Computational Intelligence (CI-MAI) EAs exercise 2

Lluís Belanche

4 Dec 2023

Optimizing a difficult function with Evolution Strategies

The application of formal tools to the minimization/maximization of multivariate functions is a long-standing problem in mathematics. It is not possible to compile the numerous approaches existing in the literature, mostly based on computing first order (the gradient) and/or second order (the Hessian) derivatives to get approximate information of the local behaviour of the function to be optimized.

Literally hundreds of works have been published in the matter, with different approaches and levels of sophistication. The best approach is to consult good specialized books on the matter, like those written by Dimitri P. Bertsekas (*Convex Optimization Theory*, *Convex Analysis and Optimization*)[1] or by Boyd and Vandenberghe (*Convex Optimization*)[2].

Moreover, constraints on the function can also be added to the fitness function. In mathematical optimization, this leads to Linear Programming, Quadratic Programming, and the like.

You are required to select a problem (function to be minimized/maximized, root finding, etc).

Ideally, you may bring your own problem, but there are many (mostly artificial) problems designed as a benchmark suite; a couple of starting webs are:

- https://www.sfu.ca/~ssurjano/optimization.html
- https://en.wikipedia.org/wiki/Test_functions_for_optimization

The exercise consists in creating or choosing a problem like the ones described above and attack it using an Evolution Strategies (ESs). The requirement for this practical work is that the search space must be a (preferably compact) subset of R^n.

The use of synthetic data allows to be in control of:

- the dimension of the problem n
- the way you represent mutation in the ES (as seen in class)
- the possible present of noise (controlled by a single hyperparameter)
- the problem hardness (controlled by a single hyperparameter) [3]

All other conditions, settings, hyperparameters, etc, are left to your decision. You can also choose any other (evolutionary or non evolutionary) method that you want to test in comparison to the ES. The problem hardness should not be extremely difficult or trivial, otherwise the study is pointless.

What to report in the comparison? An obvious choice is to set a predefined performance and then report the (average):

- execution time needed to achieve a given performance
- number of generations needed to achieve a given performance

[1] Course notes (that cover the same material as the textbook) freely available on his website: http://web.mit.edu/dimitrib/www/home.html

[2] Freely available at https://web.stanford.edu/ boyd/cvxbook/bv_cvxbook.pdf

[3] It is desirable that you know the theoretically optimal solution.

- number of fitness function calls needed to achieve a given performance
- Fraction of times the algorithm reaches a given performance

  … as a function of population size, n or problem hardness. Alternatively, one can let the algorithm run for a predefined number of generations and study all the other quantities. The study is quite flexible and left to your criterion.

If you use R, a good tool is to use *Rmarkdown* to produce the document, integrating LaTeX code and R code. If you do, have a look at

https://bookdown.org/yihui/rmarkdown/

There are other similar tools for other languages.

Important information:

- **Write a brief pdf document (7 sheets maximum, including everything) that describes only the relevant information (problem setup, previous work, work done by you, discussion and conclusions)**
- **Delivery date: no later than December 28, 2023, via the Racó**
- **To be done in groups of 2 students**
- **Please include all names in the final document and upload only one document per group**
- **Add a plain text file named "README.txt" with complete instructions about how to obtain your results**
- **If you use ChatGPT (or another similar tool) in the document, indicate it every time you do. We want to evaluate your work, not someone else's!**
- **There is no obligation to use any particular programming language; but I ask you to limit the choice to R, Julia, MATLAB and python. Choose any library that you find adequate as a complement, provided you cite them**