



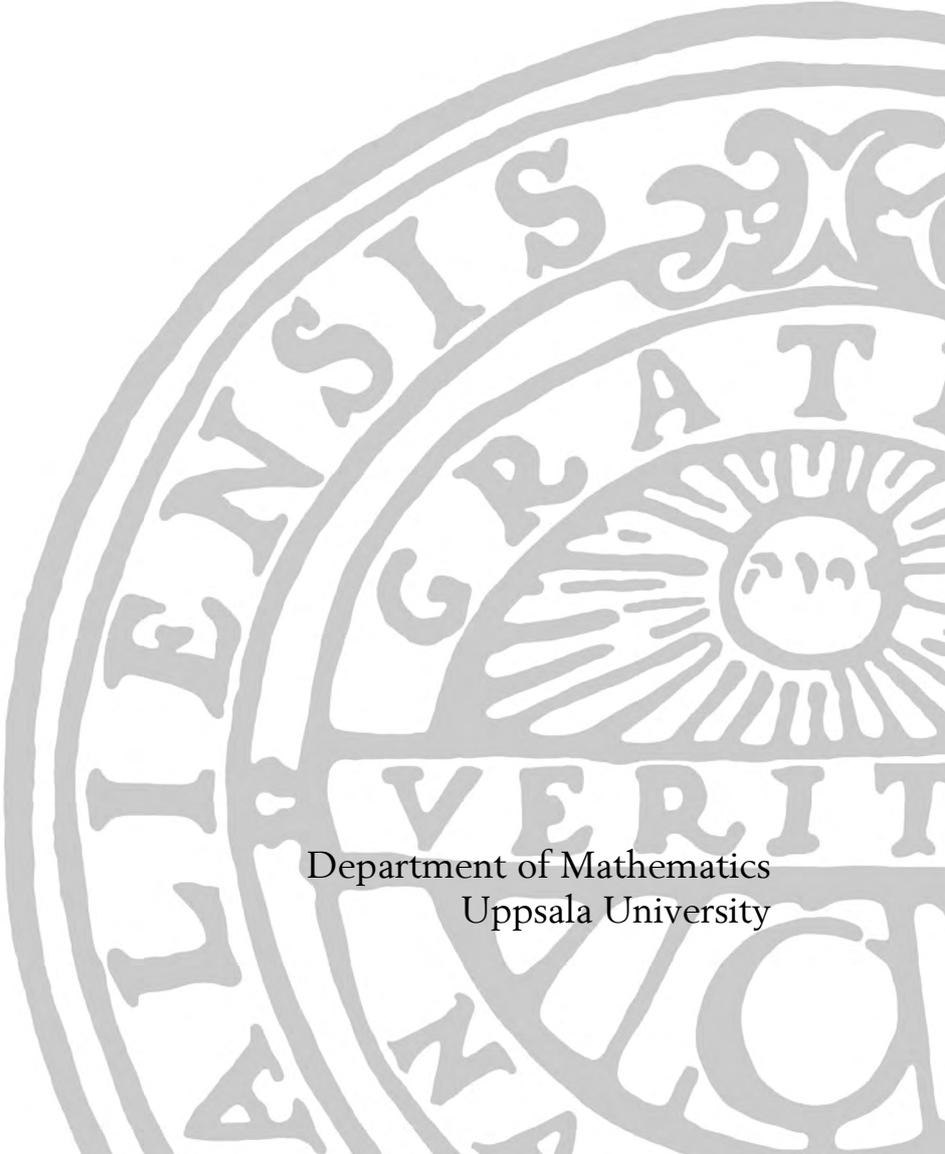
UPPSALA  
UNIVERSITET

U.U.D.M. Project Report 2018:7

# A simulational approach to parking

Erik Bertse

Examensarbete i matematik, 15 hp  
Handledare: Alejandro Luque  
Examinator: Martin Herschend  
Juni 2018

A large, faint watermark of the Uppsala University seal is visible in the bottom right corner of the page. The seal is circular and contains the Latin motto "ALIIENSIS GRATIA VERITAS" around a central sunburst design.

Department of Mathematics  
Uppsala University



# A simulational approach to parking

Degree Project C in Mathematics

Erik Bertse

Uppsala University  
Spring 2018

## **Abstract**

How can your choice of parking along a street impact your fellow parkers? In this project, we investigate the effects of different parking strategies on a street. In our simulational approach, we have implemented a model of a street, visited by parkers with different strategies. We have then investigated how our defined metrics of efficiency are impacted by different strategy choices. The results show that some strategies are more significant than others, and we can treat them as belonging to two different families of strategies. We also see that the length plays a role, in relation to a given strategy. If we want our fellow parkers to fit on our streets, this document will tell you how not to park.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Method</b>	<b>3</b>
2.1	The Model . . . . .	3
2.2	Measurements . . . . .	5
2.3	The Implementation . . . . .	6
2.3.1	Simulated time and space . . . . .	7
2.3.2	On random numbers . . . . .	8
2.3.3	Data output and mass simulation . . . . .	8
<b>3</b>	<b>Results</b>	<b>9</b>
3.1	The effects of $C_t$ on the simulation . . . . .	9
3.2	The effects of choice of spot strategy . . . . .	12
3.3	Hypothesis testing for different strategies . . . . .	13
3.4	The impact of street length on street utilization . . . . .	15
3.5	Combining drivers with various strategies . . . . .	17
3.6	Simulations with permanent cars . . . . .	19
<b>4</b>	<b>Discussion</b>	<b>21</b>
4.1	Model choices and suggested improvements . . . . .	21
4.2	Measurement Issues . . . . .	21
4.2.1	Choice of measurement . . . . .	22
4.2.2	Technical issues in measurements . . . . .	22
4.3	Floating point arithmetic considerations . . . . .	22
<b>5</b>	<b>Conclusions</b>	<b>23</b>
<b>6</b>	<b>References</b>	<b>24</b>

# 1 Introduction

In this project, we tackle the problem of parking on a busy street.

In today’s society, parking is an highly relevant problem. It is in short supply on our streets, and we are always striving to allow willing drivers to park, and with the rise of autonomous cars, it is even more important that we can find good strategies to utilize, so as to maximize efficiency when parking.

In this project, we have taken a simulational approach to investigating this problem. We have implemented a model for cars parking along a street, by constructing a simulation of this street. Cars arrive at a steady pace, and stay for a certain amount of time, in such that way that we keep the street relatively full all the time. We have not assumed that the street has predefined parking spots, but is modelled as a continuous line. The details of our implementation of the model are presented below, in section 2.3.

We define the quantities that we will be interested in measuring, so we can evaluate efficiency in a more stringent way. We then proceed to measure these quantities, in different situations. Our main point of investigation is the difference between two types of family of strategies, that can intuitively be understood as representing confident and less confident drivers (the cynical person would say ‘empathic’ and ‘nonempathic’ or ‘unsympathetic’ drivers, but we will not a priori pass judgement). In the following sections, we present and discuss how our measurements change, and we do this for the two families of strategies both in combination and isolation.

## 2 Method

In this section we first present our approach to modelling parking along a street. Here we give a conceptual overview of the conditions we are simulating. We also present the data points that we use to measure the performance of the street configuration.

After this, we present the implementation of the model, and summarize its construction.

### 2.1 The Model

We simulate a street, of length  $L$ , on which cars arrive, park, stay for a certain time interval, and then depart. The street is modelled as a continuous street without predetermined parking slots. We simulate, on a minute basis, the arrival and departure of cars in the street. In each simulated minute, a certain number of cars arrive. This is governed by the poisson parameter  $\lambda_{\text{cpm}}$ . Although the simulation allows  $\lambda_{\text{cpm}}$  to be set manually, in this text, we have restricted ourselves to the case  $\lambda_{\text{cpm}} = 1$ . Effectively, we are simulating one car per minute.

Each car,  $\mathbf{C}$ , can has the following properties:

- $\mathbf{C}_{cl}$  - The length of the car.

- $\mathbf{C}_{st}$  - The amount of iterations (simulated minutes) each car stays parked.
- $\mathbf{C}_S$  - An assignment of selection strategy.
- $\mathbf{C}_p$  - The cars position on the street (assigned only if the car parks)

Both the values of  $\mathbf{C}_{cl}$  and  $\mathbf{C}_{st}$  are normally distributed. Following classical notation, we write:

$$\mathbf{C}_{cl} \sim \mathcal{N}(\mu_{cl}, \sigma_{cl}^2) \quad \mathbf{C}_{st} \sim \mathcal{N}(\mu_{st}, \sigma_{st}^2)$$

where  $\mu$  and  $\sigma$  is the mean and the standard deviation of each car attribute, respectively.

In our model, we have settled for a fixed  $\sigma_{cl} = 0.3$  and  $\mu_{cl} = 4.5$ , so we put  $\sigma_{cl} = 0.3$  to make sure that we don't have cars that are too short or long. Employing the 66-95-99 rule, we can make sure that 95% of our cars are in the range (3.9, 5.1) which we consider reasonable.

In our model, any car can be parked anywhere in the street, subject to the restriction that a *safety distance*,  $\delta_{sd}$ , must be upheld between cars. The endpoints of the street are exceptions to this. A car can park 'tightly' against the endpoints of the street.

Given two consecutively parked cars on the street, we will refer to the space between the cars as a *spot*.

Each arriving car is assigned a *strategy*,  $S$ , that determines where that car will choose to park. Each car will make two decisions: how to select its position within a given spot, and how to select a spot. We denote the strategies corresponding to these two decisions by  $s_p$  and  $s_s$ , respectively.

A strategy,  $S = (s_p, s_s)$ , is then a choice of pair:

$$S \in \{\mathbf{L}, \mathbf{M}, \mathbf{R}\} \times \{l, s, r, f\}$$

The possible selections of  $s_p$  have the following meaning:

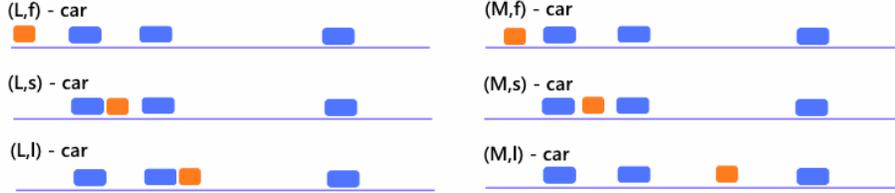
- $\mathbf{L}$  - The car parks as far left as possible in its spot
- $\mathbf{M}$  - The car parks in the middle of its spot.
- $\mathbf{R}$  - The car parks randomly in its spot.

The possible selections of  $s_s$  have the following meaning:

- $l$  - The car chooses the largest spot.
- $s$  - The car chooses the smallest spot.
- $r$  - The car chooses a random spot.
- $f$  - The car chooses the first available spot.

A car always parks according to the strategy it has been assigned. In this text, we will often refer to a car by its strategy. For example, a car that always selects the largest spot and then parks in the middle of that spot will be referred to as

an  $(\mathbf{M}, l)$ -parker, or  $(\mathbf{M}, l)$ -car. We will also sometimes refer to a collection of strategies with fixed first element, for example, we will use the notation  $(\mathbf{M}, \cdot)$  to talk about the family of strategies with  $s_p = \mathbf{M}$ . Below we present a small graphic that illustrates parking strategies for all determinations except the random strategy families  $(\mathbf{R}, \cdot)$  and  $(\cdot, r)$ .



The duration a car stays parked on the street has been chosen to follow a normal distribution, which is dictated by an indirect parameter  $C_t$  that we introduce here. We put the mean stay duration  $\mu_{st}$  as :

$$\mu_{st} = \frac{C_t L}{(\mu_{cl} + 2\delta_{sd})\lambda_{cpm}}$$

where  $C_t$  is in principle any positive value, but usually falls within the range  $0.5 < C_t \leq 1.0$ .

In essence, the parameter  $C_t$  allows us to control the extent to which the cars are ‘competing’ against each other for a place to park, in the following sense: If we put,  $C_t = \lambda_{cpm}$ , then  $\mu_{st} = L/(\mu_{cl} + 2\delta_{sd})$  and we have that each car stays, on average, as many minutes as it takes to fill the street, given the assumption that each car take up approximately  $(\mu_{cl} + 2\delta_{sd})$  meters of space (a discussion of this assumption see section 4.2). In particular, this means that, if we start the simulation with an empty street, the first car will, on average, leave approximately when the street is full. By setting this coefficient to values greater than  $\lambda_{cpm}$ , we can saturate the simulation. For values  $C_t < \lambda_{cpm}$ , we expect that, on average, the street is less than full, and this will allow most arriving cars to park. We verify the usefulness of this coefficient  $C_t$  in section 3.1 below.

For most measurements below we set

$$\sigma_{st} = 0.1\mu_{st}$$

as the standard deviation  $\sigma_{st}$  for the time each car stays, so that the time each car stays is no more than 20% from its mean, in 95% of cases.

## 2.2 Measurements

We are interested in measuring the following characteristics of the simulation:

- **PS** - This value represents the amount of parkable space in the street, in percent. This is defined as the empty space on the street, that could be used for parking, i.e. the empty space on the street excluding space that is part of the mandatory safety distance.
- $T_n$  - Time to *failure condition*, the number of simulated minutes the simulation runs until  $n$  cars cannot park.

Both of these data points represent the ‘capacity’ of the simulation. If the street is full of cars, then **PS** will show how much of the space on the street is ‘wasted’.

We measure **PS** in the following way: If  $\{C_1 \dots C_n\}$  are  $n$  cars parked on the street, then let  $D$  be the set of distances between *consecutive* cars. We put:

$$d_i = \text{distance between the front of } C_i \text{ and back of } C_{i+1}$$

As special cases, we also put:

$$\begin{aligned} d_0 &= \text{distance to the back of } C_1 \\ d_n &= \text{distance from the front of } C_n \text{ to end of street} \end{aligned}$$

Then we define parkable space **PS** as:

$$\mathbf{PS} = \sum_{i=0}^n \hat{d}_i \quad \text{where} \quad \hat{d}_i = \begin{cases} d_i - 2\delta_{sd} & \text{if } d_i > 2\delta_{sd} \text{ and } 0 < i < n. \\ d_i - \delta_{sd} & \text{if } d_i > \delta_{sd} \text{ and } (i = 0 \text{ or } n). \\ 0 & \text{otherwise} \end{cases}$$

**PS** is then the amount of free space that is not utilized, but available for parking without breaking the  $\delta_{sd}$ -restriction.

Often, we will be interested in the average and minimal value of **PS** in a simulation. We will denote these values by  $\mathbf{PS}_{\text{Avg}}$  and  $\mathbf{PS}_{\text{Min}}$ . Since we are interested in the measurement of these quantities when the street is full, a lower value of **PS** is, in some sense, better.

The measurement of  $T_n$  is much more straight-forward. We measure the number of simulated minutes that pass before  $n$  cars arrive but fail to park. For a given simulation, this is always an integer, but since we run many simulations and average the results, we are often not plotting integers in the plots in section 3.

In all simulations below, we start measuring our data point from the time step on which the first car leaves the street, and onward.

## 2.3 The Implementation

Here we discuss the details of how the model above is implemented, how we generate random numbers for the simulation, and how we perform mass simulations and extract data from these simulations.

### 2.3.1 Simulated time and space

The entire simulation is written in ANSI C, using only standard C libraries.

The street is simulated as an array of cars, where each car is represented by a `Car` struct. The simulation always ensures that the order of the cars in the array matches the order of the cars on the street, where we consider the leftmost parked car as first.

As the length of the street,  $L$ , is a parameter of the simulation, this array is allowed to become arbitrarily long.

Each time interval (which model a minute) is represented simply by a loop iteration, and so in *each* time-step the following happens:

1. A number (with expected value  $\lambda_{cpm}$ ) of cars arrive at the street (effectively we allocate temporarily, this many new `Car` structs). Each car is then assigned values for  $\mathbf{C}_{cl}$ ,  $\mathbf{C}_{st}$  and  $\mathbf{C}_S$ , where the first two mentioned values are randomly generated, and governed by the parameters  $\mu_{cl}, \sigma_{cl}$  and  $\mu_{st}, \sigma_{st}$ , respectively. The value of  $\mathbf{C}_S$  is determined by a program parameter fed to the simulation upon execution.
2. Each car, in order, check whether they can park on the street. This means that each cars checks whether there is enough space for the car to fit, while respecting the security distance  $\delta_{sd}$ . If there is available space in which to park, the car gets assigned its position according to its strategy, and put in the street array, i.e. the car parks. The cars park ‘one-by-one’, e.g. if  $\mathbf{C}_1, \mathbf{C}_2$  are two cars arriving in a given minute,  $\mathbf{C}_2$  would not even evaluate whether it can park before  $\mathbf{C}_1$  has decided on where to park, and parked there.
3. All cars that are supposed to leave at this minute do so.
4. Various calculations are made, gathered for output, and the output is written to an auxilliary file.

The simulation allows for combining two strategies  $S_1, S_2$  in a given simulation, and the percentage of cars assigned  $S_1$ , as compared to  $S_2$  is given as a parameter of the simulation. For example, if we would run the simulation with 50%  $S_1$ -parkers and 50%  $S_2$ -parkers, each car would be randomly assigned either  $S_1$  or  $S_2$ , using a uniform distribution, so that on average there would arrive equally many  $S_1$  and  $S_2$ -parkers.

The time loop will keep executing until a certain end condition is met. The simplest one of these is to simply allow the simulation to run for a certain amount of iterations, but we can also halt the simulation for any condition we choose, such as halting it when the amount of cars that arrived but did not park reaches a certain number, as we do to measure  $T_n$ .

### 2.3.2 On random numbers

The simulation requires a lot of random numbers. We generate random numbers from various distributions, in general several times per car. Since ANSI C doesn't have built-in support for generating a random number from any distribution other than a uniform distribution, we write the algorithms that generate numbers from these distributions ourselves, by first randomizing a uniformly distributed value in the interval (0,1), and then using transform algorithms.

We use the following algorithms to produce random numbers from uniformly distributed numbers:

- Box Mueller transform [1]
- Knuths Algorithm for Poisson numbers [2] .

We use the built-in C function `rand()` to generate the uniformly distributed values. At the start of each simulation, this pseudo-random number generator is reseeded with a seed that is supplied as a parameter to the program. This allows us, for example, to run the simulation many times, for various seeds, and take averages, but still allows us to replicate results deterministically.

### 2.3.3 Data output and mass simulation

For our purposes, it is useful to perform a large number of simulations, automatically. In particular, we would like to study how simulation parameters change the outcome of the simulation and values of our data points. Since the simulation uses a lot of random numbers, we would also like to rerun the simulation for each configuration multiple times, changing only the seed, to be able to take averages, to get a more accurate result.

We run the simulation in a Windows environment. We use a script to execute the simulation a fixed number of times, in a particular configuration. In each execution of the simulation, a file is produced, containing data (such as current **PS**, number of cars not parked so far, and other) for each executed time-step.

Thus, running a script that executes 200 simulations will result in 200 files, each containing detailed data for the timesteps of the corresponding simulation. Furthermore, an aggregate file is produced that contains a summary of each simulation execution, in this case, containing 200 summaries, one of each simulation.

For every simulation, we keep track of 14 different output values that are written into the aggregate file.

We are then free to investigate the data that the simulations produced. The tool of choice is *gnuplot* [3] , that allows us to write scripts to generate plots, so as to visualize the data. We also utilize gnuplots ability for simple statistical calculations (such as computing averages) when working with the data.

### 3 Results

In this section, we present our results. We start with an investigation into the effects of the parameter  $C_t$  on our data points and its relation to  $\delta_{sd}$ , the effects of the  $L$  for some configurations, and the results of allowing drivers of mixed strategies park on the street. Finally we briefly discuss the effects of initializing the simulation with cars to a certain percentage.

#### 3.1 The effects of $C_t$ on the simulation

As described above, we have a coefficient  $C_t$  that dictates the time each car stays parked in the simulated street. Here we discuss the influence of this parameter on  $\mathbf{PS}$  and  $T_n$ .

We know that a small value of  $C_t$  implies a small value of  $\mu_{st}$ . Effectively, if  $\mu_{st}$  is very low, all cars leave immediately after arriving. We expect that if  $C_t = \lambda_{cpm}$ , then, on average, each car stays as long as it takes for an empty street to be filled with cars, and so, we expect low values of  $\mathbf{PS}$ , for values  $C_t \approx \lambda_{cpm}$ . We reiterate that for all our experiments, we put  $\lambda_{cpm} = 1$ , and we therefore plot  $C_t$  in the interval  $(0, 1)$ .

In the following exposition, we have fixed  $L = 300$ ,  $\mu_{cl} = 4.5$  and  $\sigma_{cl} = 0.3$ , and we want to compare the two strategies  $(\mathbf{L}, l)$  and  $(\mathbf{M}, l)$ . We are running the simulation 10 times, each with different seeds, for a hundred values of  $C_t \in [0, 1]$ , in increments of 0.01. We do this for  $(\mathbf{L}, l)$  and  $(\mathbf{M}, l)$ -parkers, for three different values for  $\delta_{st}$ .

In figure 1 below, we are measuring  $\mathbf{PS}_{Avg}$  (see section 2.2), and we have plotted an ideal line for comparison. On this ideal line, the value of  $\mathbf{PS}_{Avg}$  exactly correspond to the values of  $100(1 - C_t)$  (Since we measure  $\mathbf{PS}_{Avg}$  in percent), with the interpretation that if  $C_t = 0.3$ , then 70% consists of parkable space, on average. We would like to know to what extent this is true.

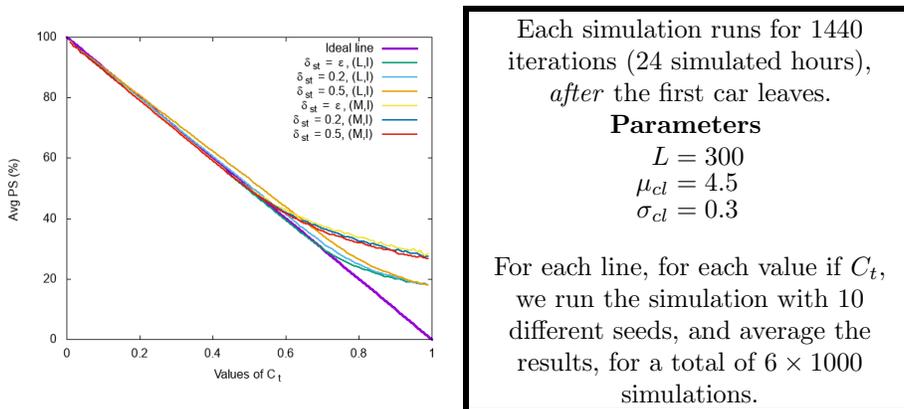


Figure 1: The average value of  $\mathbf{PS}$  for  $C_t \in [0, 1]$

We make the following observations about the figure: Both  $(\mathbf{L}, l)$  and  $(\mathbf{M}, l)$ -drivers follow this ideal, but only up to a certain value. For  $(\mathbf{L}, l)$ -cars, we see that after around  $C_t \approx 0.7$ ,  $\mathbf{PS}$  starts declining and reaches approximately 20%. For  $(\mathbf{M}, l)$ -cars, after around  $C_t \approx 0.6$ ,  $\mathbf{PS}$  starts declining and reaches approximately 30%. We can suspect that the reason for these values of  $C_t$ , is that for lower values, the street is simply never completely full during a simulation. Since we want to measure the difference between parking strategies when not fitting is a possibility, we would like to put a  $C_t$  that allows the street to be full, but not stay too long.

To get a better estimate of an appropriate value for  $C_t$ , we perform the same simulations as in figure 1, but instead measuring  $\mathbf{PS}_{\text{Min}}$ . The results are shown in figure 2 below.

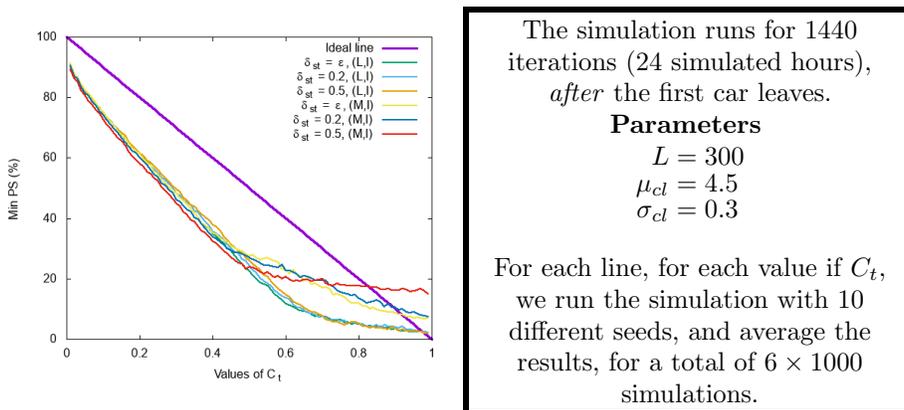


Figure 2: The minimum value of  $\mathbf{PS}$  for  $C_t \in [0, 1]$

We can clearly see a similar pattern here,  $\mathbf{PS}_{\text{Min}}$  is lower for  $(\mathbf{M}, l)$ -drivers. We can also see that for approximately  $C_t < 0.5$ , the street never gets completely full for the  $(\mathbf{M}, l)$ -parkers. For  $(\mathbf{L}, l)$ -parkers to fill the street  $C_t$  has to be approximately 0.6 or higher.

We can see that, again, the  $(\mathbf{L}, l)$  hit the same minimum parking space for the tested values of  $\delta_{sd}$ . For  $(\mathbf{M}, l)$ -parkers, all parkers stabilized, with a slight increase for minimum parkable space for  $\delta_{sd} = 0.5$ .

We can also see that neither  $\mathbf{PS}_{\text{Avg}}$  nor  $\mathbf{PS}_{\text{Min}}$  are changed significantly, regardless of  $\delta_{sd}$ , with almost identical results for  $\mathbf{PS}_{\text{Avg}}$ . Since  $\mathbf{PS}_{\text{Avg}}$  does not vary with  $\delta_{sd}$ , we will fix a value of  $\delta_{sd} = 0.2$  for forthcoming simulations.

Please note that, because of numerical instability that arise due to floating point calculations, to model ‘no safety distance’, we put  $\delta_{sd} = \epsilon := 10^{-7}$ , instead of  $\delta_{sd} = 0$  (See section 4.3 for more on this).

Next we present the same plot for the measuring  $T_{40}$ , the time it takes until 40 cars arrive but cannot park. Study figure 3.

Note that we plot only for values of  $C_t$  from 0.7 to 1. For smaller values, especially for  $(\mathbf{L}, l)$ -parkers, the condition to terminate is never satisfied. A car can always park.

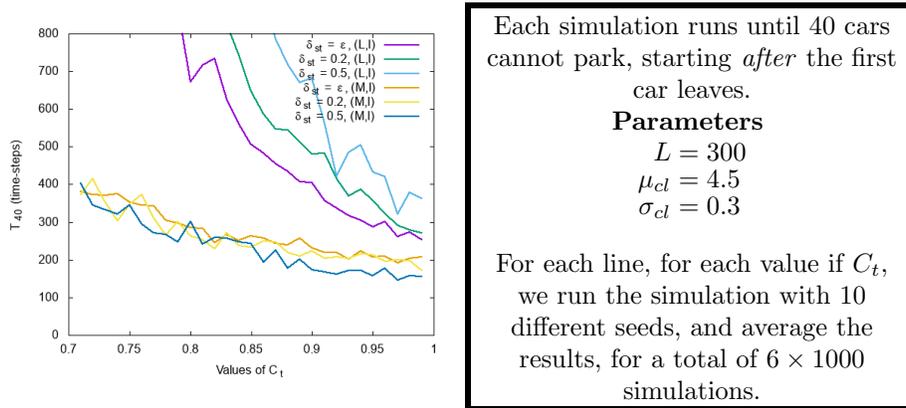


Figure 3: The value of  $T_{40}$  as a function of  $C_t$  for a street with  $L = 300$

Here we see a clear difference between  $(\mathbf{M}, l)$  and  $(\mathbf{L}, l)$ . For  $(\mathbf{M}, l)$ ,  $T_{40}$  is stable and gently declining, while for any smaller value  $C_t$  than around 0.70, for  $(\mathbf{L}, l)$ ,  $T_{40}$  is effectively infinite. Also here, we don't see a very significant change for other values of  $\delta_{st}$ . We can note that for  $(\mathbf{L}, l)$ , and a fixed  $C_t$ , larger values of  $\delta_{st}$  result in a slightly larger value of  $T_{40}$ . This is to be expected. Given that  $L$  and  $C_t$  is fixed, each car stays for a shorter duration (by definition of  $\mu_{st}$ ), and so  $T_{40}$  increases.

The data in figure 3 is averaged each over 10 simulations. Even so, we can see that  $T_{40}$  can vary a lot for the same value of  $C_t$ , especially for  $(\mathbf{L}, l)$ -parkers. For this strategy, and for a smaller  $C_t$ , a car not parking is a rather rare occurrence, and the time it takes for 40 cars to not park can vary a lot, and be quite unpredictable.

We can now make an informed choice of an appropriate value of  $C_t$  and  $\delta_{sd}$ . We put  $C_t = 1$  for most of the upcoming simulations. This allows us to measure  $T_n$  for both  $(\mathbf{L}, \cdot)$ -parkers and  $(\mathbf{M}, \cdot)$ -parkers. We also ensure that the street is filled during the simulation, so that cars can be rejected from the street at a steady rate. We put  $\delta_{sd} = 0.2$ , to conform with a realistic distance, but as the above discussion shows, changing  $\delta_{sd}$  changes the simulation only slightly.

Because of this observation, we will not be considering or simulating how a car can physically maneuver into its spot, for a given  $\delta_{sd}$ , since this does not change the result significantly. We therefore feel free to restrict our investigation to using a larger  $\delta_{sd}$ , under which a car realistically maneuver into its spot.

### 3.2 The effects of choice of spot strategy

In this section, we are investigating the effects of the second parameter within each of the family of strategies, that determines the selection of spot. To this end, we will be comparing our measurements for the strategies  $(\mathbf{L}, s_s)$ , and  $(\mathbf{M}, s_s)$  for all possible values of  $s_s$ .

We start with a simple measurement. For a fixed length, what strategy allows for the lowest average value of **PS**? Here we fix the parameters  $C_t = 1$ , and selected an  $L = 300$ . The results are summarized in figure 4.

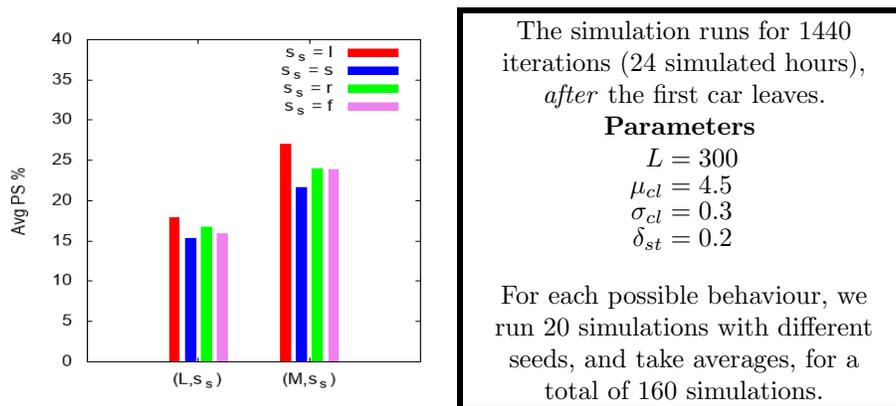


Figure 4: Average **PS** for each possible choice of strategy

We see that given a strategy  $S = (s_p, s_s)$ , the value of  $s_p$  will determine the average **PS** to a greater extent. Furthermore, for both  $S = (\mathbf{L}, \cdot)$  and  $S = (\mathbf{M}, \cdot)$ , we can see that that  $s_s = l$  gives the largest value of **PS**, while  $s_s = s$  gives the smallest value. It is interesting to note that choosing the first available spot resulted in a higher **PS** than picking the smallest spot. We can conclude that for a full street, we can utilize the street better if we park in the smallest spot, not the first available.

We now repeat the same experiment, measuring  $T_{40}$  instead. We keep  $C_t = 1$ , so that  $T_{40}$  does not vary quite as much, as this can happen, especially for lower values of  $C_t$ , as described in the previous subsection. To get a good value, we rerun the simulation 50 times, and take the average value of  $T_{40}$ . The results are summarized in figure 5.

The simulation reaches the failure condition fastest with  $s_s = l$ , for both  $(\mathbf{L}, \cdot)$  and  $(\mathbf{M}, \cdot)$ -parkers. Note that for  $(\mathbf{L}, \cdot)$  taking the first available spot resulted larger  $T_{40}$  than the smallest spot. For  $(\mathbf{M}, \cdot)$ -parkers, selecting the smallest spot is the better strategy.

We can now conclude that the selection of  $s_p$  determines the outcome of the simulation more than the determination of  $s_s$ . Assigning  $s_s = l$  results in the largest amount of **PS**, and the smallest value of  $T_{40}$ , for both selections of  $s_p$ .

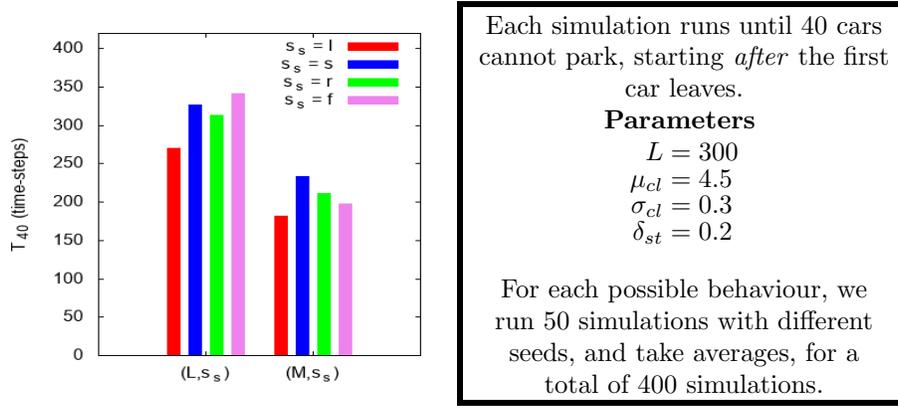


Figure 5:  $T_{40}$  for each possible choice of strategy

### 3.3 Hypothesis testing for different strategies

In this subsection, we dissect the data in figure 5 in detail. Since the variance for the values of  $T_{40}$  is large, we would like to test whether the data in the figure is statistically significant. Since we are averaging 50 points in figure 5, and take averages, we would like to know, given fixed  $s_s$ , if the mean value of  $T_{40}$  for  $(\mathbf{L}, s_s)$  is equal the mean value of  $T_{40}$  for  $(\mathbf{M}, s_s)$ .

We will be performing the two sample t-test for equal means. [4] The reason we chose this test is that it allows for unknown and different variances, which is the case here. Note that since we have sampled 50 points for each strategy, we can assume that the mean of  $T_{40}$  follows a normal distribution, because of the central limit theorem, so we are justified in using this test.

Let  $T_{40}^S$  denote the mean value of  $T_{40}$  under the strategy  $S$ . We formulate four hypotheses, one for each possible value of  $s_s$ , like so:

$$H_0^{(\mathbf{L}=\mathbf{M}, s_s)} : T_{40}^{(\mathbf{L}, s_s)} = T_{40}^{(\mathbf{M}, s_s)}$$

Because of how we formulate the test hypothesis, we will be performing a two-tailed test. For our choice of t-test, the test statistic is given by:

$$T = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{s_1^2/N_1 + s_2^2/N_2}}$$

where  $\bar{x}$  is the measured mean,  $s^2$  is the sample variance, and  $N$  is the sample size. We mean to reject the null hypothesis, at a confidence level  $\alpha$ , if

$$|T| > t_{1-\alpha/2, df}$$

where  $t_{1-\alpha/2, df}$  is the critical value of the  $t$ -distribution with  $df$  degrees of freedom. Here,  $df$  is given by the formula:

$$\frac{1}{df} = \frac{1}{N_1 - 1} \frac{(N_2 s_1^2)^2}{(N_2 s_1^2 + N_1 s_2^2)^2} + \frac{1}{N_2 - 1} \frac{(N_1 s_2^2)^2}{(N_2 s_1^2 + N_1 s_2^2)^2}$$

While this formula looks rather contrived, in our case, we have  $N_1 = N_2 = 50$ , simplifying the expression considerably. Note that since  $t_{1-\alpha/2,df}$  depends on  $df$ , which in turn depends on the sample variances, the critical values for which we reject the null hypotheses varies between the tests. We are interested in a significance level  $\alpha = 0.05$ , and we use tabulated values to look up values of  $t_{1-\alpha/2,df}$ .

Since we are testing for equal means, the confidence intervals below give the interval for the value of the difference of the means of  $T_{40}$ , for the strategies under consideration. For example, the confidence interval given for  $H_0^l$  signifies that we are 95% confident that the difference between the means of  $T_{40}$  for  $(\mathbf{L}, l)$  and  $(\mathbf{M}, l)$ -parkers, lies within the interval specified. The intervals are computed according to the formula:

$$(\bar{x}_1 - \bar{x}_2) \pm (t_{1-\alpha/2,df}) \sqrt{s_1^2/N_1 + s_2^2/N_2}$$

Now we have all we need. We find the following values of  $T$  for each of our hypotheses:

Hypothesis	$x_1$	$x_2$	$s_1^2$	$s_2^2$	Reject $H_0$ ?	Conf. interval:
$H_0^{(\mathbf{L}=\mathbf{M},l)}$	269.76	181.22	9404.7	1246.2	Yes	$88.52 \pm 29.1$
$H_0^{(\mathbf{L}=\mathbf{M},s)}$	326.7	233.64	10255	4058.0	Yes	$93.03 \pm 33.7$
$H_0^{(\mathbf{L}=\mathbf{M},r)}$	313.34	211.24	8060.4	2931.5	Yes	$102.1 \pm 29.5$
$H_0^{(\mathbf{L}=\mathbf{M},f)}$	341.76	198.32	19557	2420.5	Yes	$143.44 \pm 41.9$

As we can see, we can safely conclude that the  $(\mathbf{L}, \cdot)$ -family of parking strategies lead to a larger  $T_{40}$ .

To compare strategies within each family, we formulate six more null hypotheses. For each family  $(\mathbf{L}, \cdot)$  and  $(\mathbf{M}, \cdot)$ , we have three null hypotheses, in the following manner:

$$\begin{aligned} H_0^{(\mathbf{L}/\mathbf{M},s=l)} : T_{40}^{(\mathbf{L}/\mathbf{M},s)} &= T_{40}^{(\mathbf{L}/\mathbf{M},l)} \\ H_0^{(\mathbf{L}/\mathbf{M},s=f)} : T_{40}^{(\mathbf{L}/\mathbf{M},s)} &= T_{40}^{(\mathbf{L}/\mathbf{M},f)} \\ H_0^{(\mathbf{L}/\mathbf{M},l=f)} : T_{40}^{(\mathbf{L}/\mathbf{M},l)} &= T_{40}^{(\mathbf{L}/\mathbf{M},f)} \end{aligned}$$

We are thus comparing every selection of  $s_s$  to every other selection of  $s_s$ , for every family of parkers. This results in the following table:

Hypothesis	$x_1$	$x_2$	$s_1^2$	$s_2^2$	Reject $H_0$ ?	Conf. interval:
$H_0^{(\mathbf{L},s=l)}$	326.7	269.74	10255	9404.7	Yes	$56.96 \pm 39.4$
$H_0^{(\mathbf{L},s=f)}$	326.7	341.76	10255	19557	No	$15.06 \pm 48.5$
$H_0^{(\mathbf{L},l=f)}$	269.74	341.76	9404.7	19557	Yes	$72.02 \pm 47.8$
$H_0^{(\mathbf{M},s=l)}$	233.64	181.22	4058.0	1246.2	Yes	$52.42 \pm 20.5$
$H_0^{(\mathbf{M},s=f)}$	233.64	198.32	4058.0	2420.5	Yes	$35.32 \pm 22.6$
$H_0^{(\mathbf{M},l=f)}$	181.22	198.32	1246.2	2420.5	Yes*	$17.1 \pm 17.0$

The star in the last row signifies that we can reject  $H_0^{(\mathbf{M},l=f)}$  at  $\alpha = 0.05$  but not at lower values of  $\alpha$ .

From the table above, we can see that for both  $(\mathbf{L}, \cdot)$  and  $(\mathbf{M}, \cdot)$ -families, choosing to park in the smallest spot results in larger  $T_{40}$ , than choosing the largest spot. We cannot be confident that  $(\mathbf{L}, f)$  and  $(\mathbf{L}, s)$  result in different means for  $T_{40}$ . This is because for these choices of strategy, the variance is very large, and the difference between means is very small, leading to a large confidence interval. In a pleasant symmetry, we can only barely reject  $H_0^{(\mathbf{M},l=f)}$ . We can conclude that for  $(\mathbf{L}, \cdot)$ -parkers, parking in first available spot is similar to parking in the smallest spot, while for  $(\mathbf{M}, \cdot)$ -parkers, parking in the first spot is similar to parking in the largest spot.

### 3.4 The impact of street length on street utilization

In the previous section, we saw that  $s_s$  has less importance than  $s_p$ . With these initial observations, we fix  $s_s = l$ , and consider how these measurements vary with  $L$ . In the following exposition, we compare  $(\mathbf{L}, l)$  and  $(\mathbf{M}, l)$ -parkers, and we fix  $C_t = 1$ . We are interested in how  $\mathbf{PS}_{\text{Avg}}$ ,  $\mathbf{PS}_{\text{Min}}$ , and  $T_{40}$  vary.

Figure 6 shows minimum and average value of  $\mathbf{PS}$  for  $(\mathbf{L}, l)$  and  $(\mathbf{M}, l)$ -parkers, where we increment  $L$  by 5 meters.

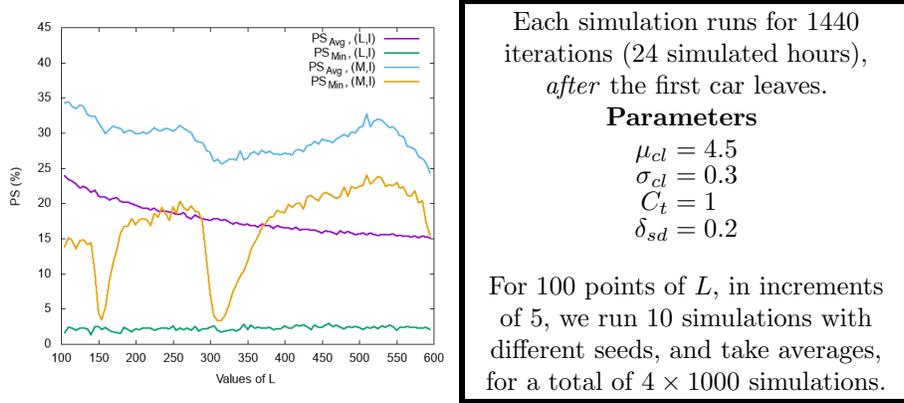


Figure 6: Average and minimum  $\mathbf{PS}$ , for  $(\mathbf{M}, l)$  and  $(\mathbf{L}, l)$ -parkers

For  $(\mathbf{L}, l)$ -parkers, both the minimal and average value of  $\mathbf{PS}$  remains stable as  $L$  increases. We can clearly see that  $\mathbf{PS}_{\text{Min}}$  is very low for all  $L$ 's, so the street is completely filled for all  $L$ 's. For  $(\mathbf{M}, l)$ -parkers, we see that for some values for  $L$ ,  $\mathbf{PS}_{\text{Min}}$  hits very small values, and it seems, for the same values of  $L$ ,  $\mathbf{PS}_{\text{Avg}}$  attains a local minimum also.

We can also note, from this figure, that for longer  $L$ 's, minimal  $\mathbf{PS}$  for  $(\mathbf{M}, l)$ -parkers is larger than average  $\mathbf{PS}$  for  $(\mathbf{L}, l)$ -parkers. In other words,  $(\mathbf{L}, l)$ -

parkers utilize more of the street in their average case, than  $(\mathbf{M}, l)$ -parkers in their best case, for these longer  $L$ 's. We can conclude that under the assumption that all cars park according to the strategy  $(\mathbf{M}, l)$ , the values of  $L$  which give minimum  $\mathbf{PS}$  will allow least space to be wasted.

We perform the same simulation, over increasing  $L$ 's, but now we measure the  $T_{40}$ . The results are summarized in figure 7.

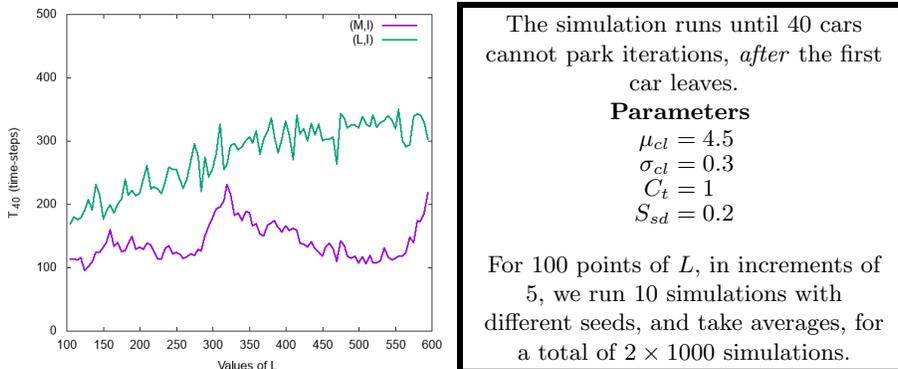


Figure 7:  $T_{40}$ , for  $(\mathbf{M}, l)$  and  $(\mathbf{L}, l)$ -parkers

We can make two observations from this graph.  $(\mathbf{L}, l)$ -parkers outperform  $(\mathbf{M}, l)$ -parkers, in the sense that in a simulation where all cars has the  $(\mathbf{L}, l)$ -strategy will run longer before  $T_{40}$  is reached. We also see a mirroring of the measurement for  $\mathbf{PS}$ . In this figure, higher is better, and the  $L$ 's for which this happens clearly correspond to the  $L$ -values of the preceding figure.

Here we present a simple inductive argument that explains the minimum values of for  $(\mathbf{M}, l)$  - parkers:

Let us assume, for the following line of reasoning the following simplified situation:

- All cars that park stay indefinitely.
- All cars have equal fixed length and  $C_{cl} = \mu_{cl}$ .

Each car  $\mathbf{C}$  that is assigned  $C_S = (\mathbf{M}, l)$  tries to park in the middle of its selected spot, and, in doing so, minimizes the space left for other cars to park in. Effectively, by parking in the middle, it halves the street into two ‘substreets’. Then, under these assumptions, each car will take up

$$X = C_{cl} + 2\delta_{sd}$$

units of parkable space in the street. This is true for all cars, except possibly the last cars that park, where the space they take up on the street might be as low as  $X$ , if the last spot is between two cars and measures  $X + 2\delta_{sd}$  exactly. If we now let  $L_0$  be any length such that:

$$L_0 = (2^n - 1)X$$

then after the first car parks, what space remains in the street is halved, so each substreet is of length

$$L_1 = \frac{(2^n - 1)X - X}{2} = (2^{n-1} - 1)X$$

so that, in general, we have:

$$L_k = (2^{n-k} - 1)X$$

After allowing  $2^n - 1$  cars to park, each substreet has length

$$L_n = (2^{n-n} - 1)X = 0$$

So by picking such an  $L_0$ , we can guarantee that, in idealized conditions, eventually, the remaining **PS** is 0. We can compute such  $L$ 's for a few values of  $n$  to check whether these are the same  $L$ 's for which the  $(\mathbf{M}, l)$ -parkers minimal **PS** falls close to 0. We find that we get the following  $L$ 's for the following  $n$ 's.

$$\begin{array}{ll} n = 5 & L = 151.9 \\ n = 6 & L = 308.7 \\ n = 7 & L = 622.3 \end{array}$$

Compare this to what we see in figure 6 and figure 7. We can see that for the same values of  $L$ , for  $(\mathbf{M}, l)$ -parkers,  $\mathbf{PS}_{\text{Min}}$  hits its dramatic minimum, just as this calculation shows. In figure 7, these calculated values allow for local maximums in  $T_{40}$ .

### 3.5 Combining drivers with various strategies

In this subsection, we will be investigating how **PS** and  $T_{40}$  change when we allow assignment of different strategies to different cars in each simulation. In the following simulations we are comparing between strategies  $(\mathbf{L}, l)$  and  $(\mathbf{M}, l)$ . This time we will be exploring  $T_{40}$  first.

Consider figure 8. We are plotting  $T_{40}$  as a function of the percentage  $p$  of  $(\mathbf{L}, l)$ -cars arriving at the street. The remaining  $100 - p$  percent of all the cars are assigned  $(\mathbf{M}, l)$ . We do this for four values of  $C_t$ , from 0.7 to 1.

We know from section 3.1 that a smaller value of  $C_t$  will allow a simulation with only  $(\mathbf{L}, l)$ -parkers to almost always let a car park once it has arrived. We can see this reflected in the large values of  $T_{40}$  for a high percentage of  $(\mathbf{L}, l)$ -parkers. Note that for  $(\mathbf{L}, l)$ , the endpoints for  $C_t = 1$ , where  $p = 0$  and  $p = 100$ , correspond to the point  $L = 300$  in figure 7.

We also note that for smaller values of  $C_t$ , we find that a small percentage of  $(\mathbf{M}, l)$  will have a dramatic impact on  $T_{40}$ . In essence, allowing a few cars to park according to the  $(\mathbf{M}, l)$ -strategy will drastically decrease the time until 40 cars cannot park, with this effect being stronger for smaller  $C_t$ . For example, for  $C_t = 0.7$ , the difference between the average value of  $T_{40}$  at 40%  $(\mathbf{L}, l)$ -parkers

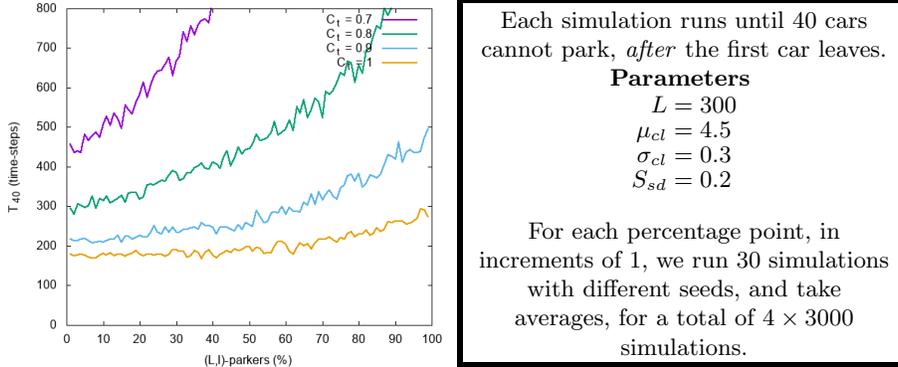


Figure 8:  $T_{40}$ , as a function of the percentage of  $(\mathbf{L}, l)$ -parkers

and 50%  $(\mathbf{L}, l)$ -parkers is about 170 simulated minutes, which is larger than the corresponding value of  $T_{40}$  at the points 0%  $(\mathbf{L}, l)$ -parkers and 100%  $(\mathbf{L}, l)$ , for  $C_t = 1$ .

The same comparison does not hold for the measurement of **PS**. In figure 9, we show average **PS**, as a function of the percentage of  $(\mathbf{L}, l)$ -parkers, for the same values of  $C_t$ . We can see that **PS** does not drop significantly as the percentage of  $(\mathbf{L}, l)$ -parkers increases. As we saw in section 3.1, **PS** decreases as  $C_t$  decreases, it did so for  $(\mathbf{L}, l)$ -parkers and  $(\mathbf{M}, l)$ -parkers individually, and it does so in combination also.

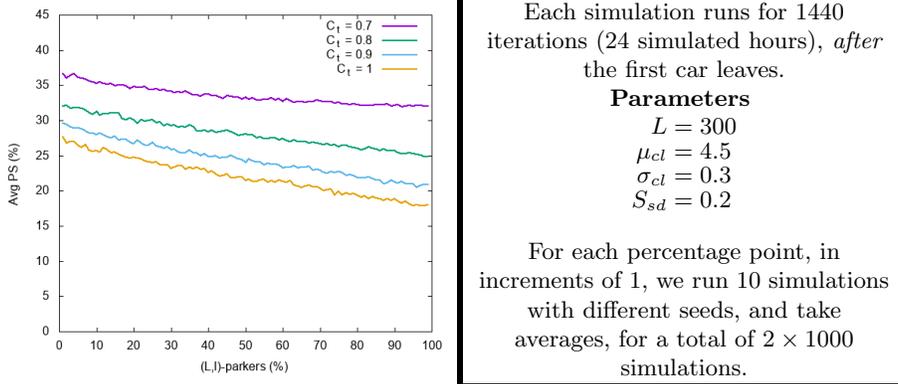


Figure 9: Average **PS**, as a function of the percentage of  $(\mathbf{L}, l)$ -parkers

Finally, we turn our attention to measure how  $\mathbf{PS}_{\text{Avg}}$  varies, as a function of  $L$  if we intermix some  $(\mathbf{M}, l)$ -drivers with  $(\mathbf{L}, l)$ -drivers, since  $\mathbf{PS}_{\text{Avg}}$  behaved differently, for different  $L$ 's, for these two strategies.

Initial observation suggested that the oscillatory behaviour of  $\mathbf{PS}_{\text{Avg}}$  for

$(\mathbf{M}, l)$  would disappear when  $(\mathbf{L}, l)$ -parkers are added. We would like to know how large the percentage of  $(\mathbf{L}, l)$ -parkers has to be before this oscillatory behaviour is imperceptible. In figure 10, we plot  $\mathbf{PS}_{\text{Avg}}$  for five different percentages of  $(\mathbf{L}, l)$ -parkers. Indeed, adding only a small percentage  $(\mathbf{L}, l)$ -parkers ‘flattens’  $\mathbf{PS}_{\text{Avg}}$ . We can see that at 25%  $(\mathbf{L}, l)$ -parkers, all of the oscillatory behaviour of  $\mathbf{PS}_{\text{Avg}}$  is gone.

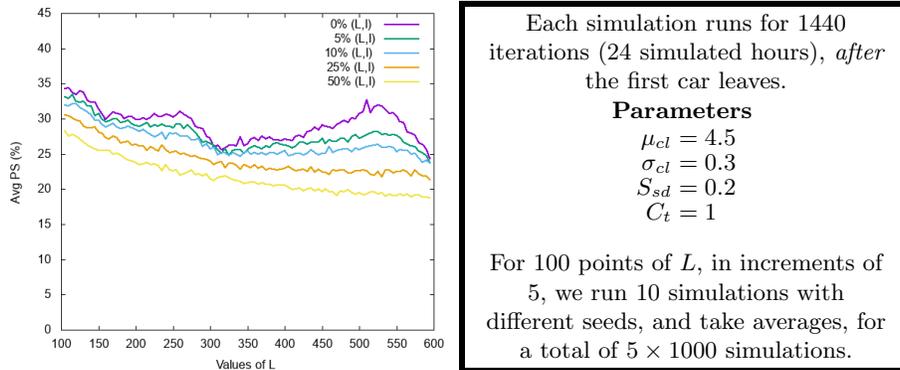


Figure 10: Average  $\mathbf{PS}$ , as a function of  $L$ , for some percentages of  $(\mathbf{L}, l)$ -parkers

### 3.6 Simulations with permanent cars

In this section, we perform our tests again, but we allow the street to be initialized to a certain percent. We are interested in whether the effects discussed in section 3.4 remain, so we allow the simulation to start with some percentage occupied. Here we are comparing cars  $(\mathbf{L}, l)$  and  $(\mathbf{M}, l)$ -parkers.  $C_t$  is fixed to 1. We are interested in whether we still get the same minimums of  $\mathbf{PS}$  for the same  $L$ -values. In these simulations the following assumptions are made:

- The simulation undergoes an initialization phase. During this phase, cars that arrive on the street are all assigned the strategy  $(\mathbf{R}, l)$ , taking the largest spot and parking in the middle of it.
- All the cars that arrived during the initialization phase are permanent, having effectively infinite value of  $\mathbf{C}_{st}$ .
- After the minute on which the percentage of  $\mathbf{PS}$  exceeds a certain value, we start collecting data points. Here  $\mu_{st}$  gets redefined as

$$\mu_{st} = \frac{C_t \mathbf{PS}_0}{(\mu_{cl} + 2\delta_{sd})\lambda_{\text{cpm}}}$$

where  $\mathbf{PS}_0$  is the value of  $\mathbf{PS}$  after initialization. We also plot  $\mathbf{PS}$  as a percentage of  $\mathbf{PS}_0$ . Effectively  $\mathbf{PS}_0$  acts as the new value for  $L$ .

Study figure 11. We can see that if we initialize the street to even a low percentage, we do not retain the minimum we saw in section 3.4. Arguably the effect is slightly there for 5% initialization, and gone at 10%. We also see that for  $(\mathbf{L}, l)$ ,  $\mathbf{PS}$  increases slightly with a slight larger initialization percentage.

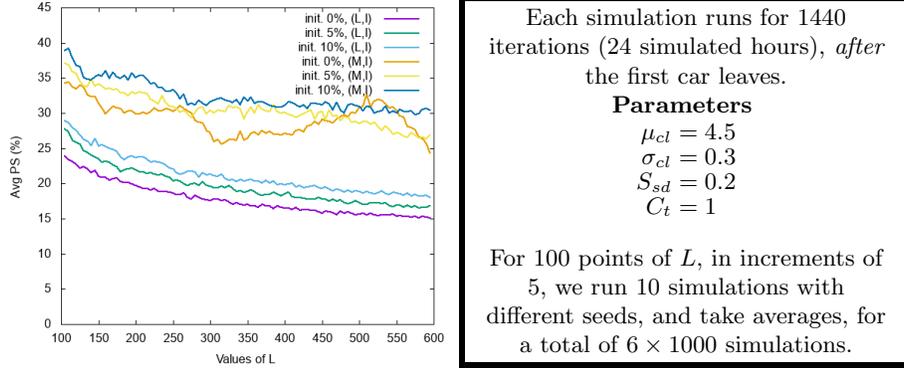


Figure 11: Average  $\mathbf{PS}$ , as a function of  $L$ , for  $(\mathbf{L}, l)$  and  $(\mathbf{M}, l)$ -parkers, with some levels of initialization

We see the same reflected in the measurements of  $T_{40}$ . At 10%,  $T_{40}$  stays flat for all  $L$ 's, and we do not retain the longer times to failure. This is shown in figure 12. In this figure we see that on  $T_{40}$  for  $(\mathbf{L}, l)$ -parkers is slightly higher without initialization. We can also see the large variance in the values for  $T_{40}$  for these parkers, the number varies quite a lot, even though we take averages over 10 simulations.

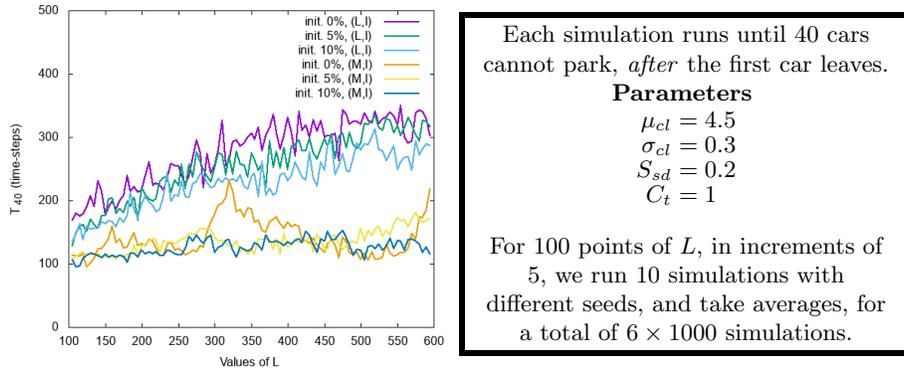


Figure 12:  $T_{40}$ , as a function of  $L$ , for  $(\mathbf{L}, l)$  and  $(\mathbf{M}, l)$ -parkers, with some levels of initialization

## 4 Discussion

We have investigated the effects of parking behaviour, and how it affects parking capacity, in our simulation. As with any model of reality, it is a simplification of reality. In this section, we discuss the model choices and measurement issues that comes with the model, how these choices can affect the results.

### 4.1 Model choices and suggested improvements

In a real situation, neither  $C_{st}$  nor the traffic flow to and from the street, dictated by  $\lambda_{\text{cpm}}$  is constant. We have made these concessions to simplify the problem. In a more complicated simulation, we could assign  $\lambda_{\text{cpm}}$  as a function of the timestep, and see how  $\mathbf{PS}$  and  $T_n$  varies as  $\lambda_{\text{cpm}}$  varies.

We want to measure  $\mathbf{PS}$  and  $T_n$  for streets that are busy. To this end, we have to have a way of automatically ensuring that cars stay long enough for the street to be full. This is the why the coefficient  $C_t$  is useful. We show in section 3 that  $C_t$ , indeed, allows us to control how full the street is during the simulation and how long it takes to reach  $T_{40}$ , and that  $\delta_{sd}$  does not affect these values significantly.

Regarding modelling time and space, a more accurate simulation could be extended in to two dimensions, with each car taking a certain area, on an empty of initialized plane.

Also, our choice of timing does not allow for control of time resolution, as each time-step is discrete. One could imagine modelling time on a more granular level, and allowing each event to take longer than one time-step. This can sometimes have effects on measurement. As described in section 2, data collection and output happens at the end of each time-step, after the cars that are supposed to leave in that time-step do so. This means that given a time-step  $i$ , we cannot capture the state of the simulation between the arrivals and departures in step  $i$ . This can introduce an error in the measurement of  $\mathbf{PS}_{\text{Min}}$ . If, in step  $i$ , there are  $n$  cars parked initially, and exactly one car arrives and one car leaves, then, at some instant, there are  $n + 1$  cars parked on the street, and so  $\mathbf{PS}_{\text{Min}}$  should decrease after iteration  $i$ , but because of this issue, we will not record this change. The ultimate reason for this is that each time step is discrete and we have to accept that, in this model, at each timestep, cars arrive, park, and leave, instantaneously.

While we do not have a specific implementation for a strategy such that cars park to the right in their spot, this is completely symmetric situation, and we consider it equivalent to the strategy  $(\mathbf{L}, l)$ .

### 4.2 Measurement Issues

Measuring information about the state of the simulation comes with its own design choices. Here we briefly discuss both our choice of measurement, and technical details that might introduce small errors into the simulation.

### 4.2.1 Choice of measurement

In this project we are measuring the quantity **PS** to estimate the occupancy of the street. The quantity **PS** evolved from a closely related quantity, the *free space* available on the street. The important difference is that free space did not take the parameter  $\delta_{sd}$  into account. Since we want to measure how much street is available for parking, this data point is less useful to us. For example, if a rather large  $\delta_{sd}$  is enforced, then the value of **PS** will accurately represent the amount of space available for parking, and not the total amount of free space on the street. We can note that early measurement showed that while free space followed some of the same patterns as **PS**, it not independent of the value of  $\delta_{sd}$  (as one would expect) which motivated us to change the data point we were interested in.

### 4.2.2 Technical issues in measurements

To define **PS** and  $\mu_{st}$ , as we do in section 2, we are making the following assumption:

Each car, when parked, occupies, on average  $\mu_{cl} + 2\delta_{sd}$  units of space

This assumption is not completely accurate, as we mentioned briefly in section 3.4. In fact, if all cars have the strategy  $(\mathbf{L}, l)$ , and the street is fully occupied, then, on average, each car only occupies  $\mu_{cl} + \delta_{sd}$  units of space. In general, a single car arriving and parking on the street can decrease **PS** (in units of space, not percent) by anywhere from  $\mu_{cl}$  to  $\mu_{cl} + 2\delta_{sd}$ . Yet, as we saw in section 3.1., this mismatch does not lead to deviation from the idealized line in the figures in that section.

When we initialize the simulation, we use parkers assigned the strategy  $(\mathbf{R}, l)$ . It is possible for two cars to arrive in the same time-step, and since we only check for the state of the street at the end of each time-step, it is possible to ‘overshoot’ the percentage mark in the initialization phase. This problem is larger for smaller values of  $L$ , since the proportion of  $\mu_{cl}$  to  $L$  is much smaller. For  $L = 300$ , and  $\mu_{cl} = 4.5$ , each car takes up approximately 2% of the whole street, which is also the reason for our selection of  $L$  in those simulations where  $L$  is fixed.

## 4.3 Floating point arithmetic considerations

Floating point arithmetic is notoriously prone to rounding error. In our simulation, we perform massive amounts of floating point arithmetic and comparisons. This has introduced numerical instability in some calculations for two calculations:

- In the generation of normally distributed numbers, we are taking a negative log. For very small floating point numbers, this function returns

‘infinity’ as its result. We therefore add  $\epsilon := 10^{-5}$  to the argument of  $\log$ , to ensure that the function returns a valid value.

- Putting  $\delta_{sd} = 0$  also leads to numerical instability, because some equality comparisons for floating point numbers fail when they should not, as the distance between the cars goes to 0. Here we put  $\epsilon := 10^{-7}$  for an arbitrarily small  $\delta_{sd}$ .

While we think these changes do not impact the data-points in any significant way, we have not performed measurements for guarantee this.

## 5 Conclusions

We have experimented with different parking strategies, to see how they affect the state of our constructed simulation. We have seen that on a busy street, if we are interested in a low vacancy on a fully occupied street, it is advisable to follow the family of strategies  $(\mathbf{L}, \cdot)$ , rather than  $(\mathbf{M}, \cdot)$ .

We have also compared how the time until 40 cars cannot park varies, depending on different spot selection strategies, where we have seen that, for both the families  $(\mathbf{L}, \cdot)$  and  $(\mathbf{M}, \cdot)$ , choosing the largest spot leads smaller values of  $T_{40}$ , and we have established that this is statistically significant. We have seen that choosing the first spot give  $(\mathbf{L}, \cdot)$ -cars results comparable to choosing the smallest spot, while  $(\mathbf{M}, f)$ -parkers give results closer to  $(\mathbf{M}, l)$ -parkers.

We have also established that if the competition for space to park is low, then parking according to  $(\mathbf{L}, \cdot)$  greatly increases time to failure  $T_{40}$ , while having little difference for  $(\mathbf{M}, \cdot)$ -parkers. Introducing  $(\mathbf{M}, l)$ -parkers to mainly  $(\mathbf{L}, \cdot)$ -parkers reduces  $T_{40}$  substantially.

We have also seen that, while for some  $L$ 's, the  $(\mathbf{M}, \cdot)$ -strategy results in lower  $\mathbf{PS}$  than for other  $L$ 's, the introduction of either  $(\mathbf{L}, \cdot)$ , even at a low percentage, will negate these effects, as does initializing the simulation to a small percentage.

## 6 References

- [1] Dustin Stansbury. *Sampling From the Normal Distribution Using the Box-Muller Transform*. 2012. Available at: <https://theclevermachine.wordpress.com/2012/09/11/sampling-from-the-normal-distribution-using-the-box-muller-transform/>.
- [2] John D. Cook. *Generating Poisson random values*. 2010. Available at: <https://www.johndcook.com/blog/2010/06/14/generating-poisson-random-values/>.
- [3] Official *gnuplot* documentation. Available at: <http://www.gnuplot.info/documentation.html>.
- [4] Alm, S. E. and Britton, T. *Stokastik: Sannolikhets-teori och statistik-teori med tillämpningar*. Liber, 2011.