

TP2: Local Search and Simulated Annealing

Goals

The objective of this sheet is to:

- Apply local search heuristics, including simulated annealing, to simple problems.

Some other information:

- The coding language for this course is Python.
- Always write tests for each exercise to assure yourselves that your code is correct.
- You can discuss in groups, but the code and tests should be written individually.

1 Local Search

Exercise 1.1. Implement a greedy local search algorithm for the group work scheduling problem and test it with different neighborhood operators.

Exercise 1.2. Implement a greedy local search algorithm to minimize a given 2D function. Test it using the [Ackley function](#) and different step sizes.

Recall the formula for the Ackley function:

$$f(x, y) = 13 - 10 \exp\left(-0.14\sqrt{x^2 + y^2}\right) - \exp\left(\frac{\cos(2\pi x) + \cos(2\pi y)}{2}\right)$$

Exercise 1.3. Implement a variant of the algorithm above, where you specify the step size λ for the first iteration, and the step size decreases as a function of the iteration number.

Test different strategies for changing the step size, such as halving it each iteration, or setting it to λ/i or λ/i^2 where i is the iteration number. Which of these strategies works best?

2 Simulated Annealing

Exercise 2.1. Implement a simulated annealing algorithm for the group work scheduling problem and compare the results with greedy local search.

Exercise 2.2. Implement a simulated annealing algorithm to minimize a given 2D function.

Test it using the [Ackley function](#) and compare to greedy local search.