CT421 Programming exercise 1: Evolutionary Search

Colm O'Riordan

January 14, 2025

1 One-max problem

Consider the simple case of evolving a string that contains all 1s in every location. Let the length of the strings be 30. The initial population should be randomly created. Use standard mutation and one-point crossover. The fitness of a solution is the number of `1`s in the string.

Plot the average fitness of the population versus the generations passed.
This exercise is to show the operation of a genetic algorithm. We know the optimal solution in advance. This represents a very easy search landscape and is included to illustrate the operation of a genetic algorithm.
When plotting the average fitness, you don't need to write code to do the plotting; feel free to output the values to a text/csv file and use excel, gnuplot or other to plot the results.

2 Evolving to a target string

This is the same as before but instead of evolving to a string of `1`s, the population will evolve to find some target string.

Define a target string (a sequence of 1s and 0s) and adopt the same approach as before with an appropriate fitness function (number of matching values). Plot the average fitness as before.

3 Deceptive Landscape

Modify the fitness function from the one-max problem, such that the fitness function is equal to the number of 1s in the string for all cases except when there are no 1s present. In this case, the fitness should be 2× `(length of the solution)`. Plot the average fitness over time.

This problem represents a deceptive problem and should be very hard for the GA to find the optimal.