Genetic Cars Simulation

By Jon

My Project

- My project is to create a Evolutionary Algorithm for a pre-existing framework.
- The pre-existing framework is Javescript simulation run on a internet browser. Example: https://rednuht.org/genetic_cars_2/

(“Reminder!, do you need it?”) Evolutionary Algorithms Overview

- An Evolutionary algorithm in general optimises a given solution to a problem using a set criteria such as time complexity.
- The algorithm itself can have several different components but overall the main ones are Selection, Crossover and Mutation operators.
  - Selection chooses the parents for the new generation
  - Crossover uses the parents chosen through selection to combine the data into new children. Such as taking half from one parent and half from another.
  - Mutation is an operator where a random set of data within the child is changed to a random number.

Selection operators Implementation

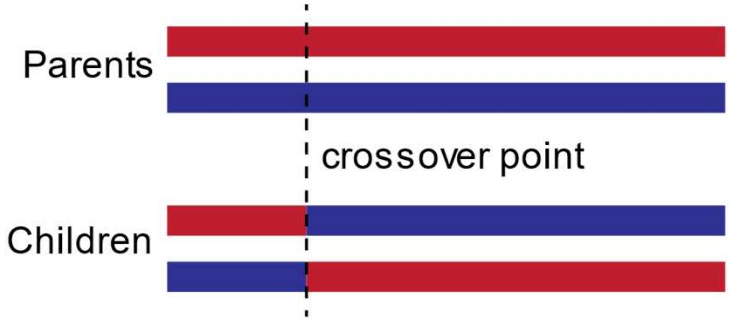
- Roulette-wheel selection (fitness proportionate), A cars fitness determines the chance of a car being chosen such as the fittest car having the highest percentage chance of being chosen compared to the weakest car.
- Tournament selection – A set of cars are randomly chosen from the population and put into another array, the fittest car in this new array is chosen as a parent.
- Uniform random selection – All cars in the population has a equal chance of being chosen.
- Tournament selection getting the strongest and weakest – Like tournament but 1 parent is the weakest car in the population and the other is the strongest.

Crossover operators implementation

- One-point crossover – only 1 point is chosen in the data
- Two-point crossover

Me: “Only two sorry”

Crossover example(“I did not make it!”)



Ref: R0oland (2013). OnePointCrossover. [image] Available at: https://commons.wikimedia.org/wiki/File:OnePointCrossover.svg [Accessed 6 Mar. 2019].

Mutation operators implementation

- 1 Mutation – only 1 mutation happens (“obvious I know”)
- Multi-mutation – Multiple mutations effect a car depending on the criteria, the criteria what I’m using is that the bottom half of the population has 3 mutations instead of the normal 1.
- Me: “Sorry still, again two operators”

Elitism implementation(what is it?)

- The car/s with the highest fitness in the population are passed through to the next generation without any mutation altering the cars data, It is also possible for Elitism to use other criteria to pass a car to the next generation.

Cluster Re-scoring

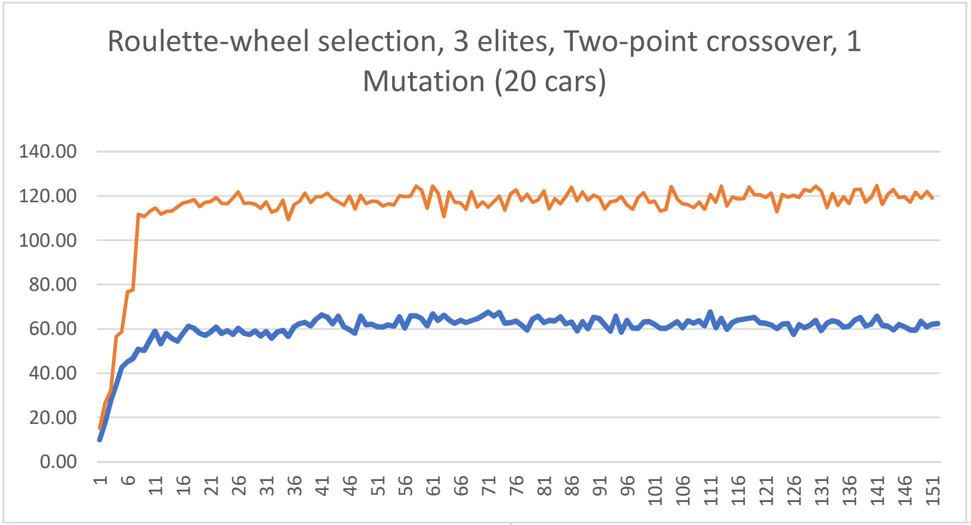
- Using clustering to change the fitness of the cars, the clustering classifies a car into clusters and using version of K-nearest neighbour to get the surrounding data points and averaging the surrounding data points fitness into the classified car’s fitness.

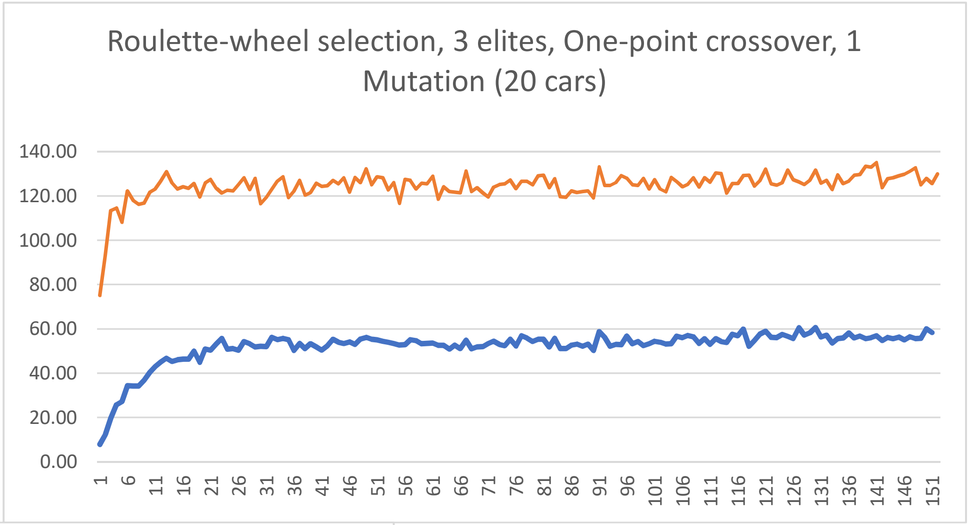
- “This would be better explained if I had images and probably a brief explanation of clustering in general…. Sorry”

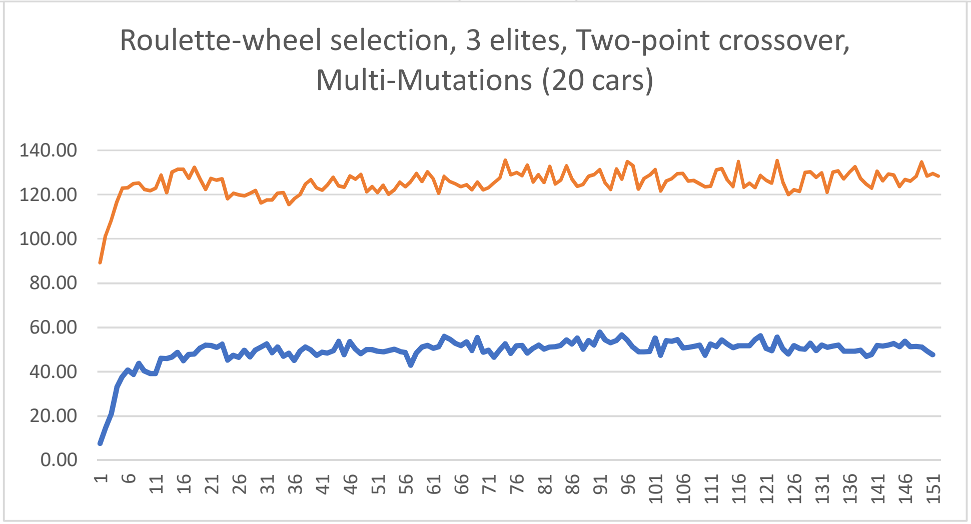
What I’m currently doing?

- Currently I’m creating test data running the simulation 10 times for 1 iteration of the Evolutionary algorithm which will be put into graphs to show the performance of the different Evolutionary algorithms operators.
- Running the tests 10 times is only preliminary but I plan on adding an extra 20 runs of the simulation to add data to back up the existing data.
- I’ve currently got 26 sets of data which takes an hour each to run.

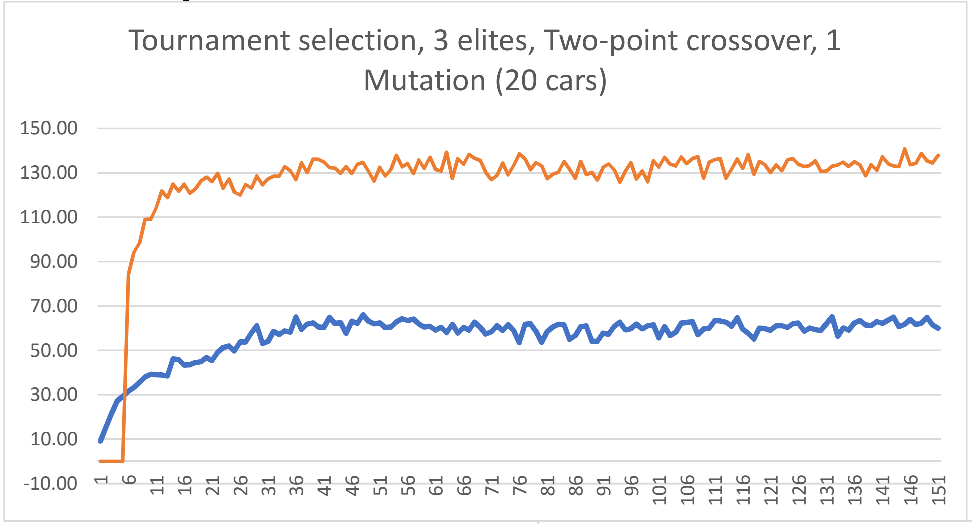
Examples Roulette Selection

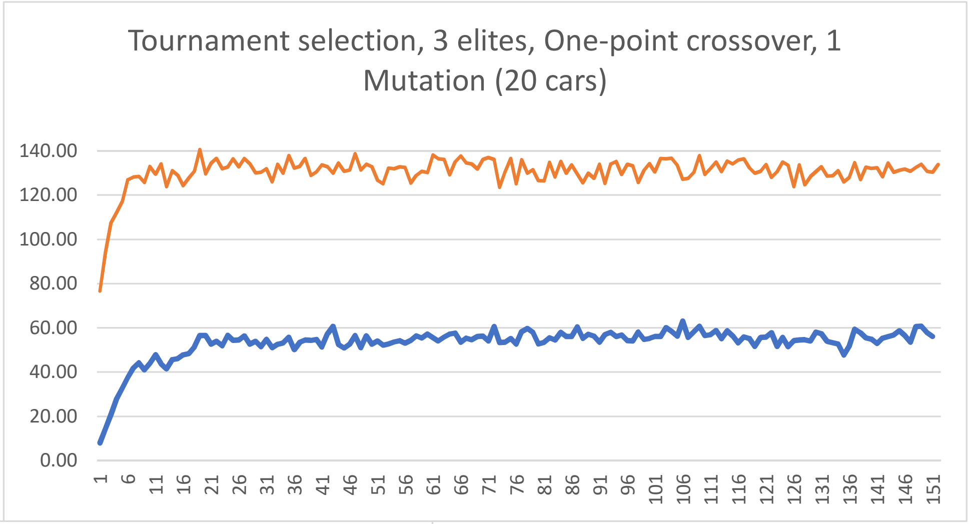


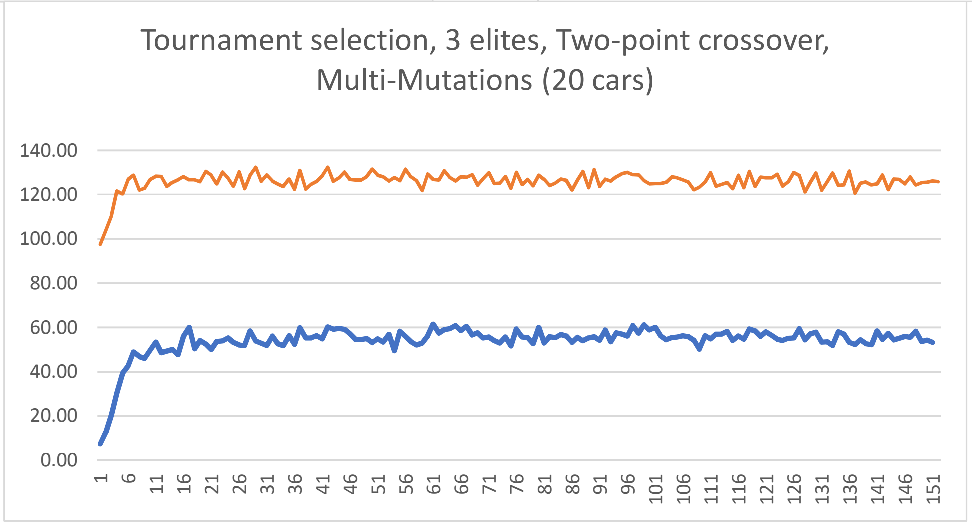




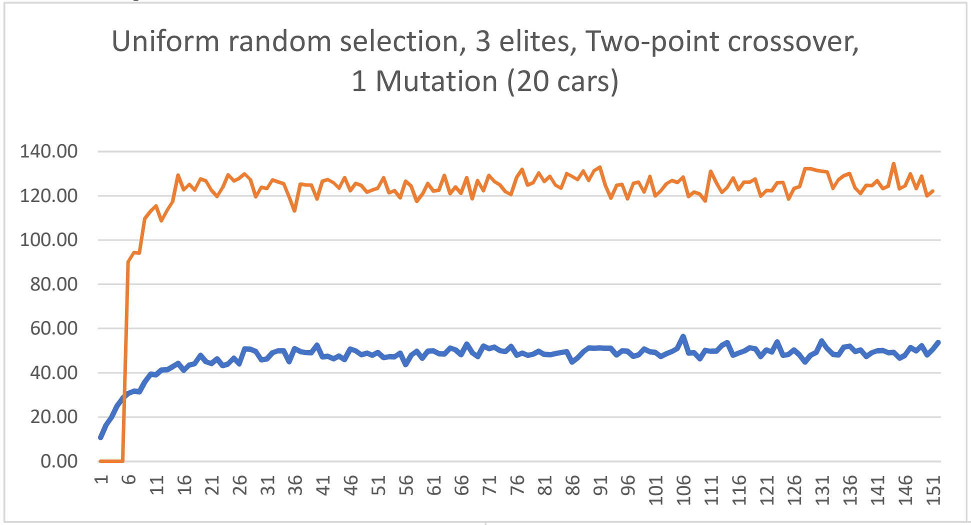
Examples Tournament Selection

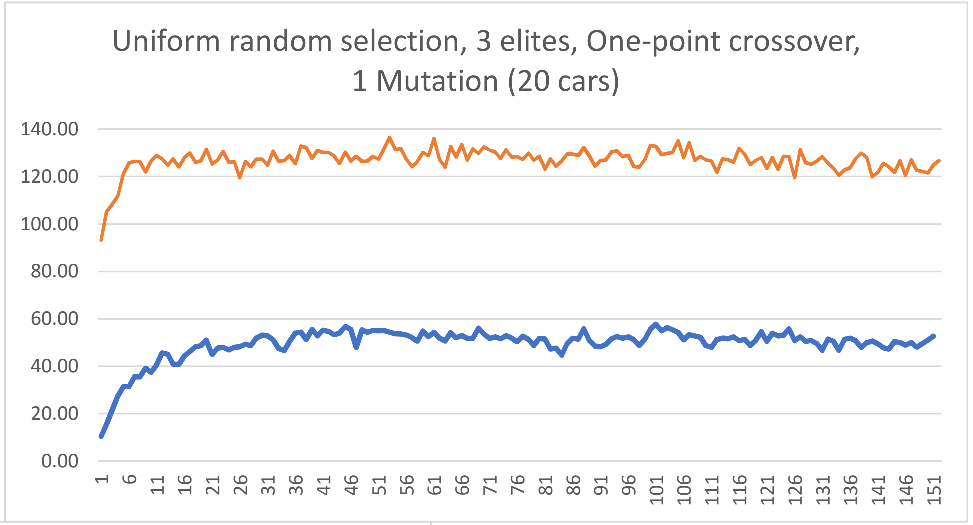


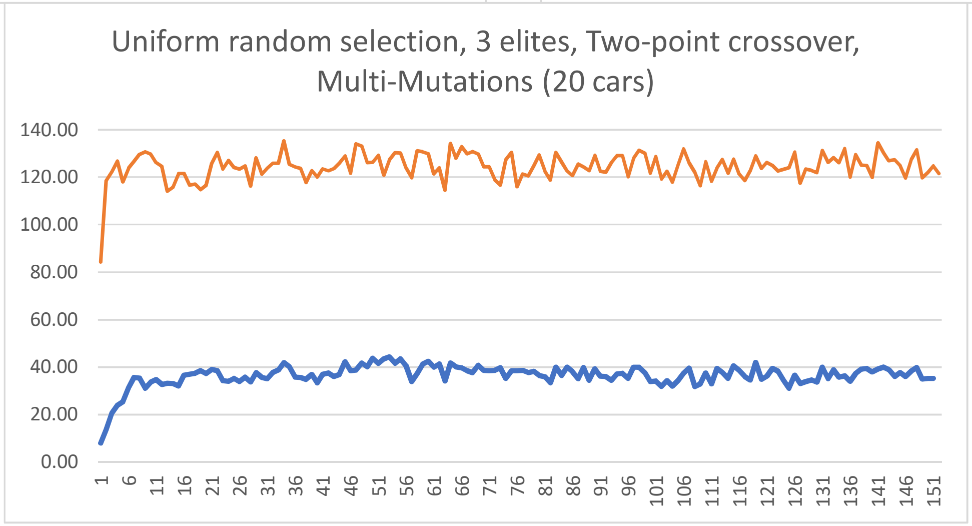




Examples Uniform Random Selection


Uniform random selection, 3 elites, Two-point crossover, 1 Mutation (20 cars)


Uniform random selection, 3 elites, One-point crossover, 1 Mutation (20 cars)


Uniform random selection, 3 elites, Two-point crossover, Multi-Mutations (20 cars)