Department of Economics and Finance

Teaching: Games and Strategies

Strategic Decision-Making in Formula 1

Supervisor:

Prof. Xavier Mathieu Raymond Venel

Candidate:

Marcello Saddemi

ID: 286291

Academic Year 2024/2025

Abstract

This paper delves into the process of strategic decision-making in Formula 1 racing by combining elements of game theory, behavioural models, and simulations. First, an overview of Formula 1 and the main factors that influence its race strategy is established. Subsequently, a literature review investigating previous works on game theory concepts applied to race strategy is provided. This is followed by a section exploring mathematical and game theoretic concepts present in real life Formula 1 racing and race strategy simulations. Lastly, several models that replicate F1 racing scenarios are presented, analyzed, and ultimately simulated in order to examine and explore strategic decision-making in F1-like races.

It is important to mention that artificial intelligence in the form of ChatGPT was used for the creation of the model's Python scripts.

Table of Contents

1. Introduction **3**
1.1 Motorsport and Formula 1 History 3
1.2 F1 and Strategies 4
1.2.1 Predictive Simulations 5
1.2.2 Pit Stop Tactics 6
1.2.3 Fuel Management 6
1.2.4 Tire Management 7
2. Literature Review **8**
3. Preliminary Mathematics **10**
3.1 F1: A Non-Cooperative Game 10
3.2 Closed-Loop Systems 11
3.3 Open-Loop Systems 12
3.4 Dynamic Games' Structure 13
4. Simulating Race Strategies **14**
4.1 Outline 14
4.2 Time Difference - 2 Driver Model 14
4.2.1 Numerical Example 16
4.2.2 Limitations and F1 Implications 18
4.3 Optimal Pit Stop Model 19
4.4 Predefined Strategies Models 21
4.4.1 Different Car Performance 21
4.4.2 Same Car Performance - Pace vs Degradation Trade-off 24
4.4.3 An Interactive Model 27
4.5 Nash Equilibrium-Based Strategy Selection Models 31
4.5.1 Best Response Model 31
4.5.2 Dynamic Interaction Model 32
4.5.2.1 Example 1 33
4.5.2.2 Example 2 33
4.5.2.3 Example 3 37
4.5.2.4 Example 4 39
5. Conclusion **43**
6. Works Cited **45**
7. Appendix **47**

1. Introduction

1.1 Motorsport and Formula 1 History

Automotive racing started soon after the development of the gasoline-powered combustion engine at the end of the 19th century. According to Encyclopædia Britannica (*Britannica*), since the invention of transportation, humans have always found a way to make it enjoyable. That marked the beginning of motorsport, and it has not stopped to fascinate and captivate the world, providing people with adrenaline rushes from witnessing high-performance vehicles race each other. Several categories, such as Rally Racing, MotoGP, and Formula 1, have allowed people to connect and bond over their shared interests, fostering a sense of community.

Formula 1, since its first official race in 1950 in France, represents the pinnacle of motorsports and continues to fascinate all over the globe, creating a fan base of over 826 million people. Since its inauguration, F1 (*F1*) has not stopped developing, introducing races in South & North America, Africa, Asia, Australia, and, most recently, the Middle East, making it a truly global championship. People are attracted to the thrill produced by the 20 drivers[1] competing for the most prestigious and sought-after prize, the F1 Drivers' Championship. Simultaneously, the teams[2] fight for the F1 Constructors' Championship and the corresponding prize money, which depends on their position in the rankings at the end of the season.

The constant thrill of watching teams implement cutting-edge technology and the finest automotive engineering to develop and deliver the most competitive car has captivated the interest of an increasing number of people in recent years. As indicated by Max Simon (*Simon*), since 2021, F1's global fanbase experienced a growth of over 12% and a general rise of interest of 5.7%. This growth is mainly explained by an increase in social media following and the impact of the "Drive to Survive" Netflix series, publicizing the sport on a transnational level.

Moreover, with the passing of years and advancements in technology, F1 has become more complex, making it more difficult for teams to get the upper hand on their competitors. Therefore, teams try to optimize every single aspect of each race, such as the setup of the car,

---

[1] Drivers currently competing in the 2025 F1 season: *Norris, Piastri, Russell, Antonelli, Hamilton, Leclerc, Verstappen, Tsunoda, Hadjar, Lawson, Alonso, Stroll, Bearman, Ocon, Bortoleto, Hülkenberg, Gasly, Doohan, Sainz, Albon*.

[2] Constructors currently competing in the 2025 F1 season: *Mclaren, Mercedes, Ferrari, Red Bull Racing, Racing Bulls, Aston Martin, Haas, Kick Sauber, Alpine, Williams*.

tire choice, and when to pit. Teams try to develop and implement both on- and off-track strategies, which allow them to unlock and use the full potential of their car. Such strategies are, for example, deciding when to pit during a race or which tires to choose at the start.

1.2 F1 and Strategies

Strategies have always played a major role in motorsport and are a crucial part of Formula 1. Teams and drivers tend to implement strategies in a multitude of situations, both before and during the race, as there are some events you can not prepare for. In F1, race strategy is all about trying to achieve the best possible race outcome. This is done through a mix of data analysis and predictive simulations, typically executed by a team of strategists and engineers to optimize certain factors, such as choosing tire compounds, when to pit, and fuel management. As stated by Catapult (*Catapult 2023*), a commercial sports technology company that develops data analysis tools, every team on the grid[3] employs a range of tactics to beat their competitors, from predicting the weather to anticipating their rivals' next move. Ultimately, race strategy is centered around maximising performance while simultaneously minimising the time lost during pits, resulting in a plan which unfolds during a race.

Data and predictive simulations play a key role in F1, as executing a winning strategy often involves drivers making split-second decisions based on real-time insights. To do so effectively, Formula 1 teams continuously gather thousands of data points throughout each race via telemetry, real-time data gathering sensors mounted on the cars. These sensors capture a vast array of information, such as tire temperature, speed, fuel consumption, and engine performance, powering millions of simulations that predict and create a plan for any scenario that can arise during a race. F1 teams rely heavily on data to make informed decisions and optimise their chances of gaining an edge on their competitors and ultimately securing championships.

[3] The F1 *"grid"* refers to the 10 teams and 20 drivers participating in the current season.

1.2.1 Predictive Simulations

Predictive models have become vital in Formula 1, combining data from sensors with simulation tools to predict certain situations in advance. Teams often rely on data from past races held at specific tracks to uncover patterns and key performance indicators that influence their strategic planning. This retrospective analysis enables them to refine crucial elements such as pit stop timing, tire degradation, and overtaking zones. Formula 1 teams employ sophisticated modelling systems to replicate race conditions, taking into account factors like weather predictions, track dynamics, and rival performance. These models enable the testing of various scenarios, ranging from when to pit and what tires to use to how much fuel to load, helping teams determine the most effective strategy by assessing potential outcomes. Furthermore, predictive simulations enhance strategic capabilities by integrating real-time data with machine learning and statistical tools. As stated by Intrafocus (*Intrafocus*), a performance management software firm, these models process numerous variables, including vehicle performance, meteorological inputs, and competitor behaviour, to estimate finishing positions and lap times. With this foresight, teams can adapt strategies dynamically during the race, capitalising on new opportunities and responding swiftly to unfolding events to improve their chances of a strong finish. During races, teams utilize algorithms such as Bayesian networks to refine their race strategies. Moreover, machine learning techniques, particularly Recurrent Neural Networks (RNNs), are used to detect intricate patterns within the data. This allows for a more nuanced understanding of race behaviour and improves the precision of performance predictions.

To better understand race strategy, F1 teams use gapper or race history plots, visual tools used to identify the time gap between drivers throughout a race. A gapper plot measures the time difference between a car and a reference car throughout a race (Figure 1, Appendix). According to Catapult (*Catapult 2024*), gapper plots help strategists monitor how the race is unfolding and allow them to simultaneously track the performance of both their cars and those of their competitors. Specifically, it plots the relative position of each car to a given zero on the y-axis, which in the case of Figure 1 is the race winner's average lap time, while the x-axis tracks the relative time progression during a race. As specified by MIA School of Race Engineering (*MIA*), gapper plots play a significant role in pit stop tactics as race engineers use them to spot undercut/overcut[4] opportunities, or to identify the best possible time to pit safely without losing track position. An example of a data analysis and predictive

[4] *Undercut:* driver pits before their rival to gain an advantage with fresher tires. *Overcut:* driver stays out longer and pits after their rival, gaining an advantage by being faster on older tires.

model tool used by 7 out of 10 teams on the current F1 grid and the FIA[5] is RaceWatch. RaceWatch is an advanced system that combines machine learning models with real-time tracking of tire degradation, fuel consumption, driver performance, and pit stop tactics. It constantly simulates essential race factors, providing strategists with refined forecasts for a wide range of race scenarios.

1.2.2 Pit Stop Tactics

A factor that influences race strategy and plays a major role in Formula 1 is the pit stop. Its execution and timing can impact the outcome of races. On average, pit stops result in a total time loss of 20-25 seconds. Thus, timing is crucial, as a wrongly-timed pit stop can have major setbacks. Pit stops allow teams to change tires, make minor adjustments to the car, or repair any damage. In 2007, a rule was introduced obliging drivers to change tire compounds at least once during a race, making one pit stop per race mandatory. This was done to make the races more interesting to watch, and resulted in an added layer of difficulty to pit stop strategy. Optimizing pit stop timing involves the simultaneous monitoring of several elements, such as tire wear, current track position, and competitors' strategies. Race strategists must stay alert and responsive throughout races, frequently adjusting their approach in real-time to seize advantages or reduce potential threats. Elements like on-track traffic, the deployment of safety cars, and shifting weather conditions all play a role in shaping pit stop tactics, making it one of the most dynamic elements in Formula 1 racing.

1.2.3 Fuel Management

Fuel management is another important aspect of race strategy, although refuelling is no longer allowed. As outlined by Motorsport Engineer (*Motorsport Engineer*), an online platform focusing on the engineering side of Formula 1, teams focus on balancing the car's fuel load to optimize performance[6] while ensuring that the car does not run out of gas. Strategists use both historical data to run simulations, such as expected race pace and fuel consumption rates, and information on the track layout to develop a fuel management plan. Moreover, race engineers must observe fuel levels throughout races and fine-tune their strategy. For example, when a safety car is introduced, engineers might want to save fuel by

---

[5] *FIA:* Fédération Internationale de l'Automobile, governing body for world motorsport.
[6] The less fuel the car has, the lighter it becomes and, as a result, the faster it laps around the circuit.

maintaining a steady pace, enabling the car to finish the race with less fuel onboard. On the other hand, if a driver is engaged in a close fight for position, the team might decide to burn more fuel to gain an advantage.

1.2.4 Tire Management

Another factor that the pit wall monitors throughout a race weekend is tire performance. Engineers observe tire wear, temperature, and degradation throughout race sessions to gather information about which tire to select and when to pit. By using predictive models, teams can measure the influence of tire strategy on race outcomes. Moreover, reviewing several variables such as track temperature and tire degradation allows teams to predict which tires will deliver the best results during a race.

2. Literature Review

This study focuses on understanding and examining the multiple ways in which game theory concepts are applied in the strategic decision-making process of Formula 1 teams. Race strategy is the ideal domain for the application of game theory, as several non-cooperative agents[7] try to select and implement a strategy that optimizes their pay-off[8]. The pay-off received is in the form of championship points, and the amount depends on the position the drivers finish the race in; the higher their placement, the more points[9] they receive. Moreover, in 2021, the head of vehicle performance of the Williams Racing[10] team (*McCabe*) stated that preparing for an F1 race is "sort of a game theory problem" itself.

F1 is a highly complex sport, where every race weekend, teams try to maximize their cars' performance to obtain the best possible result during the race. However, car performance is often not solely enough, as teams try to get the upper hand on their competitors by anticipating their decisions and applying winning strategies.

At the end of the 2010s, McLaren Racing[11] Limited (*Mulholland*) looked at the optimization of fuel consumption, pit stops, and their influence on lap times during F1 races in their paper *"Formula One Race Strategy"*. They explored how game theory and mathematical concepts such as algebra, integration, differentiation, and graphical analysis can be applied to a simplified F1 model to enhance race strategy and maximize pit stop planning. They focused on the effect that different initial amounts of fuel at the start of a race have on total race time, and calculated the additional lap time brought about by carrying more fuel. All this was done to ultimately decipher the optimal pit stop lap during a Formula 1 race.

From then onwards, many official papers and blogs have examined Formula 1 race strategy through a game-theoretic lens. Gordon McCabe (*McCabe*), in his blog post, investigated whether Nash equilibria, a state where no player benefits from changing their strategy, were applicable in racing scenarios. In a similar direction, Artem Filatov (*Filatov*), in his blog post, analyzed how mathematics and game theory can be used to model and optimize race strategy decision-making.

Over the last couple of years, simulation techniques have been used to solve the pit stop strategy optimization problem. The model created by Maria Michele Crudele et al.

---

[7] *Agents*: entities that make decisions that affect the results of the game.
[8] *Pay-off*: reward received for a certain strategy chosen in a game.
[9] *Points system*: only drivers finishing in positions 1 to 10 receive points, respectively: *25, 18, 15, 12, 10, 8, 6, 4, 2, 1.*
[10] *Williams Racing* is a British Formula 1 team founded by Sir Franck Williams and Patrick Head in 1977.
[11] *McLaren Racing* is a British Formula 1 racing team founded by Bruce McLaren in 1963.

(*Crudele*), "*Formula 1 Grand Prix Simulator: a Dynamic Game Theory Approach*", investigates F1 drivers' optimal strategies. They established a simulator where each driver is a rational decision-maker who makes their decisions based on the Nash equilibrium. The simulation considers several factors such as tire wear, speed, and drivers' experience to predict their next moves, such as whether they would overtake or defend. The simulator utilizes dynamic programming to update the status of each driver and the evolving race conditions after every lap. This resulted in a model that mirrors real-world F1 racing behaviour, where regardless of the drivers' starting position, the more experienced they are and the more their car performs, the higher the probability that they will end up dominating the race. Analogously, J. Bekker and W. Lotz (*Lotz*) established a model in their paper, "*Planning Formula One Race Strategies Using Discrete-Event Simulation*", that is meant to assist F1 teams in planning race strategies. The authors developed a simulator using discrete-event simulation[12] (DES) that mimics and replicates real-world racing events, such as overtaking maneuvers and pit stops, to help improve and maximize teams' race outcomes. They implemented real-world data on base lap times, starting grid positions, and fuel loads, to analyze the influence of various race strategies on race outcomes and the impact of random events on strategy success. Also utilizing modelling, Heilmeier et al. (*Heilmeier*) used historical data on practice and qualifying sessions, machine learning, and neural networks to optimize pit stop strategies. Correspondingly, Andrew Phillips (*Phillips*), in his blog F1metrics, published an article discussing his process of building a mathematical race simulator mirroring Formula 1 competitions. The model aimed to recreate F1 races to examine and investigate the effect of certain strategies on race outcome.

Later in the paper, taking inspiration from the above stated works, several simplified models replicating strategic decision-making during F1 races are developed, explained, and simulated.

[12] *DES definition:* method used to model processes which vary at determined time instances. It focuses on states and events, rather than tracking system states continuously.

3. Preliminary Mathematics

Mathematical and game-theoretic concepts play a fundamental role in Formula 1's strategic decision-making. In the papers covered in the previous chapter, we got a glimpse into how mathematical and game-theoretic principles, such as the Nash equilibrium and many more, are utilized in the creation of simulations and analysis of F1 race strategies. To fully grasp why these were used and why I decided to apply some of these concepts to my own models, we must first understand what they are and mean.

3.1 F1: A Non-Cooperative Game

Formula 1 can be considered a non-cooperative game, as each driver and team tries to maximize their results, without directly cooperating with rival teams. ScienceDirect (*ScienceDirect*) states that non-cooperative game theory is based on interactive strategic decision making, where agents act independently and selfishly. Players aim to find strategies that maximize their outcomes without binding agreements or cooperating with other game participants. Non-cooperative game theory furnishes insights into strategic decision-making processes among agents that share the same interest and are maximizing personal utility. However, non-cooperative games often do not provide the most efficient outcome, as they do not always ensure Pareto efficiency, a condition where no player can change strategy and be made better off without damaging another player's pay-off.

If the non-cooperative model obtains an equilibrium, it is usually displayed in the form of a Nash equilibrium. A Nash equilibrium can be defined as a combination of strategies such that no player can increase its pay-off by deviating to a different strategy while all other players' strategies remain constant. As stated by John Nash (*Nash*) himself, *"Any n-tuple of strategies, one for each player, may be regarded as a point in the product space obtained by multiplying the n strategy spaces ... One such n-tuple counters another if each player's strategy yields the highest obtainable expectation against the others' strategies… A self-countering n-tuple is called an equilibrium point."* Simply put, in a game with several agents, where each one selects a strategy, a certain set of strategies maximize the players' pay-offs. If each player in the pay-off maximizing set holds their selected strategy, then no one is willing to switch to a different strategy. This set of strategies is called the Nash equilibrium. Each player's strategy in the Nash equilibrium is therefore the best response to the strategies of other players.

Moreover, non-cooperative games can be split into 2 different groups based on the information available. There are complete information games, where all players have full knowledge of the game, and incomplete information games, where agents have limited knowledge of the game. Formula 1 is therefore an example of an incomplete information game, as drivers' strategies and preferences are hidden and not shared between teams. Due to a consideration of simplicity, I chose to create models which are complete information games where all drivers know each other's possible strategies. It reduces complexity as there are fewer unknowns, and the simulation does not need to model players' anticipation and reaction to opponents' moves.

3.2 Closed-Loop Systems

Formula 1 is full of unknowns and unexpected events during races, resulting in unpredictable race conditions. As race strategy is continuously updated in response to the constantly evolving track environment, strategic decision-making in F1 can be regarded as a closed-loop game. As mentioned by Fudenberg et al. (*Fudenberg*), in their article "*Open-Loop and Closed-Loop Equilibria in Dynamic Games with Many Players*" open-loop and closed-loop systems are two different information structures used for multi-stage dynamic games. Closed-loop models are often a more realistic representation of real-life scenarios, as players can observe and react to opponents' actions, and all past play up to the current stage is known. Information on the game is available to players, allowing them to analyze their opponents' moves and adjust their strategies accordingly in real time. As a result, agents consider the probability of opponents shifting from the equilibrium. Closed-loop equilibria are often subgame perfect, meaning that the strategy profiles represent a Nash equilibrium in all subgames of the dynamic game. This indicates that the players' flexibility to adapt their strategies throughout the game results in outcome-maximizing actions at each decision point, not just overall. Nash equilibria are therefore often stable and resist any deviations from the equilibrium. However, closed-loop systems can be complex to examine due to their dynamic nature, as the interconnection between agents, coupled with the anticipation and prediction of opponents' next moves, leads to a more complex strategic landscape. This complexity also influences the calculation of the Nash equilibrium, making it computationally more intensive and occasionally impossible to solve.

To establish a model that mimicked reality as much as possible, I decided to start by designing an F1 simulation based on a closed-loop system. The idea was to create a 2 driver model, where one driver's actions would impact the strategies chosen by the second driver. However, coding a Python script for the model turned out to be quite intricate and complex, as dynamic programming[13] was required. Mainly my limited Python knowledge, and the restricted computational power of my computer, kept me from continuing the application of dynamic programming and closed-loop systems. Thus, I decided to focus on a simpler game theoretic principle, the open-loop system, which mimics reality less but significantly decreases the computational complexity of the simulation.

3.3 Open-Loop Systems

In contrast to closed-loop models, open-loop systems are distinguished by players choosing their strategy at the beginning of the game and committing to it. Agents act on their behalf, with no information on opponents' choices. It is presupposed that players cannot detect each other's actions and strategies, and can therefore not adjust their strategies throughout the game. Due to the lack of information available, players define their actions based on their interests and objectives. According to Stewart et al.[14] (*Stewart*), open-loop Nash equilibria are not always Pareto-efficient, as other outcomes could exist from which all agents would benefit. Inefficiencies rise as a consequence of the lack of feedback during games, and players fixing their strategy at the start. Thus, players may be incentivized to deviate from their pre-chosen strategy if it becomes disadvantageous throughout the game, leading to subgame imperfection. Their initially chosen strategy might not end up being the optimal one once the model unfolds, leading to the formation of Nash equilibria that lack Pareto efficiency.

Nonetheless, the simplicity advantages received from designing an open-loop simulation significantly outweigh the associated inaccuracies. Thus, I chose to implement open-loop systems in my simulations.

[13] *Dynamic programming:* method used for solving complex problems, by breaking them down into smaller subproblems and solving them singularly.
[14] Researchers from the International Institute for Applied System Analysis.

3.4 Dynamic Games' Structure

While developing my initial models, I looked at which factors were necessary to develop the structure of dynamic games. State and action spaces, transitions, and pay-off functions play a fundamental role in defining the formal layout of strategic models. State spaces represent all possible scenarios and positions of a model at a given point in time. For example, lap 1 in a 10-lap race is considered a state of the race. At each state, players have a certain set of possible actions they can take (overtake, defend, etc.), shaping the action space. Transitions are the rules or probabilities that occur between states, determining how the model continues after an action. Each strategy or action chosen during a game result in a certain reward received, that is determined by the pay-off function. A pay-off function is a mathematical representation that assigns a value to each possible outcome of a game, quantifying each players' satisfaction associated with that result.

In the models displayed in the next chapter, I integrated the mathematical and game theoritic knowledge outlined earlier into the creation of my own F1 race simulations.

4. Simulating Race Strategies

4.1 Outline

Throughout the process of model building, I underwent multiple approaches and established various simulations to examine strategic decision-making in F1. However, to keep a logical and coherent flow in this paper, I decided to only include models that I found relevant and provide meaningful insights. Thus, I excluded models involving dynamic programming, as I ultimately decided against pursuing this path due to the computational complexity involved. I also decided not to include models based on Monte Carlo simulations due to the lack of accuracy associated with random sampling, and simulations investigating the existence of Mixed Nash equilibria, due to the lack of significant findings. It is also worth reminding that ChatGPT was used to assist me in the coding of Python scripts to run my models.

4.2 Time Difference - 2 Driver Model

To replicate real-life racing and understand strategic decision-making in Formula 1, I started by designing a simple model that takes into account a few variables that can have an impact on race outcomes. I included some of the most important factors of race strategy: tire management, fuel management, and the driver's driving style. The aim of this first model was to measure and examine the impact these three variables have on strategy choice.

I started my investigation by creating a simple model that examines the influence of tire management, fuel management, and driving styles on the time difference between two cars during a race. The model was set up in a way that replicated a race between two drivers, where the time difference between the two drivers' cars was measured and updated throughout the race. The aim was to find out how different driving styles, such as aggressive and passive driving, would impact tire degradation and fuel consumption, and how that in turn would influence state variables, such as the time difference at each lap of the race between the two drivers. Pit stops were excluded from this model for simplicity.

In this simulation, two drivers are fighting for position[15]. To establish this scenario, I set up the following state-action framework to analyse the impact different strategies have on the time difference between two drivers.

Driver 1 (leader):	Driver 2 (follower):
If Passive: - Tire degradation per lap: d1 - Fuel consumption per lap: f1 If Aggressive: - Tire degradation per lap: 2d1 - Fuel consumption per lap: 2f1 t(n) = current time difference at lap *n* t(n+1) = new time difference	If Passive: - Tire degradation per lap: d2 - Fuel consumption per lap: f2 If Aggressive: - Tire degradation per lap: 2d2 - Fuel consumption per lap: 2f2 t(n) = current time difference at lap *n* t(n+1) = new time difference

Table 1: Drivers' available strategies and their impact on the steady state variables.

I started by defining state variables, such as the time difference between the two drivers, and established a system where driving styles influence the amount of tire degradation and fuel consumed during a lap. As can be seen in Table 1 above, aggressive driving results in twice as much tire degradation and fuel consumption as passive driving for both drivers, mimicking real-world racing, where faster driving is often associated with higher tire degradation and fuel consumption. Utilizing these variables and parameters, I created formulas that test and calculate how various driving styles affect the time difference between two drivers.

I began by predefining strategies for drivers' actions. For example, for a given lap, I set that both drivers would be passive. Therefore, I created functions that would have as an output t(n+1), the new time difference after the lap was completed, and as an input, state variables: tire degradation, fuel consumption, and t(n), the current time difference. I established several formulas that calculate the new time difference (t(n+1)) between drivers at the end of a lap based on their chosen driving styles. These formulas can be seen in Figure 2 in the Appendix.

$$(d1 + f1) + t(n) - (d2 + f2) = t(n+1)$$

The formula above is an example where both drivers have predefined actions passive. Therefore, parameters d1, f1, d2, and f2 are used. This formula takes the current amount of

[15] *"fighting for position":* one driver (the follower) attempts an overtake, while the driver ahead (leader) defends his position.

fuel and tire degradation for both drivers, and correspondingly either subtracts or adds those values from the current time difference to find the new time difference. As driver 1 is set as the leading car (Table 1), and therefore wants to defend his position by extending the gap t(n) to driver 2 following behind, his tire degradation and fuel consumption (d1 + f1) are added to the current time difference t(n). Conversely, driver 2's tire degradation and fuel consumption (d2 + f2) are subtracted from t(n) as he is trying to decrease the gap to driver 1 as much as possible to attempt an overtake. Thus, this formula produces the new time difference based on the actions and driving styles chosen by the drivers during the previous lap, and each strategy results in a different outcome during the race.

4.2.1 Numerical Example

When put to the test, the model displayed interesting results. To set up the simulation, I defined both the tire degradation and fuel consumption per lap for both drivers, and set the time difference between the two drivers to 3 seconds at race start. I set the race length to 5 laps, and to increase the probability of an overtake occuring, I set that driver 2, the car behind, would have a faster car than the leader. Driver 2's tire degradation per lap is higher for both passive and aggressive driving styles compared to driver 1's (respectively 0.5 vs. 0.3, and 1 vs. 0.6), thus, resulting in faster lap times. Regarding the drivers' strategies, I randomly selected the drivers' race strategies. The following information regarding the 5 lap race simulation is displayed in Table 2 below.

Driver 1 (leader):	Driver 2 (follower):
Race strategy (From left to right, left is lap 1): P A A P A If *Passive*: - Tire deg (d1): 0.3 per lap - Fuel cons. (f1): 0.1 per lap If *Aggressive*: - Tire deg (2d1): 0.6 per lap - Fuel cons. (2f1): 0.2 per lap Race length (n=5): 5 laps Time difference at race start (t(0)) = 3 seconds t(n+1): new time difference	Race strategy (From left to right, left is lap 1): A P A A P If *Passive*: - Tire deg (d2): 0.5 per lap - Fuel cons. (f2): 0.1 per lap If *Aggressive*: - Tire deg (2d2): 1 per lap - Fuel cons. (2f2): 0.2 per lap

Table 2: Variable values and assigned race strategies for the 5 lap race simulation.

The outcomes of the 5 lap race simulation were the following, and can be seen in Table 3 below: Driver 2 reduced the time difference from 3 seconds at the start to 1.4 seconds at the end of lap 5, catching up to driver 1 in front. This illustrates that having a faster car setup significantly influences the outcome of a race, as both drivers' strategies weren't too different. Both drivers had aggressive driving styles for 3 laps of the race, and had 2 laps in which they drove passively. Thus, the major difference that allowed driver 2 to reduce the time difference to driver 1 by 1.6 seconds was his faster car setup. Furthermore, when analysing the race strategies chosen, we can see that aggressive driving results in a gain in the time difference for both drivers. For example, on lap 1, driver 1 is passive while driver 2 is aggressive. Driver 1 gains a total of 0.4 seconds during that lap, while driver 2 gains 1.2 seconds, resulting in a net gain of 0.8 seconds for driver 2. Here we can see that driving aggressively and the faster setup given to driver 2 results in him significantly decreasing the gap to driver 1. If driver 1 wants to defend his position by driving as fast as possible, the best he can do is drive aggressively, such as in laps 2, 3 and 5, where the gains made by driver 2 were reduced to -0.2 in laps 2 and 5 (driver 1 gained 0.2 seconds on driver 2, as driver 2 was driving passively) and 0.4 in lap 3 (driver 2 is driving aggressively). This highlights the fact that driver 1's best chance to defend his position against a faster driver 2 behind is driving aggressively. However, in the long term, for example, if the race length is extended to 50 laps, driver 2 will eventually overtake driver 1.

5 Lap Race Simulation	
t0 = 3seconds Driver 1's strategy: PAAPA Driver 2's strategy: APAAP	
Lap 1: *Passive - Aggressive* (d1 + f1) + t0 - (2d2 + 2f2) = t1 → (0.3 + 0.1) + 3 - (1 + 0.2) = 2.2	*t1* = 2.2 seconds
Lap 2: *Aggressive - Passive* (2d1 + 2f1) + t1 - (d2 + f2) = t2 → (0.6 + 0.2) + 2.2 - (0.5 + 0.1) = 2.4	*t2* = 2.4 seconds
Lap 3: *Aggressive - Aggressive* (2d1 + 2f1) + t2 - (2d2 + 2f2) = t3 → (0.6 + 0.2) + 2.4 - (1 + 0.2) = 2	*t3* = 2 seconds
Lap 4: *Passive - Aggressive* (d1 + f1) + t3 - (2d2 + 2f2) = t4 → (0.3 + 0.1) + 2 - (1 + 0.2) = 1.2	*t4* = 1.2 seconds
Lap 5: *Aggressive - Passive* (2d1 + 2f1) + t4 - (d2 + f2) = t5 → (0.6 + 0.2) + 1.2 - (0.5 + 0.1) = 1.4	*t5* = 1.4 seconds
End result: *t5*, the time difference between the 2 drivers at the end of the 5 lap race, is equal to **1.4 seconds**.	

Table 3: Calculations and outcomes of the 5 lap race simulation.

4.2.2 Limitations and F1 Implications

Although this model provides useful and interesting insights into F1 racing's strategic decision-making, there are quite a few limitations to it. The first and most significant limitation is that drivers do not get penalized for driving aggressively. In real-life racing, driving aggressively is linked to higher tire degradation and fuel consumption, which in the long term results in slower lap times or having to pit earlier. In the Time Difference Model, this is the opposite, as drivers are quicker when having higher tire degradation and consume more fuel, as they gain extra speed from driving aggressively. A further limitation is that the model is solely based on the time difference between the two drivers, without taking into consideration lap times. Lap times are significant outputs of driver behaviour, and by only modelling relative time difference, drivers' absolute performance is somewhat ignored.

Nonetheless, in this simulation, there are some valuable implications for strategy selection in Formula 1. This model displays that timing your strategies is significant, because choosing when to be aggressive is as relevant as choosing the frequency. Driver 2 reduces the time difference significantly more when driver 1 is passive than when he is aggressive.

Lastly, the simulation also underlines the value of models in race planning, as they allow teams to analyse several strategies before races.

4.3 Optimal Pit Stop Model

From this simulation onwards, I took the previous model's limitations into consideration, and tried to establish models that replicate real-world racing more closely. In this model, I decided to focus on solving the optimal pit stop problem, and established a simulation that finds the optimal lap to pit for a single driver completing an n-lap long race. The only factors that influence pit stop timing in this model are tire degradation, fuel consumption, race length, and the pit stop duration. All other factors, such as competitors (other drivers) and driving styles, have not been taken into consideration.

Moreover, in comparison to the previous model, some adjustments and changes were made. I established a new formula, which, in contrast to the formulas used in the Time Difference Model, does not calculate the time difference between 2 cars, but it computes the driver's lap time. The new formula can be seen below.

$$\text{Lap time } (L(n)) = 75 - (0.5d) + (0.5f)$$

In this formula, lap time is calculated based on three terms: a constant 75, 0.5d, and 0.5f. Lap time is measured in seconds, and the average race lap time in Formula 1 is 75 seconds, therefore, I chose it as the constant. The terms 0.5d and 0.5f represent, respectively the tire degradation and fuel consumption, and their impact on lap times. Tire degradation and fuel consumption accumulate in this model, mimicking real-life racing. Both parameters start at 100% and then slowly decrease as the race unfolds. On one hand, the more tires degrade, the slower the car becomes due to a decrease in traction. On the other hand, the more fuel is consumed, the lighter the car's weight, therefore, the quicker it laps. This is represented in the formula above; lap time L(n) increases when tire degradation d increases (starting at 100% → accumulation causes d to decrease, but virtually tire degradation is increasing). When fuel consumption increases, and thus f decreases (starts at 100% and decreases as fuel is consumed), lap time L(n) decreases.

Based on these assumptions, I established a model simulating a race with a single driver. The aim was to find out when the optimal pit time would be, minimizing the total time needed to complete the race. To solve this problem, I established the following race simulation displayed in Table 4 below.

Race length (n): 57 laps Lap time = L(n): 75 - (0.5d) + (0.5f) Tire degradation (d): - decreases by 3% per lap - starts at 100% - If a pit stop occurs, tire degradation goes back to 100% Fuel consumption (f): - decreases by 1.5% per lap - starts at 100% Pit stop duration (P): 20 seconds p = number of pit stops Total race time = (Σ L(n)) + pP

Table 4: Optimal Pit stop model simulation.

This model simulates a 57-lap long race scenario, where a driver wishes to minimize total race time. I set that tire degradation decreases by 3% per lap, while fuel consumption decreases by 1.5% per lap. Thus, the driver knows that he has to pit at least once, otherwise his tires will not last until the end of the race, as he will be forced to retire once his tires reach 0%. Pit stop duration was set to 20 seconds, mirroring the average time lost for a pit stop in F1. The total race time is calculated using the following formula: (Σ L(n)) + pP, which represents the sum of all lap times throughout the whole race, plus the time lost during pit stops.

This simulation aims to find the optimal pit stop strategy that allows the driver to complete the race in the least amount of time possible. To find out what the best strategy is, I established a Python script that runs all possible strategies and prints the optimal one. The code can be seen in Figure 3 in the Appendix. It calculates and examines all possible pit stop strategies for the driver, taking into account all the parameters set and displayed in Table 4. The result after running the code is that the optimal pit stop occurs on Lap 28, with a total race time of 4274 seconds (71.2 minutes). This illustrates that lap 28 is the optimal balance point, as pitting earlier would mean having higher tire degradation at the end of the race, while pitting later would mean staying out too long with worn tires, sacrificing race time. Thus, this result indicates that mid-race pit stops, approximately halfway through a 57-lap race, establish an optimal trade-off between tire degradation, fuel consumption (weight loss), and the time lost during pit stops. Moreover, although this model is a simplified version of what F1 teams use, it illustrates how predefined parameters, such as tire degradation and fuel consumption, can influence strategic decision-making both before and during a race.

4.4 Predefined Strategies Models

To understand how distinct race strategies impact total race time and car performance, I designed multiple models based on predefined strategy selection during F1 races. These simulations build upon the Optimal Pit Stop Model's core structure and enable controlled experimentation with several combinations of drivers' actions, such as aggressive, passive, and pit stops, offering insight into the trade-offs drivers face in real-life F1 racing. Three models are presented in this section, forming a complete analysis on how car setup, strategic timing, and driver interaction shape the results of Formula 1 races.

4.4.1 Different Car Performance

In this model, I developed a system that focuses on calculating a driver's total race time based on the different strategies and driving styles chosen. This was done to see how different driving styles influence lap times throughout a race. I added a second driver to this race, to compare strategies between drivers, and see which strategies used during races minimize total race time. To establish this simulation and test some strategies, I created a simple 3-lap race, where drivers' strategies were predefined. The three different possible strategies drivers could use during a lap included the two different driving styles also mentioned in the Time Difference Model, aggressive and passive, and the possibility of doing a pit stop. In a 3-lap race, where drivers have three different strategies to choose from, there are a total of 27 different pure strategies[16] per driver and 729 distinct strategy profiles[17] present. Thus, I utilized strategy vectors[18] to display the strategies picked by drivers for each

[16] *Pure strategy:* plan of action (strategy) that tells a player what to do at every step of the game.
[17] *Strategy Profile:* a collection of strategies, one for each player in the game.
[18] *Strategy vectors:* list of strategies chosen by each player.

lap before the race start: (X1, X2, X3) → X1 represents driver 1's strategy chosen for lap 1, X2 for lap 2, and X3 for lap 3. (Y1, Y2, Y3) represent the same thing for driver 2.

In this model, I established distinct formulas for lap time calculation, which vary depending on the driving style selected. In real-life racing, different strategies and driving styles result in varying lap times. For example, aggressive driving results in quicker lap times than passive driving. Thus, I chose to replicate this phenomenon and designed the following formulas for the calculation of the different strategies' lap times.

Aggressive:

Lap time (L(n)) = 70 - (0.5d) + (0.5f)

Passive:

Lap time (L(n)) = 75 - (0.5d) + (0.5f)

Pit:

Lap time (L(n)) = 75 - (0.5d) + (0.5f)

As illustrated in the formulas above, aggressive driving, compared to passive driving, results in quicker lap times. I decreased the constant in the aggressive driving formula from 75 to 70 to lower the lap time when driving aggressively, therefore symbolizing that choosing aggressive as a strategy results in faster lap times. When pitting, for simplicity, I chose that the drivers complete the lap passively, and on top of that, they lose 20 seconds during the pit stop while tire degradation goes back to 100%.

Taking into account the limitations of the Time Difference Model, I tried to implement some improvements to this model to better mimic real-life racing. Thus, I included that aggressive driving would result in higher tire degradation than passive driving, and in the long run make drivers suffer, mirroring what occurs in real-world racing. All of the relevant information describing this model is summarized in Table 5 below.

<table>
<tr><td colspan="2">Race length: 3 laps
Total race time: $\Sigma L(n)$
Tire degradation (d) and Fuel consumption (f) at the start: 100%</td></tr>
<tr><td>Driver 1:
Strategy vector: (X1, X2, X3)

Possible strategies: Aggressive (A), Passive (P), Pit (Pit)

If Aggressive:
- $L(n) = 70 - (0.5d1) + (0.5f1)$
- Tire degradation (d1): -4% per lap
- Fuel consumption (f1): -3% per lap

If Passive:
- $L(n) = 75 - (0.5d1) + (0.5f1)$
- Tire degradation (d1): -2% per lap
- Fuel consumption (f1): -2% per lap

If Pit:
- $L(n) = 75 - (0.5d1) + (0.5f1)$
- +20 seconds
- Tire degradation (d1): -2% per lap
- Fuel consumption (f1): -2% per lap
- At the end of the lap tire degradation (d1): goes back to 100%</td><td>Driver 2:
Strategy vector: (Y1, Y2, Y3)

Possible strategies: Aggressive (A), Passive (P), Pit (Pit)

If Aggressive:
- $L(n) = 70 - (0.5d2) + (0.5f2)$
- Tire degradation (d2): -5% per lap
- Fuel consumption (f2): -3% per lap

If Passive:
- $L(n)$: $75 - (0.5d2) + (0.5f2)$
- Tire degradation (d2): -3% per lap
- Fuel consumption (f2): -2% per lap

If Pit:
- $L(n)$: $75 - (0.5d2) + (0.5f2)$
- +20 seconds
- Tire degradation (d2): -3% per lap
- Fuel consumption (f2): -2% per lap
- At the end of the lap tire degradation (d2): goes back to 100%</td></tr>
</table>

Table 5: Predefined strategies model's parameter values and formulas.

As displayed in Table 5 above, driver 2 has higher tire degradation than driver 1. This was done on purpose to investigate how different car setups influence lap time and race time, and again, to mirror real-life racing more closely, as opponents' cars rarely have perfectly equal performance. The first thing I looked into was exploring how the two drivers' setups differed in total race time when both selected the same race strategy. To calculate the race and lap times, I once again created a Python script that computed total race times based on the strategies chosen by the drivers and the parameters defined in Table 5. The Python script can be found in the Appendix, Figure 4.

Thus, I decided to establish and test two identical strategy vectors for both drivers.

Strategies:

Driver 1: (P, P, P) → passive in all 3 laps of the race

Driver 2: (P, P, P) → passive in all 3 laps of the race

The results of running the above strategies on Python were the following: Driver 1 completed the race in 225 seconds, while driver 2 completed the race in 228 seconds. Due to driver 2's tire degradation disadvantage, it took him 3 seconds longer to complete the race. This underlines the fundamental role that car performance plays in race outcomes. This is further highlighted by the following example, where I set the following strategies.

Strategies:

Driver 1: (P, P, A) → passive, passive, aggressive

Driver 2: (A, A, P) → aggressive, aggressive, passive

Here, the results further highlight the fact that car performance has a large influence on race outcome. Although driver 2 drove aggressively in 2 laps while driver 1 only did so in one, they both had the same total race time of 220.5 seconds. Driver 2's higher tire degradation kept him from finishing the race in less time than driver 1, despite driving aggressively in 2 laps, while driver 1 only did so in one. This exemplifies that car performance is a major determinant of race results, as even with a better strategy it is difficult to beat your opponent's superior car.

4.4.2 Same Car Performance - Pace vs Degradation Trade-off

To focus solely on how different strategy choices affect race outcomes, I wanted to test how different strategies affect the race outcomes of two cars that have equal performance. Therefore, I slightly adapted the parameters illustrated above in Table 5 to equalize the cars' performance. The only changes made to the variables were adjusting the tire degradation and fuel consumption for both drivers in all driving styles, resulting in two equally performing cars.

This allows for an isolated investigation of strategic decision-making on race outcomes. It enabled me to conduct an analysis on how the timing of strategies affects race outcomes and which strategy combinations end up being the most beneficial for drivers over a 3 lap race. I started this process by testing the same strategy for both drivers, to be sure that the car's performance is equal.

Strategies:

Driver 1: (P, P, P) → passive in all 3 laps of the race

Driver 2: (P, P, P) → passive in all 3 laps of the race

This resulted in both drivers finishing the race in 225 seconds as expected. From here, I tried to test how setting an aggressive lap at different laps of the race would affect total race time. For example, I tested the following two strategies.

Strategies:

Driver 1: (A, P, P) → aggressive, passive, passive

Driver 2: (P, A, P) → passive, aggressive, passive

In this case, driver 1 completed the race in 221.5 seconds, while driver 2 interestingly crossed the finish line in 221 seconds. This shows that the timing of a strategy plays a crucial role. Not only does it matter whether a driver is aggressive or passive, but more importantly, when they choose to be aggressive or passive. Driver 2 completed the race in 0.5 seconds less than driver 1 because he managed his tires better. Driver 1 sacrificed his lap times in laps 2 and 3 by being aggressive at the start, resulting in higher tire degradation and worn rubber for the rest of the race. If drivers are immediately aggressive at the start of the race, they sacrifice future lap time because of significant tire degradation affecting their lap times. Thus, being aggressive at the start results in slower lap times in the future. This strategy scenario replicates real-life F1 racing, where race pace and degradation are seen as a trade-off.

Let us break this 3-lap race down lap by lap.

Lap 1: Driver 1 (A): 70.5s; Driver 2 (P): 75s

Lap 2: Driver 1 (P): 75.5s; Driver 2 (A): 70.5s

Lap 3: Driver 1 (P): 75.5s; Driver 2 (P): 75.5s

The Lap times above display the loss that driver 1 incurs in laps 2 and 3 by being aggressive in lap 1. Driver 1 completes laps 2 and 3 passively in 75.5 seconds. Contrastingly, driver 2, by being aggressive in lap 2, loses less time in total, because his first passive lap on fresher tires results in a faster lap (75 seconds) than driver 2's passive laps on used rubber. Driver 2 on lap 3 drives at the same pace as driver 1, as his tire degradation reaches the same level as driver 1's after his aggressive second lap. This underlines the importance of strategic timing in Formula 1 races, and that using the right strategy at the right time can result in gaining an upper hand on opponents. This racing scenario is further exemplified in the following example:

Strategies:

Driver 1: (A, P, P) → aggressive, passive, passive

Driver 2: (P, P, A) → passive, passive, aggressive

In this simulation, driver 1 finished the 3-lap race in 221.5 seconds, while driver 2 completed it in 220.5. This further highlights the time benefit drivers receive from driving aggressively later in the race and saving their tires at the beginning. This illustrates the benefits and importance of timing strategies correctly.

A further racing scenario I wanted to examine was the timing of pit stops in this 3-lap race, and find out when the optimal pit occurs. In the case where drivers are forced to pit at least once during the race, there are three different laps at which drivers can pit, resulting in 3 different possible strategies. For simplicity, and to isolate the effects of pit stop timing, I set all non-pit strategies to aggressive. The three different strategies are shown below.

1. (Pit, A, A)
2. (A, Pit, A)
3. (A, A, Pit)

The results were interesting, as the first strategy (Pit, A, A) was the second fastest, completing the race in 234.5 seconds. The second strategy vector, (A, Pit, A), ended up being the quickest strategy, finishing the race in 234 seconds. The last, and slowest strategy profile (A, A, Pit) completed the race in 237.5 seconds. To understand why this occurred, we must examine each strategy's lap times and see in which cases drivers tend to lose time and where they are quicker.

Laps	1. (Pit, A, A)	2. (A, Pit, A)	3. (A, A, Pit)
1	95 seconds	70.5 seconds	70.5 seconds
2	69.5 seconds	95.5 seconds	71 seconds
3	70 seconds	68 seconds	96 seconds
Total race time	234.5 seconds	234 seconds	237.5

Table 6: Optimal pit stop problem strategy analysis.

As shown in Table 6 above, the third strategy, where the pit stop is made at the end of the race, is by 3 seconds the slowest. This time loss can be justified by the high tire degradation in the first 2 laps, and the extra loss of time by pitting on the last lap. By making a pit stop on the last lap, drivers are not benefitting from the fresher tires, as the race ends once new tires are equipped.

Strategies 1 and 2 have a similar total race time, with strategy 2 being only half a second quicker. The small advantage received from strategy 2 is due to the utilization of fresh tires at the end, occurring in combination with the decrease in the amount of fuel in the car (lighter car), resulting in a quick final lap of 68 seconds. This scenario mimics real-world racing, as pitting too early in the race gives little gains, because tires degrade before the race ends. It also highlights that pitting midway through races is the quickest option, minimizing total race time.

4.4.3 An Interactive Model

One vital factor that plays a major role in Formula 1 has been excluded from the models until now: interaction. Interaction between cars is what makes motorsport and F1 racing so breathtakingly interesting to watch. Thus, I wanted to try and implement drivers' interactions into my models to see how they influence strategic decision-making.

I designed a model, taking the Predefined Strategies Model's core structure as a base, which includes interaction between 2 drivers and examines the effect of possible overtakes and corresponding penalties on strategy choice. I added a part to the Predefined Strategies Model which, on one hand, allows the driver behind to overtake more easily, but, on the other hand, penalizes both drivers for fighting for position. To replicate this real-life overtaking scenario I established the following conditions to incentivize and boost overtaking throughout the simulation.

If driver 1's total race time (total_time1) > driver 2's total race time (total_time 2), **and** total_time1 - total_time2 ≤ 5 seconds, **and** driver 1 is aggressive ("A") → **then** total_time1 increases by 5 seconds, and total_time2 increases by 15 seconds.

If driver 2's total race time (total_time2) > driver 1's total race time (total_time 1), **and** total_time2 - total_time1 ≤ 5 seconds, **and** driver 2 is aggressive ("A")→ **then** total_time2 increases by 5 seconds, and total_time1 increases by 15 seconds.

The following information is summarized in Table 7 below.

If total_time1 > total_time2 **and** total_time1 - total_time2 ≤ 5 seconds **and** Driver 1 is "A" → **then** total_time1: +5 seconds total_time2: +15 seconds	**If** total_time2 > total_time1 **and** total_time2 - total_time1 ≤ 5 seconds **and** Driver 2 is "A" → **then** total_time1: +15 seconds total_time2: +5 seconds

Table 7: Explanation of how the overtaking condition works.

This simulates the situation in which one driver is behind another, trying to attempt an overtake. Lets start by looking at the scenario where driver 1 is behind driver 2. For driver 1 to be behind driver 2 at a certain point of a race, driver 1's total race time at that point must be larger than the total race time of driver 2, thus, the inequality total_time1 > total_time2 must hold. For driver 1 to attempt an overtake on driver 2, he must be close to the car infront. In this model, to incentivize and boost overtakes, the car behind overtakes once its at a distance of 5 seconds or less from the car infront. Therefore, driver 1 overtakes driver 2 when total_time1 - total_time2 ≤ 5 seconds. Additionally, to overtake, driver 1 must be driving aggressively. If these 3 conditions are met, then driver 1 overtakes driver 2. The overtake is registered by driver 1 receiving a time penalty of 5 seconds and driver 2 obtaining a time penalty of 15 seconds. They both receive a penalty because overtaking and battling leads to a loss of time for all drivers involved, replicating the consequences of overtaking in real-life racing. However, driver 2 receives a 10 second longer penalty, allowing driver 1 to virtually pass and overtake him.

I continued by establishing a python script which simulates this model, and started to test how different strategies affect total race time and drivers' finishing position. The Python code can be found in Figure 5 in the Appendix. One should keep in mind that driver 1 and 2 have different car setups, as displayed in Table 8 below, resulting in distinct car performance.

Driver 1:	Driver 2:
If *Aggressive*: - Tire degradation (d1): -4% per lap - Fuel consumption (f1): -3% per lap	If *Aggressive*: - Tire degradation (d2): -5% per lap - Fuel consumption (f2): -3% per lap
If *Passive*: - Tire degradation (d1): -2% per lap - Fuel consumption (f1): -2% per lap	If *Passive*: - Tire degradation (d2): -3% per lap - Fuel consumption (f2): -2% per lap
If *Pit*: - Tire degradation (d1): -2% per lap - Fuel consumption (f1): -2% per lap - Tire degradation goes back to 100%	If *Pit*: - Tire degradation (d2): -3% per lap - Fuel consumption (f2): -2% per lap - Tire degradation goes back to 100%

Table 8: Different Paramters between drivers.

I began by investigating both drivers' behaviour over a five lap race where they chose the same strategy. Thus, I simulated a race in which both drivers selected aggressive as their driving style for the whole race. The results were quite surprising, as 2 overtakes occurred throughout the race.

Strategies:

Driver 1: (A, A, A, A, A) → only aggressive

Driver 2: (A, A, A, A, A) → only aggressive

The whole race can be visualized in Table 9 below.

5 Lap Race: Aggressive Strategy				
Laps	**Driver 1 Lap Time** (in seconds)	**Driver 2 Lap Time** (in seconds)	**Driver 1 Accumulated Race time** (in seconds)	**Driver 1 Accumulated Race time** (in seconds)
1 (overtake)	70.5	71	70.5 + 15 (overtake penalty)	71 + 5 (overtake penalty)
2	71	72	156.5	148
3	71.5	73	228	221
4 (overtake)	72	74	300 + 5 (overtake penalty)	310 + 15 (overtake penalty)
5	72.5	75	377.5	385

Table 9: 5 Lap race simulation.

In this simulation, driver 2 overtakes driver 1 at the end of the first lap, as driver 1 gains a slight advantage on him throughout the first lap. At the end of lap 1, driver 2 is behind driver 1 by 0.5 seconds, and all conditions for driver 2 to overtake driver 1 are met. Thus, at the end of lap 1 driver 2 overtakes him. However, in laps 2, 3, and 4 driver 1 catches up to driver 2, and at the end of lap 4, driver 1 is 5 seconds behind driver 2 and all conditions for an overtake are met, so driver 1 goes past driver 2. Driver 1 completed the race before driver 2, respectively in *377.5* seconds and *385* seconds. Due to driver 1's faster car setup and driver 2 suffering from higher tire degradation, driver 1 manages to catch up to driver 2 and ends up winning the race. Therefore, tire degradation and car performance play a major role in determining race outcomes. As driver 1 had a favourable car set up compared to driver 2 (lower tire degradation per lap) he managed to catch up to driver 2 and overtake him for the win.

A limitation to this model is driver 2's overtake at the end of lap 1. Although driver 1 had a faster first lap, driver 2 overtook him at the end of the lap because all conditions for an overtake boost were met, providing driver 2 with an unfair advantage.

The conditions established at the start (Table 7) do not exactly mirror real-world Formula 1 racing. The driver leading the race after the first lap will most probably lose his position to the driver behind, because it is uncommon to gain 5 seconds on your rival in the first lap. Therefore, the driver who completes the first lap in the lead will always be penalized and lose his position at the end of the lap when the second driver is behind by 5 seconds or less. This is quite unrealistic, as in F1 racing the driver with the faster first lap often holds his position for a long time.

4.5 Nash Equilibrium-Based Strategy Selection Models

After establishing various models where I determined drivers' strategies ex-ante, I developed simultaions in which drivers select their strategies based on best response mechanisms. The aim is to analyze and identify the best-possible strategies that drivers choose based on the strategies selected by their opponents. This was done by designing models that simulate all strategy combinations, establish pay-off matrices, and determine their Nash equilibria.

4.5.1 Best Response Model

The first model I established is the Best Response Model which builds upon the Predefined Strategies Model while retaining its core structure. The drivers have the same set of strategies to choose from, and lap times are calculated in same way aswell. The Best Response Model extends the previous one by adding systems and functions which find and examine the nash equilibria present in race simulations.

The aim of this model is to find the best response strategies for both drivers competing in a race. To find these strategies, the first step is the generation of the strategy space, which consists of all possible strategy combinations. The race strategy outcomes, represented by the corresponding total race times, are stored in a pay-off matrix, with driver 1's strategies assigned to the rows, and driver 2's strategies to the columns. Each cell of the matrix is checked, until the cell in which each driver's individual race time is minimized is found, given the other driver's strategy. This process is utilized to identify pure-strategy Nash equilibria in the race simulations. To execute this mechanism and find the drivers' best-responses to their opponents' strategies, the Python script displayed in Figure 6 in the Appendix was utilized.

Due to the computational complexity associated with the generation of all possible strategy combinations, issues started to arise. For example, in a five lap long race with three different actions drivers can choose from each lap, Python had to analyse a total of 59049 strategy profiles to find the Nash equilibrium. Thus, due to the computational intensity, from a race length of seven laps (4,782,969 different strategy profiles) onwards Python would stop working. To increase race length, I tried to decrease the amount of strategies Python would have to examine, while simultaneously making the model more realistic.

4.5.2 Dynamic Interaction Model

To solve the difficulties encountered in the Best Response Model, I made some adjustments and improved model efficiency in order to decrease computational intensity and simulate longer races. I implemented conditions and rules which not only decrease complexity, but also make the race simulations mimic real-life racing more closely.

I started by establishing a condition where strategies without a pit stop are rejected, as in Formula 1 drivers must at least pit once during a race. Moreover, other factors were limited, such as the aggressiveness of drivers. In real-life racing, F1 drivers can not drive aggressively forever, as they need to manage their resources such as tires and fuel until the end of the race.

Furthermore, during a race each driver has to manage their ERS (Energy Recovery System) battery, which they can use for extra horsepower during an overtake, or to defend their position. This battery is fully loaded at the start of the race, and when deployed uses up its power, and restores it via braking or turbocharger heat. Therefore, drivers can use this battery to their advantage, but must actively manage it as it is a finite resource. To mirror this racing scenario I set that each driver can not be aggressive for more than two consecutive laps.

Additionally, I decided to include driver interaction in this model, implementing an improved version of the Predefined Strategies Interaction Model, fixing the limitations and challenges I encountered.

In this refined model, I got rid of the overtake penalty system, and introduced a new system that tracks overtakes during a race simulation. I established a mechanism that detects overtakes based on total race time after each lap. For example, if the total race time of driver 1 is higher than the total race time of driver 2 at the end of lap 2, then driver 2 is leading the race. If at the end of the next lap (lap 3), the total race time of driver 1 is lower than the total race time of driver 2, then an overtake in lap 3 must have occurred. This system is applied throughout the whole race simulation, and keeps track of all the overtakes that take place. The Python script for this model can be found in Figure 7 in the Appendix.

4.5.2.1 Example 1

By decreasing the strategy space as a result of the conditions added to the model, it was possible to simulate races up to the length of 7 laps. Thus, I began by simulating a 7 lap race with the parameters displayed in Table 8, and examined the drivers' best responses to the other driver's chosen strategy. The results were the following.

Nash Equilibrium:

Driver 1: (A, A, Pit, A, P, A, A) → total race time: 510 seconds

Driver 2: ((A, A, Pit, A, P, A, A)) → total race time: 518 seconds

Overtakes: 0

Both drivers selected the same strategy. However, it took driver 1 8 seconds less to complete the race than driver 2. This 8 second disadvantage occurs due to driver 2's higher tire degradation throughout the race (5% vs. 4% per lap). Nonetheless, he can not do anything to improve his result with his current car setup. By picking the above strategy, he is maximising his race outcome and total race time relative to the strategy chosen by driver 1, making it a best response and part of the Nash equilibrium. Driver 2 can not improve his total race time by unilaterally changing his strategy.

4.5.2.2 Example 2

To investigate how varying levels of tire degradation and fuel consumption per lap affect the strategies selected by drivers, I attempted to alter the race simulation's parameters and examined how these changes impacted decision-making.

I decided to investigate how drivers' strategy changes in reaction to increased tire degradation and fuel consumption per lap. Thus, I decided to explore the strategies drivers would choose as best responses if they were given the car setups displayed in Table 10 below.

Driver 1:	Driver 2:
If *Aggressive*: - Tire degradation (d1): -7% per lap - Fuel consumption (f1): -3% per lap	If *Aggressive*: - Tire degradation (d2): -14% per lap - Fuel consumption (f2): -7% per lap
If *Passive*: - Tire degradation (d1): -5% per lap - Fuel consumption (f1): -2% per lap	If *Passive*: - Tire degradation (d2): -10% per lap - Fuel consumption (f2): -5% per lap
If *Pit*: - Tire degradation (d1): -5% per lap - Fuel consumption (f1): -2% per lap - Tire degradation goes back to 100%	If *Pit*: - Tire degradation (d2): -10% per lap - Fuel consumption (f2): -5% per lap - Tire degradation goes back to 100%

Table 10: Driver's car setups.

The Nash Equilbrium received was the following.

Nash Equilibrium:

Driver 1: (A, A, Pit, A, P, A, A) → total race time: 534 seconds

Driver 2: (A, A, Pit, A, Pit, A, A) → total race time: 530 seconds

Overtakes: Lap 7: driver 2 overtakes driver 1

Compared to the previous example (Example 1) both drivers have higher tire degradation per lap, and driver 2 has higher fuel consumption per lap as well. In this simulation, driver 2 has double the tire degradation per lap driver 1 has in both the aggressive and passive state (respetively 14% vs. 7% and 10% vs. 5%). Driver 2 also has higher fuel consumption per lap in both states (respectively 7% vs. 3% and 5% vs. 2%).

As displayed in the Nash equilibrium above, driver 1 chose the same strategy as in the previous example. However, due to higher tire degradation it took him 24 seconds longer to complete the 7 lap race. Driver 2 having double the tire degradation of driver 1, and a slightly higher fuel consumption, chose to implement a different race strategy. Instead of only pitting once, driver 2 made a pit stop twice, losing time to driver 1 due to the second pit stop but ultimately benefitting from new tires, and finishing the race ahead of driver 1.

To analyse how the race unfolded, I broke it down lap by lap in Table 11 below.

7 Lap Race						
Laps	**Driver 1 Strategy**	**Driver 2 Strategy**	**Driver 1 Lap Time** (in seconds)	**Driver 2 Lap Time** (in seconds)	**Driver 1 Accumulated Race time** (in seconds)	**Driver 2 Accumulated Race time** (in seconds)
1	A	A	72	73.5	72	73.5
2	A	A	74	77	146	150.5
3	Pit	Pit	80.5 + 20	84.5 + 20	246.5	255
4	A	A	68	64	314.5	319
5	P	Pit	74.5	71.5 + 20	389	410.5
6	A	A	71.5	58	460.5	468.5
7 (overtake)	A	A	73.5	61.5	534	530

Table 11: Second example, 7 Lap race simulation.

Remembering that tire degradation and fuel consumption start at 100%, and decrease throughout the race accordingly to the parameters outlined in Table 10 above, Table 12 below exhibits the changes in tire degradation and fuel consumption during the 7 lap race.

Laps	**Driver 1 Strategy**	**Driver 2 Strategy**	**Driver 1 Accumulated Tire Degradation** (in %)	**Driver 2 Accumulated Tire Degradation** (in %)	**Driver 1 Accumulated Fuel Consumption** (in %)	**Driver 2 Accumulated Fuel Consumption** (in %)
1	A	A	93	86	97	93
2	A	A	86	72	94	86
3	Pit	Pit	81	62	92	81
4	A	A	93	86	89	74
5	P	Pit	88	76	87	69
6	A	A	81	86	84	62
7	A	A	74	72	81	55

Table 12: Changes in tire degradation and fuel consumption.

Throughout this race simulation, we can see that the 2 drivers' distinct car performances resulted in different strategic decisions. In laps 1 and 2 both drivers drove aggressively, and driver 2 immediately suffered from higher tire degradation and fell behind driver 1 by 4.5 seconds at the end of lap 2 (150.5s vs. 146s). On lap 3 both drivers chose to make a pit stop, losing 20 seconds but acquiring new tires. Due to higher degradation, driver 2 had 62% of his tires left at the end of lap 3 compared to driver 1's 81%. On lap 4 both drivers chose the same strategy and drove aggressively, however, this is where driver 2 begins to catch up to driver 1. Driver 2 has a faster lap than driver 1 (64s vs. 68s), as he starts to gain from his higher fuel consumption advantage, making his car lighter every lap. On lap 4, as shown in Table 12 above, driver 2 has less fuel left in his tank than driver 1 (74% vs. 89%), making his car significantly lighter and resulting in quicker lap times. Driver 2's lap time loss due to high tire degradation is compensated in the long run by high fuel consumption. On lap 5 the drivers strategies diverge. Driver 1 selects passive, managing his tires, while driver 2 decides to make a second pit stop, in order to benefit from fresher rubber on the last two laps of the race. At the end of lap 5, driver 2 was behind driver 1 by 21.5 seconds after his pit stop, making it quite difficult for driver 2 to recover and win the race. On laps 6 and 7 both drivers choose to drive aggressively again, trying to minimize their lap times in order to win the race. Driver 1 completes the last 2 laps in 71.5 and 73.5 seconds, while driver 2 completes them in 58 and 61.5 seconds. Driver 2 laps significantly faster than driver 1 in the last 2 laps due to both the tire advantage received after pitting and the lighter car, resulting in driver 2 overtaking driver 1 on lap 7 and winning the race. This underlines the advantage driver 2 received at the end of the race from having a significantly lighter car, due to higher fuel consumption compared to driver (55% vs 81% of fuel left in the tank after lap 7) .

In this simulation driver 1 can allow himself to make fewer pit stops due to his advantageous tire efficiency. Driver 2 is obligated to pit more often due to high tire degradation, but significantly benefits from his higher fuel consumption, especially towards the last laps of the race. This model highlights how tire degradation and fuel consumption impact not only lap time, but also strategic decision-making in F1, as choosing and timing your strategies correctly based on your car's performance is crucial to beat your opponents.

4.5.2.3 Example 3

I established another simulation to investigate how strategy choice would be impacted if only driver 2's strategy from the previous example was marginally modified. I further increased driver 2's tire degradation and fuel consumption per lap for his aggressive state (d2: 14% → 17%, f2: 7% → 8%), and made passive driving more convenient by lowering tire degradation and moderately increasing fuel consumption per lap (d2: 10% → 9%, f2: 5% → 6%). The following information is summarized in Table 13 below.

Driver 1:	Driver 2:
If *Aggressive*: - Tire degradation (d1): -7% per lap - Fuel consumption (f1): -3% per lap If *Passive*: - Tire degradation (d1): -5% per lap - Fuel consumption (f1): -2% per lap If *Pit*: - Tire degradation (d1): -5% per lap - Fuel consumption (f1): -2% per lap - Tire degradation goes back to 100%	If *Aggressive*: - Tire degradation (d2): -17% per lap - Fuel consumption (f2): -8% per lap If *Passive*: - Tire degradation (d2): -9% per lap - Fuel consumption (f2): -6% per lap If *Pit*: - Tire degradation (d2): -9% per lap - Fuel consumption (f2): -6% per lap - Tire degradation goes back to 100%

Table 13: Driver's car setups.

The Nash Equilbrium for this simulation was the following.

Nash Equilibrium:

Driver 1: (A, A, Pit, A, P, A, A) → total race time: 534 seconds

Driver 2: (A, Pit, A, Pit, P, P, A) → total race time: 527 seconds

Overtakes: Lap 3: Driver 2 overtakes driver 1,

Lap 4: Driver 1 overtakes driver 2, Lap 7: Driver 2 overtakes driver 1

These adjustments done to driver 2's car setup resulted in a more dynamic and interactive race with distinct strategies selected by the driver. Driver 1, having the same parameters as in the previous model chose the same strategy, while driver 2 adjusted his strategy accordingly to the changes in tire degradation and fuel consumption. As aggressive driving became slightly less advantageous due to an increase in tire degradation, driver 2 made his pit stops earlier in the race and utilized the more beneficial passive driving style. To examine what occurred throughout the race, I looked at the drivers' lap times.

7 Lap Race						
Laps	**Driver 1 Strategy**	**Driver 2 Strategy**	**Driver 1 Lap Time** (in seconds)	**Driver 2 Lap Time** (in seconds)	**Driver 1 Accumulated Race time** (in seconds)	**Driver 2 Accumulated Race time** (in seconds)
1	A	A	72	74.5	72	74.5
2	A	Pit	74	81 + 20	146	175.5
3 (overtake)	Pit	A	80.5 + 20	67.5	246.5	243
4 (overtake)	A	Pit	68	74 + 20	314.5	337
5	P	P	74.5	62.5	389	399.5
6	A	P	71.5	64	460.5	463.5
7 (overtake)	A	A	73.5	63.5	534	527

Table 14: Third example, 7 Lap race simulation.

Laps	**Driver 1 Strategy**	**Driver 2 Strategy**	**Driver 1 Accumulated Tire Degradation** (in %)	**Driver 2 Accumulated Tire Degradation** (in %)	**Driver 1 Accumulated Fuel Consumption** (in %)	**Driver 2 Accumulated Fuel Consumption** (in %)
1	A	A	93	83	97	92
2	A	Pit	86	74	94	86
3	Pit	A	81	83	92	78
4	A	Pit	93	74	89	72
5	P	P	88	91	87	66
6	A	P	81	82	84	60
7	A	A	74	65	81	52

Table 15: Changes in tire degradation and fuel consumption.

In this simulation, as displayed in Table 14 above, more interaction and overtakes occured due the significant difference in race strategies selected between the drivers. On only 3 out of 7 laps (laps, 1, 5, and 7) both drivers selected the same driving style. Driver 2's high tire degradation, especially when driving aggressively, forced him to choose a 2 stop strategy. The 2 drivers made their pit stops at different points in time throughout the race, resulting in several overtakes and many changes in race lead. Driver 1 had an advantage in the first laps of the race due to lower tire degradation, as illustrated in Table 15 above. However, once driver 2 completed his second pit stop on lap 4, he caught up to driver 1 and ultimately overtook him on the last lap for the race win. This lap time advantage driver 2 receives at the end of the race is a consequence of the higher fuel consumption he benefits from throughout the race. Moreover, it is worth to note that once driver 2 completes his pit stop on lap 4, he does 2 consecutive passive laps. This is due to the better tire to fuel efficiency received from driving passively, and the long term tire advantage received from not degrading new tires immediately after pitting. By driving aggressively driver 2 degrades his tires 17% per lap and consumes 8% of fuel, while driving passively he burns 9% of his tires per lap and 6% of fuel. Driving passively provides a more advantageous tire degradation to fuel consumption ratio, as by not driving aggressively 8% of his tire per lap is saved, and only 2% less of fuel are consumed. Thus, for the long-term driving passively is substantially more beneficial to driver 2 than driving aggressively, as you manage your tires while consistently consuming a similar amount of fuel.

Therefore, drivers choose their strategy based on a careful balance between immediate performance and long-term efficiency. Driver 2's decision to drive passively after making his second pit stop on lap 4 reflects an effort to manage tires and ultimately optimize his overall race outcomes. This behavior highlights how strategy selection is not solely driven by lap time optimization in the short term, but also by considerations of resource management that influence competitiveness over the entire race duration.

4.5.2.4 Example 4

Lastly, I examined how making aggressive driving a lot more costly in terms of tire degradation would impact strategy choice. Therefore, I increased driver 2's tire degradation to 25% per lap, and made passive driving more beneficial, by decreasing the tire degradation per lap and increasing the fuel consumption slightly more. The aim was to see how high drivers would manage having high tire degradation when driving aggressively, whilst also having an advantageous passive option. The parameters for this simulation are displayed in Table 16 below.

Driver 1:	Driver 2:
If *Aggressive*: - Tire degradation (d1): -14% per lap - Fuel consumption (f1): -7% per lap	If *Aggressive*: - Tire degradation (d2): -25% per lap - Fuel consumption (f2): -11% per lap
If *Passive*: - Tire degradation (d1): -10% per lap - Fuel consumption (f1): -5% per lap	If *Passive*: - Tire degradation (d2): -8% per lap - Fuel consumption (f2): -6% per lap
If *Pit*: - Tire degradation (d1): -10% per lap - Fuel consumption (f1): -5% per lap - Tire degradation goes back to 100%	If *Pit*: - Tire degradation (d2): -8% per lap - Fuel consumption (f2): -6% per lap - Tire degradation goes back to 100%

Table 16: Driver's car setups.

The following Nash equilibrium was received.

Nash Equilibrium:

Driver 1: (A, A, Pit, A, Pit, A, A) → total race time: 530 seconds

Driver 2: (P, A, Pit, P, P, P, P) → total race time: 522 seconds

Overtakes: Lap 5: Driver 2 overtakes driver 1

Driver 2 completed the race in first position and did so in 8 seconds less than driver 1. Driver 1's strategy is similar to the ones we have seen in the previous examples, alternating aggressive driving with frequent pit stops, in order to take advantage of fresher tires. Contrastingly, driver 2's strategy is different to the ones we have seen so far. Driver 2, on 5 laps out of 7, drives passively, and nonetheless ends up winning the race. He only selects aggressive driving once due to the high tire degradation of 25% per lap. Thus, driver 2 frequently chose passive driving to take advantage of the significantly lower tire degradation compared to when driving aggressively (8% vs. 25% per lap). Inversely, driver 1 has a less beneficial passive driving set up than driver 2, consuming less fuel (f1: 5% vs. f2: 6% per lap) and burning more tire rubber per lap (d1: 10% vs. f1: 8%). However, driver 1 has an advantage in aggressive driving, degrading his tire significantly less compared to driver 2

(14% vs. 25% per lap). To analyse how the race unfolded, and see how driver 2 ended up winning, I looked at the overall race progression.

7 Lap Race						
Laps	**Driver 1 Strategy**	**Driver 2 Strategy**	**Driver 1 Lap Time** (in seconds)	**Driver 2 Lap Time** (in seconds)	**Driver 1 Accumulated Race time** (in seconds)	**Driver 2 Accumulated Race time** (in seconds)
1	A	P	73.5	76	73.5	76
2	A	A	77	78	150.5	154
3	Pit	Pit	84.5 + 20	84 + 20	255	258
4	A	P	64	64.5	319	322.5
5 (overtake)	Pit	P	71.5 + 20	65.5	410.5	388
6	A	P	58	66.5	468.5	454.5
7	A	P	61.5	67.5	530	522

Table 17: Fourth example, 7 Lap race simulation.

Laps	**Driver 1 Strategy**	**Driver 2 Strategy**	**Driver 1 Accumulated Tire Degradation** (in %)	**Driver 2 Accumulated Tire Degradation** (in %)	**Driver 1 Accumulated Fuel Consumption** (in %)	**Driver 2 Accumulated Fuel Consumption** (in %)
1	A	P	86	92	93	94
2	A	A	72	67	86	83
3	Pit	Pit	62	59	81	77
4	A	P	86	92	74	71
5	Pit	P	76	84	69	65
6	A	P	86	76	62	59
7	A	P	72	68	55	53

Table 18: Changes in tire degradation and fuel consumption.

Table 17 above exhibits how the race simulation unfolded, and where each driver gained and lost time. Excluding the time added to lap times due to pit stop penalties, driver 1 was quicker in every lap except lap 5. However, due to to the extra pit stop on lap 5 and the corresponding addition of 20 seconds to his total race time, he finished the race behind driver 2. Driver 2 drove a more resource-efficient strategy and stayed consistent by driving passively for the last four laps of the race after his pit stop on lap 3. Despite the seemingly “slower” strategy, he completed the race ahead due to better resource management and less pit stops. As illustrated in Table 18, driver 1’s tire degradation varies between highs and lows due to the resets after pit stops and aggressive driving. Driver 2 has a more consistent decrease in tire degradation, especially from lap 4 onwards, highlighting his efficient management of tires.

This race simulation highlights that aggressive driving results in faster lap times, but requires drivers to complete more pit stops due to higher tire degradation. Passive driving is slower, but with less needs for pit stops drivers can gain an advantage over the full length of a race. Thus, the choice and timing of strategies is more important than optimizing individual lap times or trying to be as fast as possible each lap, as displayed in this example, managing your resources effectively is also a vital factor.

Throughout these simulations, in order to best respond to opponents’ strategies, drivers tend to utilize the driving styles from which they benefit the most. In a Nash equilibrium, each player chooses an action or strategy that represents their best response to the other player’s strategy. In the last simulation, for driver 1, aggressive driving in combination with 2 pit stops was the most advantageous option, minimizing his total race time. Conversely, driver 2, having a more favourable set up for passive driving, takes advantage of it, and mainly drives passively in order to maximize his race outcome. Neither driver has an incentive to unilaterally deviate from their selected strategy, as it would not improve their pay-off of winning the race. Therefore, both drivers are driving optimally in response to one another's selected strategies, characterizing a Nash equilibrium.

5. Conclusion

F1 is a complex sport with several predictable and unpredictable variables shaping race outcomes. To control and influence race results, teams gradually rely more and more on strategy and predictive simulations. These instruments play a major role in guiding the actions drivers and teams select both prior and during races, making them vital components of the sport. Throughout this paper, strategic decision-making in Formula 1 was investigated, in order to understand how strategic choices are taken throughout races. Particular regard was given to tire degradation, fuel consumption, and pit stop tactics, factors that play a significant role in race strategy formulation. Their impact on strategic planning was analysed through a series of simulation models that aim to replicate real-life F1 race scenarios.

Although the models discussed in the previous chapter have various limitations, several interesting findings arose. The simulations underlined that optimal strategies were not always intuitive, as passive driving styles could result in better race outcomes due to fewer pit stops. Moreover, the optimal strategies drivers choose tend to vary depending on the different car performance they have at their disposal. Different levels of tire degradation and fuel consumption influence the way drivers approach their races and select strategies. For example, a car with high tire degradation tends to make pit stops more often or drive more passively than drivers with lower degradation and advantageous fuel efficiency. In the Best Response Model we saw that drivers adjust their actions based on their opponent's strategy, resulting in strategies that are mutual best responses and form a Nash equilibrium. Another key finding observed in the Predefined Strategies Model is the importance of timing strategies correctly. Even with identical car setups, strategy influences race outcomes, as timing aggressiveness optimally or pitting on the ideal lap can significantly impact drivers' total race times. In the models that include interaction we noticed drivers adapting their strategies based on the penalties and benefits administered when overtaking, resulting in models that mirror real world F1 racing to some extent.

However, the models also have considerable limitations. The most apparent is the simplicity of the race simulations, as in real-life Formula 1 racing many variables impact strategic decision-making, more than the ones utilized in my models. Furthermore, for the sake of simplicity, the simulations designed were based on complete information games, while in reality F1 races are considered incomplete information games, where drivers and teams do not reveal their strategies.

Throughout the process of designing my models I encountered various difficulties regarding the creation of the Python scripts. As I attempted to increase the race length of my simulations, I realized that the computational complexity increased as the number of strategies Python had to analyze incremented exponentially. To solve this issue I tried decreasing the number of strategies, but the maximum race length I achieved was 7 laps.

Lastly, an improvement I would suggest for future studies would be to include random events like crashes, safety cars, and mechanical failures in models, in order to mimic real world racing more closely.

6. Works Cited

Britannica. "Automobile racing | History, Types & Safety." https://www.britannica.com/sports/automobile-racing. Accessed 14 April 2025.

Catapult 2023. "F1 Strategy: How Formula 1 Teams Optimize Analysis and Racing." 25 May 2023, https://www.catapult.com/blog/formula-1-race-strategy-analysis. Accessed 14 April 2025.

Catapult 2024. "Race Analysis Software | Strategy and Circuit | Catapult." 6 Mar. 2024, www.catapult.com/solutions/racewatch. Accessed 18 April 2025.

Crudele, Michele Maria, Gabriele Del Fiume, and Alessandro Marcomini. *Formula 1 Grand Prix Simulator: A Dynamic Game Theory Approach*. DFA and DEI - University of Padova, 2021. file:///Users/ye/Downloads/FerrariGT_8.pdf. Accessed 26 April 2025.

F1. "Everything you need to know about F1 – Drivers, teams, cars, circuits and more | Formula 1®." https://www.formula1.com/en/latest/article/drivers-teams-cars-circuits-and-more-everything-you-need-to-know-about.7iQfL3Rivf1comzdqV5jwc. Accessed 12 April 2025.

Filatov, Artem. "F1 Strategy: Optimization, Probability, and Game Theory." *Medium*, 3 May 2023, medium.com/@artemfilatov_62210/f1-strategy-optimization-probability-and-game-theory-3be82abb3654. Accessed 1 May 2025.

Fudenberg, Drew, and Jean Tirole. "Open-Loop and Closed-Loop Equilibria in Dynamic Games with Many Players." *Journal of Economic Theory*, vol. 44, no. 1, Feb. 1988, pp. 1–18. *Academic Press*, https://doi.org/10.1016/0022-0531(88)90093-2. Accessed 20 May 2025.

Heilmeier, Alexander. "Virtual Strategy Engineer: Using Artificial Neural Networks for Making Race Strategy Decisions in Circuit Motorsport." *Applied Sciences*. file:///Users/ye/Downloads/applsci-10-07805.pdf. Accessed 1 May 2025.

Intrafocus. "The Data-driven Race to Victory." 18 Apr. 2024, www.intrafocus.com/2024/04/the-data-driven-race-to-victory. Accessed 14 April 2025.

Lotz, J. Bekker and W. "Planning Formula One Race Strategies Using Discrete-Event Simulation." *The Journal of the Operational Research Society*, vol. 60, no. 7, 2009, pp. 952–61. *JSTOR*, www.jstor.org/stable/40206814. Accessed 27 April 2025.

McCabe, Gordon. *Formula 1 Strategy and Nash Equilibrium*. mccabism.blogspot.com/2016/01/formula-1-strategy-and-nash-equilibrium.html. Accessed 1 May 2025.

MIA. "How Do Teams Analyse F1 Race Strategy?" *How Do Teams Analyse F1 Race Strategy? | the MIA*, 8 Feb. 2022, www.schoolofraceengineering.co.uk/blog/post/15986/how-do-teams-analyse-f1-race-strategy. Accessed 12 April 2025.

Motorsport Engineer. Admin. "How Race Strategy Works in Formula 1." 18 May 2023, motorsportengineer.net/how-race-strategy-works-in-formula-1. Accessed 17 April 2025.

Mulholland, William. "Formula One Race Strategy." *stem.org.uk*, www.stem.org.uk/system/files/elibrary-resources/legacy_files_migrated/550-14_Car_Racing.pdf. Accessed 26 April 2025.

Nash, John F. "Equilibrium Points in N -person Games." *Proceedings of the National Academy of Sciences*, vol. 36, no. 1, Jan. 1950, pp. 48–49, doi:10.1073/pnas.36.1.48. Accessed 19 May 2025.

Phillips, Andrew. "Building a Race Simulator." *f1metrics,* 3 Sept. 2015, f1metrics.wordpress.com/2014/10/03/building-a-race-simulator. Accessed 1 May 2025.

ScienceDirect. "Noncooperative Game." https://www.sciencedirect.com/topics/economics-econometrics-and-finance/noncooperative-game. Accessed 19 May 2025.

Simon, Max. "Formula One (F1) | History, Drivers, Teams, Championships, & Controversies." *Britannica*, https://www.britannica.com/sports/Formula-One-automobile-racing. Accessed 13 April 2025.

Stewart, John Q., and William W. Warntz. *Physics of Population Distribution*. International Institute for Applied Systems Analysis, 1983, https://pure.iiasa.ac.at/id/eprint/2238/7/WP-83-074.pdf. Accessed 21 May 2025.

7. Appendix

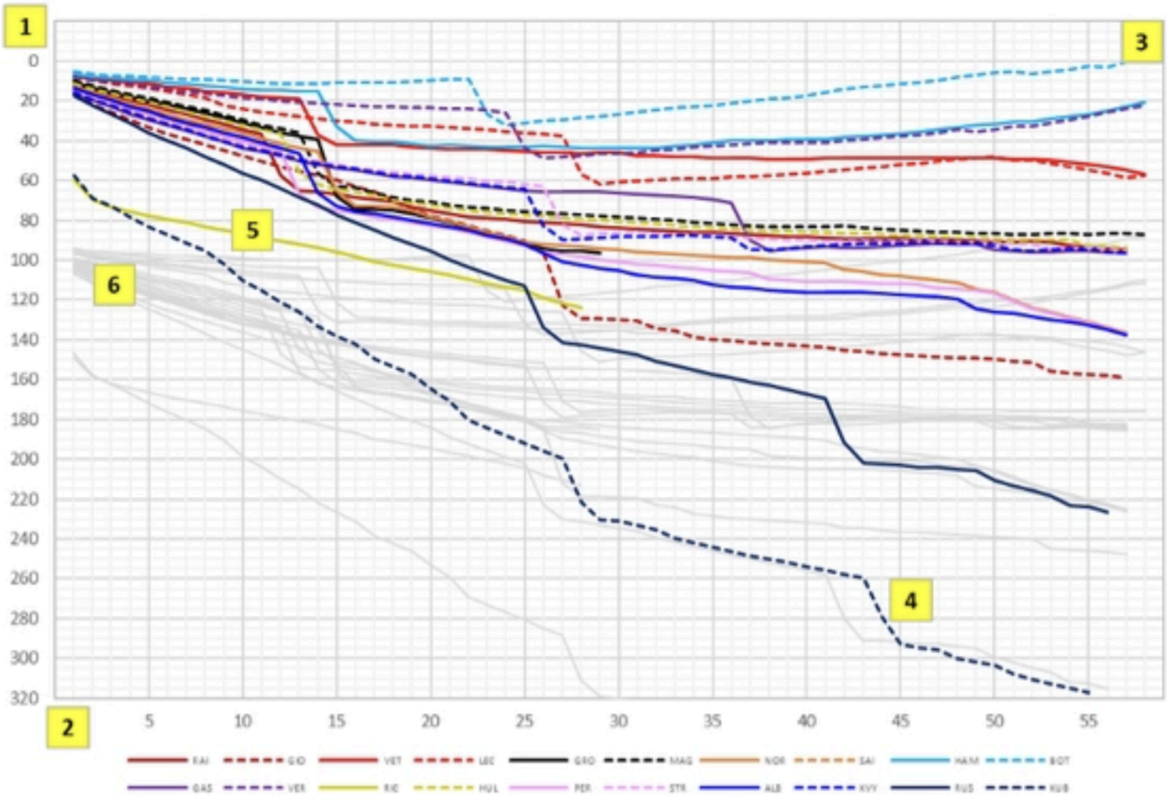

Figure 1: Gapper plot displaying the race history for the 2019 Australian Grand Prix (racewatch).

<u>Aggressive - Aggressive</u>

$$(2d1 + 2f1) + t0 - (2d2 + 2f2) = t1$$

<u>Aggressive - Passive</u>

$$(2d1 + 2f1) + t0 - (d2 + f2) = t1$$

<u>Passive-Aggressive</u>

$$(d1 + f1) + t0 - (2d2 + 2f2) = t1$$

<u>Passive-Passive</u>

$$(d1 + f1) + t0 - (d2 + f2) = t1$$

Figure 2: Time Difference - 2 Drivers Model's formulas.

```python
import numpy as np

def simulate_race(pit_lap):
    laps = 57
    tyre_deg = 100  # Tyre degradation starts at 0%
    fuel = 100      # Fuel percentage
    total_time = 0  # Total race time in seconds

    for lap in range(1, laps + 1):
        lap_time = 75 - (0.5 * tyre_deg) + (0.5 * fuel)
        total_time += lap_time

        tyre_deg -= 3  # Tyre degradation per lap
        fuel -= 1.5    # Fuel consumption per lap

        if tyre_deg <= 0:
            return float('inf')  # Car retires, invalid pit strategy

        if lap == pit_lap:
            total_time += 20  # Pit stop penalty (estimated at 20 seconds)
            tyre_deg = 100  # New tyres

    return total_time

def find_optimal_pit():
    best_pit_lap = 1
    best_time = float('inf')

    for pit_lap in range(1, 57):  # Try different pit lap strategies
        total_time = simulate_race(pit_lap)
        if total_time < best_time:
            best_time = total_time
            best_pit_lap = pit_lap

    return best_pit_lap, best_time

optimal_pit, min_time = find_optimal_pit()
print(f"Optimal pit stop: Lap {optimal_pit} with total race time: {min_time:.2f} seconds")
```

Figure 3: Optimal Pit Stop Model's Python script.

```
import itertools

def simulate_race(strategy1, strategy2):
    l1=len(strategy1)

    laps=l1

    total_time1 = 0
    total_time2 = 0

    # Initial values for both drivers
    tyre_deg1, fuel1 = 100, 100
    tyre_deg2, fuel2 = 100, 100

    for lap in range(laps):
        # Driver 1
        if strategy1[lap] == 'A':  # Aggressive
            tyre_deg1 -= 4
            fuel1 -= 3
            lap_time1 = 70 - (0.5 * tyre_deg1) + (0.5 * fuel1)  # Aggressive lap time
        elif strategy1[lap] == 'P':  # Passive
            tyre_deg1 -= 2
            fuel1 -= 2
            lap_time1 = 75 - (0.5 * tyre_deg1) + (0.5 * fuel1)  # Passive lap time
        elif strategy1[lap] == 'Pit':
            # Passive-style driving this lap
            tyre_deg1 -= 2
            fuel1 -= 2
            lap_time1 = 75 - (0.5 * tyre_deg1) + (0.5 * fuel1)
            lap_time1 += 20  # pit stop time
            tyre_deg1 = 100  # reset after lap

        total_time1 += lap_time1

        # Driver 2
        if strategy2[lap] == 'A':  # Aggressive
            tyre_deg2 -= 5
            fuel2 -= 3
            lap_time2 = 70 - (0.5 * tyre_deg2) + (0.5 * fuel2)  # Aggressive lap time
        elif strategy2[lap] == 'P':  # Passive
            tyre_deg2 -= 3
            fuel2 -= 2
            lap_time2 = 75 - (0.5 * tyre_deg2) + (0.5 * fuel2)  # Passive lap time
        elif strategy2[lap] == 'Pit':
            # Passive-style driving this lap
            tyre_deg2 -= 3
            fuel2 -= 2
            lap_time2 = 75 - (0.5 * tyre_deg2) + (0.5 * fuel2)
            lap_time2 += 20  # pit stop time
            tyre_deg2 = 100  # reset after lap

        total_time2 += lap_time2

    return total_time1, total_time2


# Generate all possible strategy combinations
strategies = ['A', 'P', 'Pit']
strategy_combinations = list(itertools.product(strategies, repeat=1))

print(strategy_combinations)

# Test specific strategies
test_strategy1 = ('P', 'P', 'P')
test_strategy2 = ('P', 'P', 'P')

time1, time2 = simulate_race(test_strategy1, test_strategy2)
print(f"Driver 1 strategy {test_strategy1}: Total time = {time1:.2f} seconds")
print(f"Driver 2 strategy {test_strategy2}: Total time = {time2:.2f} seconds")
```

Figure 4: Predefined Strategies Model's Python script.

```python
import itertools

def simulate_race(strategy1, strategy2):
    l1 = len(strategy1)
    laps = l1

    total_time1 = 0
    total_time2 = 0

    # Initial values for both drivers
    tyre_deg1, fuel1 = 100, 100
    tyre_deg2, fuel2 = 100, 100

    overtake_log = []

    for lap in range(laps):
        # Driver 1
        if strategy1[lap] == 'A':  # Aggressive
            tyre_deg1 -= 4
            fuel1 -= 3
            lap_time1 = 70 - (0.5 * tyre_deg1) + (0.5 * fuel1)
        elif strategy1[lap] == 'P':  # Passive
            tyre_deg1 -= 2
            fuel1 -= 2
            lap_time1 = 75 - (0.5 * tyre_deg1) + (0.5 * fuel1)
        elif strategy1[lap] == 'Pit':
            tyre_deg1 -= 2
            fuel1 -= 2
            lap_time1 = 75 - (0.5 * tyre_deg1) + (0.5 * fuel1)
            lap_time1 += 20  # pit stop time
            tyre_deg1 = 100

        total_time1 += lap_time1

        # Driver 2
        if strategy2[lap] == 'A':  # Aggressive
            tyre_deg2 -= 5
            fuel2 -= 3
            lap_time2 = 70 - (0.5 * tyre_deg2) + (0.5 * fuel2)
        elif strategy2[lap] == 'P':  # Passive
            tyre_deg2 -= 3
            fuel2 -= 2
            lap_time2 = 75 - (0.5 * tyre_deg2) + (0.5 * fuel2)
        elif strategy2[lap] == 'Pit':
            tyre_deg2 -= 3
            fuel2 -= 2
            lap_time2 = 75 - (0.5 * tyre_deg2) + (0.5 * fuel2)
            lap_time2 += 20  # pit stop time
            tyre_deg2 = 100

        total_time2 += lap_time2

        # Overtake check at end of lap
        if total_time1 > total_time2 and (total_time1 - total_time2) <= 5:
            total_time1 += 5
            total_time2 += 15
            overtake_log.append(f"Lap {lap+1}: Driver 1 overtakes Driver 2")
        elif total_time2 > total_time1 and (total_time2 - total_time1) <= 5:
            total_time2 += 5
            total_time1 += 15
            overtake_log.append(f"Lap {lap+1}: Driver 2 overtakes Driver 1")

    return total_time1, total_time2, overtake_log
```

```python
# Test specific strategies
test_strategy1 = ['A', 'A', 'A', 'A', 'A']
test_strategy2 = ['Pit', 'P', 'A', 'A', 'A']

time1, time2, overtakes = simulate_race(test_strategy1, test_strategy2)

print(f"Driver 1 strategy {test_strategy1}: Total time = {time1:.2f} seconds")
print(f"Driver 2 strategy {test_strategy2}: Total time = {time2:.2f} seconds")
print("Overtake log:")
for event in overtakes:
    print(event)
```

Figure 5: Predefined Strategies: An Interactive Model's Python script.

```python
import itertools
from multiprocessing import Pool, cpu_count

def simulate_race(strategy1, strategy2):
    laps = len(strategy1)
    total_time1 = 0
    total_time2 = 0
    tyre_deg1, fuel1 = 100, 100
    tyre_deg2, fuel2 = 100, 100
    overtake_log = []

    prev_leader = 1 if total_time1 < total_time2 else 2

    for lap in range(laps):
        prev_time1 = total_time1
        prev_time2 = total_time2

        # Driver 1
        if strategy1[lap] == 'A':
            tyre_deg1 -= 14
            fuel1 -= 7
            lap_time1 = 70 - 0.5 * tyre_deg1 + 0.5 * fuel1
        elif strategy1[lap] == 'P':
            tyre_deg1 -= 10
            fuel1 -= 5
            lap_time1 = 75 - 0.5 * tyre_deg1 + 0.5 * fuel1
        elif strategy1[lap] == 'Pit':
            tyre_deg1 -= 10
            fuel1 -= 5
            lap_time1 = 75 - 0.5 * tyre_deg1 + 0.5 * fuel1 + 20
            tyre_deg1 = 100
        total_time1 += lap_time1

        # Driver 2
        if strategy2[lap] == 'A':
            tyre_deg2 -= 25
            fuel2 -= 11
            lap_time2 = 70 - 0.5 * tyre_deg2 + 0.5 * fuel2
        elif strategy2[lap] == 'P':
            tyre_deg2 -= 8
            fuel2 -= 6
            lap_time2 = 75 - 0.5 * tyre_deg2 + 0.5 * fuel2
        elif strategy2[lap] == 'Pit':
            tyre_deg2 -= 8
            fuel2 -= 6
            lap_time2 = 75 - 0.5 * tyre_deg2 + 0.5 * fuel2 + 20
            tyre_deg2 = 100
        total_time2 += lap_time2

        new_leader = 1 if total_time1 < total_time2 else 2
        if new_leader != prev_leader:
            overtake_log.append(f"Lap {lap+1}: Driver {new_leader} overtakes Driver {prev_leader}")
            prev_leader = new_leader

    return total_time1, total_time2, overtake_log
```

```python
def is_valid_strategy(strategy):
    if 'Pit' not in strategy:
        return False
    count = 0
    for s in strategy:
        if s == 'A':
            count += 1
            if count > 2:
                return False
        else:
            count = 0
    return True

def simulate_wrapper(args):
    s1, s2 = args
    t1, t2, overtakes = simulate_race(list(s1), list(s2))
    return (s1, s2, t1, t2, overtakes)

def find_nash_equilibria_parallel(laps):
    strategies = ['A', 'P', 'Pit']
    all_strats = list(itertools.product(strategies, repeat=laps))
    valid_strats = [s for s in all_strats if is_valid_strategy(s)]

    pairs = list(itertools.product(valid_strats, repeat=2))
    with Pool(processes=cpu_count()) as pool:
        results = pool.map(simulate_wrapper, pairs)

    payoff_matrix = {(s1, s2): (t1, t2) for s1, s2, t1, t2, _ in results}

    best_response_1 = {}
    for s2 in valid_strats:
        min_time = float('inf')
        best_s1s = []
        for s1 in valid_strats:
            t1, _ = payoff_matrix[(s1, s2)]
            if t1 < min_time:
                min_time = t1
                best_s1s = [s1]
            elif t1 == min_time:
                best_s1s.append(s1)
        for s1 in best_s1s:
            best_response_1[(s1, s2)] = True

    best_response_2 = {}
    for s1 in valid_strats:
        min_time = float('inf')
        best_s2s = []
        for s2 in valid_strats:
            _, t2 = payoff_matrix[(s1, s2)]
            if t2 < min_time:
                min_time = t2
                best_s2s = [s2]
            elif t2 == min_time:
                best_s2s.append(s2)
        for s2 in best_s2s:
            best_response_2[(s1, s2)] = True
```

```
    nash_equilibria = []
    for s1, s2, t1, t2, overtakes in results:
        if (s1, s2) in best_response_1 and (s1, s2) in best_response_2:
            nash_equilibria.append((s1, s2, t1, t2, overtakes))

    return nash_equilibria

def run_nash_search_parallel(lap_count):
    nash_eqs = find_nash_equilibria_parallel(lap_count)

    print(f"\n\U0001F3AF Nash Equilibria for {lap_count} laps (with multiprocessing):\n")
    for i, (s1, s2, t1, t2, overtakes) in enumerate(nash_eqs, 1):
        print(f"{i}. Driver 1: {s1}, Time: {t1:.2f} | Driver 2: {s2}, Time: {t2:.2f}")
        print("   Overtake log:")
        if overtakes:
            for line in overtakes:
                print("     ➤", line)
        else:
            print("     – No overtakes")
        print()

if __name__ == "__main__":
    run_nash_search_parallel(lap_count=7)
```

Figure 6: Best Response Model’s Python script.

```python
import itertools
from multiprocessing import Pool, cpu_count

def simulate_race(strategy1, strategy2):
    laps = len(strategy1)
    total_time1 = 0
    total_time2 = 0
    tyre_deg1, fuel1 = 100, 100
    tyre_deg2, fuel2 = 100, 100
    overtake_log = []

    prev_leader = 1 if total_time1 < total_time2 else 2

    for lap in range(laps):
        prev_time1 = total_time1
        prev_time2 = total_time2

        # Driver 1
        if strategy1[lap] == 'A':
            tyre_deg1 -= 4
            fuel1 -= 3
            lap_time1 = 70 - 0.5 * tyre_deg1 + 0.5 * fuel1
        elif strategy1[lap] == 'P':
            tyre_deg1 -= 2
            fuel1 -= 2
            lap_time1 = 75 - 0.5 * tyre_deg1 + 0.5 * fuel1
        elif strategy1[lap] == 'Pit':
            tyre_deg1 -= 2
            fuel1 -= 2
            lap_time1 = 75 - 0.5 * tyre_deg1 + 0.5 * fuel1 + 20
            tyre_deg1 = 100
        total_time1 += lap_time1

        # Driver 2
        if strategy2[lap] == 'A':
            tyre_deg2 -= 5
            fuel2 -= 3
            lap_time2 = 70 - 0.5 * tyre_deg2 + 0.5 * fuel2
        elif strategy2[lap] == 'P':
            tyre_deg2 -= 3
            fuel2 -= 2
            lap_time2 = 75 - 0.5 * tyre_deg2 + 0.5 * fuel2
        elif strategy2[lap] == 'Pit':
            tyre_deg2 -= 3
            fuel2 -= 2
            lap_time2 = 75 - 0.5 * tyre_deg2 + 0.5 * fuel2 + 20
            tyre_deg2 = 100
        total_time2 += lap_time2

        new_leader = 1 if total_time1 < total_time2 else 2
        if new_leader != prev_leader:
            overtake_log.append(f"Lap {lap+1}: Driver {new_leader} overtakes Driver {prev_leader}")
            prev_leader = new_leader

    return total_time1, total_time2, overtake_log
```

```
def is_valid_strategy(strategy):
    if 'Pit' not in strategy:
        return False
    count = 0
    for s in strategy:
        if s == 'A':
            count += 1
            if count > 2:
                return False
        else:
            count = 0
    return True

def simulate_wrapper(args):
    s1, s2 = args
    t1, t2, overtakes = simulate_race(list(s1), list(s2))
    return (s1, s2, t1, t2, overtakes)

def find_nash_equilibria_parallel(laps):
    strategies = ['A', 'P', 'Pit']
    all_strats = list(itertools.product(strategies, repeat=laps))
    valid_strats = [s for s in all_strats if is_valid_strategy(s)]

    pairs = list(itertools.product(valid_strats, repeat=2))
    with Pool(processes=cpu_count()) as pool:
        results = pool.map(simulate_wrapper, pairs)

    payoff_matrix = {(s1, s2): (t1, t2) for s1, s2, t1, t2, _ in results}

    best_response_1 = {}
    for s2 in valid_strats:
        min_time = float('inf')
        best_s1s = []
        for s1 in valid_strats:
            t1, _ = payoff_matrix[(s1, s2)]
            if t1 < min_time:
                min_time = t1
                best_s1s = [s1]
            elif t1 == min_time:
                best_s1s.append(s1)
        for s1 in best_s1s:
            best_response_1[(s1, s2)] = True

    best_response_2 = {}
    for s1 in valid_strats:
        min_time = float('inf')
        best_s2s = []
        for s2 in valid_strats:
            _, t2 = payoff_matrix[(s1, s2)]
            if t2 < min_time:
                min_time = t2
                best_s2s = [s2]
            elif t2 == min_time:
                best_s2s.append(s2)
        for s2 in best_s2s:
            best_response_2[(s1, s2)] = True

    nash_equilibria = []
    for s1, s2, t1, t2, overtakes in results:
        if (s1, s2) in best_response_1 and (s1, s2) in best_response_2:
            nash_equilibria.append((s1, s2, t1, t2, overtakes))

    return nash_equilibria
```

```python
def run_nash_search_parallel(lap_count):
    nash_eqs = find_nash_equilibria_parallel(lap_count)

    print(f"\n\U0001F3AF Nash Equilibria for {lap_count} laps (with multiprocessing):\n")
    for i, (s1, s2, t1, t2, overtakes) in enumerate(nash_eqs, 1):
        print(f"{i}. Driver 1: {s1}, Time: {t1:.2f} | Driver 2: {s2}, Time: {t2:.2f}")
        print("   Overtake log:")
        if overtakes:
            for line in overtakes:
                print("    ➤", line)
        else:
            print("    – No overtakes")
        print()

if __name__ == "__main__":
    run_nash_search_parallel(lap_count=7)
```

Figure 7: Dynamic Interaction Model’s Python script.