Formula 1 Grand Prix Simulator: a Dynamic Game Theory Approach

Crudele Michele Maria (2021858) | Del Fiume Gabriele (2016821) | Marcomini Alessandro (2024286)

Abstract

In the general framework of multiplayer competitions Formula One classifies as one of the most strategic sports due to the large number of decisions each driver must take and the huge impact these actions have on drivers performances as a whole. In this work we design and implement a fully characterized simulator for F1 races which aims to provide a mathematical explanation of drivers' optimal strategies based on the rules of Game Theory. Results show how race outcomes commonly observed in real life can be explained in terms of drivers' choices being the natural best joint strategies (Nash Equilibria) of a game of rational decision makers. Therefore, our implementation sets the basis for further developments in the framework of game theory as well as a tool to combine long-term and short-term decision strategies for each pilot while seeking the optimal set of choices to achieve the best possible results.

1. Introduction

Among motorsports, Formula One is undoubtedly the most popular worldwide with millions of supporters and investments for several billions dollars every year. The hidden battle among Constructors to power their cars with state-of-the-art technology and carefully designed aerodynamics combines with drivers and teams selecting car's settings (such as tyre compound, amount of fuel, pitstops timing...) according to the track and presumed competitors' choices in each race: this makes F1 one of the most strategic games on the sport panorama. Therefore, while trying to look analytically for the optimal strategy in a given moment of the race, it becomes of interest how to properly formalize the problem in a mathematical way and look for an optimal solution taking into account the detailed information decision-makers possess: this can be achieved in the framework of Game Theory (GT).

This work aims to develop a fully working simulator for a Formula One Grand Prix founding the strategies of drivers on Game Theory models. In particular, we focus on the in-race decisions of drivers (such as whether to attack or not the driver in front) and we relate several real-life details of a race to parameters which will dynamically modify the payoffs of the players. As a result, our implementation allows an user to specify the initial conditions of the race, leading the latter to evolve according to the decisions of drivers in which "no participant can gain by a unilateral change of strategy if the strategies of the others remain unchanged".

We take especial care in modeling our full system taking into account as many details of real-life races as possible, such as tyre aging (and its impact on velocity), tyre management ability for each driver, in-race events like crashes or safety car, DRS time window for overtakes, undercut overtakes, fast pitstops and erroneous pitstops, etc. All this brings a concrete contribution in making each run of our simulation a fully-working fictitious race on its own.

Related work

As mentioned, F1 stands out as one of the most suitable real-life domains for GT applications due to the presence of many competitive agents playing in a well-defined environment. However, pre-race investigations can lead only to a limited amount of strategies to adopt, namely setting the tyre compound, the fuel amount and the pitstop schedule, along with some "B plans" to initiate in specific conditions. This framework can be treated as a static game of complete information, since all actors (drivers) must take decision simultaneously at the beginning of the race knowing the payoffs and alternatives of competitors. The head of vehicle performance of Williams F1 Racing team himself stated that preparing a Grand Prix is "sort of game theory problem" (interview in [2]) and several studies tried to relate teams' decisions to game theoretical foundations either in general cases (as in [1]) or on specific circuits (like in the Monaco GP, [5]).

In this work we take a completely new approach to estimate and study the strategic solutions of drivers inside the race, instead of just at the beginning. In fact, to the best of our knowledge the vast majority of GT studies on F1 focus on *a priori* strategies, treating the drivers' game as static, but tell us nothing on the real-time realization of the race. However, despite being the whole preparation of a F1 race a strategic game on its own, during the race itself drivers need to take tens of decisions in order to find the optimal way for success.

Here we model the decision problems at each lap as a dynamic game between pilots which eventually results in a race simulation based on game-theoretically optimal actions. The power of our architecture lays in its ability to reproduce real-life races and its high compatibility with pre-race strategic settings: therefore, combining previous results with our work we could achieve an operating and fully characterized race simulator completely relying on Game Theory optimal strategies.

2. GT model

The core of our implementation is the decision-making module which we design on the basis of a competitive model and we solve following the rules of Game Theory. We model the drivers as rational decision-makers and aim to describe the choices they face in a mathematical way, so for them to decide according to their rationality.

Consider the following frame of the race: a fast driver is approaching the driver in front of him and has to decide either to attack him to perform an overtake or to wait. In a real-life scenario he has to evaluate in a fraction of seconds many different variables: the velocity of his car and the one of the opponent, the personal skill-set of both players, the probability of the attempted overtake to be successful and the eventuality of a car crash due to the craziness or their driving style. On the other hand, if the back driver attacks, the opponent can make a choice himself: either to try to defend himself (trying to keep the position but consuming more his tyres by battling) or leaving the attacker overtake for free (if he knows he is much slower than the attacker). All the variables involved in the decisions could be in principle lap-dependent (for example, after one lap velocities will be different due to tyre consume and therefore this evaluation may lead to a different outcome).

Here we take everything into account by properly parameterizing the payoff of each driver in each situation so to enter the game-theoretical framework. While doing so, we design a dynamic game of complete imperfect information of two players: the back driver (Attacker, A), which plays first and can take actions "attack" (a) or "not to attack" (n), and the driver in front (Defender, D), who does not have to make a choice if A does not attack but decides among "defend" (d) and "not to defend" (n) when A attacks. If the joint strategy of the two players is (a, d), then a battle arises: in this case we model the outcome of the attempted overtake as a "move by Nature", i.e. the decision of an external player (Nature, N) according to which the attempt can either be successful (✓), failed (✗) or resulting in a crash (☠). Here the information is complete because each player has full knowledge of the game status (i.e., everyone knows all the past, the sequence of actions, the payoffs, the probabilities of outcomes and knows that everybody knows everything) but imperfect because the presence of Nature makes the final choice to be taken according to expectations, and not certainty.

		D: n	D: d
A	a	$u_A(a,n), u_D(a,n)$	$u_A(a,d), u_D(a,d)$
	n	$u_A(n,n), u_D(n,n)$	$u_A(n,d), u_D(n,d)$

Table 1: Normal form of the overtake game. An example of sensible numerical values in table 3. Notice that $u_A(a,d)$ and $u_D(a,d)$ are expected utilities, while $u_A(n,n) = u_A(n,d)$; the same holds for D (when A chooses n, game ends).

The extensive form of our game is depicted in figure 1: the numerical values of the payoffs at the end of each path are fine-tuned like the other parameters of the implementation and can be changed to study different behaviours of players. However, we use the following rationale to define them: if an overtake happens with no duel ((a,n)), the payoffs will be x and $-x$ for A and D respectively; if an overtake happens after a duel ((a,d)) the previous payoffs will be lowered by -0.5 to model the tyre wear caused by the fight; if A and D fight but there is no overtake, the payoffs will have to model only the tyre wear (-0.5 for both drivers); crash has to be the worst scenario, in which both the drivers obtain the lowest utility; if A does not attack, nothing happens, so the payoffs will be both equal to 0. An example of the payoffs obtained using this rationale is showed in Table 3. Of course, assuming players to be rational allows us to set the Nash Equilibrium (NE) of the game (which is also a Sub-game Perfect Equilibrium, SPE) as the realization of what happens in the in-race situation we are investigating (i.e., we make drivers follow the NE in the simulation). Starting from figure 1 we can derive the normal form of the game: this is reported in table 1.

As mentioned, N is an "hidden player" of the game which is responsible for the final outcome but has no strategic interests in the result. Its practical role is the one of randomly deciding the result of the duel. In order to do so, it is necessary to accurately model the probabilities linked to each outcome: these will depend on drivers' parameters at the $i-$th lap. We define the successful overtake probability as:

$$\alpha = \alpha\left(\boldsymbol{\theta}_A^{(i)}, \boldsymbol{\theta}_D^{(i)}\right) = \frac{v_A^{(i)} - v_D^{(i)}}{v_{max}} + \frac{s_A^{(i)} - s_D^{(i)}}{s_{max}} \quad (1)$$

While the crashing probability as:

$$\gamma = \gamma\left(\boldsymbol{\theta}_A^{(i)}, \boldsymbol{\theta}_D^{(i)}\right) = \frac{c_A^{(i)} \cdot c_D^{(i)}}{c_{max}^2} - \frac{s_A^{(i)} \cdot s_D^{(i)}}{s_{max}^2} \quad (2)$$

where $\boldsymbol{\theta}_A^{(i)}$ is the set of A's parameters at the $i-$th lap

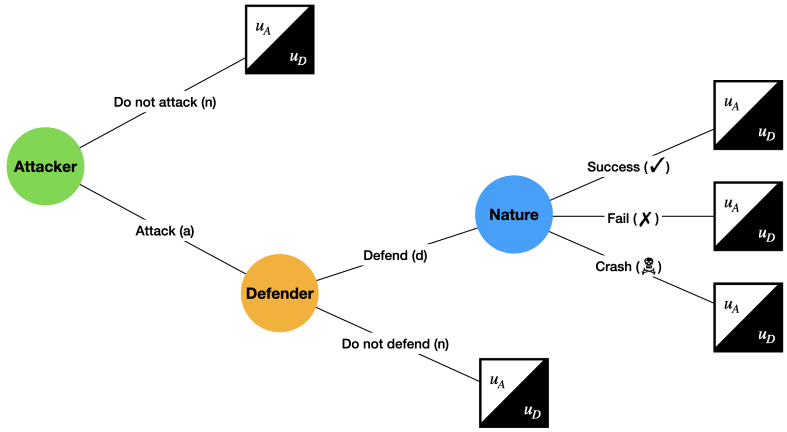


Figure 1: Overtake game for Attacker, Defender and Nature

($v_A^{(i)}$ is A's velocity at the $i-$th lap, s its skill level and c its craziness), and same applies to D. Namely, the probability of A to overtake D is large when A has higher velocity and/or skills than D, while the probability for A and D to crash depends on the craziness and the ability of the two drivers. Note that the former changes for each lap, since the velocity of each driver depends on other factors of the race (such as tyre age), while the second is determined at the start. In our implementation the parameters are fine-tuned so to have $\alpha \in [10\% : 90\%]$ and $\gamma \in [5\% : 40\%]$.

The last tool we need is a way to find the NE. This can be done by backward induction: starting from the right hand side of figure 1 (D moves second), we focus on the yellow node and take as D's choice the one which maximizes his payoff. Please notice that while the payoffs for the path (a,n) are well defined, the ones for the path (a,d) are to be taken as the expected payoffs over all possible Nature's choices with their probabilities, meaning:

$$u_A(a,d) = (1-\alpha-\gamma)\cdot u_A(a,d;\text{✗}) + \alpha \cdot u_A(a,d;\checkmark) + \gamma \cdot u_A\left(a,d;\text{☠}\right) \tag{3}$$

where for example $u_A(a,d;\checkmark)$ is the payoff of A for the strategy (a,d) when the overtake is successful. Similar equations can be written for the Defender.

The choice of D cancels out one of the two paths on the bottom yellow node. At this point, we proceed with backward induction and make A take the action that maximizes his payoff. This way we find the SPE of the game. When equal payoffs arise, we assume each player to be generous (i.e., backward induction returns the path that maximizes the opponent's payoff).

The game and solution we described above applies for an arbitrary pair of drivers where a battle for the position is possible. In the following paragraphs we describe how we managed to create a fully working simulator of a Formula One race by employing this GT model in the realization of duels. Moreover, building a simulation has the benefit of naturally varying the parametric conditions of the game, allowing us to investigate the solutions of hundreds of games with the same extensive form but different expected payoffs.

3. Dataset

We decide to run simulations for the the official 20 drivers of Formula One season 2021, which we describe by the means of different parameters crawled from 2 debate portals (Corriere dello Sport [3], SportSkeeda [4]). The parameters which used in the model are:

- Maximum velocity $[v]$: car speed in range (4,10)
- Skill $[s]$: the skill of the driver in the range (1,5)
- Craziness $[c]$: the driver's nature to take risks in the range (1,7)
- Tyre Age $[ty]$: the condition of the tyre (all drivers begin the race with $ty = 100\%$)

Moreover, each driver is identified with a three-letters code and a starting position in the grid. The attributes were modeled by us based on the data referring to the the last F1 seasons: "Velocity" and "Skill" parameters are set in relation to the constructors' ranking and drivers' final rankings (reported in [3]). Drivers belonging to the same team are driving the same car and thus have the same max speed. For what concerns the values of the "Craziness" parameter, we set them in the range from 1 to 7 according the ranking of damages (in thousands dollars) each driver caused during the season (available in [4]).

4. Simulator model and implementation

From a starting grid specified by the user, our implementation performs an iterative procedure to update the status of drivers at each lap, which takes into account several real-life details of a Grand Prix. This is composed by some fundamental modules:

- A GT-based system to determine the optimal choices of the drivers;
- An overtake actuator that performs actual overtakes on the basis of drivers' choices;
- A inter-lap routine which updates all relevant parameters at each lap (e.g. velocity, tyre age, etc).

We review them in better details.

GT-based overtake module

This module simply actuates what has been described in section 2: given a pair of drivers (A,D), we compute the values of α and γ for the specific game and apply backward induction to retrieve the optimal choices of the players.

Overtake system and actuator

While investigating real-world F1 races, one typical situation that drivers face is the one in which a potentially faster car is "covered" (and so slowed down) by an opponent. Moreover, overtakes are possible only when attacker and defender are close enough, in a process which is enhanced by what's technically called "DRS time window". For the sake of simplicity we could also state that in an overtake process the time gain of the attacker is equal to the time loss of the defender, while it is well known that when two drivers battle they stress their tyres more than when they are running free.

Inspired by all this, we model the overtake procedure as follows: we set a time window Δt_{ot} which defines the maximum absolute time difference between drivers for an overtake to be possible (i.e, for each couple of drivers D_1, D_2 with $T_2 > T_1$ we have that D_2 can attack D_1 if and only if $T_2 - T_1 < \Delta t_{ot}$) and we launch the GT module described above for each couple of drivers satisfying this constrain (with particular attention to the case in which more than two drivers lay inside the time window, see below).

On the basis of the outcome of the previous module, three scenarios can arise for the attacker-defender pair, namely the three paths of figure 1:

1. **Equilibrium (n,n) (equivalent to (n,d))**: the optimal choice for the attacker is not to attack, and therefore the game comes to end. This is the case of highly improbable overtake or highly probable crash. Here the defender will simply conserve its position and while updating the total time of the drivers we make sure that the attacker keeps a total time slightly above the one of the defender. The attacker loses the possibility to perform other attacks in this lap.
2. **Equilibrium (a,n)**: for the defender is inconvenient to defend, since either the attacker is highly favoured or the crash probability is high. The attacker has therefore the chance of a free overtake, which we model as a swap in the absolute time of attacker and defender: A has now the lowest absolute time and therefore an higher ranking. Performing overtake costs a certain tyre wear ty_{wear}^{battle} to A, while for D nothing happens since he did not fight. If there are other drivers in the attacking time window of A, he can continue to attack (GT module called again).
3. **Equilibrium (a,d)**: both attacker and defender have good reasons to fight, since expected payoffs are convenient for both. The result of the outcome is decided by Nature (via random variable generation), while both players degrade their tyres because of fight. If the overtake is successful, we swap A and D's total times and A can continue to attack; if it fails, we proceed as for the (n,n) scenario. Finally, if Nature decides for a crash, both players receive a (considerate) time penalty t_{crash} (yet, we decided not to make them retire not to change the total number of drivers).

Update to the next lap

After all possible overtakes have been considered and all players have made their choices on the basis of Game Theory techniques, we update the *status quo* of the race by considering the evolution of all the parameters subject to dynamical changes among time, namely:

- Augment all tyres age: at each lap the degradation of tyres is given by a fixed parameter ty_{wear}^{std} divided by the skill of each driver (i.e, that for all players the fixed degradation per lap is a value in the range $[1\% : 5\%]$). This represents the huge role that drivers' abilities play in the tyre management in actual races.
- Check tyres age: we control which drivers have tyres which are too ruined (under a threshold ty_{life}^{min}. If tyres are too degraded, a pitstop is forced to collect new ones.
- Identify undercutters: it is not rare that in real situations a driver who's facing difficulties in overtaking an

opponent despite being faster decides to anticipate the pitstop to have higher velocity and free path, waiting for the opponent to have his pitstop later and eventually fall behind ("undercut overtake"). In our simulator, there could be situations for which it is convenient for a driver to attack but the probability of success is still small and Nature chooses multiple times to deny the overtake. Therefore, we set a parameter f^{uc}_{max} on the maximum number of attacks a driver can fail before deciding to perform undercut (and therefore to enter pitstop).

- Perform pitstops: for players willing to pitstop we reset the 100% tyre life at the price of a slight time penalty t_{ps}. Moreover, to add randomness to our implementation, we account mechanics to make a mistake with a little probability p^{ps}_{err} in a way that penalizes the driver more (with a time $t^{err}_{ps} > t_{ps}$).

- Update drivers' velocity: at this point we calculate the new values of drivers' velocities to be used in the GT module in the next lap. These are given by the maximum velocity of each driver multiplied by his tyre age (%): this reflects the fact that that drivers with new tyres will always be quicker, while velocity decreases over time due to tyre wear. Moreover, attacking frequently ruins tyres more and so decreases velocity faster. Finally, skilled drivers will ruin their tyres more slowly and so keep an higher overall velocity.

- Update total time: finally, we update the absolute time of each driver by adding the time over this lap to the total. For each driver, the time for the single lap is calculated as the maximum between his projected time (length of the circuit divided by player's velocity at this lap, L/v_i) and the projected time of the driver in front of him, if he did not manage to overtake him. This models the fact that sometimes slower but more skilled drivers are indeed creating a bottleneck for faster drivers behind them. Moreover, here we consider also the possibility of a safety car entry with probability p_{sc}: if this happens, the distance among all players compresses up to a minimum margin equal to the starting grid, preserving positions. At each lap the ranking can be found by looking at the total time of the race, from minimum to maximum.

5. Results

Based on the simulator implementation described in the section 4, we propose here different results by varying the number of laps and drivers' starting positions. What we observe is that regardless of the starting order and for a number of laps $n \to \infty$, the drivers with the highest skill (s_i) and speed (v_i) will always find themselves at the head of race. This correctly represents what in reality happens and means that rational drivers which are aware of their superior abilities decide to attack and gain positions: this confirms the correct working of our model.

5.1. Simulations

Figure 2 reports the evolution of a Gran Prix simulation with 300 laps (parameters reported in table 3). One can observe:

- Left side: drivers in their initial positions;
- Center: driver's position, represented by lines of different colors, that can changes for each lap because of any overtake, crash or pitstop;
- Right side: drivers in their final positions.

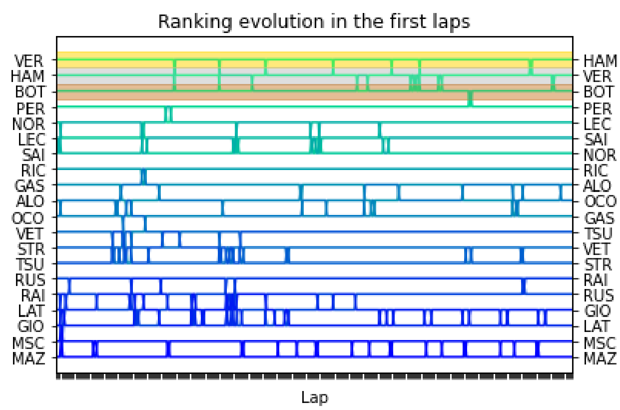


Figure 2: Simulation with 300 laps with ascending ranking

The starting grid is the one in Table 2: here we are adopting an "ascending" (ASC) configuration, namely the one in which the best drivers (according to the last seasons and with higher parameters in our code) occupy the first positions, following the one found on the website [3]. Analyzing the results, it is possible to recognize six clusters of drivers on the basis of their overtakes. For example, looking at the first group (from VER to PER) we notice that these 4 drivers only overtake each other: this is related to their v_i

Position	driver code	Position	driver code
1	VER	11	OCO
2	HAM	12	VET
3	BOT	13	STR
4	PER	14	TSU
5	NOR	15	RUS
6	LEC	16	RAI
7	SAI	17	LAT
8	RIC	18	GIO
9	GAS	19	MSC
10	ALO	20	MAZ

Table 2: Starting grid for the ascending case (ASC), i.e. best drivers start first.

and s_i, which are higher than the ones of other drivers. Figuratively, this means that ”faster pilots run a race on their own”, competing just among themselves: this is a quite common phenomenon in modern F1. Moreover, the top positions of the final ranking (on the right of Figure 2) are very similar to those represented in the starting grid: again, this strengthens the thesis expressed previously.

We run further analyses by keeping the same driver parameters but reversing their starting positions and placing best drivers last (”descending” order, DESC). In this way we want to test the simulator with a situation of full disadvantage for the drivers with high v_i and s_i. Results are reported in figure 3, while a focus on the starting part of the race is displayed in 4

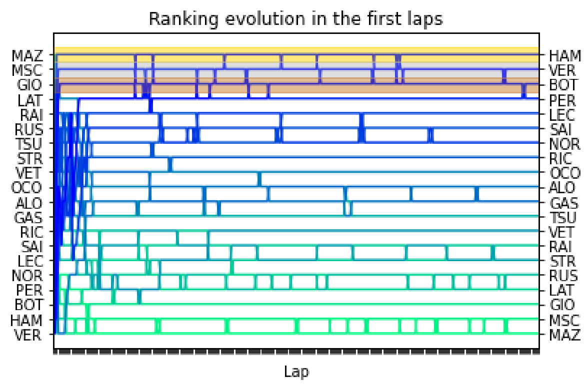


Figure 3: Simulation with 300 laps with descending ranking

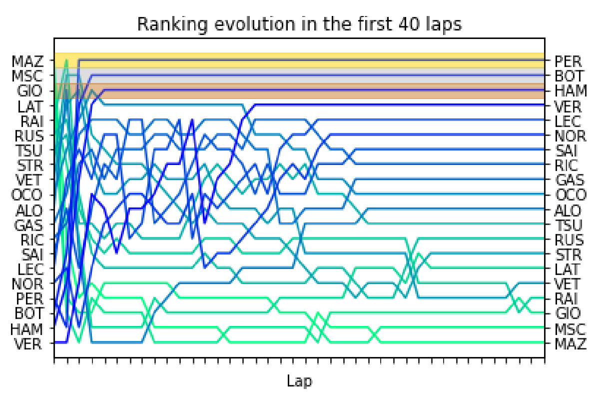


Figure 4: Focus on the first 40 laps, descending ranking

The final positions represented in Figure 3 are very similar to the previous simulation. This means that drivers with the highest parameters were able to conquer the top positions of the final grid, despite the initial disadvantage. Another interesting observation is that this process takes mainly place in the first 40 laps of the race, where we register a considerate amount of overtakes. After this point the Grand Prix goes to balance itself, with a behaviour similar to the one of figure 2. This makes us suspect the existence of a sort of ”dynamical equilibrium” configuration (more on this in the following sections).

5.2. Parameters settings

Table 3 reports the parameters settings for the simulations above.

Parameter	**Value (a.u.)**	**Parameter**	**Value (a.u.)**
Δt_{ot}	2.0	t_{crash}	7.0
L	10.0	t_{ps}	3.0
p_{sc}	1%	t_{ps}^{err}	7.0
v_i	[4.0 : 10.0]	p_{err}^{ps}	5%
s_i	[1.0 : 5.0]	ty_{life}^{min}	30%
c_i	[1.0 : 7.0]	ty_{wear}^{std}	$5\%/s_i$
α	[10% : 90%]	ty_{wear}^{battle}	2%
γ	[2% : 40%]	f_{max}^{uc}	3
Payoff		**Value (a.u.)**	
$u_A(n,n), u_D(n,n)$		$0.0; 0.0$	
$u_A(a,n), u_D(a,n)$		$3.0; -3.0$	
$u_A(a,d;✗), u_D(a,d;✗)$		$-0.5; -0.5$	
$u_A(a,d;✓), u_D(a,d;✓)$		$2.5; -3.5$	
$u_A\left(a,d;☠\right), u_D\left(a,d;☠\right)$		$-7.0; -7.0$	

Table 3: Example of parameter settings for the simulations (in arbitrary units).

5.3. Time evolution of the race

Figure 5 depicts the total (absolute) time of drivers while entering the first laps of the race. It is easy to see that while the race evolves the gap between faster and slower cars becomes more and more significant. In particular, it is noticeable how after 250 laps drivers are grouped in subsets together with others with similar parameters (as mentioned before). One evident example is the one of the two slowest drivers which have the higher total time: their time curves make a braid due to them continuously overtaking one the other (which can be seen clearly also in figure 2) and the gap with the antepenultimate driver is very large.

At this point of the race, drivers are able only to overtake opponents in the same cluster (since recovering time from drivers in the group in front is unfeasible). This is indeed quite realistic: in real races gaps arise naturally and ultimately drivers compete only with opponents driving similar power cars. We can expect that, since we modeled the drivers' velocities to be possibly dissimilar one from the other, after a large enough amount of laps we will see gaps arising between almost all the drivers, resulting in a form of dynamical equilibrium. As a matter of fact this is observed, as we further investigate in section 5.4.

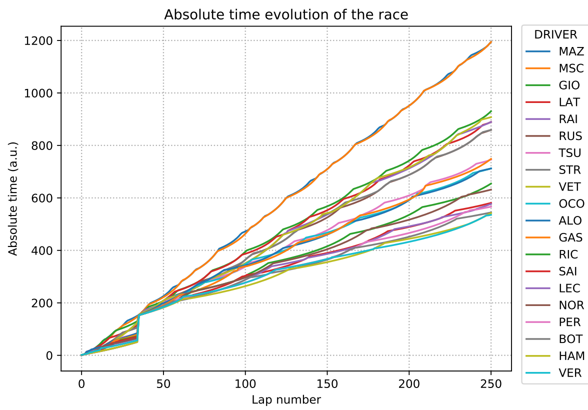


Figure 5: Total time of the race per driver (first 250 laps, ASC configuration)

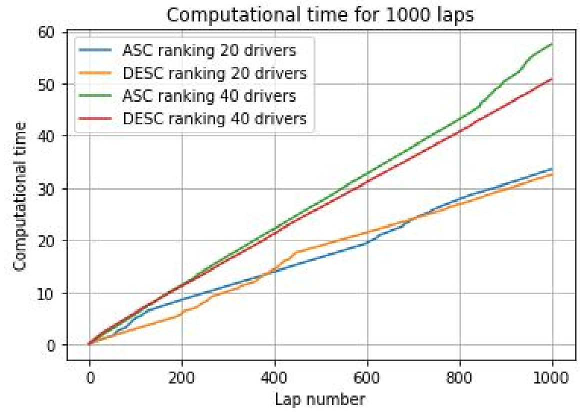


Figure 6: CPU time by varying ranking order and drivers number

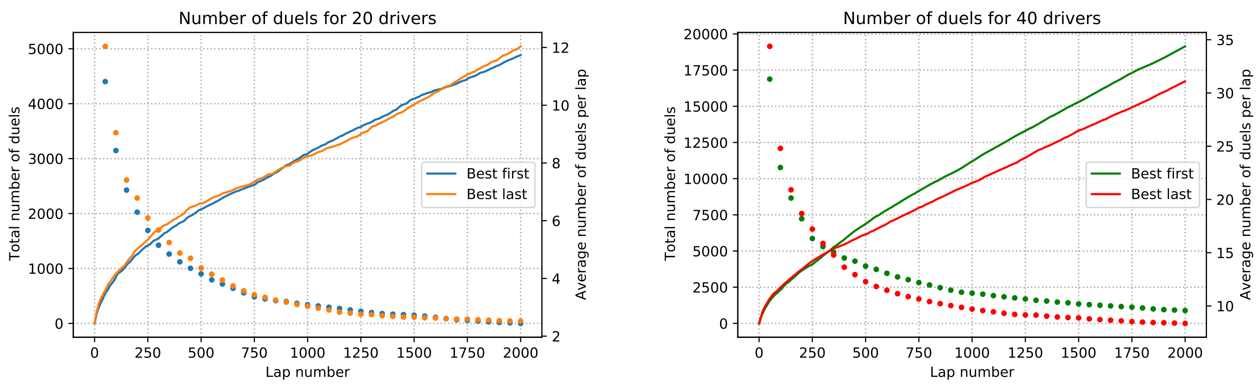
Since this behaviour of the model leads to a diminish of the number of potential duels per lap, inspired by real races we introduce a safety car mechanism that activates randomly at each lap with a probability p_{sc} which results in ”compacting the group” again: when safety car is triggered all drivers slow down until the point in which they have a minimum time gap with respect to the driver in front and in the back, preserving positions. This behaviour can be seen in figure 5 around Lap 40.

5.4. Scaling of the number of operations

While trying to understand how the complexity of our model scales, it becomes of paramount importance to estimate the contribution of the game-theoretical decision system we model and its impact on the computational cost. Figure 7 displays the total number of calls of the GT module as a function of the number of laps (continuous lines) for the original case of the twenty regular drivers and the one of forty drivers, together with the average per lap (dotted lines). The 40 drivers case is achieved by considering for each driver a ”twin” which shares the same race parameters (e.g. velocity, skill,...) and therefore undergoes similar situations in the race. It is worth mentioning that this particular configuration allows to investigate the behaviour of the decision module for $\alpha = 0$: in this case, players with same abilities are unable to overtake each other and therefore once a couple of twin drivers reunites (given the fact that they may start at far away positions in grid) they will proceed at similar speed and perform similar choices.

By looking at figure 7(a) we can point out some interesting results. First of all, we observe that the total GT decisions do not increase in a constant way overall, but present a fast-growing behaviour in the first part of the race and grow linearly after. This can be interpreted as the race reaching a dynamical equilibrium stage: after enough iterations, fastest drivers are in front and slow drivers in the back, with a gap between them which is destined to increase more and more. Indeed, this behaviour has been seen and depicted in figure 5: after many iterations the only decisions that are taken in a lap are the ones regarding drivers in the middle-back of the ranking which are kept at a distance smaller than Δt_{ot} by pitstop time penalty and fast tyre consumption. Another interesting insight is given by the almost perfect overlap between the curves for the case ”best-drivers-first” and the one ”best-drivers-last”: while one may expect that the number of operations grows faster when the fastest drivers are in the back due to the numerous attempts required to clear their way up to the top, this analysis shows that the starting grid has no significant impact on the computational side while running the simulation. Moreover, the equilibrium behaviour is reached after a similar number of iterations and proceeds with the same amount of average iterations per second.

If we now look at figure 7(b) and account for ”twin drivers”, we face a different situation. First of all, enlarging the number of drivers enlarges the number of potential duels and therefore the number of GT decisions: by looking at the y axis one can observe how values scale of a factor 2^2 by doubling the contenders. One the other hand, we can see that the shape of the curves is pretty similar, presenting the two distinct trends discussed above. Moreover, we see that here again the dynamical equilibrium is reached after the same amount of laps of the previous case, suggesting that this is a specific of the race parameters and independent of the number of drivers. Finally, a key difference lays in the fact that this time the two linear trends evolve with non-identical slopes: this is likely due to the fact that the equilibrium configurations that the system reaches are different. In fact, while we set the best drivers to be on top of the starting grid we ensure that most of the twin players find themselves one before the other at a certain point, meaning



(a) (b)

Figure 7: Number of duels for different starting conditions for 20 drivers (a) and 40 drivers(b), corresponding to the number of calls of the GT decision module. Continuous lines represent the cumulative amount, dotted lines the average per lap.

that they will always be within the overtake window and we will have to run the GT decision module for this pair at each lap. However, if the fastest drivers are in the back it can happen that while they overtake all the slowest ones they create a gap between pairs of slow twins which is eventually impossible to overcome, due to the limited time recovery they can make. As a result, fewer pairs of twins battle at each lap and the overall number of operations grows slower.

In all cases we see how the number of operations per lap is high at the beginning (race assessment transient) and decreases down to a steady number for large enough iterations. This is in accordance with the observations above.

5.5. Computational analysis

Some final considerations can be done regarding the computational time of the simulator, where the machine we use is a MACBOOK PRO (Processor: Intel® Core™ i5 dual-core 2,3 GHz; Memory: 8GB 2133 MHz LPDDR3).

We report four simulations with 1000 laps achieved by varying the drivers number (20 or 40, as in the previous section) and the starting grid (ASC or DESC configuration). Results are represented in Figure 6. Analyzing the 20 drivers case, it is easy to see that for a number of laps $n \gg 1$ the trends display the same tendency, meaning that changing the initial positions of the drivers we are not affecting the computation times. Since we register an higher number of overtakes with the DESC configuration, this result tells us that the whole overtake module we implemented has no significant impact on the CPU time. In the 40 drivers case the situation is different: the 2 lines have the same trend until around lap 800 and from this point on the computation times change slightly. Generally speaking, the trends of the computational times remark the ones of the complexity analysis of section 5.4 (figure 7).

6. Conclusions

In our work we managed to design and implement a fully working simulator of a Formula One Grand Prix based on game-theoretically optimal joint strategies for rational drivers. We successfully modeled several in-race decision variables and took into consideration real-life dynamics coming from the study of experimental races. The behaviour of players follows the one we expect: skilled drivers are conscious of their abilities and gain the top of the race by choosing to attack, despite of initial conditions.

Therefore, our work proposes an innovative tool either to analyse race prospects for real-life applications or to deepen the study of game-theoretical models in an automatized environment. In particular, further investigation could reinforce the model by including other race variables (e.g., weather conditions) and combining the decision model we proposed with some pre-game strategy study (such as tyre types, pre-arranged pitstop schedule,...). Other useful insights could be achieved by considering longer-term payoffs (e.g., championship points and a zero-sum games), redefining the overtake and crash probabilities or extending the horizon of the game to the $n-$ players case.

References

[1] Formula 1 strategy and Nash equilibrium . Accessed: 2021-11-22.

[2] F1 teams facing "game theory" strategy battle in Turkish GP. Accessed: 2021-11-22.

[3] F1: le classifiche piloti e costruttori dopo il Gp del Qatar. Accessed: 2021-11-21.

[4] Which F1 driver accrued highest crash damage costs in 2021 season? Accessed: 2021-11-22.

[5] The Game Theory of Formula 1: Winning the Monaco Grand Prix. Accessed: 2021-11-22.