Scheduling Electric Buses with Stochastic Driving Times

Philip de Bruin
Department of Information and Computing Sciences, Utrecht University, The Netherlands

Marjan van den Akker
Department of Information and Computing Sciences, Utrecht University, The Netherlands

Han Hoogeveen
Department of Information and Computing Sciences, Utrecht University, The Netherlands

Marcel van Kooten Niekerk
Department of Information and Computing Sciences, Utrecht University, The Netherlands
Qbuzz BV, The Netherlands

Abstract

To try to make the world more sustainable and reduce air pollution, diesel buses are being replaced with electric buses. This leads to challenges in scheduling, as electric buses need recharging during the day. Moreover, buses encounter varying traffic conditions and passenger demands, leading to delays. Scheduling electric buses with these stochastic driving times is also called the Stochastic Vehicle Scheduling Problem. The classical approach to make a schedule more robust against these delays, is to add slack to the driving time. However, this approach doesn't capture the variance of a distribution well, and it doesn't account for dependencies between trips. We use discrete event simulation in order to evaluate the robustness of a schedule. Then, to create a schedule, we use a hybrid approach, where we combine integer linear programming and simulated annealing with the use of these simulations. We show that with the use of our hybrid algorithm, the punctuality of the buses increase, and they also have a more timely arrival. However, we also see a slight increase in operating cost, as we need slightly more buses compared to when we use deterministic driving times.

2012 ACM Subject Classification Computing methodologies → Planning and scheduling; Computing methodologies → Modeling and simulation; Computing methodologies → Planning under uncertainty; Computing methodologies → Discrete-event simulation

Keywords and phrases Electric Vehicle Scheduling Problem, Simulated Annealing, Hybrid Algorithm, Simulation, Stochastic Driving Times

Digital Object Identifier 10.4230/OASIcs.ATMOS.2023.14

1 Introduction

In an effort to make the world more sustainable, electrification in the public transport sector is becoming more and more important. More specifically, everywhere in The Netherlands, diesel buses are replaced with their electric counterparts. This, however, introduces more constraints when scheduling these buses. In their current status, electric buses are constrained in their range, and need to be recharged during the day. For diesel buses, this is not an issue, as they could generally drive the whole day on a single tank.

The introduction of range restrictions on the buses leads to various additional problems. Electric buses either need to recharge, or swap their batteries during the day. This makes the scheduling of electric buses more difficult, as we not only need to determine the routes for the buses, but also when to recharge the vehicle. This results in an NP-hard problem [12].

There are several approaches to solve this problem, as is also discussed in a recent review by Perumal et al. [11]. However, most of these solutions assume deterministic driving times. This is not a realistic assumption, as traffic conditions and passenger loads vary from day to

23rd Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2023).
Editors: Daniele Frigioni and Philine Schiewe; Article No. 14; pp. 14:1–14:19
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

day, which causes delays. These delays can cause various issues, such as delay propagation in our network or inconveniences for the end-user, as it results in longer waiting times, delayed arrivals, and possibly missed transfers. We investigate using stochastic driving times in order to make our schedules more robust against these delays. Furthermore, we also consider the driving behaviour of the bus drivers, since someone with a more sporty driving style has a different energy consumption than someone who drives more conservatively. The scheduling of these buses with the use of stochastic variables is called the Stochastic Electric Vehicle Scheduling Problem (stochastic E-VSP).

A classical way to deal with stochastic driving times, is to include some slack based on their distribution. This slack time could be based on a factor of the mean of the distribution, or a percentile of the distribution. With this, all the stochastic driving times are converted back into deterministic ones. However, this results in an approximation where the different trips do not affect each other. Also, the variance of the distribution is only partly accounted for, as we will not encounter the more extreme delays that realistically could still occur. Thus, this approximation may not be very realistic. A better way to deal with stochastic driving times is to work with the expected start- and end-times of a trip given the trips that are driven before it. Unfortunately, it is very time-consuming, if not outright infeasible, to compute these values exactly. This could be solved by estimating these values. An approach to do this, is to use simulations in a local search algorithm [15]. However, the use of simulations is computationally very expensive. Therefore, in their research on parallel machine scheduling, Passage et al. [9] chose to assume normal distributions, which makes calculating the expected start- and end-times a lot simpler and very quick. This also yielded better results compared to the use of simulation, unless we do a lot of simulations each iteration of the local search, which slows down the algorithm.

However, since driving times of trips on the same day are dependent, we focus on the use of simulations inside a local search algorithm, as this will give us a better view on the robustness of our solution. Using simulations in a local search algorithm is computationally expensive. Doing only a few simulations is great runtime-wise, but might not give a correct view on which solution is better. Thus, we minimize the number of simulations, while making sure that we can make a "correct" decision. To do this, we tested several techniques. Namely, Optimal Computation Budget Allocation, Indifference Zones, and a self-developed method based on *t*-tests. For an explanation and comparison of these techniques, we refer to [4].

For the local search algorithm, we extend the simulated annealing approach used by ten Bosch et al. [14], by including robustness and simulations to evaluate solutions in each iteration. Their method is based on a column-generation approach by van Kooten Niekerk et al. [16], where each column represents the schedule of a single vehicle. However, instead of solving a pricing problem to find new columns, they use simulated annealing to find a solution and use the vehicle schedules in this solution as columns. These are then recombined into a final solution by an ILP-solver.

For simulating a schedule, we need to know the distribution of the driving time. To determine these, we worked with Qbuzz, a major bus company from The Netherlands, who provided data and insights of their operations. We looked at historic data and determined the various sources of delays. Furthermore, we also found that the driving times in the simulation depend on each other, with the main idea that people taking the bus in the morning, will also take the bus back in the afternoon. Thus, if we have higher passenger demands in the morning, we will also see these higher passenger demands in the afternoon, likely resulting in higher driving times both in the morning and the afternoon. With this, we determined the distributions to use in our simulation. A detailed overview of this is given in Appendix A.

Our contribution. We present an algorithm that takes into account the relevant sources of delays to increase the robustness and punctuality of a schedule. Hereto, we combine local search with simulation and integer linear programming. The probability distributions are determined by analysing historical driving times and weather data.

The rest of this paper is organized as follows. In Section 2, we will first discuss the relevant literature for this problem. In Section 3, we will describe the problem into more detail. Then in Sections 4 and 5 we will go into the details of our model, where we discuss our local search approach and its extension with simulation. Lastly, we run experiments to test our model in Section 6, which we will discuss in Section 7.

2 Literature Overview

In this section, we discuss some of the literature on the planning of electric vehicles and the use of stochastic driving times. In recent years, there has been an increasing focus on the study of this problem due to its relevance in the context of electric vehicles. Considering the scheduling of electric vehicles can be viewed as scheduling vehicles with resource constraints, taking into account the limitations imposed by their range. In 1983, Raff [12] studied this VSP problem with any resource constraint and show that is NP-hard. In 2007, Wang and Shen [17] expanded this problem, adding fuelling time constraints. Here, they specifically focus on electric vehicles.

In a recent review, Perumal et al. [11] divide the research of this problem into different challenges and different methodologies to overcome these challenges. For the recharging of these electric buses, multiple technologies can be considered [7, 3]. The main technologies considered are battery swapping and the use of recharging stations. For example, the use of battery swapping is studied by Chao and Xiaohong [2]. They solve the resulting problem using a genetic algorithm. There is more focus on the use of recharging stations. Wen et al. [18] present a large neighbourhood search heuristic for solving E-VSP with recharging stations, where they assume the charging time to be linear in the charging volume. However, this assumption of linear charging times is not realistic and, as shown by Olsen and Kliewer [8], may lead to infeasible routes, as not enough time is planned for charging. Van Kooten Niekerk et al. [16] incorporated such non-linear charging times and proposed a column-generation approach to solve E-VSP. Ten Bosch et al. [14] built on this approach, by using simulated annealing to solve the pricing problem.

The use of stochastic driving times in the E-VSP problem is novel. Tang et al. [13] propose a branch-and-price framework for solving E-VSP under both static and dynamic traffic conditions. They do this by using a so-called buffer-distance, which makes sure that the bus does not run out of charge while in traffic. Furthermore, while they propose a model to avoid running out of charge due to the traffic conditions, they still use the average travel time for cost and delay calculations. So, while they look at stochastic driving conditions, they still solve it deterministically. Bie et al. [1] use a Non-dominated Sorting Genetic Algorithm with the elitist strategy (NGSA-II) to solve E-VSP for stochastic driving times. For their recharging strategy, they set a range in which the battery's state of charge is allowed to vary. They recharge a bus when it is idle, i.e. when it is currently waiting for its next trip to start.

3 Problem Description

In the Vehicle Scheduling Problem (VSP), we are given a set of trips $\bar{T}$. These trips consist of a departure and arrival location, a planned starting time, and a driving time. The goal is to schedule a set of identical vehicles such that every trip in $\bar{T}$ is driven. For this, we minimize the costs of using these vehicles. These costs consist of a fixed cost, a cost per kilometer driven, and a cost per *block*. In this case, a *block* is a set of trips driven after each other without going back to the depot. A cost for these blocks is included to penalize situations where a bus only drives a single trip before going back to the depot. For this problem, we consider only one depot location. All vehicles must start and end their route at this location.

As we work with electric vehicles, we get the E-VSP problem. Since electric vehicles have a lower range than their non-electric counterparts, we need to consider how and when to charge these vehicles. When recharging the battery, we take the battery life into consideration This is because certain charging strategies could significantly degrade the battery life, resulting in more maintenance costs. Therefore, we follow the approach of van Kooten Niekerk et al. [16]. They looked at the *Depth-of-Discharge* (DoD), which is a percentage that indicates how much the battery is discharged. Based on the number of charge cycles of a battery and the current DoD, they estimated the cost of charging, given the DoD at the start and the end of charging. We use the exact same cost for our charging sessions. Lastly, we need a charging strategy for these vehicles. For this, we make use of so-called *opportunity charging*. Thus, we charge a bus whenever possible for as long as possible. Note that this charging strategy also minimizes the DoD over the whole trip. Furthermore, charging a vehicle takes time. For calculating this time, we take the same approach as van Kooten Niekerk et al. [16], thus we assume the charging time to consist of two linear parts. Here, we assume that charging from 0% to 80% takes the same time as charging from 80% to 100%.

As we alluded to before, we want the created schedules to be robust against delays. This is why we use stochastic driving times instead of deterministic ones. However, we also need a measure for the robustness of a schedule. For this, we compare the robustness of a given schedule using simulation, where we compare the planned and actual starting times of a trip, because a delayed vehicle will start its next trip late when there is not enough slack between the trips. For this measure, we use a piecewise-linear function, where we penalize being less than 3 minutes late significantly less than being more than 3 minutes late. Doing this over multiple simulations, and taking the mean, gives a good score for the robustness.

Summarizing, we schedule electric buses to perform a set of trips, where we minimize operational costs, a cost for the battery lifetime, and the robustness penalty.

4 The Hybrid Algorithm

As mentioned before, we use the same approach as ten Bosch et al. [14] to solve the E-VSP problem, which we expand with simulations in order to solve stochastic E-VSP. We first look at how they set up their local search. For this, they take a set-covering MIP as a basis. Remember that $\bar{T}$ is the set of trips that need to be driven. Then, let V be the set of all possible vehicle tasks. Here, a vehicle task, from now on task for short, is a set of trips that can be driven by a single vehicle. For a task $v \in V$, we can calculate its cost C_v. This cost is the sum of three components, as described in Section 3.

With this, we can formulate the master problem. We use the variable x_v to denote if a task v is chosen. Furthermore, we have the parameter r_{vt} to denote that a trip $t \in \bar{T}$ is in v. Then our objective is to

$$\textbf{minimize} \sum_{v \in V} x_v C_v. \tag{1}$$

Which is subject to the constraints:

$$\sum_{v \in V} r_{vt} x_v = 1 \qquad \forall t \in \bar{T}, \tag{2}$$

$$x_v \in \{0, 1\} \qquad \forall v \in V. \tag{3}$$

Here, Equation (2) ensures that we drive every trip in the final schedule and Equation (3) sets the domain of our decision variables.

A common way to solve this problem, is to use column generation and find columns by solving a pricing problem. However, ten Bosch et al. [14] have shown a better way to solve this problem. They use simulated annealing to find a set of vehicle tasks. We use the approach. Finally, we include these tasks in the restricted master problem and solve it to find our final solution. However, in our setup we cannot calculate C_v directly, because of the use of stochastic variables. Thus, we need to estimate the cost of a task. We do this by simulating them and taking the average. To calculate the number of simulations that are required each iteration, we look at the results of de Bruin [4]. We use the t-test method they developed, as this seems to be a good compromise between runtime and solution quality.

5 Robustness

In order to simulate a task or a complete schedule, we make use of discrete-event simulation. Each vehicle (or task) in the schedule has multiple subtasks. These subtasks are essentially everything that needs to be driven, thus trips, deadheads, or going to and from the depot. The discrete-event simulation consists of two events: the start and end of a subtask. During this simulation, we also need to keep track of the state of charge in order to calculate the minimum required charging times to make sure that buses do not run out of charge. Thus, we also simulate the energy consumption, which we explain further in Section 5.2.

To integrate this into our simulated annealing algorithm, we perform multiple simulations for a given solution and return the average result. However, when comparing two solutions, we need to make sure that they are compared fairly. Specifically, the randomness of the driving times and energy consumption can cause a worse solution to be "lucky" and outperform the better solution. In order to make comparisons more fair, we employ a technique called *Common Random Numbers* (CRN) [6]. With this technique, we make sure that both solutions get the same realizations of driving times, thus solutions cannot gain an advantage by drawing shorter driving times. However, this technique is not applicable to the energy consumption, which we will explain further in Section 5.2.

5.1 Simulating Driving Times

To simulate the driving times, we first need to find appropriate distributions for them. To do this, we analysed historic driving times that were provided by the bus company Qbuzz. This analysis is available in Appendix A. Here, we also found that driving times within the same day are somewhat dependent on each other. To capture this behaviour in our simulation, we need to create scenarios where, for example, longer driving times in the morning also lead to longer driving times in the evening, allowing us to create days with higher passenger demands that could lead to higher driving times over the whole day. To accomplish this, we generate *instances* of simulated driving times. These contain the simulated driving times of all the trips on a single day.

We create a set of instances for multiple types of situations. First, we have a set of instances for "normal" days. These are days with an average passenger load and mostly average driving times. But, as we want our schedules to be robust against delays, we will also create scenarios for busier days, where we have more passengers and thus more above average driving times. By simulating a mix of these situations, we make our schedules robust against these busy days, while maintaining a good schedule under more normal loads.

To start a simulation, we randomly select one of these instances and simulate the whole schedule with it. Before running the simulations, we also decide how many of each of the instance types we simulate per iteration. This is set beforehand, such that we always run the simulation with the same distribution of instance types. Here, we also ensure that each instance type is accounted for.

5.2 Simulating Energy Consumption

In our simulation, we account for different bus drivers having different driving styles, and thus different energy consumption figures. However, since we do not create a crew schedule, we need to estimate this, as we do not know who is driving when. Therefore, we create three scenarios, namely for 95%, 100%, and 105% of the base energy consumption. Then, before the vehicle pulls out of the depot, we select one of these scenarios randomly. We assume that these driving styles do not have an influence on the driving time. This might not be a completely realistic assumption, but we do not expect the driving style to have a big effect on the driving time.

As drivers need breaks, bus drivers may be swapped along the route. Since we do not create a crew schedule, we use a more high-level model. Here, we allow these driver swaps at the start of every trip. However, to make sure that drivers are not swapped too frequently, a driver needs to drive the bus for at least 2 hours before he is allowed to be swapped. After these 2 hours, we try to swap the drivers as soon as possible. Note that since we employ only three different driving styles, this does not always lead to a change in energy consumption.

Unfortunately, this approach does not allow for CRN to be used on these stochastic energy consumptions. Since we are comparing different bus routes, it is unreasonable to assume that every trip is still driven by the same driver. Furthermore, the actual distance driven could also be significantly different between these routes. This could be solved by using the same driver scenario over the whole solution, but this could lead to unrealistically large energy usage. Another approach would be to select a driver scenario per trip, however this could lead to an excessive number of driver swaps. For these reasons, we will not use CRN for these driver scenarios in our model.

6 Experiments and Results

To test our algorithm, we compare the use of stochastic driving times with using deterministic driving times. These deterministic driving times are given in the input data and form the basis of the stochastic driving times. Note that the deterministic driving times already contain some slack in order to make the schedule more robust. For our input, we use several instances that were provided by Qbuzz. These instances are from various regions of The Netherlands, namely the regions of Dordrecht, Groningen, and Utrecht. A short overview of these instances is provided in Table 1.

We use 15 simulated annealing runs to generate trips for the restricted master problem. These trips are then combined into a final solution using CPLEX version 22.1 as our ILP solver. Note that the simulated annealing can handle both stochastic and deterministic driving times.

In order to have a fair comparison between our final solutions, we simulate these 1 000 times. This is also done for solutions that were created with deterministic driving times. Thus, we can fairly compare the robustness between each method. Here it is important

Table 1 Overview of the used datasets and their parameters.

Dataset	#Trips	#Lines	Battery Capacity (kWh)
`dmg`	631	8	232
`gn345`	463	3	184
`qlink`	590	3	160
`zst`	317	2	232

to note that we do not employ CRN in these simulations. However, due to the number of simulations we use, we do not expect this to result in unfairness, since using a lot of simulations reduces the variance.

6.1 General results

First we compare the lateness, maximum DoD, and number of vehicles used. Averages of these statistics over multiple solutions are shown in Table 2. In this table, we use L to denote the set containing the lateness values for each trip. We define the lateness as the difference between the planned starting time of a trip and the earliest time a bus could depart for this trip. This can be negative, and a positive value means the trip started late. Furthermore, these values are in minutes. This means that $\bar{L}$ denotes the mean lateness of all the trips, and we use L_{95} to denote the 95th percentile of the lateness. We define the punctuality to be the percentage of trips that started on time. Lastly, the column "Mean Late" denotes the mean of the set $\{x \in L \mid x > 0\}$. Thus, it is the average number of minutes a bus starts late, given that it starts late. From these results, we see some reductions in the lateness of a trip, and also a reduction in the "Mean Late" statistic. However, this sometimes comes at the cost of having to use more buses. We checked these results with Qbuzz, confirming that these lateness values are similar to what they encounter in practice.

Table 2 Various statistics regarding the final solutions calculated with either deterministic or stochastic driving times.

		#Vehicles	$\bar{L}$	L_{95}	Mean Late	Punctuality	Max DoD
Dataset	Driving times						
`dmg`	Deterministic	45.0	-3.8	2.0	2.9	89.5%	53.2%
	Stochastic	52.2	-4.5	1.0	2.0	92.7%	58.3%
`gn345`	Deterministic	92.0	-2.3	1.2	4.7	93.4%	95.1%
	Stochastic	94.0	-2.9	1.0	3.3	94.7%	95.2%
`qlink`	Deterministic	37.8	0.8	5.0	5.1	83.2%	52.9%
	Stochastic	45.8	-3.0	1.6	2.7	91.8%	62.3%
`zst`	Deterministic	38.6	-5.9	0.0	2.4	98.4%	69.8%
	Stochastic	37.4	-6.2	0.0	2.2	98.5%	70.0%

6.2 Recombination

We also compare our simulated annealing runs with the results from the recombination. For this, we compare the scores of the schedules. We use 15 simulated annealing runs for the recombination. During our simulated annealing, we keep multiple of our best solutions, which are used for the recombination, i.e. to be included in the MIP (of restricted master problem). From a certain point in our simulated annealing we will collect the new best solutions, however after collecting a new best solution, we wait a few iterations before collecting the next one. This is done in order to not collect solutions that just differ in one neighbour. This means that we collect about 20 to 40 solutions per simulated annealing run, which results in about 7 000 to 56 000 columns depending on the dataset that is used. Recall, that one simulated annealing solution is a complete schedule resulting in multiple vehicle tasks, and hence multiple columns. Lastly, we set the time limit of the ILP to 20 minutes in order to reduce the total computation time.

We show the average result for different statistics in Table 3. Note that the runtime consists of both the recombination and the score calculation of the columns, which is why some instances report a time that is above 20 minutes. Also, the "Improvement" denotes the percentage improvement compared to the best simulated annealing score, where a negative value means that the recombination did not improve compared to the simulated annealing. We do not give the ILP solver an initial solution. Thus, the solver will not necessarily come up with a solution that is better than simulated annealing in the given timeframe, which is why we see these datasets run into the time limit of 20 minutes. Furthermore, these are also the only tests with fairly big integrality gaps, and they do not show an improvement compared to the simulated annealing. However, the results on the other tests are quite promising, as they show 1 to 3 percent improvements compared to simulated annealing, which is a big improvement cost-wise.

Table 3 Various statistics regarding the performance of the recombination. Here, "Gap" denotes the gap to the LP relaxation, and "Improvement" denotes the percentage improvement compared to the best simulated annealing score.

		#Columns	Gap	Improvement	Time (s)
Dataset	Driving Times				
`dmg`	Deterministic	20 523.7	5.889%	−2.377%	1 200.41
	Stochastic	27 566.7	7.411%	−3.425%	1 251.85
`gn345`	Deterministic	33 320.8	0.013%	1.545%	467.80
	Stochastic	55 770.7	0.038%	2.372%	760.99
`qlink`	Deterministic	21 931.6	0.005%	3.645%	18.70
	Stochastic	25 805.3	4.672%	−1.237%	1 238.82
`zst`	Deterministic	7 037.4	0.009%	2.713%	74.39
	Stochastic	23 271.2	0.010%	3.206%	245.42

6.3 Lateness

To better understand the robustness of our solutions, we first look at the histogram of the lateness values (the set L) we encountered in our simulations. This is displayed in Figure 1. There are a few things to notice. First, we see that for most trips the bus is early by about 2 minutes or less, which is normal and expected behaviour. We also see some peaks at -10 and -15 minutes. This is especially clear in the `qlink` dataset. These peaks correspond to the frequency of some lines in these datasets. We suspect that these peaks are due to the dataset not containing many lines, thus the only way to increase robustness is to keep a bus reserve at the starting location of the trip. One way to do this, is to arrive just when the next bus departs, essentially arriving 1 trip early. Furthermore, for every dataset, there is also a big peak at 0 minutes. This is partly due to the buses charging until their trip starts, but also due to tight planning. Comparing the lateness between deterministic and stochastic driving times, we see that in case of stochastic driving times, a bus generally arrives earlier, which is in line with our other results.

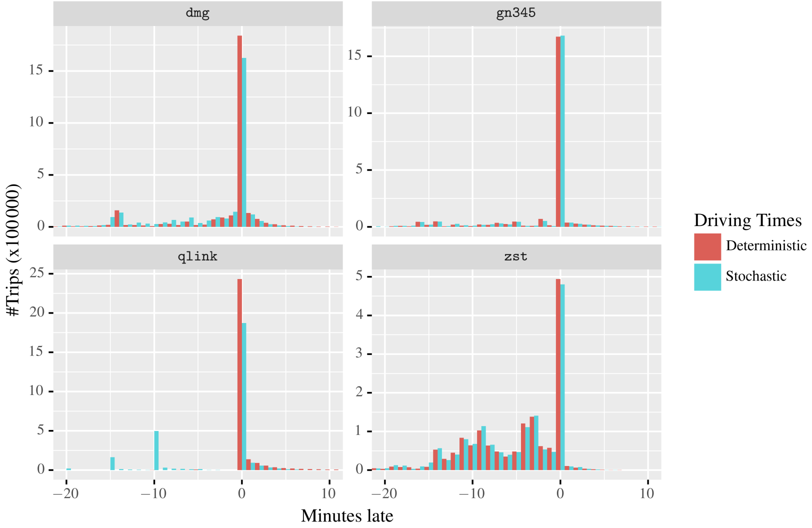

Figure 1 Histogram of the set L.

We ran our algorithm with different penalty factors for the lateness, in order to get a better overview of how the stochastic driving times compare to the deterministic driving times. Thus, we verify what happens when we change the importance of the lateness factor in the solution score. For this, we look at both the punctuality and the mean lateness.

First, punctuality. We compare this with the operating costs and the number of vehicles used. These comparisons are shown in Figures 2 and 3 respectively. In these figures we see that using stochastic driving times, generally leads to solutions with a better punctuality, but they use more vehicles. This is also reflected in Figure 2, where we see higher operating costs for the same punctuality. Note that the operating cost also contains a time component. Thus, buses that need to wait for their trip to start increase the operating cost. The stochastic model includes more slack in the schedule, and not having enough slack, or waiting time, might result in lateness penalties.

We also compare the lateness itself to the operating cost and the number of vehicles used. For the lateness, we use the mean minutes late statistic ($\bar{L}$). The results of these comparisons are shown in Figures 4 and 5 respectively. These figures show results that are similar to the punctuality, where we decrease the average lateness for a bit more vehicle usage, which is reflected in the operating costs.

6.4 Depth of Discharge

We also ran our algorithm with different penalty factors for the DoD, to see what happens when we prioritize battery costs more. In Figure 6, we see the maximum DoD for different number of vehicles used in the final solution. For most of the datasets, we see very similar results between the use of stochastic and deterministic driving times. Meaning that the maximum DoD is more or less the same for both deterministic and stochastic driving times.

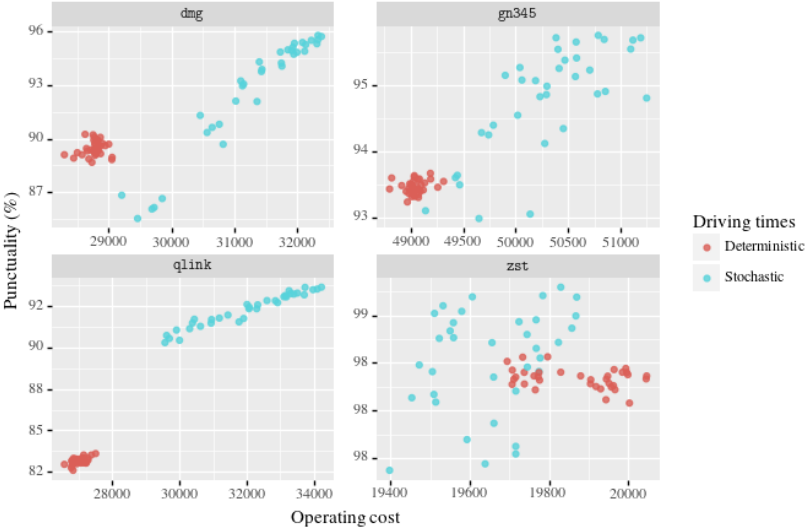

Figure 2 Punctuality compared to the operating cost.

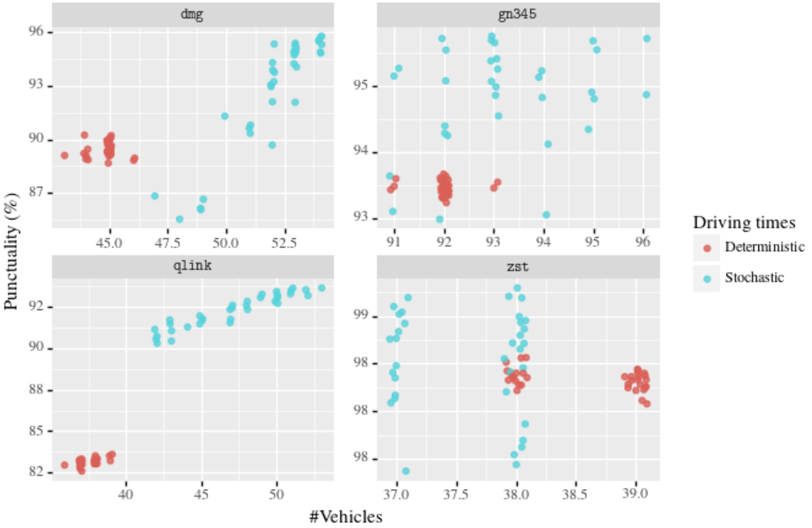

Figure 3 Punctuality compared to the number of vehicles used.

The main difference here is the number of vehicles a solution requires. This seems intuitive, as a higher DoD means that a bus can drive longer without recharging. Hence, fewer vehicles are required in order to drive all trips, but this have a negative effect on the lifetime of the battery.

7 Conclusion

In this paper, we show a model to solve E-VSP with stochastic driving times. For this, we used a hybrid algorithm including a local search approach in the form of simulated annealing and a set covering ILP. We extended an existing simulated annealing approach for the case of deterministic driving times by including robustness and simulations, such that we could use it with stochastic driving times.

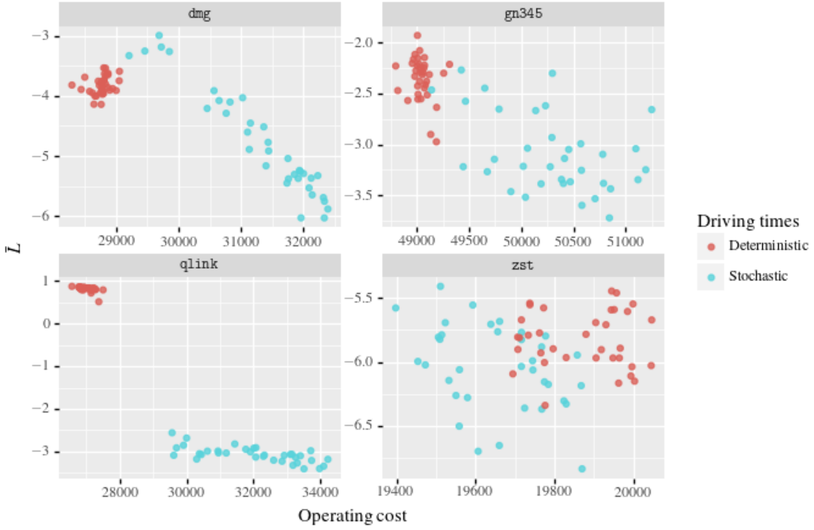

Figure 4 Mean minutes late compared to the operating cost.

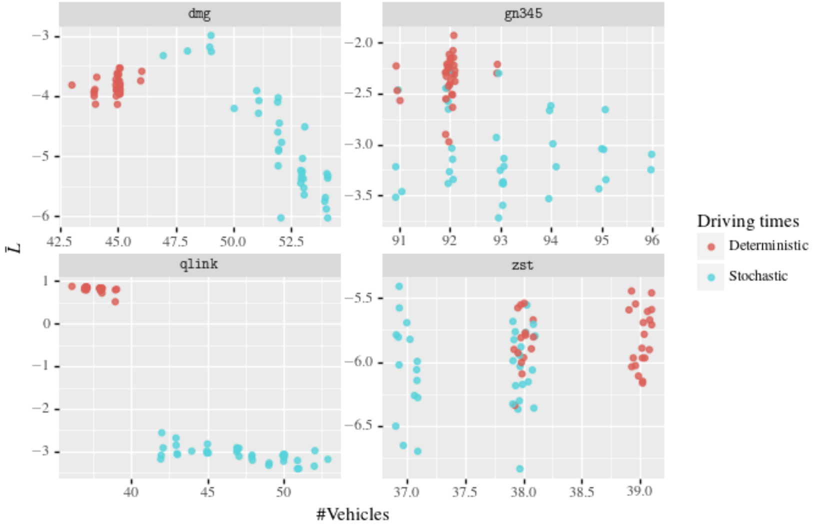

Figure 5 Mean minutes late compared to the number of vehicles used.

We compared the robustness by using stochastic variables in Table 2 and Figure 1. Here we see a decrease in lateness compared to using deterministic driving times. This is not only true for the average lateness, but more importantly also for the worse cases. That is, we see reduction of the 95th percentile of the lateness. From this, we can conclude that the use of these stochastic driving times indeed increases robustness of our schedules. However, this comes at a small cost. In general, these solutions could require about the same number of vehicles, although on average solutions created with stochastic driving times require slightly more vehicles. Furthermore, the operating cost of these solutions is also higher. This is due to the additional vehicles required, and the extra waiting time that is sometimes needed.

From the results of the combination of different simulated annealing runs, we see improvements on the solution quality of up to 3%. However, for some of our experiments the ILP ran into the time limit of 20 minutes, where it did not find any improvements compared to

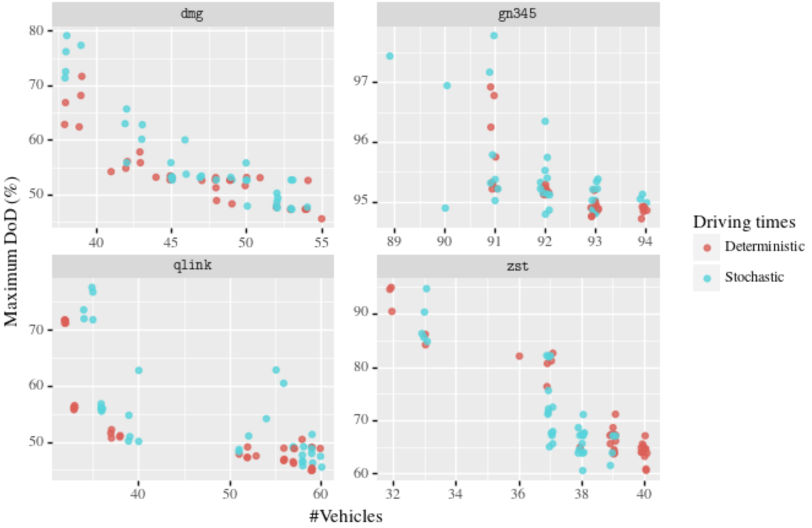

Figure 6 Maximum DoD for different number of vehicles used.

the simulated annealing. This is especially visible in the integrality gaps reported in Table 3, where the experiments that ran into the time limit have a fairly big gap compared to the experiments that did not run into this limit. Given more time, these instances should find a solution that is at least as good as the solution found by the simulated annealing. For the other instances, we can improve our best solution quite quickly. Thus, although it is not always improving our best result, this extra step seems a good addition to our simulated annealing.

7.1 Future Research

We showed that our model for stochastic E-VSP is quite successful in creating more robust schedules. This approach could be further enhanced to increase the applicability of our results. One of the assumptions we made for our stochastic driving times is that we use the same distribution for every line in every direction. This is not necessarily realistic, as there are lines that solely cross city centers, but also lines that cover longer distances. Buses on these lines encounter different traffic conditions and stopping patterns, and thus they could end up with different distributions for their driving time. Our work could be extended to include a distinction between these different line types, which would require more research into how these distinctions should be made and also the distributions that are required.

The model itself could also be further enhanced. Currently, we make use of opportunity charging and do not look at the price of electricity. Hence, future extensions could look at making charging plans, taking these prices into account.

References

1 Yiming Bie, Jinhua Ji, Xiangyu Wang, and Xiaobo Qu. Optimization of electric bus scheduling considering stochastic volatilities in trip travel time and energy consumption. *Computer-Aided Civil and Infrastructure Engineering*, 36(12):1530–1548, 2021. doi:10.1111/mice.12684.

2 Zhu Chao and Chen Xiaohong. Optimizing battery electric bus transit vehicle scheduling with battery exchanging: Model and case study. *Procedia - Social and Behavioral Sciences*, 96:2725–2736, 2013. doi:10.1016/j.sbspro.2013.08.306.

3 Zhibin Chen, Yafeng Yin, and Ziqi Song. A cost-competitiveness analysis of charging infrastructure for electric bus operations. *Transportation Research Part C: Emerging Technologies*, 93:351–366, 2018. doi:10.1016/j.trc.2018.06.006.

4 Philip de Bruin. Scheduling electric buses with stochastic driving times. mathesis, Utrecht University, 2022. URL: https://studenttheses.uu.nl/handle/20.500.12932/41969.

5 KNMI. Hourly weather station readings. https://daggegevens.knmi.nl/klimatologie/uurgegevens. Accessed: 2022-05-24.

6 Averill M. Law. *Simulation Modeling and Analysis*. McGraw-Hill, 5th edition, 2015.

7 Jing-Quan Li. Battery-electric transit bus developments and operations: A review. *International Journal of Sustainable Transportation*, 10(3):157–169, 2016. doi:10.1080/15568318.2013.872737.

8 Nils Olsen and Natalia Kliewer. Scheduling electric buses in public transport: Modeling of the charging process and analysis of assumptions. *Logistics Research*, 13(1):4, 2020. doi:10.23773/2020_4.

9 G. J. P. N. Passage, J. M. van den Akker, and J. A. Hoogeveen. Local search for stochastic parallel machine scheduling: improving performance by estimating the makespan. In *European Conference on Stochastic Optimization*, 2017.

10 Jayakrishna Patnaik, Steven Chien, and Athanassios Bladikas. Estimation of bus arrival times using APC data. *Journal of Public Transportation*, 7(1):1–20, 2004. doi:10.5038/2375-0901.7.1.1.

11 Shyam S. G. Perumal, Richard M. Lusby, and Jesper Larsen. Electric bus planning & scheduling: A review of related problems and methodologies. *European Journal of Operational Research*, 301(2):395–413, 2022. doi:10.1016/j.ejor.2021.10.058.

12 Samuel J. Raff. Routing and scheduling of vehicles and crews : The state of the art. *Computers & Operations Research*, 10(2):63–67, 1983. doi:10.1016/0305-0548(83)90030-8.

13 Xindi Tang, Xi Lin, and Fang He. Robust scheduling strategies of electric buses under stochastic traffic conditions. *Transportation Research Part C: Emerging Technologies*, 105:163–182, 2019. doi:10.1016/j.trc.2019.05.032.

14 W. ten Bosch, J. A. Hoogeveen, and M. E. van Kooten Niekerk. Scheduling electric vehicles by simulated annealing with recombination through ILP. Submitted for publication, 2021.

15 Marjan van den Akker, Kevin van Blokland, and Han Hoogeveen. Finding robust solutions for the stochastic job shop scheduling problem by including simulation in local search. In Vincenzo Bonifaci, Camil Demetrescu, and Alberto Marchetti-Spaccamela, editors, *Experimental Algorithms, 12th International Symposium, SEA 2013, Rome, Italy, June 5-7, 2013. Proceedings*, volume 7933 of *Lecture Notes in Computer Science*, pages 402–413. Springer, 2013. doi:10.1007/978-3-642-38527-8_35.

16 Marcel E. van Kooten Niekerk, J. M. van den Akker, and J. A. Hoogeveen. Scheduling electric vehicles. *Public Transport*, 9(1-2):155–176, 2017. doi:10.1007/s12469-017-0164-0.

17 Haixing Wang and Jinsheng Shen. Heuristic approaches for solving transit vehicle scheduling problem with route and fueling time constraints. *Applied Mathematics and Computation*, 190(2):1237–1249, 2007. doi:10.1016/j.amc.2007.02.141.

18 M. Wen, E. Linde, S. Ropke, P. Mirchandani, and A. Larsen. An adaptive large neighborhood search heuristic for the electric vehicle scheduling problem. *Computers & Operations Research*, 76:73–83, 2016. doi:10.1016/j.cor.2016.06.013.

A Driving Time Analysis

In order to find good distributions for the driving times, we analysed historic driving times. This data is mainly from the region of Dordrecht, The Netherlands. This data was provided by Qbuzz, the bus company that serves this region. We looked at total time of trips driven in this region during 2019. It contains the information about the delay at the start of a trip, the planned driving time of the trip, the actual driving time of the trip, the dwell time during the trip, and the driving distance. Note that the dwell time is the time a bus stands still at a stop. Furthermore, it contains 27 different routes with an average length of 11.8 km and on average 22 stops. In total, this dataset contains 77 937 trips.

To analyse these driving times, we will first remove some outliers from the dataset. For this, we require the dwell time to be non-negative and not bigger than the total driving time, as values outside this range are simply not possible. Furthermore, we look at the average speed of the bus. This has to be between 0 and 80 kilometers per hour. Lastly, we also filter trips based on their delay at the start of the trip. We observed some trips to start exactly 1 hour before or after their planned time, suggesting an error in linking the bus with the exact trip they drove. Thus, we filter trips based on the z-score of their delay at the start. The z-score is the number of standard deviations by which this value is above or below the mean of the observed values. In this case, we remove trips where the z-score of the delay at the start is bigger than 2.5. After filtering, our dataset contains 76 485 different trips.

A.1 Variables

To create distributions for the driving times, we first investigate different sources for variation in the driving times. For this, we look into the time of day, the weather conditions, and also the effect of the dwell time.

A.1.1 Time of Day

One source of variation in the driving times is the time of day. Traffic conditions vary over the day, where mornings and afternoons are usually more busy due to people commuting to work or back home. For the same reasons, we also expect there to be more passengers, thereby increasing the dwell time and thus the total driving time. These variations are already accounted for in the bus schedule, as illustrated in Figure 7, where we observe higher planned driving times in the morning and late afternoon.

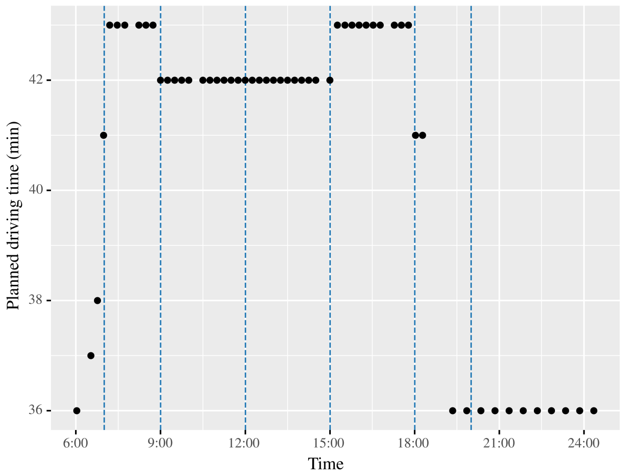

Figure 7 Example of planned driving times over a single day. The blue lines indicate the time periods defined in Table 4.

Looking at the full data, we extract the average driving time as a percentage of the planned driving time. This is plotted in Figure 8, where we grouped each trip by the hour it departs in. In this figure, we do not see big differences in these percentages over the whole day. However, we still create different distributions for different periods of the day. We base this division on the work of Patnaik et al. [10] and the planned driving times. For our simulation, we use the time periods defined in Table 4. These time periods are also indicated by the blue lines in Figures 7 and 8. These do largely correspond to the time periods used by Qbuzz for, for example, their deadhead driving time calculations. The main difference being that we define more time periods.

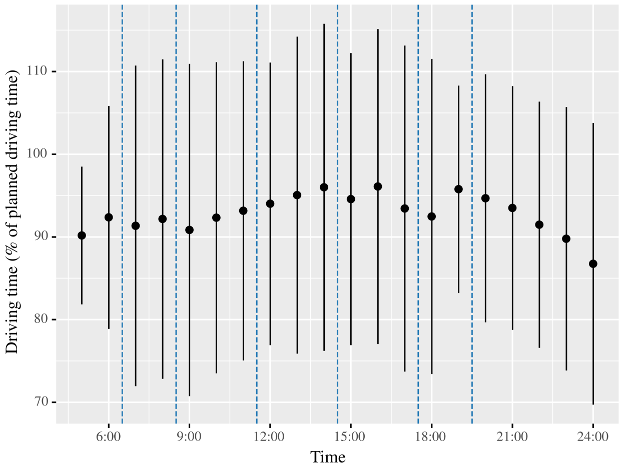

Figure 8 Average driving time plus/minus two times its standard deviation as a percentage of the planned driving over a whole day. The blue lines indicate the time periods defined in Table 4.

Table 4 Time periods used.

Time Period	Description
Early Morning	4:00 till 6:59
Morning Peak	7:00 till 8:59
Late Morning	9:00 till 11:59
Early Afternoon	12:00 till 14:59
Afternoon Peak	15:00 till 17:59
Evening	18:00 till 19:59
Late night	20:00 and later

A.1.2 Weather

Another variable we investigated is the effect of the weather on the driving times. We expect the driving times to be higher on days with bad weather. The reasoning behind this is that we expect more people to take either public transport or go by car, thus increasing driving times due to traffic conditions and higher passenger loads.

To test this hypothesis, we used the hourly weather data of 2019 made publicly available by the KNMI [5]. For this, we used the readings from the weather station in Rotterdam, which is closest to Dordrecht. We use information about the duration of rainfall (DR) and the total amount of rainfall (RH) during the timeblock of an hour. For every trip, we calculate the duration and total amount of rainfall during the day the trip took place, the morning of the day the trip took place, and the hour in which the trip departed. For this, we define rain during the morning to be any rain that falls between 6:00 and 9:00, while rain during the day is defined as any rain that falls between 6:00 and 20:00. We performed correlation tests on these variables and the driving time, using Pearson's correlation coefficient. These coefficients are shown in Figure 9. Unfortunately, these tests indicate no relationship between the driving time and various variables indicating rainfall.

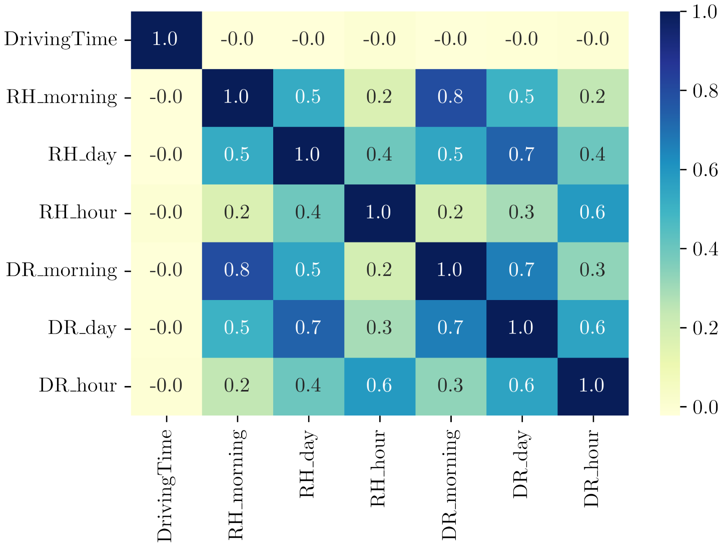

Figure 9 Pearson's correlation coefficients between various rainfall parameters and the driving time.

To see why this is the case, we looked at the driving times under various rain conditions. We looked at the average rain intensity in millimeters per hour. This is done for both for the whole day (excluding the night) and within a certain hour. We use the rain intensity as this would be the most accurate classifier within the available data. Another factor that could be taken into consideration is, for example, the size of the rain droplets. However, we do not have data for that and this is usually not reported in the weather reports, so we do not expect this to be a major factor when people decide how they travel.

We classify an average rain intensity of 3 mm/h or less to be light rain, and higher values are classified as rain. The driving times under these conditions are shown in Figure 10. This indicates that there are not always significant differences between rain or no rain. We also note that the amount of rain does not predict the driving time very well, as the figure shows that higher intensities of rain sometimes lead to lower driving times than when there is no rain.

From this we conclude that we cannot use these weather patterns in our simulations, because it remains unclear how they influence the driving times. We saw that in some scenarios there do not seem to be significant differences, and also that heavier rain did not necessarily lead to higher driving times. This could be due to passenger behaviour, where for some weather conditions people go by bus rather than by bike, while for other weather conditions people just stay at home. We could not verify this behaviour as we do not have access to passenger data for this route. Thus, we do not include these weather patterns in our simulations, since we can not draw conclusions from our current data.

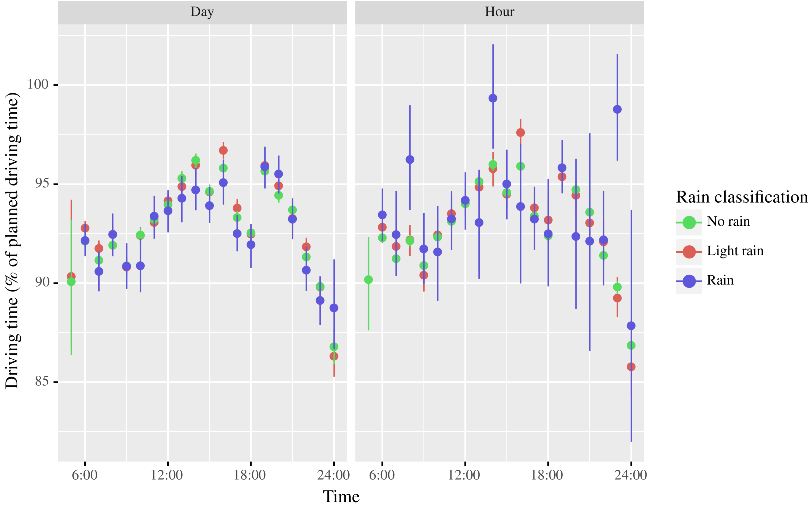

Figure 10 Mean driving times with their 95% confidence interval under different rain conditions during the day. Here, light rain has an average rain intensity of 3 mm/h or less, and more than 3 mm/h is classified as rain.

A.1.3 Number of Passengers

The last variable we looked at is the effect of passenger numbers on the driving times. While we do not have exact passenger data, we do have information about the dwell times, which gives an indication of how busy a trip is, since more people moving in or out of the bus leads to longer dwell times. The dwell time could be a significant part of the total driving time, thus we have to understand its influence.

To get a better understanding of how the driving times are influenced and by how much, we group the dwell times into three categories. For each line, we calculate the 70th and 90th percentiles of the dwell time and use these to categorize the dwell time of a specific trip. Then we create three groups with driving times. Group 1 contains driving times, where the dwell time is below the 70th percentile of the dwell time of that trip. Group 2 contains driving times, where the dwell time is above the 70th percentile and below the 90th percentile. Lastly, group 3 contains the remaining driving times. Grouping on these categories gives us insight into the mean and standard deviation of these driving times. These are shown in Table 5.

Table 5 Mean and standard deviation of the driving time (as percentage of the planned driving time) grouped by the dwell time category.

	Driving time (% of planned driving time)		
	Mean	Standard deviation	#Trips
Group 1	91.77	9.23	53 519
Group 2	95.72	7.54	15 285
Group 3	99.45	8.62	7 681

From this table we can already see some differences between the driving times with the different dwell times. To confirm that these differences are also significant, we performed Welch's unequal variances *t*-test. Our null-hypothesis in these tests is: "The means of the driving times from the two tested dwell time categories are equal." For these tests, we will use $\alpha = 0.005$. This is lower than the usual 0.05, because we perform multiple t-tests. The p-values for these tests are shown in Table 6. Note that some of these values are 0.0, meaning that they are too small to be represented by a 64-bit floating point number. All these p-values are lower than our chosen α, thus the means of the driving times in these categories are significantly different. Note these p-values seem exceptionally low, which is due to the number of trips in each category.

Table 6 p-Values of Welch's unequal variances t-test we performed on the driving times in the different dwell time categories.

	Group 1	Group 2	Group 3
Group 1	–	0.0	0.0
Group 2	0.0	–	$7.69 \cdot 10^{-219}$
Group 3	0.0	$7.69 \cdot 10^{-219}$	–

For our implementation, it is important to know if there are any patterns in which these higher dwell times happen. For example, are there certain days on which most of the trips encounter higher dwell times? We mainly looked at patterns over a whole day, as our simulation model generates driving times for a whole day. We found that these higher dwell times could occur during the whole day. However, we could not find patterns in this. Hence, we make sure to generate higher driving times over the whole day.

A.2 Distributions

Now that we know which variable to account for, we can fit distributions on the historical data. To do this, we fit different distributions on the driving times for trips departing in the time periods defined in Table 4. Based on Appendix A.1.3, we only use driving times with a dwell time that is less than the 70th percentile of the dwell time for that trip. This is to create a baseline distribution, that is not influenced by the more busy days. Then, in our simulation model, we set a probability to generate driving times for a busy day, in which case all simulated driving times are multiplied by a set factor. We base these factors on the results shown in Table 5. Thus, with a 20% probability we will generate driving times that are 5% higher and with a 10% probability we will generate driving times that are 10% higher.

We fitted normal distributions for the driving times in each period. These fits are shown in Figure 11. For some time periods, we used a single normal distribution to fit the data to, but for others we used a combination of two normal distributions to create a better fit. Thus, these distributions are a mixture of the distributions $N(\mu_1, \sigma_1^2)$ and $N(\mu_2, \sigma_2^2)$ with the weights p and $1 - p$ respectively. The parameters we use for these distributions are given in Table 7.

In our simulation, we only use these distributions to generate the driving times of trips, which means that deadheads and trips to and from the depot use deterministic driving times. This is because we only have data on the planned driving times. However, these driving times vary less in general, since they are not influenced by passenger loads. These deterministic driving times still vary over the day to account for different traffic conditions; we vary these according to specified time periods given in our input data.

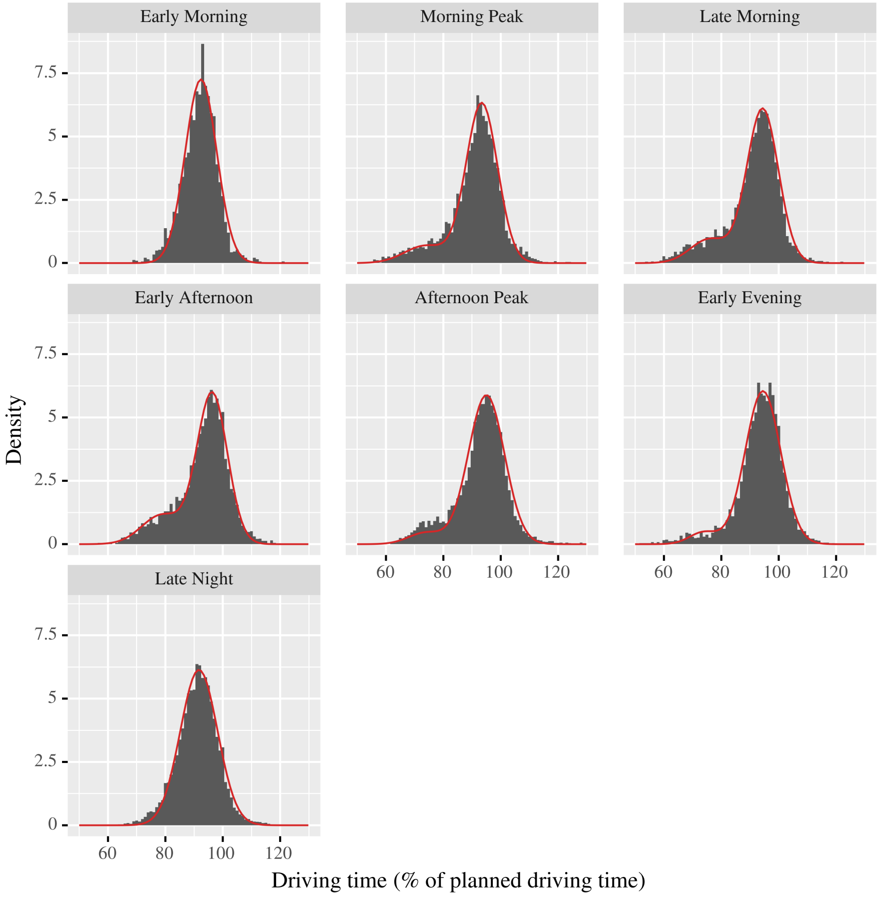

Figure 11 Histograms and the fitted probability density function of the driving time distribution for each time period. Here, the density is the probability of a certain driving time occurring.

Table 7 Parameters of the fitted driving time distributions.

	p	μ_1	σ_1	μ_2	σ_2
Early Morning	1.00	0.924	0.055		
Morning Peak	0.87	0.934	0.055	0.740	0.075
Late Morning	0.84	0.944	0.055	0.760	0.067
Early Afternoon	0.79	0.963	0.053	0.790	0.072
Afternoon Peak	0.93	0.950	0.063	0.740	0.061
Early Evening	0.94	0.945	0.062	0.740	0.050
Late Night	1.00	0.917	0.065		