UE304C: Master's Project Work for DSAI

Lhassane IDOUMGHAR

lhassane.idoumghar@uha.fr

DSAI master at UFAZ

Outline

- Presentation of the Project
- Deliverables
- Deadlines

Presentation of the project

Master’s courses

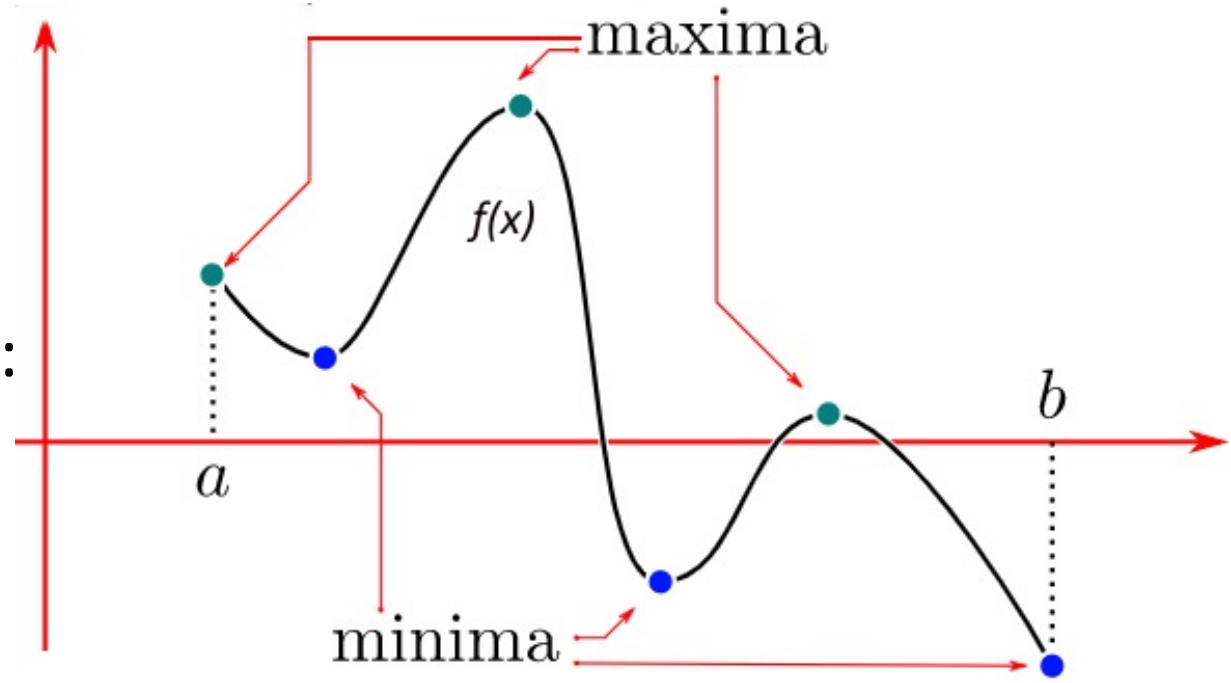
- Master 1
  - MUE107C: Project Work 1 for DSAI Development Project – implementation of several metaheuristics algorithms (DE, PSO, GWO, GA, etc.)
  - MUE202C: Parallel programming – Introduction to CUDA programming.
- Master 2
  - MUE303C: Massively parallel stochastic optimization and artificial evolution.

Goal optimization problem

$$\min_{x \in S \subseteq \mathbb{R}^n} f(x)$$

We seek to find a global solution, such that:

$$x^* = \arg\min f(x) : f(x^*) \leq f(x) \; \forall \; x \in S$$



To resolve this problem, we propose in this Master's project to study a massively parallel metaheuristic based on differential evolution.

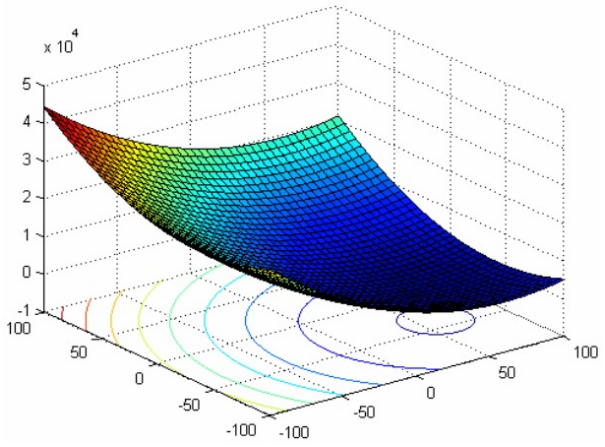
Dimensionality

- optimization can be done on multi-dimensional spaces

2D Shifted Sphere's function

$$\sum_{i=1}^{D} z_i^2 + f\_bias_1 \,,\ \mathbf{z} = \mathbf{x} - \mathbf{o} \,,\ \mathbf{x} = [x_1, x_2, ..., x_D]$$

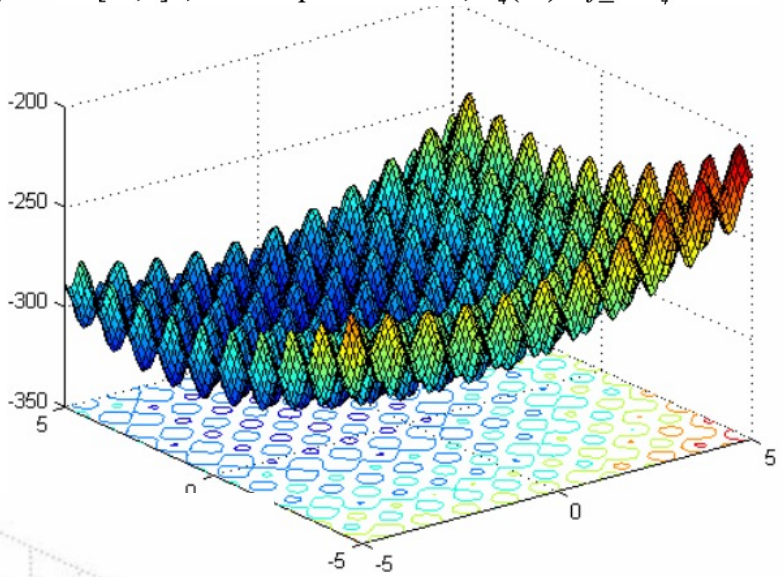
$\mathbf{x} \in [-100,100]^D$, Global optimum: $\mathbf{x}^* = \mathbf{o}$, $F_1(\mathbf{x}^*) = f\_bias_1 = -450$



2D Shifted Rastrigin's function

$$\sum_{i=1}^{D} (z_i^2 - 10\cos(2\pi z_i) + 10) + f\_bias_4,\ \mathbf{z} = \mathbf{x} - \mathbf{o},\ \mathbf{x} = [x_1, x_2, ..., x_D]$$

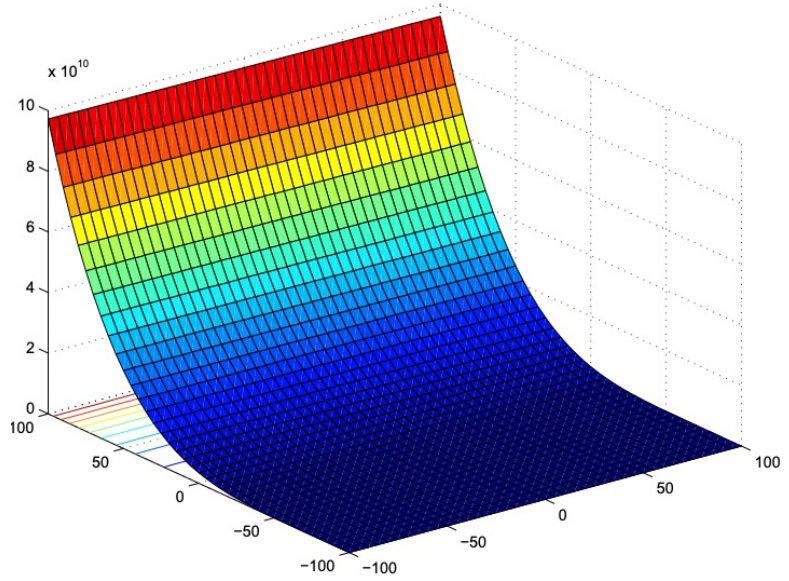
$\mathbf{x} \in [-5,5]^D$, Global optimum $\mathbf{x}^* = \mathbf{o}$, $F_4(\mathbf{x}^*) = f\_bias_4 = -330$

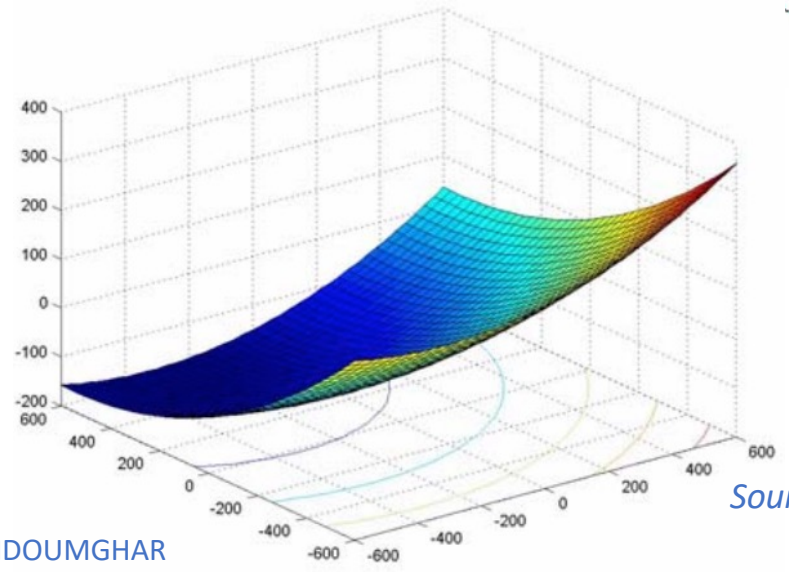


2D Shifted Rosenbrock's function

$$\sum_{i=1}^{D-1} (100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2) + f\_bias_3,\ \mathbf{z} = \mathbf{x} - \mathbf{o} + 1,\ \mathbf{x} = [x_1, x_2, ..., x_D]$$

$\mathbf{x} \in [-100,100]^D$, Global optimum $\mathbf{x}^* = \mathbf{o}$, $F_3(\mathbf{x}^*) = f\_bias_3 = 390$





2D Shifted Griewank's function

$$\sum_{i=1}^{D} \frac{z_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 + f\_bias_5\ ,\ \mathbf{z} = (\mathbf{x} - \mathbf{o}),\ \mathbf{x} = [x_1, x_2, ..., x_D]$$

$\mathbf{x} \in [-600,600]^D$, Global optimum $\mathbf{x}^* = \mathbf{o}$, $F_5(\mathbf{x}^*) = f\_bias_5 = -180$

Source: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.515.821&rep=rep1&type=pdf

Resources

- Several files are provided, where there is:
  - A research paper explaining an example of the Improved CUDA-Based Implementation of Differential Evolution on GPU (An_Improved_CUDA-Based-Implementation_of_Different.pdf).
  - A research article explaining: *A Unified Differential Evolution Algorithm for Global Optimization.* Also, this paper describes the mathematical formulation of the objective functions (DE-BenchmarksFunctions.pdf).
  - Instruction of how to install Cuda and DE under googlecolab.
    - CudaDE.ipynb file to download from colab
    - C/C++/CUDA source code with an implementation of massively parallel version of DE

The project- Instructions

1. Each pair of students will study massively parallel version of DE algorithm code.
2. Read the attached pdf files (*An_Improved_CUDA-Based-Implementation_of_Different.pdf*, *DE-BenchmarksFunctions.pdf*)
3. Implement the mutation strategy described in *DE-BenchmarksFunctions.pdf* (eq. 13)
4. The evaluation of your implementation will be done on the mathematical benchmark functions: Shifted Rastrigin's Function, Shifted Rosenbrock's Function, Shifted Griewank's function, Shifted Sphere's Function. For more details about these functions, please see:
   https://www.sfu.ca/~ssurjano/optimization.html
   http://www.cmap.polytechnique.fr/~nikolaus.hansen/Tech-Report-May-30-05.pdf
   https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.515.821&rep=rep1&type=pdf
5. The user can specify the dimension (Dim) of the studied problem
6. improve the quality of your results (try others crossovers operators, other mechanisms, etc.).

The project

- Finally, you will need to compare your implementation with other algorithms in the literature, following this procedure:
  - Population size: 50, 100, 500, 1000.
  - Dimension of each benchmark: Dim = 10, 50, 100.
  - The algorithm terminates once the maximal number of function evaluations is reached, which is set to 10^4 X Dim, e.g. for a 10D problem, the maximal number of function evaluations is 10^5.
  - Calculate the mean and standard deviation of the global minimum values obtained after 25 runs.
  - To compare your results with other algorithms, you can :
    - Implement sequential version of DE algorithm.
    - You can also use the code that you developed last year.
    - Use results obtained presented in *An_Improved_CUDA-Based-Implementation_of_Different.pdf*
    - *Implement these functions under EASEA Library*.
  - Use the statistical analysis (Wilcoxon and/or Kruskal-Wallis based on your results) described on the following web link: https://machinelearningmastery.com/statistical-hypothesis-tests-in-python-cheat-sheet/)

Deliverables

Final document: Scientifique paper

- The document to be produced will take the form of a scientific article (take inspiration from *An_Improved_CUDA-Based-Implementation_of_Different.pdf*).
- Structure of the paper (word or latex format https://www.ieee.org/conferences/publishing/templates.html):

  Title
  Abstract,
  Keywords,
  1. Introduction,
  2. GPU programming
  3. Differential Evolution
     3.1. Overview of sequential version of DE
     3.2 Description of Massively parallel version of DE (UML Diagrams: sequence diagrams, User Diagram, etc.)
  4. Experimental results
     Setup
     Benchmark functions
     Results: experimental protocol, graphics, tables, etc (presentation of the results obtained by sequential and massively parallel versions of DE (Dim = 10, 50, 100, etc., Population Size = 50, 100, 500, 1000. Different mutation strategies, Running time - curves).
  5. Conclusion

Deadlines

Planning

- Nov 12th: composition of student groups. Please complete the following file: https://uhafr-my.sharepoint.com/:x:/g/personal/lhassane_idoumghar_uha_fr/EZkKdkFIECFEncrcT-rnjzEByAu6uxVBdJEwpxmo89T2Mw?e=Qxr6fu

- Dec 20th and Dec 23th : Each pair of students will work on this project.

- Dec 24th: Each pair of students will have a 30 min time slot to present their work:
  Their algorithms,
  Their specification,
  Their design and results, etc.

- Final deliverables due on Dec 24th at midnight : presentation, paper, code, etc.