Racing Strategy in Formula One

Mercedes AMG Petronas F1 Team Complexity Miniproject Proposal

A race in F1 consists of a set number of laps around a circuit. Each driver has a choice of two types of tyres with different performance and wear characteristics, both of which must be used during the race. A pitstop costs time, but staying out on worn out tyres can cost much more. A race strategy involves deciding when to have pitstops, and which tyres to use, and needs to consider what competitors might do. It should avoid releasing a car from the pits into a convoy of slower cars, because it will cost time to pass them again. It should also consider the likelihood of safety cars, weather, and when to slow down to conserve fuel and conserve tyres. Sporting and technical rules change from year-to-year, so it is important to have a flexible approach to the modelling of strategy.

We would like to investigate agent-based modelling for strategy. This miniproject should make simplifying assumptions about the problem, relaxing these assumptions can be later work.

The first step of the project would be to write a stochastic race simulator, which will provide the environment for informing the agent-based model in the second part of the project. Initial positions and time deltas of the cars would be input, and then a simple algorithm performed to see how the positions change each lap. See the section below for details. It is anticipated this part of the project would take about a week. We would help with required models and data to get this part to work, if required.

The second and main part of the project would be to replace the simple pitting approach of the first with an agent-based model, or any other approach which might be interesting to try. There is a lot of scope and freedom for this part of the project, and most of the time should be spent on it. It would be useful to have this part of the software as modular as possible, so that different strategy algorithms can be compared easily.

Stochastic Race Simulator

Below is a suggestion, based on a large amount of earlier experience, for how the first part of the project would work. The aim is to see how race positions and time gaps change from one lap to the next.

Inputs: Race position, time-behind-leader, and some parameters, for each car.

Algorithm:

1. Calculate the laptime of each car. Add this laptime to time-behind-leader of the car.
2. A car pits after a pre-determined number of laps since its last pitstop. If a car pits then the standard time taken to pit is added to its time-behind-leader.
3. If any car now has a time-behind-leader that's less than the car in front, it has caught the car in front on that lap. It overtakes with a calculated probability, if not it takes the car in front's time-behind-leader, with a small gap added.
4. Positions are then updated by sorting by time-behind-leader, and then time-behind-leader is renormalized (all translated so it becomes zero for the leader).

Laptime is calculated by adding a car's base laptime (provided as a parameter) to a simple model which gives laptime delta as a function of fuel levels and tyre degradation, so would take number of laps since start of race and number of laps since last pitstop as inputs.

Overtaking probability on a particular lap is calculated as a function of time-behind-leader differences, the driver in front's defending ability, and the driver behind's overtaking ability.

Outputs: Updated race position and time-behind-leader, for each car.