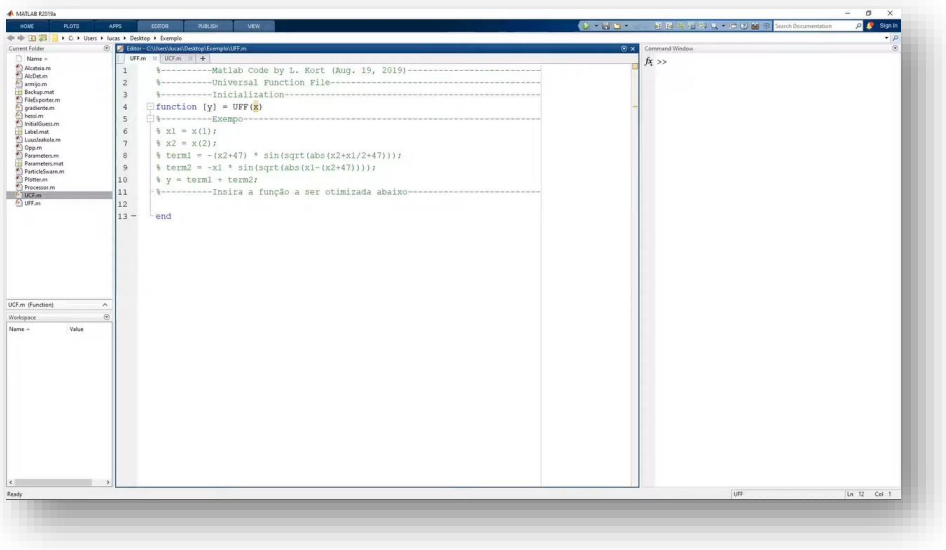
OPP

Optimization tool

Summary

Letter to the user

Opp is a simple optimization tool. Created by me in 2018, Opp came up as a tool to perform numerous optimization processes quickly and simplify the comparison between different optimization methods.



Written for MatLab, Opp, in its first versions, works as a tree of .m files that needed to be modified. For ease of use, these files had predefined editing zones.

The original source code is still available for download on GitHub here.

In 2021, Opp entered a new phase. The project has been resumed and entirely rewritten for Python 3.x. A graphical interface was added, the tool gained more optimization methods, and with the new structure, Opp is now faster and even easier to use than ever before.

Opp's goal is to democratize optimization. It was thought to be extremely easy to use and versatile. Projected to be effective for all audiences, Opp does not require prior knowledge of optimization, and you will find in this text all the information you will need.

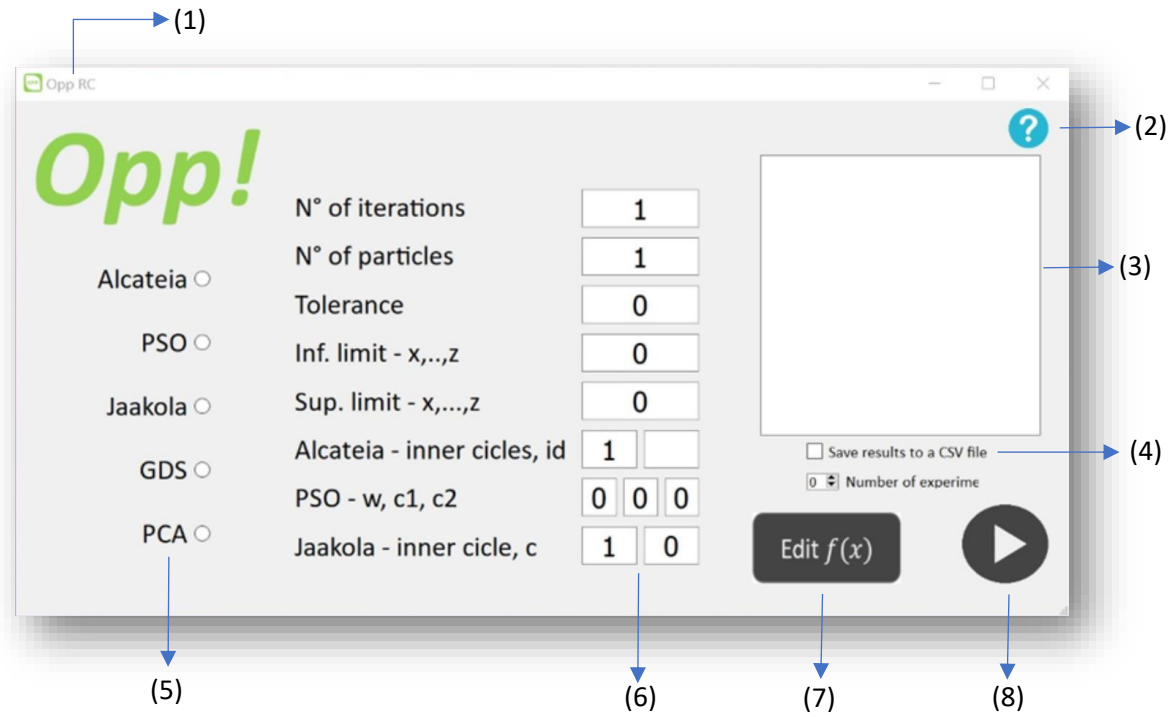
The version you are using now has a long history and required a lot of work and dedication, not only from me but from several other people who contributed directly and indirectly to this project. I hope you like it and, even more importantly, that it's useful. Several papers were written and published with the support of Opp. The tool will be updated frequently and gaining resources to continue being useful, whether in academic research, commercial, or any other optimizations of interest.

Good optimization!

Lucas Kort

Opp interface

“Extremely versatile and easy to use. Opp's goal is to democratize optimization. With an intuitive interface, several guidance messages programmed into the console, and easy method configuration, you don't need the experience to get good results in a short time.”



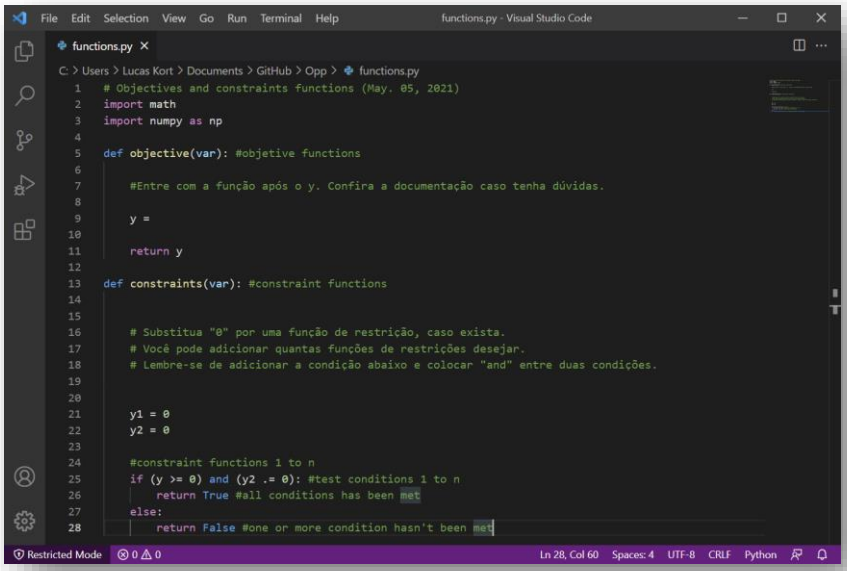
(1) Application's version
(2) Help button
(3) Application console
(4) Export result in .csv file
(5) Numeric Method's Tree
(6) Parameters' setup
(7) Edit function button
(8) Run button

Quick start

Opp was designed to be easy to use. Based on the Poka-yoke philosophy, your options are limited and intuitive.

First step:

Press the ***Edit Functions button*** (7). The objective and restriction function editing window will open.



In this window, you should enter your objective and restrictive functions (if any). Note that Opp is built in Python and needs the Math library for advanced commands. In other words, before trigonometric functions, such as sine and cosine, logarithms and exponents, it is necessary to place the math code before the function. A complete table of all functions can be found at the end of this document. Another important detail is that Python uses ** for powers.

Second step:

Select within the ***Numeric Methods' tree*** (5) which optimization method(s) you want to use. Opp supports multiple methods to be executed at the same time.

Third step:

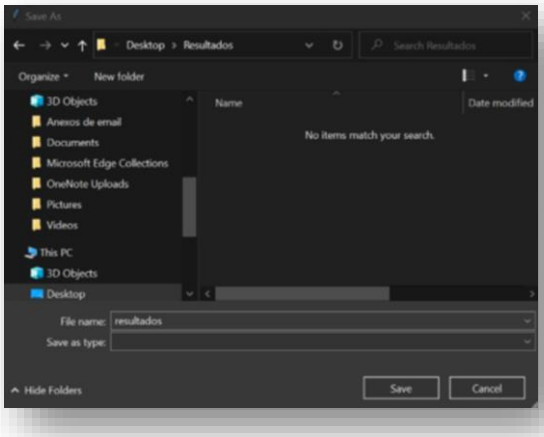
In the Optimization ***Parameters' setup*** (6), enter the basic optimization settings and method-specific parameters. If you don't have programming experience, try running the method first with the default configuration and modify the method with small increments/decrements with each execution, while following the evolution of the results.

Fourth step:

When ready, click ***Run*** (8) to run the numerical methods, with the settings entered.

Fifth step:

If you have chosen the option ***Export result in .csv file*** (4), a window will open at the end of each execution, and you will have to enter a save location. The file is compatible with the main productivity programs, including and in particular Microsoft Excel.



Remember

- Pay attention to the ***console*** (3). It will provide important information about the process and help you set up methods and fix common errors.
- Make it a habit to check if your current ***version*** (1) is the last one. For this, visit https://github.com/LukeKort/Opp/releases regularly.
- Whenever you need it, this document is at the ***help button*** (2).
- It is also possible to save the graphics generated at runtime.

Acknowledgments

Opp was only possible thanks to the collaboration of many people. Thanks to the entire GitHub community, Python.org, and the Stack Overflow community, who took all my questions away and made this code better.

Thanks to Anaconda, the platform on which Opp was written on. Thanks to Riverbank Computing for make PyQt available, where the interface of this program was created on. And thanks to the entire Python community. Their work has made coding easier.

Special thanks to Andressa Machado for her support and collaboration