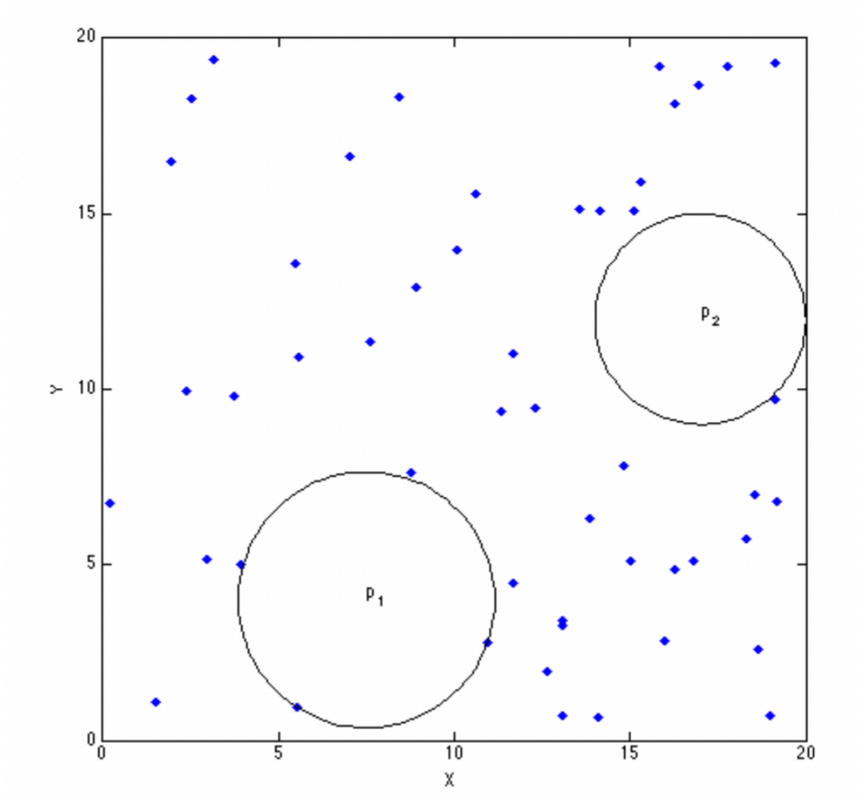
Evolutionary Computation and Swarm Intelligence

Task 1: Genetic Algorithms



Question 1: Think about what each of the genetic operators means for this simplistic genome. Geometrically speaking, what would a 1-point crossover look like in this case? What about mutation?

Since the genome is represented by only two cartesian coordinates (x_i,y_i), a 1-point crossover means that for two individuals, (x_1,y_1) and (x_2,y_2), we swap their x and y coordinates to get - (x_1,y_2) and (x_2,y_1).

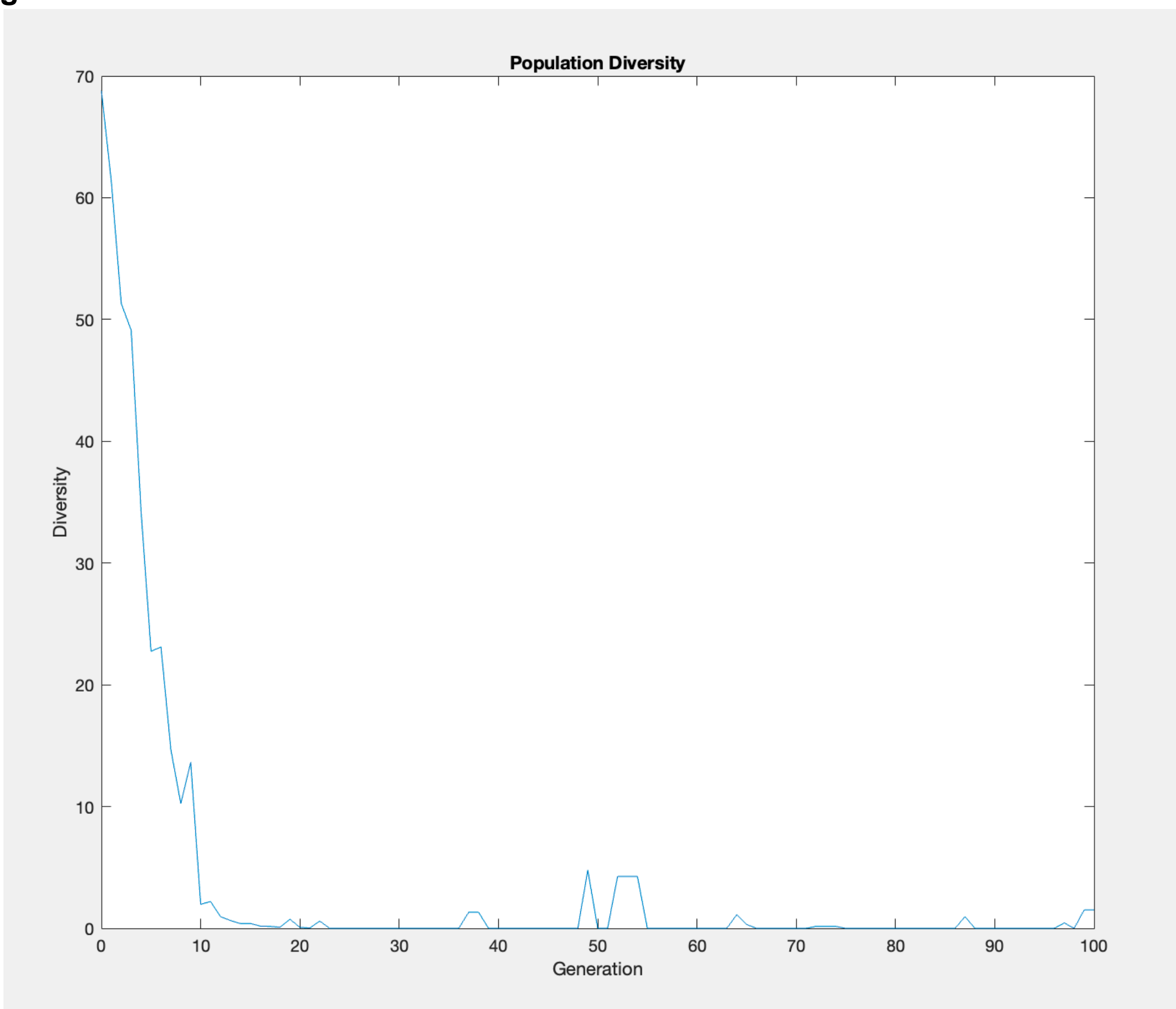
For mutation, we simply shift the centres by a random noise so -
$x_1' = x_1$ + random noise
$y_1' = y_1$ + random noise

Question 2: Would you expect more improvement in this problem to result from crossovers or mutations? Why? Is that what you would normally expect?

We expect better results from mutation here because crossover can only yield 4 possible individuals from 2. That is a very limited search space out of all continuous points in the geometric space. Mutation can provide a different individual every time resulting in higher chance of a better individual.

Usually, crossover works better when the situation deals with combinatorial problems with a genome that is more readily crossed with other individuals. Mutation in these cases can result in losing an important bit of information in the genome.

Plot 1: Include a plot showing the change in diversity of the population by generation.



Question 3: What are the possible sources of population diversity when using genetic algorithms? How are those sources of diversity reflected in the shape of the diversity curve in your plot?

1. Initialization - Randomly initializing the population ensures a broad exploration of the search space from the beginning.

2. Crossover (Recombination) - Single-point, multi-point, or uniform crossover combines genetic material from parents, creating new variations.

3. Mutation - Randomly modifying genes (e.g., changing values, flipping bits) promotes genetic diversity.

We can see the curve in three stages -

1. Initially, the diversity is high due to random initialization and there is no impact of crossover or mutation.
2. Then the diversity declines rapidly due to crossover limiting the new individuals to high fitness and converges quickly.
3. Finally, we can see the effects of mutation after around 10 generations where it tries to introduce small changes in population but the chance of getting high fitness through mutation is rare.

4. **Cloning & Elitism -** Clone the best fitness individual - reduces population diversity. Cloning high-performing individuals, especially when elitism is enabled (replace.elitist = true), causes certain individuals to persist across generations. This reduces diversity as the same genomes dominate the population.

5. **Replacement Strategy** – Using 'replace.func = all' means each new generation completely replaces the old population (except the elite individual), which can accelerate convergence and reduce diversity quickly if selection pressure is high.

Question 4: Why is population diversity important?

1. Preventing Premature Convergence - If diversity is too low, the population becomes homogeneous, and the algorithm gets stuck in a local optimum.

2. Ensuring Effective Exploration - High diversity allows the algorithm to explore a wider range of possible solutions, reducing the risk of missing the global optimum.

3. Balancing Exploration and Exploitation

- **Exploration:** High diversity promotes trying new solutions.
- **Exploitation:** Low diversity refines good solutions.
- Maintaining a balance ensures the GA doesn’t waste time exploring too much or converging too early.

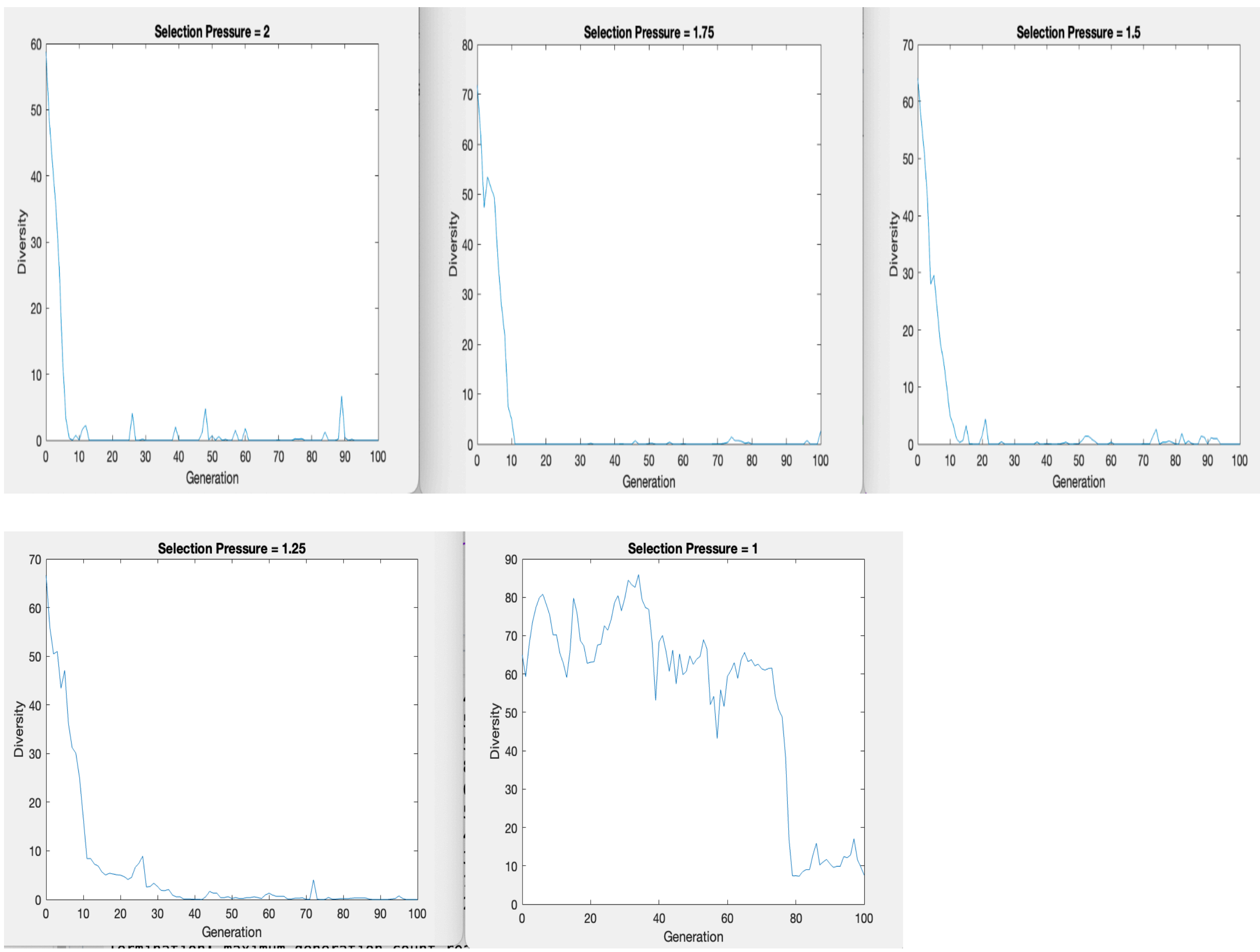
Question 5: What's the difference between rank and fitness selection? Why does that make rank based selection better at avoiding premature convergence?

Fitness selection means that individuals are selected with probability proportional to their fitness.

Rank selection implies that individuals are first sorted based on their fitness score then selected on their position in the list.

This is better because in fitness selection a highly fit individual will be picked almost always as compared to rank selection. The rank selection gives smoother selection pressure and avoids domination by a single high-fitness individual early on.

Plot 2: For each pressure value of 2, 1.75, 1.5, 1.25 and 1 provide one plot of the population diversity over 100 generations using rank based selection.



Question 6: What selection pressure results in the most promising looking diversity curve? Run the algorithm several times using that pressure setting. Report the fitness value and position you were able to locate?

We get the best curve at pressure of 1 since there are still around 10 distinct individuals after 100 generations vs nearly zero for other pressure values.

1) Best individual had a fitness of 2.4680 at [5.9775 8.5035]
2) Best individual had a fitness of 3.1703 at [6.9704 4.1168]
3) **Best individual had a fitness of 3.5982 at [15.9050 11.2884]**
4) Best individual had a fitness of 3.5065 at [7.6396 3.9450]
5) Best individual had a fitness of 3.5288 at [7.6874 4.1308]

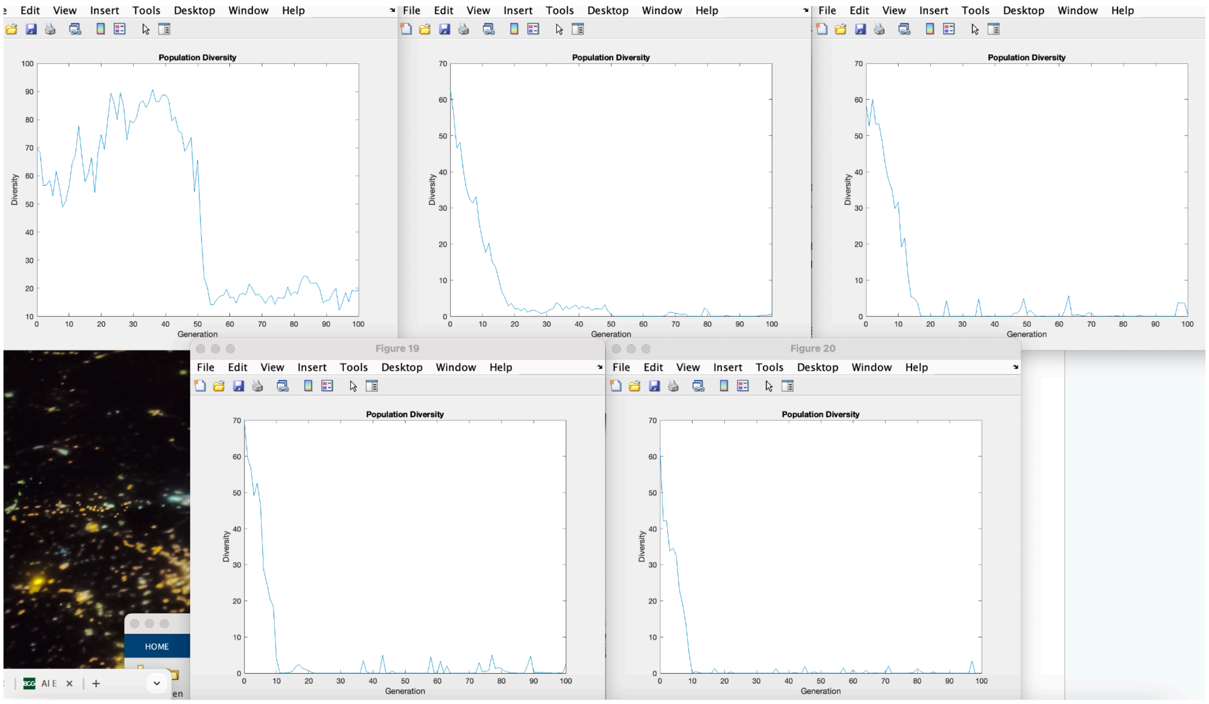
NOTE: Setting **selection pressure** in a genetic algorithm means controlling how strongly better individuals are favored during selection. Higher pressure increases the chance of selecting fitter individuals, speeding up convergence but risking premature convergence; lower pressure maintains diversity but slows progress.

Question 7: What are the possible downsides of using elitism?
Elitism ensures that the best individuals always survive, which can lead to a rapid loss of diversity. As these elite solutions dominate, genetic variation in the population decreases, making it difficult for the algorithm to explore new areas of the search space. This often results in **premature convergence.**

Since elite individuals always survive, weaker individuals are frequently discarded, even if they contain valuable genetic diversity. This reduces the ability of the GA to explore new solutions. This makes the GA behave more like a greedy algorithm rather than an evolutionary process.

Question 8: What settings did you use? How well did the solver perform on the more difficult maps? Explain any difference in performance you observe.

Star 2



Pressure 2:

gen=100   Fitness worst=2.9291 mean=2.9291 best=2.9291

Time: 1.5211 Termination: maximum generation count reached.

Best individual had a fitness of 2.9291 at [9.3339 5.2313]

Pressure 1:

gen=100   Fitness worst=0.1141 mean=0.5810 best=2.7080

Time: 1.3552 Termination: maximum generation count reached.

Best individual had a fitness of 2.7080 at [9.2540 4.5647]

Pressure 1.5:

gen=100   Fitness worst=2.9097 mean=2.9097 best=2.9097

Time: 1.4474 Termination: maximum generation count reached.

Best individual had a fitness of 2.9097 at [17.0903 15.6012]

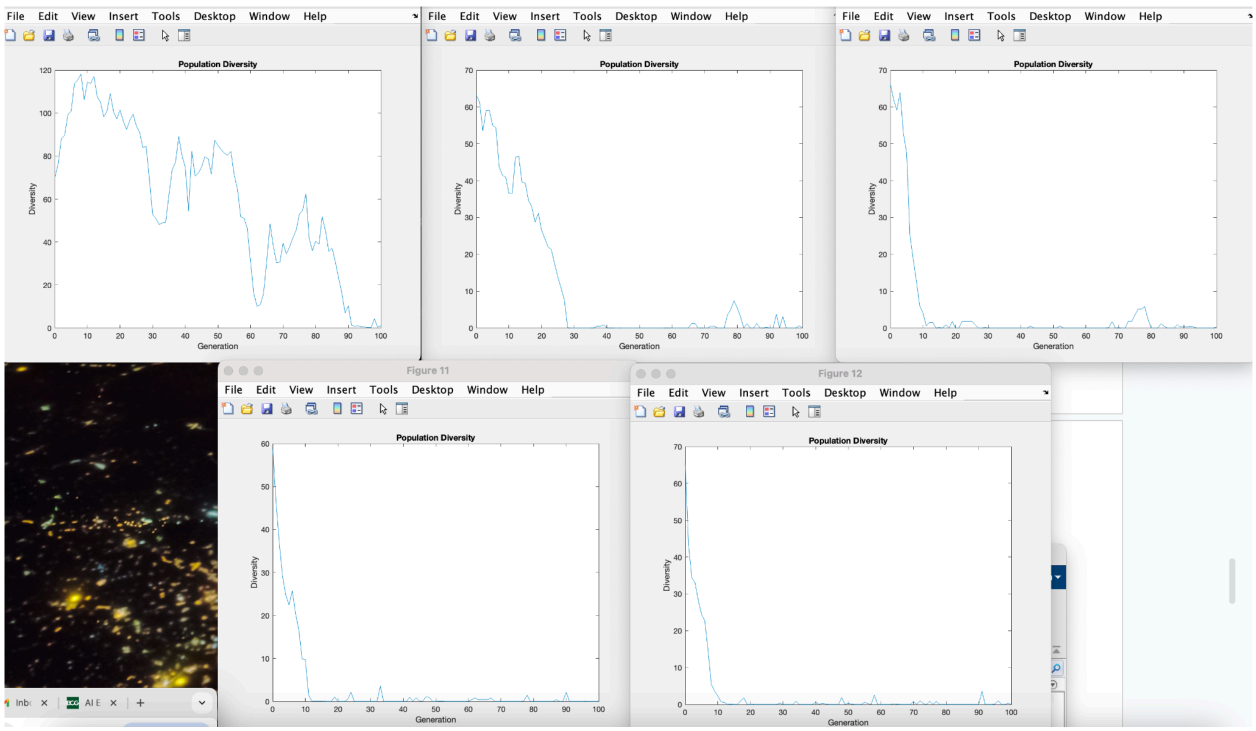
Pressure 1.75:

gen=100   Fitness worst=1.3926 mean=2.9253 best=2.9566

Time: 1.404 Termination: maximum generation count reached.

Best individual had a fitness of 2.9566 at [15.9040 16.6718]

Star 3



Pressure 1:

Time: 1.6008

Termination: maximum generation count reached.

Best individual had a fitness of 2.4320 at [5.6322 12.0715] and was born in generation 4 via crossover

Pressure 1.25:

Time: 1.4012

Termination: maximum generation count reached.

Best individual had a fitness of 2.7542 at [6.0423 5.4818] and was born in generation 5 via crossover

Pressure 1.5:

Time: 1.4876

Termination: maximum generation count reached.

Best individual had a fitness of 2.7211 at [6.2455 5.9575] and was born in generation 2 via crossover

Pressure 1.75:

Time: 1.5248

Termination: maximum generation count reached.

Best individual had a fitness of 2.4098 at [7.1084 7.0717] and was born in generation 47 via mutation

Pressure 2:

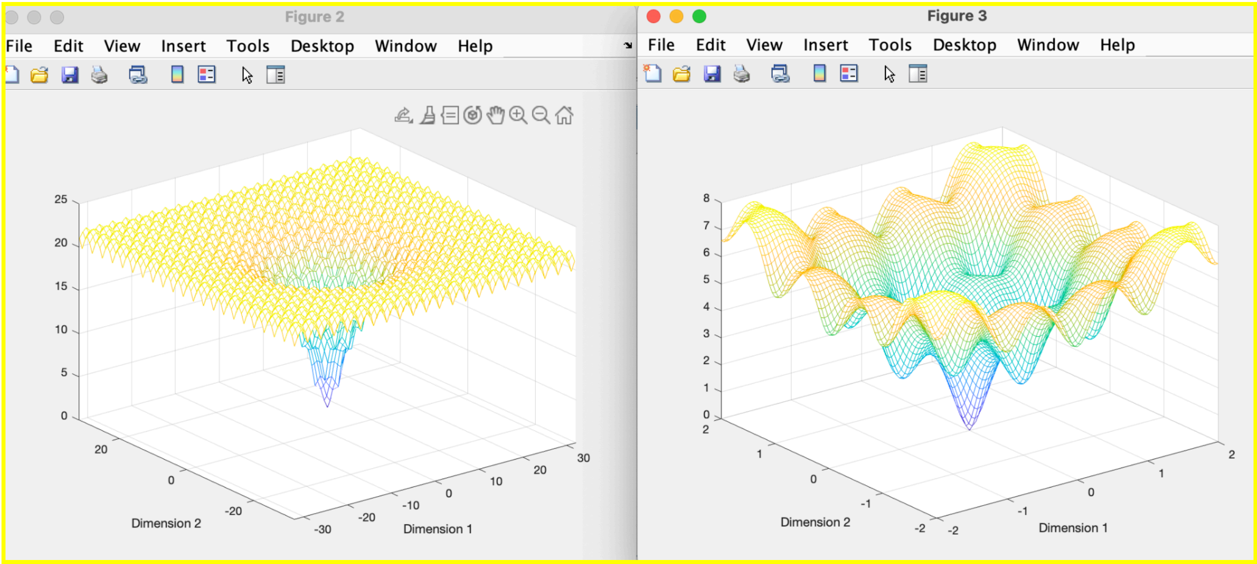
Time: 1.5272

Termination: maximum generation count reached.

Best individual had a fitness of 2.3188 at [6.9441 3.9753] and was born in generation 1 via crossover

Lower fitness in more difficult maps because there are more chances of getting stuck in the local minima.

Minimizing a Function:



Question 9: Why would the global minimum be difficult to find using a gradient descent method?

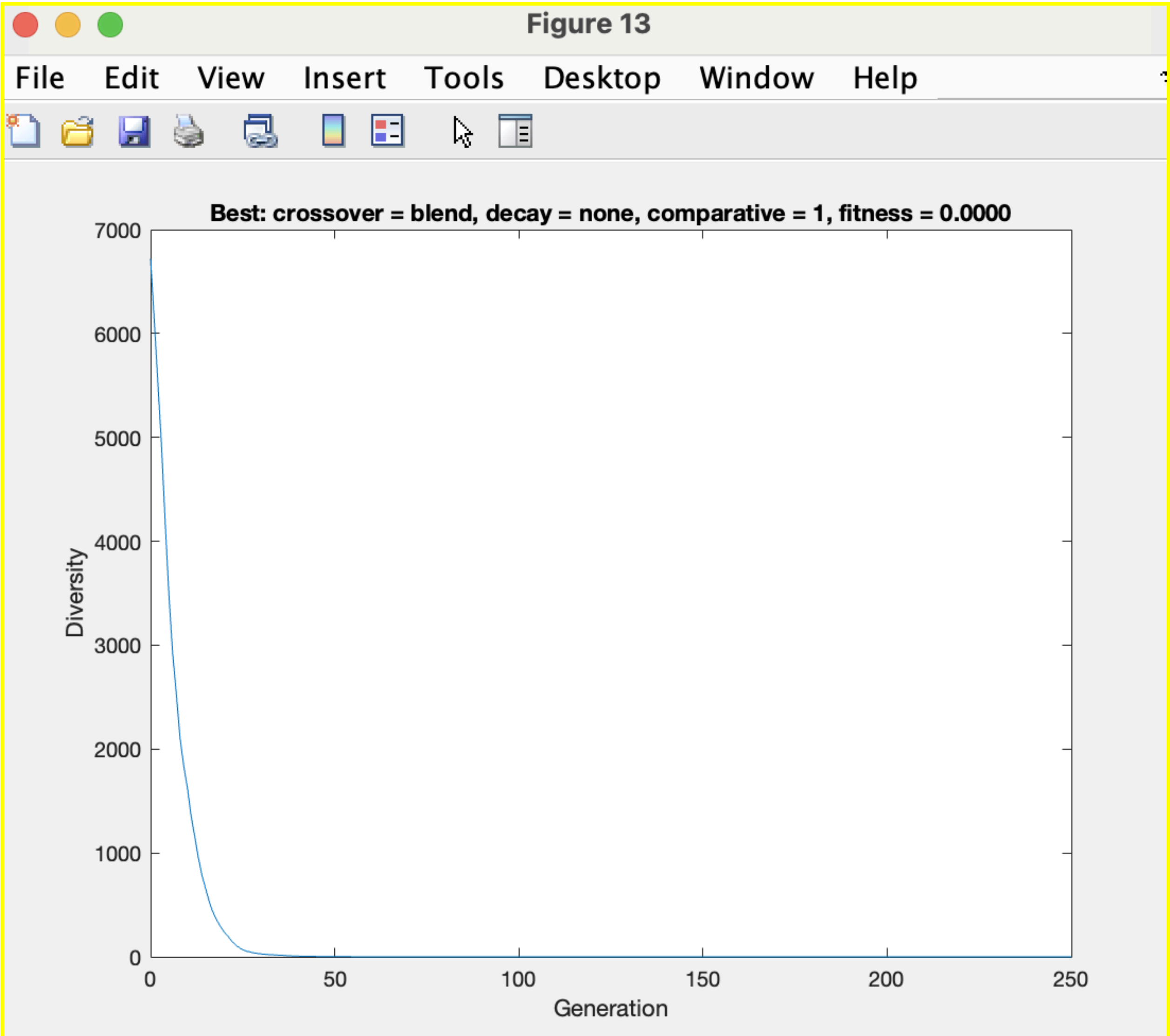
The global minimum of Ackley's function is difficult to find using gradient descent because the function contains many local minima that surround the global minimum. It is a local search method that updates a solution based on the local gradient (slope) of the function.

As a result:

- It can easily get trapped in a local minimum, especially if the starting point is not close to the global minimum.
- The gradient becomes small in flat regions, which slows down or completely stops progress.
- Since the function's surface has a rugged, bowl-like landscape, with ripples and dips, a gradient-based method lacks the exploration capability to jump out of poor regions.

In contrast, genetic algorithms are more likely to escape local minima due to stochastic operations like mutation and crossover, and thus have a better chance of reaching the global minimum.

Question 10: Explore at least ten different combinations of settings in total, make sure each setting is run several times. Write down (e.g., in a table), for each set-up what parameters you used, the average fitness you achieved. Motivate your choice of explored parameter combinations.

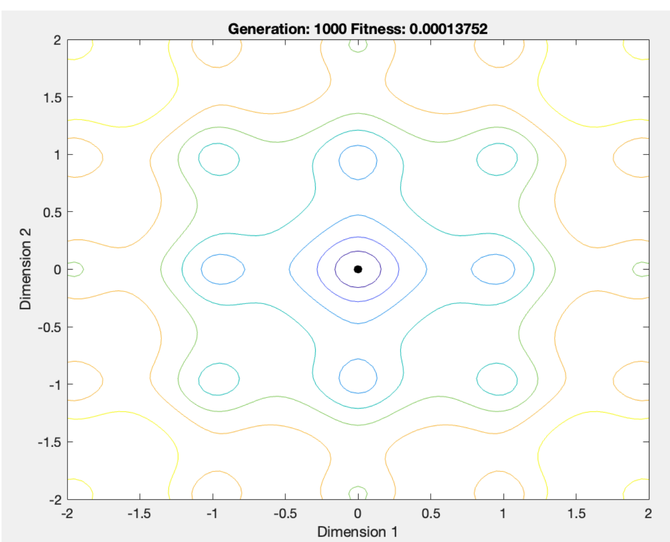


crossover_func	mutate_decay	replace_comparative	average_fitness
arithmetic	none	TRUE	2.7499
arithmetic	none	FALSE	2.3984
arithmetic	linear	TRUE	3.5099
arithmetic	linear	FALSE	3.4920
arithmetic	exponential	TRUE	2.2576

arithmetic	exponential	FALSE	3.1505
blend	none	TRUE	3.7185e-06
blend	none	FALSE	5.769e-06
blend	linear	TRUE	5.4475e-06
blend	linear	FALSE	3.007e-06
blend	**exponential**	**TRUE**	**2.8583e-06**
blend	exponential	FALSE	4.5443e-06
linear	none	TRUE	14.0578
linear	none	FALSE	12.7673
linear	linear	TRUE	19.9112
linear	linear	FALSE	19.8974
linear	exponential	TRUE	19.8962
linear	exponential	FALSE	19.8974

From the table, we can see that blend works best as the crossover, followed by arithmetic and then linear. Using comparative replacement seems like a better option in all cases. The mutation decay doesn't impact the fitness to a large extent.

Question 11: Try using GAsolver to find the minimum for the 100 dimensional Ackley's function, using the best parameters you found in the previous question. How well does it perform now and how can you improve the performance?



For 100 iterations, it is not ending up in the global maxima; but when we increase the iterations to 1000, it converges to the global maxima. As the dimensions have increased, we need more iterations to converge.

Question 12: What are the inputs and outputs of the networks evolved in the video?

Inputs are the pixels of the current game screen which is processed into a simpler image map represented as the black and white boxes.

Outputs are the possible buttons that can be pressed on the game controller, in this case, the A,B,X,Y and up, down, left, right

Question 13: What is the major challenge when doing crossover between neural networks, which NEAT addresses, apart from the fact that weights in one network are unlikely to mean the same thing in another?

- Each new connection or neuron gets a unique identifier. This allows meaningful combinations of structures during crossover.

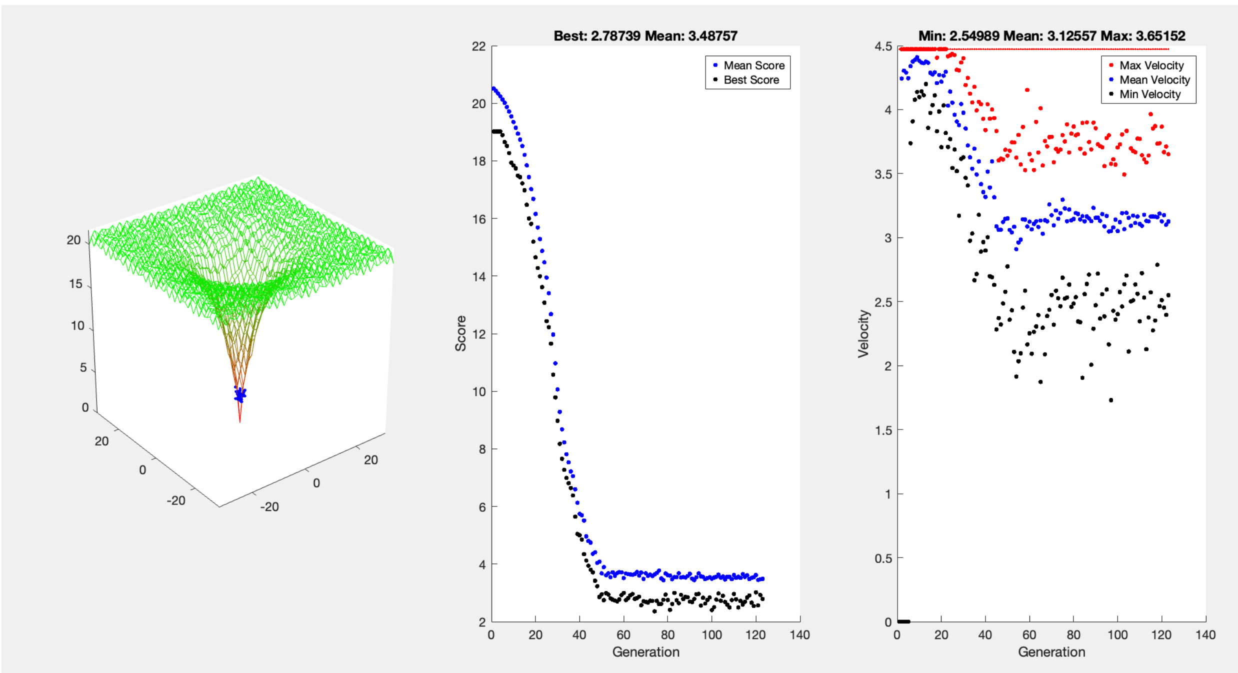
- Instead of forcing offspring to match parent structures, NEAT increases complexity gradually, where networks start simple and evolve more layers and neurons over generations.

- NEAT groups similar structures together and applies crossover only within similar architectures, preventing destructive recombination

Task 2: Particle Swarm Optimization

Question 14: What happens to the maximum velocity of the swarm over time when using velocity clamping? How does the maximum velocity compare with the theoretical velocity limit (the red line at the top of the velocity plot)?.

For max velocity set to 1 -

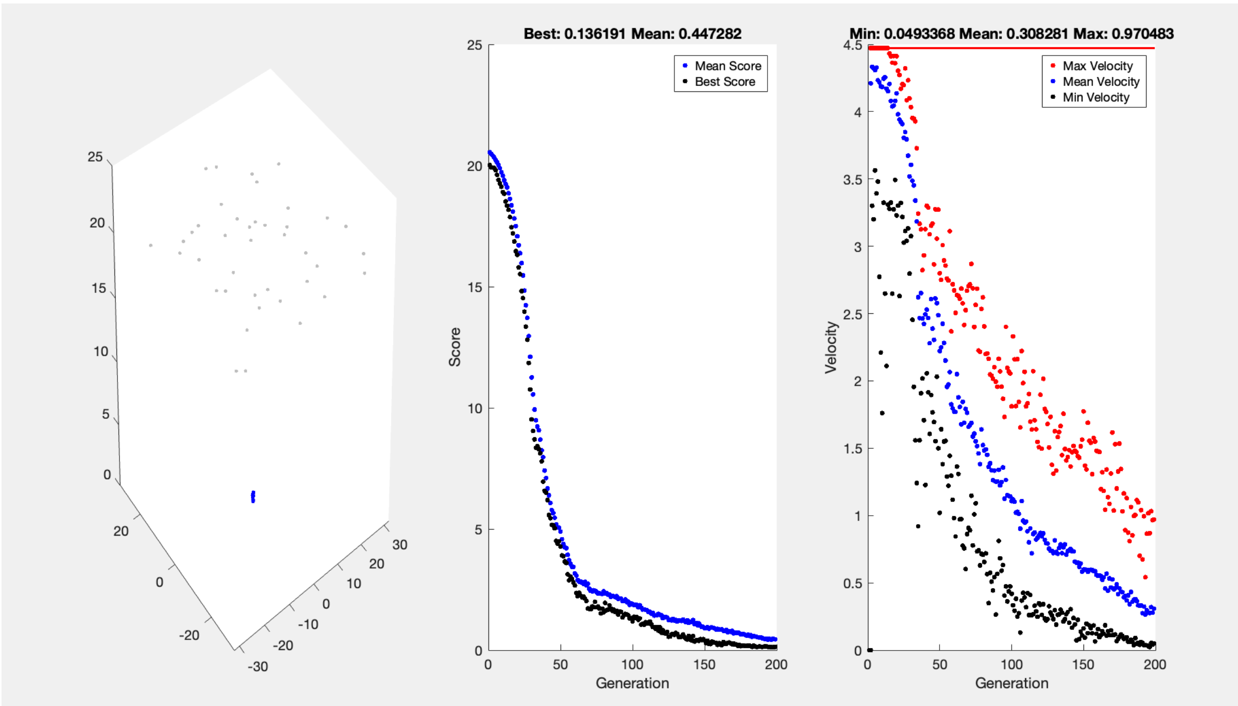


The maximum velocity decreases rapidly in the initial generations and then becomes relatively constant. **This suggests that the swarm is converging to the solution.**

It always remains <= the theoretical max velocity limit (vector magnitude for max velocity of 1).

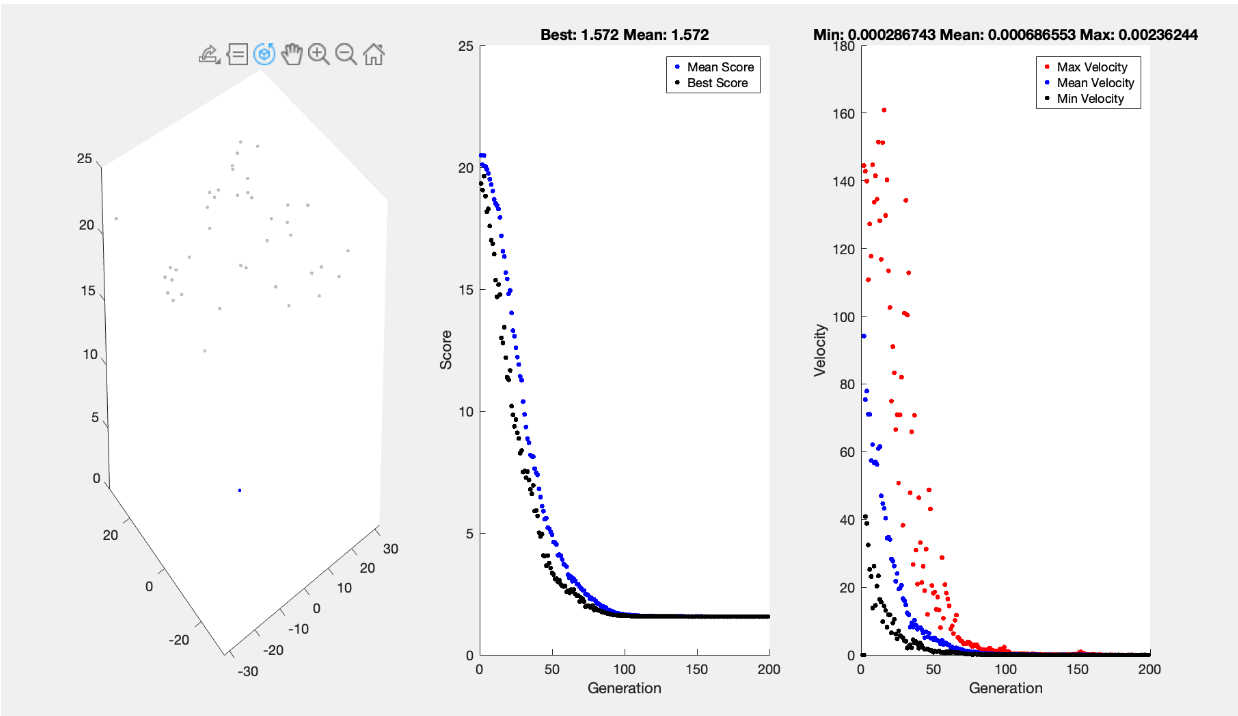
Question 15: What values for w, c1 and c2 worked best on this problem? How close did this swarm come to locating the global optimum?

Running PSO with w=0.7968, c1=1.4900, c2=1.4900, VelLimit=1 worked the best. It almost converged to 0. **-**



Question 16: Was velocity clamping still necessary to prevent swarm explosion, or were you able to find a combination of values that kept the swarm together?

Running PSO with w=0.5000, c1=1.7500, c2=1.7500, VelLimit=[] -

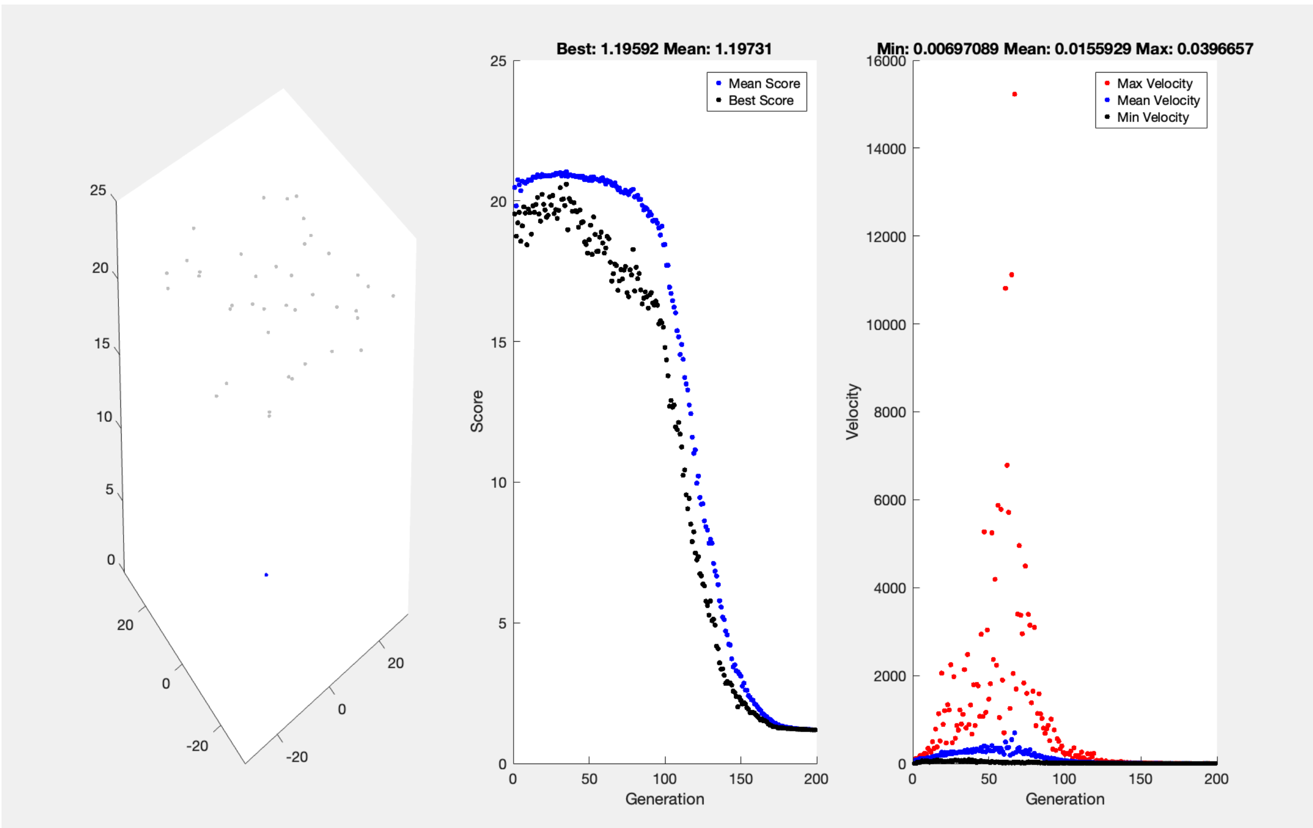


In this case, velocity clamping **was not used**, but the swarm still converged almost to the minima.

In both the cases, setting low inertia weight (<1) seems to work the best.

Question 17: What was the best combination of decaying inertia weights, and social and cognitive coefficients? What was the best fitness value you found using that Combination?

Running PSO with w=[0.9;0.4], c1=1.4900, c2=1.4900, VelLimit=[] -



This combination almost causes the swarm to converge. So having the possibility of low inertia weight gives more chance for the swarm to get to the minima.

Question 18: What is the equation for the value of the constriction coefficient in terms of φ?

The equation -

$$K = \frac{2}{\phi - 2 + \sqrt{\phi^2 - 4\phi}}, \text{where } \phi = \phi_1 + \phi_2 > 4$$

We know from the formula of constriction coeff. that **χ =** 0.2679

Question 19: Consider the constriction equation with $\varphi 1 = 4$, $\varphi 2 = 2$. What is the constriction coefficient for these values? What values of w, c1 and c2 would we have to use in Equation 3 in order to implement constriction with these values?

$$vid(t) = w*vid(t-1) + c1*r1[pid - xid(t-1)] + c2*r2[pgd - xid(t-1)]$$

$$= \chi vid(t-1) + \chi*\varphi 1*r1(pid - xid(t-1)) + \chi*\varphi 2*r2(pgd - xid(t-1))$$

To satisfy the above relation, we get -

W = χ

C1 = $\chi*\varphi 1$

= $\chi*4$

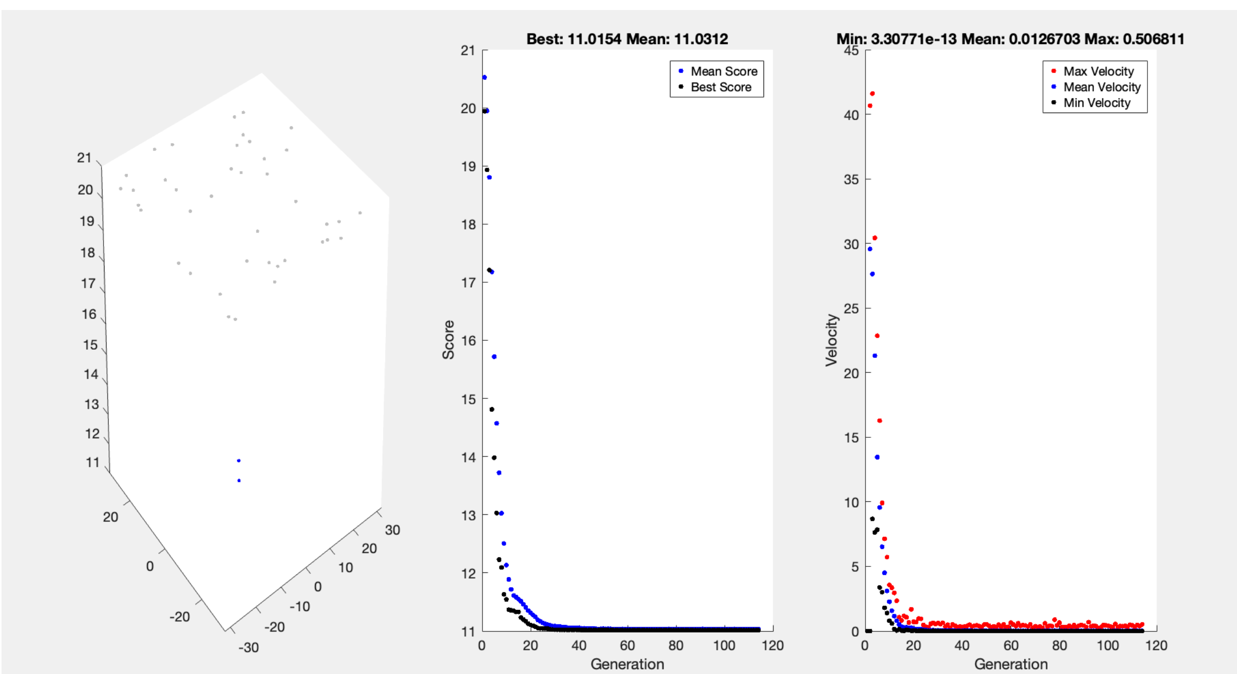
C2 = $\chi*\varphi 2$

= $\chi*2$

We know from the formula of constriction coeff. that χ = 0.2679

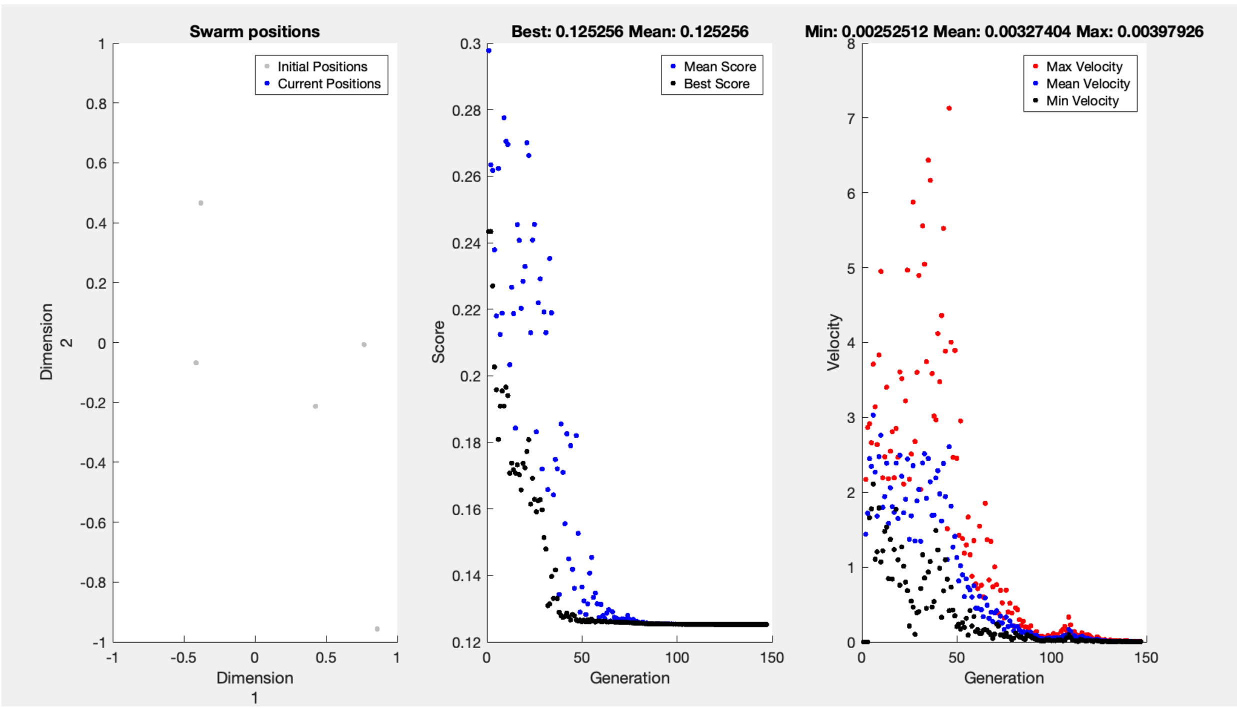
So W = 0.2679, C1 = 1.0716, C2 = 0.5358

Question 20: Describe the behavior of the swarm when using constriction. Does itlocate the global optimum? How quickly does it converge?



With Constriction, the PSO converges in 20 generations, but it doesn't arrive at the global minima. We got better results with other sets of parameters. (increase iterations; we still have some velocity at the end).

Training a Neural Network using PSO



Question 21: Given the network topology we used in Lab 1 and the problem definition above, we've defined the problem as having 9 dimensions. What possible explanation is there for that?

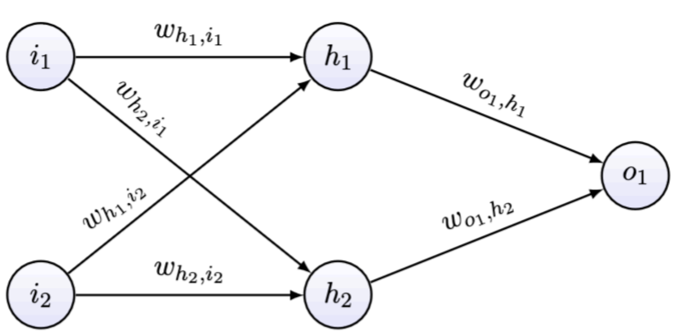
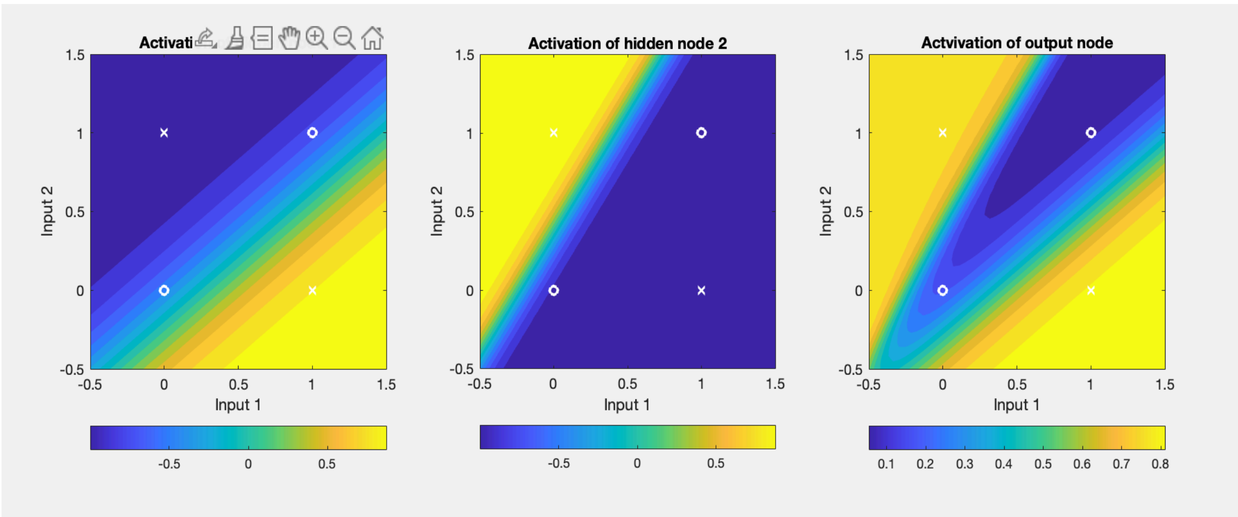


Figure 1: A 2-2-1 MLP.

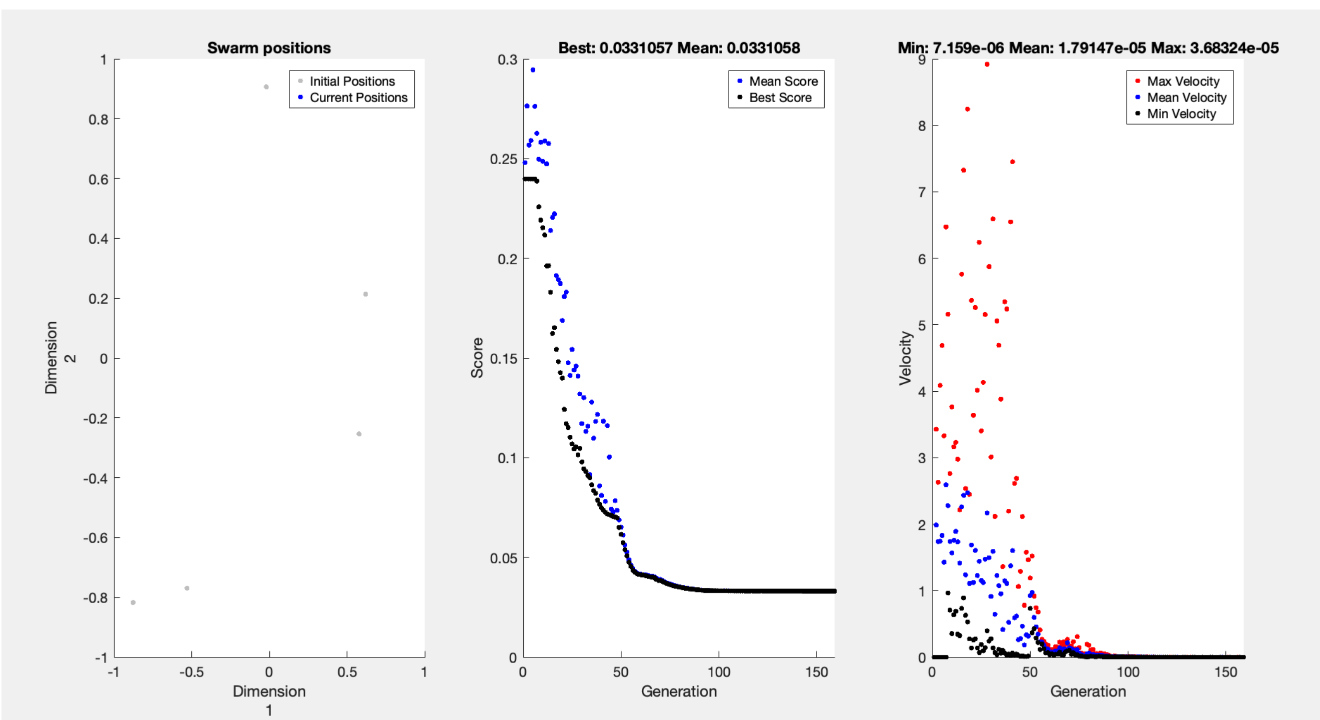
As seen from Lab 1, the neural network has 1 hidden layer and 2 inputs. This constitutes 6 weight values. When we also include the bias, we get 3 additional weight values for a total of 9 which is the dimension of our output.

Plot 3: Include the output from plot_xor for the best solution you were able to Locate.

Best solution with parameters - w=0.5000, c1=1.7500, c2=1.7500,

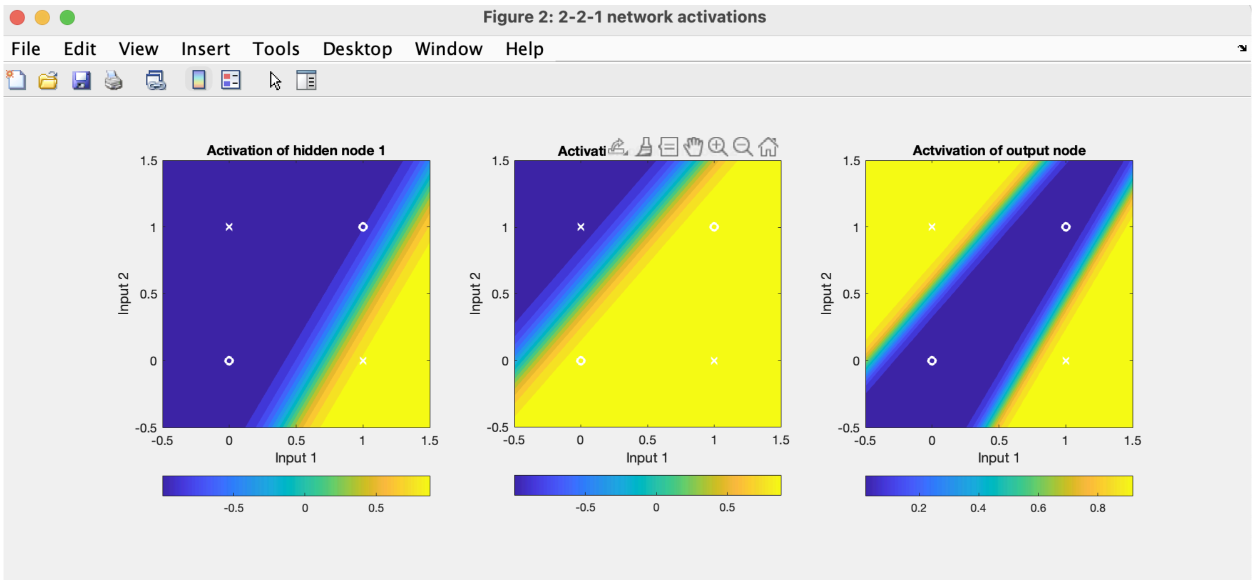


(Try with a bigger population size)



Question 22: How does the plot of the PSO solution compare with the plots you got in lab 1? Why might these plots be different?

Lab 1 plot -



Possible reasons why PSO plots are blurrier -

- PSO explores a wider range of solutions before converging, potentially leading to noisier decision boundaries.
- Gradient descent, with its step-wise weight updates, refines a single solution more precisely.
- PSO initializes a population of solutions and optimizes collectively.
- Gradient descent starts from a single initialization and updates iteratively.
- Random initialization can cause results to be varied each time

Question 23: How well did this lab help to understand concepts from the lectures? Is there a specific concept that the lab has helped to clarify for you? Is there a concept from the relevant lectures that should have been covered more?

So so. Helped for GA, PSO not so much.