MUE107C
Project Work 1 for DSAI Development Project

Cecilia Zanni-Merk

Scientific Coordinator of the DSAI master at UFAZ

Agenda

- Presentation of the Project
- Deliverables
- Deadlines

Presentation of the project

Meta-heuristics

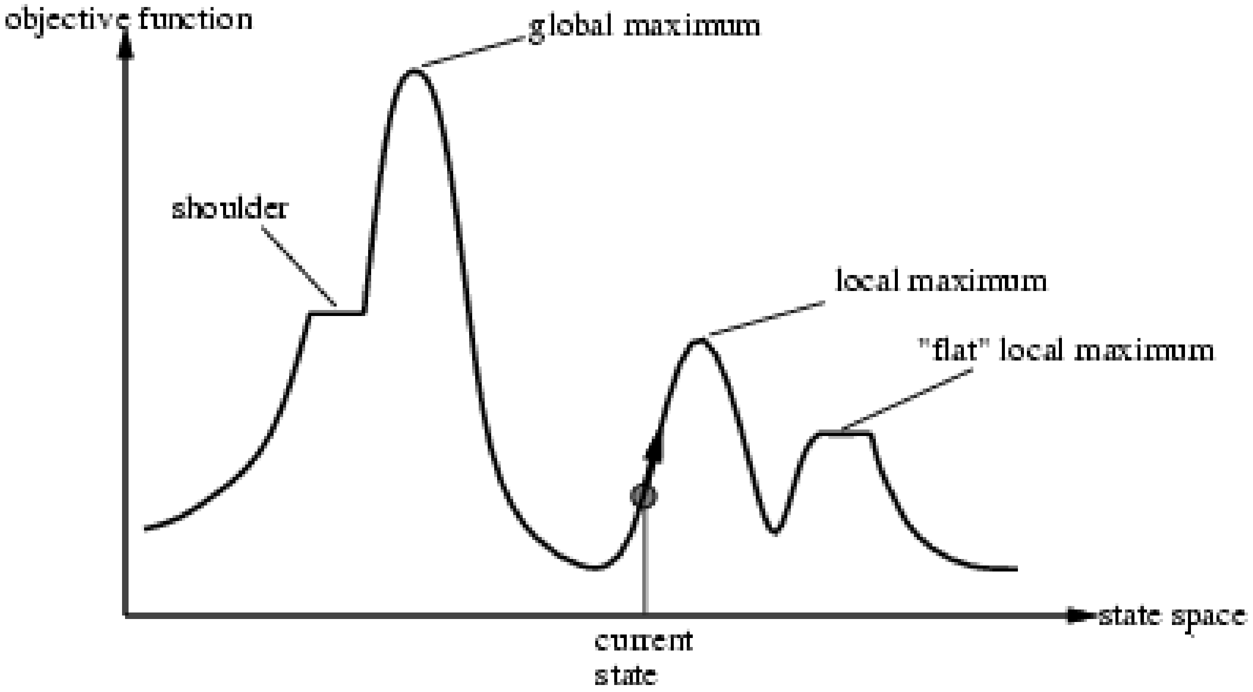
- Metaheuristics are optimisation algorithms aimed at solving difficult optimisation problems for which no more efficient classical method is known.
  - Often coming from the fields of operational research, engineering or artificial intelligence.
- They differ from classical state space search algorithms in that **when a certain state is explored, the algorithm no longer knows anything about the previous states and the path taken up to the current state**.
  - The disappearance of this memory function within these algorithms has strong repercussions on their performance !

Performance

- Where all classical algorithms are often very memory-greedy, local exploration algorithms are much lighter.
- This is the main reason why the industry has adopted this type of algorithm for many optimisation systems.
- The good news ... they are, for the most part, **easily parallelizable**.

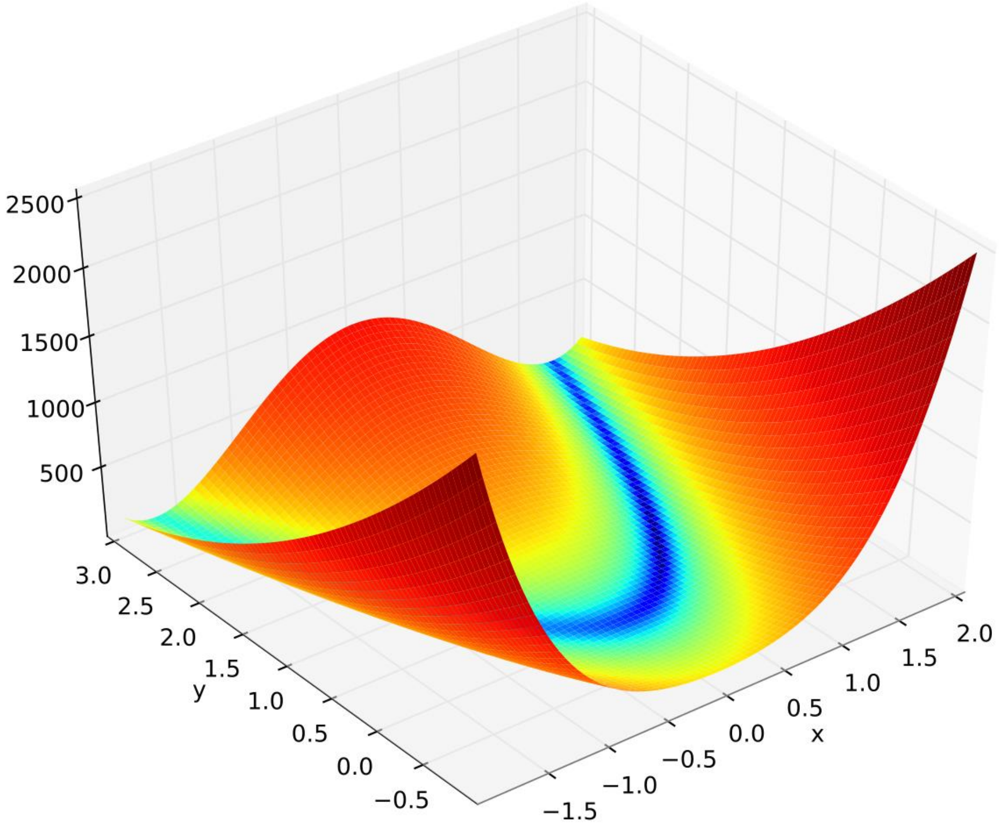
More specifically …

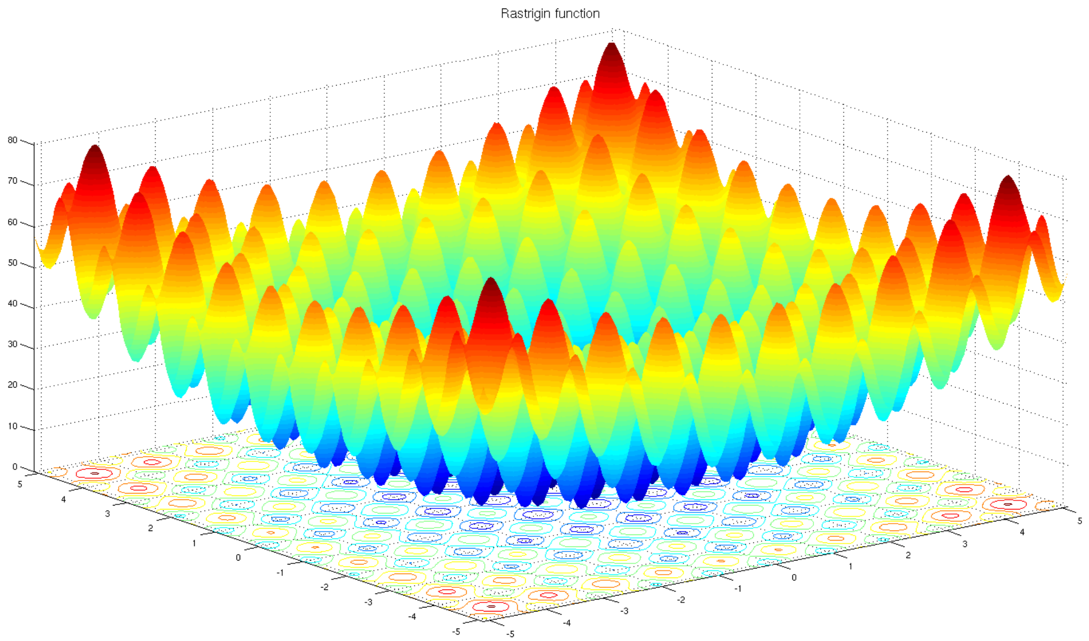
- These are stochastic iterative algorithms that try to find the global optimum of a difficult optimisation problem, without being trapped by local optima.



Dimensionality

- Of course, optimisation can be done on multi-dimensional spaces …





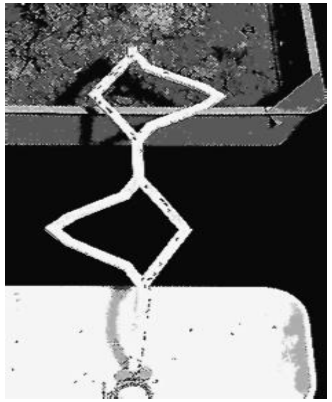
Rosenbrock function
Rastrigin function

Examples …

- Meta-Heuristics are often inspired by physics (**simulated annealing**), evolutionary biology (**genetic algorithms**) or ethology (**ant colony** or **particle swarm optimization** algorithms).

Evolution ?

Ants



Social behavior

Fish schools

Birds flocks

Bee swarms

The project

1. Each pair of students will implement **one** meta-heuristic algorithm, among the following:
    - Backtracking Search Optimization Algorithm (BSA)
    - Differential Evolution Algorithm (DEA)
    - Gravitational Search Algorithm (GSA)
    - Particle Swarm Optimization (PSO)
    - Grey Wolf Optimizer (GWO)
    - Genetic Algorithm (GA)
    - Whale Optimization Algorithm (WOA)
    - Harmony Search Algorithm (HSA)
    - Imperialist Competitive Algorithm (ICA)
    - Bat Optimization Algorithm (BOA)
    - Firefly Optimization algorithm (FOA)

Random Draw now!!!

The project

2. The test on the evaluation of your implementation will be done on the mathematical benchmark functions: ***Ackley Function***, ***Rastrigin Function***, ***Rosenbrock Function***, ***Schwefel Function***
   - For more details about these functions, please see: https://www.sfu.ca/~ssurjano/optimization.html
3. Finally, you will need to compare your implementation with other algorithms in the literature, following this procedure:
   - Population size: 30.
   - Dimension of each benchmark: 30.
   - The arithmetic precision value used in this project is 10-^16 (we suppose that the objective function is equal to Zero if f < 10^-16).
   - The maximum number of times the objective function is evaluated is 2,000,000.
   - Number of runs: 30
   - Calculate the mean and standard deviation of the global minimum values obtained after 30 runs.
   - Compare your results with the results obtained by ***provided*** Matlab code.

Resources

- On Moodle you will find a folder for each algorithm, where there is
  - A research article explaining the algorithm
  - Matlab source code with an implementation of each meta-heuristic algorithm, for you to compare the performance of your implementation with it.

Deliverables

Specification document

- Introduction
  - This is an informal description of the project and its context.
  - It should include, among other things, the following information
    - A list of the main functions,
    - The different users and their characteristics,
    - hardware and software constraints.
  - Detailed requirements
    - It is the contractual party strictly speaking since it formalizes the need
    - It consists of 3 distinct parts:
      - the functional specifications
      - interface specifications
      - operational specifications (performance, safety, ...)
  - These different elements can be based in use case diagrams, a domain model diagram, mock-ups and a navigation diagram in UML, depending on the needs.

Design document

- Three sections
  - Introduction
  - Preliminary design
  - Detailed design
- Introduction
  - This is an informal description of the project and its context.
  - It is often relatively similar to the introduction to a specification document.
  - It should therefore include the following information
    - the list of the main functions,
    - the different users and their characteristics,
    - hardware and software constraints.

Design document

- Preliminary design
  - This step consists in carrying out a macroscopic design, i.e. leading to a breakdown into packages with the external signatures of each package.
  - This step can be based in UML on :
    - A domain model diagram (if not specified) ;
    - System sequence diagrams (if not specified);
    - Mock-ups (if not specified);
    - Navigation activity diagrams (if not specified);Interaction diagrams;
    - Preliminary design class diagram;
    - A breakdown into packages and the external signatures of each package.
    - A deployment diagram

Design document

- Detailed design
  - This step consists in detailing by package the elements constituting them.
  - Concretely, it is mainly a matter of specifying the class attributes and methods of all participating classes and grouping them together in a class diagram. The methods of a package that will be considered as non-trivial will have to be commented and their operation detailed by pseudo-code.

Final report

- Must contain
    - Specifications (Specifications document updated if necessary)
    - Design (Design document updated if necessary),
    - Justified technical choices
    - Possible improvements (not foreseen in the specs)
    - Evaluation of the performance of your implementation of the algorithm on the suggested functions
    - Comparison of the performance of your implementation of the algorithm with the provided Matlab source code

Deadlines

Planning

- ***Week 11*** (Nov 24th and Nov 27th) : Each pair of students will have a 15 / 20 min time slot to present
  - Their algorithms, informally
  - Their specification
  - Their design
- ***Deliverables due on Nov 27th at midnight*** : presentation, specification document, design document

Planning

- ***Weeks 13 and 14*** (Dec 7th, 8th, 10th, 15th): Four slots to work on the implementation of your meta-heuristic algorithm.
- ***Exam week*** (date not fixed yet): Each pair of students will have 15 / 20 minutes to present their implementation and discuss their results
  - Presentation, demonstration (live or video)
- ***Deliverables due the exam day at midnight***: Presentation, video, final report