Car model: Finding solutions to traffic problem

Riyad Kettaf

June 2023

Contents

1 Description of physics variables

Symbol	Description	Value
a_{max}	maximum acceleration of vehicle i	$0.73m.s^{-2}$
b	maximum deceleration	$1.67m.s^{-2}$
$v_{0,i}$	maximum speed of vehicle i	$30m.s^{-1}$
T_i	Reaction time of driver i	$1.6s$
$S_{0,i}$	Minimum distance allowed	$2m$
δ	empirical acceleration coefficient	4
v_i	vehicle i speed	$[0, v_{0,i}]$
S_i	Distance between vehicles i and i-1	$[S_{0,i}, +\infty[$
Δv	Speed difference between vehicles i and i-1	$v_{i-1} - v_i$

[1]

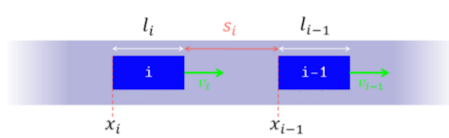


Figure 1: $S_i = x_i - x_{i-1} - l_i$

2 Introduction

Traffic congestion may seem as it is a problem only for those directly affected by it and that it's a problem that we naturally overexagerate due to our impatient

[1]Values taken from: Li, Daofei & Liu, Ao. (2021). Personalized Highway Pilot Assist Considering Leading Vehicle's Lateral Behaviours

nature. However they are the source of many problems that affect us such as :[2]

- 10 days lost in traffic per year, per person
- 15% of driving related emissions
- Emission duplication during congestion situations
- 22 billion euros in loss in France projected in 2030

3 Elementary interaction model: IDM implementation

3.1 IDM interaction

The fundamental formula used throughout this problem is as followed:

$$\frac{dv_i}{dt} = a_{max}(1 - \frac{v_i}{v_{0,i}}^{\delta} - \frac{S_{0,i} + v_i.T_i + \frac{v_i.\Delta v}{2.\sqrt{a_{max}.b}}}{S_i}^{2}))$$

To better understand what each part of this formula's use is, we'll now run specific simulations. A problem that this equation gives us is the difficulty to find the position and speed at every instant because it's a differential equation that isn't easy by any means to solve. To do that, we use Taylor's formula to approximate position and speed: For a given car with speed v and position x:

$$v_t = v_{t-1} + \Delta t * \frac{dv}{dt}(t)$$

$$x_t = x_{t-1} + \Delta t * v_t$$

With Δt being the time interval between t and $t-1$

3.2 Model coherence

First, we do a simulation on a road with a still vehicle at the end and an initially still vehicle at the start. Then, we launch the simulation and we see what happens.

Figure 2: $t = 0$

[2]Source:TomTom Traffic

Figure 3: $t = 10s$

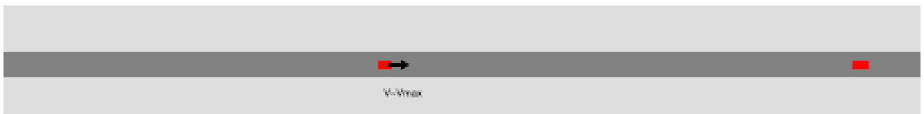


Figure 4: $t = 20s$

Figure 5: $t = 40s$

Figure 6: $t- > +\infty$

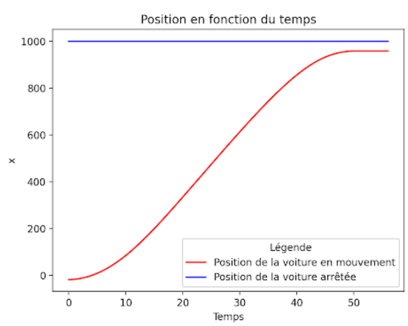


Figure 7: Time position plot

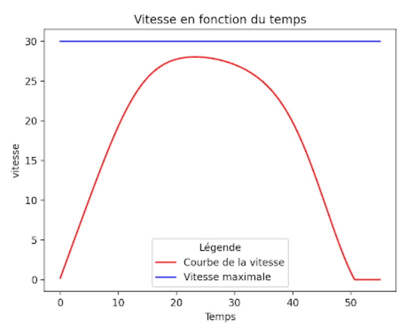


Figure 8: Time speed plot

What we can take to positive observations from this experiment: First, the car behaves as is should; it goes through the main movement phases such as accelerating steeply until reaching its top speed, then stagnating at that speed before steeply braking when approaching the still car. Then, the two different plots enable us to verify another necessary condition that our model needs to verify : the car's position and speed are continuous over time.

3.3 Single car specification

The intelligent driver model is defined for situations where you have a car behind another "leading" car, therefore, we also need to define a formula that gives the movement of the first car on the road. To do that, we take the original formula, and we imagine that the car is behind another one, but that this one is located infinitely far from the car, so $S_i = +\infty$. With this affirmation, the IDM formula is as follows:

$$\frac{dv_i}{dt} = a_{max}(1 - \frac{v_i}{v_{0,i}}^{\delta})$$

Reiterating the previous simulation without a still car at the end of the road, and applying this formula, we now have the position and speed plots below.

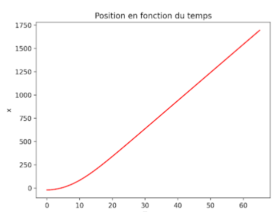


Figure 9: Time position plot

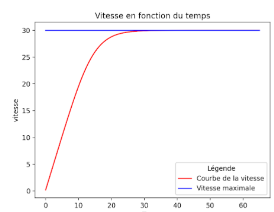


Figure 10: Time speed plot

4 Mesoscopic route system model

Now that we have defined how cars behave with one another, we also need to define lane changing, insertion and exiting lanes and system entry debit to better simulate a trafic situation

4.1 Lane changing

To simulate lane changing, we first suppose that they are instantaneous. This means that instead of rotating and crossing from the current road to the wanted road, the cars will be able immediately switch to their wanted road (as long as it's adjacent). We do that to save computing power as it can be costly to compute the rotation and render it on a large scale system.

To further accurately program lane changes, we need to establish conditions that cars need to verify to be able to "lawfully" execute a lane change. We have two conditions: First, if a car wants to switch lanes, it needs to be at least $D = 80m$ away from the closest car on that lane, Then, it also needs to make

Figure 11: Lane changing

sure that on its own lane, The distance between itself and the closest car S_i, is inferior to αv_i where α is an empirically defined constant (here $\alpha = 5$).

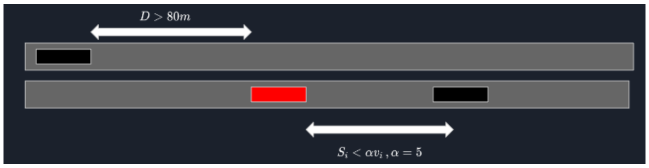


Figure 12: Lane changing

4.2 Entry and exit lanes

This part doesn't really have any theoretical value, but is interesting in terms of coding. To model entry and exit lanes, we need to adopt something similar to the previous model, but adapt it to a 2 dimensional situation. To do that, we simply seperate our formula into their x and y coordinates. And just like with straight roads, cars have to respect conditions to exit the entry lane or enter the exiting lane. Here, they consist of making sure that there's enough space on the straight road when exiting the entry lane, and making sure that there's enough space on the exit lane when exiting the straight roads

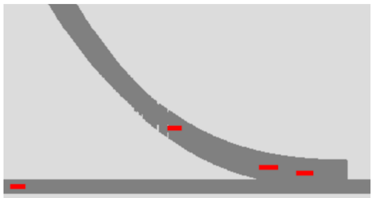


Figure 13: Entry lane

Figure 14: Exit lane

4.3 Car debit

To represent the frequency that the cars enter the system, we need to compute the probability that a car spawns at time t. To do that, we can use many probabilistic densities, but the one that we're going to choose for this problem is a normal one. We choose it because you can easily modulate the standard deviation and expectation, which will be useful in our simulations, and because of it realistically representing real life situations (rush hours).

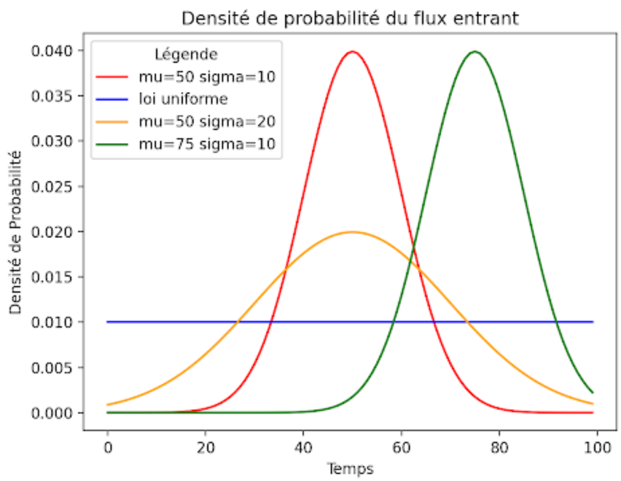


Figure 15: Normal density function

The corresponding density is as follows:

$$\forall t \in \mathbb{R}, f(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{t-\mu}{\sigma})^2} \tag{1}$$

with σ the standard deviation and μ the expectancy. (Here the density is equal to the probability because when computing, time is rendered as discrete, therefore, we technically do the integral of the density function using counting measure)

5 User behavior study

User choices can greatly impact how our system works on a macroscopic level, therefore, we also need to render user behavior. And even though user behavior isn't a variable that we can really tweak yet, with the emergence of self driving vehicles, the way that automated systems will "behave" is going to be fundamental to make sure to have high quality road systems.

5.1 IDM parameter individual variation

In the physics variables table, some values are fixed even though, in reality they depend on the user. So, if G is the function that takes as arguments an expectancy and a standard deviation and returns a random number according to a normal law that has the same parameters, then the physics variable becomes:

Symbol	Description	Value
a_{max}	maximum acceleration of vehicle i	$G(0.73, 0.2)m.s^{-2}$
b	maximum deceleration	$G(1.67, 0.2)m.s^{-2}$
$v_{0,i}$	maximum speed of vehicle i	$30m.s^{-1}$
T_i	Reaction time of driver i	$G(1.6, 0.5)s$
$S_{0,i}$	Minimum distance allowed	$2m$
δ	empirical acceleration coefficient	4
v_i	vehicle i speed	$[0, v_{0,i}]$
S_i	Distance between vehicles i and i-1	$[S_{0,i}, +\infty[$
Δv	Speed difference between vehicles i and i-1	$v_{i-1} - v_i$

5.2 Paths

If we have a road system with many cars, we won't simply make our cars travel in a single line. In real life, people are on the road because they have somewhere to go. To represent this, at every instant, every car in the system will have a destination and try (more about that in behavioral study below) to get to that destination.

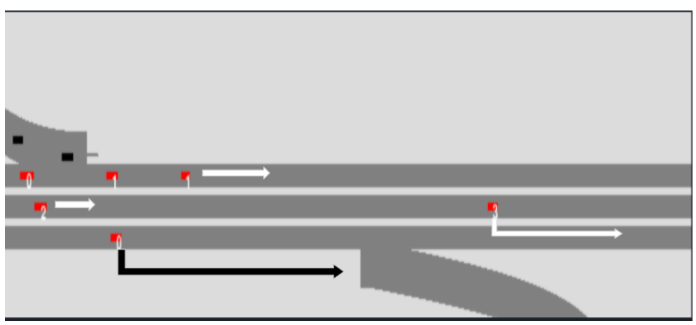


Figure 16: Path attribution example

5.3 Egoism vs Altruism

To observe behavior's effects on a road system, we'll simplify the problem by supposing that there are only two kinds of behavior: Egoistical which consists in getting to one's path independentely of the effect on other road users (while still following overtaking rules) and Altruistic which consists in letting other cars insert themselves in front of yourself by slowing down and changing lanes if and only if your lane change won't slow down vehicles on the lane that you're inserting on.

Figure 17: Red:Egoistic,Green:Altruistic

6 Simulations

6.1 Egoism study

Let's consider a road system with only egoistic drivers and see what happens:

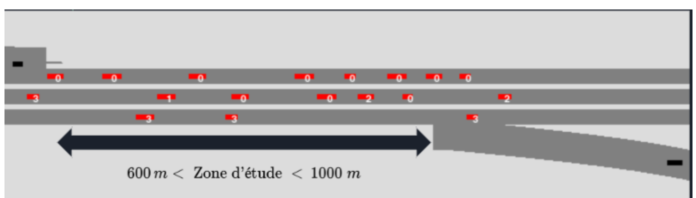


Figure 18: Zone d'études: Space interval where the amount of cars is taken from in congestion graphs

Just from figure 18 (taken at $t = 100s$), we can observe a high traffic situation and this observation is confirmed by analytical analysis.

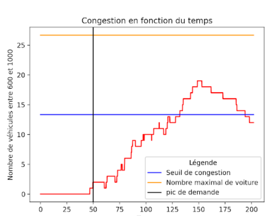


Figure 19: Number of cars in "Zone d'études" ($\mu = 40, \sigma = 20$)

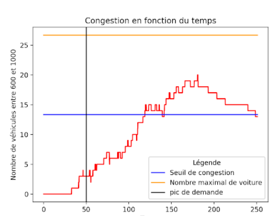


Figure 20: Number of cars in "Zone d'études" ($\mu = 50, \sigma = 30$)

We observe that independently of the probability law's parameters,for more than 75 seconds, the number of cars is over the "Seuil de congestion" which corresponds to a congestion threshold which corresponds to around 4 cars in a 400 m road segment (12/13 on aggregate for the 3 roads in figure 18).
However, we'll also notice that this behavior enables almost every user to successfully get to their destination.

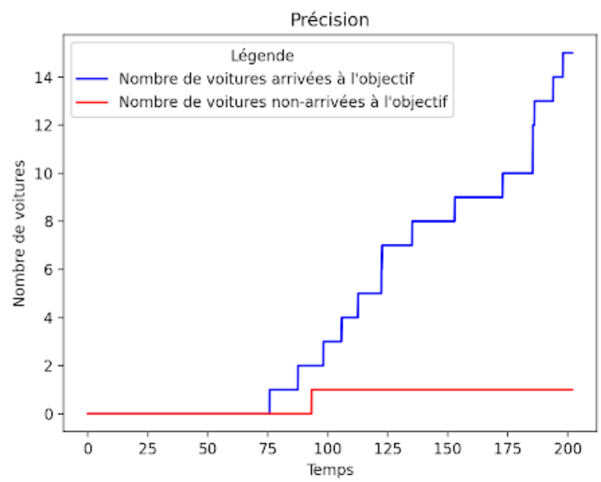
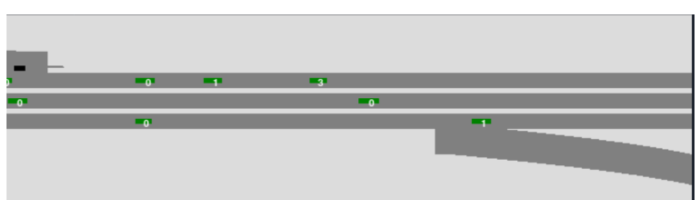


Figure 21: Number of cars getting to their destination (blue) over time vs number of them failing to (red)

6.2 Altruism

Let's reiterate the previous simulation, but instead of giving the cars an egoistic behavior, let's give them an altruistic one. We immediately see that there's much less congestion (screenshot taken at $t = 100s$)



We can confirm this analytically: However, this behavior induces one major issue, the vast majority of cars are unable to get to their destination because of their behavioral constraints

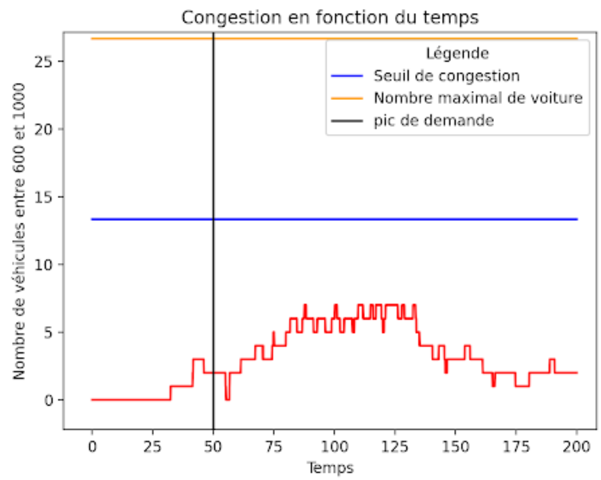


Figure 22: Number of cars in "Zone d'études" over time ($\mu = 40, \sigma = 20$)

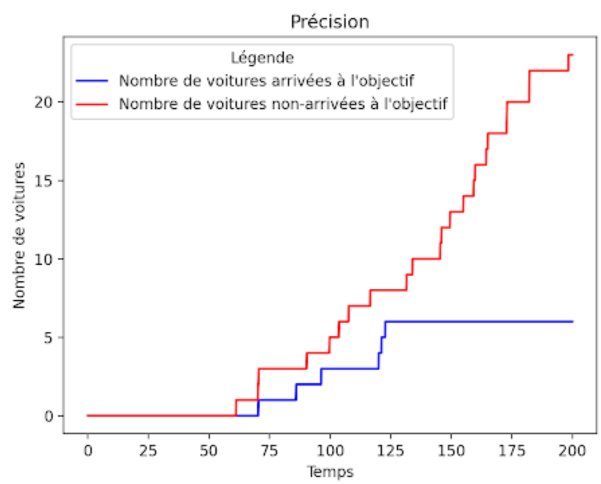


Figure 23: Number of cars getting to their destination (blue) vs number of them failing to(red)

6.3 Algorithmic behavior

In the past two subsection, cars had only one attitude and didn't change it independently of the context (where they are) that they're in. To take this into account, let's consider our road system as a graph where the nodes are the roads and arcs represent the possibility for a car to go from one node to another. Using this data structrure, let's apply the following algorithm to our

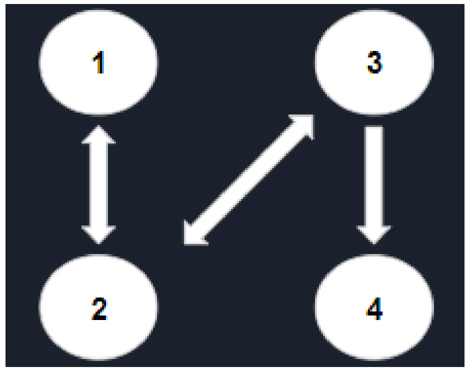


Figure 24: Road graph for the road system that we used (4 is the exit lane)

road system:

Data: Road graph $G = (S, A)$
Result: Updates vehicle behaviors
for v *in vehicles* **do**
  if $(v.road, v.objective) \in A$ **then**
    $v.behavior = egoistic$
  end
  else
    $v.behavior = altruistic$
  end
end

We now get these congestion and "accuracy" (proporition of cars getting to their destination) graphs:

We observe that the algorithm managed to keep low congestion levels and that it drastically increased "accuracy" compared to a strictly altruistic behavior scheme.

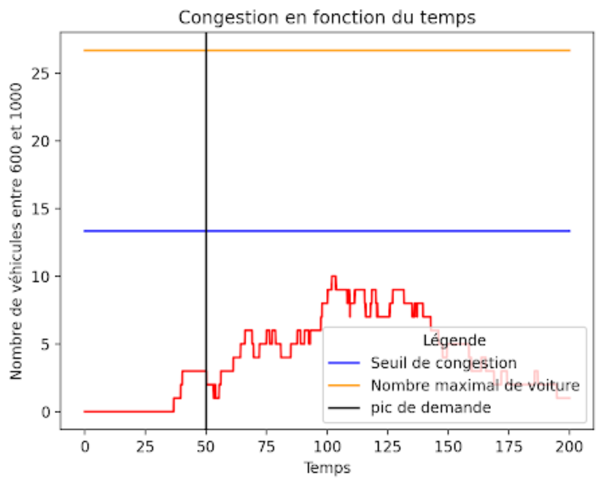


Figure 25: Number of cars in "Zone d'études" over time ($\mu = 40, \sigma = 20$)

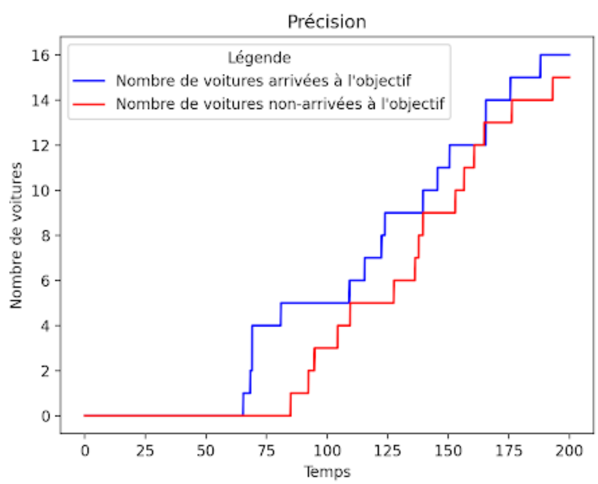


Figure 26: Number of cars getting to their destination (blue) vs number of them failing to(red)

6.4 Smart traffic lights: Intro

As said previously, user behavior related solutions can only be adopted in the future, but another variable that is susceptible to affect traffic and that we can immediately work with is traffic light management. For all the study, we'll consider the following road system where lights are very prevalent:

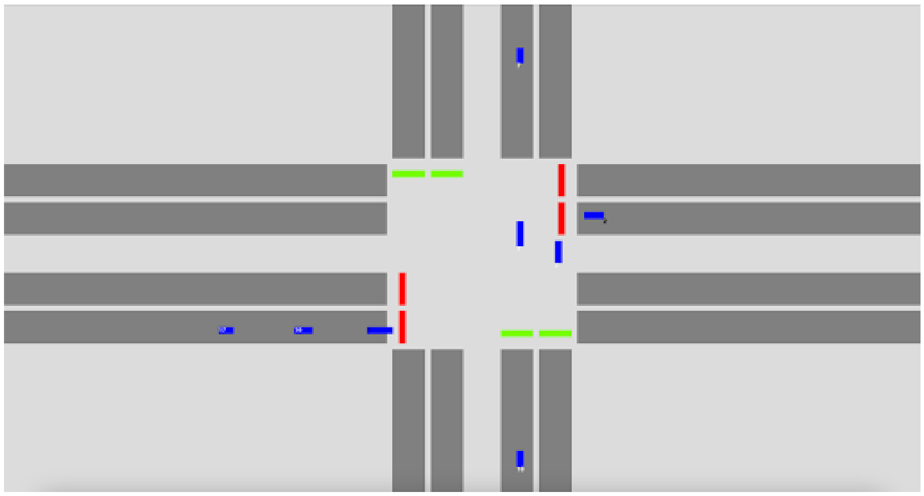


Figure 27: Road system

To see how we can optimize the way lights work, we need to see how they've mainly been working so far; Here's an algorithm that describes it:

Data: Road:Timer,color (green_time,yellow_time,red_time):duration values for each color
Result: Updates light color
if *road.timer* $<$ *green_time* **then**
  road.color=green
else if *road.timer* $<$ *yellow_time* **then**
  road.color=yellow
else if *road.timer* $<$ *red_time* **then**
  road.color=red
else
  road.timer=0
end

Using this traffic light algorithm on figure 27, we get the following congestion graph:

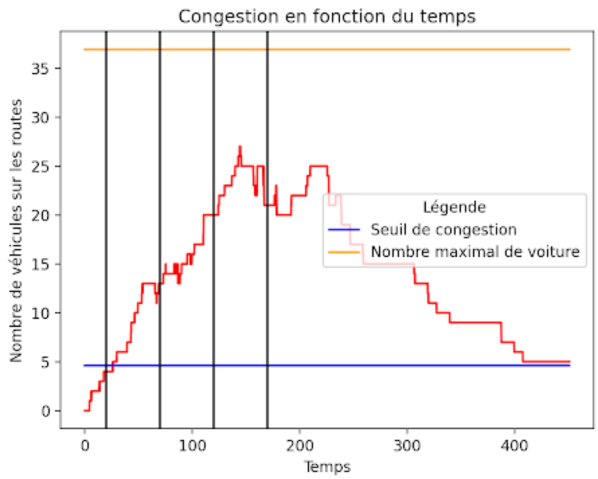


Figure 28: Number of cars in road system over time (black bands represent expectancy values for the different roads (direction))

6.5 Smart traffic lights: Adaptative algorithm

An issue with the previous algorithm is that it didn't take into account what was happening on the roads, and it could to situations where you had many cars waiting for a green light, while on the perpendicular road, the light was green but there wasn't even any cars on the road. To solve that, let's consider that our roads have n sensors each (aligned) that can each detect or not if there's a car, and let's consider the following algorithm with its correponding congestion graph:

```
Data: Road:Timer,color (green_time,yellow_time,red_time):duration values
      for each color
Result: Updates light color
green_time=0 ;
i=0 ;
while i < n do
    if c_i then
        i+=1 ;
        green_time+=10;
    else
        while Opposing light is green do
            Wait;
        end
    end
    road.timer=0 ;
    road.color=green ;
    while road.timer < green_time do
        road.color=green ;
    end
    while road.timer < green_time + yellow_time do
        road.color=yellow ;
    end
    while road.timer < green_time + yellow_time + red_time do
        road.color=red ;
    end
end
```

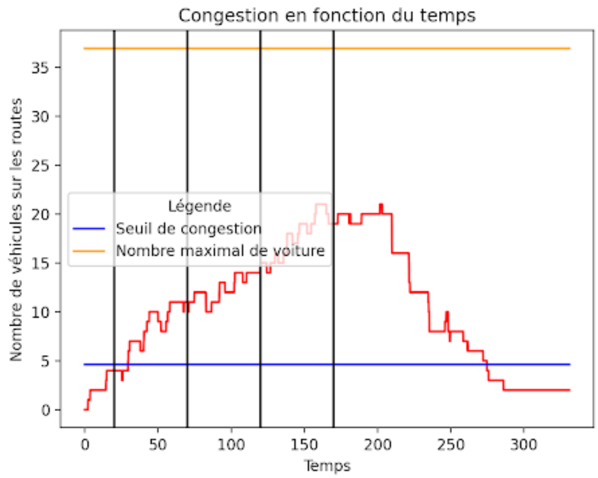


Figure 29: Number of cars in road system over time, in comparison with figure 28,during peak traffic, there are less cars and the system loses it's cars way faster (400s+ vs 200s)

7 conclusion

We can conclude two things from the simulations done in this article: Over the long term, to reduce traffic congestion in any type of road system (especially highways), modulating driving behavior algorithms for self driving cars could be interesting. Moreover over the short term, using smart traffic lights (with sensors for instance) is a realistic and very effective solutions to solve traffic light related congestion.