UPPSALA UNIVERSITET
DEPARTMENT OF MATHEMATICS
Denis Gaidashev

Modeling Complex Systems (1MA256)
Spring 2023

LAB 5: Genetic algorithms. Fractals

The deadline for this assignment is June 11th.

Please submit by email, upload to GitHub or any other "box", if the files/data are too large (provide a link). All code should be included. Feel free to submit videos illustrating your results where appropriate, by email or uploaded elsewhere such as vimeo or youtube. You may work in groups of size 1-7, and only one group member needs to submit the assignment. State clearly the members of the group.

Part 1: For this problem we will imagine we have a painter robot similar to the robot which picked up cans in the lectures. We will use this robot to paint the floor of a room. To make it interesting, the painter starts at a random place in the room, and paints continuously. We will also imagine that there is exactly enough paint to cover the floor. This means that it is wasteful to visit the same spot more than once or to stay in the same place.

To see if there is an optimal set of rules for the painter to follow, you will create a genetic algorithm. You may write your own code from scratch or use painter_play.m or painter_play.py, linked in Studium, as starting points. As inputs, this function receives:

a) A chromosome: A 1×54 array of numbers between 0 and 3 that shows how to respond (0: no turn, 1: turn left, 2: turn right, 3: random turn left/right) in each of the 54 possible states. The state is the state of the squares forward/left/right and the current square. Let $[c, f, l, r]$ denote states of the current square, forward square, left square and right square respectively. Write 0 for empy, 1 for wall/obstruction and 2 for painted. Note that $c \in \{0, 2\}$ and $f, l, r \in \{0, 1, 2\}$ so there are $2 \times 3^3 = 54$ possible states.

b) An environment: A 2D array representing a rectangular room. Empty (paintable) space is represented by a zero, while furniture or extra walls are represented by ones. Outside walls are automatically created by painter_play().

The function painter_play() then uses the rule set to guide a painter, initially placed in the room with a random position and direction, until the paint can is empty. Note that the painter does not move when it tries to walk into a wall or furniture. The efficiency (total fraction of paintable space covered) is then given as an output, as well as the $x - y$ trajectory (i.e. the positions of the painter at each time step) of the painter. To see that the painter works, you can try passing it an empty room for an environment and a trivial chromosome. For example, a chromosome consisting of all 3's produces a kind of random walk.

Now do the following: Create 100 random chromosomes in a 100×54 matrix, as well as a 30×60 empty room. Create a genetic algorithm to evolve this population over 200 generations, playing each chromosome several times and storing the chromosomes average efficiency as the fitness. You may choose any rule for picking the next generation from the previous one so long as it includes crossovers and mutation and that individuals with higher fitness are more likely to have offspring in next generation. (An example is to use a two-point crossover with a mutation rate of 0.005 per locus per generation.) Plot the final set of chromosomes. Plot an example trajectory of one of the more successful chromosomes (or make a video).

Part 2: Write a code which does the following:

1) Take any three points in a plane to form a triangle.

2) Randomly select any point inside the triangle and consider that your current position.

3) Randomly select any one of the three vertex points (with some probabilities $p_1+p_2+p_3 = 1$).

4) Move half the distance from your current position to the selected vertex.

5) After first hundred steps, start plotting the current position.

6) Repeat from step 3).

Question 1: What do you obtain as the limit set if $p_1 = p_2 = p_3$? What do you obtain if they are not equal?

Question 2: Write out the iterated function system which describes this process.