iscte

INSTITUTO
UNIVERSITÁRIO
DE LISBOA

# Introduction to Machine Learning — 2022/2023

## Evolutionary Algorithms

> These exercises should be solved using Python notebooks (Jupyter) due to the ability to generate a report integrated with the code. It is assumed you are proficient with programming. All answers must be justified and the results discussed and compared to the appropriate baselines.
> Each exercise is scored 1 point. Max score of the assignment is 4 points. Optional exercises help achieving the max score by complementing the errors or mistakes in the mandatory exercises.
> **Deadline:** October 22nd, 2022

Imagine a simplification of the mastermind game where we need to figure out a specific pattern. In our simplification we only have two colors and we can vary the pattern size. The pattern will be a bit sequence of a specific size.

## Exercise 1

Build the simulation environment. Develop the following functionalities:

a) Create a function that produces a random bit pattern of a specific size (the bit pattern may be coded as a string of 0s and 1s)

b) Create a function that will generate random patterns and measure how many attempts and how much time does it take to generate the "correct" bit pattern. Make two graphs on the evolution of attempts / time *vs* the number of bits in the pattern (2, 4, 8, 12, 16, …). Each "point" in the graph should be a box-plot based on the results of 30 trials. Use a fixed set of seeds to be able to reproduce the experiments. Stop when run-times on your machine surpass one hour to gather the results for a given number of bits even if you have not reached 16 bits. Compare the difference in attempts and run-times for each pattern size.

c) Create an evaluation function that measures the proximity to a pattern, resulting in a single number that should be zero if the pattern is an exact match and growing as the difference between the attempted pattern and the “correct” pattern increases.

d) Create a function inverse to the previous one that measures the “fitness” of the guessed pattern. This function should have a maximum value when the guessed pattern matches exactly the “correct” pattern and decreases as distance between patterns increases.

## Exercise 2

Create a function to mutate (flip one bit) a given pattern. Use this function in a cycle where the change is accepted only if it generates a better solution (i.e., a solution with higher fitness). Stop after 1000 mutations or when the pattern generated by mutation is equal to the solution. Does it always converge to the solution? If not, can you understand if you could generate a better pattern than the last sequence by using a single mutation?

## Exercise 3

Generate a random set of patterns (a population, for example, 100 patterns) and evaluate each of the patterns. Select the best 30% and, based on them generate by mutation a new set of 100: 30% of the new population corresponds to the 30% best and the remaining 70% are the ones generated by mutation. Repeat the processes until the best evaluation stagnates. Compare fairly the search methods tested so far, including their run-times.

## Exercise 4

Generate a random set of patterns (a population, for example, 100 patterns) and evaluate each of the patterns. Select the best 30% and, based on them, generate by crossover a new set of 100: 30% of the new population corresponds to the 30% best and the remaining 70% are the ones generated by crossover. Repeat the processes until the best evaluation stagnates. Compare fairly the search methods tested so far, including their run-times.

## Exercise 5 [Optional]

Explain the changes that would be necessary in the evaluation function, mutation, and crossover to deal with a similar problem where the size of the target bit pattern would be unknown.

**Recommendation:** try to implement your ideas in order to better understand the necessary changes. Include the code in the submission.

## Exercise 6 [Optional]

Explain the changes that would be necessary in the evaluation function, mutation, and crossover to deal with a similar problem where the pattern would be of decimal digits and not binary.

**Recommendation:** try to implement your ideas in order to better understand the necessary changes. Include the code in the submission.

## Exercise 7 [Optional]

Explain the changes that would be necessary in the evaluation function, to apply this procedure to the robot-maze problem in the previous set of exercises (Reinforcement Learning).