1 The Canadian Grand Prix

It's the 2022 Formula 1 season and the FIA (Fédération Internationale de l'Automobile) has decided to change their qualifying rules to make it more challenging for the drivers[1].

In this short note, we build a toy model to find the best time where a driver can confirm his best qualifying time.

1.1 Problem statement and Model formulation

Qualifying is held Saturday, the day before the race, and sets drivers positions on the starting grid for the race. In a nutshell, drivers have one hour to make their fastest lap which will determine their staring position for the race.

This year, a small change to the rules to make qualifying more challenging: the lap that determines the driver's position on the grid is his last one and the driver can exit the race at any time. The driver can choose at the beginning of each lap k, where $0 \leq k \leq N-1$, whether he is satisfied with his last lap time w_k or wants to try his luck for another lap. Drivers are also rewarded for exiting early as their final lap time gets discounted by a rate r for each of the remaining laps. We want to find the optimal moment when to exit the race minimizing out lap time.

The lap time is a complicated function of many parameters (car mechanics, fuel available, tire quality ...). To simplify our model, we represent the time lap as a uniformly distributed random variable $w_k \sim \mathcal{U}(70, 85)$[2].

We want to make this problem similar to that of slide 45 of [2]. Minimizing a lap time is equivalent to maximizing its inverse. Therefore we consider inverse lap times $\omega_k \equiv \frac{1}{w_k}$. This is the same as considering $\omega_k \sim \mathcal{U}(142, 167)$ where $167 \simeq 10^4/85$ and $142 \simeq 10^4/70$[3].

We formulate the problem as a dynamic programming problem, where we have:

- the N stages are exactly the maximum number of laps allowed, $k = 0, 1, \ldots, N-1$,
- the action is $u_k = 0$ if we stay in the race and $u_k = 1$ if we decide to exit. If $u_k = 1$ for some $0 \leq k \leq N-1$, then $u_l = 0$ for $l \neq k$. And finally $u_0 = 0$,
- the state at each stage is given by the current time lap, where $x_0 = 0$ and

$$x_{k+1} = \begin{cases} \omega_k & \text{if } u_{0 \leq i \leq k} = 0, \\ \Delta & \text{if not.} \end{cases}, \tag{1}$$

[1] We only change rules for the sake of the homework

[2] these bounds are in seconds and are based on the real average time lap scores found in [1]

[3] We multiplied by 10^4 to get rid of the floating small numbers. This does not affect the maximization problem.

- the expected optimal time lap is given by,

$$J_k(x_k) = \begin{cases} 0 & \text{if } x_k = \Delta, \\ x_N & \text{if } k = N \text{ and } x_N \neq \Delta, \\ \max\left\{(1+r)^{N-k} x_k, \mathbb{E}\left[J_{k+1}(\omega_k)\right]\right\} & \text{if not} \end{cases} \tag{2}$$

- we define V_k as,

$$\begin{aligned} V_k(x_k) &= \frac{J_k(x_k)}{(1+r)^{N-k}} \\ &= \begin{cases} \max\left\{x_k, \mathbb{E}\left[V_{k+1}(x_{k+1})/(1+r)\right]\right\} & \text{if } k < N, \\ x_N & \text{if } k = N, \end{cases}, \end{aligned}$$

- we define α_k as,

$$\begin{aligned} \alpha_k &= \frac{\mathbb{E}\left[J_{k+1}(\omega_k)\right]}{(1+r)^{N-k}} \\ &= \frac{\mathbb{E}\left[V_{k+1}(\omega_k)\right]}{(1+r)} \\ &= \frac{\mathbb{E}\left[V_{k+1}(x_{k+1})\right]}{(1+r)} \\ &= \begin{cases} \frac{1}{(1+r)}\left(\mathbb{E}\left[\omega_k \mathbb{I}\left[\omega_k > \alpha_{k+1}\right]\right] + \alpha_{k+1}\mathbb{P}\left[\omega_k \leq \alpha_{k+1}\right]\right) & \text{if } k \neq N, \\ 0 & \text{if } k = N, \end{cases}, \end{aligned} \tag{3}$$

- following the definition of α_k, the actions take the form,

$$u_k = \begin{cases} 1 & \text{if } x_k \neq \Delta \text{ and } x_k \geq \alpha_k, \\ 0 & \text{if not} \end{cases}$$

The optimal decision, when there is still a choice, is to exit the race, i.e qualifying time $x_k = \omega_{k-1}$, if and only if $x_k \geq \alpha_k$. The analytical solution of of α_k is given by,

$$\begin{aligned} \alpha_k &= \frac{1}{1+r}\left(\mathbb{E}\left[\omega_k \mathbb{I}\left[\omega_k > \alpha_{k+1}\right]\right] + \alpha_{k+1}\mathbb{P}\left[\omega_k \leq \alpha_{k+1}\right]\right) \\ &= \frac{1}{1+r}\left(\int_{t_0}^{t_1} \omega_k \mathbb{I}\left[\omega_k > \alpha_{k+1}\right] \frac{1}{t_1 - t_0} \mathrm{d}\omega_k + \alpha_{k+1}\int_{t_0}^{\max(t_1, \alpha_{k+1})} \frac{1}{t_1 - t_0} \mathrm{d}\omega_k\right) \\ &= \frac{1}{1+r}\left(\frac{1}{2}\frac{t_1^2 - \max\left(t_0, \alpha_{k+1}\right)^2}{t_1 - t_0}\mathbb{I}\left[t_1 \geq \alpha_{k+1}\right] + \alpha_{k+1}\frac{\max\left(t_0, \alpha_{k+1}\right) - t_0}{t_1 - t_0}\right), \end{aligned} \tag{4}$$

where $t_0 = 142$s and $t_1 = 167$s.

2 Solution

2.1 Instances

We create two instances to solve, one small $N = 3$ and the other larger $N = 15$.

- **Small instance:** We put $N = 3$ and generate ω_k using the uniform distribution $\mathcal{U}(142, 167)$. We find

  $$\omega_0 = 146.79, \qquad \omega_1 = 157.55, \qquad \text{and } \omega_2 = 152.94.$$

  Then we compute α_k using equation (4). By definition $\alpha_{N=3} = 0$. Now we solve recursively for $k = 2$,

  $$\begin{aligned}\alpha_2 &= \frac{1}{1+r}\left(\frac{1}{2}(t_1 + t_0)\right) \\ &= 154.34,\end{aligned}$$

  where we set $r = 0.05$.

  For $k = 1$ we have,

  $$\begin{aligned}\alpha_1 &= \frac{1}{1+r}\left(\frac{1}{2}\frac{t_1^2 - \alpha_2^2}{t_1 - t_0} + \alpha_3\frac{\alpha_2 - t_0}{t_1 - t_0}\right) \\ &= 157.39.\end{aligned}$$

- **Large instance:** We put $N = 15$ and generate ω_k using the uniform distribution $\mathcal{U}(142, 167)$. We find

  $$\begin{array}{rrrrr} \omega_0 = 146.79, & \omega_1 = 157.55, & \omega_2 = 152.94, & \omega_3 = 161.63, & \omega_4 = 161.5 \\ \omega_5 = 148.81, & \omega_6 = 148.91, & \omega_7 = 162.05, & \omega_8 = 165.95, & \omega_9 = 163.9 \\ \omega_{10} = 150.95, & \omega_{11} = 154.52, & \omega_{12} = 159.09, & \omega_{13} = 159.82, & \omega_{14} = 151.26. \end{array}$$

  We then compute α_k recursively starting with $\alpha_1 5 = 0$ and we find,

  $$\begin{array}{rrrrr} \alpha_1 = 163.4, & \alpha_2 = 163.29, & \alpha_3 = 163.15, & \alpha_4 = 163.00, & \alpha_5 = 162.81 \\ \alpha_6 = 162.58, & \alpha_7 = 162.30, & \alpha_8 = 161.95, & \alpha_9 = 161.52, & \alpha_{10} = 160.94 \\ \alpha_{11} = 160.17, & \alpha_{12} = 159.08, & \alpha_{13} = 157.39, & \alpha_{14} = 154.34, & \alpha_{15} = 0. \end{array}$$

Algorithm 1 Backward Chaining for Stochastic case[2]

Require: $N \geq 0$
  for $x \in X_N$ **do**
    $J_N(x) \leftarrow g_N(x)$
  end for
  for $k = N-1, \dots, 0$ **do**
    $J_k(x) \leftarrow \min_{u \in U_k(x)} \mathbb{E}\left[g\left(x_k, u_k, \omega_k\right) + J_{k+1}\left(f\left(x_k, u_k, \omega_k\right)\right)\right]$
    $\mu_k^*(x) \leftarrow \arg\min_{u \in U_k(x)} \mathbb{E}\left[g\left(x_k, u_k, \omega_k\right) + J_{k+1}\left(f\left(x_k, u_k, \omega_k\right)\right)\right]$ ▷ Optimal policy
  end for

2.2 Results with the original model

We want to compute the optimal policy in two ways: first with using the analytical form, then with the dynamic programming algorithm 1.

- **Small instance:** We computed in section 2.1 the analytical solutions for α_k and generated ω_k as well.

  - for $k = 1$, we have $x_1 = \omega_0 = 146.79 \leq \alpha_1 = 157.39$ and therefore $u_1 = 0$,
  - for $k = 2$, we have $x_2 = \omega_1 = 157.55 \geq \alpha_2 = 154.34$ and therefore $u_2 = 1$,
  - for $k = 3$, since $u_2 = 1$, then $u_3 = 0$.

  Therefore the analytical solution is $\boldsymbol{u} = (0, 1, 0)$.

  Now we solve using dynamic programming. We find the following,

  - for $k = 3$, we have $J_3(x_3) = \omega_2 = 152.94$,
  - for $k = 2$, we have

  $$\begin{aligned} J_2(x_2) &= (1+r)\max(\omega_1, \alpha_2) \\ &= 1.001 \cdot 157.55 \\ &= 157.70755, \end{aligned}$$

    and $u_2 = 1$,
  - for $k = 1$, we have

  $$\begin{aligned} J_1(x_1) &= (1+r)^2\max(\omega_0, \alpha_1) \\ &= 1.001^2 \cdot 157.39 \\ &= 157.705, \end{aligned}$$

    and $u_1 = 0$.

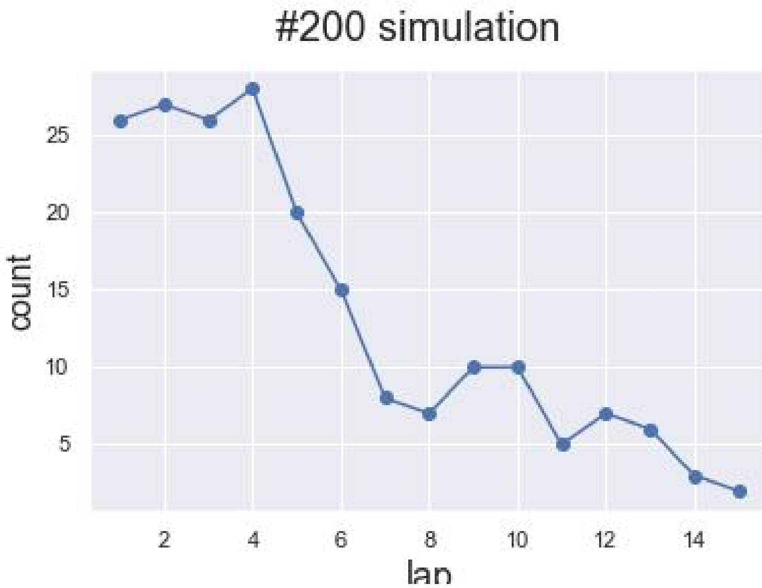

Figure 1: plot of the average optimal policy of 200 simulation.

We do find the same solutions as those of the analytical solution $\boldsymbol{u} = (0, 1, 0)$.

- **Large instance:** We computed in section 2.1 the analytical solutions for α_k and generated ω_k as well. For the analytical solution, we find,

$$\boldsymbol{u} = (0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0).$$

Now we solve using dynamic programming. We find the following,

$$\begin{aligned}\boldsymbol{J} &= (165.7, 165.42, 165.12, 164.8, 164.44, 164.05, 163.61, 163.19,\\ &\quad 166.95, 164.72, 160.82, 159.56, 159.41, 159.98, 151.26)\\ \boldsymbol{u} &= (0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0),\end{aligned}$$

where we used `np.round_(J, 2)` for clarity.

Again we obtain the same solutions as the analytical version, we exit at lap number eight.

In order to get a better idea for best optimal policy, we run a simulation over 200 samples and take the average lap exit. We present a plot of the simulation result in figure 1. We can see that the best policy is $\boldsymbol{u} = (0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$.

The average lap time is $\tau = 187.61600627421979$s.

2.3 Results with a modified model

Let's use a more realistic distribution of ω_k. We choose a a normal distribution centred around the average lap time, i.e $\omega_k \sim \mathcal{N}\left(\mu, \sigma^2\right)$

$$\mu = \frac{t_1 + t_0}{2}, \qquad \sigma = \frac{t_1 - t_0}{2}.$$

In this case we have to compute again the analytical value of α_k using equation (3). Given the complication of this expression, we use `Mathematica` to solve α_k as follow,

```
In[1]:= f1 = 1/(Sqrt[2 Pi] σ) Integrate[ x Exp[-(x-μ)^2/(2σ^2)] dx, {x, a, ∞}]
        f2 = a/(Sqrt[2 Pi] σ) Integrate[  Exp[-(x-μ)^2/(2σ^2)] dx, {x, -∞, a}]
        α = (f1 + f2)/(1+r) // N
```

- **Small instance:** We put $N = 3$ and generate ω_k using the normal distribution $\mathcal{N}\left(\mu, \sigma^2\right)$. We find

  $$\omega_0 = 160.39, \qquad \omega_1 = 139.61, \qquad \text{and } \omega_2 = 172.41.$$

  Then we compute α_k using equation (3) with the help of `Mathematica`. By definition $\alpha_{N=3} = 0$. Now we solve recursively for $k = 2$, we find $\alpha_2 = 154.346$. For $k = 1$ we have, $\alpha_1 = 189.277$.

  - for $k = 1$, we have $x_1 = \omega_0 = 160.39 \leq \alpha_1 = 189.277$ and therefore $u_1 = 0$,
  - for $k = 2$, we have $x_2 = \omega_1 = 139.61 \leq \alpha_2 = 154.346$ and therefore $u_2 = 0$,
  - for $k = 3$, we have $x_3 = \omega_2 = 172.41 \geq \alpha_3 = 0$ and therefore $u_3 = 1$,

  Therefore the analytical solution is $\boldsymbol{u} = (0, 0, 1)$.

  Now we solve using dynamic programming. We find the following,

  - for $k = 3$, we have $J_3(x_3) = \omega_2 = 172.41$,
  - for $k = 2$, we have

    $$\begin{aligned} J_2\left(x_2\right) &= (1 + r)\max(\omega_1, \alpha_2) \\ &= 1.001 \cdot 154.346 \\ &= 154.50, \end{aligned}$$

    and $u_2 = 0$,

– for $k = 1$, we have

$$\begin{aligned} J_1(x_1) &= (1+r)^2 \max(\omega_0, \alpha_1) \\ &= 1.001^2 \cdot 189.277 \\ &= 189.65, \end{aligned}$$

and $u_1 = 0$.

We do find the same solutions as those of the analytical solution $\boldsymbol{u} = (0, 0, 1)$.

- **Large instance:** We put $N = 15$ and generate ω_k using the normal distribution $\mathcal{N}\left(\mu, \sigma^2\right)$. We find

$$\begin{array}{rrrrr} \omega_0 = 160.39, & \omega_1 = 139.61, & \omega_2 = 172.41, & \omega_3 = 150.59, & \omega_4 = 145.49 \\ \omega_5 = 165.59, & \omega_6 = 165.24, & \omega_7 = 146.54, & \omega_8 = 154.7, & \omega_9 = 126.47 \\ \omega_{10} = 168.88, & \omega_{11} = 166.9, & \omega_{12} = 166.42, & \omega_{13} = 129.23, & \omega_{14} = 150.32. \end{array}$$

We then compute α_k recursively starting with $\alpha_1 5 = 0$ and we find,

$$\begin{array}{rrrrr} \alpha_1 = 189.277, & \alpha_2 = 188.462, & \alpha_3 = 187.566, & \alpha_4 = 186.573, & \alpha_5 = 185.465 \\ \alpha_6 = 184.215, & \alpha_7 = 182.788, & \alpha_8 = 181.134, & \alpha_9 = 179.181, & \alpha_{10} = 176.816 \\ \alpha_{11} = 173.846, & \alpha_{12} = 169.916, & \alpha_{13} = 164.232, & \alpha_{14} = 154.346, & \alpha_{15} = 0. \end{array}$$

We have just computed the analytical solutions for α_k and generated ω_k as well. For the analytical solution, we find,

$$\boldsymbol{u} = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0).$$

Now we solve using dynamic programming. We find the following,

$$\begin{aligned} \boldsymbol{J} = (&191.94, 190.93, 189.83, 188.63, 187.33, 185.88, 184.25, 182.40, \\ &180.26, 177.70, 174.54, 170.432, 166.75, 154.50, 150.32) \\ \boldsymbol{u} = (&0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0), \end{aligned}$$

where we used `np.round_(J, 2)` for clarity.

Again we obtain the same solutions as the analytical version, we exit at lap number 13.

In order to get a better idea for best optimal policy, we run a simulation over 200 samples and take the average lap exit. We present a plot of the simulation result in figure 2. We can see that the best policy is $\boldsymbol{u} = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0)$.

The average lap time is $\tau_m = 168.4492617944438$s. The relative error (with respect to the value of the objective function) between the two models is $err \simeq 10.21\%$.

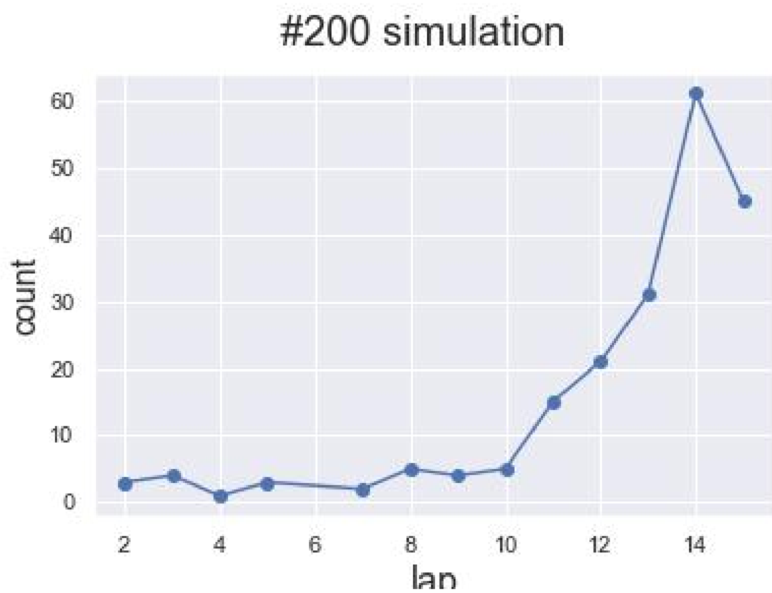

Figure 2: plot of the average optimal policy of 200 simulation.

The main difference between figure 1 and 2 is that in the latter, it's better to exit at the end of the race instead of the beginning. This makes sense, since with a Gaussian distribution, we will always have better chances to perform near average μ while our chances of performing extremely poorly or extremely good are very low given σ.

The main modification we did to adjust the model to a Gaussian distribution is to compute α_k using `Mathemmatica`. The rest of the code is similar to the original model.

3 Comments

3.1 Inspiration

This note is inspired mainly from [2] and [3].

3.2 Instructions for code

We provide a `.ipynb` file as well as a `.nb` file. for the modified model, in order to change the instances `n`, one has to compute α_k using the `.nb` file and copy those values back to the `.ipynb` file.

References

[1] F1 LAPS, *Average F1 2020 Lap Times per Track* https://www.f1laps.com/articles/f12020/track-laptime-averages/.

[2] Emma Frejinger, *IFT 6521. Programmation Dynamique, Lecture Slides*, Modèles stochastiques sur horizon fini, page 45.

[3] Maya Abou-Rjeili, *IFT 6521: Devoir 3*, 2021.