UNIVERSITY OF LONDON

INTERNATIONAL PROGRAMMES

BSc Computer Science and Related Subjects

CM3020 Artificial Intelligence

Mid Term Assignment Part B Report

Author: LEE YONG HUA

Student Number: 10246454

Date of Submission: 30th June 2025

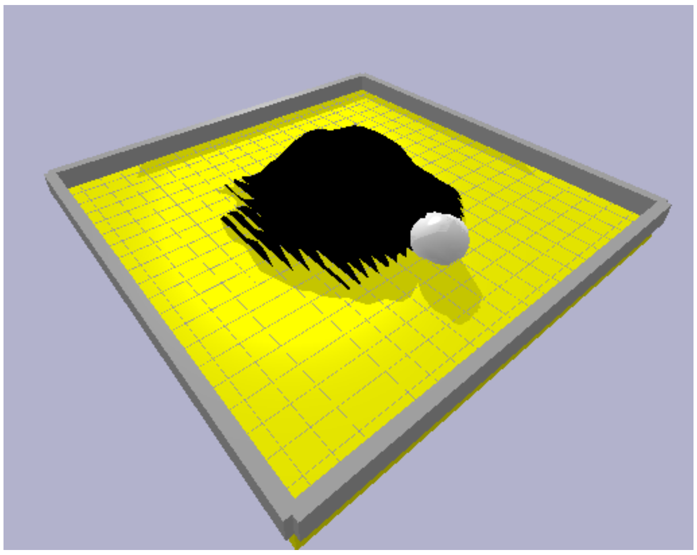
Part B Report

Introduction

In this project the main goal is to build on top of the codebase of Genetic Algorithm(GA) case to adapt the evolutionary algorithms so that creature evolve to complete a new task of climbing to the top of the mountain rather than traverse on the bottom of the mountain. We leverage the custom PyBullet environment (cw_envt.py) that renders a sandbox with procedurally generated mountain. By integrating our GA code into this environment and redefining fitness to reward vertical elevation, we investigate how different hyperparameter settings such as population size, mutation rate, elite fraction, and generations of the mountain climbing setup.

Objective

The main objective of the project is to extend the existing GA creature framework, rather than wandering a flat arena. The "Creature" learn how to climb a procedurally generated mountain in a Pybullet sandbox. To achieve this, I integrated the previously developed GA modules such as population, genome, and creature into a new "cw_envt_ga.py" as shown in the diagram below.

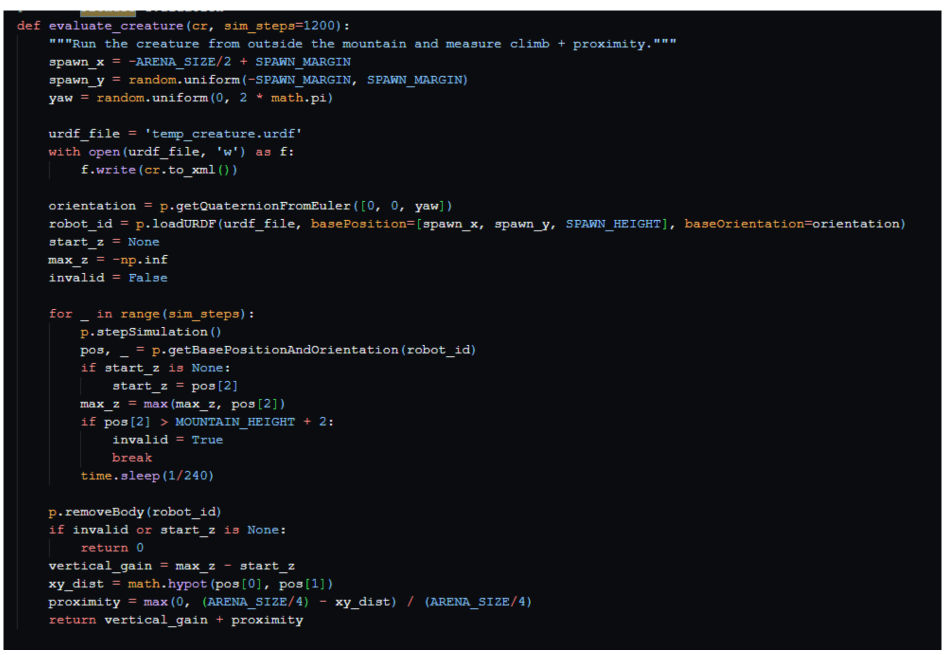


Original “distance_traveled” fitness function with a height-based metrics that rewards maximal vertical elevation and rigorously evaluate how different evolutionary parameters affect both climb efficiency and convergence speed.

Fitness Overview

In this project, fitness is entirely focused on vertical evaluation to replace original “distance_travlled” function. Each creature stimulated run collects the 3D position of its body link at every timestep. The core idea is to scan through this recorded position, extract the z-coordinate which is the height of each link at every movement. The main purpose of fitness evaluation is to determine the maximum elevation reached during the evaluation window of the simulation.

According to the fitness evaluation function “evaluate_creature” shown below, each creature in the population is first initiated outside the mountain and simulated for a fixed number of frames, and return a scalar score composed of its vertical gain.



```python
def evaluate_creature(cr, sim_steps=1200):
    """Run the creature from outside the mountain and measure climb + proximity."""
    spawn_x = -ARENA_SIZE/2 + SPAWN_MARGIN
    spawn_y = random.uniform(-SPAWN_MARGIN, SPAWN_MARGIN)
    yaw = random.uniform(0, 2 * math.pi)

    urdf_file = 'temp_creature.urdf'
    with open(urdf_file, 'w') as f:
        f.write(cr.to_xml())

    orientation = p.getQuaternionFromEuler([0, 0, yaw])
    robot_id = p.loadURDF(urdf_file, basePosition=[spawn_x, spawn_y, SPAWN_HEIGHT], baseOrientation=orientation)
    start_z = None
    max_z = -np.inf
    invalid = False

    for _ in range(sim_steps):
        p.stepSimulation()
        pos, _ = p.getBasePositionAndOrientation(robot_id)
        if start_z is None:
            start_z = pos[2]
        max_z = max(max_z, pos[2])
        if pos[2] > MOUNTAIN_HEIGHT + 2:
            invalid = True
            break
        time.sleep(1/240)

    p.removeBody(robot_id)
    if invalid or start_z is None:
        return 0
    vertical_gain = max_z - start_z
    xy_dist = math.hypot(pos[0], pos[1])
    proximity = max(0, (ARENA_SIZE/4) - xy_dist) / (ARENA_SIZE/4)
    return vertical_gain + proximity
```

During the simulation, if the creature remaining near the mountain’s centre, the fitness will return 0 and the agent will punished by “fly too high”. During each generation, we call “evaluate_creature(cr, sim_steps)” for every cr in the population to assign “cr.fitness”. Once all creatures have been evaluated, those fitness value will report two key metrics which is "average fitness", the mean fitness value of all return score and the "best fitness", corresponding to the single highest "evaluate_creature" score. By reference on these metrics, each generation provides a clear convergence profile, showing how quickly and effectively different GA settings feedback on the mountain climbing performance.

Experiment Protocol

To evaluate how GA settings impact on mountain climbing performance, we employed a controlled factorial design in which we systematically varied four key parameters while keeping all other variables constant. These four key parameters are population size, mutation rate, and elite fractions.

- **Population Size**

  Define the number of creatures in each generation by increase the population size from 20, 50, and 100. The larger populations increase the genetic diversity and exploration of the search space.

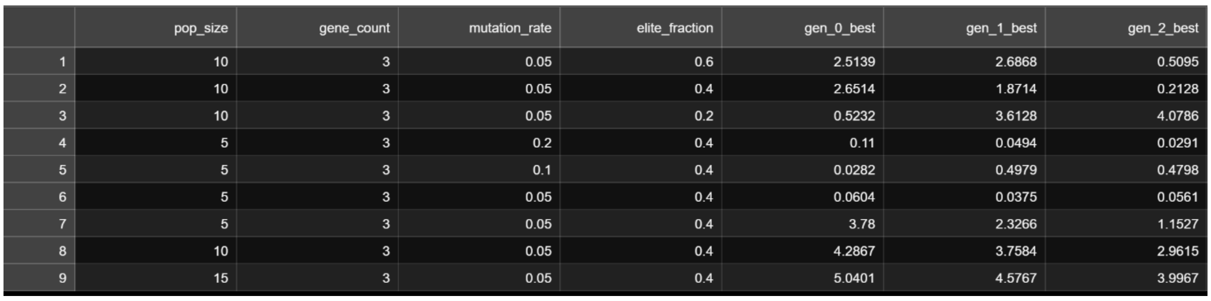
- **Mutation Rate**

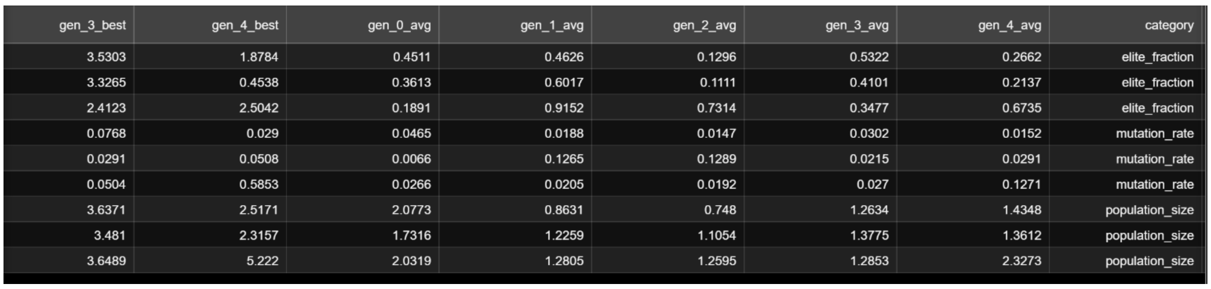
  Probability of the joint in GA will be randomly altered during reproduction. Higher mutation rates promote exploration of different behaviours by increasing the mutation rates from 0.01, 0,05, and 0,1

- **Elite Fraction**

  The proportion of top-performing individuals carried over unchanged into next generation by increasing the elite fraction from 0.05, 0.1, and 0,2. Keeping more top performers in each generation helps retain good traits by can make the population similar, while keeping fewer top performance can let us explore more new traits but will slow down overall improvement.

For each combination of the experiment, we perform a single run of the GA, initializing a fresh population and for each generation evaluate every creature by calling the "evaluate_creature" function. This function will return a fitness value which is the vertical gain of the creature climb to the top of the mountain and we log both the best and average fitness value into a CSV file. Once all the generations complete, the CSV file contains a row per generation with columns shown as the diagram below.





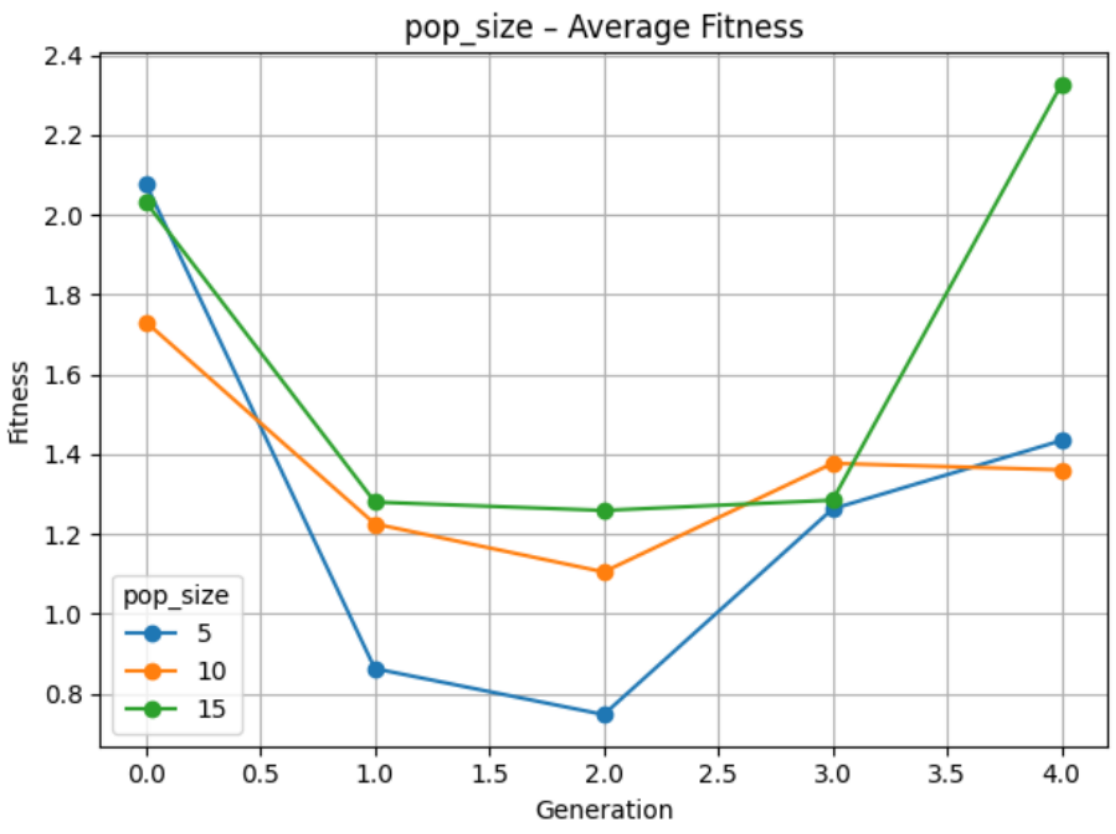
	pop_size	gene_count	mutation_rate	elite_fraction	gen_0_best	gen_1_best	gen_2_best	gen_3_best	gen_4_best	gen_0_avg	gen_1_avg	gen_2_avg	gen_3_avg	gen_4_avg	category
1	10	3	0.05	0.6	2.5139	2.6868	0.5095	3.5303	1.8784	0.4511	0.4626	0.1296	0.5322	0.2662	elite_fraction
2	10	3	0.05	0.4	2.6514	1.8714	0.2128	3.3265	0.4538	0.3613	0.6017	0.1111	0.4101	0.2137	elite_fraction
3	10	3	0.05	0.2	0.5232	3.6128	4.0786	2.4123	2.5042	0.1891	0.9152	0.7314	0.3477	0.6735	elite_fraction
4	5	3	0.2	0.4	0.11	0.0494	0.0291	0.0768	0.029	0.0465	0.0188	0.0147	0.0302	0.0152	mutation_rate
5	5	3	0.1	0.4	0.0282	0.4979	0.4798	0.0291	0.0508	0.0066	0.1265	0.1289	0.0215	0.0291	mutation_rate
6	5	3	0.05	0.4	0.0604	0.0375	0.0561	0.0504	0.5853	0.0266	0.0205	0.0192	0.027	0.1271	mutation_rate
7	5	3	0.05	0.4	3.78	2.3266	1.1527	3.6371	2.5171	2.0773	0.8631	0.748	1.2634	1.4348	population_size
8	10	3	0.05	0.4	4.2867	3.7584	2.9615	3.481	2.3157	1.7316	1.2259	1.1054	1.3775	1.3612	population_size
9	15	3	0.05	0.4	5.0401	4.5767	3.9967	3.6489	5.222	2.0319	1.2805	1.2595	1.2853	2.3273	population_size

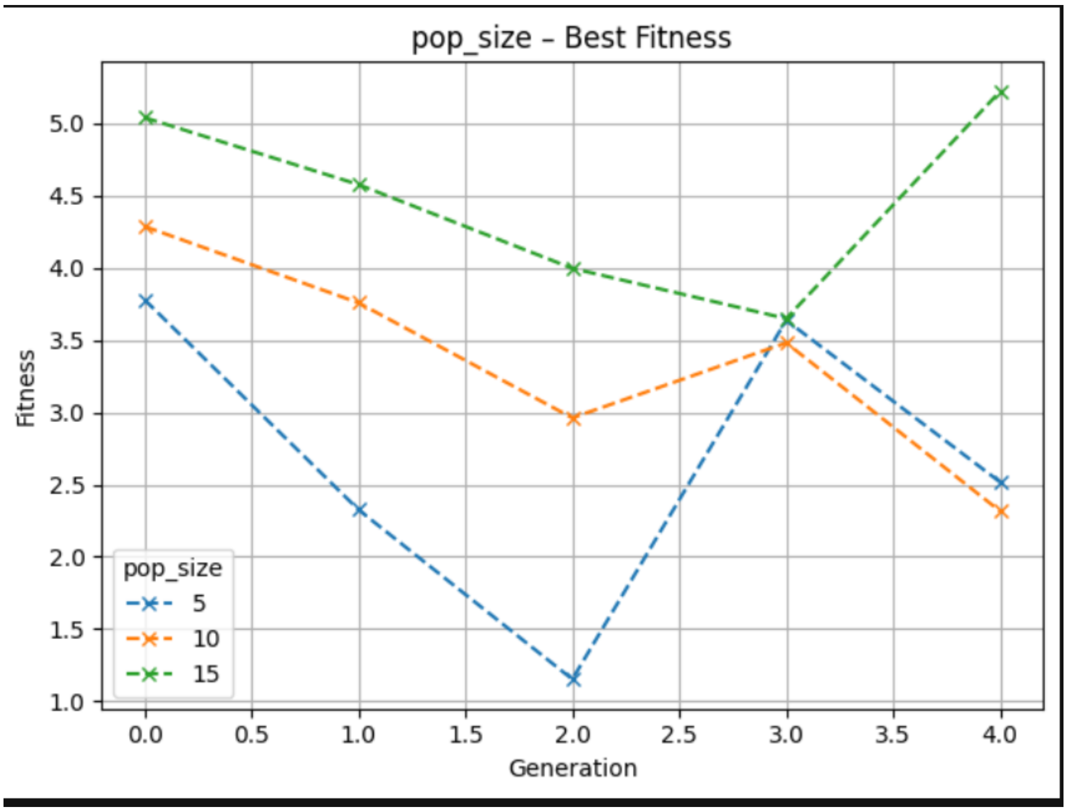
After all simulations run finished, I loaded these CSV files into the analysis environment in jupyter notebook to produce convergence plots of best and average fitness value against generation of each category.

Result and Discussion

The graphs shown below is plotted according to the result collect from the simulation with different key hyperparameters change.

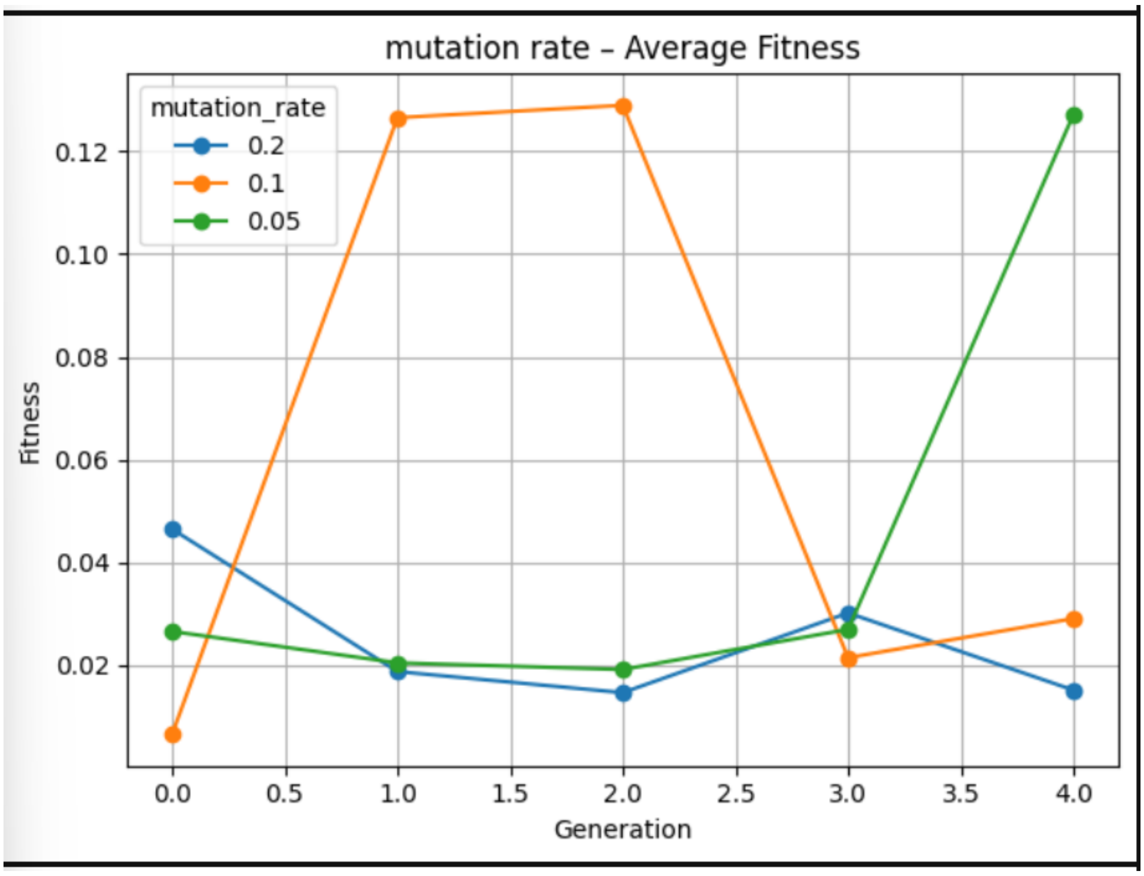
Population

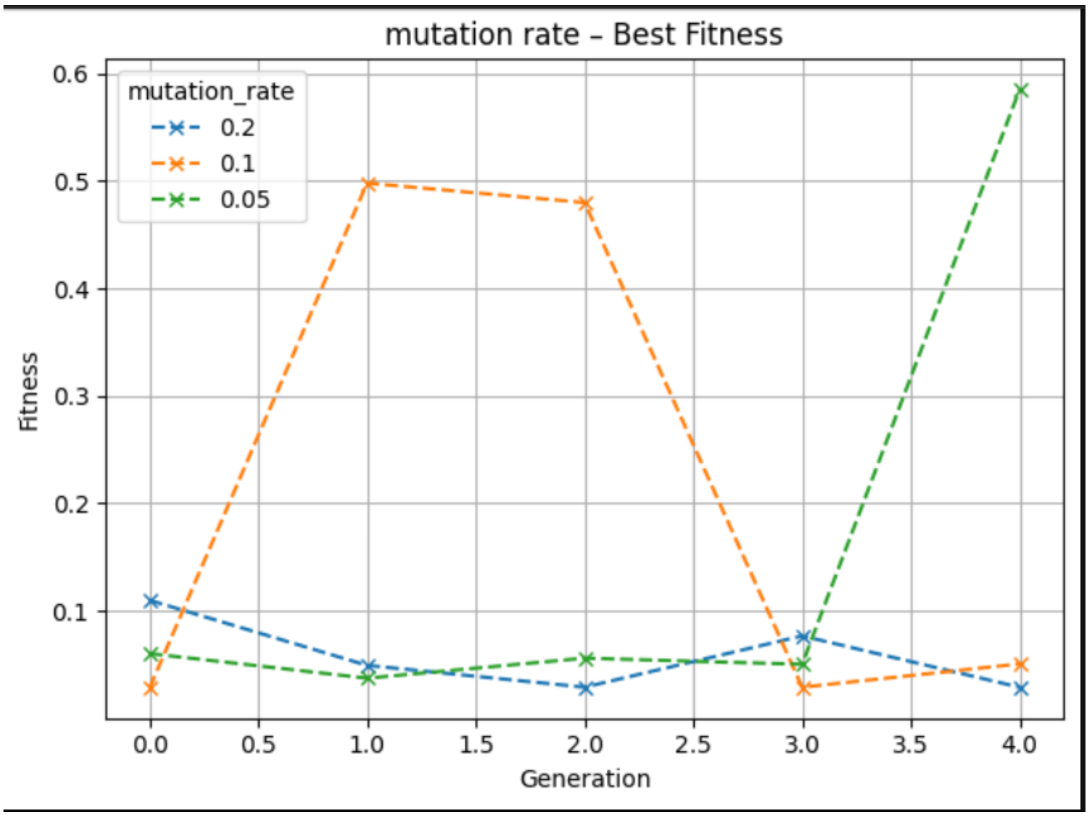

pop_size – Average Fitness
Fitness
Generation
pop_size
5
10
15


pop_size – Best Fitness
Fitness
Generation
pop_size
5
10
15

All three populations being with relatively high in “best” and “average” fitness, but these are random individuals sampled from the search space. As the GA kicks in weaker offspring showing a decreasing value to replace the initial elite performance. Around generation 3, each population size starts to recover and started to drive both average and best curves upward. The largest population of 15 rebounds the fastest and consistently, with its average fitness rising from 1.25 to 2.33 and its best fitness surging past its initial peak. This shows that a bigger gene pool preserves diversity and enabling more robust discovery of effective climbing.

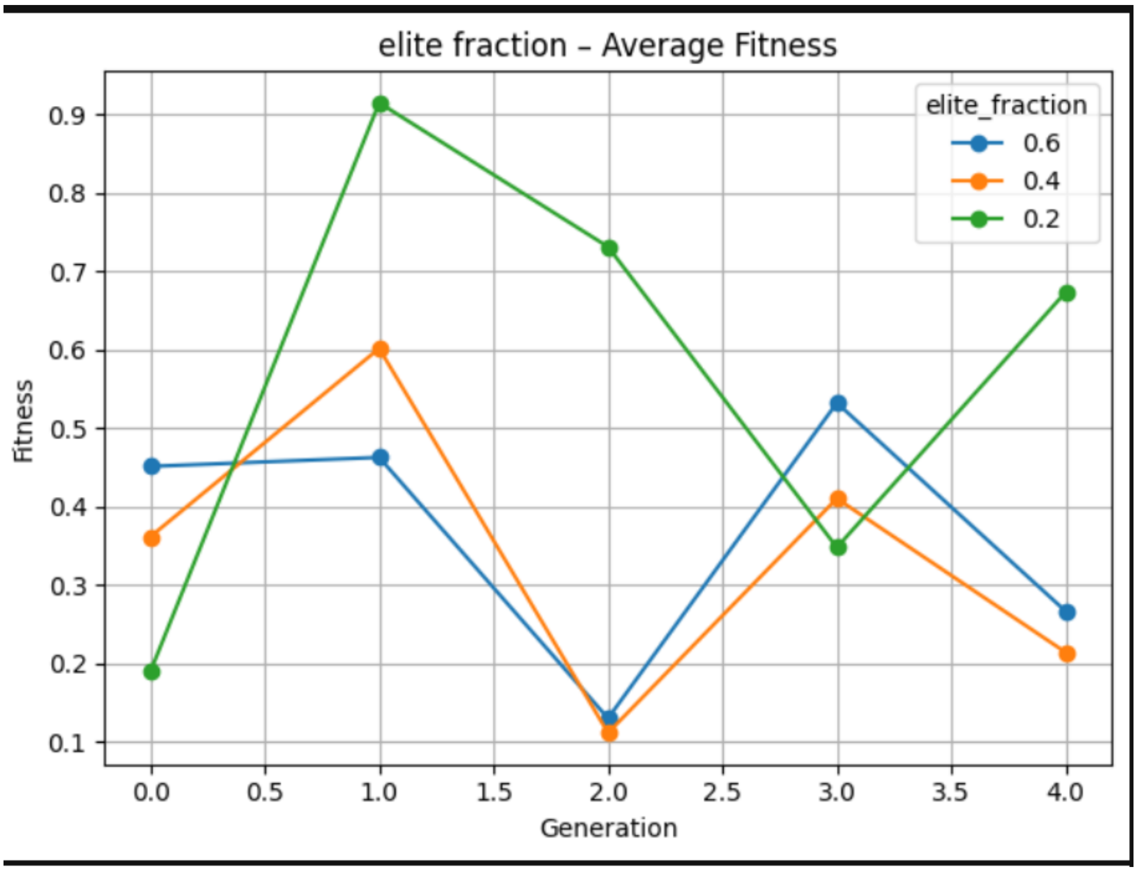
Mutation Rate

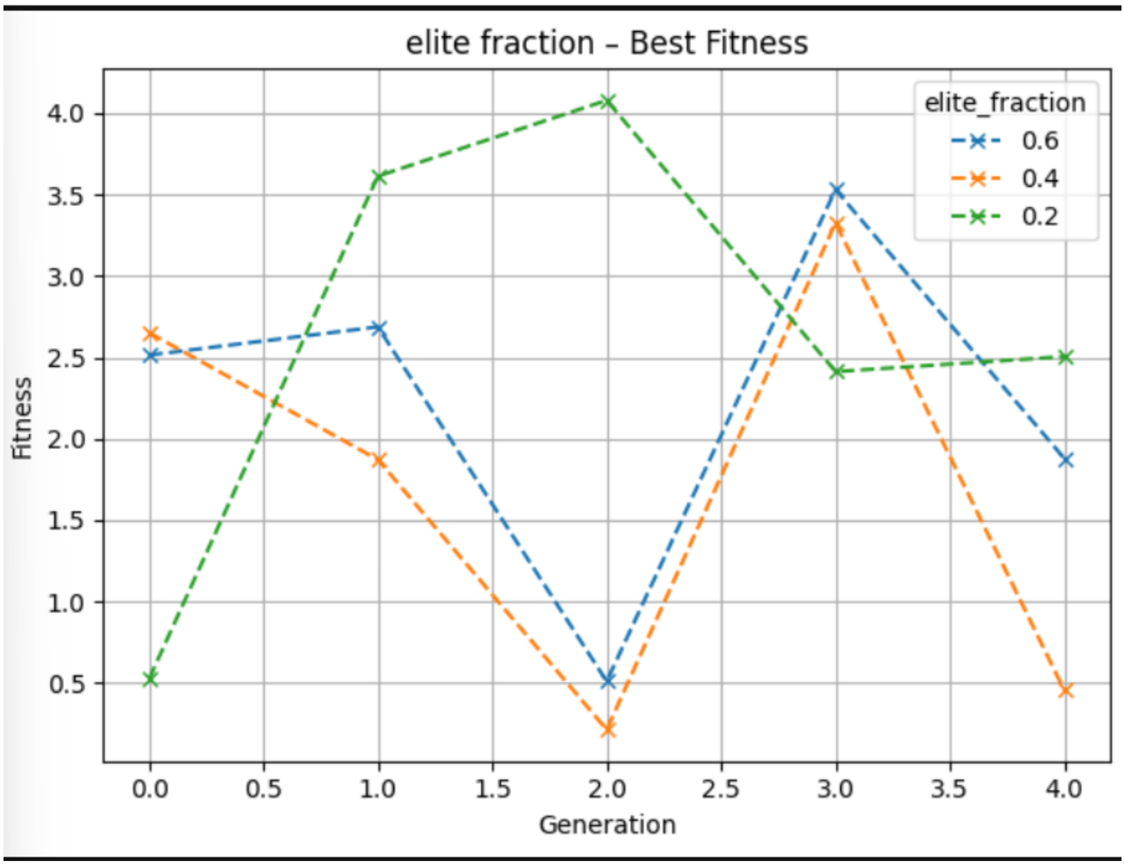




According to these two-mutation rate best and average fitness plotted graph, I observed a high mutation rate of 0.2 injects randomness to affect that both average and best fitness drop quickly and never bounce back to the peak point which indicate overwhelms any strategy refinement. Moderate mutation rate of 0.1 produces an impressive spike in average fitness and best fitness performed in generation 1 and 2 that showing a rapid discovery of climbing behaviours. Besides that, a low mutation rate of 0.05 yields the most reliable progress as both average and best fitness increase slowly but steadily through the first three generations and in generation 4 it increases rapidly and overcome both high mutation settings. Overall, these trends illustrated in the graphs show that lower mutation rate schedule best balances discovery and consolidation to produce the strongest creature climber.

Elite Fraction





According to the elite fraction fitness graph, population starts with the modest baseline fitness but immediately increases with the strong climbing strategies in a low elite fraction of 0.2. Besides that, the high elite fraction of 0.6 preserves top performance but not constant with bouncing performance as best fitness in generation 1 is around 2.5, collapse in generation 2 and rebounding back in generation 3 to around 3.5. Medium elite fraction of 0.4 falls achieve the modest early gain around with a average of 0.6 and best of 1.9 in generation 1, collapsing in generation 2 then rebounding back in generation 3. Overall, using a small elite fraction accelerates climbing performance but make results more volatile, while a larger elite fraction delivers stronger initial breakthrough at the cost of consistency over time.

Conclusion

This project demonstrates that adapting GA creature framework to a mountain climbing task is both straightforward and effective, by redefining fitness as a combination of vertical gain and proximity to complete the task. Variation of population size, mutation rate, and elite fraction revealed clear trends. The larger population and lower mutation rate present a highest and most reliable climbing, moderate mutation can spark rapid at the starting stage of the simulation, but it will have an unstable breakthrough. Small elite fractions drive bold innovation at the expense of consistency.

In the future, GA should incorporate multi objective criteria such as structure stability and energy efficiency to yield a more stable climber. Surrogate modelling should be introduced to mitigate the computational expense as this can accelerate convergence by prioritizing regions of the search space. Lastly, GA promise to advance evolutionary robotics towards a more adaptive, resilient agents capable to tacking an wider range of real-world climbing and exploring challenges.