UPPSALA UNIVERSITY
DEPARTMENT OF MATHEMATICS
Fiona Skerman

Resit version
Summer 2020

Lab 6: Genetic Algorithms

The deadline for this sheet is midnight Monday 31st of August.

Please submit hand-ins on Studentportalen. All code should be included. Please feel free to submit videos illustrating your results where appropriate, via Studentportalen or uploaded elsewhere. You may work in groups of size 1-4, and only one group member needs to submit the assignment. State clearly the members of the group.

6. Painting and sometimes climbing robot

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. NB: The first question is the same as in Lab 6 in the course.

To see if there is a 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` as starting points.

As inputs, this function receives

a A chromosome: A 1x54 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 $[c, f, \ell, r]$ 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 $f, \ell, r \in \{0, 1, 2\}$ so there are $2 \times 3^3 = 54$ possible states. (For the last question $c \in \{0, 1, 2\}$ so this will need to be extended to 81 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 3s produces a kind of random walk. Now do the following:

1. Create 50 random chromosomes in a 50x54 matrix, as well as a 20x40 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 single-point crossover with a mutation rate of 0.002 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). Is this what you expected? **(4 points)**.

2. Plot the genetic diversity in the population vs generation. We can also use as a measure of genetic diversity the average number of pairwise differences

$$D(t) = \frac{1}{\ell} \sum_{k=1}^{\ell} \frac{1}{N(N-1)} \sum_{ij, i \neq j} I(c_{ik}(t) \neq c_{jk}(t)) \tag{1}$$

   where $c_{ik}(t)$ is the value of the ith chromosome at locus k at time t, the indicator I is one if the inequality is true and zero otherwise, N is the number of chromosomes and ℓ is the length of each chromosome. Is this what you expected? Explain any trend that you see. **(2 points)**

3. **A (badly) climbing painter robot**. We now make a change to the room and to how the automata operates. Add some (low) walls to the empty room (say about 50-100 square metres in total). If the painter tries to move into (low) walls it succeeds with probability 1/2. (If the painter tries to move into the 4 boundary walls it does not succeed). Each step the painter makes a decision (i.e. looks up its chromosome) what to try to do based on the states of its current square and the forward, left and right squares - for this the low wall counts as a wall. Notice that it is now possible for the current position to be a wall and

hence the chromosome length needs to be increased to 81 (from 54).

a. For the room with (low) walls. Create 50 random chromosomes, and 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. Plot an example trajectory of one of the more successful chromosomes (or make a video).

b. Repeat (a) except now train the chromosomes on an empty room. How well do the more successful of these chromosomes perform on the empty room and on the room with (low) walls.

For each of (a) and (b) take one of your highly evolved chromosomes, and plot the trajectory (or make a video). How do the strategies compare? **(4 points)**