University of Konstanz Winter Semester, 2025
Cyber-physical Systems Group Assignment 3
Group member name(s): Benedict Finsterwalder, Florian Shabani
Group member UID(s): 1564941, 1587539

INF-16580: Evolutionary Robotics

Introduction

In this task we created a small simulation of a basic hill climbing algorithm. The goal was to reach the exact target string "charles darwin was always seasick". We started from a random string of the same length made only of lowercase letters and spaces. In every generation one random position was changed to a random new character. After this change we checked the fitness, which was the number of correct characters in the correct positions. If the fitness did not decrease, we accepted the new string. Otherwise we returned to the previous one. The process ended as soon as the target string was reached.

Solution Method

We implemented the algorithm in a Jupyter Notebook which is uploaded in our Github. First we generate a random starting string. Then we change exactly one character at a random position and compare the fitness before and after the change. If the new string has equal or higher fitness, we kept it and continued with the next generation. The program prints the current string in each generation such that the progress is easy to follow.

The task also asked why this problem can be viewed as a good-natured optimization problem. This is reasonable because each position in the string can be improved independently. A mutation that matches the correct character always increases the fitness, and a wrong one does not get accepted. This means the search always moves uphill and never loses progress. The landscape therefore has a simple structure without traps.

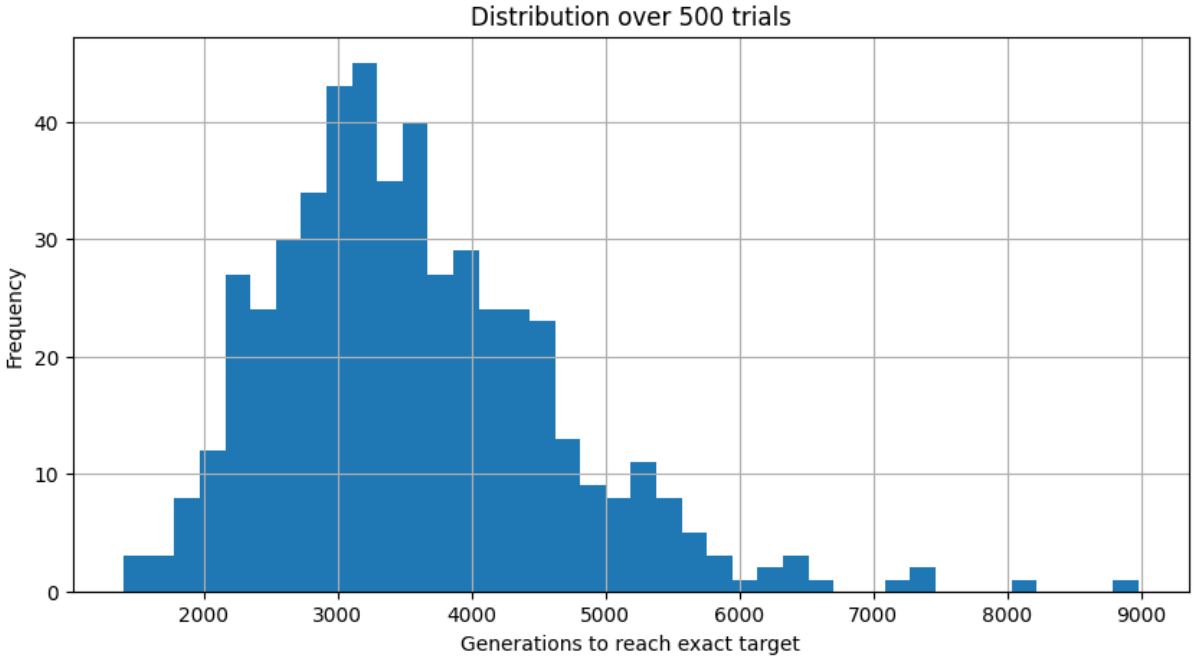
To test this, we ran many independent experiments. A separate script repeated the hill climbing procedure several hundred times. Each run recorded how many generations were needed to reach the exact target. We also used multiprocessing to make the runs faster. After all trials finished we calculated the mean, median, and standard deviation. A histogram was created to show how the number of generations was distributed.

Results [1.75 pages]

Across all experiments the algorithm always reached the target string. The number of generations varied from run to run, but the values stayed in a similar range. The fitness increased step by step without ever going down because worse mutations were rejected. The summary statistics showed a stable mean and median, and the standard deviation was moderate.

Empirical results over 500 trials:
mean generations = 3558.504
median generations = 3396.0
std deviation = 1059.544
min / max = 1400 / 8979

The histogram of the results had one main peak and a longer tail toward higher values. This shape makes sense because some positions needed more attempts before the correct character appeared. Overall the results supported the idea that the task has a smooth fitness landscape with no misleading paths.



Conclusion

The hill climbing algorithm worked well and always reached the correct string. The experimental results matched the theoretical explanation of why this problem is easy to optimize. The fitness increased steadily, and the simple mutation rule was enough to guide the search toward the final solution. The tests showed that even a very basic evolutionary method can solve this type of problem efficiently.