source, recommended) [3] or *sublime* [4].

#### 6.1.1 Test Driven Development

*unittest* has some internal magic that :

- discovers all classes inheriting from *unittest.TestCase*
- then runs its *setUp* function
- then runs all methods that are named *test\_\**
- then prints a summary (passed vs failed tests)

And more. We've already seen other methods to do unitary tests (using the `__name__ == __main__` trick in the program containing some *\_test\_func* ). *unittest* is just more practical when there is a large set of tests as here. See all *unittest* frameworks online<sup>7</sup>.

6. <https://gitlab.ensta.fr/paun/mo101-agile-tdt/tree/master>

7. <http://docs.python.org/3/library/unittest.html>

### 6.1.2 Installing the testing frameworks

```
conda install -c anaconda pytest
```

Additionally, although it should not be needed :

```
conda install -c anaconda unittest2
```

### 6.1.3 Running tests

Option A) (preferred) To run a given test :

```
python -m unittest test_name
```

e.g. :

```
python -m unittest tests_binary
```

if tests\_binary.py is in the main binary repository

Option B) Run all tests in tests folder (-v for verbose mode) :

```
py.test -v
```

### 6.1.4 Links

[1] Github (alternative to gitlab, easier) : <sup>8</sup> -> request discount for student -> individual, using your ensta email. Option b : create gitlab account <sup>9</sup>.

[2] Create and add your ssh key to gitlab <sup>10</sup> or github <sup>11</sup>

[3] Atom editor <sup>12</sup>

[4] Sublime editor <sup>13</sup>

[5] Python overloading operator API : <https://docs.python.org/3.4/library/operator.html>

### 6.1.5 Troubleshooting

— Q : Using gitlab :

```
bash: warning: setlocale: LC_ALL: cannot change locale (en_US.UTF-8)
fatal: protocol error: bad line length character: No s
```

A : You can try setting the locale language :

```
export LC_ALL=fr_FR.UTF-8
```

you can also try run the command *ssh-add* (this commands adds the ssh key to the ssh agent).

---

8. <https://education.github.com/>

9. <http://gitlab.ensta.fr/>

10. <https://gitlab.ensta.fr/help/ssh/README>

11. <https://help.github.com/articles/adding-a-new-ssh-key-to-your-github-account/>

12. <https://atom.io>

13. <https://www.sublimetext.com/3>