

Assignment: Using Simulated Annealing and Genetic Evolution to optimize pitstop strategy for multiple F1 cars

Barry Denby

November 30, 2020

1 Introduction

NOTE: read the whole assignment brief first before implementing it contains very important information

In this assignment you will be tasked with doing four things:

- Create a simulation of F1 tyres taking into account tyre degradation on each lap (we will refer to tyre degradation from here on in as deg for short) that takes into account the fuel on board the car and how hard the car works the tyres
- A mapping onto the simanneal library that will use Simulated Annealing algorithm to determine the best strategy
- A mapping onto the deap library that will use genetic evolution to determine the best strategy
- Using both mappings to determine the best strategy in a number of situations for three cars at a time

In terms of effort the vast majority of your time will be spent on the first part in creating the simulation. The mappings to both libraries will not take long, and the experimentation to find the best strategy will be fairly quick.

It is recommended that when you have completed your simulations that you test them with the test cases and graphs given below. You should print out graphs of your own for comparison as it is much easier to verify this visually than to pour over a set of numbers and figure out the general pattern that way. **Remember if you produce a bad simulation your results from numerical optimisation cannot be trusted.**

Read the details of the simulation carefully and if you have any questions do not hesitate to ask. You will need to have full clarity on the simulation in order for implementation to be correct and therefore your numerical optimisation to be correct

2 Simulation Details

The reason why F1 teams run many simulations of this nature is that they need to know the fastest way to complete a race, generally there are two main variables involved how many pitstops should be done over the race and what tyre compounds to use. What we will simulate here will not be to their level of complexity but will roughly simulate what happens.

They generate many strategies to cover many different situations. Teams will generally have three dry tyre compounds available to them on each weekend. For the purposes of our called the Soft, Medium, and Hard. They have the following properties:

- Soft: Most grip, fastest lap time, only lasts a short time
- Medium: in between on all properties between Soft and Hard
- Hard: Least grip, slowest lap time, lasts a long time

Also teams must use at least two different compounds during the race, meaning there is one mandatory pitstop and compound change to be made during the race. However, we are going to simulate a rule tweak here. **We will require that all three compounds are used during a race requiring a minimum of two stops.**

Tyres will be modelled with three different wear phases. A linear phase where the grip loss of the tyre is predictable and linear in nature, A high wear phase where the tyre is still usable but is losing more lap time and grip with each passing lap. Finally the last phase where all usable grip is exhausted and the tyre loses a lot of performance (in F1 terms its referred to as a tyre “falling off the cliff”). The high wear phase will be modelled as a compounding effect after each lap of wear (similar to compound interest). A tyre and its properties therefore will be modelled with the following four parameters:

- `initial_grip`: the grip in seconds per lap the tyre will give as new.
- `initial_deg`: the linear loss in grip per lap expressed as a fraction of time
- `switch_point`: the point at which grip loss changes from linear to high wear
- `switch_deg`: a multiplier that is applied to degradation each lap to determine how much grip is lost

Based on the above you can model your three tyre compounds with the following properties respectively:

- Soft: 2.0, 0.03, 1.8, 1.2
- Medium: 1.6, 0.02, 1.3, 1.2
- Hard: 1.1, 0.01, 0.85, 1.2

For example the soft tyre is worth 2 seconds a lap to begin with but will lose 3 hundreds of a second (0.03 seconds) per lap in the linear phase. The high wear phase starts when overall grip is less than or equal to 1.8 after which the grip loss multiplies by a factor of 1.2 on each lap.

You may also assume the following constants for the purposes of simulation which is designed to approximate the Hungarian GP at the Hungaroring:

- A base lap time of 76 seconds
- A 70 lap race
- each pitstop will add 30 seconds in total to the overall race time
- each car starts with 105Kg of fuel (yes it is measured in Kg not litres as litres is not precise enough in F1)
- each car will use 1.5Kg of fuel per lap

How much fuel is in the car will have an effect on the tyre degradation. Less fuel on board means less wear. You will be required to implement a fuel effect on degradation (will be a multiplier between 0.83 and 1.0) and the degradation must be scaled by this. To calculate the fuel effect you take the current fuel on board and divide it by six times the total fuel at the start of the race. you will then add 0.83 to this number. The reason for this is that the fuel will make up 1/6th of the total mass of the car at the start of the race (i.e. $5/6 = 0.83$)

A discussion of the classes and methods that you will need will be given below:

3 Classes

Discussion of the classes that you will need in order to write the simulation

3.1 Tyre

A class to represent and model a tyre. you will need the following functions in order to represent it

- `__init__`: constructor of the class that takes in all four tyre properties mentioned earlier. you will need to keep a copy of these initial properties for other methods.
- `__eq__`: tests for equality between two tyres. The equality is that all the initial parameters are the same between both tyres
- `addLap(current_fuel, grip_loss_factor)`: simulates the wear of the tyre after another lap has been completed. The calculation of wear should be done according to these rules
 - In the linear phase grip should be reduced by the deg scaled by the fuel effect
 - In the high wear phase (where grip is above 0.2 seconds) the degradation should be multiplied by the `switch_deg`. The grip should then be reduced by this deg scaled by fuel effect.
 - when grip falls below 0.2 it should be set to zero i.e. the tyre has completed fallen off the cliff

The first two are further scaled by the `grip_loss_factor` in that cars that are easier on the tyres will reduce the amount of grip lost

- `calculateLapTime(fuel, lap_time_factor)`: calculates and returns the lap-time based on the current fuel level and state of the tyre. The calculation starts off with the base lap time being scaled by the `lap_time_factor` (allows us to simulate faster and slower cars) and is adjusted as follows:
 - if the grip is above or equal 0.2 then subtract the grip from the laptime
 - if the grip falls below 0.2 then add 2 seconds as the tyre is considered to be unsafe and about to destroy itself.
 - reduce the laptime according to 2 times the fuel load. at 105Kg there should be zero reduction in time. at 0Kg you should be subtracting 2 seconds.
- `reset`: resets the tyre to its initial state

3.2 Car

This is an implementation of car and a race strategy. It will contain a `grip_loss_factor` denoting how hard a car works a tyre. Higher values will wear the tyre out quicker and lower values will make the tyre last longer. Similarly there is a `lap_time_factor` that will denote how fast this car is. Lower values produce quicker laptimes which higher values produce slower laptimes.

The Car class will also contain details of a strategy including: an initial starting tyre, lap numbers for when each pitstop is to be performed, and the tyres to be fitted at each stop. An example two stop strategy may look like this:

- initial tyre: medium
- number of stops: 2
- pit in laps: 20, 50
- pit tyres: soft, hard

This states we start the race on the medium tyre. At the end of lap 20 we will pit in and put on a new set of softs, and on lap 50 we will come in again to put on a new set of hards which we will use to get to the end of the race.

You will need the following functions in your code:

- `__init__`: constructor of the class that takes in all of the parameters needed as shown above including a count of how many laps a race will be in total. this should also construct a variable for the total race time on this strategy (to be used as energy or fitness) and also should prepare a list for storing each laptime of the race on this strategy
- `isValidStrategy`: determines if this is a valid race strategy by checking for the following conditions. if one of these fails then this is not a valid strategy:
 - there are less than 2 pit stops
 - there are more stops than tyres and vice versa
 - not all three compounds of tyre is used in the strategy

- the first stop cannot occur on lap zero
- the last stop cannot occur on the last lap
- the laps for pitstops must increase monotonically and cannot overlap
- `simulateRace()`: simulates a race using this strategy to calculate an overall racetime. The rules for calculating the racetime are as follows
 - reset all tyres to their initial state, clear the lap times, set the car to start with 105Kg of fuel and fit the starting tyre.
 - if this is an invalid strategy set the racetime to 10,000 seconds and return
 - to account for the standing start of the race start set the initial racetime to 5 seconds.
 - simulate each lap by calculating the laptime, add wear to the tyre and reduce the fuel by a single lap, add the calculated laptime to the overall racetime.
 - if this is a pit in lap then the laptime is the laptime plus the time for a pitstop. make sure to change the tyre
 - record each individual laptime in a list
- `energy()`: used for simulated annealing this should return the overall race-time of the strategy
- `chooseRandomTyre()`: will be needed for making moves in simulated annealing. Is equally likely to return any one of the three tyre compounds
- `changeCompound()`: chooses a stop at random (all equally likely) and replace the tyre on that stop with a random tyre
- `changeLap()`: chooses one of the pit in laps at random and flips a coin to determine if we should increment or decrement that lap
- `move()`: needed for simulated annealing. This should flip a coin to determine whether to change a tyre compound or a lap number using the helper functions above.

3.3 StrategyAnnealer

Wrapper around the Car class for the purposes of mapping it to simulated annealing. it should take in a Car object for the state field of the annealer. the move and energy functions should be mapped onto the move and energy functions of the Car class.

4 Advice

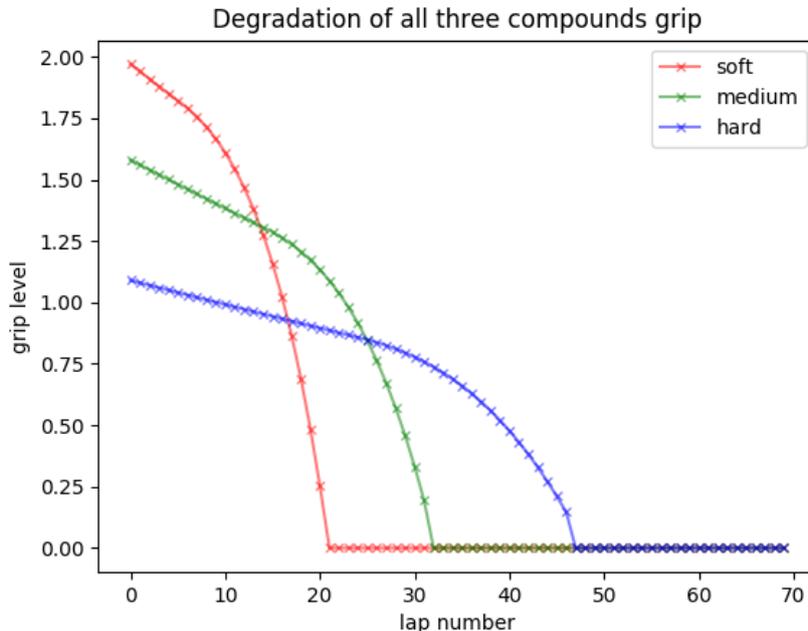
My own advice for tackling this assignment. When implementing the simulation you will need something to see if your calculations are correct in your simulation. The way we would suggest doing this is the following

1. start by modelling the degradation in grip. start on a full fuel load for all three compounds and run them all the way to lap 70. Put the lap numbers in a list and the lap times for all three compounds in three separate lists. Plot all three on a graph using matplotlib. Your aim is to get a similar graph to ours below. also do this for a fuel load of 60Kg to test fuel effect, again the aim is to get something similar to ours below.
2. do the same thing for lap times for each tyre and compare it with the graphs below.
3. finally test a 2 stop strategy starting on the medium and pitting for the soft on lap 20 and the hard on lap 50 and print out the lap times to test the overall simulation. You should get something similar to that below.
4. Once all the above correlates then you should be able to do the simulated annealing and genetic evolution work

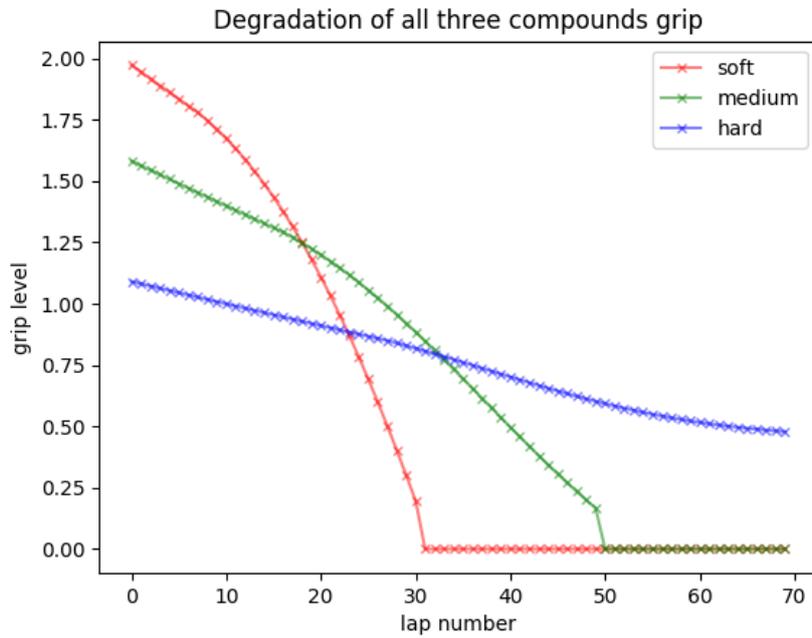
5 Graphs

5.1 Grip deg of all three tyres

Grip degradation of all three tyres on a full fuel load. Soft is red, medium is green, and hard is blue. the crossover point for the soft and medium is about lap 13.84 and the crossover point for medium and hard is around 25.16.



Grip degradation of all three tyres starting with a 60Kg fuel load. The crossover point for the soft and medium is 18.12 laps and the crossover point for the medium and hard is 32.81 laps.



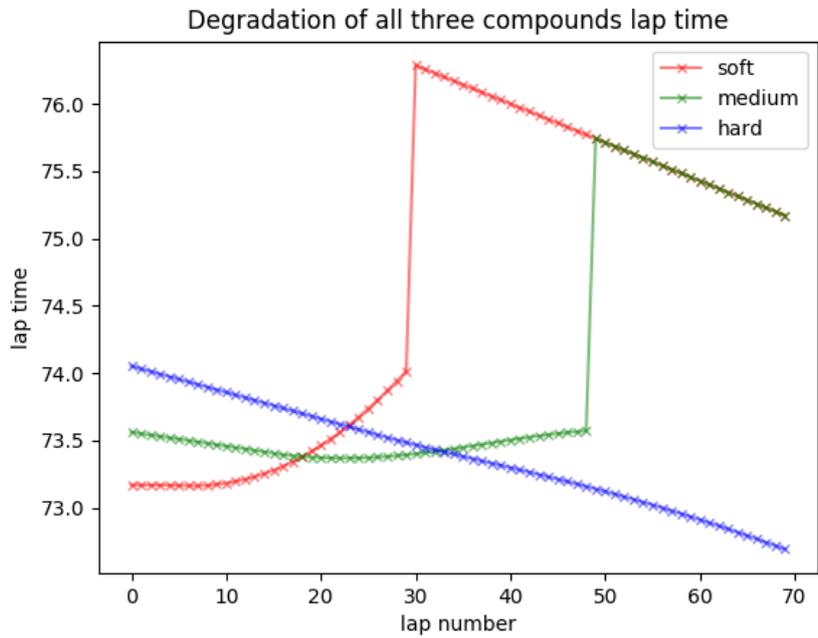
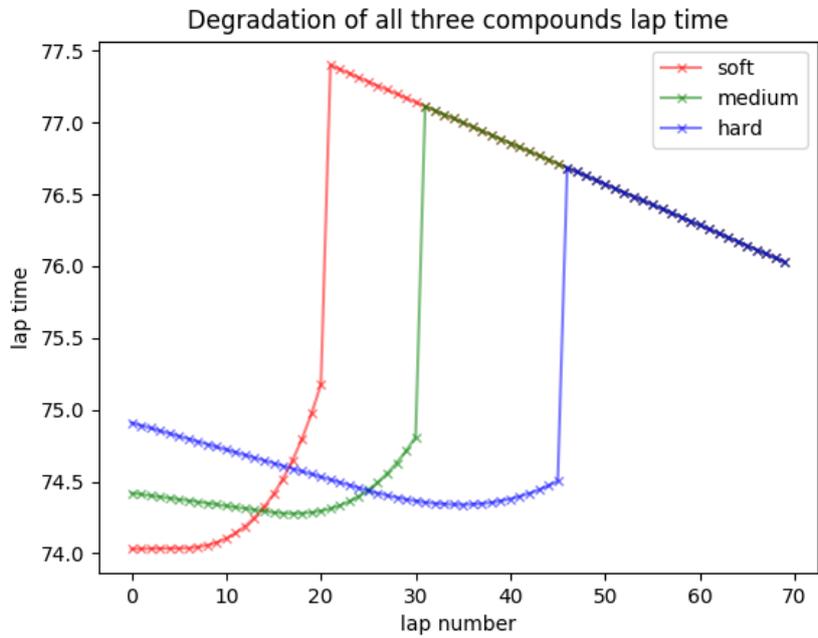
For both graphs be aware that the grip loss factor and lap time factor were both set to 1.

5.2 Laptime relative to tyre degradation and fuel effect

Lap time charts of tyres starting from 105Kg and also from 60Kg that takes into account tyre degradation and fuel effect. Note they get faster to begin with but will fall off the cliff.

In the 105Kg case the crossover point for soft and medium should be around 16.29, medium and hard should be at 25.01. In the 60Kg case the crossovers should be at 17.97 and 32.35 respectively.

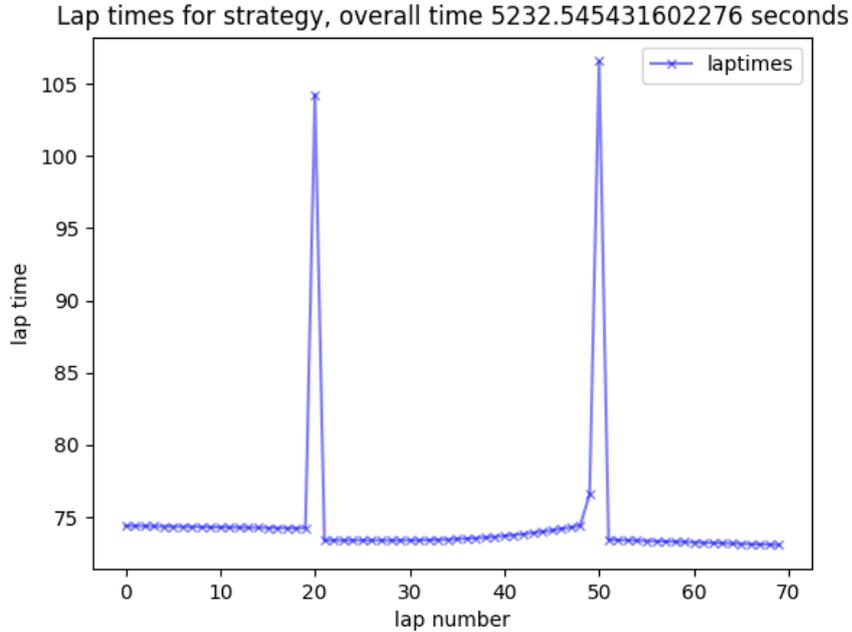
For both graphs be aware that the grip loss factor and lap time factor were both set to 1.



5.3 Sample strategy

Sample one stop strategy represented as laptimes. The lap times that are two spikes is because we've added the 30 seconds for the pitstop. The strategy here is start on the sample strategy listed above. Again the grip loss factor and lap

time factor were both set to 1.



5.4 Other information

An optimal race time will be somewhere between 5200 and 5400 seconds depending on the strategy (around 105 minutes total). You will also need to adapt this simulation to run for genetic evolution when we cover that section of the notes.

In order to generate the graphs above we have done the following:

- for the tyres deg 70 laps from a full fuel load was simulated with 1.5Kg of fuel removed each lap and fuel effect applied to the tyres. There were no stops in between. This was done for all three compounds.
- for the lap times 70 laps from a full fuel load was also simulated with with 1.5Kg of fuel removed each lap and fuel effect applied. There were no stops in between and this was also done for all three compounds.
- To generate the data for the graphs a list of all lap numbers was made for the x axis and a list of deg/laptimes was made for the y axis.
- for the graphs that have three lines there were three sets of laptimes generated and three plot commands were called in matplotlib before showing the graph.

When you get to simulating the race strategies for all three cars you will use the following factors

Team	Colour	Lap Time Factor	Grip Loss Factor
Mercedes	Green	0.987	0.996
Ferrari	Red	0.99	0.99
Red Bull	Blue	0.991	0.982

6 Notes

You are required to submit this assignment by 2021-01-24 (Sunday 24th of January) by 23:55. You are required to submit two separate components to the Moodle

- A single python file containing all of your code.
- A PDF containing documentation of your experiments along with graphs
If you do not provide documentation your code will not be marked.

There are also a few penalties you should be aware of

- Code that fails to compile will incur a 30% penalty before grading. At this stage you have zero excuse to produce non compiling code. I should be able to open your project and be able to compile and run without having to fix syntax errors.
- The use of libraries outside the python SDK and permitted libraries (simanneal, deap, matplotlib) will incur a 20% penalty before grading. You have all you need in the standard SDK. I shouldn't have to figure out how to install and use an external library to get your app to work
- The standard late penalties will also apply

You should be aware that I will remove marks for the presence of bugs anywhere in the code and this will incur a deduction of between 1% and 15% depending on the severity. If you have enough of these bugs it is entirely possible that you may not score very many marks overall. I want robust bug free code for this that closely matches the graphs above

Also note that the percentage listed after the bracket is the maximum mark you can obtain if you complete that many brackets without error.

7 Tasks

1. Write the simulation using the detail above and test it works by correlating your graphs against the ones above (50%)
2. Map the simulation so it can run using simulated annealing to compute the ideal strategy (65%)
3. Map the simulation so it can run using genetic algorithms to compute the ideal strategy (80%)
4. Generate strategies using both simulated annealing and genetic algorithms for the following situations. You must detail what the best strategy is used (initial tyre, pitstop laps, and tyres used)for all three cars, write a small discussion of the strategy (time taken, any interesting effects, total discussion should be no more than 1,000 words). Also state which algorithm generated the result (100%)
 - All three qualified in the top 10 so will start the race on the soft tyre doing two stops.
 - All three qualified in the top 10 but Mercedes is starting on the Medium while the others are on soft, all doing two stops.
 - Ferrari and Red bull qualified in the top 10 starting on medium but Mercedes have a car starting at the back of the grid on the hard tyre. Ferrari and Red bull will do two stops while Mercedes will do three.