

Lab 4 - Natural Computational Methods

Alejandro Villaron and David Håkansson

March 2021

1 Question 1:

Considering the fact that we are exchanging tails, and our genome is encoded with only two variables, then only the Y-axis coordinates would be switched during 1-point crossover. If we look at the example image what would happen is: p1 would "go up" to the Y value of p2, and p2 would "go down" to the Y value of P1 originally. You can also look at this as switching X coordinates since $(x1, y1), (x2, y2) \Rightarrow (x1, y2), (x2, y1) = (x2, y1), (x1, y2)$ and say that the points "go right" and "go left".

From our understanding, mutation in this example would correspond to taking the y or the x value and completely randomizing it. Which geometrically is equivalent to taking a point p, drawing a line that is parallel to axis which mutation occurs and the parallel line goes through the original point p. Then anywhere on the parallel line our new point will occur.

Cloning would just be copying the point.

2 Question 2:

More improvement would be expected from the mutation technique since with crossover only the tails are swapped, the Y values in this case, while with mutation the change would affect both axis. Also the amount diversity that can be generated from crossover is very low since no matter what type of crossover from 2 data points $p_1 = (x_1, y_1), p_2 = (x_2, y_2)$ the new points that can be generated is $p_1, p_2, p_3 = (x_1, y_2), p_4 = (x_2, y_1)$.

Plot 1

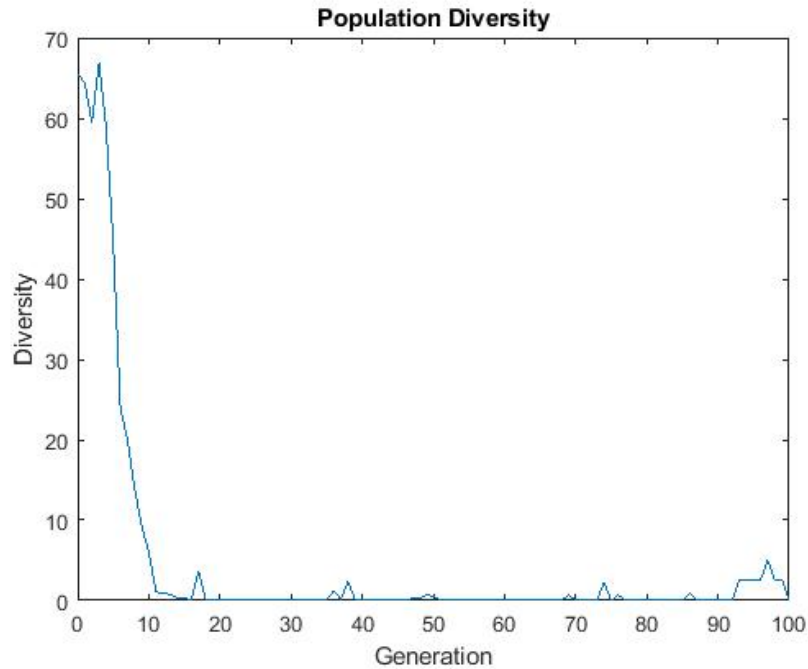


Figure 1: *plot showing the change in diversity of the population by generation.*

3 Question 3:

Our primary source of diversity is through mutation, we can observe this in the graph if we look at when the spikes happen. Crossover should not have that much of an effect on diversity since the amount new data points that can be generated from crossover is highly limited.

4 Question 4:

If we do not have enough diversity in the population that then the amount of search space we explore will be very low and we will converge very quickly.

5 Question 5:

Rank doesn't give as high values of probability as fitness selection since the probability of a point being selected in rank fitness is dependant to the rank rather than the fitness. For example if we in the first generation we get a point that has 5 times of the second best point then we would give that point a very high probability of being selected in fitness selection which would lead to us

converging to that point very quickly without really exploring a lot (premature convergence).

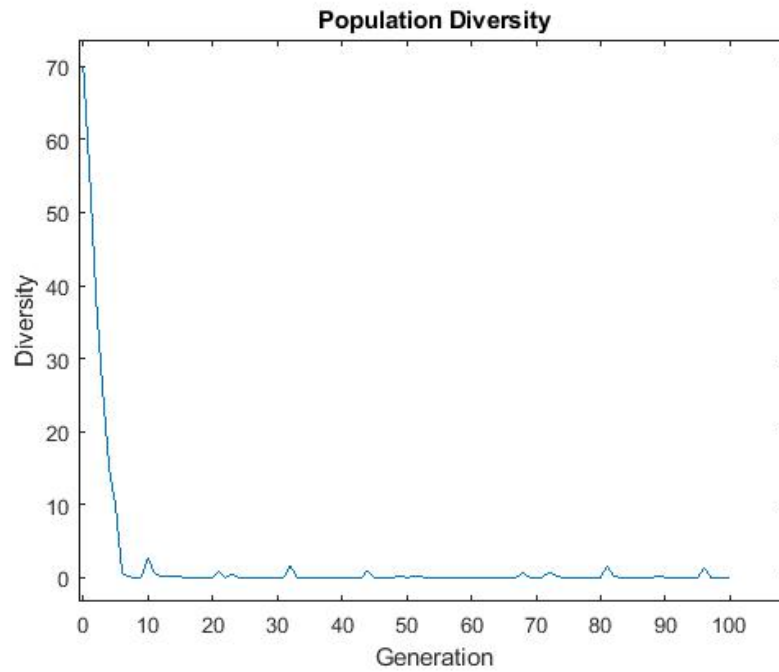


Figure 2: *change in diversity of the population after using a rank based selection with $p = 2$*

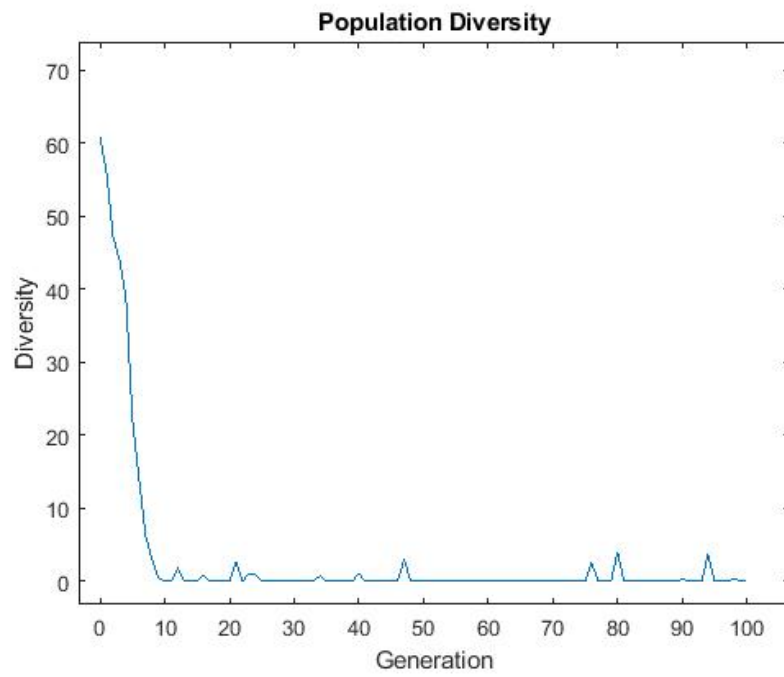


Figure 3: *change in diversity of the population after using a rank based selection with $p = 1.75$*

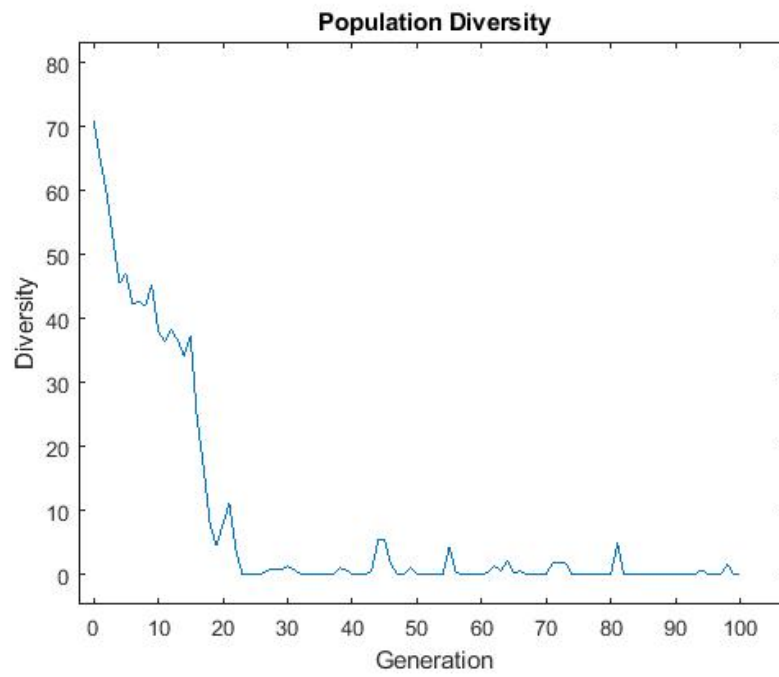


Figure 4: *change in diversity of the population after using a rank based selection with $p = 1.5$*

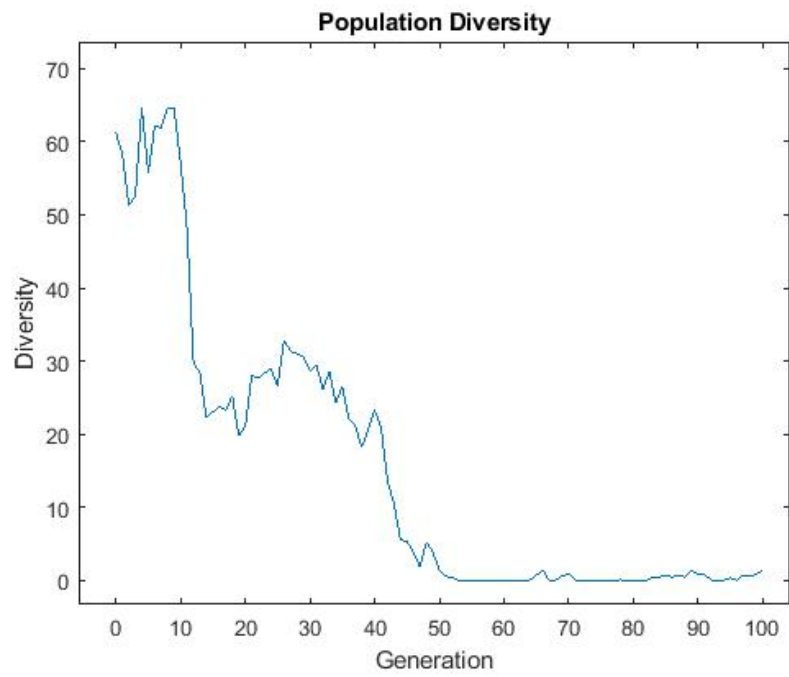


Figure 5: *change in diversity of the population after using a rank based selection with $p = 1.25$*

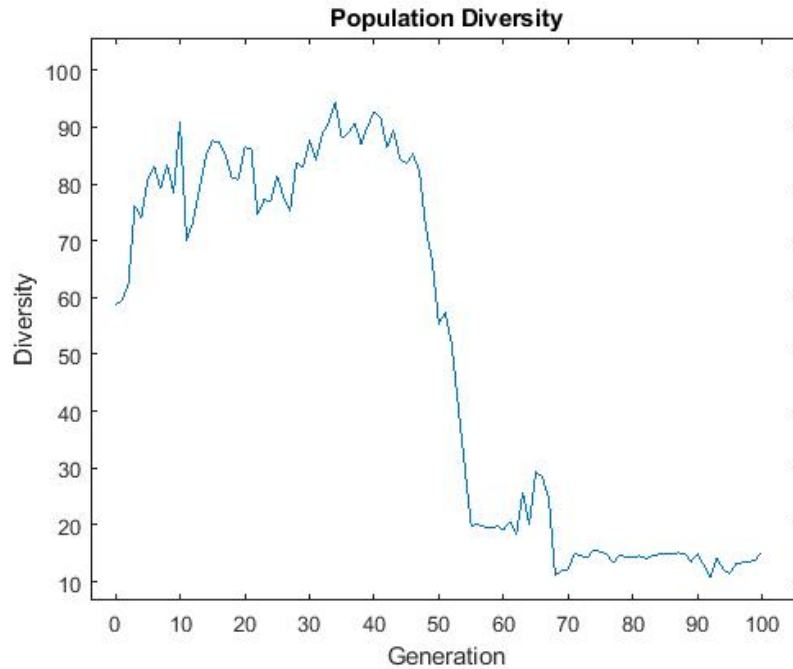


Figure 6: *change in diversity of the population after using a rank based selection with $p = 1.0$*

6 Question 6:

By looking at the diversity curves we thought that 1.25 was the best one. We can think of the graph in two phases when the graph has high diversity that corresponds to us looking for local maximums, while when it has low diversity it tries to move towards the best local maximum it has found. So in order to find a good local maximum we need to have high diversity for sufficiently long and in order for us to converge to the local maximum we need to have low diversity for sufficiently long. The best results we got for pressure = 1.25, were a fitness value of 3.4925 with the position [7.2691 3.9919].

7 Question 7:

It will lead to us converging a lot quicker which can lead to premature convergence.

8 Question 8:

We used the same settings as on star1 and got what we think are good solutions for star2 and star3. In star2 we got a fitness of 2.99 and it was in an area where the fitness could be further increased if it moved a little(it was possibly around the global optimum). As for star3 we managed to get a point in the area of the global optimum although it never got to it. Not all solution moved towards the global optimum though.

9 Question 9:

Because the algorithm would get stuck on the local minimum instead of converging into the global minimum.(If the question refers to Ackley's function, then it is extra difficult because of all the local minimums)

10 Question 10:

Using mutate decay could we imagine could be good since it would make a lot of mutation initially which could help us a lot to find the global minimum and with the reduced mutation later on it could help us get to the local minimum.

Using mutate proportional might be good since then if we find the global optimum we are less likely to mutate away from it.

Having different decreasing crossover weight we imagine would be worse than having the same weights since having decreasing weights would make use converge faster.

When we have same as the crossover weights we use this vector: ones*0.1(the 0.1 is probably unnecessary) When we have decreasing as crossover weights we use this vector:

mutate.decay	mutate.proportional	crossover.weight	avg fitness
none	true	same	11.641
linear	true	same	12.659
exponential	true	same	11.722
none	false	same	4.338
linear	false	same	5.968
exponential	false	same	4.59
none	true	decreasing	11.0912
linear	true	decreasing	13.329
exponential	true	decreasing	11.8798
none	false	decreasing	4.8987
linear	false	decreasing	6.2986
exponential	false	decreasing	4.6285

11 Question 11:

After running it three times with the best configuration obtained in the previous question, we got the results of : 17.1151, 16.4954, 15.0606. Which is a bit worse fitness than when we had 20 dimensions, but considering the number of dimensions increased by quite a lot it is hard to say if this is a bad result or not. One way to increase the performance would be to increase the population size. Another way would be to increase the number of generations.

12 Question 12:

The input is an image of the screen with the elements that the player can stand on simplified into a white square and elements that are moving simplified into a black square.

The outputs of the neuron are which buttons are pressed by the network.

13 Question 13:

When doing naive crossover it is possible that a link is formed between nodes that the offspring do not have. This is handled by NEAT through innovations, which is a data structure that contains information about links and nodes created through the use of global IDs. With those innovation IDs we can do crossover in a cleaver way where we know the prerequisites for certain innovations.

14 Question 14:

It decreases drastically, always in a range of values between 3 and 4 times the velocity limit. The relationship with the theoretical velocity limit is a difference of values of approximately the velocity limit. For example, for the velocity limit of 30, the maximum value avg was 106, and the theoretical velocity limit was 135.

15 Question 15:

c_1	c_2	Initial Weight	Velocity clamp	Best Score
2	2	1	off	20.74
2	2	1	5	7.621
2	2	1	30	19.4313
1.49	1.49	0.7968.	off	2.3639
1.49	1.49	0.7968.	5	0.487
1.49	1.49	0.7968.	30	1.1809
2	1.3	0.7968	5	1.8066
1.3	2	0.7968	5	2.227
1.3	1.3	0.7968	5	0.1410

From the table we can see that the best value we got was 0.1410 and we achieved this when $c_1 = c_2 = 1.3$

16 Question 16:

When we used $c_1 = c_2 = 1.49$ and with an initial weight of 0.7968 we managed to avoid swarm explosion without velocity clamp.

17 Question 17:

We achieved a fitness value of 0.841 with the following combination of values: $C1 = 2$, $c2 = 1.2$, decaying weight = [0.8,0.4]

18 Question 18:

$$\chi = \frac{2}{\phi - 2 + \sqrt{\phi^2 - 4\phi}}$$

19 Question 19:

$$\chi = 0.268, w = \chi, c_1 = \phi_1 * \chi = 1.072, c_2 = \phi_2 * \chi = 0.536$$

20 Question 20:

The particles converge very quickly, in generation 10-15 it has about the same score as in generation 175. However it does not seem to move towards the global optimum and only gets best scores around 10.

21 Question 21:

Those 9 dimension correspond to the 6 weights and the 3 biases of the network used in the Lab 1.

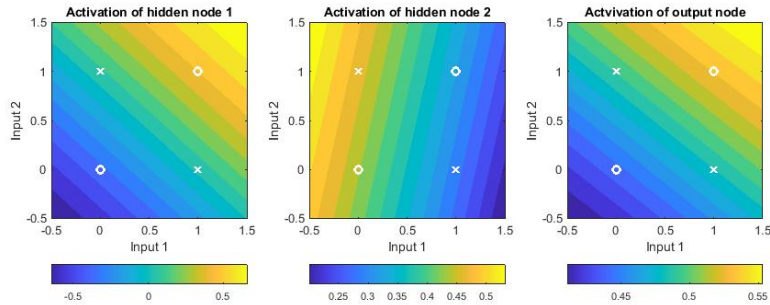


Figure 7: *Solutions for the xor problem after adjusting the neural network's weights using particle swarm optimization*

We did not get any good solutions despite trying various configurations, the reason behind this we think is the fact that we have small population size, even when only using 7 particles it took several minutes to train so we only used a population of 5 for most experiments. If that is not the case we do not get why we did not get any good solutions, we tried having the constriction parameters from the question above and the best parameter from question 15 among other things like having high weights for the personal best but not for the global best.

22 Question 22:

The different outputs of the activation function seem to be within a much smaller range. maybe this is because gradient decent reaches the local minima while we are not ensured to reach it for PSO. We also get better results from lab1 than we get now that might be because of parameters or it is because gradient decent is better for this problem.

23 Question 23:

We think the lab wraps up the content of the lectures and is useful to understand the relevant concepts.