Trackmania AI

Saadan Ashraf

29223

Headings

Introduction

Code links

Track details

EA Implementation

PSO Implementation

Analysis

Possible Improvements

Introduction

By using a popular game “Trackmania”, I have created a program that uses algorithms from our course and applies them (Evolutionary Algorithm & Particle Swarm Optimization)

The AI’s decision making is based on real-time screen capture analysis and heuristics

The intention is to optimize the algorithm such that the car in the game can finish the track as efficiently as possible

Efficiently means minimum time and minimum resets (Reset is when the car is in ‘stuck’ state for several seconds)

Code Links

Basic Code (Without Optimization, works)

https://drive.google.com/file/d/1KfimzyZxDQZOsaZILve4r0CMwjvg3p0H/view?usp=drive_link

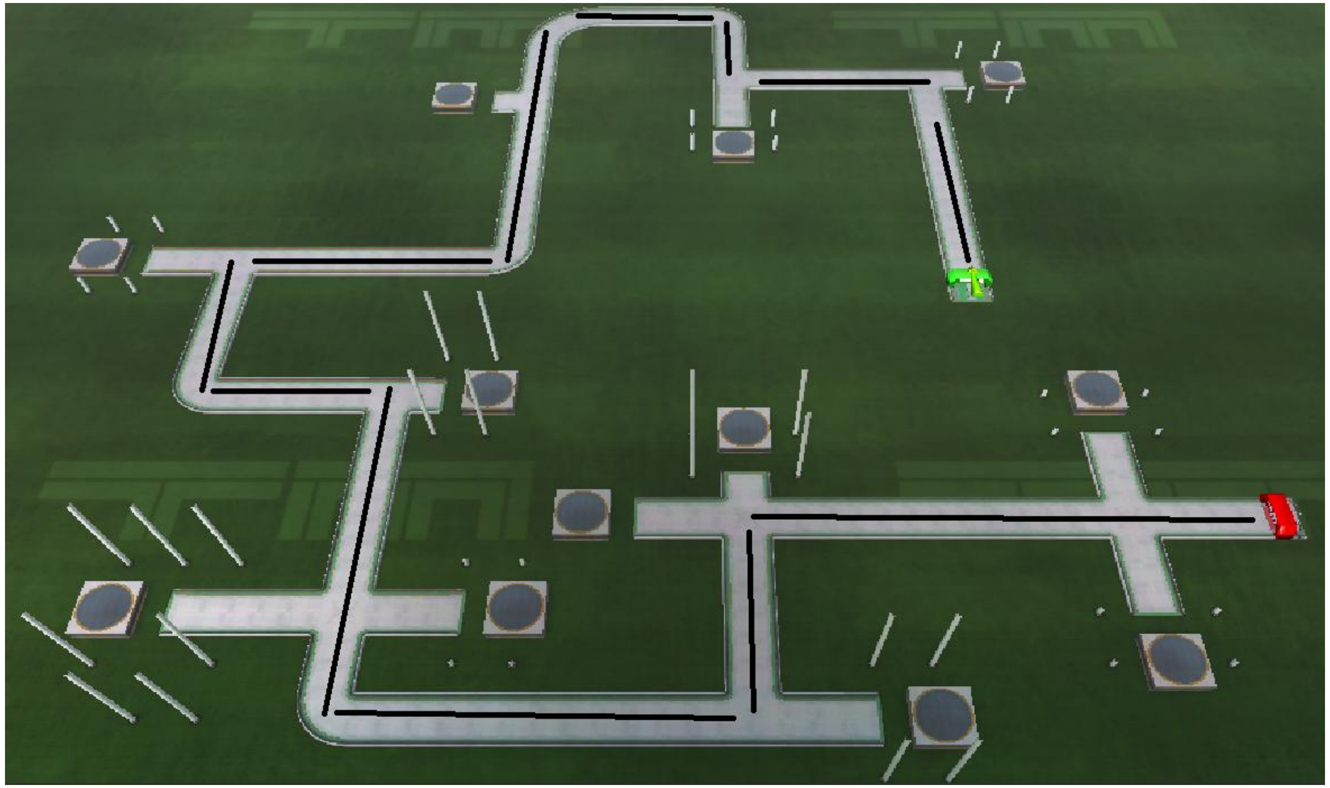
Final Code with EA

https://drive.google.com/file/d/17TRHFXDl4RlHF4Zj4yRtnlCz9e05HNii/view?usp=drive_link

Final Code with PSO

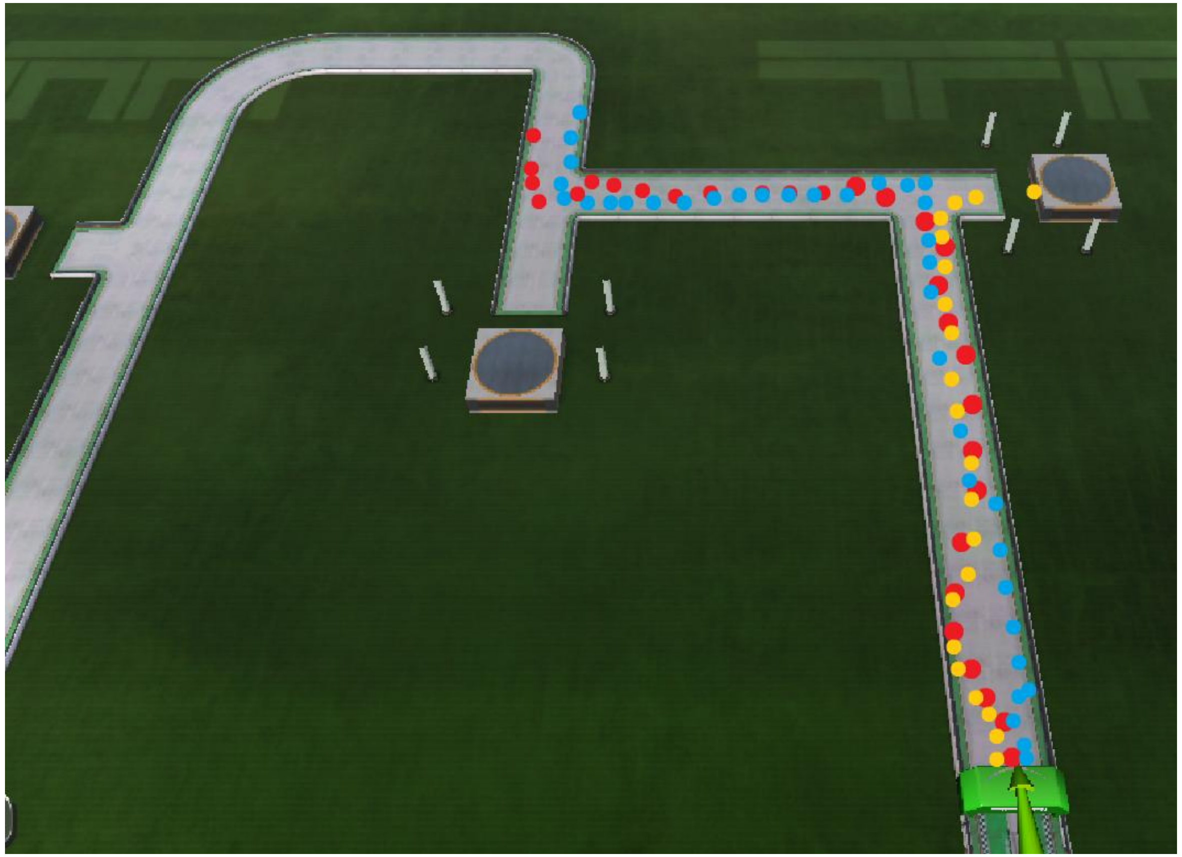
https://drive.google.com/file/d/1WbT03_plG3o3ALuJhRwTLAikeDT-vdE1/view?usp=drive_link

Track details



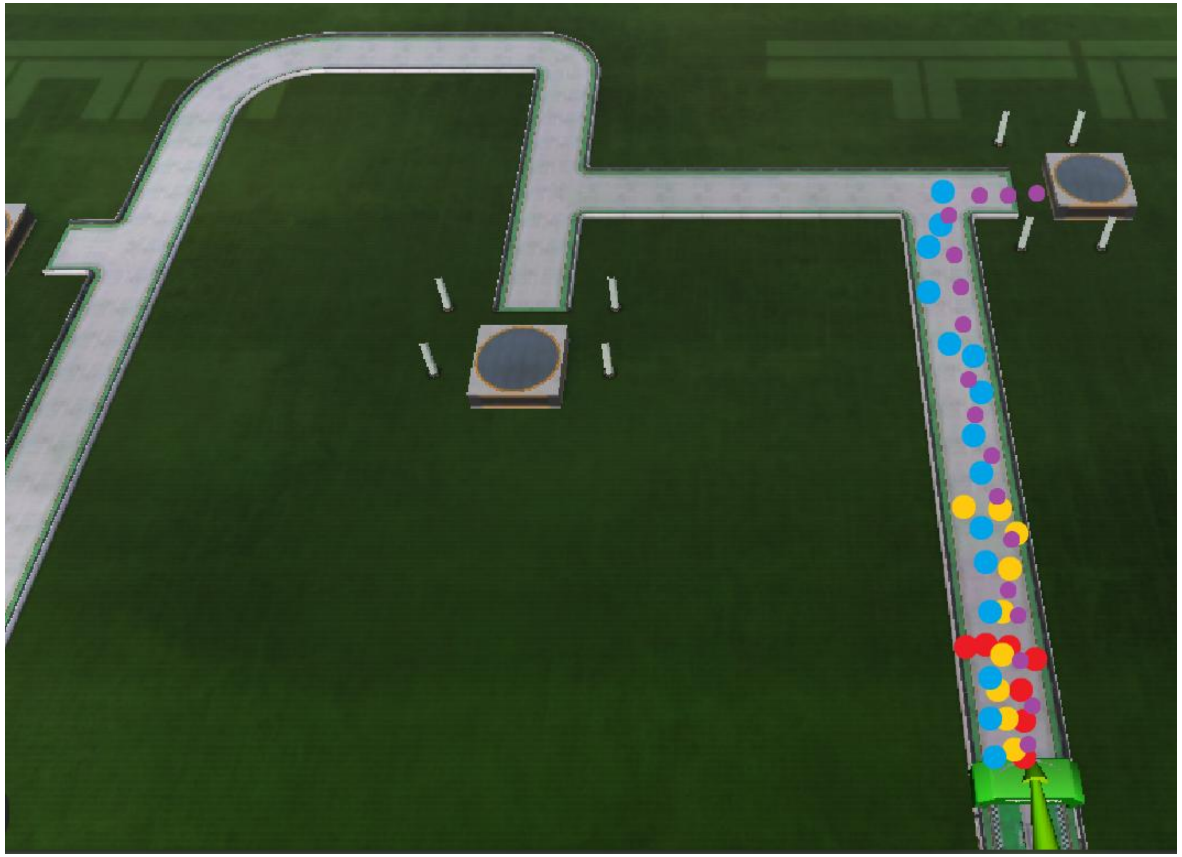
- The track is inspired by a maze to implement a heuristic approach for decision making of AI
- There are a total of 21 turns, 8 lead to the finish line
- The track is a total of 2165 meters
- Every wrong turn has a blockage which causes the car to be in ‘stuck’ state and causes a reset (car set to initial position at starting line)
- The code takes screenshot of the game state, and the AI makes decision based on the current game state

EA Implementation



- Evolutionary Algorithm was able to go through the first stretch without halting / being stuck
- It can be seen for the first 3 runs, 1/3 times it took a wrong turn and causes a reset
- 2/3 times it took the correct first turn and went ahead
- In all cases, the car crashes to the side walls but continues to go ahead by increasing distance from the wall

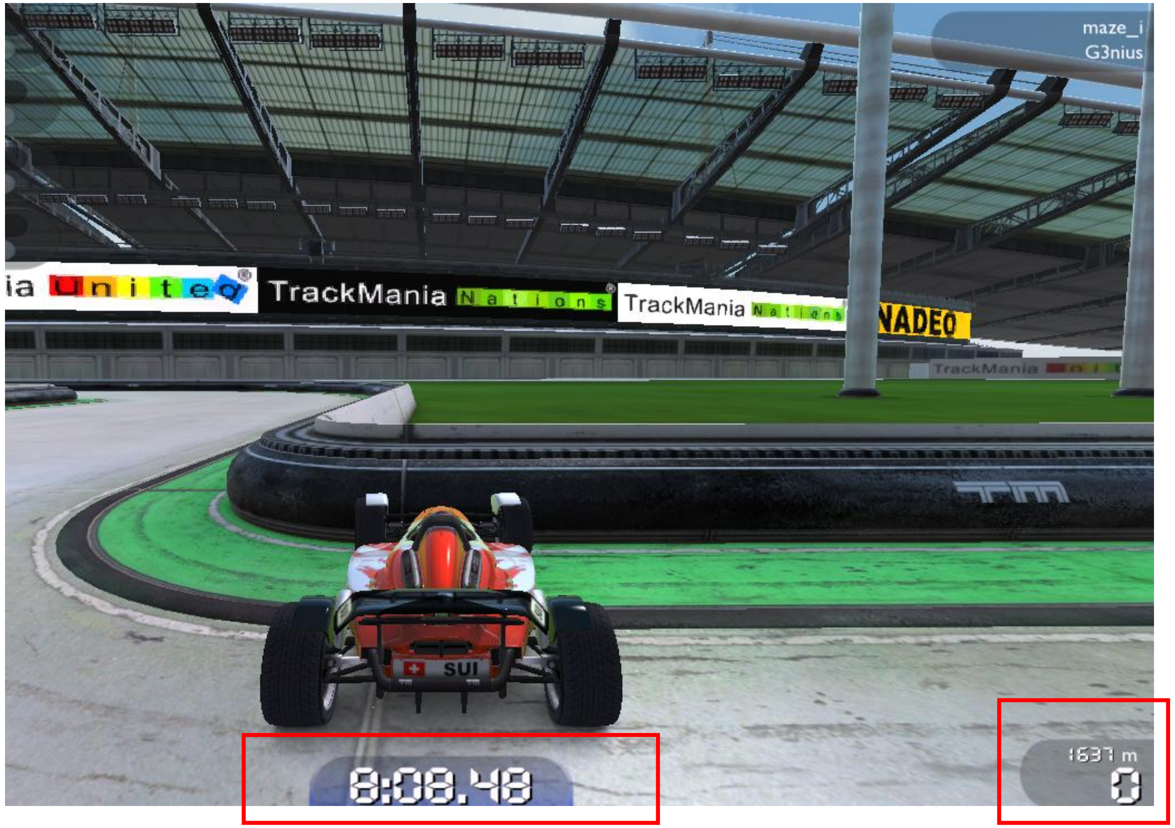
PSO Implementation



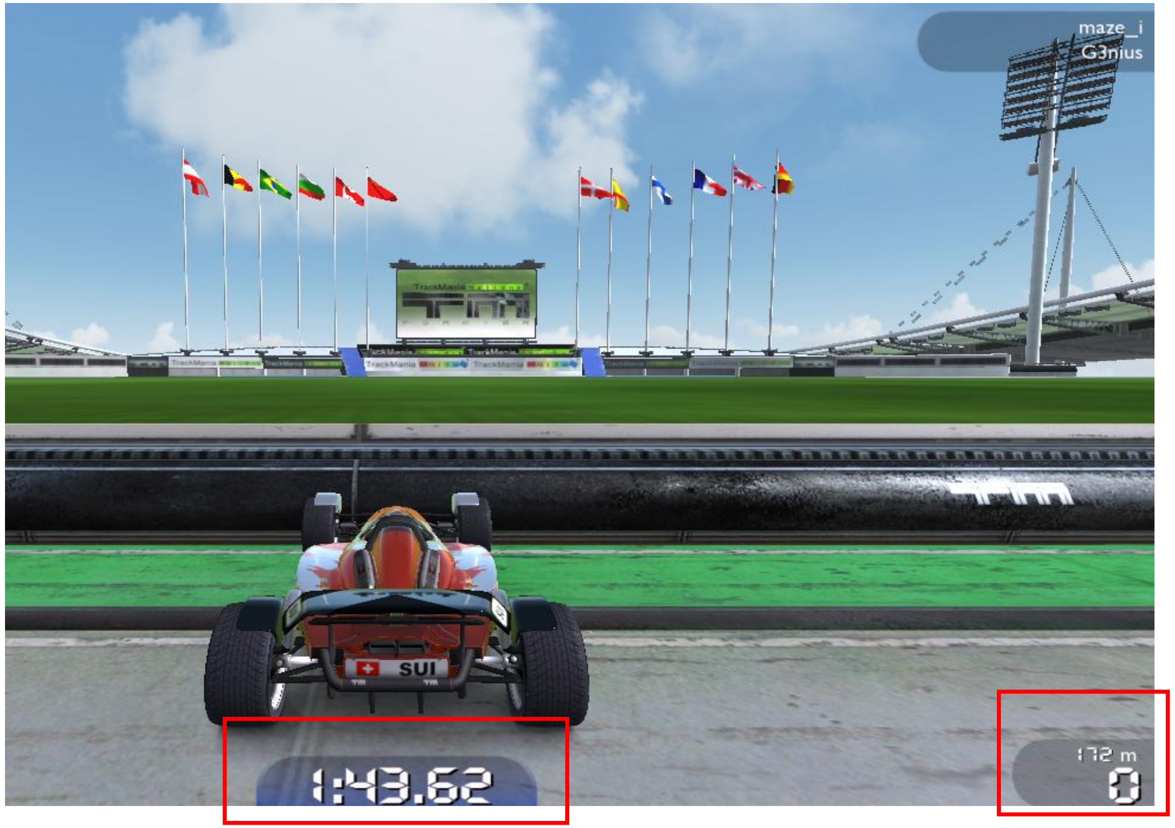
- PSO was inefficient compared to EA
- In multiple tries, the car was unable to reach the first turn as it got stuck as highlighted by the red and yellow path
- After a few resets, the car was able to reach the first turn but got stuck straight ahead
- The car crashed to side walls and took more time and attempts to recover than EA

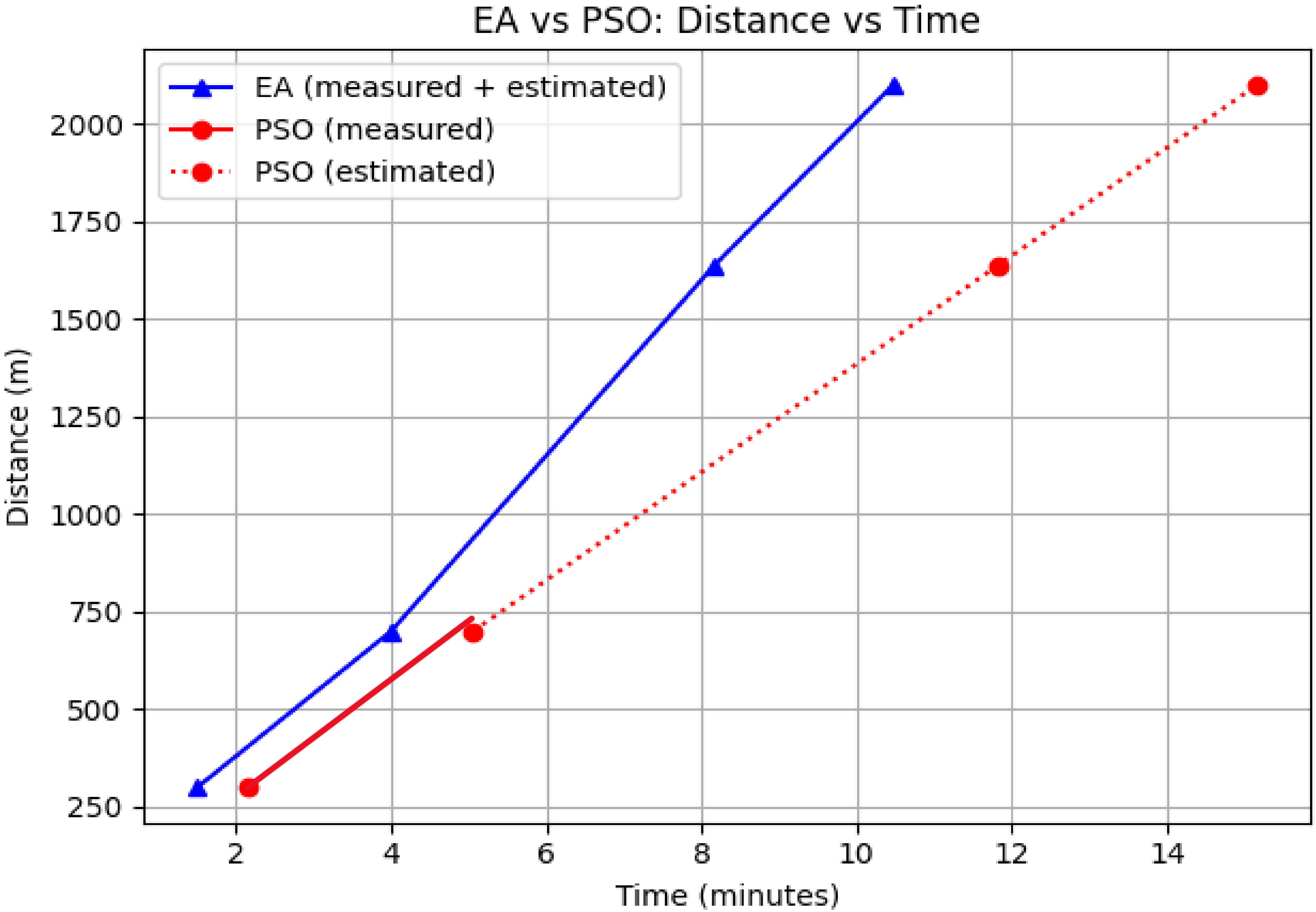
Analysis

EA



PSO




EA vs PSO: Distance vs Time
EA (measured + estimated)
PSO (measured)
PSO (estimated)
Distance (m)
2000
1750
1500
1250
1000
750
500
250
Time (minutes)
2
4
6
8
10
12
14

Possible Improvements

With an addition of an API, I would be able to optimize the code better as the feedback will be live in terms of values (Speed, Coordinates, etc.) instead of screenshot decision-making as PSO and EA are value dependent

Usage of Reinforcement Learning with a memory-enabled Neural Network would allow fast adaptation of algorithm and faster learning than a penalty-based algorithm as it requires multiple tries to converge to an ideal balance between reward and penalty

A Hybrid approach (PSO + EA) can be used for better results, but it will need a lot of trial and error as mutation and computation of swarm will complicate values often

Instead of resetting the game state on ‘stuck’, we can add the option to input ‘back’ key to reverse or backtrack the moves taken by storing all the inputs of the run

Use 2D FFT (Fast Fourier Transform) to capture the screen and visualise it better (grayscale)

Thank you