



Universitatea Națională  
de Știință și Tehnologie  
POLITEHNICA BUCUREȘTI



FACULTATEA DE  
AUTOMATICĂ ȘI  
CALCULATOARE

UNIVERSITATEA NAȚIONALĂ DE ȘTIINȚĂ ȘI TEHNOLOGIE  
POLITEHNICA DIN BUCUREȘTI  
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
INGINERIA SISTEMELOR

## OPTIMIZĂRI

*Calcularea strategiei de oprit la boxe*  
*“PitStrategist”*

Grupa: 321AC

Student: Dodi Mircea-  
Ovidiu

## Cuprins

1. Descrierea problemei .....	3
2. Formularea matematică a problemei.....	4
3. Soluționarea problemei .....	5

## 1. Descrierea problemei

În cadrul curselor de Formula 1, lupta dintre piloți nu este dusă exclusiv pe circuit – echipele rulează numeroase simulări pentru a determina cea mai bună strategie pentru opritul la boxe, ce are un mare rol în rezultatul final al cursei.

Prin prezentul proiect, este creat un model ce ia în considerare parametri reali, precum degradarea cauciucurilor sau cantitatea de combustibil aflată la bordul mașinii pentru a încerca determinarea unei strategii optime de oprire la boxe.

Vom considera formatul cursei de tip “Sprint Race”, de mai puține ture, în care toți piloții folosesc cel mai moale compus de pneuri disponibil și conduc aproape de limitele mașinii, pentru a elimina nevoia mai multor variabile față de cazul cursei, în care sunt puse la dispoziție 3 tipuri de cauciucuri și strategiile piloților sunt mult mai complexe și variate.

### Considerente practice:

Aceste simulări, chiar și în practică, nu vor fi folosite 1:1 în realitate mereu. Numărul de variabile ce determină timpul unui tur este imens, de la vreme, temperatura circuitului, vânt, cantitatea de fragmente de pneu depuse pe timpul cursei, stilul de condus al pilotului ș.a.m.d. De asemenea, hazardul intervine adesea, prin factor uman, vreme neașteptată sau schimbătoare, accidente ce pot scoate un safety car, ritmul celorlalți șoferi etc. Aceste simulări sunt folosite drept un punct de plecare pentru alegerea strategiei folosite sau pentru teste, când pilotul conduce în condiții ideale pe circuit, în absența concurenților pentru culegerea de date privind mașina.

## 2. Formularea modelului matematic

### 2.1. Variabile utilizate.

În modelul matematic prezent, formulăm problema în termeni ai programării liniare.

Variabilele folosite sunt:

$N$  – numărul de ture impus pentru atingerea distanței de 306km, conform regulilor.

MaxLapsPerStint – număr maxim de ture pe un set de pneuri, pentru a evita exploziile.

PitStopTime – timpul pierdut în medie pentru o oprire la boxe.

baseLapTime – timp mediu estimat pentru o tură în timpul cursei, cu 4 secunde peste cel mai rapid din calificări

degradation\_rate – constantă prin care estimăm efectul degradării pneurilor asupra turei.

fuel\_consumption\_effect – similar, estimăm efectul consumului de combustibil.

stint\_lap[] – vector ce ține evidența a câte ture au trecut de la ultima oprire.

(Conform regulamentului F1, alimentarea în timpul cursei este interzisă din 2010).

### 2.2. Modelul propriu-zis și constrângerile.

Obiectivul nostru este terminarea cât mai rapidă a cursei, prin urmare vrem un timp total al cursei cât mai mic. Funcția obiectiv este deci timpul său, egal cu suma dintre timpul total al tururilor și durata pit-stop-urilor:

```
1 # Objective function - minimize total race time
2 @objective(model, Min, sum(lap_time_with_degradation_and_fuel(baseLapTime, stint_lap[i], i)
3                           + PitStopTime * x[i] for i in 1:N-1))
```

Data fiind sintaxa Julia, aceasta funcție este notată drept “@objective”, iar ca parametri ia “model” – modelul, cuprins din funcție și constrângeri și “Min”, scopul fiind minimizarea sa.

Constrângerile:

1. în prima tură nu se poate intra la boxe
2. în ultima tură nu se poate intra la boxe
3. în fiecare cursă minim o oprire trebuie făcută
4. distanță de minim un tur între opriri
5. nu mai mult de 25 de ture pot fi făcute pe același set de cauciucuri.
6. cursa trebuie parcursă în întregime
7. maxim o oprire per tur

Ultimele două sunt triviale, însă sunt plasate pentru a evita eventualele aberații în rezultate.

```
1  # Number of pit stops within the allowed range
2  @constraint(model, sum(x) >= MinPitStops)
3  @constraint(model, sum(x) <= MaxPitStops)
4
5  # Ensure that the race can be completed
6  for i in 1:(N-MaxLapsPerStint)
7      @constraint(model, sum(x[j] for j in i:(i+MaxLapsPerStint-1)) >= 1)
8  end
9
10 # Ensure the first and Last Laps cannot be pit stops - regulations
11 @constraint(model, x[1] == 0)
12 @constraint(model, x[N-1] == 0)
```

Am luat în calcul influența degradării pneurilor și micșorarea cantității de combustibil asupra timpilor, prin funcția următoare:

```

1 # Lap time calculation with degradation and fuel consumption effect
2 function lap_time_with_degradation_and_fuel(baseLapTime::Int, stint_lap::Int, lap_number::Int,
3     degradation_rate::Float64, fuel_consumption_effect::Float64)::Float64
4     return baseLapTime + (stint_lap - 1) * degradation_rate - (lap_number - 1) * fuel_consumption_effect
5 end

```

Aici cele două constante definite precedent își au rolul.

### 3. Soluționarea problemei

În urma definirii problemei, înlocuim variabilele descrise precedent.

Drept exemplu numeric, luăm drept sursă timpii scoși în timpul Marelui Premiu al Austriei 2023: <https://f1pace.com/p/2023-austrian-gp-race-pace/>.

```

1 # Define season and circuit-specific data: The 2023 Austrian Grand Prix held at the Red Bull Ring for example.
2 N = 25 # Total number of laps
3 MaxLapsPerStint = 9 # Maximum number of laps that can be performed on a stint
4 PitStopTime = 25 # Time wasted in a pit stop (in seconds)
5 MinPitStops = 1 # Minimum number of pit stops, enforced by regulations
6 MaxPitStops = N # Maximum number of pit stops
7 baseLapTime = 68 # Projected race Lap time, 5s off pole
8 degradation_rate = 0.25 # avg rate of Lap time increase due to tyre wear per lap (in seconds)
9 fuel_consumption_effect = 0.08 # avg Lap time decrease per lap due to fuel consumption (in seconds)

```

Am folosit optimiser-ul GLPK, alternativa GNU la CPLEX, ce aplică Simplex: <https://www.gnu.org/software/glpk/>.

Rezultatele au fost:

```

Objective function value, race time: 1683.17s
-----
Optimal pit strategy:
Pitting on lap: 9
Pitting on lap: 16

```

Codul precum și output-ul complet, unde se poate vizualiza evoluția timpilor în fiecare tur:



pitStopStrat.ipynb



out.txt